UNIVERSIDADE DO ALGARVE

ESTIMAÇÃO EM PEQUENOS DOMÍNIOS COM MODELOS ESPACIOTEMPORAIS DE NÍVEL ÁREA

APÊNDICES

LUÍS MIGUEL SOARES NOBRE DE NORONHA E PEREIRA

DOUTORAMENTO EM
MÉTODOS QUANTITATIVOS APLICADOS À
ECONOMIA E À GESTÃO
NA ESPECIALIDADE DE ESTATÍSTICA

2009

ÍNDICE DE APÊNDICES

Apêndice 1 – Variáveis de estratificação

A1.1 Valor de Volume de Negócios (em Euros)

Escalão	Descrição		
1	4,99	-	249.398,95
2	249.403,94	-	748.196,85
3	748.201,83	-	2.493.989,49
4	2.493.994,47	-	7.481.968,46
5	7.481.973,44	-	14.963.936,91
6	14.963.941,90	-	24.939.894,85
7	24.939.899,84	-	34.915.852,79
8	34.915.857,78	-	74.819.684,56
9	74.819.689,55	-	124.699.474,27

Fonte: Adaptado de INE (2001*b*).

A1.2 NUTSIII

NUTSIII	Código	Designação
111	1	Minho-Lima
112	2	Cávado
113	3	Ave
114	4	Grande Porto
115	5	Tâmega
116	6	Entre Douro e Vouga
117	7	Douro
118	8	Alto Trás-os-Montes
161	9	Baixo Vouga
162	10	Baixo Mondego
163	11	Pinhal Litoral
164	12	Pinhal Interior Norte
165	13	Dão-Lafões
166	14	Pinhal Interior Sul
167	15	Serra da Estrela
168	16	Beira Interior Norte
169	17	Beira Interior Sul
16A	18	Cova da Beira
16B	19	Oeste
171	20	Grande Lisboa
172	21	Península de Setúbal
16C	22	Médio Tejo
185	23	Lezíria do Tejo
181	24	Alentejo Litoral
182	25	Alto Alentejo
183	26	Alentejo Central
184	27	Baixo Alentejo
150	28	Algarve

Fonte: Assembleia da República (2002).

A1.3 Concelhos da Área Metropolitana de Lisboa

Concelho	Designação
1105	Cascais
1106	Lisboa
1107	Loures
1109	Mafra
1110	Oeiras
1111	Sintra
1114	Vila Franca de Xira
1115	Amadora
1116	Odivelas
1502	Alcochete
1503	Almada
1504	Barreiro
1506	Moita
1507	Montijo
1508	Palmela
1510	Seixal
1511	Sesimbra
1512	Setúbal

Fonte: Assembleia da República (2002).

A1.4 Concelhos da Área Metropolitana do Porto

Concelho	Designação
0107	Espinho
1304	Gondomar
1306	Maia
1308	Matosinhos
1312	Porto
1313	Póvoa de Varzim
1315	Valongo
1316	Vila do Conde
1317	Vila Nova de Gaia

Fonte: Assembleia da República (2002).

Apêndice 2 – Dimensões populacional e amostrais reais

A2.1 Dimensões populacional e amostrais e taxas de sondagem em cada estrato ao longo de sete trimestres

Estrato	Dimensão Populacional	Dimensão Amostral							Taxa de Sondagem						
		1º T	2º T	3º T	4º T	5º T	6º T	7º T	1º T	2º T	3º T	4º T	5º T	6º T	7º T
001	38	7	8	8	6	5	7	9	18,4%	21,1%	21,1%	15,8%	13,2%	18,4%	23,7%
002	5	1	1	2	1	0	2	2	20,0%	20,0%	40,0%	20,0%	0,0%	40,0%	40,0%
003	2	1	1	1	1	0	1	1	50,0%	50,0%	50,0%	50,0%	0,0%	50,0%	50,0%
004	2	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
005	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
006	82	9	8	10	7	5	4	7	11,0%	9,8%	12,2%	8,5%	6,1%	4,9%	8,5%
007	15	4	3	3	2	3	3	5	26,7%	20,0%	20,0%	13,3%	20,0%	20,0%	33,3%
008	14	2	2	2	3	1	1	1	14,3%	14,3%	14,3%	21,4%	7,1%	7,1%	7,1%
009	3	0	0	0	0	0	0	1	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	33,3%
010	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
011	49	6	6	6	5	5	5	5	12,2%	12,2%	12,2%	10,2%	10,2%	10,2%	10,2%
012	9	4	4	5	5	4	3	4	44,4%	44,4%	55,6%	55,6%	44,4%	33,3%	44,4%
013	2	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
014	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
015	4	1	0	0	0	0	0	0	25,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
016	3	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
017	9	0	1	2	0	1	0	1	0,0%	11,1%	22,2%	0,0%	11,1%	0,0%	11,1%
018	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
019	52	3	6	5	5	4	4	3	5,8%	11,5%	9,6%	9,6%	7,7%	7,7%	5,8%
020	9	3	2	3	2	4	4	3	33,3%	22,2%	33,3%	22,2%	44,4%	44,4%	33,3%
021	4	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
022	69	6	6	5	5	7	7	7	8,7%	8,7%	7,2%	7,2%	10,1%	10,1%	10,1%
023	9	2	2	2	2	2	2	1	22,2%	22,2%	22,2%	22,2%	22,2%	22,2%	11,1%
024	7	2	2	2	2	3	3	3	28,6%	28,6%	28,6%	28,6%	42,9%	42,9%	42,9%
025	2	1	1	1	0	0	0	0	50,0%	50,0%	50,0%	0,0%	0,0%	0,0%	0,0%
026	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
027	68	3	5	6	4	3	2	2	4,4%	7,4%	8,8%	5,9%	4,4%	2,9%	2,9%
028	13	2	2	2	1	1	2	1	15,4%	15,4%	15,4%	7,7%	7,7%	15,4%	7,7%
029	10	1	1	2	2	1	2	2	10,0%	10,0%	20,0%	20,0%	10,0%	20,0%	20,0%
030	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
031	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
032	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
033	175	8	5	7	6	7	5	7	4,6%	2,9%	4,0%	3,4%	4,0%	2,9%	4,0%
034	46	7	9	9	10	9	7	9	15,2%	19,6%	19,6%	21,7%	19,6%	15,2%	19,6%
035	32	4	3	4	3	3	4	3	12,5%	9,4%	12,5%	9,4%	9,4%	12,5%	9,4%
036	5	1	1	1	1	1	1	1	20,0%	20,0%	20,0%	20,0%	20,0%	20,0%	20,0%
037	1	1	1	1	1	1	1	1	100%	100%	100%	100%	100%	100%	100%
038	21	4	6	5	4	4	7	4	19,0%	28,6%	23,8%	19,0%	19,0%	33,3%	19,0%
039	4	3	2	3	1	2	3	3	75,0%	50,0%	75,0%	25,0%	50,0%	75,0%	75,0%
040	32	5	5	3	5	5	5	5	15,6%	15,6%	9,4%	15,6%	15,6%	15,6%	15,6%
041	2	1	1	1	1	1	1	1	50,0%	50,0%	50,0%	50,0%	50,0%	50,0%	50,0%
042	2	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
043	13	1	3	2	2	4	3	5	7,7%	23,1%	15,4%	15,4%	30,8%	23,1%	38,5%
044	2	2	2	2	2	0	1	0	100%	100%	100%	100%	0,0%	50,0%	0,0%
045	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
046	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
047	139	10	14	13	7	8	5	5	7,2%	10,1%	9,4%	5,0%	5,8%	3,6%	3,6%
048	17	5	4	4	4	4	6	4	29,4%	23,5%	23,5%	23,5%	23,5%	35,3%	23,5%
049	7	2	3	2	3	2	2	2	28,6%	42,9%	28,6%	42,9%	28,6%	28,6%	28,6%
050	4	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
051	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
052	2	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
053	34	3	4	4	2	1	3	2	8,8%	11,8%	11,8%	5,9%	2,9%	8,8%	5,9%
054	4	2	1	1	2	2	2	2	50,0%	25,0%	25,0%	50,0%	50,0%	50,0%	50,0%
055	2	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%

(continua na página seguinte)

(continuação da tabela A2.1)

Estrato	Dimensão Populacional	Dimensão Amostral							Taxa de Sondagem						
		1º T	2º T	3º T	4º T	5º T	6º T	7º T	1º T	2º T	3º T	4º T	5º T	6º T	7º T
056	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
057	4	1	1	0	0	0	0	0	25,0%	25,0%	0,0%	0,0%	0,0%	0,0%	0,0%
058	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
059	2	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
060	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
061	2	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
062	50	8	5	4	4	5	9	6	16,0%	10,0%	8,0%	8,0%	10,0%	18,0%	12,0%
063	7	1	1	1	0	0	1	1	14,3%	14,3%	14,3%	0,0%	0,0%	14,3%	14,3%
064	4	1	1	1	1	1	1	1	25,0%	25,0%	25,0%	25,0%	25,0%	25,0%	25,0%
065	2	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
066	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
067	15	1	0	2	0	3	2	1	6,7%	0,0%	13,3%	0,0%	20,0%	13,3%	6,7%
068	1	1	1	1	1	1	1	0	100%	100%	100%	100%	100%	100%	0,0%
069	3	0	1	0	1	0	0	0	0,0%	33,3%	0,0%	33,3%	0,0%	0,0%	0,0%
070	2	1	0	0	0	0	0	0	50,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
071	18	5	5	2	3	4	4	5	27,8%	27,8%	11,1%	16,7%	22,2%	22,2%	27,8%
072	2	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
073	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
074	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
075	7	1	0	0	0	0	0	0	14,3%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
076	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
077	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
078	107	8	8	8	5	7	8	6	7,5%	7,5%	7,5%	4,7%	6,5%	7,5%	5,6%
079	23	5	4	4	6	6	4	5	21,7%	17,4%	17,4%	26,1%	26,1%	17,4%	21,7%
080	4	1	1	0	0	1	1	1	25,0%	25,0%	0,0%	0,0%	25,0%	25,0%	25,0%
081	2	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
082	105	10	10	11	11	6	10	10	9,5%	9,5%	10,5%	10,5%	5,7%	9,5%	9,5%
083	28	4	6	6	6	6	4	7	14,3%	21,4%	21,4%	21,4%	21,4%	14,3%	25,0%
084	6	2	3	3	3	3	3	3	33,3%	50,0%	50,0%	50,0%	50,0%	50,0%	50,0%
085	2	1	1	1	1	2	2	2	50,0%	50,0%	50,0%	50,0%	100%	100%	100%
086	93	5	6	5	5	5	4	7	5,4%	6,5%	5,4%	5,4%	5,4%	4,3%	7,5%
087	29	7	6	7	7	5	6	6	24,1%	20,7%	24,1%	24,1%	17,2%	20,7%	20,7%
088	7	0	0	1	1	2	2	2	0,0%	0,0%	14,3%	14,3%	28,6%	28,6%	28,6%
089	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
090	17	2	4	4	2	5	4	4	11,8%	23,5%	23,5%	11,8%	29,4%	23,5%	23,5%
091	2	1	2	2	2	2	2	2	50,0%	100%	100%	100%	100%	100%	100%
092	4	1	0	0	0	0	0	0	25,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
093	2	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
094	47	7	7	6	2	6	5	5	14,9%	14,9%	12,8%	4,3%	12,8%	10,6%	10,6%
095	7	2	3	2	2	2	1	1	28,6%	42,9%	28,6%	28,6%	28,6%	14,3%	14,3%
096	2	0	1	1	0	1	1	1	0,0%	50,0%	50,0%	0,0%	50,0%	50,0%	50,0%
097	3	0	0	0	0	1	0	0	0,0%	0,0%	0,0%	0,0%	33,3%	0,0%	0,0%
098	1	1	0	0	0	0	0	0	100%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
099	4	0	0	0	0	0	0	1	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	25,0%
100	6	1	2	2	1	2	3	3	16,7%	33,3%	33,3%	16,7%	33,3%	50,0%	50,0%
101	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
102	10	2	1	2	1	4	2	5	20,0%	10,0%	20,0%	10,0%	40,0%	20,0%	50,0%
103	2	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
104	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
105	9	2	1	1	2	3	3	4	22,2%	11,1%	11,1%	22,2%	33,3%	33,3%	44,4%
106	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
107	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
108	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
109	92	9	8	10	9	8	7	6	9,8%	8,7%	10,9%	9,8%	8,7%	7,6%	6,5%
110	15	3	4	4	4	4	4	4	20,0%	26,7%	26,7%	26,7%	26,7%	26,7%	26,7%
111	10	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
112	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
113	17	4	4	2	2	4	3	5	23,5%	23,5%	11,8%	11,8%	23,5%	17,6%	29,4%
114	4	2	2	2	2	1	2	2	50,0%	50,0%	50,0%	50,0%	25,0%	50,0%	50,0%
115	2	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
116	3	1	1	0	0	0	0	0	33,3%	33,3%	0,0%	0,0%	0,0%	0,0%	0,0%
117	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
118	2	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
119	167	12	8	11	9	9	9	8	7,2%	4,8%	6,6%	5,4%	5,4%	5,4%	4,8%
120	32	4	7	5	6	7	8	6	12,5%	21,9%	15,6%	18,8%	21,9%	25,0%	18,8%
121	22	2	3	1	0	0	1	0	9,1%	13,6%	4,5%	0,0%	0,0%	4,5%	0,0%

(continua na página seguinte)

(continuação da tabela A2.1)

Estrato	Dimensão Populacional	Dimensão Amostral							Taxa de Sondagem						
		1º T	2º T	3º T	4º T	5º T	6º T	7º T	1º T	2º T	3º T	4º T	5º T	6º T	7º T
122	4	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
123	456	10	9	10	9	8	7	10	2,2%	2,0%	2,2%	2,0%	1,8%	1,5%	2,2%
124	129	9	10	7	7	5	6	8	7,0%	7,8%	5,4%	5,4%	3,9%	4,7%	6,2%
125	129	4	5	3	3	2	3	5	3,1%	3,9%	2,3%	2,3%	1,6%	2,3%	3,9%
126	31	5	5	3	4	4	3	5	16,1%	16,1%	9,7%	12,9%	12,9%	9,7%	16,1%
127	10	3	4	2	2	3	2	3	30,0%	40,0%	20,0%	20,0%	30,0%	20,0%	30,0%
128	3	1	0	0	1	1	1	1	33,3%	0,0%	0,0%	33,3%	33,3%	33,3%	33,3%
129	2	1	1	0	0	0	0	0	50,0%	50,0%	0,0%	0,0%	0,0%	0,0%	0,0%
130	2	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
131	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
132	48	7	7	5	7	6	4	5	14,6%	14,6%	10,4%	14,6%	12,5%	8,3%	10,4%
133	5	0	0	0	0	1	1	1	0,0%	0,0%	0,0%	0,0%	20,0%	20,0%	20,0%
134	11	1	1	1	1	2	2	2	9,1%	9,1%	9,1%	9,1%	18,2%	18,2%	18,2%
135	77	7	7	6	5	6	8	8	9,1%	9,1%	7,8%	6,5%	7,8%	10,4%	10,4%
136	17	7	7	7	5	7	7	9	41,2%	41,2%	41,2%	29,4%	41,2%	41,2%	52,9%
137	13	1	1	1	1	0	1	1	7,7%	7,7%	7,7%	7,7%	0,0%	7,7%	7,7%
138	3	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
139	210	11	13	9	9	7	6	8	5,2%	6,2%	4,3%	4,3%	3,3%	2,9%	3,8%
140	29	2	2	1	2	3	3	3	6,9%	6,9%	3,4%	6,9%	10,3%	10,3%	10,3%
141	19	5	5	5	5	5	3	5	26,3%	26,3%	26,3%	26,3%	26,3%	15,8%	26,3%
142	10	0	0	1	0	1	0	0	0,0%	0,0%	10,0%	0,0%	10,0%	0,0%	0,0%
143	1	1	1	1	1	1	1	1	100%	100%	100%	100%	100%	100%	100%
144	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
145	45	10	12	9	10	9	9	9	22,2%	26,7%	20,0%	22,2%	20,0%	20,0%	20,0%
146	7	3	3	3	3	2	2	2	42,9%	42,9%	42,9%	42,9%	28,6%	28,6%	28,6%
147	4	4	4	4	4	4	3	3	100%	100%	100%	100%	100%	75,0%	75,0%
148	1	1	1	1	1	1	1	1	100%	100%	100%	100%	100%	100%	100%
149	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
150	76	11	12	8	6	5	7	7	14,5%	15,8%	10,5%	7,9%	6,6%	9,2%	9,2%
151	11	2	1	2	1	2	2	2	18,2%	9,1%	18,2%	9,1%	18,2%	18,2%	18,2%
152	10	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
153	41	6	7	6	5	5	5	6	14,6%	17,1%	14,6%	12,2%	12,2%	12,2%	14,6%
154	4	1	0	1	1	1	1	1	25,0%	0,0%	25,0%	25,0%	25,0%	25,0%	25,0%
155	4	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
156	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
157	1	1	0	0	0	0	0	0	100%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
158	1	1	1	0	0	1	0	0	100%	100%	0,0%	0,0%	100%	0,0%	0,0%
159	2	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
160	2	1	1	1	1	1	1	1	50,0%	50,0%	50,0%	50,0%	50,0%	50,0%	50,0%
161	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
162	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
163	8	2	2	2	1	2	2	1	25,0%	25,0%	25,0%	12,5%	25,0%	25,0%	12,5%
164	121	16	14	14	14	11	8	10	13,2%	11,6%	11,6%	11,6%	9,1%	6,6%	8,3%
165	15	6	6	6	5	5	5	5	40,0%	40,0%	40,0%	33,3%	33,3%	33,3%	33,3%
166	13	1	1	1	1	1	1	1	7,7%	7,7%	7,7%	7,7%	7,7%	7,7%	7,7%
167	2	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
168	29	4	4	6	5	4	5	4	13,8%	13,8%	20,7%	17,2%	13,8%	17,2%	13,8%
169	2	1	1	1	0	2	1	2	50,0%	50,0%	50,0%	0,0%	100%	50,0%	100%
170	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
171	24	5	5	6	4	6	5	5	20,8%	20,8%	25,0%	16,7%	25,0%	20,8%	20,8%
172	4	2	1	1	1	1	1	1	50,0%	25,0%	25,0%	25,0%	25,0%	25,0%	25,0%
173	25	6	5	4	2	6	4	5	24,0%	20,0%	16,0%	8,0%	24,0%	16,0%	20,0%
174	5	1	1	1	1	1	2	2	20,0%	20,0%	20,0%	20,0%	20,0%	40,0%	40,0%
175	2	1	1	0	0	1	0	0	50,0%	50,0%	0,0%	0,0%	50,0%	0,0%	0,0%
176	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
177	15	4	4	3	2	3	2	3	26,7%	26,7%	20,0%	13,3%	20,0%	13,3%	20,0%
178	1	0	1	0	0	0	0	0	0,0%	100%	0,0%	0,0%	0,0%	0,0%	0,0%
179	98	8	5	8	4	5	5	7	8,2%	5,1%	8,2%	4,1%	5,1%	5,1%	7,1%
180	7	2	2	2	2	2	2	0	28,6%	28,6%	28,6%	28,6%	28,6%	28,6%	0,0%
181	5	2	2	3	1	1	1	1	40,0%	40,0%	60,0%	20,0%	20,0%	20,0%	20,0%
182	26	3	3	4	3	1	4	3	11,5%	11,5%	15,4%	11,5%	3,8%	15,4%	11,5%
183	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
184	1	1	1	1	1	1	1	1	100%	100%	100%	100%	100%	100%	100%
185	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
186	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
187	62	7	5	3	3	7	9	8	11,3%	8,1%	4,8%	4,8%	11,3%	14,5%	12,9%

(continua na página seguinte)

(continuação da tabela A2.1)

Estrato	Dimensão Populacional	Dimensão Amostral							Taxa de Sondagem						
		1º T	2º T	3º T	4º T	5º T	6º T	7º T	1º T	2º T	3º T	4º T	5º T	6º T	7º T
188	8	1	1	1	1	1	1	1	12,5%	12,5%	12,5%	12,5%	12,5%	12,5%	12,5%
189	4	3	3	3	3	3	3	2	75,0%	75,0%	75,0%	75,0%	75,0%	75,0%	50,0%
190	1	1	1	1	0	0	1	0	100%	100%	100%	0,0%	0,0%	100%	0,0%
191	41	6	4	6	2	5	4	6	14,6%	9,8%	14,6%	4,9%	12,2%	9,8%	14,6%
192	5	2	2	2	2	2	2	2	40,0%	40,0%	40,0%	40,0%	40,0%	40,0%	40,0%
193	3	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
194	1	1	0	0	0	0	1	1	100%	0,0%	0,0%	0,0%	0,0%	100%	100%
195	2	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
196	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
197	1	1	1	1	1	1	1	1	100%	100%	100%	100%	100%	100%	100%
198	50	4	4	3	4	3	4	3	8,0%	8,0%	6,0%	8,0%	6,0%	8,0%	6,0%
199	8	3	3	3	3	3	3	2	37,5%	37,5%	37,5%	37,5%	37,5%	37,5%	25,0%
200	6	1	1	0	0	0	0	0	16,7%	16,7%	0,0%	0,0%	0,0%	0,0%	0,0%
201	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
202	2	0	0	0	0	1	1	1	0,0%	0,0%	0,0%	0,0%	50,0%	50,0%	50,0%
203	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
204	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
205	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
206	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
207	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
208	4	1	0	0	0	0	0	0	25,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
209	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
210	2	1	1	1	1	1	1	1	50,0%	50,0%	50,0%	50,0%	50,0%	50,0%	50,0%
211	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
212	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
213	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
214	3	0	1	0	0	0	0	0	0,0%	33,3%	0,0%	0,0%	0,0%	0,0%	0,0%
215	2	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
216	2	1	1	0	0	0	0	0	50,0%	50,0%	0,0%	0,0%	0,0%	0,0%	0,0%
217	22	2	2	3	2	5	1	5	9,1%	9,1%	13,6%	9,1%	22,7%	4,5%	22,7%
218	2	0	1	1	1	1	0	1	0,0%	50,0%	50,0%	50,0%	50,0%	0,0%	50,0%
219	13	3	3	4	4	5	1	4	23,1%	23,1%	30,8%	30,8%	38,5%	7,7%	30,8%
220	24	4	4	5	4	6	2	6	16,7%	16,7%	20,8%	16,7%	25,0%	8,3%	25,0%
221	3	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
222	2	0	1	1	1	1	0	1	0,0%	50,0%	50,0%	50,0%	50,0%	0,0%	50,0%
223	21	3	4	5	4	4	1	4	14,3%	19,0%	23,8%	19,0%	19,0%	4,8%	19,0%
224	2	0	0	0	0	1	0	0	0,0%	0,0%	0,0%	0,0%	50,0%	0,0%	0,0%
225	2	1	0	0	0	0	0	0	50,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
226	319	12	18	16	16	14	14	13	3,8%	5,6%	5,0%	5,0%	4,4%	4,4%	4,1%
227	111	7	10	10	8	9	5	10	6,3%	9,0%	9,0%	7,2%	8,1%	4,5%	9,0%
228	25	3	4	5	4	2	4	3	12,0%	16,0%	20,0%	16,0%	8,0%	16,0%	12,0%
229	7	0	0	2	2	0	1	1	0,0%	0,0%	28,6%	28,6%	0,0%	14,3%	14,3%
230	2	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
231	4	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
232	1	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
233	2	0	0	0	0	0	0	0	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%	0,0%
234	1	1	1	1	1	1	1	1	100%	100%	100%	100%	100%	100%	100%
Total	**4671**	**486**	**498**	**478**	**414**	**441**	**424**	**467**	**10,4%**	**10,7%**	**10,2%**	**8,9%**	**9,4%**	**9,1%**	**10,0%**

1º T – 1º Trimestre (1º Trimestre de 2002); 2º T – 2º Trimestre (2º Trimestre de 2002);

3º T – 3º Trimestre (3º Trimestre de 2002); 4º T – 4º Trimestre (4º Trimestre de 2002);

5º T – 8º Trimestre (1º Trimestre de 2003); 6º T – 6º Trimestre (2º Trimestre de 2003);

7º T – 7º Trimestre (3º Trimestre de 2003).

Apêndice 3 – Matriz de pesos espaciais

A3.1 Matriz de pesos espaciais estandardizada por linhas

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0,25	0	0,25	0,25	0	0	0	0,25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0,25	0	0,25	0,25	0	0	0,25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0,20	0,20	0	0,20	0,20	0	0	0,20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0,17	0,17	0	0,17	0,17	0,17	0	0	0	0	0,17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0,25	0,25	0	0	0	0,25	0	0	0	0,25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0,25	0	0	0,25	0	0	0	0	0,25	0	0	0,25	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0,25	0,25	0	0,25	0	0,25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0,25	0	0,25	0	0	0	0,25	0	0	0,25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0,25	0	0,25	0,25	0,25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0,20	0	0,20	0	0	0	0	0	0	0,2	0	0	0,20	0,20	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0,14	0,14	0	0,14	0,14	0,14	0	0	0,14	0	0	0	0,14	0	0	0	0	0	0
13	0	0	0	0	0,13	0,13	0,13	0	0,13	0,13	0	0,13	0	0	0,13	0,13	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0,20	0	0	0	0	0,20	0,20	0	0	0	0,20	0	0	0,20	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0,25	0,25	0	0	0,25	0	0,25	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0,20	0	0	0	0	0	0,20	0	0,20	0	0,20	0,2	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0,25	0	0,25	0	0,25	0	0	0	0	0	0	0,25	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0,20	0	0,20	0,20	0,20	0,20	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0,33	0	0	0	0	0	0	0	0	0,33	0	0	0,33	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,33	0	0,33	0	0,33	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,25	0	0	0,25	0,25	0	0,25	0	0
22	0	0	0	0	0	0	0	0	0	0	0,20	0,20	0	0,20	0	0	0	0	0	0	0	0	0,20	0	0,20	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0,14	0	0	0	0	0	0	0	0,14	0,14	0,14	0,14	0	0	0,14	0,14	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,25	0	0	0	0	0,25	0,25	0,25
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0,20	0	0	0,20	0	0	0	0	0,20	0,20	0	0	0,20	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,20	0	0,20	0,20	0,20	0	0,20	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,33	0	0,33	0	0,33
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0,50	0	0	0,50	0

Apêndice 4 – Limites aceitáveis de áreas e preços segundo a tipologia por fogo

A4.1 Limites inferiores e superiores das áreas para cada uma das tipologias dos fogos (em metros quadrados)

Tipologia por fogo	IPTH		IABH	
	Limite Inferior	**Limite Superior**	**Limite Inferior**	**Limite Superior**
Moradia com tipologia inferior ou igual a T3	26	380	26	397
Moradia com tipologia superior ou igual a T4	78	520	78	530
Apartamento com tipologia inferior ou igual a T1	26	115	26	155
Apartamento com tipologia T2	54	160	54	167
Apartamento com tipologia T3	68	220	68	229
Apartamento com tipologia superior ou igual a T4	78	340	78	373

Fonte: Adaptado de INE (2001*b*).

A4.2 Limites inferiores e superiores dos preços de transacção por metro quadrado para cada uma das tipologias dos fogos em 2002 e 2003 (em Euros)

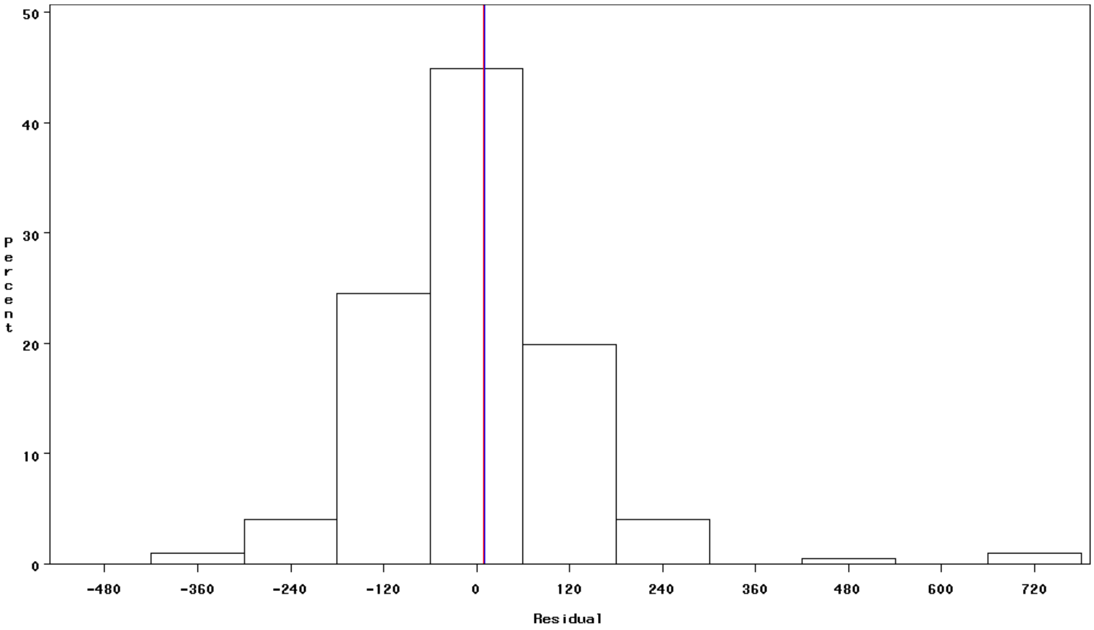
Tipologia por fogo	2002		2003	
	Limite Inferior	**Limite Superior**	**Limite Inferior**	**Limite Superior**
Moradia com tipologia inferior ou igual a T3	185	3.965	174	4.242
Moradia com tipologia superior ou igual a T4	216	4.697	201	5.077
Apartamento com tipologia inferior ou igual a T1	304	4.182	290	4.145
Apartamento com tipologia T2	287	3.218	277	3.439
Apartamento com tipologia T3	244	3.335	233	3.886
Apartamento com tipologia superior ou igual a T4	223	5.057	217	4.778

A4.3 Limites inferiores e superiores dos preços de avaliação bancária por metro quadrado para cada uma das tipologias dos fogos em 2001, 2002 e 2003 (em Euros)

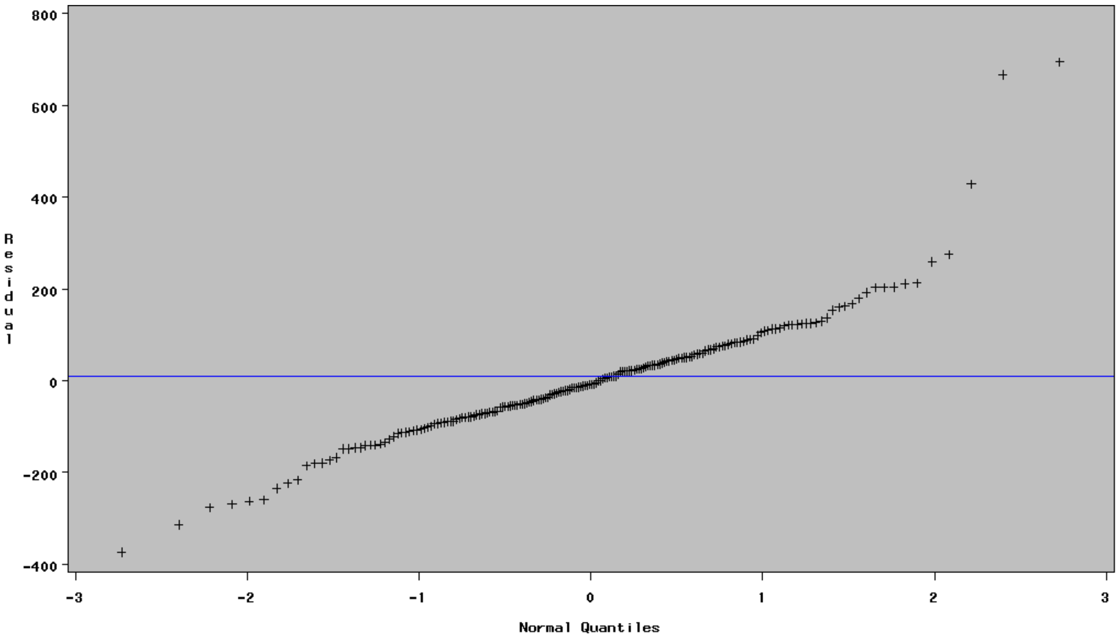
Tipologia por fogo	2001		2002		2003	
	Limite Inferior	Limite Superior	Limite Inferior	Limite Superior	Limite Inferior	Limite Superior
Moradia com tipologia inferior ou igual a T3	249	4.376	258	4.692	267	5.170
Moradia com tipologia superior ou igual a T4	260	4.553	270	4.734	281	5.052
Apartamento com tipologia inferior ou igual a T1	375	4.928	379	5.003	383	6.003
Apartamento com tipologia T2	337	3.810	341	3.871	346	4.649
Apartamento com tipologia T3	309	4.076	314	4.129	319	5.071
Apartamento com tipologia superior ou igual a T4	319	5.041	324	4.987	329	5.660

Apêndice 5 – Análise dos resíduos do modelo de regressão linear simples

A5.1 Histograma dos resíduos do modelo de regressão linear simples



A5.2 Gráfico Q-Q dos resíduos do modelo de regressão linear simples



Apêndice 6 – Dimensões pseudo-populacional e amostral de unidades primárias no estudo por simulação *design-based*

A6.1 Dimensões populacional e amostral e taxa de sondagem em cada estrato

Estrato	Código do Estrato	Dimensão Populacional	Dimensão Amostral	Taxa de Sondagem
01	101010000	43	5	11,6%
02	101020000	117	5	4,3%
03	101030000	60	5	8,3%
04	101041304	66	3	4,5%
05	101041306	93	5	5,4%
06	101041308	92	3	3,3%
07	101041312	268	10	3,7%
08	101041313	24	4	16,7%
09	101041315	39	3	7,7%
10	101041316	17	2	11,8%
11	101041317	179	7	3,9%
12	101050000	44	2	4,5%
13	101060000	65	4	6,2%
14	101070000	19	1	5,3%
15	101080000	20	2	10,0%
16	102010000	139	6	4,3%
17	102020000	138	10	7,2%
18	102030000	132	6	4,5%
19	102040000	19	3	15,8%
20	102050000	55	3	5,5%
21	102060000	2	1	50,0%
22	102070000	4	1	25,0%
23	102080000	9	2	22,2%
24	102090000	15	3	20,0%
25	102100000	14	3	21,4%
26	103010000	117	6	5,1%
27	103011109	24	3	12,5%
28	103021105	226	9	4,0%
29	103021106	770	13	1,7%
30	103021107	67	3	4,5%
31	103021110	112	7	6,3%
32	103021111	274	8	2,9%
33	103021114	61	9	14,8%
34	103021115	106	5	4,7%
35	103021116	52	3	5,8%
36	103031502	8	1	12,5%
37	103031503	152	10	6,6%
38	103031504	33	3	9,1%
39	103031506	31	3	9,7%
40	103031507	35	3	8,6%
41	103031508	17	2	11,8%
42	103031510	113	5	4,4%
43	103031511	34	2	5,9%
44	103031512	74	5	6,8%
45	103040000	49	3	6,1%
46	103050000	68	3	4,4%
47	104010000	24	2	8,3%
48	104020000	13	2	15,4%
49	104030000	32	3	9,4%
50	104040000	23	3	13,0%
51	105010000	471	14	3,0%
Total	**-**	**4659**	**229**	**4,9%**

Apêndice 7 – Grupos de domínios

A7.1 Constituição dos grupos de domínios

Grupo	NUTSIII	Designação	Dimensão média amostral
1	166	Pinhal Interior Sul	1
1	167	Serra da Estrela	1
2	118	Alto Trás-os-Montes	4
2	168	Beira Interior Norte	2
2	169	Beira Interior Sul	3
2	16A	Cova da Beira	5
3	117	Douro	8
3	182	Alto Alentejo	8
3	184	Baixo Alentejo	9
4	111	Minho-Lima	11
4	112	Cávado	14
4	115	Tâmega	10
4	116	Entre Douro e Vouga	14
4	164	Pinhal Interior Norte	15
4	165	Dão-Lafões	12
4	181	Alentejo Litoral	12
4	183	Alentejo Central	16
5	113	Ave	20
5	161	Baixo Vouga	29
5	163	Pinhal Litoral	28
5	16C	Médio Tejo	21
5	185	Lezíria do Tejo	22
6	114	Grande Porto	189
6	162	Baixo Mondego	54
6	16B	Oeste	41
6	171	Grande Lisboa	277
6	172	Península de Setúbal	165
6	150	Algarve	45

Apêndice 8 – Procedimento MIXED do SAS para estimação dos modelos pelo método da MVR

A8.1 Modelo seccional de Fay-Herriot:

```
Proc mixed data=cron MMEqSol covtest METHOD=REML MAXITER=100;
   class i;
   model y=x / solution ddfm=RESIDUAL;
   random i / solution G V;
   repeated /group=i;
   parms /parmsdata=par
hold=2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29;
run; quit;
```

A8.2 Modelo espacial de Nicola Salvati:

```
proc mixed data=cron MMEqSol covtest METHOD=REML MAXITER=100;
   class i;
   model y = x / solution ddfm=RESIDUAL;
   random i / type = sp(lin)(F1-F28);
   repeated /group=i;
   parms /parmsdata=par2
hold=3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30;
run; quit;
```

A8.3 Modelo seccional e cronológico de Rao-Yu:

```
Proc mixed data=cron MMEqSol covtest METHOD=REML MAXITER=100;
   class i t;
   model y=x / solution ddfm=RESIDUAL;
   random i / solution G V;
   random t / subject=i type=ar(1);
   repeated /group=i*t;
   parms /parmsdata=par
hold=4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33
,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,
63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,9
2,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,1
16,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,1
38,139,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,1
60,161,162,163,164,165,166,167,168,169,170,171,172,173,174,175,176,177,178,179,180,181,1
82,183,184,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199;
run; quit;
```

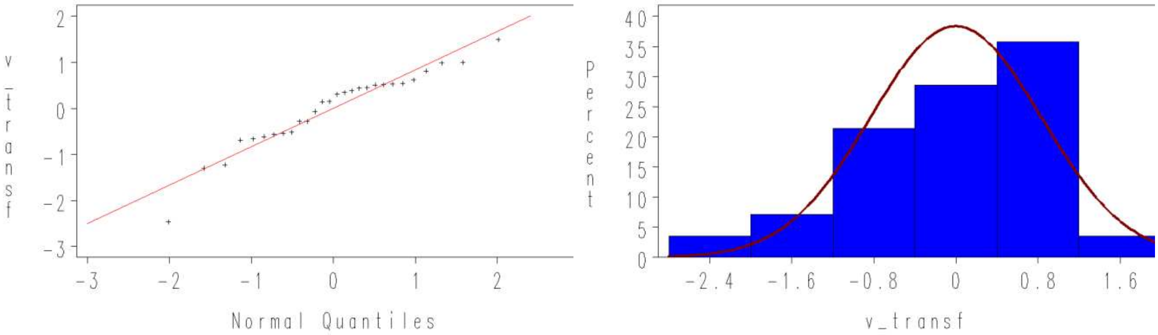
A8.4 Modelo espaciotemporal de Luís Pereira:

```
Proc mixed data=cron MMEqSol covtest METHOD=REML MAXITER=100;
   class i t;
   model y=x / solution ddfm=residual;
   random i / type = sp(lin)(col1-col28);
   random t / subject=i type=ar(1);
   repeated /group=i*t;
 parms /parmsdata=par2
hold=5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,3
4,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63
,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,
93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116
,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138
,139,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,160
,161,162,163,164,165,166,167,168,169,170,171,172,173,174,175,176,177,178,179,180,181,182
,183,184,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,200;
run; quit;
```

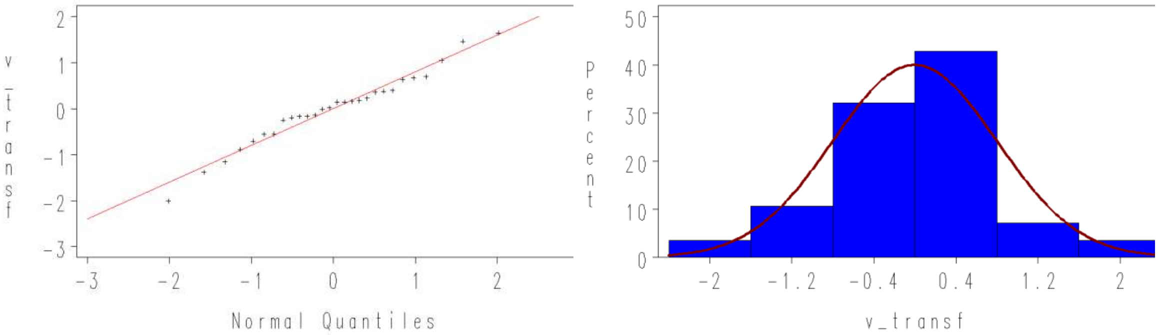
Apêndice 9 – Estudo da normalidade dos erros do modelo de Fay-Herriot

A9.1 Gráfico Q-Q e histograma dos efeitos aleatórios transformados, para cada trimestre

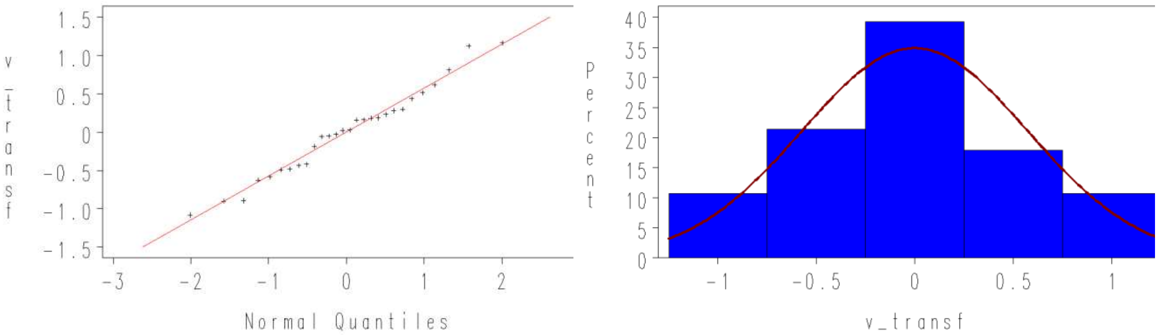
***t*=1**



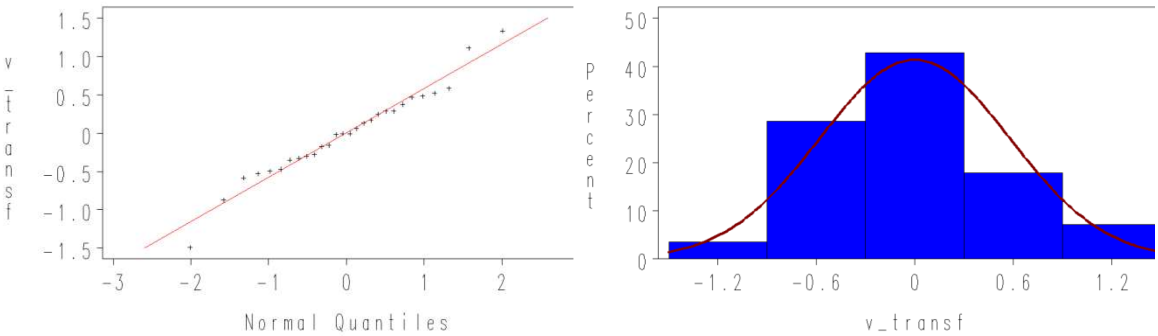
***t*=2**



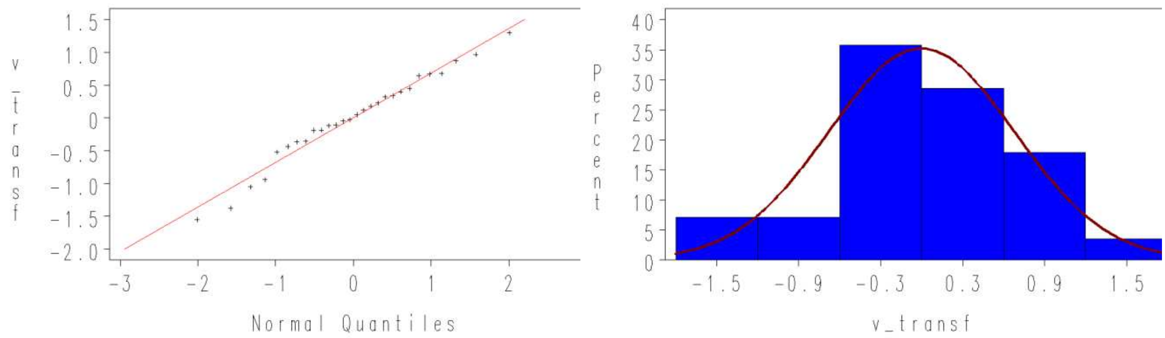
***t*=3**



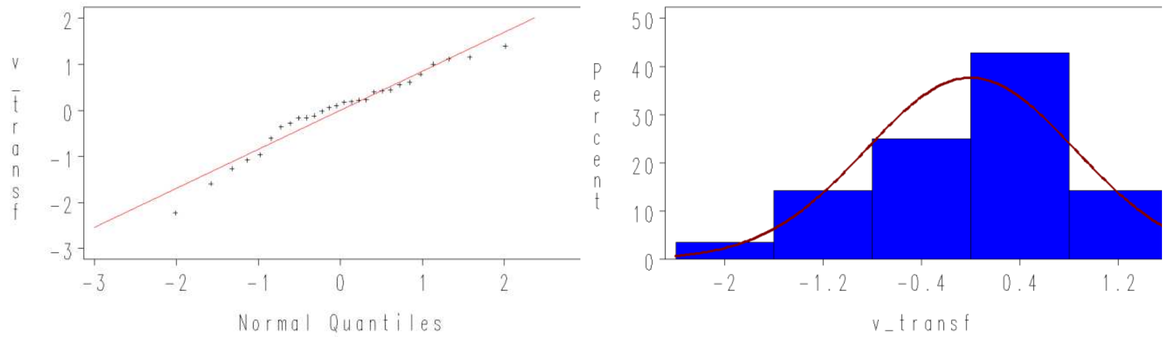
***t*=4**



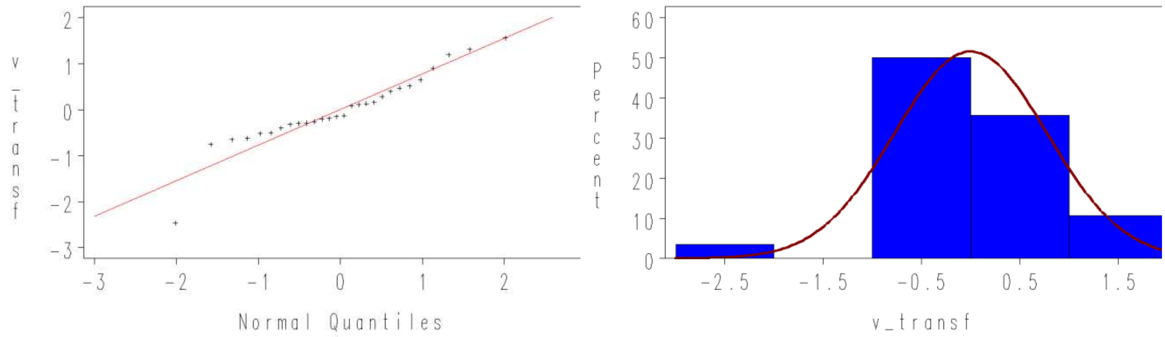
***t*=5**



***t*=6**



***t*=7**

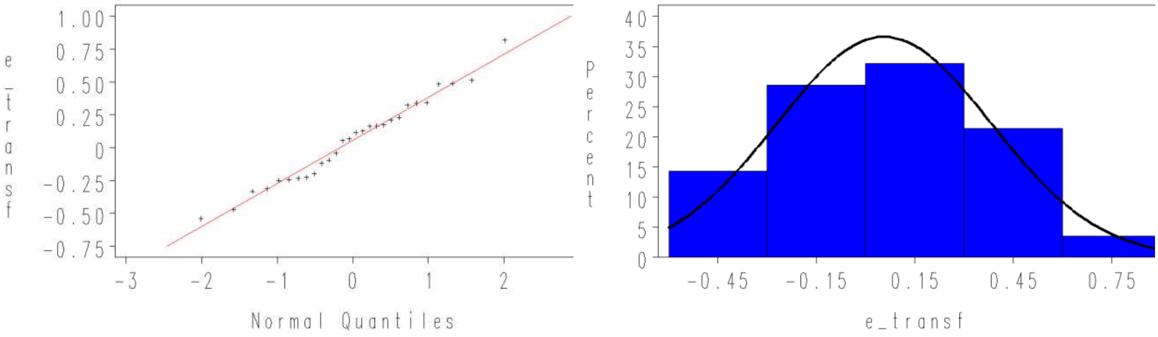


A9.2 Medidas de assimetria e curtose dos efeitos aleatórios

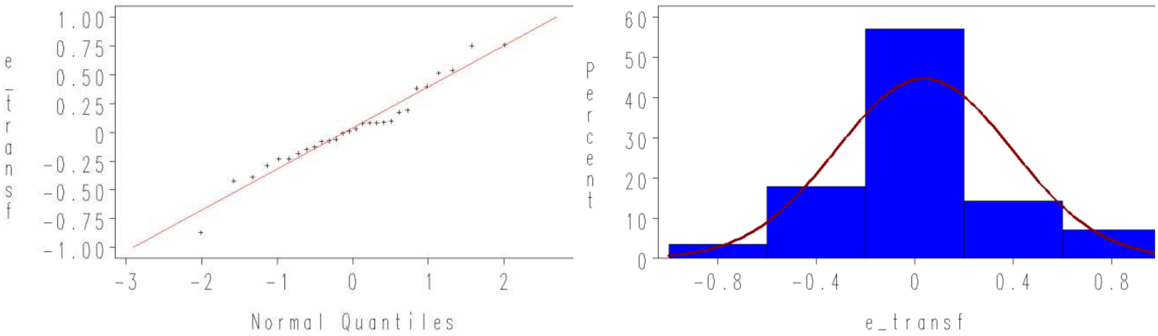
t	Assimetria	Curtose
1	-0,926	1,504
2	-0,303	0,786
3	0,114	-0,239
4	-0,052	1,146
5	-0,475	0,196
6	-0,778	0,672
7	-0,629	3,024

A9.3 Gráfico Q-Q e histograma dos erros da sondagem transformados, para cada trimestre

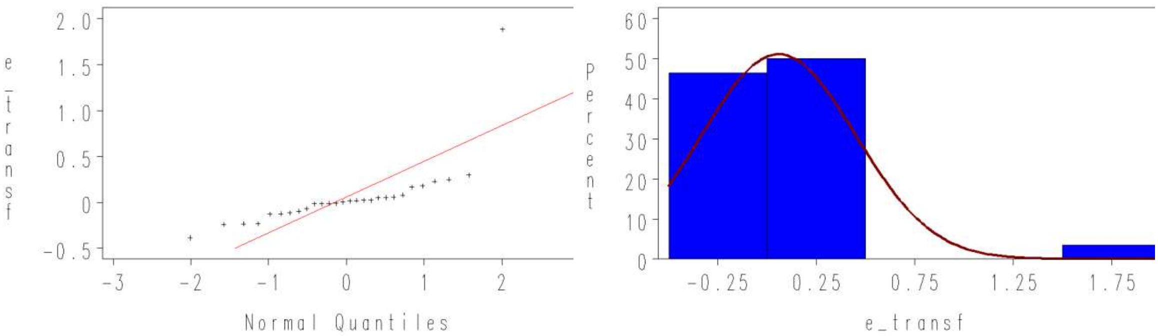
***t*=1**



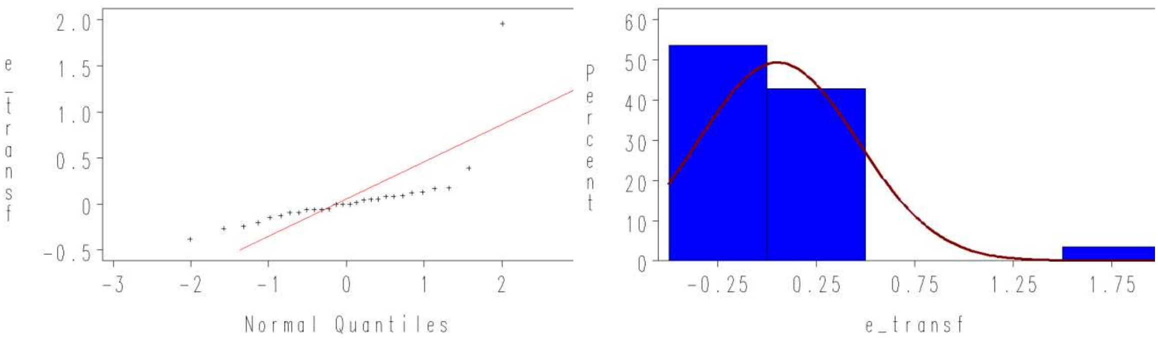
***t*=2**



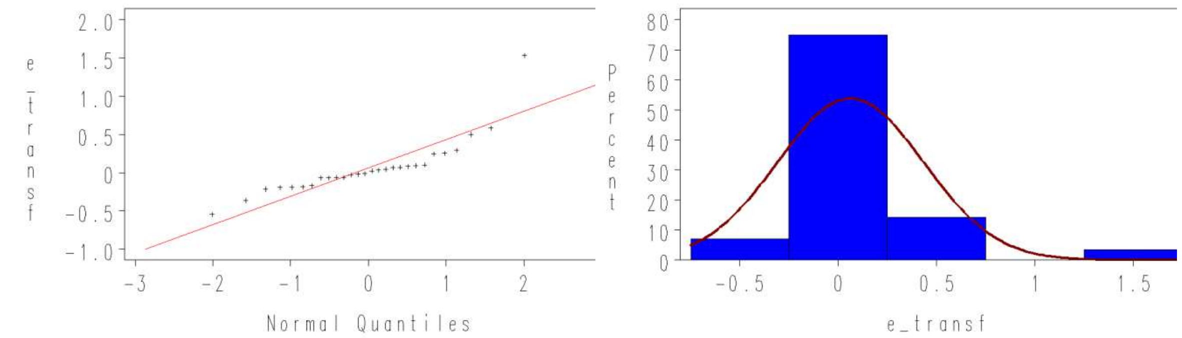
***t*=3**



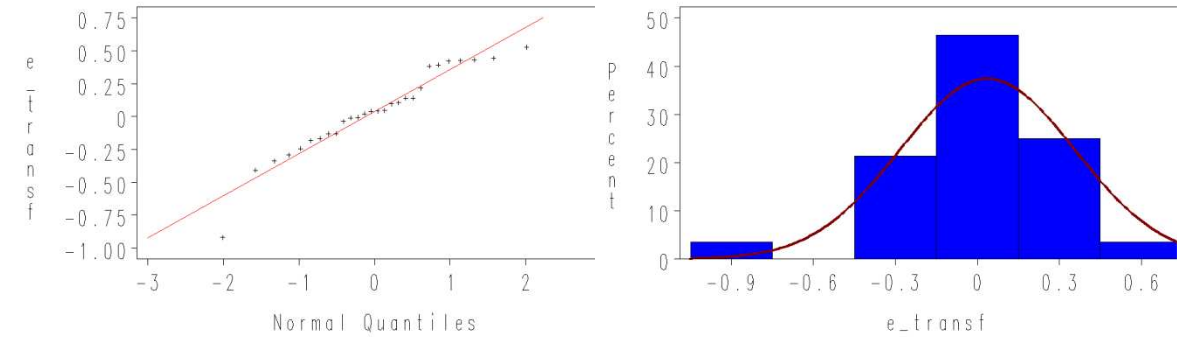
***t*=4**



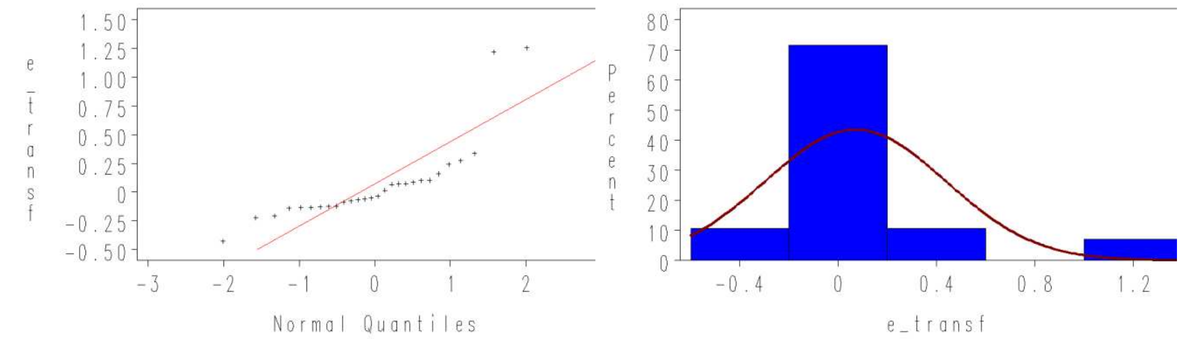
***t*=5**



***t*=6**



***t*=7**



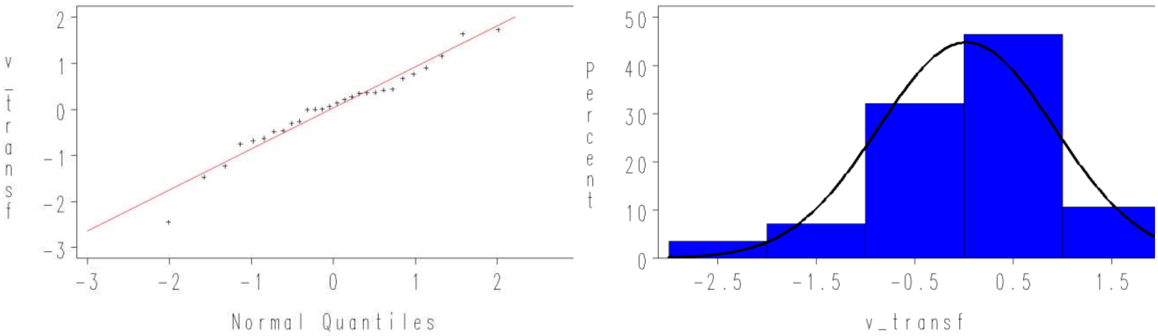
A9.4 Medidas de assimetria e curtose dos erros da sondagem

t	**Assimetria**	**Curtose**
1	0,212	-0,329
2	0,041	0,834
3	3,979	18,972
4	4,081	19,558
5	2,375	8,829
6	-0,778	1,509
7	2,397	6,318

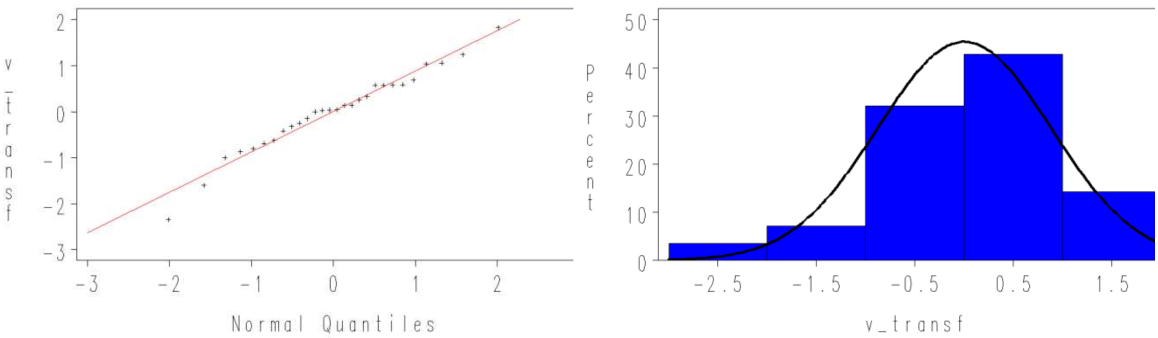
Apêndice 10 – Estudo da normalidade dos erros do modelo de Salvati

A10.1 Gráfico Q-Q e histograma dos efeitos aleatórios transformados, para cada trimestre

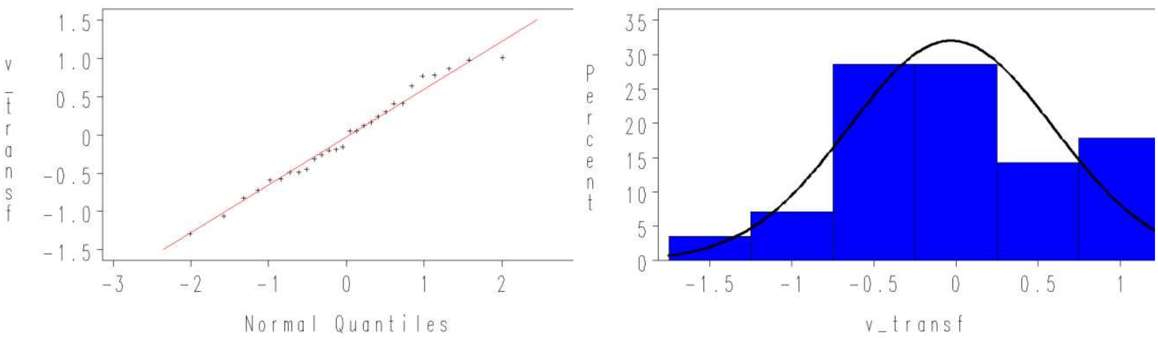
***t*=1**



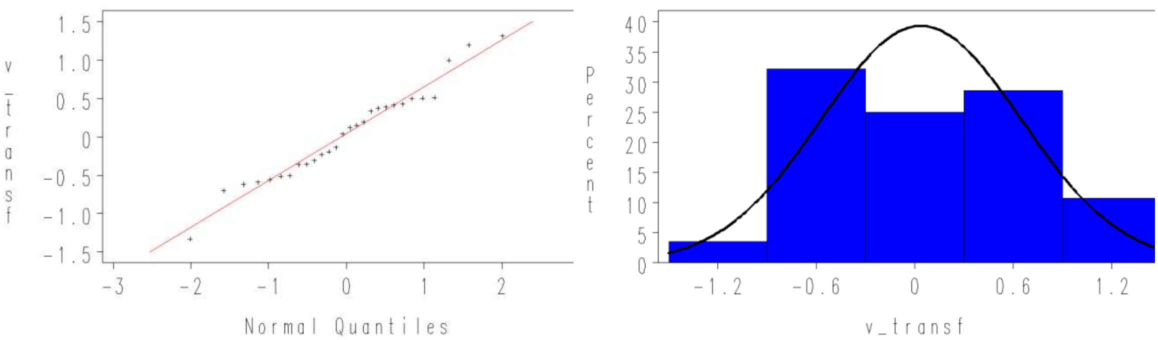
***t*=2**



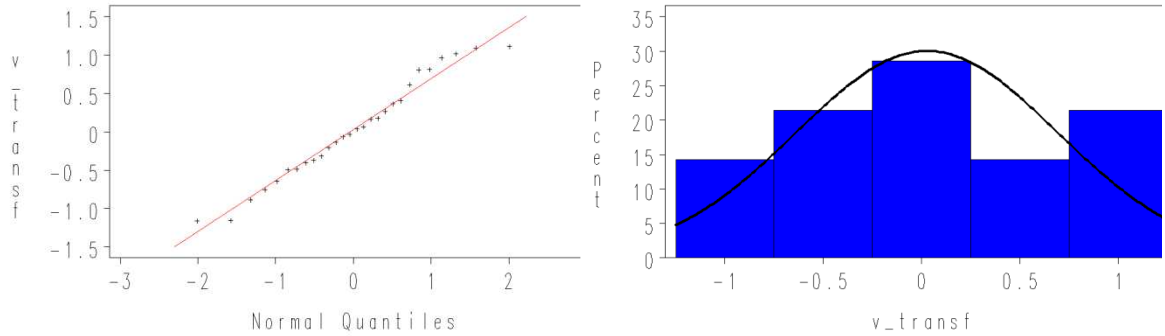
***t*=3**



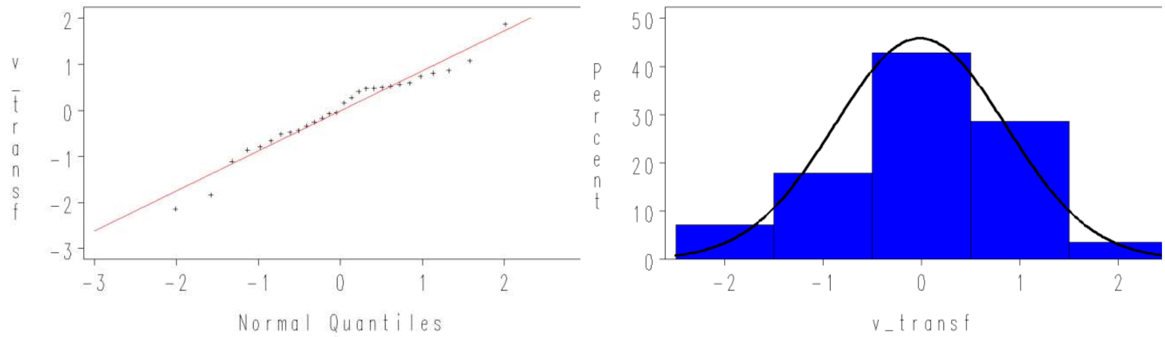
***t*=4**



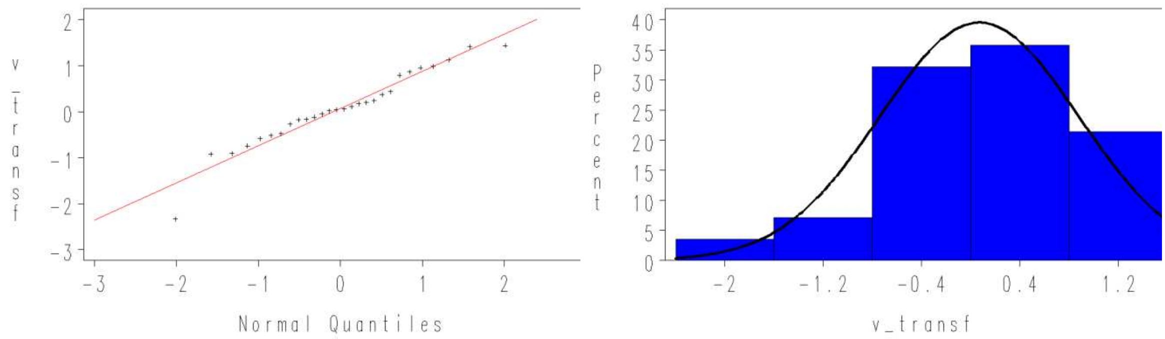
***t*=5**



***t*=6**



***t*=7**

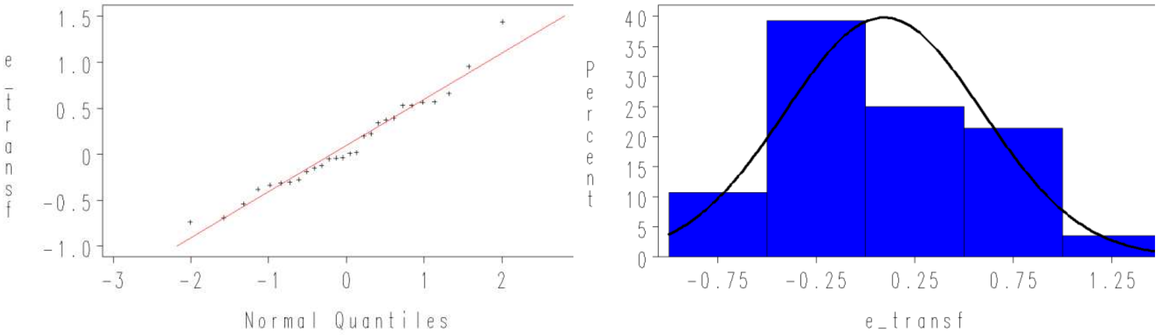


A10.2 Medidas de assimetria e curtose dos efeitos aleatórios

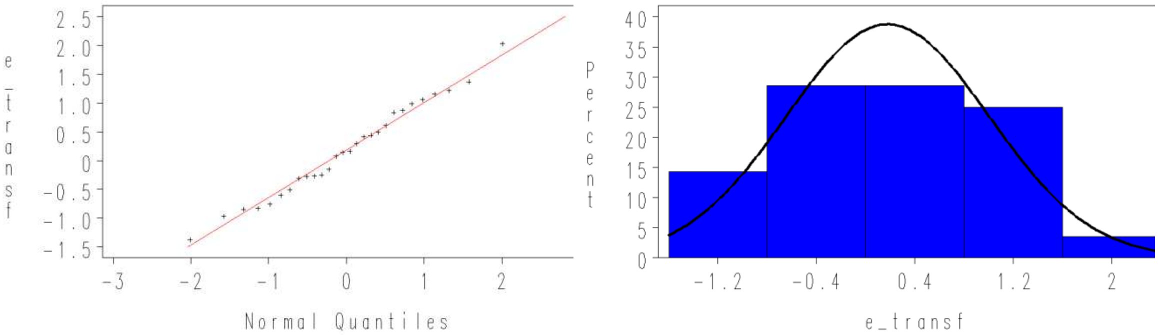
t	Assimetria	Curtose
1	-0,555	1,331
2	-0,495	0,993
3	-0,025	-0,722
4	0,146	0,025
5	0,019	-0,801
6	-0,485	0,688
7	-0,643	1,668

A10.3 Gráfico Q-Q e histograma dos erros da sondagem transformados, para cada trimestre

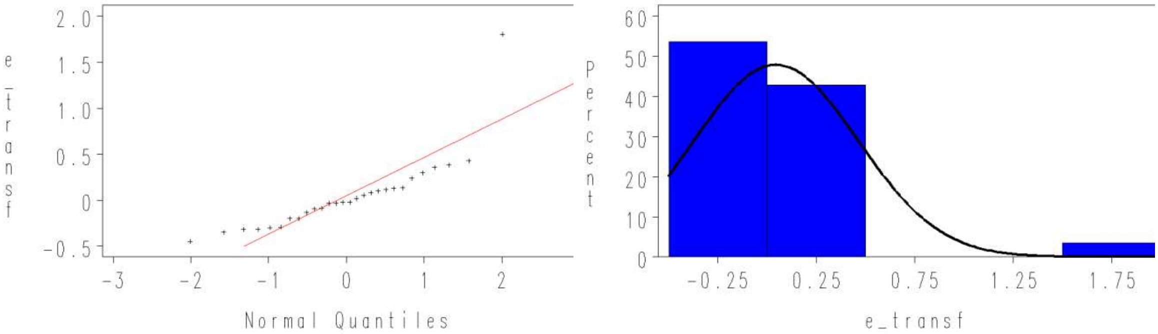
***t*=1**



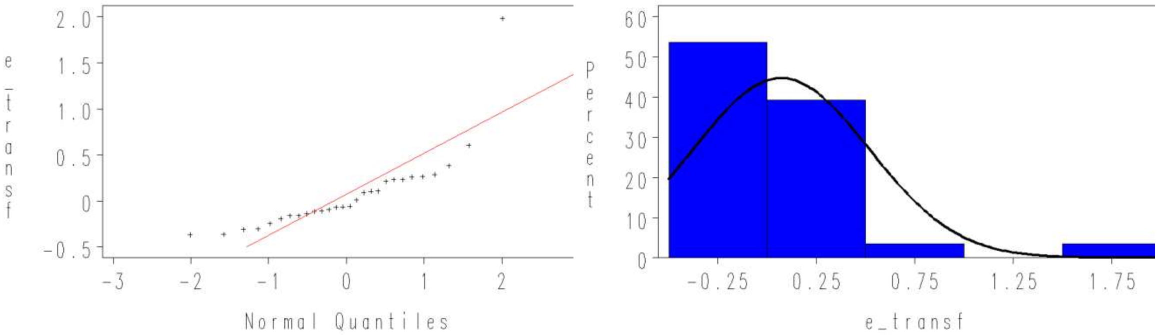
***t*=2**



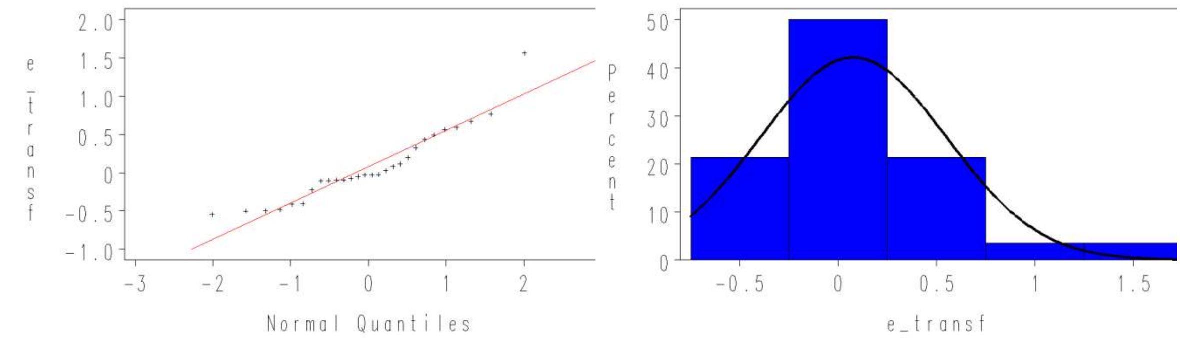
***t*=3**



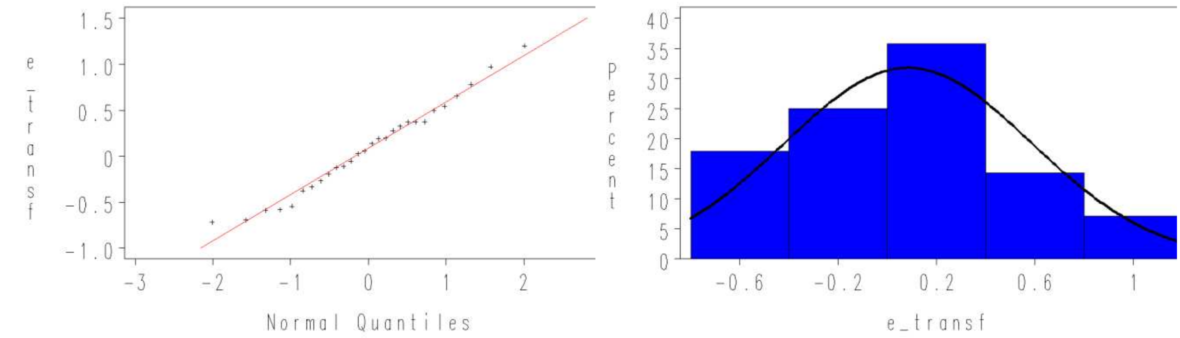
***t*=4**



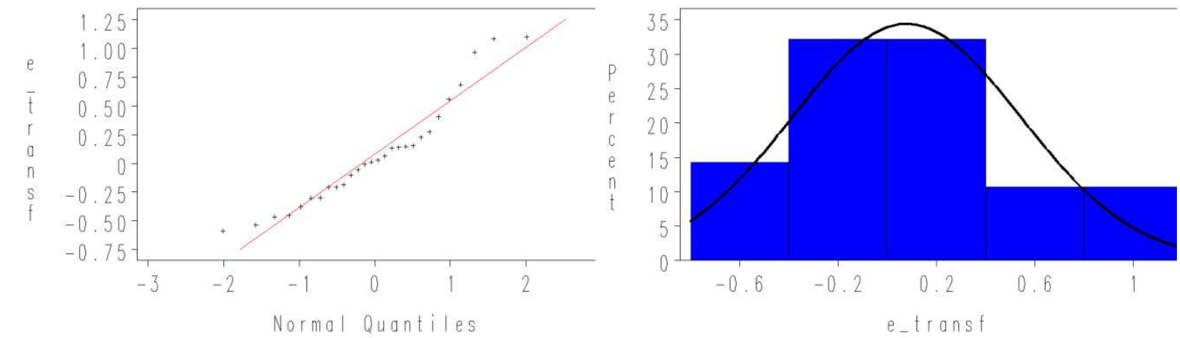
t=5



t=6



t=7

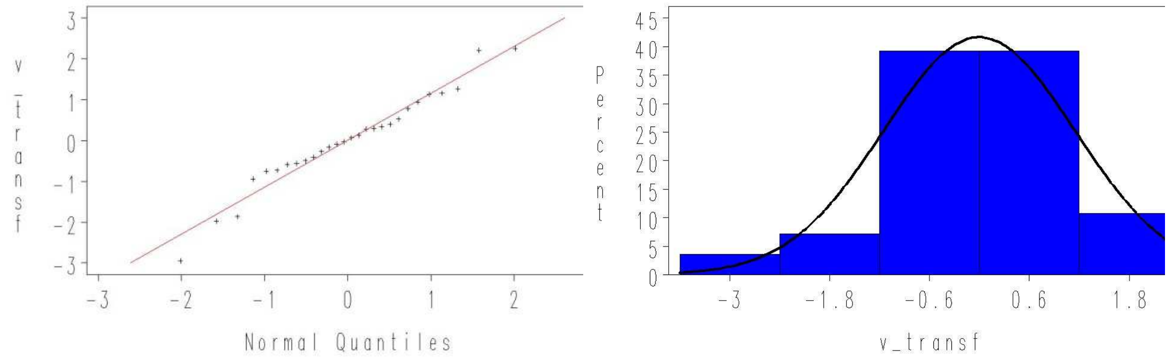


A10.4 Medidas de assimetria e curtose dos erros da sondagem

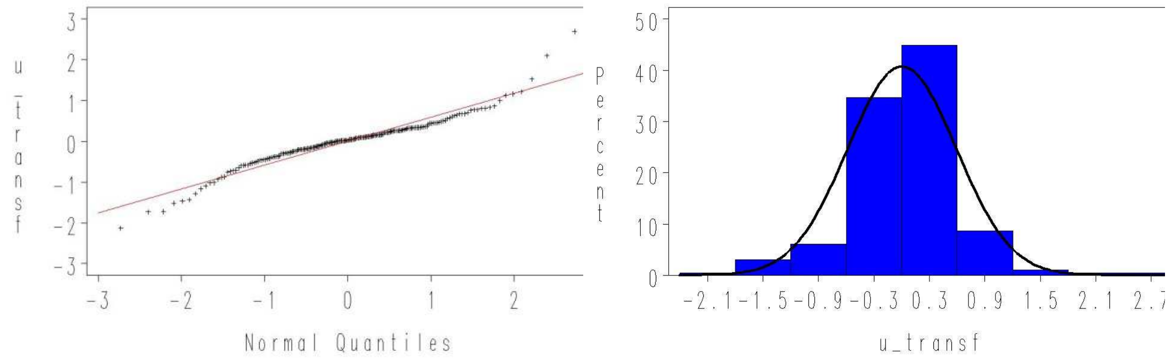
t	Assimetria	Curtose
1	0,620	0,516
2	0,187	-0,503
3	2,858	11,623
4	3,060	12,504
5	1,175	2,175
6	0,238	-0,438
7	0,820	0,173

Apêndice 11 – Estudo da normalidade dos erros do modelo de Rao-Yu

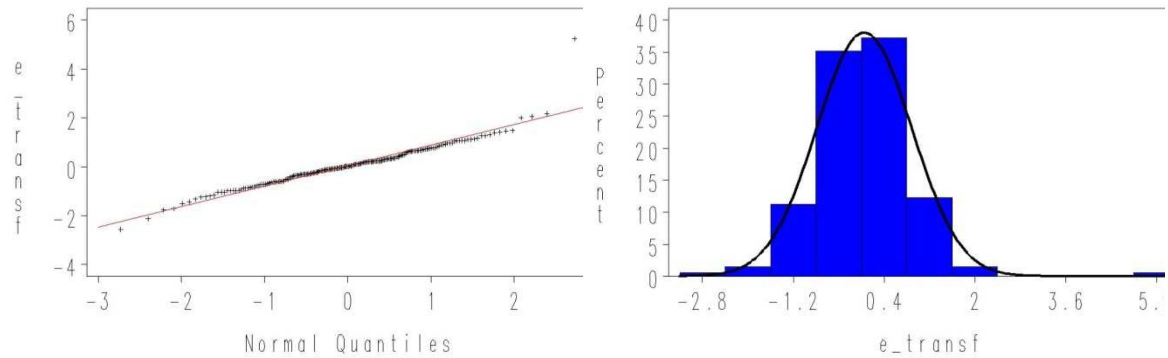
A11.1 Gráfico Q-Q e histograma dos efeitos aleatórios de domínio transformados



A11.2 Gráfico Q-Q e histograma dos efeitos aleatórios de domínio-tempo transformados



A11.3 Gráfico Q-Q e histograma dos erros da sondagem transformados

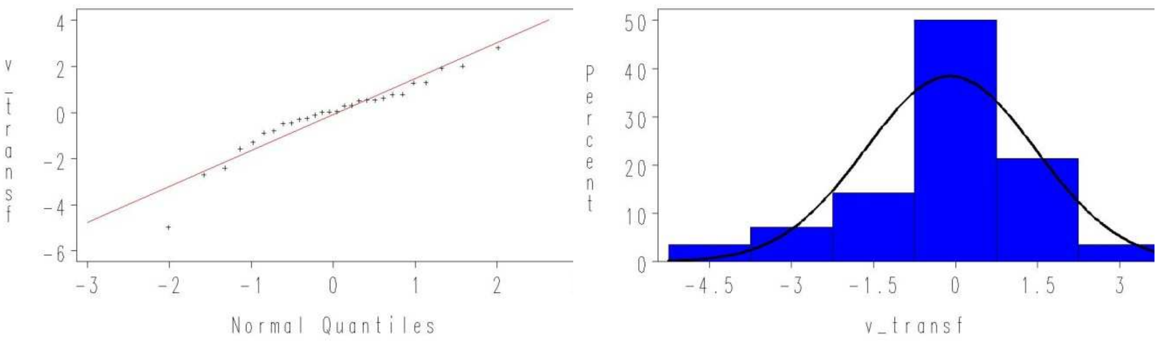


A11.4 Medidas de assimetria e curtose

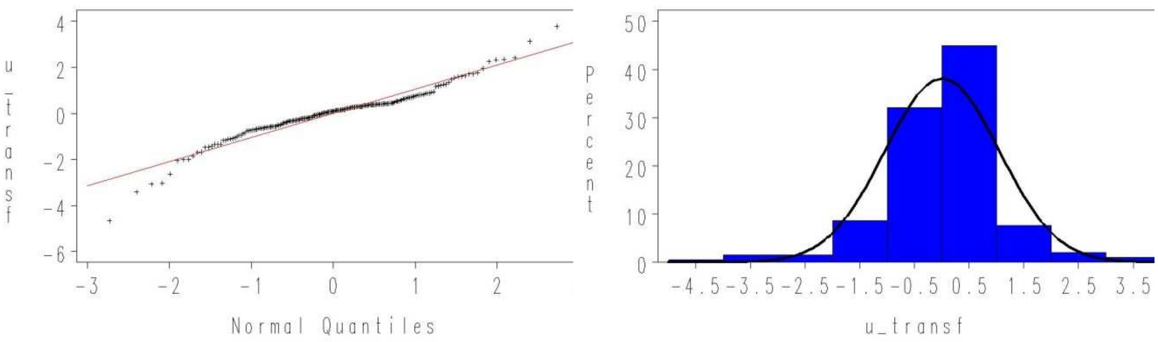
Medida	Efeitos aleatórios de domínio	Efeitos aleatórios de domínio-tempo	Erros da sondagem
Assimetria	-0,342	0,057	1,076
Curtose	0,917	3,822	7,086

Apêndice 12 – Estudo da normalidade dos erros do modelo espaciotemporal

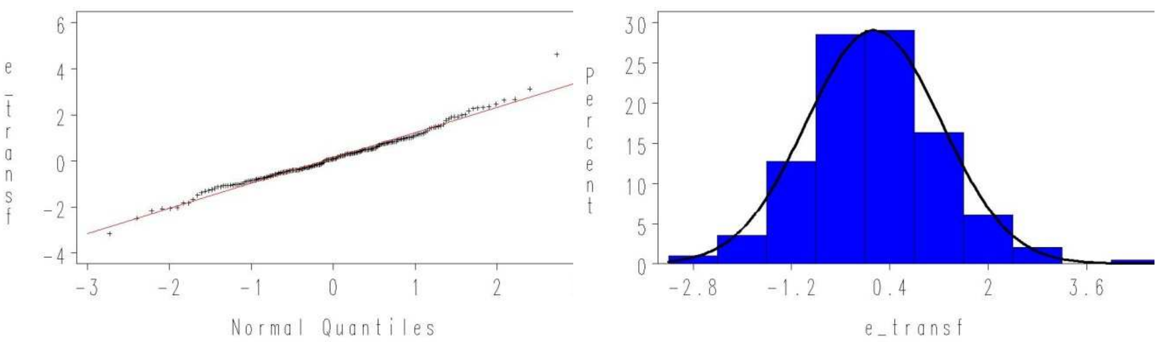
A12.1 Gráfico Q-Q e histograma dos efeitos aleatórios de domínio transformados



A12.2 Gráfico Q-Q e histograma dos efeitos aleatórios de domínio-tempo transformados



A12.3 Gráfico Q-Q e histograma dos erros da sondagem transformados



A12.4 Medidas de assimetria e curtose

Medida	Efeitos aleatórios de domínio	Efeitos aleatórios de domínio-tempo	Erros da sondagem
Assimetria	-1,102	-0,434	0,413
Curtose	2,713	1,259	3,391

Apêndice 13 – Medidas da qualidade dos estimadores do grupo A (*L*=100)

A13.1 Enviesamento

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=1**							
111	-19,1	105,9	-64,1	-58,6	-42,8	-19,8	-26,1	-28,9
112	-18,0	77,6	-92,7	-34,6	-11,7	-1,0	-0,2	-11,4
113	6,8	98,0	-72,2	-3,0	-7,0	-7,4	1,2	0,2
114	-22,0	62,0	-107,3	-40,8	-61,0	-42,2	-21,1	-18,6
115	3,3	239,9	69,7	13,6	62,8	37,9	30,6	13,6
116	-2,5	109,5	-60,5	-24,5	-37,8	-53,5	-6,4	-9,9
117	-28,6	63,4	-107,0	-52,8	-78,7	-80,7	-54,1	-42,0
118	-23,7	47,3	-123,1	-39,2	-31,5	-3,0	-31,4	-30,8
161	-19,7	-58,3	-83,6	-38,7	-54,9	-36,6	-16,5	-28,2
162	0,9	-2,7	5,2	0,1	-50,0	-52,0	18,7	4,0
163	-5,9	50,5	23,3	-3,7	-33,2	-35,8	-2,3	2,9
164	-2,0	-39,0	-88,2	-30,7	-112,7	-46,2	-54,0	-11,0
165	-28,3	-56,0	-96,9	-60,2	-92,3	-60,0	-44,7	-35,8
166	0,5	320,9	277,2	2,3	51,6	15,3	12,4	8,9
167	-4,6	182,2	135,6	-0,7	22,0	-1,7	6,3	4,5
168	-0,5	122,2	57,8	9,1	35,9	85,0	7,7	7,0
169	-45,1	-119,8	-160,9	-72,4	-191,8	-73,4	-123,4	-59,1
16A	-59,2	147,8	105,7	-22,4	-18,0	-53,6	-25,3	11,4
16B	-33,7	-196,1	66,9	-9,2	24,0	31,1	1,8	3,3
171	30,3	8,0	-14,1	4,4	-47,6	17,1	-27,3	7,2
172	-2,0	76,2	33,7	9,5	14,0	1,2	11,0	2,6
16C	3,4	-39,5	144,8	17,6	26,4	36,0	6,1	11,4
185	-19,6	-113,6	74,2	-2,4	20,6	-20,8	18,8	15,6
181	-71,7	-87,3	-132,4	-89,8	-98,6	7,4	-54,5	-91,9
182	-26,1	25,4	-39,0	-26,6	-95,8	-38,8	-63,8	-27,7
183	-5,3	-150,1	-202,8	-39,2	-145,3	-93,4	-69,6	-73,9
184	-19,6	-133,4	-209,4	-37,8	-94,9	-20,7	-30,0	-35,7
150	32,9	37,6	74,4	54,2	59,3	59,5	49,6	31,0

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
				***t*=2**				
111	-3,6	29,5	-119,5	-71,3	-98,9	-85,8	-57,6	-57,7
112	-23,6	-37,6	-194,6	-42,6	-82,5	-76,5	-39,6	-24,2
113	4,9	103,0	-51,2	-5,3	5,3	4,9	17,1	0,1
114	-2,1	119,8	-11,3	-6,5	-0,2	14,1	16,0	0,3
115	5,9	125,1	-31,1	-3,4	-6,3	-29,4	26,2	-4,2
116	-3,7	115,7	-35,6	-16,0	-15,3	-25,8	-0,8	-9,4
117	0,8	102,0	-54,9	-12,8	-20,3	-26,7	-3,8	-2,6
118	-20,2	-51,4	-211,3	-35,6	-86,8	-65,1	-59,8	-41,5
161	-27,6	-86,0	-99,1	-48,0	-75,4	-60,0	-55,2	-31,7
162	4,8	49,2	81,7	28,8	0,9	-1,6	23,5	15,7
163	0,1	104,1	94,3	9,3	11,4	2,7	8,6	3,5
164	-4,5	58,9	16,2	-6,1	-21,8	26,8	-52,9	1,9
165	-20,0	24,9	-6,8	-5,0	-15,9	5,4	-15,2	-3,8
166	1,6	266,1	211,1	4,1	25,9	3,5	24,5	7,5
167	-1,4	186,8	143,3	2,6	25,6	2,5	17,2	5,1
168	-21,6	73,1	5,7	-32,9	7,5	57,0	-59,8	-18,7
169	-0,8	68,4	43,2	48,1	-16,3	88,3	-59,2	29,3
16A	-21,1	222,5	185,1	21,6	52,6	16,6	20,6	51,5
16B	-15,1	-252,8	24,7	6,3	-15,4	-0,1	-6,7	5,5
171	26,9	14,5	-3,4	2,5	-68,8	-13,7	-39,7	1,2
172	13,5	53,7	5,7	5,4	-28,4	-46,0	9,6	0,6
16C	-3,8	-127,8	76,0	18,2	-22,4	-14,4	-13,6	6,7
185	-22,4	-118,7	88,3	6,3	16,4	-24,0	34,0	12,7
181	-19,8	-100,7	-119,8	-61,0	-103,1	-15,1	-75,9	-90,8
182	66,8	12,6	-49,0	62,2	-106,8	-43,1	-98,9	-19,8
183	12,3	-124,0	-168,7	-43,4	-114,9	-58,3	-97,4	-56,4
184	-28,1	-15,3	-83,3	-47,1	1,0	62,6	-14,6	-22,3
150	28,5	32,1	28,5	23,4	-1,7	-5,3	22,1	15,4
NUTSIII				***t*=3**				
111	-12,9	18,9	-101,8	-69,2	-97,4	-86,5	-50,9	-52,4
112	-1,5	117,7	-9,0	-15,7	36,6	40,1	18,7	-1,5
113	11,7	77,4	-47,8	-1,9	-4,9	-7,7	5,2	0,6
114	-1,2	80,3	-26,3	-6,2	-11,9	2,3	-0,5	0,6
115	16,7	113,7	-15,4	11,6	5,6	-10,9	18,9	15,0
116	-3,5	101,8	-21,8	-5,0	-12,7	-20,7	-1,1	-2,2
117	-3,4	134,2	5,3	-2,2	10,8	7,2	7,4	3,8
118	-7,7	-23,5	-153,7	-16,8	-38,5	-27,2	-7,9	-5,4
161	-3,1	-44,5	-37,7	-7,8	-26,2	-8,4	-9,0	-4,8
162	9,6	71,9	119,8	47,9	34,8	36,3	33,5	26,1
163	-7,7	63,0	66,9	11,2	-7,2	-8,7	5,0	6,5
164	-9,8	-37,9	-58,2	-15,2	-68,4	-24,5	-15,2	-0,9
165	0,8	58,8	46,5	9,9	14,9	38,7	2,5	2,1
166	-158,6	-82,8	-116,3	-160,1	-234,1	-255,8	-127,2	-121,5
167	1,7	144,3	117,7	6,9	8,3	-8,1	5,5	3,2
168	0,4	-47,4	-69,9	-5,4	-30,9	-20,1	-5,8	-2,5
169	-55,6	-15,5	-22,4	-48,4	-89,0	23,7	-30,4	-11,5
16A	-71,4	166,1	158,4	-60,1	2,7	-35,4	-5,7	-15,7
16B	-12,7	-284,3	9,4	-2,6	-30,8	-15,9	-9,7	1,6
171	21,7	-1,7	0,6	8,8	-49,3	8,9	-11,4	11,5
172	27,7	107,7	85,2	45,9	40,7	25,3	30,3	19,0
16C	-2,7	-146,9	74,8	14,5	-21,7	-13,1	2,4	4,3
185	22,0	35,6	161,7	45,1	81,0	44,5	36,1	37,2
181	-260,8	-507,5	-651,6	-287,6	-606,0	-505,5	-437,3	-444,7
182	-12,7	248,0	97,6	13,4	16,7	69,7	19,1	24,3
183	11,7	32,7	-116,3	-45,3	-69,3	-8,7	-28,2	-29,3
184	-7,0	-43,3	-196,2	-40,9	-97,2	-34,4	-14,2	-31,4
150	12,9	18,1	-46,9	-33,2	-81,8	-97,6	10,8	-27,2

NUTSIII	$\widehat{\mu}_i^{dir}$	$\widehat{\mu}_i^{sinQ}$	$\widehat{\mu}_i^{sinR}$	$\widehat{\mu}_i^{com}$	$\widehat{\mu}_i^{FH}$	$\widehat{\mu}_i^{NS}$	$\widehat{\mu}_{it}^{RY}$	$\widehat{\mu}_{it}^{LP}$
				***t*=4**				
111	4,9	97,8	-54,8	-55,7	-40,2	-35,9	-42,3	-32,1
112	-6,1	147,3	-17,7	-4,1	39,8	37,2	6,3	-1,4
113	16,3	157,0	-5,0	11,5	43,6	35,1	11,5	5,5
114	-6,0	81,7	-41,8	-16,6	-22,3	-11,2	-8,9	-6,1
115	-1,1	160,6	-9,7	-12,4	17,1	-2,1	6,6	-1,4
116	-0,5	199,2	41,1	5,3	40,6	25,7	8,0	3,0
117	3,6	96,4	-73,7	-3,2	-20,3	-27,9	-0,8	-1,8
118	-1,6	62,7	-113,6	-8,2	1,2	13,1	-3,8	-3,1
161	-31,5	-66,6	-89,9	-60,7	-64,3	-47,3	-40,7	-33,0
162	3,1	32,2	72,9	22,2	-13,0	-19,6	17,8	11,4
163	-5,2	107,7	82,0	24,3	22,5	16,6	17,0	12,6
164	-9,0	-29,8	-88,7	-22,6	-66,9	-25,7	-17,3	-5,2
165	-16,0	6,1	-40,9	-42,4	-43,7	-18,9	-19,0	-31,3
166	-0,2	301,4	237,6	1,9	45,7	19,3	8,3	6,1
167	-1,8	141,1	65,0	4,2	-0,5	-13,8	3,4	1,8
168	-8,1	5,2	-62,6	-22,1	-16,9	11,5	-26,1	-15,1
169	-2,8	2,3	-36,4	-8,7	-77,8	29,4	-67,7	-3,0
16A	-53,5	204,1	162,2	-25,7	27,6	-16,6	6,6	11,0
16B	-26,0	-244,6	34,9	6,2	-3,2	2,6	-4,0	9,0
171	12,5	13,0	20,2	17,7	-51,3	0,7	-13,6	6,6
172	37,9	100,4	65,4	41,9	20,9	3,6	50,4	27,5
16C	-1,6	-160,7	21,7	0,3	-54,3	-45,4	-22,7	-3,6
185	-4,8	111,8	114,0	39,0	49,9	6,5	42,8	27,3
181	-328,3	-405,6	-692,3	-324,3	-628,3	-526,6	-464,5	-475,8
182	19,1	381,1	94,1	65,7	29,7	64,2	16,9	48,4
183	1,6	85,1	-201,9	-26,0	-138,0	-89,6	-40,0	-41,5
184	-8,4	193,9	-93,3	-28,5	-4,4	45,5	1,7	-13,3
150	30,5	31,0	104,1	56,3	51,1	35,9	41,2	27,8
NUTSIII				***t*=5**				
111	-17,1	174,6	-9,9	-10,8	21,3	21,9	7,8	-1,3
112	-12,0	27,4	-172,3	-59,0	-74,2	-76,4	-57,5	-70,1
113	9,4	145,9	-51,1	-10,9	26,5	18,7	9,0	0,3
114	-9,7	106,9	-41,7	-16,4	-16,9	-9,3	-4,5	-4,8
115	-8,7	172,0	-31,5	-15,5	14,7	-3,8	0,0	-3,5
116	-9,1	200,6	10,6	-1,2	28,2	11,2	2,8	-0,5
117	0,8	145,8	-60,7	-7,3	-1,6	-7,5	1,1	-1,7
118	-4,4	112,8	-97,5	-9,0	18,2	26,2	-1,9	-2,8
161	-5,2	26,4	25,5	10,4	41,6	46,2	0,9	3,1
162	-4,6	82,2	150,6	31,3	61,3	49,3	28,9	25,9
163	-7,0	32,7	23,9	2,9	-19,5	-31,9	-13,2	-0,4
164	-18,0	59,3	7,5	-12,1	-11,0	42,1	-69,1	-3,3
165	-11,1	7,7	-22,0	-22,7	-11,3	3,3	-26,0	-15,5
166	-13,1	238,8	169,1	-11,6	3,3	-9,3	-5,4	-6,4
167	0,1	185,3	132,0	4,5	35,5	9,0	5,3	3,9
168	13,9	-45,5	-107,6	-0,5	-29,5	-4,7	-21,7	-5,2
169	3,5	-44,2	-69,4	-18,9	-95,2	8,7	-61,7	-23,3
16A	-21,7	257,1	219,4	-5,3	88,5	44,6	25,4	25,9
16B	-14,3	-248,9	5,7	1,8	-10,9	-5,4	5,0	0,5
171	15,2	-19,0	-89,7	-16,6	-118,3	-48,6	-27,3	-10,9
172	33,5	157,8	64,3	43,2	29,1	7,0	49,8	33,8
16C	-25,5	-134,7	51,4	-6,3	-31,5	-36,2	-9,7	4,6
185	-22,9	14,6	77,8	8,8	42,8	10,5	30,5	11,3
181	-222,7	-275,7	-437,7	-232,7	-370,0	-284,7	-308,5	-307,2
182	-4,3	124,1	-73,9	-19,9	-69,3	-27,0	-19,8	-8,0
183	16,8	211,1	25,6	17,5	74,6	130,0	37,1	2,9
184	-9,9	8,1	-188,9	-44,7	-66,3	-10,7	-43,3	-39,6
150	8,3	8,0	61,5	29,4	20,3	-2,7	45,1	17,1

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
				***t*=6**				
111	6,5	146,7	-32,8	-21,2	-7,9	-15,2	1,1	-14,0
112	-1,7	36,0	-147,0	-41,8	-59,9	-62,6	-12,3	-26,9
113	2,8	121,1	-61,2	-15,2	12,3	1,3	18,9	-3,9
114	-7,2	142,7	-31,2	-11,9	9,0	12,2	2,4	1,9
115	0,4	156,5	-27,3	-5,3	9,7	-5,5	2,5	0,1
116	-5,2	198,1	17,2	-3,1	24,6	5,0	4,6	0,2
117	-1,5	193,6	9,6	0,8	14,7	10,4	3,5	0,6
118	40,5	-3,9	-188,6	30,1	-46,8	-38,1	-12,3	-3,6
161	-22,7	-26,0	-54,8	-36,8	-18,4	-7,1	-19,6	-19,7
162	5,9	-6,6	5,4	5,7	-29,4	-41,0	-19,0	2,3
163	-1,4	142,1	130,5	26,2	61,9	27,5	21,2	12,3
164	-17,0	126,4	74,4	10,0	17,6	71,4	-48,9	2,3
165	-40,9	22,9	-13,9	-34,2	-2,1	9,3	-53,6	-31,1
166	-1,2	328,5	277,9	0,7	52,8	21,0	11,5	7,6
167	2,2	240,8	198,1	4,9	60,3	24,5	10,9	5,5
168	-2,8	-94,7	-156,6	-19,5	-74,8	-45,5	-48,9	-17,3
169	-6,7	-66,4	-112,2	-54,3	-129,5	-12,9	-104,8	-42,5
16A	-23,7	279,9	235,3	30,5	112,9	68,9	47,4	65,7
16B	-1,5	-368,8	-86,1	-30,3	-66,0	-62,3	-28,0	-14,3
171	16,5	-1,2	-13,4	-2,7	-29,8	-6,4	-11,0	1,9
172	27,8	86,9	42,0	29,3	43,5	9,3	34,7	21,5
16C	25,7	-105,8	137,5	45,4	33,5	21,7	3,0	16,9
185	-3,3	-163,0	48,3	-0,5	18,5	-17,0	13,3	3,7
181	1,0	77,4	34,9	-7,0	102,2	157,4	50,4	-5,5
182	10,4	47,7	-39,6	-23,1	-53,2	-0,1	-57,2	-16,2
183	7,9	-50,3	-126,5	-39,6	-55,1	0,8	-28,4	-30,3
184	15,0	-92,9	-171,8	-20,0	-55,1	-11,6	-26,2	-30,0
150	8,7	4,1	-104,8	-63,0	-85,0	-112,1	-44,8	-47,3
NUTSIII				***t*=7**				
111	-21,0	199,4	24,0	-25,3	49,7	34,7	-19,1	-3,2
112	2,1	233,9	57,7	17,7	104,7	87,4	25,9	7,2
113	-9,4	124,6	-51,8	-20,2	24,3	11,0	0,7	-1,8
114	16,3	93,7	-79,9	-11,3	-16,6	-15,6	-29,9	-13,6
115	1,2	215,6	38,9	12,4	65,6	27,8	25,8	4,5
116	-9,8	217,2	41,2	4,5	48,5	24,8	1,5	-2,7
117	9,0	166,9	-10,2	6,3	15,5	6,7	4,3	-0,3
118	-5,6	24,2	-153,2	-7,7	-5,6	-1,4	-2,5	-1,7
161	-30,7	-24,3	-20,3	-27,3	13,1	7,0	-9,8	-15,7
162	-0,1	6,3	36,7	9,2	14,0	3,8	9,2	3,3
163	-8,6	76,0	73,5	13,4	29,8	8,9	1,2	6,0
164	-10,9	54,3	28,8	-1,5	-15,9	26,9	-16,7	-1,6
165	-13,5	-106,3	-122,4	-35,0	-72,3	-66,1	-24,9	-16,9
166	2,5	349,1	328,6	4,0	85,8	37,4	12,4	9,0
167	-4,6	140,1	100,2	1,2	19,1	-3,6	6,1	2,8
168	8,4	-52,4	-84,8	2,0	-24,8	-7,6	-1,0	0,0
169	30,8	-69,5	-93,0	-19,6	-105,2	7,5	-63,2	-13,2
16A	-13,0	103,2	89,5	38,8	-4,5	-66,8	1,4	15,5
16B	-2,0	-289,4	53,5	21,9	55,0	37,4	-4,1	6,8
171	11,2	-2,2	32,8	17,4	16,5	22,7	-3,1	6,6
172	31,6	56,4	53,2	37,8	67,8	16,9	38,8	25,3
16C	-9,6	-216,3	86,2	16,7	1,6	-33,4	-14,9	12,0
185	-30,7	-95,0	78,1	-6,4	47,8	-5,3	2,1	3,6
181	-94,7	-177,4	-320,0	-113,1	-213,2	-163,2	-132,6	-157,1
182	-2,3	121,4	-30,7	0,2	-42,9	0,0	-71,6	-13,8
183	3,8	105,9	-40,7	-15,2	19,2	43,8	23,3	-12,8
184	24,3	-169,4	-321,9	-1,1	-183,5	-128,6	-87,9	-95,4
150	3,9	3,1	-12,3	-11,3	13,3	-38,2	38,8	-7,8

A13.2 EQM

NUTSIII	$\hat{\boldsymbol{\mu}}_i^{dir}$	$\hat{\boldsymbol{\mu}}_i^{sinQ}$	$\hat{\boldsymbol{\mu}}_i^{sinR}$	$\hat{\boldsymbol{\mu}}_i^{com}$	$\hat{\boldsymbol{\mu}}_i^{FH}$	$\hat{\boldsymbol{\mu}}_i^{NS}$	$\hat{\boldsymbol{\mu}}_{it}^{RY}$	$\hat{\boldsymbol{\mu}}_{it}^{LP}$
	***t*=1**							
111	13.224	13.683	4.336	8.074	2.520	2.399	2.898	2.978
112	6.121	8.089	8.883	6.148	869	1.567	1.139	1.305
113	2.073	11.901	5.455	2.135	214	279	182	180
114	4.264	7.908	11.935	5.596	4.725	2.506	1.629	1.540
115	5.427	59.813	5.103	5.064	5.596	2.858	2.212	1.239
116	6.993	14.573	3.874	5.098	1.847	3.525	1.296	1.483
117	10.982	6.025	11.744	9.304	6.612	7.015	4.833	4.372
118	6.006	4.095	15.512	6.712	2.621	2.349	3.884	3.946
161	5.723	4.504	7.208	4.916	3.736	2.859	1.244	1.941
162	11.099	1.672	579	4.467	2.955	3.760	2.803	2.574
163	6.161	3.629	757	3.371	1.374	1.761	1.237	1.524
164	7.899	2.301	8.054	6.693	15.433	2.995	4.321	751
165	21.504	4.020	9.621	15.430	10.562	6.931	7.157	7.189
166	1.799	103.851	77.061	1.831	3.257	379	166	81
167	1.371	34.010	18.641	1.436	661	6	46	21
168	2.542	15.539	3.789	2.539	2.325	10.539	835	823
169	14.259	15.227	26.109	16.475	41.218	8.910	22.531	7.327
16A	20.122	22.726	11.413	20.641	2.551	7.721	6.733	8.650
16B	10.112	39.295	4.946	7.833	1.072	2.557	2.537	2.928
171	7.588	3.534	1.858	4.016	3.822	1.510	2.450	1.687
172	2.819	8.402	1.919	1.798	767	771	1.036	806
16C	4.614	2.118	21.186	5.345	1.001	1.862	768	928
185	9.111	15.330	5.800	8.641	1.659	2.654	3.031	2.951
181	54.383	12.656	17.957	29.917	11.158	23.278	6.114	10.986
182	17.048	4.279	1.734	11.295	10.515	4.034	7.254	4.948
183	14.941	26.977	41.415	15.369	22.274	10.413	6.245	6.868
184	5.163	20.695	44.063	6.206	10.832	2.151	2.709	3.204
150	6.473	5.908	6.235	5.824	4.229	4.996	4.502	2.632

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
				t=2				
111	22.063	1.491	14.586	16.567	10.434	9.237	4.764	4.935
112	5.325	1.914	38.346	7.990	8.804	8.491	3.482	1.928
113	2.792	11.152	3.032	2.515	349	409	957	576
114	1.862	15.304	716	982	353	636	976	514
115	5.746	16.160	1.440	4.844	1.181	2.334	2.396	1.731
116	4.302	13.977	1.606	2.594	615	1.295	793	1.002
117	1.675	10.896	3.513	1.577	608	1.024	291	251
118	5.759	3.095	45.279	6.848	9.489	6.212	5.917	4.320
161	11.794	8.154	10.098	9.363	6.708	5.536	4.807	3.845
162	5.008	3.549	7.484	5.080	419	985	1.395	1.313
163	1.963	11.625	9.168	2.310	360	304	312	253
164	4.094	4.031	699	2.311	1.798	1.490	4.225	657
165	12.074	1.251	376	7.159	984	1.825	1.449	1.518
166	1.600	71.312	45.184	1.649	880	303	922	59
167	1.415	35.447	20.973	1.554	893	7	521	27
168	19.628	5.768	908	16.611	4.640	10.249	9.564	9.584
169	10.436	5.356	2.156	6.367	2.804	10.656	6.606	2.296
16A	10.045	50.109	34.645	13.077	4.203	4.498	1.933	5.591
16B	7.148	64.394	1.115	3.834	786	1.495	842	1.442
171	10.948	7.337	3.104	4.719	6.677	1.474	2.976	1.843
172	8.103	8.074	1.261	2.952	1.513	3.108	1.559	1.725
16C	4.984	16.695	6.054	3.911	810	482	556	771
185	11.551	16.339	8.176	10.428	1.819	2.804	4.442	3.700
181	33.126	15.367	15.107	22.668	12.639	24.231	8.332	10.939
182	42.977	3.435	2.684	33.985	12.527	4.172	12.486	3.373
183	8.204	19.384	28.773	12.928	14.946	5.700	13.020	6.196
184	10.722	3.259	7.251	8.063	683	6.129	1.892	2.313
150	10.272	7.801	1.934	3.690	653	1.225	1.605	1.934
NUTSIII				*t*=3				
111	24.314	1.133	10.662	15.132	10.114	9.341	5.522	5.658
112	6.275	14.500	427	3.397	2.147	3.288	2.062	1.145
113	4.315	6.668	2.596	3.258	222	350	357	389
114	2.704	7.609	1.709	2.020	457	301	298	304
115	4.999	13.503	660	4.036	1.277	1.700	2.278	2.235
116	2.332	11.066	766	1.765	468	996	274	289
117	1.673	18.594	446	1.072	230	158	308	210
118	2.563	1.117	24.085	3.228	2.830	1.844	527	433
161	8.100	2.913	1.756	4.729	1.490	1.784	1.923	1.966
162	7.144	6.551	15.615	9.525	1.603	2.297	3.007	2.498
163	3.301	4.878	4.792	3.462	413	604	870	938
164	5.552	2.120	3.766	5.176	8.000	1.836	1.980	1.178
165	3.713	4.215	2.471	2.889	708	3.268	492	558
166	58.116	7.440	14.115	57.113	64.032	75.700	39.315	39.512
167	1.105	21.464	14.320	1.289	123	101	44	13
168	1.336	2.911	5.286	1.474	2.488	1.544	85	26
169	25.799	1.045	790	15.370	10.679	3.909	5.869	4.927
16A	16.841	28.393	25.374	14.462	1.332	5.565	2.813	3.519
16B	7.524	81.459	978	3.086	1.576	1.889	1.880	1.948
171	6.571	2.956	4.498	5.081	4.503	1.352	1.830	1.534
172	4.981	13.809	9.314	5.132	2.370	1.579	1.867	1.312
16C	5.750	22.032	5.903	5.261	1.123	666	787	839
185	15.623	24.692	26.688	15.369	8.216	4.328	4.826	5.020
181	240.664	310.655	425.642	242.107	371.513	283.413	214.266	225.528
182	11.448	99.072	9.859	11.707	1.170	7.142	3.332	3.688
183	9.586	41.913	13.945	8.742	5.919	2.191	2.144	2.015
184	7.146	34.106	38.794	10.462	11.364	2.915	2.011	3.160
150	22.903	20.048	4.575	11.540	7.448	10.679	5.522	3.844

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=4**							
111	18.277	10.998	3.277	9.333	2.230	3.090	3.952	3.334
112	5.003	22.879	535	2.425	2.314	2.851	896	894
113	5.341	25.893	243	3.404	2.277	1.630	926	997
114	2.767	8.771	2.866	2.096	973	519	631	603
115	4.924	26.868	350	3.671	1.862	1.327	1.882	1.600
116	4.647	40.991	1.919	3.364	2.211	1.163	1.034	901
117	1.421	10.382	5.678	1.550	581	1.114	3	4
118	2.409	4.909	13.228	2.907	383	674	388	367
161	9.512	5.375	8.387	8.569	4.710	3.821	3.887	3.427
162	4.811	2.471	6.801	4.226	585	1.277	1.651	1.312
163	3.555	12.514	7.013	3.952	1.011	981	1.570	1.477
164	4.181	1.587	8.108	4.558	7.860	1.692	1.611	578
165	11.423	811	1.891	5.282	2.529	2.632	3.090	3.348
166	1.438	91.498	56.722	1.470	2.464	596	106	49
167	1.693	20.503	4.581	1.673	35	253	20	4
168	6.676	674	4.207	6.626	2.395	2.932	4.674	4.128
169	11.361	833	1.555	6.457	9.351	4.025	10.366	2.702
16A	15.403	42.460	26.526	14.971	2.163	4.393	3.909	4.501
16B	9.604	60.512	2.268	5.977	683	1.728	1.835	1.894
171	6.769	4.216	4.888	5.472	4.964	1.371	2.207	2.263
172	7.291	13.049	6.256	5.639	1.303	1.064	4.730	2.290
16C	5.161	26.306	742	3.468	3.939	2.930	2.022	1.930
185	13.825	67.025	13.584	13.573	3.477	2.037	3.822	2.388
181	316.140	275.678	480.315	278.389	400.894	307.848	249.082	271.787
182	32.203	235.199	9.294	28.679	1.652	5.855	3.807	6.588
183	7.206	100.688	41.263	9.775	21.141	10.665	3.860	4.121
184	7.405	117.074	8.994	6.848	742	3.879	1.016	1.594
150	7.194	6.160	13.130	7.909	3.365	2.408	3.371	2.144
NUTSIII	***t*=5**							
111	9.055	32.066	344	4.575	860	1.929	1.476	1.446
112	21.494	2.043	29.936	18.977	6.080	7.280	5.194	6.714
113	3.984	22.623	2.827	3.131	1.063	725	752	692
114	3.209	13.816	3.054	2.459	740	530	639	652
115	2.173	30.802	1.262	1.744	780	510	483	454
116	3.461	41.722	323	1.724	1.337	611	635	686
117	1.614	22.421	3.998	1.784	48	127	4	3
118	1.607	13.843	9.893	1.755	717	1.336	218	228
161	6.295	2.385	1.038	2.906	2.263	3.676	1.071	1.120
162	4.957	9.203	24.431	7.605	4.505	3.635	2.404	2.193
163	5.359	2.683	891	3.430	736	1.591	1.125	1.059
164	13.791	4.739	291	8.938	1.067	2.592	8.169	2.080
165	10.478	1.476	697	7.788	989	2.490	2.288	2.178
166	2.406	58.096	28.949	2.346	165	339	160	197
167	1.593	35.565	17.672	1.763	1.789	106	36	17
168	5.848	3.207	11.876	6.449	2.877	2.315	3.960	2.835
169	20.829	3.410	5.041	14.837	12.528	3.909	9.065	4.841
16A	4.837	67.430	48.327	4.106	9.281	5.130	1.188	1.243
16B	8.975	63.200	1.097	3.967	860	1.821	1.687	2.229
171	4.911	3.781	11.867	5.409	17.180	3.909	2.427	1.661
172	10.297	27.761	6.414	7.899	1.967	1.346	5.490	3.580
16C	9.215	19.129	3.012	7.206	1.683	2.051	1.706	1.997
185	7.843	26.879	6.457	6.927	2.781	1.914	2.403	1.584
181	170.585	138.778	192.883	158.359	144.044	111.598	122.948	129.700
182	2.650	55.594	5.689	2.732	6.233	1.788	755	440
183	10.189	92.014	1.064	6.644	7.372	21.438	5.468	2.980
184	6.770	40.858	35.919	10.360	5.459	1.922	4.842	3.652
150	7.301	5.980	6.588	5.284	1.181	1.120	4.624	2.032

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	t=6							
111	14.298	22.780	1.289	6.468	536	1.775	1.516	1.374
112	7.516	2.255	21.905	9.064	4.089	5.295	1.245	1.844
113	4.705	15.682	4.009	3.515	402	346	983	958
114	2.775	22.188	1.623	2.070	486	618	467	481
115	1.860	25.394	1.119	1.586	462	434	251	261
116	3.252	40.361	510	1.962	1.194	464	551	549
117	1.277	38.376	475	1.065	504	264	58	17
118	7.475	841	36.033	6.514	3.086	2.341	1.268	1.218
161	6.962	1.501	3.217	4.662	810	1.506	1.485	1.692
162	8.025	1.292	752	3.823	1.421	2.769	1.859	1.960
163	4.371	21.199	17.351	6.351	4.501	1.420	1.874	1.637
164	7.416	16.612	5.850	6.494	1.013	6.787	4.730	1.378
165	18.440	1.279	409	10.271	1.229	2.998	5.761	5.504
166	1.256	108.525	77.517	1.273	3.786	672	149	60
167	1.795	58.665	39.469	1.867	4.929	766	137	30
168	3.221	9.517	24.961	4.998	8.338	4.357	5.358	1.714
169	16.063	5.087	12.856	15.320	20.105	3.903	16.329	5.779
16A	9.601	79.034	55.602	18.543	14.950	9.258	6.338	10.882
16B	4.981	136.592	7.892	5.154	5.529	6.114	1.988	1.509
171	5.925	3.701	3.693	4.402	2.886	1.481	1.345	1.487
172	6.926	10.120	3.030	3.911	2.959	1.320	2.884	2.094
16C	4.148	11.654	19.186	6.427	1.685	963	576	1.322
185	7.013	27.778	2.591	5.558	1.138	1.707	1.449	1.217
181	35.229	9.786	2.258	23.953	14.972	51.535	10.368	6.573
182	11.137	4.286	1.792	8.955	3.760	1.370	4.359	1.353
183	8.782	4.927	16.225	8.331	3.825	2.646	1.845	2.157
184	7.473	10.942	29.733	8.113	4.072	2.238	1.754	1.905
150	10.706	9.657	12.269	8.778	8.110	13.810	3.408	3.574
NUTSIII	*t*=7							
111	15.678	45.224	826	7.738	3.472	3.386	2.381	2.336
112	6.239	59.264	3.526	4.890	13.673	10.610	2.915	1.686
113	3.960	19.836	2.892	3.497	864	476	650	731
114	8.563	16.500	7.065	7.528	1.130	1.156	3.259	2.776
115	3.629	50.582	1.729	3.111	5.763	1.848	2.376	1.165
116	3.550	51.926	1.898	2.926	3.447	1.378	829	841
117	1.512	31.517	362	1.105	377	88	22	0
118	1.477	4.021	23.757	1.565	105	77	17	9
161	8.724	1.321	723	2.796	911	1.914	2.008	2.379
162	4.015	990	2.039	2.032	772	1.060	1.323	1.141
163	3.871	6.462	5.662	3.963	1.425	660	755	894
164	5.357	3.465	1.038	4.115	853	1.635	1.382	721
165	7.227	11.895	15.178	9.201	7.310	7.646	2.299	1.744
166	1.698	122.450	108.208	1.725	9.485	1.873	163	84
167	1.233	20.056	10.314	1.395	573	15	41	8
168	2.380	3.227	7.414	2.516	1.498	1.115	322	328
169	11.838	5.366	8.850	9.684	13.411	3.481	6.915	3.046
16A	5.528	11.245	8.224	3.765	830	9.586	538	679
16B	8.144	84.210	3.641	6.376	3.479	2.979	1.269	1.629
171	4.090	2.132	4.671	3.833	1.667	1.813	830	860
172	6.615	4.636	4.293	5.112	5.784	1.671	3.319	2.452
16C	5.857	47.195	7.911	5.998	584	2.020	1.436	1.508
185	6.677	19.148	6.531	5.934	2.927	1.493	979	884
181	52.387	58.730	103.340	57.049	53.442	54.184	35.578	45.911
182	16.692	31.112	1.165	10.300	3.606	2.098	9.729	3.440
183	9.028	33.686	2.172	5.462	1.136	3.994	3.096	2.389
184	11.500	44.733	103.844	11.938	36.374	19.712	14.512	15.420
150	9.899	9.812	1.811	4.215	1.460	2.898	5.241	2.092

A13.3 Variância

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=1**							
111	12.861	2.459	221	4.634	685	2.008	2.217	2.141
112	5.797	2.066	285	4.950	733	1.566	1.139	1.174
113	2.026	2.306	239	2.126	164	224	180	180
114	3.782	4.066	425	3.929	1.001	727	1.183	1.194
115	5.416	2.259	246	4.879	1.652	1.425	1.277	1.055
116	6.987	2.581	213	4.498	417	664	1.255	1.386
117	10.164	2.006	301	6.521	424	503	1.910	2.610
118	5.445	1.856	347	5.174	1.627	2.340	2.895	2.999
161	5.333	1.109	217	3.415	720	1.520	972	1.148
162	11.098	1.665	552	4.467	459	1.060	2.453	2.558
163	6.127	1.082	213	3.357	274	482	1.231	1.515
164	7.895	780	280	5.749	2.724	861	1.405	630
165	20.706	887	230	11.807	2.035	3.335	5.158	5.904
166	1.799	849	244	1.825	595	143	12	2
167	1.350	812	261	1.435	175	3	6	1
168	2.541	599	454	2.456	1.038	3.309	775	774
169	12.229	885	231	11.235	4.431	3.524	7.294	3.838
16A	16.619	871	235	20.142	2.227	4.845	6.095	8.520
16B	8.974	832	472	7.748	496	1.593	2.534	2.917
171	6.669	3.471	1.659	3.997	1.555	1.216	1.706	1.635
172	2.815	2.596	781	1.707	572	769	914	799
16C	4.602	559	211	5.037	306	567	731	797
185	8.726	2.419	300	8.636	1.234	2.223	2.676	2.708
181	49.239	5.036	417	21.852	1.428	23.222	3.143	2.537
182	16.369	3.632	214	10.588	1.331	2.529	3.188	4.180
183	14.913	4.459	302	13.834	1.164	1.691	1.401	1.403
184	4.781	2.901	236	4.776	1.825	1.722	1.810	1.928
150	5.390	4.497	698	2.884	717	1.461	2.037	1.671

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	t=2							
111	22.050	623	297	11.488	662	1.883	1.449	1.603
112	4.768	497	495	6.173	2.002	2.641	1.910	1.343
113	2.768	538	407	2.486	321	385	666	576
114	1.857	961	589	940	353	437	719	514
115	5.711	507	471	4.833	1.141	1.469	1.708	1.713
116	4.288	584	336	2.338	380	628	792	914
117	1.675	497	495	1.413	194	310	277	245
118	5.352	453	623	5.578	1.958	1.968	2.342	2.598
161	11.030	760	275	7.056	1.027	1.931	1.763	2.841
162	4.985	1.126	802	4.248	418	983	841	1.065
163	1.963	784	281	2.223	230	296	239	241
164	4.074	561	437	2.274	1.321	773	1.428	654
165	11.676	632	330	7.134	730	1.796	1.216	1.504
166	1.597	487	621	1.632	209	290	321	2
167	1.413	556	447	1.547	237	0	223	1
168	19.159	418	875	15.530	4.583	6.994	5.993	9.233
169	10.435	675	293	4.051	2.539	2.865	3.095	1.436
16A	9.601	595	379	12.610	1.434	4.223	1.509	2.941
16B	6.919	503	504	3.794	550	1.495	796	1.412
171	10.222	7.127	3.092	4.713	1.947	1.287	1.398	1.841
172	7.922	5.194	1.228	2.923	705	995	1.468	1.725
16C	4.969	366	276	3.581	310	274	371	726
185	11.049	2.258	372	10.388	1.549	2.228	3.284	3.538
181	32.733	5.228	762	18.944	2.005	24.002	2.569	2.690
182	38.509	3.276	283	30.119	1.120	2.316	2.703	2.981
183	8.052	4.000	322	11.044	1.747	2.296	3.531	3.016
184	9.930	3.024	320	5.845	682	2.207	1.680	1.816
150	9.457	6.773	1.122	3.142	651	1.197	1.119	1.697
NUTSIII	*t*=3							
111	24.148	776	294	10.341	630	1.863	2.932	2.909
112	6.273	636	347	3.151	807	1.681	1.713	1.143
113	4.177	670	314	3.255	198	290	329	388
114	2.702	1.160	1.018	1.981	314	295	298	304
115	4.721	586	424	3.902	1.245	1.582	1.923	2.011
116	2.319	708	292	1.740	306	566	273	284
117	1.661	590	417	1.067	114	107	254	196
118	2.504	563	473	2.947	1.350	1.103	463	403
161	8.090	932	335	4.668	802	1.713	1.841	1.943
162	7.052	1.376	1.272	7.226	391	980	1.882	1.818
163	3.241	904	314	3.336	362	529	845	896
164	5.456	686	373	4.946	3.326	1.237	1.749	1.177
165	3.712	755	307	2.790	486	1.772	486	554
166	32.968	580	580	31.483	9.229	10.282	23.144	24.739
167	1.102	635	455	1.242	54	35	14	3
168	1.336	667	399	1.445	1.531	1.140	51	20
169	22.707	804	289	13.030	2.750	3.346	4.944	4.795
16A	11.749	795	290	10.855	1.324	4.315	2.780	3.274
16B	7.362	634	889	3.079	628	1.637	1.785	1.945
171	6.098	2.953	4.497	5.004	2.075	1.273	1.700	1.401
172	4.217	2.219	2.053	3.021	715	936	948	950
16C	5.742	458	310	5.051	651	495	782	820
185	15.138	23.423	548	13.334	1.662	2.346	3.523	3.634
181	172.640	53.070	1.089	159.403	4.255	27.921	23.027	27.742
182	11.286	37.571	335	11.529	893	2.284	2.967	3.095
183	9.448	40.845	425	6.692	1.122	2.116	1.349	1.159
184	7.096	32.228	290	8.790	1.916	1.734	1.808	2.172
150	22.735	19.721	2.379	10.440	755	1.160	5.405	3.102

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
				t=4				
111	18.253	1.428	274	6.232	615	1.803	2.162	2.303
112	4.965	1.182	222	2.408	729	1.469	856	892
113	5.077	1.239	218	3.271	372	398	794	967
114	2.731	2.102	1.120	1.821	478	394	552	566
115	4.923	1.084	255	3.517	1.571	1.323	1.839	1.598
116	4.647	1.317	229	3.336	566	504	970	891
117	1.407	1.088	253	1.540	168	334	3	1
118	2.407	978	333	2.840	382	502	374	357
161	8.521	933	311	4.886	575	1.584	2.228	2.338
162	4.801	1.435	1.481	3.733	416	895	1.335	1.182
163	3.528	916	292	3.362	505	705	1.280	1.318
164	4.099	700	241	4.046	3.381	1.031	1.312	551
165	11.168	774	218	3.483	624	2.273	2.728	2.366
166	1.438	671	264	1.467	379	222	37	12
167	1.690	599	356	1.656	34	62	8	1
168	6.610	647	288	6.139	2.110	2.801	3.995	3.899
169	11.353	828	229	6.381	3.296	3.163	5.787	2.693
16A	12.541	807	222	14.309	1.400	4.117	3.867	4.380
16B	8.926	689	1.053	5.939	673	1.721	1.819	1.813
171	6.613	4.046	4.479	5.157	2.328	1.371	2.023	2.220
172	5.854	2.977	1.984	3.882	866	1.051	2.185	1.534
16C	5.158	474	270	3.468	994	871	1.505	1.917
185	13.802	54.535	589	12.053	987	1.994	1.989	1.642
181	208.329	111.185	982	173.243	6.120	30.544	33.361	45.414
182	31.840	89.941	442	24.365	768	1.738	3.522	4.244
183	7.203	93.441	513	9.101	2.091	2.640	2.262	2.398
184	7.335	79.486	285	6.035	723	1.807	1.013	1.419
150	6.261	5.199	2.299	4.742	749	1.115	1.673	1.371
NUTSIII				*t*=5				
111	8.761	1.584	246	4.457	405	1.447	1.416	1.444
112	21.350	1.293	234	15.499	569	1.439	1.883	1.801
113	3.896	1.344	216	3.011	362	377	671	691
114	3.115	2.381	1.314	2.188	455	443	619	629
115	2.097	1.227	271	1.505	565	496	483	442
116	3.377	1.474	211	1.723	541	486	628	686
117	1.614	1.174	312	1.731	45	71	3	0
118	1.588	1.108	382	1.674	387	649	214	221
161	6.267	1.688	390	2.797	536	1.545	1.070	1.111
162	4.936	2.443	1.743	6.623	744	1.200	1.571	1.522
163	5.310	1.611	319	3.421	354	571	951	1.059
164	13.466	1.223	235	8.792	945	819	3.390	2.068
165	10.356	1.417	215	7.274	862	2.479	1.611	1.938
166	2.233	1.078	348	2.212	154	253	130	156
167	1.593	1.210	241	1.743	530	24	8	1
168	5.655	1.138	290	6.449	2.005	2.293	3.491	2.809
169	20.817	1.457	227	14.481	3.456	3.833	5.253	4.296
16A	4.365	1.345	207	4.078	1.453	3.140	541	573
16B	8.769	1.252	1.064	3.964	741	1.792	1.662	2.229
171	4.680	3.420	3.819	5.132	3.182	1.543	1.683	1.542
172	9.175	2.870	2.286	6.032	1.122	1.297	3.005	2.440
16C	8.564	982	375	7.166	691	740	1.611	1.976
185	7.317	26.665	404	6.849	953	1.804	1.473	1.457
181	121.000	62.743	1.316	104.232	7.145	30.534	27.748	35.347
182	2.631	40.188	223	2.334	1.427	1.059	364	376
183	9.906	47.469	410	6.338	1.814	4.526	4.092	2.971
184	6.671	40.793	231	8.360	1.064	1.807	2.969	2.087
150	7.233	5.917	2.807	4.421	770	1.112	2.586	1.740

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
				***t*=6**				
111	14.255	1.254	212	6.020	473	1.544	1.515	1.177
112	7.513	958	303	7.314	496	1.378	1.094	1.118
113	4.697	1.013	264	3.283	250	345	626	943
114	2.723	1.823	649	1.929	405	470	461	478
115	1.860	887	371	1.558	367	403	245	261
116	3.225	1.132	216	1.953	590	438	530	549
117	1.275	878	382	1.065	288	155	46	16
118	5.834	825	450	5.609	900	889	1.116	1.205
161	6.448	826	211	3.308	471	1.456	1.102	1.305
162	7.990	1.248	724	3.790	554	1.085	1.498	1.955
163	4.369	993	317	5.666	675	664	1.426	1.486
164	7.128	624	318	6.393	705	1.689	2.338	1.373
165	16.766	753	215	9.100	1.225	2.912	2.885	4.534
166	1.254	636	303	1.272	1.001	231	16	1
167	1.790	702	240	1.843	1.297	166	18	1
168	3.213	548	450	4.616	2.740	2.285	2.968	1.414
169	16.018	676	261	12.366	3.344	3.738	5.353	3.976
16A	9.040	685	252	17.612	2.194	4.507	4.090	6.568
16B	4.979	542	481	4.235	1.175	2.228	1.204	1.306
171	5.653	3.699	3.513	4.394	1.999	1.440	1.225	1.484
172	6.152	2.560	1.269	3.052	1.064	1.234	1.682	1.630
16C	3.490	459	271	4.370	560	490	567	1.038
185	7.002	1.198	260	5.558	797	1.417	1.271	1.203
181	35.228	3.791	1.044	23.904	4.531	26.748	7.825	6.543
182	11.029	2.006	222	8.422	926	1.370	1.091	1.091
183	8.720	2.399	228	6.766	791	2.645	1.036	1.240
184	7.247	2.302	215	7.711	1.035	2.103	1.065	1.002
150	10.630	9.640	1.288	4.805	892	1.239	1.402	1.338
NUTSIII				***t*=7**				
111	15.237	5.476	250	7.097	1.002	2.180	2.016	2.326
112	6.235	4.576	200	4.578	2.718	2.970	2.247	1.634
113	3.873	4.310	204	3.089	276	355	650	728
114	8.296	7.719	686	7.401	855	912	2.362	2.592
115	3.628	4.111	214	2.957	1.463	1.074	1.711	1.145
116	3.454	4.756	203	2.906	1.098	764	827	833
117	1.431	3.661	258	1.066	138	43	3	0
118	1.445	3.437	292	1.507	74	75	11	6
161	7.783	732	313	2.052	738	1.865	1.913	2.132
162	4.015	950	696	1.948	576	1.046	1.238	1.130
163	3.796	682	260	3.783	537	581	754	857
164	5.238	522	206	4.113	600	909	1.102	718
165	7.044	585	203	7.973	2.081	3.279	1.679	1.460
166	1.691	554	200	1.709	2.130	478	10	3
167	1.212	431	279	1.394	207	2	4	0
168	2.310	478	230	2.512	885	1.057	321	328
169	10.890	534	203	9.299	2.344	3.424	2.918	2.872
16A	5.359	602	208	2.259	809	5.119	536	438
16B	8.140	458	779	5.895	450	1.580	1.252	1.583
171	3.964	2.127	3.595	3.531	1.394	1.296	821	817
172	5.615	1.453	1.460	3.682	1.192	1.385	1.818	1.811
16C	5.765	391	477	5.719	581	907	1.215	1.363
185	5.736	10.114	427	5.892	638	1.465	974	871
181	43.414	27.271	930	44.247	7.970	27.543	17.988	21.221
182	16.686	16.365	220	10.300	1.769	2.098	4.606	3.249
183	9.014	22.481	517	5.232	767	2.075	2.553	2.227
184	10.910	16.033	214	11.937	2.704	3.183	6.782	6.311
150	9.884	9.803	1.661	4.087	1.283	1.437	3.734	2.032

A13.4 Erro absoluto

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=1**							
111	91	106	64	75	44	37	40	41
112	63	78	93	67	23	32	26	28
113	35	98	72	37	11	13	10	10
114	52	72	107	64	61	43	31	30
115	59	240	70	58	67	45	41	29
116	69	110	61	61	39	54	28	30
117	93	65	107	86	79	81	59	52
118	64	52	123	69	40	39	50	51
161	59	60	84	59	56	41	26	33
162	85	32	20	44	51	56	42	39
163	65	52	24	44	33	36	29	33
164	68	42	88	68	113	46	54	21
165	123	57	97	109	96	73	73	72
166	33	321	277	34	52	15	12	9
167	30	182	136	30	22	2	6	4
168	40	122	58	43	39	86	24	24
169	94	120	161	109	192	79	128	69
16A	118	148	106	132	32	61	57	82
16B	83	196	67	75	28	43	40	43
171	69	46	34	46	52	32	39	33
172	43	80	37	34	23	22	26	23
16C	54	41	145	59	28	37	21	24
185	77	114	74	79	34	40	44	44
181	203	94	132	140	99	136	59	93
182	114	53	39	84	97	51	74	55
183	104	151	203	105	145	93	72	76
184	55	134	209	64	95	36	40	44
150	64	61	74	70	59	62	55	43

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=2**							
111	114	32	120	107	99	86	59	59
112	55	39	195	68	83	77	43	31
113	45	103	51	43	15	17	25	19
114	36	120	20	24	15	21	24	19
115	63	125	32	56	26	42	42	33
116	54	116	36	42	21	29	21	24
117	32	102	55	33	21	27	13	13
118	60	52	211	65	89	68	66	53
161	85	86	99	84	76	62	59	48
162	55	52	82	62	16	25	30	30
163	36	104	94	39	15	14	14	12
164	50	59	21	36	30	33	54	20
165	81	30	16	50	26	34	31	31
166	34	266	211	35	26	12	25	8
167	29	187	143	30	26	3	17	5
168	116	73	24	95	50	89	81	71
169	79	68	43	61	44	91	75	34
16A	78	223	185	98	60	60	36	70
16B	66	253	28	44	24	30	22	31
171	86	70	44	55	70	31	45	34
172	74	73	28	39	32	47	30	34
16C	52	128	76	52	25	19	20	20
185	86	119	88	88	35	40	56	50
181	152	107	120	125	103	140	78	91
182	167	47	49	132	107	52	106	49
183	73	125	169	95	117	63	101	67
184	74	47	83	70	21	64	29	33
150	81	66	37	47	21	29	32	36
NUTSIII	***t*=3**							
111	129	27	102	111	97	86	60	61
112	65	118	17	41	41	47	36	27
113	49	77	48	43	11	13	14	14
114	43	81	35	37	17	14	13	14
115	56	114	20	47	26	33	34	32
116	40	102	23	34	18	25	13	13
117	33	134	17	25	12	9	13	11
118	40	27	154	45	40	29	14	12
161	73	47	38	55	34	34	36	36
162	69	73	120	83	35	39	43	40
163	47	64	67	50	16	20	24	25
164	59	40	58	58	73	33	36	28
165	48	59	47	45	20	43	18	19
166	194	83	116	192	234	256	138	134
167	27	144	118	29	9	8	6	3
168	29	48	70	31	31	21	6	3
169	122	26	23	78	95	48	55	50
16A	97	166	158	91	28	46	38	38
16B	67	284	25	37	35	34	35	35
171	62	43	54	57	56	30	35	32
172	57	109	86	60	42	35	37	30
16C	62	147	75	62	29	21	20	22
185	95	130	162	103	83	55	56	55
181	424	508	652	415	606	505	437	445
182	89	249	98	98	25	76	48	52
183	71	172	116	78	70	38	38	37
184	67	162	196	80	97	45	30	39
150	121	112	58	81	82	98	57	49

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=4**							
111	104	98	55	75	40	43	52	46
112	60	147	19	36	41	42	23	24
113	58	157	12	41	44	36	25	26
114	42	84	46	38	25	17	18	18
115	60	161	14	47	38	29	37	32
116	53	199	41	47	42	29	26	24
117	31	96	74	32	20	28	1	2
118	39	64	114	44	16	22	17	16
161	79	67	90	82	65	49	50	46
162	56	41	73	55	20	29	33	29
163	49	108	82	54	27	25	33	32
164	44	33	89	48	68	28	22	14
165	89	23	41	58	45	43	49	50
166	30	301	238	30	46	19	8	6
167	33	141	65	34	4	14	3	2
168	67	21	63	72	36	50	61	52
169	81	23	36	60	89	50	89	38
16A	96	204	162	103	43	44	52	58
16B	73	245	39	57	21	33	36	35
171	67	51	56	58	58	31	38	39
172	69	102	67	64	30	26	60	41
16C	59	161	23	47	58	49	37	35
185	94	216	114	100	53	38	52	40
181	464	429	692	433	628	527	465	477
182	144	381	94	140	33	69	47	69
183	65	287	202	75	138	90	49	50
184	62	259	93	68	21	48	25	30
150	69	63	104	75	52	40	47	37
NUTSIII	***t*=5**							
111	78	175	15	45	25	37	31	31
112	123	39	172	117	74	77	60	71
113	48	146	51	46	29	23	21	21
114	47	107	46	41	21	17	19	19
115	37	172	32	33	23	19	18	18
116	46	201	14	28	32	21	20	20
117	31	146	61	34	5	8	1	2
118	32	113	98	34	22	30	12	12
161	64	39	27	41	42	50	26	26
162	58	84	151	73	62	50	38	37
163	57	42	25	44	23	34	28	27
164	89	61	13	56	23	46	74	33
165	77	30	23	60	24	39	36	35
166	38	239	169	38	9	12	8	9
167	33	185	132	34	35	9	5	4
168	63	49	108	68	45	37	57	44
169	110	49	69	99	104	48	87	55
16A	51	257	219	49	90	67	31	32
16B	75	249	27	44	24	34	33	38
171	55	48	92	57	119	52	40	33
172	84	158	68	77	37	29	64	52
16C	77	135	51	67	36	40	33	37
185	71	156	78	70	47	37	41	31
181	344	303	438	330	370	287	312	314
182	41	184	74	43	69	31	22	16
183	84	219	27	58	78	133	63	44
184	64	194	189	78	67	33	54	47
150	70	61	67	60	28	28	55	36

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
				t=6				
111	81	147	33	50	18	33	32	30
112	72	39	147	80	60	64	27	36
113	55	121	61	50	17	14	25	24
114	43	143	34	37	18	21	17	18
115	34	157	29	32	16	16	11	11
116	46	198	19	34	30	18	18	18
117	27	194	17	25	15	11	5	3
118	67	24	189	61	51	43	26	24
161	63	32	55	54	23	32	29	31
162	70	29	22	42	32	46	33	33
163	53	142	131	65	62	31	36	34
164	62	126	74	61	27	72	53	28
165	113	29	17	66	25	43	62	58
166	27	328	278	27	53	21	12	8
167	34	241	198	35	60	24	11	5
168	46	95	157	58	80	53	63	34
169	97	66	112	113	134	51	118	65
16A	82	280	235	122	117	90	75	96
16B	53	369	86	57	67	64	35	30
171	57	48	49	52	43	31	30	31
172	69	89	45	50	47	30	47	39
16C	48	106	138	63	36	25	19	27
185	71	163	48	63	29	32	30	28
181	151	81	39	111	104	182	95	64
182	83	52	40	68	55	27	58	28
183	80	60	126	79	56	43	36	40
184	68	95	172	74	58	36	34	37
150	85	81	105	81	85	112	47	49
NUTSIII				*t*=7				
111	95	199	25	65	55	49	38	37
112	61	234	58	59	107	91	44	33
113	50	125	52	50	26	18	21	22
114	74	97	80	73	27	27	49	41
115	46	216	39	46	70	36	39	28
116	49	217	41	47	53	32	23	23
117	31	167	15	25	15	7	4	0
118	31	49	153	31	6	5	3	2
161	74	28	22	37	25	37	36	39
162	52	26	38	36	22	25	28	27
163	51	76	74	54	34	22	22	24
164	52	55	29	42	20	32	23	18
165	58	106	122	73	75	71	38	33
166	34	349	329	34	86	37	12	9
167	28	140	100	30	19	4	6	3
168	39	53	85	41	29	23	14	14
169	89	70	93	84	107	47	76	47
16A	53	103	90	51	21	72	18	21
16B	72	289	54	65	55	47	28	33
171	51	36	57	49	33	35	23	23
172	66	58	56	58	70	35	50	41
16C	62	216	86	67	17	39	31	32
185	61	113	78	61	49	32	26	25
181	181	199	320	188	218	189	152	177
182	107	131	31	76	53	32	87	46
183	78	138	41	60	28	52	46	41
184	92	177	322	94	187	134	108	115
150	79	80	34	48	31	45	55	37

Apêndice 14 – Medidas da qualidade dos estimadores do grupo A (*L*=500)

A14.1 Enviesamento

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	t=1							
111	-16,7	108,1	-61,2	-58,0	-42,8	-16,7	-24,4	-28,1
112	-6,0	79,6	-88,8	-24,6	-10,5	1,6	0,8	-10,4
113	3,6	100,1	-68,9	-5,5	-8,3	-9,0	-0,3	-1,2
114	-18,3	64,8	-107,9	-36,8	-61,5	-41,9	-19,7	-17,3
115	-1,2	242,0	73,1	10,3	60,7	35,0	24,6	9,0
116	1,0	111,8	-57,9	-25,4	-35,8	-52,0	-4,4	-9,2
117	-30,3	65,4	-102,9	-53,3	-77,5	-79,8	-54,5	-42,5
118	-23,5	49,2	-118,6	-39,2	-32,0	-2,5	-30,0	-28,8
161	-9,7	-53,8	-81,9	-34,9	-49,4	-28,6	-5,4	-16,7
162	4,7	2,8	3,8	0,9	-48,1	-49,8	19,6	4,0
163	-12,1	54,9	25,2	-6,5	-33,5	-36,1	-4,2	1,1
164	1,9	-35,3	-84,3	-22,5	-113,3	-48,9	-52,9	-13,1
165	-1,8	-52,0	-93,8	-38,3	-90,6	-53,0	-38,3	-26,8
166	0,4	324,8	280,5	2,3	52,6	15,6	12,2	8,7
167	0,8	186,0	139,2	4,9	23,0	-1,5	6,1	4,4
168	-2,1	125,5	63,1	5,6	38,5	89,7	6,1	4,9
169	-43,6	-115,8	-157,7	-68,8	-187,6	-72,5	-106,6	-46,8
16A	-53,4	151,8	108,9	-18,9	-17,3	-52,7	-24,7	11,2
16B	-20,6	-192,3	65,9	4,5	22,1	33,0	2,9	5,5
171	9,5	-3,9	-19,4	-7,7	-50,4	13,2	-30,5	1,4
172	-5,4	66,0	31,3	5,8	12,6	-2,3	10,0	2,1
16C	3,1	-36,3	146,8	16,8	28,8	40,0	6,8	13,0
185	-8,0	-111,1	74,5	9,9	19,2	-20,9	16,4	12,7
181	-37,6	-83,6	-133,0	-76,0	-96,5	14,0	-61,2	-94,6
182	-25,8	28,6	-37,2	-30,4	-95,3	-40,5	-60,7	-28,8
183	-8,1	-146,6	-202,4	-39,9	-144,1	-92,6	-67,7	-71,0
184	-13,7	-130,6	-206,1	-32,6	-94,3	-21,7	-29,7	-34,7
150	12,7	16,1	72,3	38,2	56,3	52,3	38,0	20,9

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
				***t*=2**				
111	-2,9	30,7	-117,3	-70,9	-98,2	-82,4	-56,7	-54,6
112	-13,8	-36,5	-189,4	-31,4	-82,0	-74,9	-40,2	-19,9
113	3,1	104,2	-47,1	-7,9	3,7	3,0	15,7	-2,2
114	2,9	121,3	-15,6	-2,3	-0,5	14,4	16,3	0,2
115	3,9	126,2	-26,2	-1,7	-5,6	-29,9	29,3	-2,6
116	-5,5	116,9	-32,5	-15,7	-13,4	-24,1	2,4	-7,6
117	-0,9	103,1	-49,8	-12,7	-20,4	-27,2	-6,0	-3,7
118	-14,1	-50,3	-205,1	-30,5	-89,1	-66,5	-62,9	-43,0
161	-33,1	-83,7	-98,6	-51,8	-72,4	-54,3	-54,5	-32,9
162	-0,4	52,0	76,0	26,5	1,4	-1,1	21,9	14,7
163	1,2	106,4	94,3	11,4	12,1	2,5	10,1	3,9
164	-6,4	60,9	20,7	-6,6	-24,5	23,2	-56,0	-1,1
165	-12,9	27,0	-3,8	0,7	-13,3	12,4	-11,4	-0,4
166	0,9	268,0	217,3	3,4	27,3	4,8	26,3	7,5
167	-0,3	188,7	147,9	3,5	26,5	2,5	18,5	5,1
168	12,8	74,8	13,7	2,1	14,5	66,3	-49,4	-7,5
169	1,1	70,6	45,3	43,6	-15,3	86,6	-56,4	28,2
16A	-21,3	224,5	188,9	20,7	53,4	17,1	21,1	51,0
16B	-7,0	-250,9	21,2	9,4	-17,3	2,0	-8,0	3,7
171	11,6	9,6	-17,9	-12,9	-72,9	-18,8	-45,9	-3,6
172	12,5	49,5	-2,3	-3,3	-30,2	-50,4	6,5	1,0
16C	6,7	-126,2	77,3	24,7	-20,2	-11,0	-11,5	8,4
185	-12,4	-118,9	86,3	13,6	17,1	-22,7	40,0	18,7
181	-13,1	-101,0	-125,3	-61,9	-99,2	-7,2	-84,2	-92,6
182	56,7	12,4	-47,3	49,5	-105,0	-43,2	-92,6	-13,9
183	1,5	-124,3	-169,8	-51,3	-114,1	-58,0	-99,8	-59,5
184	-18,9	-15,5	-80,5	-42,1	2,0	61,2	-18,3	-22,4
150	16,2	15,1	21,0	12,7	-4,2	-12,1	13,3	7,1
NUTSIII				***t*=3**				
111	-10,8	23,6	-102,6	-80,6	-96,8	-83,5	-47,8	-47,5
112	5,8	122,0	-8,6	-9,4	37,1	41,8	18,5	-0,3
113	5,4	81,8	-47,7	-7,6	-6,3	-9,8	2,6	-3,3
114	1,1	86,1	-29,8	-5,4	-12,7	2,2	-2,3	-0,9
115	29,4	117,7	-14,5	21,9	6,9	-10,1	20,4	16,5
116	-4,8	106,3	-22,0	-6,5	-12,5	-20,6	-0,7	-2,0
117	-0,5	138,3	6,2	-0,1	9,4	5,6	3,2	0,2
118	-2,6	-19,5	-152,6	-9,7	-39,4	-27,4	-8,5	-6,3
161	-9,7	-45,4	-38,9	-14,6	-23,5	-2,8	-9,5	-6,4
162	8,3	70,8	115,8	46,1	34,1	35,8	29,0	23,6
163	-4,6	62,2	65,9	14,4	-7,2	-9,0	2,9	4,1
164	-14,8	-38,6	-57,7	-21,4	-72,9	-29,2	-19,3	-5,8
165	5,1	58,0	46,6	16,9	17,8	46,4	5,5	5,4
166	-170,7	-83,5	-114,9	-172,1	-234,8	-256,5	-117,3	-111,3
167	1,2	143,6	118,7	6,3	9,2	-7,7	5,4	3,1
168	1,4	-48,1	-69,2	-4,1	-26,6	-16,5	-4,9	-2,1
169	-63,4	-16,4	-22,7	-54,0	-89,5	21,6	-35,8	-20,2
16A	-68,6	165,3	158,1	-57,7	3,8	-35,0	-6,2	-16,5
16B	-16,2	-285,0	6,2	-5,8	-33,0	-14,0	-11,9	-1,5
171	7,5	-15,4	-7,1	0,2	-53,9	3,8	-19,4	3,9
172	6,9	95,8	80,0	28,9	37,1	19,9	24,3	13,1
16C	-0,4	-147,5	73,8	16,8	-22,1	-11,9	1,7	3,7
185	24,7	42,7	159,4	52,3	80,7	45,4	36,5	39,2
181	-249,1	-496,8	-655,2	-281,3	-595,6	-491,7	-421,6	-429,4
182	-16,2	257,0	96,3	15,2	15,8	67,2	14,1	17,3
183	11,4	42,1	-118,1	-45,3	-68,1	-7,8	-25,9	-27,8
184	-7,0	-35,0	-196,5	-40,0	-96,2	-35,1	-11,7	-28,9
150	13,4	13,7	-52,5	-34,6	-83,3	-103,9	9,1	-28,5

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	t=4							
111	3,2	100,0	-54,6	-62,5	-40,6	-34,2	-43,9	-30,9
112	3,0	149,3	-15,4	-3,0	41,6	39,7	6,6	-0,5
113	3,5	159,1	-3,2	4,7	42,3	32,7	6,9	0,9
114	-5,5	84,4	-46,4	-16,5	-23,6	-12,1	-7,5	-4,7
115	-3,5	162,5	-6,5	-11,1	17,4	-2,8	7,4	-1,3
116	6,7	201,3	42,3	11,9	41,6	26,4	11,0	5,8
117	2,0	98,4	-70,5	-4,8	-20,1	-28,3	-0,6	-1,6
118	0,6	64,5	-109,4	-5,7	1,5	13,7	-3,8	-3,0
161	-21,6	-65,0	-90,1	-51,0	-60,6	-41,1	-31,7	-25,6
162	0,5	34,2	67,1	21,1	-13,0	-19,9	16,2	9,5
163	-7,7	109,3	82,0	21,9	21,7	15,5	12,6	8,1
164	-10,0	-28,3	-85,8	-23,3	-70,0	-29,5	-19,6	-7,4
165	-9,1	7,6	-39,0	-38,8	-38,5	-8,9	-6,3	-19,8
166	-2,8	302,8	240,9	-0,7	46,8	19,9	8,1	5,8
167	0,4	142,4	69,4	7,7	0,3	-13,1	3,4	1,8
168	1,9	6,5	-58,9	-12,5	-11,1	17,4	-20,5	-10,0
169	-3,0	3,9	-35,3	-3,0	-78,4	26,9	-69,1	-5,8
16A	-54,9	205,6	163,6	-28,6	28,8	-16,5	4,6	9,2
16B	-18,0	-243,2	30,5	7,3	-6,0	3,5	-9,5	3,8
171	3,3	6,1	8,0	7,6	-55,5	-3,9	-18,9	4,3
172	15,1	94,4	58,2	22,6	16,0	-2,8	36,8	16,4
16C	-0,2	-159,5	22,0	1,6	-51,6	-41,5	-16,6	1,2
185	-0,8	112,6	111,7	42,4	49,6	7,2	38,3	25,3
181	-314,0	-404,4	-696,4	-322,2	-616,3	-511,9	-442,3	-447,2
182	28,9	382,2	92,7	66,9	30,0	62,2	18,7	49,8
183	-13,2	86,2	-203,7	-44,9	-136,4	-89,3	-39,6	-39,7
184	-5,3	194,9	-93,2	-32,8	-2,5	44,1	1,6	-12,3
150	11,8	13,0	96,1	37,6	47,8	28,1	32,7	19,4
NUTSIII	*t*=5							
111	-13,5	172,0	-10,4	-7,8	21,3	23,9	8,1	-1,0
112	-16,4	25,0	-171,0	-65,3	-72,1	-73,6	-55,0	-66,3
113	14,3	143,5	-50,1	-5,6	25,5	16,8	8,8	0,4
114	-5,7	103,7	-46,6	-16,7	-18,5	-10,7	-5,0	-5,1
115	-10,3	169,7	-29,7	-15,7	12,8	-6,2	-0,4	-4,1
116	-5,1	198,1	10,7	3,8	28,9	11,5	5,0	1,3
117	-0,5	143,5	-58,6	-7,7	-1,0	-7,5	1,0	-1,6
118	-0,6	110,7	-94,9	-5,3	18,5	26,6	-1,9	-2,7
161	-10,9	29,0	23,8	7,6	44,9	51,2	2,2	3,8
162	-6,5	85,4	144,8	28,6	61,7	49,0	27,1	24,2
163	-10,0	35,3	22,7	-2,5	-19,8	-33,1	-13,6	-0,6
164	-30,7	61,5	8,8	-24,9	-12,9	38,3	-70,9	-6,7
165	-3,1	10,1	-22,0	-14,2	-5,6	12,8	-19,4	-6,6
166	-14,3	240,9	171,5	-12,8	3,7	-9,3	-5,5	-6,5
167	0,9	187,6	133,4	5,2	36,2	8,7	5,4	3,9
168	19,7	-43,3	-105,7	4,4	-25,0	-0,1	-17,8	-1,4
169	8,3	-41,7	-69,6	-16,8	-96,0	5,2	-64,0	-26,1
16A	-22,0	259,4	219,8	-6,2	90,2	45,2	24,3	24,7
16B	-8,2	-246,6	1,5	1,6	-11,8	-2,6	6,3	3,0
171	2,6	-29,7	-98,9	-31,0	-122,5	-52,2	-32,6	-14,8
172	6,9	148,0	57,4	17,7	24,1	0,0	36,8	20,5
16C	-17,2	-132,7	49,7	-1,8	-32,3	-36,1	-12,6	1,3
185	-18,2	15,6	76,0	17,1	42,2	11,2	26,6	8,6
181	-225,7	-274,2	-442,5	-234,7	-360,0	-273,6	-292,7	-289,9
182	-0,9	125,3	-74,1	-16,4	-69,0	-27,5	-18,2	-6,0
183	6,4	212,4	23,8	6,0	75,0	129,4	35,7	2,1
184	-8,3	9,3	-189,2	-43,9	-64,8	-11,7	-41,7	-38,5
150	7,6	8,9	53,8	25,7	18,1	-10,2	41,4	13,2

NUTSIII	$\widehat{\mu}_i^{dir}$	$\widehat{\mu}_i^{sinQ}$	$\widehat{\mu}_i^{sinR}$	$\widehat{\mu}_i^{com}$	$\widehat{\mu}_i^{FH}$	$\widehat{\mu}_i^{NS}$	$\widehat{\mu}_{it}^{RY}$	$\widehat{\mu}_{it}^{LP}$
				***t*=6**				
111	5,5	141,2	-33,8	-28,6	-7,3	-13,1	2,6	-11,6
112	6,6	31,2	-145,6	-32,8	-58,6	-60,4	-13,0	-27,8
113	2,9	116,2	-60,3	-18,5	11,1	-1,1	18,8	-4,7
114	-5,1	136,1	-36,0	-13,1	7,3	10,4	0,0	-0,5
115	-3,9	151,9	-25,3	-7,7	8,8	-7,7	1,5	-1,4
116	-4,8	192,8	17,1	-1,2	24,0	3,8	2,7	-2,4
117	1,5	189,0	11,7	3,0	15,3	10,3	3,5	0,4
118	25,5	-8,4	-186,1	15,8	-46,6	-38,2	-15,2	-6,9
161	-9,6	-24,3	-55,7	-29,5	-14,3	-1,0	-13,3	-13,5
162	8,5	-4,6	0,3	-0,7	-27,9	-40,2	-15,8	2,7
163	-2,3	144,0	127,8	27,0	63,2	26,6	25,8	16,6
164	-12,6	127,9	75,9	14,3	18,6	70,6	-45,4	4,6
165	-23,1	24,5	-14,0	-27,5	3,8	18,6	-43,1	-15,7
166	-0,1	329,9	279,3	1,8	53,3	20,6	11,8	7,7
167	3,4	242,3	198,6	6,1	60,9	23,4	11,3	5,5
168	-0,7	-93,3	-154,0	-18,7	-72,5	-42,1	-50,1	-16,8
169	-10,3	-64,9	-111,3	-48,6	-130,0	-15,9	-108,5	-46,1
16A	-11,2	281,4	236,0	44,0	113,8	68,8	47,5	67,0
16B	-3,9	-367,5	-90,0	-31,5	-67,5	-60,3	-24,3	-10,3
171	5,1	-6,4	-25,5	-17,1	-33,6	-12,5	-15,6	0,4
172	5,5	82,6	34,9	12,2	38,7	1,7	27,2	14,0
16C	17,3	-104,6	135,4	34,7	31,6	20,3	1,2	13,9
185	-1,5	-166,6	46,3	7,2	19,4	-15,4	17,0	6,4
181	-2,2	71,0	28,5	-10,2	108,2	164,1	51,7	1,9
182	10,2	43,1	-39,4	-24,5	-53,1	-0,9	-56,9	-13,3
183	8,7	-55,4	-127,9	-38,3	-54,0	1,5	-30,7	-31,6
184	8,4	-98,0	-172,9	-23,2	-53,5	-12,8	-27,5	-30,3
150	3,7	2,8	-112,0	-65,6	-86,5	-119,4	-49,9	-50,3
NUTSIII				***t*=7**				
111	-10,7	186,8	22,4	-18,5	50,8	36,8	-17,9	-0,6
112	7,6	222,3	57,2	20,5	108,4	91,1	26,7	9,0
113	-2,8	113,4	-51,9	-16,7	22,0	7,6	-1,2	-3,5
114	-2,2	78,7	-84,2	-24,4	-18,6	-18,3	-32,6	-16,4
115	3,9	204,7	39,2	15,0	66,2	26,7	25,9	5,5
116	-13,7	205,4	40,5	1,7	49,2	24,5	2,6	-2,3
117	1,6	156,6	-9,3	-0,1	16,2	6,5	4,3	-0,2
118	-3,7	14,2	-151,9	-5,8	-5,1	-0,9	-2,5	-1,5
161	-18,7	-23,6	-22,5	-18,7	19,3	14,3	-2,6	-8,8
162	2,4	7,1	32,3	8,6	15,6	4,4	10,5	4,5
163	-7,6	76,7	71,7	13,5	29,7	7,5	1,9	6,6
164	-10,7	54,8	28,9	-0,4	-16,8	23,9	-17,7	-2,5
165	-16,2	-105,8	-123,1	-38,4	-69,6	-60,9	-25,6	-17,7
166	0,8	349,7	328,3	2,3	86,4	36,3	12,3	9,0
167	0,8	140,6	101,3	6,7	20,0	-3,5	6,2	2,8
168	6,1	-51,9	-84,2	-0,3	-22,0	-4,8	0,2	1,3
169	30,3	-68,9	-93,1	-20,3	-105,8	3,8	-70,4	-21,5
16A	-15,1	103,8	88,6	32,7	-4,4	-69,0	-0,9	14,9
16B	-6,4	-288,9	48,9	20,5	53,9	38,6	-2,2	8,9
171	4,1	-10,0	22,1	8,1	13,2	15,8	-6,8	4,2
172	1,1	50,0	46,6	16,1	62,5	8,0	29,0	14,4
16C	-0,7	-215,9	82,9	25,7	-0,9	-36,1	-16,9	8,3
185	-20,8	-94,9	75,1	2,8	47,9	-5,0	2,8	4,0
181	-102,8	-177,1	-325,2	-120,1	-206,3	-158,4	-128,3	-150,0
182	-3,3	121,6	-31,9	-4,5	-44,6	-2,9	-67,2	-7,9
183	4,7	106,1	-44,2	-13,6	22,1	44,0	26,6	-11,0
184	8,7	-169,2	-323,0	-18,0	-180,9	-128,6	-82,1	-90,4
150	10,1	9,2	-19,4	-11,8	13,6	-45,8	42,4	-6,8

A14.2 EQM

NUTSIII	$\widehat{\mu}_i^{dir}$	$\widehat{\mu}_i^{sinQ}$	$\widehat{\mu}_i^{sinR}$	$\widehat{\mu}_i^{com}$	$\widehat{\mu}_i^{FH}$	$\widehat{\mu}_i^{NS}$	$\widehat{\mu}_{it}^{RY}$	$\widehat{\mu}_{it}^{LP}$
	***t*=1**							
111	13.809	14.222	3.993	8.583	2.469	2.289	2.917	3.071
112	4.971	8.464	8.221	5.062	773	1.441	981	1.207
113	2.431	12.387	5.021	2.389	224	295	189	191
114	4.605	8.381	12.131	5.816	4.827	2.457	1.580	1.514
115	4.809	60.890	5.625	4.707	5.489	2.707	2.133	1.357
116	6.006	15.144	3.582	4.235	1.720	3.418	1.017	1.123
117	12.198	6.337	10.942	10.018	6.407	6.823	4.723	4.251
118	6.626	4.332	14.495	7.503	2.638	2.395	3.616	3.642
161	5.533	4.073	6.939	4.931	3.250	2.304	1.124	1.474
162	10.381	1.776	653	4.690	3.002	3.851	2.997	2.760
163	5.750	4.160	854	3.591	1.448	1.835	1.311	1.557
164	8.600	2.072	7.435	6.859	15.291	3.172	4.114	972
165	17.053	3.645	9.053	14.404	9.931	6.016	5.686	5.248
166	1.585	106.422	78.978	1.613	3.406	363	159	78
167	1.485	35.465	19.683	1.609	724	5	42	20
168	2.707	16.387	4.550	2.609	2.734	11.977	777	773
169	13.677	14.345	25.138	15.012	39.609	9.116	18.643	5.765
16A	20.002	23.963	12.133	21.063	2.570	8.060	6.844	8.819
16B	7.928	37.847	4.883	6.460	967	2.637	1.866	2.147
171	7.293	3.524	2.445	4.141	4.360	1.709	2.826	1.927
172	4.165	6.975	1.909	2.810	935	1.026	1.123	968
16C	3.433	1.913	21.773	4.198	1.215	2.325	745	955
185	7.729	14.579	5.874	6.813	1.441	2.470	2.735	2.728
181	44.790	11.660	18.169	26.350	11.432	27.052	7.088	11.799
182	15.293	4.185	1.605	10.031	10.616	4.495	7.247	5.356
183	14.028	25.623	41.301	14.036	22.159	10.933	6.370	6.866
184	4.926	19.745	42.747	6.097	10.615	2.461	2.622	3.059
150	6.952	5.610	6.051	5.673	4.042	4.285	3.394	2.105

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	t=2							
111	19.190	1.617	14.077	14.582	10.317	8.642	4.664	4.390
112	4.539	1.872	36.437	6.775	8.579	8.059	3.745	1.506
113	2.760	11.436	2.670	2.468	288	350	896	525
114	2.497	15.757	822	1.615	336	602	988	457
115	5.537	16.485	1.218	4.334	1.326	2.536	2.492	1.818
116	4.230	14.306	1.428	2.671	576	1.255	686	949
117	1.923	11.165	3.037	1.734	587	1.013	362	258
118	6.209	3.023	42.764	7.434	9.654	6.152	6.475	4.494
161	12.868	7.969	10.013	9.420	6.278	4.765	4.486	3.644
162	5.239	4.128	6.565	4.987	505	1.138	1.409	1.490
163	2.289	12.319	9.178	2.615	375	319	358	290
164	4.209	4.414	919	2.789	1.765	1.411	4.514	805
165	10.844	1.526	378	6.402	1.073	2.427	1.465	1.736
166	1.384	72.415	47.925	1.428	971	255	1.132	58
167	1.480	36.326	22.378	1.594	958	7	645	27
168	24.508	6.129	1.172	21.792	4.155	10.691	8.912	9.197
169	11.999	5.833	2.368	6.422	2.544	10.744	6.256	2.386
16A	10.787	51.167	36.109	14.306	4.291	4.936	2.184	5.708
16B	6.101	63.595	944	3.090	846	1.472	878	1.317
171	9.249	7.420	3.489	4.556	7.470	2.009	3.542	2.270
172	9.017	7.789	1.237	3.367	1.752	3.672	1.556	1.897
16C	4.698	16.390	6.268	4.009	782	490	591	740
185	10.348	16.416	7.819	9.489	1.893	2.815	4.767	3.725
181	36.232	15.494	16.454	25.896	12.748	27.671	9.719	11.820
182	36.391	3.470	2.538	27.226	12.527	4.689	12.483	4.185
183	10.922	19.501	29.167	15.574	14.813	6.128	14.301	6.485
184	7.466	3.304	6.823	5.935	743	6.095	2.033	2.183
150	8.716	6.839	1.561	3.588	845	1.490	1.259	1.637
NUTSIII	*t*=3							
111	20.868	1.424	10.836	14.246	10.032	8.845	5.021	5.051
112	6.840	15.596	419	3.711	2.232	3.458	2.122	1.337
113	4.757	7.443	2.590	3.499	268	428	495	629
114	2.570	8.703	1.940	2.031	480	283	347	351
115	6.139	14.518	626	4.977	1.454	1.868	2.383	2.332
116	2.382	12.085	785	1.710	460	998	265	294
117	1.851	19.781	446	1.261	185	105	185	144
118	2.475	1.010	23.746	2.815	2.740	1.719	522	438
161	7.241	3.133	1.875	4.148	1.369	1.606	1.915	2.022
162	7.869	6.600	14.707	9.513	1.654	2.408	2.844	2.525
163	3.172	4.902	4.673	3.313	401	595	744	784
164	5.096	2.282	3.696	4.788	8.047	1.873	1.727	1.140
165	4.403	4.233	2.480	3.568	937	4.559	769	826
166	60.719	7.645	13.755	59.698	63.170	74.654	36.305	36.510
167	1.472	21.345	14.542	1.673	148	88	45	13
168	1.500	3.083	5.179	1.624	1.988	1.221	70	22
169	27.744	1.192	816	17.598	10.424	4.244	6.944	6.449
16A	16.306	28.236	25.298	13.901	1.350	5.970	2.957	3.790
16B	8.602	81.976	961	4.015	1.686	1.724	1.953	2.082
171	5.974	3.314	4.536	4.321	5.123	1.625	2.386	1.872
172	4.651	11.484	8.482	4.854	2.347	1.604	1.873	1.396
16C	4.498	22.286	5.774	4.287	1.153	724	820	855
185	13.379	26.288	25.985	14.754	8.275	4.512	5.402	5.833
181	227.064	302.224	430.452	234.762	362.682	275.396	209.738	222.845
182	10.729	105.297	9.641	11.444	1.425	7.330	3.865	4.014
183	12.836	44.426	14.392	11.774	5.955	2.681	2.541	2.342
184	6.662	34.878	38.922	9.758	11.072	3.092	1.933	2.656
150	22.268	20.225	5.155	10.236	7.919	12.217	6.170	4.515

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=4**							
111	18.732	11.559	3.283	9.035	2.285	3.027	4.092	3.450
112	4.397	23.579	491	2.391	2.489	3.083	924	925
113	4.883	26.653	261	2.814	2.196	1.498	862	931
114	3.154	9.397	3.304	2.480	1.029	520	594	573
115	4.601	27.586	331	3.446	1.868	1.301	1.893	1.532
116	4.234	41.958	2.048	2.933	2.345	1.283	1.380	1.180
117	1.464	10.853	5.257	1.560	556	1.122	4	4
118	1.957	5.226	12.329	2.334	381	718	384	372
161	8.711	5.442	8.458	7.727	4.278	3.190	3.091	2.815
162	5.135	3.047	6.018	4.363	621	1.394	1.738	1.461
163	4.284	13.152	7.046	4.621	947	903	1.350	1.254
164	4.617	1.717	7.634	4.798	7.934	1.818	1.798	653
165	11.273	1.068	1.775	5.814	2.220	2.695	2.821	2.713
166	1.526	92.555	58.349	1.547	2.600	588	115	45
167	1.485	21.064	5.207	1.648	38	224	23	4
168	7.316	888	3.794	7.073	2.238	2.909	4.490	3.956
169	13.534	1.096	1.505	7.505	9.018	4.264	11.084	3.062
16A	15.491	43.340	27.021	14.920	2.246	4.837	4.107	4.679
16B	9.358	60.026	2.013	5.173	640	1.534	2.147	2.066
171	7.190	4.266	4.570	5.070	5.700	1.797	2.923	2.714
172	7.134	12.020	5.395	5.442	1.371	1.299	4.072	2.239
16C	6.721	26.072	787	5.080	3.891	2.862	2.439	2.362
185	11.081	67.424	13.098	10.493	3.492	2.090	3.796	2.770
181	302.309	275.137	486.041	283.025	390.459	298.895	240.245	256.280
182	30.565	236.368	9.070	26.086	1.899	5.999	4.377	7.152
183	7.878	101.237	42.041	12.258	20.883	11.102	4.762	4.360
184	5.166	117.774	9.008	5.839	741	3.809	1.062	1.517
150	6.609	4.907	11.562	6.945	3.392	2.259	3.412	2.287
NUTSIII	***t*=5**							
111	7.480	31.118	358	3.584	861	2.051	1.368	1.420
112	18.164	1.888	29.478	17.828	5.701	6.844	4.813	6.217
113	4.519	21.891	2.719	3.716	1.079	726	907	857
114	3.183	13.083	3.542	2.630	780	519	614	617
115	2.401	29.984	1.139	1.983	797	541	504	447
116	3.528	40.677	322	1.775	1.323	619	622	659
117	1.575	21.740	3.728	1.625	46	119	10	3
118	1.653	13.326	9.375	1.810	767	1.429	220	226
161	6.101	2.464	970	2.676	2.451	3.960	953	1.048
162	5.650	9.637	22.791	8.110	4.614	3.709	2.537	2.356
163	5.269	2.794	844	3.025	771	1.718	1.177	1.204
164	14.147	4.962	299	8.933	1.149	2.397	8.808	2.575
165	11.796	1.462	695	8.097	1.027	2.866	2.256	2.206
166	1.936	59.063	29.735	1.884	168	306	180	201
167	1.615	36.349	18.030	1.785	1.860	95	52	16
168	6.185	2.970	11.448	7.023	2.730	2.183	4.055	2.927
169	21.605	3.142	5.079	14.933	12.203	4.061	8.905	4.784
16A	5.368	68.592	48.530	4.674	9.688	5.720	1.282	1.294
16B	9.001	62.026	1.115	4.373	940	1.670	1.664	2.053
171	4.939	4.439	13.730	6.425	18.377	4.571	3.097	2.049
172	10.729	24.885	5.677	8.129	1.957	1.577	4.925	3.360
16C	8.372	18.552	2.864	6.321	1.685	2.096	1.997	2.273
185	7.426	27.658	6.200	6.661	2.743	1.925	2.234	1.583
181	174.679	139.719	197.221	161.517	140.907	110.157	119.391	124.423
182	2.802	57.023	5.717	2.960	6.263	2.060	802	434
183	10.405	93.899	992	6.309	7.654	21.619	5.669	3.189
184	6.523	42.026	36.046	9.972	5.214	2.056	4.481	3.355
150	8.227	7.200	5.811	5.530	1.409	1.567	4.751	2.332

NUTSIII	$\widehat{\mu}_i^{dir}$	$\widehat{\mu}_i^{sinQ}$	$\widehat{\mu}_i^{sinR}$	$\widehat{\mu}_i^{com}$	$\widehat{\mu}_i^{FH}$	$\widehat{\mu}_i^{NS}$	$\widehat{\mu}_{it}^{RY}$	$\widehat{\mu}_{it}^{LP}$
				***t*=6**				
111	10.180	21.055	1.349	3.508	517	1.689	1.358	1.229
112	8.871	1.826	21.473	10.137	3.938	5.095	1.367	1.965
113	5.077	14.395	3.875	3.949	434	432	1.073	1.143
114	2.833	20.135	1.984	2.312	383	486	415	435
115	2.387	23.866	983	1.933	502	514	267	289
116	3.188	38.190	496	2.143	1.191	500	550	566
117	1.529	36.515	488	1.261	559	261	64	18
118	5.747	805	35.041	5.178	2.855	2.135	1.115	1.006
161	6.072	1.537	3.309	4.367	632	1.334	1.225	1.565
162	8.060	1.450	769	3.425	1.388	2.847	1.652	1.896
163	4.955	21.869	16.678	7.177	4.629	1.423	2.114	1.835
164	6.171	17.073	6.055	5.680	940	6.350	4.120	1.026
165	17.362	1.464	397	9.647	1.156	3.252	4.332	4.016
166	1.419	109.587	78.264	1.442	3.849	623	174	60
167	1.682	59.514	39.675	1.761	5.021	688	152	31
168	2.854	9.339	24.132	4.698	7.688	3.696	5.679	1.563
169	17.051	4.986	12.636	14.890	19.818	4.196	17.546	6.618
16A	10.568	79.991	55.944	20.896	15.177	9.686	6.361	11.058
16B	5.125	135.670	8.605	5.814	5.746	5.622	1.823	1.448
171	5.722	3.867	4.319	4.409	3.486	2.110	1.869	1.979
172	7.633	9.476	2.557	4.307	2.862	1.532	2.594	2.113
16C	3.958	11.458	18.608	5.814	1.534	951	533	1.152
185	5.987	29.049	2.409	4.746	1.208	1.743	1.696	1.417
181	29.761	9.091	1.917	18.627	18.591	57.021	9.999	7.041
182	10.856	3.998	1.763	8.627	3.847	1.768	4.388	1.540
183	9.588	5.629	16.597	9.021	3.776	2.877	2.026	2.212
184	6.465	12.053	30.099	6.619	3.887	2.433	1.547	1.644
150	18.171	16.848	13.895	12.982	8.689	15.901	4.546	4.781
NUTSIII				***t*=7**				
111	14.153	40.389	743	6.463	3.517	3.475	2.256	2.361
112	6.759	54.035	3.470	5.106	13.953	10.941	2.567	1.493
113	3.488	17.199	2.888	3.221	871	500	765	842
114	8.685	13.963	7.740	7.680	1.115	1.097	3.553	3.048
115	4.568	46.017	1.739	3.851	5.982	1.890	2.538	1.368
116	3.656	46.989	1.836	2.756	3.400	1.304	739	746
117	1.598	28.204	330	1.158	413	81	21	0
118	1.478	3.658	23.343	1.558	84	56	16	7
161	8.277	1.465	810	2.880	913	1.788	1.644	2.022
162	4.285	1.230	1.711	2.431	853	1.169	1.339	1.203
163	3.679	6.725	5.399	3.744	1.436	673	797	905
164	5.037	3.653	1.031	3.682	840	1.279	1.236	624
165	6.646	11.910	15.344	8.370	6.878	6.940	2.609	1.955
166	1.379	122.991	107.969	1.397	9.595	1.735	159	83
167	1.417	20.304	10.533	1.642	628	14	41	8
168	2.069	3.286	7.306	2.208	1.276	901	324	335
169	12.425	5.416	8.864	10.914	13.181	3.514	7.330	2.836
16A	6.038	11.515	8.059	5.055	717	10.122	621	681
16B	6.401	84.015	3.132	4.685	3.390	2.924	1.303	1.655
171	4.756	2.508	3.884	3.905	1.823	2.012	1.154	1.257
172	7.009	4.143	3.555	5.367	5.371	1.656	2.820	2.172
16C	5.083	47.079	7.329	5.759	509	2.274	1.515	1.506
185	5.554	19.308	6.048	4.849	2.961	1.441	773	719
181	56.639	59.152	106.629	60.548	53.527	54.637	34.978	44.229
182	16.319	31.462	1.232	10.276	3.454	2.292	9.095	3.120
183	8.785	34.151	2.452	5.638	1.359	4.377	2.916	2.272
184	10.861	44.969	104.510	12.248	35.339	19.964	13.601	14.890
150	11.313	9.850	1.952	4.750	1.709	4.059	5.673	2.210

A14.3 Variância

NUTSIII	$\hat{\boldsymbol{\mu}}_i^{dir}$	$\hat{\boldsymbol{\mu}}_i^{sinQ}$	$\hat{\boldsymbol{\mu}}_i^{sinR}$	$\hat{\boldsymbol{\mu}}_i^{com}$	$\hat{\boldsymbol{\mu}}_i^{FH}$	$\hat{\boldsymbol{\mu}}_i^{NS}$	$\hat{\boldsymbol{\mu}}_{it}^{RY}$	$\hat{\boldsymbol{\mu}}_{it}^{LP}$
	***t*=1**							
111	13.531	2.529	242	5.214	634	2.011	2.322	2.278
112	4.934	2.125	337	4.458	664	1.438	981	1.100
113	2.418	2.372	269	2.358	155	214	189	190
114	4.269	4.182	478	4.460	1.049	702	1.190	1.213
115	4.807	2.323	280	4.601	1.806	1.482	1.527	1.276
116	6.005	2.654	228	3.589	436	714	997	1.038
117	11.279	2.063	359	7.173	398	456	1.754	2.441
118	6.076	1.909	425	5.965	1.617	2.389	2.717	2.811
161	5.438	1.178	225	3.713	812	1.483	1.094	1.194
162	10.359	1.768	638	4.689	686	1.371	2.614	2.745
163	5.603	1.149	221	3.549	325	533	1.294	1.556
164	8.597	828	330	6.351	2.456	785	1.314	800
165	17.050	942	256	12.940	1.724	3.210	4.217	4.530
166	1.585	902	277	1.608	635	119	11	2
167	1.485	863	303	1.585	195	3	6	1
168	2.703	636	574	2.577	1.249	3.931	740	749
169	11.776	940	257	10.284	4.403	3.854	7.277	3.572
16A	17.145	925	264	20.704	2.271	5.281	6.232	8.692
16B	7.503	884	537	6.440	478	1.547	1.858	2.117
171	7.202	3.509	2.070	4.081	1.817	1.536	1.894	1.925
172	4.136	2.624	932	2.776	776	1.021	1.023	964
16C	3.423	594	219	3.914	387	726	698	786
185	7.665	2.244	322	6.715	1.072	2.032	2.468	2.566
181	43.379	4.671	467	20.574	2.123	26.856	3.346	2.854
182	14.625	3.369	222	9.105	1.526	2.853	3.558	4.529
183	13.962	4.135	324	12.445	1.408	2.366	1.780	1.826
184	4.737	2.691	265	5.034	1.726	1.992	1.742	1.854
150	6.791	5.350	825	4.213	869	1.552	1.953	1.670

NUTSIII	$\widehat{\mu}_i^{dir}$	$\widehat{\mu}_i^{sinQ}$	$\widehat{\mu}_i^{sinR}$	$\widehat{\mu}_i^{com}$	$\widehat{\mu}_i^{FH}$	$\widehat{\mu}_i^{NS}$	$\widehat{\mu}_{it}^{RY}$	$\widehat{\mu}_{it}^{LP}$
				***t*=2**				
111	19.181	674	322	9.555	670	1.856	1.452	1.406
112	4.349	538	557	5.788	1.859	2.442	2.126	1.111
113	2.751	581	455	2.405	274	341	649	521
114	2.489	1.039	580	1.610	336	396	723	457
115	5.521	548	529	4.331	1.295	1.641	1.631	1.811
116	4.200	631	370	2.424	397	672	680	891
117	1.923	538	557	1.573	172	276	326	245
118	6.010	490	702	6.501	1.713	1.734	2.521	2.647
161	11.772	962	284	6.736	1.032	1.815	1.521	2.562
162	5.239	1.425	795	4.284	503	1.137	928	1.275
163	2.288	992	287	2.485	229	313	257	275
164	4.167	710	490	2.745	1.164	871	1.383	804
165	10.677	799	363	6.401	897	2.273	1.334	1.736
166	1.383	616	700	1.417	226	232	441	2
167	1.480	703	501	1.582	257	0	301	1
168	24.344	529	985	21.788	3.945	6.298	6.472	9.141
169	11.997	854	317	4.518	2.309	3.249	3.080	1.591
16A	10.334	752	422	13.879	1.443	4.645	1.737	3.109
16B	6.051	636	495	3.002	546	1.468	815	1.303
171	9.114	7.328	3.170	4.388	2.154	1.654	1.439	2.257
172	8.861	5.340	1.231	3.356	838	1.137	1.513	1.896
16C	4.654	463	291	3.397	374	368	460	670
185	10.194	2.286	366	9.305	1.602	2.298	3.170	3.374
181	36.060	5.293	755	22.064	2.914	27.619	2.630	3.248
182	33.175	3.317	302	24.779	1.508	2.823	3.902	3.993
183	10.920	4.050	321	12.942	1.783	2.758	4.348	2.941
184	7.107	3.062	350	4.165	739	2.356	1.699	1.682
150	8.454	6.610	1.122	3.427	827	1.344	1.081	1.586
NUTSIII				***t*=3**				
111	20.751	867	312	7.742	655	1.874	2.739	2.797
112	6.807	711	346	3.622	855	1.710	1.782	1.337
113	4.728	749	318	3.441	227	333	489	618
114	2.569	1.297	1.052	2.002	319	278	342	350
115	5.274	655	415	4.499	1.406	1.766	1.968	2.059
116	2.358	792	302	1.668	304	574	264	290
117	1.851	659	409	1.261	97	74	175	144
118	2.468	629	459	2.721	1.185	968	450	398
161	7.147	1.072	358	3.935	815	1.598	1.825	1.980
162	7.800	1.581	1.305	7.388	494	1.129	2.002	1.969
163	3.151	1.039	335	3.107	349	513	735	767
164	4.876	788	369	4.332	2.729	1.020	1.355	1.105
165	4.377	868	313	3.282	620	2.409	739	797
166	31.574	667	557	30.067	8.034	8.876	22.549	24.117
167	1.471	730	443	1.633	63	29	16	3
168	1.498	767	392	1.608	1.280	949	46	18
169	23.729	924	300	14.685	2.415	3.776	5.660	6.040
16A	11.594	914	301	10.570	1.335	4.743	2.918	3.518
16B	8.340	729	923	3.981	598	1.527	1.810	2.079
171	5.919	3.075	4.486	4.321	2.215	1.610	2.008	1.857
172	4.604	2.311	2.080	4.021	971	1.207	1.280	1.225
16C	4.498	526	330	4.006	664	581	817	841
185	12.769	24.460	579	12.015	1.769	2.449	4.067	4.293
181	165.000	55.420	1.123	155.606	7.918	33.674	32.031	38.434
182	10.467	39.235	358	11.213	1.174	2.821	3.667	3.714
183	12.705	42.653	453	9.723	1.313	2.621	1.871	1.570
184	6.613	33.655	301	8.156	1.823	1.861	1.796	1.822
150	22.088	20.036	2.401	9.040	977	1.426	6.088	3.702

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=4**							
111	18.721	1.549	306	5.133	639	1.857	2.162	2.498
112	4.388	1.282	255	2.382	757	1.507	881	925
113	4.871	1.343	250	2.792	406	428	814	931
114	3.124	2.279	1.150	2.207	473	373	537	550
115	4.589	1.175	288	3.323	1.567	1.293	1.838	1.530
116	4.189	1.428	261	2.790	613	589	1.258	1.146
117	1.460	1.179	286	1.536	152	323	4	1
118	1.957	1.060	366	2.301	379	531	370	362
161	8.245	1.219	343	5.121	610	1.498	2.084	2.158
162	5.135	1.874	1.511	3.917	452	997	1.477	1.371
163	4.226	1.196	324	4.143	474	663	1.190	1.188
164	4.517	914	274	4.254	3.035	945	1.414	598
165	11.189	1.011	250	4.307	734	2.616	2.781	2.323
166	1.518	876	297	1.547	409	191	48	11
167	1.484	783	389	1.588	38	52	11	1
168	7.313	845	321	6.917	2.115	2.605	4.072	3.857
169	13.525	1.081	261	7.497	2.874	3.540	6.316	3.029
16A	12.478	1.054	254	14.103	1.418	4.564	4.086	4.595
16B	9.034	900	1.083	5.120	604	1.521	2.056	2.051
171	7.180	4.229	4.505	5.012	2.617	1.782	2.565	2.696
172	6.906	3.112	2.013	4.933	1.115	1.291	2.720	1.969
16C	6.721	620	302	5.078	1.228	1.142	2.162	2.360
185	11.080	54.746	620	8.691	1.032	2.038	2.329	2.130
181	203.742	111.614	1.013	179.182	10.624	36.808	44.576	56.261
182	29.728	90.289	474	21.606	1.001	2.131	4.029	4.672
183	7.703	93.802	544	10.240	2.270	3.130	3.191	2.782
184	5.138	79.792	317	4.761	735	1.860	1.060	1.365
150	6.469	4.736	2.328	5.530	1.108	1.470	2.340	1.910
NUTSIII	***t*=5**							
111	7.297	1.548	250	3.523	409	1.480	1.302	1.419
112	17.895	1.264	221	13.558	496	1.426	1.784	1.828
113	4.314	1.313	206	3.684	430	445	829	857
114	3.150	2.326	1.373	2.353	437	403	588	591
115	2.295	1.198	255	1.738	634	502	504	430
116	3.502	1.440	208	1.760	486	487	597	657
117	1.575	1.147	294	1.565	44	63	9	0
118	1.652	1.082	360	1.781	426	722	217	219
161	5.983	1.621	406	2.619	437	1.338	948	1.034
162	5.608	2.346	1.818	7.290	801	1.304	1.805	1.772
163	5.169	1.547	330	3.019	378	625	993	1.203
164	13.201	1.175	222	8.314	982	929	3.784	2.531
165	11.787	1.360	213	7.894	996	2.702	1.879	2.162
166	1.731	1.035	328	1.721	154	219	149	159
167	1.614	1.163	227	1.757	548	21	22	1
168	5.797	1.093	272	7.004	2.108	2.183	3.738	2.925
169	21.536	1.399	228	14.651	2.989	4.034	4.806	4.105
16A	4.884	1.291	200	4.635	1.552	3.680	692	685
16B	8.934	1.203	1.113	4.370	801	1.664	1.624	2.044
171	4.933	3.557	3.959	5.466	3.368	1.844	2.034	1.829
172	10.682	2.986	2.378	7.815	1.375	1.577	3.571	2.938
16C	8.077	943	389	6.318	641	794	1.837	2.271
185	7.096	27.415	421	6.369	964	1.800	1.528	1.508
181	123.734	64.507	1.375	106.432	11.316	35.319	33.720	40.379
182	2.802	41.318	223	2.692	1.506	1.304	471	397
183	10.364	48.803	427	6.273	2.027	4.878	4.394	3.184
184	6.454	41.940	233	8.041	1.013	1.919	2.743	1.876
150	8.170	7.120	2.916	4.868	1.080	1.462	3.035	2.158

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
				***t*=6**				
111	10.149	1.117	208	2.687	463	1.518	1.351	1.096
112	8.827	853	277	9.060	506	1.449	1.198	1.191
113	5.069	901	242	3.607	310	431	719	1.121
114	2.807	1.623	690	2.141	330	379	415	435
115	2.372	790	341	1.874	424	455	264	287
116	3.164	1.007	203	2.142	617	486	543	561
117	1.527	781	351	1.251	324	155	52	18
118	5.096	735	415	4.927	685	675	882	959
161	5.980	945	206	3.495	427	1.333	1.048	1.382
162	7.988	1.429	769	3.425	608	1.230	1.403	1.889
163	4.950	1.137	332	6.447	636	714	1.450	1.561
164	6.013	715	291	5.476	594	1.363	2.060	1.004
165	16.827	862	202	8.892	1.142	2.907	2.477	3.770
166	1.419	728	277	1.439	1.006	199	35	1
167	1.671	804	222	1.724	1.308	139	24	1
168	2.854	627	415	4.350	2.428	1.927	3.165	1.280
169	16.946	774	240	12.528	2.916	3.944	5.770	4.491
16A	10.442	785	232	18.960	2.233	4.956	4.108	6.572
16B	5.110	621	511	4.820	1.193	1.989	1.230	1.342
171	5.697	3.826	3.668	4.115	2.355	1.954	1.625	1.979
172	7.602	2.647	1.342	4.158	1.367	1.529	1.857	1.916
16C	3.659	525	280	4.612	533	540	531	959
185	5.985	1.279	267	4.695	832	1.506	1.408	1.376
181	29.757	4.048	1.106	18.522	6.878	30.090	7.330	7.038
182	10.752	2.142	207	8.026	1.031	1.767	1.147	1.362
183	9.513	2.561	230	7.550	862	2.875	1.084	1.212
184	6.395	2.458	212	6.079	1.026	2.268	789	725
150	18.158	16.840	1.362	8.683	1.210	1.634	2.053	2.254
NUTSIII				***t*=7**				
111	14.038	5.507	242	6.119	933	2.118	1.938	2.361
112	6.701	4.601	193	4.686	2.211	2.637	1.852	1.412
113	3.480	4.334	196	2.943	385	443	764	829
114	8.680	7.762	657	7.083	771	761	2.488	2.781
115	4.553	4.134	204	3.626	1.600	1.174	1.869	1.338
116	3.468	4.783	196	2.753	980	702	733	741
117	1.596	3.681	244	1.158	152	38	2	0
118	1.463	3.456	275	1.524	57	55	10	5
161	7.926	909	303	2.529	540	1.583	1.637	1.946
162	4.279	1.180	666	2.358	610	1.150	1.228	1.183
163	3.621	847	252	3.563	552	617	793	861
164	4.923	648	198	3.682	558	708	921	618
165	6.385	726	197	6.894	2.030	3.235	1.955	1.643
166	1.378	688	193	1.391	2.133	415	8	3
167	1.416	535	263	1.597	226	2	3	0
168	2.032	593	219	2.208	794	877	324	333
169	11.507	664	195	10.503	1.986	3.500	2.380	2.372
16A	5.808	747	202	3.988	698	5.363	620	459
16B	6.360	568	745	4.266	482	1.434	1.298	1.575
171	4.739	2.409	3.395	3.839	1.648	1.761	1.109	1.239
172	7.008	1.646	1.387	5.109	1.469	1.593	1.980	1.965
16C	5.082	485	459	5.098	508	968	1.230	1.437
185	5.121	10.302	412	4.841	665	1.416	765	703
181	46.074	27.778	887	46.117	10.958	29.531	18.530	21.721
182	16.309	16.670	213	10.255	1.465	2.283	4.582	3.057
183	8.763	22.899	497	5.452	872	2.440	2.210	2.150
184	10.785	16.332	207	11.923	2.628	3.419	6.860	6.711
150	11.211	9.764	1.576	4.609	1.524	1.958	3.878	2.163

A14.4 Erro absoluto

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=1**							
111	94	108	61	76	44	35	40	41
112	56	80	89	60	22	31	25	27
113	38	100	69	39	12	13	11	11
114	54	75	108	64	62	43	31	30
115	57	242	73	58	67	44	40	31
116	61	112	58	55	37	53	25	26
117	100	68	103	89	78	80	58	51
118	68	55	119	75	40	38	48	48
161	58	55	82	60	51	36	26	29
162	82	34	21	46	50	55	44	42
163	63	56	26	46	34	37	29	32
164	72	38	84	66	113	49	53	22
165	104	53	94	101	94	65	64	59
166	31	325	281	31	53	16	12	9
167	31	186	139	33	23	2	6	4
168	42	126	63	42	43	91	23	23
169	88	116	158	102	188	78	111	59
16A	117	152	109	134	31	60	57	82
16B	73	192	66	69	27	44	34	37
171	69	48	40	49	57	33	43	35
172	51	70	36	41	24	25	27	24
16C	45	38	147	51	31	42	22	24
185	72	111	75	72	32	39	43	43
181	181	90	133	139	100	141	65	95
182	108	52	37	79	98	53	74	58
183	101	147	202	103	144	95	72	76
184	54	131	206	62	95	37	39	43
150	66	59	72	66	57	55	47	37

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	t=2							
111	104	34	117	101	98	83	58	56
112	52	37	189	64	82	76	44	28
113	42	104	47	41	14	15	24	18
114	40	121	23	31	14	20	24	17
115	61	126	29	52	28	44	43	33
116	52	117	33	41	19	29	20	24
117	35	103	50	34	21	28	15	13
118	60	50	205	66	90	68	68	53
161	87	84	99	85	73	57	58	47
162	57	55	76	60	18	27	30	30
163	39	106	94	42	16	14	15	13
164	52	61	25	41	31	33	57	22
165	77	32	15	47	27	39	30	33
166	30	268	217	31	27	12	26	7
167	30	189	148	31	26	2	19	5
168	134	75	27	116	49	92	76	68
169	88	71	45	64	42	91	71	36
16A	83	225	189	104	62	63	40	71
16B	63	251	25	41	24	31	23	29
171	78	70	48	54	76	36	50	38
172	77	70	28	43	35	52	31	35
16C	51	126	77	52	24	18	20	20
185	82	119	86	85	35	41	58	49
181	159	108	125	137	104	143	86	96
182	151	47	47	115	106	54	103	52
183	84	126	170	107	115	66	104	70
184	63	47	80	63	22	62	31	32
150	72	62	32	42	23	32	28	32
NUTSIII	*t*=3							
111	118	31	103	106	97	84	57	57
112	66	122	17	42	42	49	37	28
113	53	82	48	47	12	15	16	17
114	40	86	37	36	18	13	15	15
115	65	118	20	55	29	35	36	34
116	39	106	23	32	17	25	13	13
117	35	138	17	27	11	7	9	9
118	38	26	153	41	40	29	14	13
161	68	48	39	53	32	32	36	37
162	70	72	116	85	36	40	43	41
163	45	63	66	48	15	19	22	23
164	57	41	58	56	76	35	34	28
165	51	59	47	47	24	52	21	22
166	200	84	115	198	235	256	128	125
167	31	144	119	33	9	8	5	3
168	32	49	69	33	27	18	5	3
169	126	27	24	82	94	53	60	56
16A	96	165	158	90	27	46	37	38
16B	73	285	25	42	35	32	35	36
171	62	46	55	53	61	32	39	34
172	56	97	81	58	41	33	36	30
16C	53	148	74	54	29	22	22	23
185	90	134	159	102	83	57	57	59
181	410	498	655	406	596	492	426	436
182	86	259	96	96	28	77	50	51
183	81	177	118	85	69	41	40	39
184	63	164	197	76	96	46	31	35
150	121	115	61	78	84	104	62	54

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=4**							
111	111	100	55	77	41	41	53	46
112	53	149	18	34	44	45	24	25
113	56	159	13	37	43	34	23	24
114	45	87	50	40	27	18	19	18
115	57	163	15	45	38	29	37	32
116	52	201	42	45	44	31	29	27
117	31	98	71	33	20	28	1	2
118	35	65	109	39	16	23	17	16
161	74	66	90	77	61	45	44	41
162	56	45	68	54	20	31	33	30
163	53	109	82	59	26	25	31	29
164	49	34	86	53	71	32	25	16
165	86	26	39	58	41	42	44	41
166	31	303	241	32	47	20	8	6
167	30	142	69	33	5	13	3	2
168	73	24	59	76	34	49	59	51
169	92	26	35	64	88	54	90	42
16A	97	206	164	102	43	46	52	58
16B	76	243	36	52	20	31	37	36
171	67	52	55	57	64	34	43	42
172	68	96	61	61	30	29	53	39
16C	69	160	24	56	57	49	41	39
185	83	217	112	87	52	37	50	43
181	458	428	696	439	616	512	446	452
182	141	383	93	135	35	69	49	69
183	69	288	204	88	137	92	53	51
184	53	262	93	63	21	47	25	30
150	64	54	96	69	50	38	46	38
NUTSIII	***t*=5**							
111	68	172	15	37	25	39	29	29
112	112	37	171	114	72	74	59	69
113	52	143	50	49	28	22	23	22
114	45	104	50	42	23	18	19	19
115	40	170	30	36	23	19	18	17
116	47	198	15	29	31	20	20	20
117	31	144	59	33	5	8	2	2
118	32	111	95	34	23	31	12	13
161	62	41	26	38	46	54	24	25
162	60	87	145	75	62	51	41	39
163	57	44	24	41	22	35	28	29
164	92	63	14	60	24	44	77	37
165	87	31	23	63	23	42	34	33
166	35	241	171	34	9	11	8	9
167	32	188	133	33	36	9	5	4
168	65	46	106	73	44	34	58	44
169	116	46	70	100	103	50	84	56
16A	55	259	220	53	91	71	32	32
16B	74	247	28	46	25	32	32	36
171	56	52	102	63	123	56	44	36
172	86	148	62	75	36	31	58	48
16C	75	133	50	64	36	40	34	38
185	68	158	76	69	46	36	39	32
181	350	303	443	336	361	287	308	309
182	42	187	74	45	69	33	22	16
183	85	222	26	58	79	133	64	45
184	62	196	189	78	65	34	50	44
150	72	67	62	59	29	32	56	39

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=6**							
111	75	141	34	44	17	31	30	28
112	75	36	146	84	59	62	28	36
113	55	116	60	54	17	16	25	26
114	43	136	39	39	16	18	16	17
115	39	152	27	34	17	17	12	12
116	46	193	19	35	29	17	18	19
117	31	189	18	28	16	11	5	3
118	56	23	186	52	49	41	26	23
161	59	32	56	54	20	29	27	30
162	72	31	22	41	32	46	32	34
163	57	144	128	71	64	32	38	36
164	59	128	76	60	26	71	50	24
165	108	31	16	59	26	46	53	48
166	30	330	279	30	53	21	12	8
167	32	242	199	33	61	23	11	5
168	44	93	154	56	76	48	63	32
169	104	65	111	110	134	51	122	69
16A	86	281	236	127	117	93	75	97
16B	57	367	90	65	68	62	34	30
171	59	49	54	54	48	37	34	36
172	71	85	40	52	45	31	43	38
16C	48	105	135	61	34	24	18	25
185	62	167	46	57	29	33	32	30
181	143	77	35	101	111	189	91	67
182	81	50	39	68	54	30	58	30
183	79	64	128	80	55	43	38	40
184	63	100	173	64	56	38	33	36
150	108	104	112	99	88	120	55	57
NUTSIII	***t*=7**							
111	93	187	23	57	56	51	36	37
112	62	222	57	57	110	93	40	30
113	47	113	52	48	26	18	22	23
114	76	85	84	76	28	27	51	46
115	53	205	39	50	71	36	39	29
116	48	205	41	44	52	31	22	21
117	32	157	15	26	16	7	4	0
118	30	48	152	31	5	4	3	2
161	73	31	24	40	25	35	32	36
162	52	28	34	39	23	27	29	27
163	50	77	72	53	34	21	23	25
164	51	55	29	42	21	29	23	17
165	59	106	123	72	72	66	38	33
166	29	350	328	29	86	36	12	9
167	30	141	101	33	20	4	6	3
168	36	52	84	38	28	21	14	14
169	89	69	93	92	108	48	79	45
16A	55	104	89	58	20	73	20	20
16B	64	289	49	56	54	47	29	33
171	55	40	50	50	34	36	27	28
172	67	54	50	58	65	33	44	38
16C	58	216	83	67	17	41	31	32
185	56	113	75	55	49	30	22	21
181	191	198	325	197	217	194	154	176
182	104	131	32	74	52	35	84	44
183	75	140	45	60	29	54	43	39
184	88	177	323	96	183	134	106	113
150	83	77	36	51	32	53	60	37

Apêndice 15 – Medidas da qualidade dos estimadores do grupo A (*L*=1.000)

A15.1 Enviesamento

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=1**							
111	-18,5	107,4	-61,2	-58,7	-42,9	-16,4	-24,4	-27,8
112	-5,6	79,0	-88,3	-25,4	-10,5	1,6	0,8	-10,6
113	4,7	99,4	-68,7	-5,2	-8,4	-9,3	-0,3	-1,2
114	-18,9	63,9	-109,2	-37,4	-62,0	-42,0	-19,8	-17,1
115	-1,3	241,3	73,4	11,1	61,8	35,1	24,2	8,4
116	1,9	111,0	-57,9	-25,9	-35,4	-51,9	-4,1	-8,9
117	-26,6	64,7	-102,3	-50,7	-76,4	-78,8	-53,7	-41,2
118	-23,7	48,6	-117,9	-39,4	-31,4	-1,5	-29,8	-28,6
161	-8,6	-53,6	-82,3	-33,6	-49,4	-27,7	-4,8	-15,6
162	8,4	3,0	2,3	2,8	-49,0	-50,2	18,9	3,6
163	-11,1	55,1	24,8	-6,8	-34,4	-37,0	-4,9	0,2
164	-2,4	-35,1	-83,8	-27,9	-114,8	-49,8	-53,4	-13,4
165	3,0	-51,8	-93,6	-36,4	-92,1	-53,5	-40,0	-28,6
166	0,8	325,0	280,8	2,6	53,4	15,8	12,2	8,7
167	1,3	186,2	139,6	5,3	23,1	-1,6	6,0	4,4
168	0,6	125,7	64,1	8,8	38,6	89,8	5,8	4,6
169	-40,1	-115,6	-157,6	-67,5	-187,8	-71,2	-106,5	-46,1
16A	-53,3	152,0	109,2	-20,1	-17,8	-53,6	-26,0	9,9
16B	-20,3	-192,1	64,5	6,3	20,0	31,0	2,0	4,7
171	2,9	-7,5	-22,4	-11,6	-54,7	9,5	-34,0	-2,6
172	-7,6	62,9	29,3	4,5	10,3	-6,0	9,7	1,5
16C	2,9	-36,2	146,6	16,6	28,8	40,3	6,0	11,9
185	-8,3	-111,1	73,6	8,6	18,3	-21,2	16,9	13,2
181	-42,9	-83,6	-134,3	-80,6	-99,0	11,4	-63,2	-95,9
182	-22,9	28,5	-37,5	-31,8	-96,4	-38,2	-60,6	-27,4
183	-7,8	-146,6	-203,3	-38,7	-146,1	-92,6	-67,2	-70,6
184	-16,3	-130,6	-205,9	-36,0	-96,1	-21,0	-30,8	-36,0
150	10,4	11,6	70,5	36,5	53,7	50,6	35,8	19,0

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
				***t*=2**				
111	-5,5	31,6	-117,7	-71,4	-99,1	-82,9	-56,8	-55,6
112	-12,3	-35,7	-189,3	-30,2	-82,7	-75,6	-39,1	-19,6
113	2,9	105,0	-47,1	-8,2	3,3	2,4	14,6	-2,7
114	3,9	122,4	-17,1	-1,7	-0,8	14,3	14,8	-0,1
115	5,0	127,0	-26,2	-1,0	-4,9	-30,2	28,8	-2,2
116	-6,1	117,8	-32,8	-16,6	-13,0	-24,1	2,3	-7,3
117	-0,4	103,9	-49,7	-12,0	-20,4	-27,2	-5,9	-3,7
118	-15,0	-49,6	-204,8	-31,4	-88,8	-65,9	-61,4	-42,6
161	-29,7	-83,2	-99,3	-49,8	-72,2	-53,1	-53,8	-31,8
162	1,5	52,7	74,2	26,4	0,1	-2,0	20,8	13,5
163	-0,4	107,0	93,5	10,2	11,5	1,8	9,2	3,4
164	-6,8	61,3	20,7	-6,6	-25,9	22,6	-55,8	-1,9
165	-12,0	27,5	-4,0	2,5	-14,0	12,3	-11,7	-1,2
166	-1,1	268,4	217,6	1,3	27,9	4,8	25,9	7,4
167	0,2	189,2	147,9	4,0	26,6	2,5	18,0	5,1
168	9,8	75,2	14,2	-0,1	16,4	68,1	-46,2	-4,9
169	-0,1	71,1	44,9	42,4	-15,2	87,9	-54,0	29,1
16A	-22,5	225,0	188,8	20,2	53,2	16,3	20,6	50,3
16B	-9,5	-250,5	19,8	7,7	-19,2	0,1	-9,1	2,4
171	5,6	6,1	-21,1	-16,5	-77,6	-22,9	-49,7	-8,0
172	7,9	46,5	-4,4	-7,6	-33,2	-54,5	3,4	-1,9
16C	7,0	-125,8	76,8	24,4	-20,2	-10,7	-10,6	9,1
185	-14,5	-119,5	85,2	10,1	16,8	-22,6	39,2	17,5
181	-13,2	-102,0	-127,0	-61,8	-101,9	-10,1	-86,2	-93,4
182	60,3	11,6	-47,7	50,3	-105,1	-40,2	-90,5	-12,7
183	-2,4	-125,2	-170,8	-53,5	-115,2	-57,5	-98,1	-58,8
184	-20,4	-16,3	-80,7	-42,0	1,2	63,1	-18,4	-23,3
150	12,0	11,6	18,9	9,5	-6,8	-13,8	11,6	5,5
NUTSIII				***t*=3**				
111	-13,8	23,7	-102,9	-79,6	-97,6	-84,0	-49,3	-48,6
112	2,9	122,1	-8,7	-11,1	37,1	41,8	18,3	-0,7
113	2,5	81,9	-47,8	-9,5	-6,6	-10,3	2,8	-3,0
114	1,9	86,2	-30,5	-5,4	-13,1	2,0	-2,4	-1,0
115	33,9	117,8	-14,5	25,3	7,4	-10,4	20,6	16,7
116	-4,9	106,4	-22,2	-6,8	-12,6	-21,0	-1,0	-2,3
117	-0,5	138,4	6,1	-0,8	9,5	5,6	3,0	-0,2
118	-4,1	-19,4	-152,6	-11,3	-39,4	-27,2	-8,4	-6,2
161	-8,7	-44,8	-39,3	-13,6	-23,4	-1,8	-8,9	-6,2
162	8,8	71,5	114,9	45,7	32,7	34,7	28,2	23,0
163	-6,2	62,7	65,5	12,7	-7,9	-9,9	2,2	3,3
164	-18,1	-38,2	-57,7	-24,8	-75,3	-30,7	-20,9	-7,2
165	7,3	58,5	46,4	19,9	17,9	46,8	5,7	5,6
166	-175,5	-83,1	-114,8	-176,7	-234,6	-257,0	-112,2	-105,9
167	0,0	144,1	118,7	5,1	9,3	-7,7	5,5	3,2
168	-0,6	-47,6	-69,2	-6,0	-27,8	-17,3	-5,2	-2,1
169	-67,0	-15,8	-22,9	-52,7	-89,0	23,4	-36,0	-20,3
16A	-71,2	165,8	157,9	-60,0	3,5	-36,0	-6,9	-17,3
16B	-15,2	-284,6	5,5	-5,6	-34,8	-15,8	-14,3	-3,8
171	1,2	-21,9	-8,6	-3,1	-58,9	-0,3	-25,4	-1,2
172	-1,9	90,2	79,0	21,4	33,6	15,5	19,6	8,4
16C	-4,4	-147,1	73,5	14,0	-22,6	-12,2	1,9	3,9
185	27,8	45,4	158,8	54,7	80,5	45,8	37,3	40,4
181	-245,6	-492,8	-656,0	-278,3	-597,4	-493,5	-426,2	-433,9
182	-9,5	260,4	96,0	21,5	15,7	70,1	16,0	18,7
183	11,5	45,6	-118,5	-45,3	-69,3	-7,4	-26,6	-28,0
184	-6,8	-31,9	-196,7	-39,9	-97,8	-34,4	-11,3	-29,1
150	9,3	9,7	-53,6	-38,5	-86,2	-105,8	6,4	-31,0

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	t=4							
111	3,0	101,9	-55,1	-60,9	-40,7	-34,0	-43,1	-30,3
112	3,3	151,0	-15,3	-2,9	42,3	40,4	7,2	0,0
113	1,3	160,8	-3,3	3,6	42,2	32,4	6,2	0,3
114	-3,8	86,6	-48,5	-15,8	-24,4	-12,6	-7,6	-4,9
115	-1,7	164,1	-6,2	-10,0	17,9	-2,9	7,1	-1,5
116	8,6	203,1	42,0	13,9	42,5	26,7	12,7	7,4
117	0,2	100,0	-70,2	-6,8	-20,3	-28,6	-0,6	-1,6
118	1,8	66,1	-108,7	-4,5	2,0	14,4	-3,4	-2,8
161	-17,7	-64,0	-90,8	-47,7	-60,2	-39,7	-29,6	-23,8
162	-1,0	35,5	64,7	18,8	-14,7	-21,4	14,5	7,8
163	-8,3	110,4	81,3	20,9	21,1	14,8	11,7	7,1
164	-10,0	-27,5	-85,5	-23,4	-71,9	-30,7	-19,7	-7,7
165	-4,5	8,5	-39,1	-36,3	-38,7	-8,9	-5,1	-19,6
166	-2,9	303,7	241,3	-0,8	47,4	20,0	8,1	5,8
167	-1,0	143,2	70,1	6,1	0,3	-13,2	3,4	1,8
168	0,2	7,4	-58,4	-12,7	-11,0	17,4	-19,4	-9,2
169	-8,2	4,8	-35,6	-4,9	-77,6	28,5	-67,1	-4,3
16A	-55,3	206,6	163,4	-28,1	28,6	-17,3	3,8	8,5
16B	-15,0	-242,3	28,5	6,9	-8,0	1,6	-11,3	2,4
171	-2,3	0,0	3,5	2,9	-60,6	-8,0	-23,0	1,6
172	4,9	89,2	55,2	14,6	12,3	-7,2	29,1	9,8
16C	-1,4	-158,8	21,5	1,2	-51,8	-41,3	-16,1	1,5
185	-0,4	114,1	110,3	40,6	49,0	7,3	37,7	25,3
181	-309,5	-402,2	-698,4	-320,6	-618,5	-514,1	-448,1	-452,4
182	26,3	384,2	91,6	61,7	30,3	65,7	22,6	52,9
183	-17,8	88,2	-204,9	-50,4	-137,8	-88,9	-39,0	-39,2
184	-8,7	196,7	-93,8	-33,4	-3,3	45,5	2,5	-12,4
150	8,5	7,5	92,9	34,1	44,8	26,1	28,7	16,1
NUTSIII	*t*=5							
111	-12,6	173,3	-10,5	-7,7	20,6	23,6	7,4	-1,7
112	-12,5	26,2	-170,7	-63,4	-71,8	-73,4	-55,3	-66,7
113	13,9	144,7	-49,9	-6,6	25,7	16,7	8,7	0,4
114	-3,6	105,4	-47,8	-15,5	-19,6	-11,5	-5,7	-5,5
115	-9,4	170,9	-29,3	-14,7	12,5	-6,9	-0,8	-4,5
116	-6,1	199,4	10,8	3,8	29,4	11,6	4,2	0,5
117	-0,7	144,7	-58,0	-7,9	-1,0	-7,5	1,1	-1,6
118	-1,6	111,8	-94,2	-6,4	19,4	27,8	-1,7	-2,6
161	-9,8	28,7	23,3	7,6	45,0	52,4	1,8	3,4
162	-5,9	84,9	143,3	28,4	60,8	48,4	26,4	23,3
163	-8,8	34,9	22,4	-2,7	-20,6	-33,9	-14,2	-1,5
164	-30,0	61,2	9,2	-25,0	-13,9	37,8	-71,5	-6,7
165	2,4	9,7	-21,9	-10,8	-6,2	12,2	-19,0	-6,2
166	-15,6	240,6	172,1	-14,1	3,5	-9,6	-5,9	-6,7
167	2,5	187,3	133,8	6,9	36,4	8,7	5,5	3,9
168	18,0	-43,6	-105,2	3,0	-25,0	-0,1	-16,6	-0,4
169	3,1	-42,1	-69,7	-17,8	-95,5	6,8	-62,1	-24,8
16A	-22,3	259,1	220,0	-5,9	90,4	44,7	24,4	24,7
16B	-14,4	-247,0	0,4	-2,3	-14,0	-4,8	4,2	0,6
171	0,6	-34,3	-101,2	-33,0	-127,5	-56,1	-37,1	-18,1
172	-3,9	143,8	55,7	10,1	20,5	-4,4	29,3	13,4
16C	-16,4	-133,0	49,4	-1,4	-32,7	-36,1	-11,9	1,8
185	-16,7	17,0	75,6	17,6	41,8	11,5	26,6	9,3
181	-217,5	-272,2	-443,8	-229,2	-363,2	-276,4	-299,1	-295,9
182	-0,4	127,0	-74,1	-15,4	-68,9	-24,6	-18,0	-5,9
183	1,8	214,2	23,3	1,5	74,7	131,0	34,9	1,3
184	-7,5	10,9	-189,3	-43,2	-65,8	-10,8	-42,6	-39,0
150	7,4	8,3	51,8	23,1	15,4	-11,9	40,5	13,1

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=6**							
111	5,1	142,8	-34,1	-28,7	-7,8	-13,1	2,5	-11,8
112	10,0	32,6	-145,3	-29,9	-58,3	-60,3	-12,2	-27,2
113	4,2	117,6	-60,1	-18,5	10,9	-1,6	18,0	-5,7
114	-2,0	138,0	-37,2	-11,1	6,0	9,4	-1,2	-1,6
115	-4,5	153,3	-24,9	-8,2	8,8	-8,3	1,0	-1,9
116	-6,9	194,4	17,1	-3,2	23,9	3,4	2,0	-3,3
117	-0,2	190,4	12,2	1,6	15,5	10,5	3,6	0,4
118	26,0	-7,1	-185,5	16,7	-45,5	-37,1	-13,8	-5,6
161	-11,9	-24,8	-56,0	-29,0	-14,1	0,2	-12,2	-12,1
162	6,9	-5,2	-1,0	-2,4	-29,6	-41,6	-16,8	1,4
163	-1,2	143,4	127,2	28,0	62,5	25,9	25,9	16,4
164	-12,8	127,5	76,2	14,0	18,3	70,7	-44,9	4,7
165	-14,2	24,1	-14,0	-22,5	3,4	18,3	-43,6	-15,4
166	-1,1	329,5	279,6	0,8	53,9	20,8	12,1	7,7
167	1,2	241,8	198,7	3,8	61,2	23,6	11,4	5,5
168	0,2	-93,8	-153,4	-17,4	-73,2	-42,6	-51,2	-17,1
169	-13,6	-65,4	-111,2	-49,1	-129,4	-14,3	-107,6	-44,6
16A	-10,8	281,0	236,2	44,9	113,4	67,9	46,5	66,5
16B	-7,5	-367,9	-90,9	-34,6	-70,0	-62,6	-26,4	-12,2
171	0,5	-11,2	-28,5	-18,8	-38,4	-16,5	-20,2	-3,5
172	-2,5	78,6	33,1	7,3	35,0	-2,8	22,0	8,5
16C	15,8	-104,9	134,8	32,9	31,2	20,1	1,3	13,7
185	-0,1	-166,6	45,8	7,2	18,9	-15,4	16,5	5,8
181	5,6	71,1	26,9	-1,5	104,5	161,2	47,5	-1,4
182	4,6	43,1	-39,4	-28,2	-52,1	3,1	-56,1	-12,3
183	6,2	-55,3	-128,3	-40,0	-54,6	2,7	-31,0	-31,2
184	9,8	-97,9	-173,2	-21,1	-54,2	-11,8	-27,5	-29,7
150	2,5	2,9	-113,7	-66,5	-89,7	-121,6	-53,2	-53,0
NUTSIII	***t*=7**							
111	-5,2	190,8	23,0	-15,1	49,8	36,4	-18,3	-1,0
112	9,1	226,0	58,1	22,4	108,9	91,6	26,3	8,6
113	-1,9	117,0	-50,9	-14,6	21,7	7,1	-2,0	-4,3
114	-0,3	83,6	-84,2	-22,8	-20,7	-20,0	-34,7	-18,5
115	7,2	208,2	40,2	18,4	67,1	26,8	25,9	5,6
116	-13,4	209,2	41,3	1,2	49,7	24,8	2,5	-2,4
117	3,0	159,9	-8,1	1,3	16,4	6,7	4,3	-0,2
118	-2,8	17,4	-150,6	-5,0	-5,0	-0,5	-2,6	-1,6
161	-10,4	-22,5	-22,1	-13,6	19,3	15,4	-0,6	-6,7
162	5,0	8,4	32,3	9,8	14,5	3,5	9,8	3,7
163	-4,8	77,8	72,3	16,2	29,1	6,8	1,4	6,1
164	-11,7	55,8	29,9	-0,7	-18,2	23,2	-18,3	-2,6
165	-17,1	-104,7	-122,3	-39,2	-70,3	-61,4	-25,9	-17,9
166	1,2	350,7	329,2	2,8	87,2	36,6	12,3	9,0
167	-1,8	141,5	102,6	3,8	20,1	-3,5	6,1	2,8
168	6,2	-51,0	-83,1	-0,2	-22,5	-5,2	0,5	1,6
169	27,3	-68,0	-92,2	-22,3	-106,3	4,8	-69,2	-20,7
16A	-14,0	104,8	89,4	33,7	-5,2	-70,5	-1,3	14,8
16B	-7,3	-288,0	48,8	18,9	51,8	36,7	-4,2	6,6
171	-0,1	-13,2	20,6	7,0	8,8	12,2	-9,4	2,0
172	-7,1	47,3	46,0	10,9	58,4	3,0	22,5	7,9
16C	-1,5	-215,0	83,1	24,6	-1,9	-36,7	-16,3	8,7
185	-17,6	-94,3	75,3	6,0	47,3	-5,0	2,4	3,6
181	-97,9	-176,2	-325,4	-116,7	-209,8	-161,1	-132,1	-153,9
182	-9,3	122,3	-31,2	-8,1	-44,4	0,2	-64,5	-7,4
183	2,7	106,9	-44,1	-14,2	20,5	44,1	26,5	-10,6
184	7,8	-168,5	-322,2	-17,9	-181,1	-127,5	-80,2	-88,1
150	6,6	7,7	-20,1	-14,9	10,5	-47,9	40,9	-8,2

A15.2 EQM

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=1**							
111	14.841	14.145	3.975	9.114	2.479	2.359	2.988	3.114
112	4.905	8.424	8.126	5.182	753	1.457	974	1.201
113	2.330	12.322	4.980	2.304	211	288	171	173
114	4.746	8.387	12.409	5.994	4.961	2.489	1.589	1.500
115	4.656	60.634	5.659	4.597	5.530	2.662	2.026	1.287
116	5.972	15.062	3.578	4.355	1.706	3.401	1.036	1.135
117	11.416	6.315	10.820	9.209	6.233	6.660	4.521	4.027
118	6.663	4.328	14.317	7.459	2.553	2.314	3.547	3.568
161	5.324	4.052	6.996	4.801	3.312	2.294	1.125	1.454
162	10.864	1.778	649	5.129	3.148	3.883	2.986	2.730
163	6.189	4.183	833	3.966	1.529	1.920	1.375	1.596
164	8.830	2.061	7.348	7.100	15.588	3.284	4.250	1.077
165	16.275	3.626	9.013	13.698	9.959	5.749	5.564	5.024
166	1.592	106.540	79.124	1.622	3.472	364	160	77
167	1.433	35.532	19.779	1.561	720	5	42	20
168	2.670	16.426	4.668	2.683	2.717	11.915	787	794
169	12.906	14.302	25.073	14.609	39.767	8.798	18.705	5.590
16A	20.207	24.020	12.172	21.048	2.600	8.362	7.157	9.146
16B	7.592	37.779	4.706	6.280	911	2.566	1.823	2.119
171	7.063	3.784	2.598	3.948	4.857	1.655	3.078	1.996
172	4.127	6.741	1.801	2.725	870	1.130	1.109	955
16C	3.466	1.903	21.691	4.239	1.201	2.319	728	913
185	7.676	14.659	5.742	6.797	1.360	2.455	2.760	2.751
181	47.574	11.818	18.504	28.907	11.731	26.953	7.612	12.245
182	15.472	4.291	1.624	10.181	10.577	4.067	6.785	4.810
183	13.668	25.769	41.659	13.552	22.731	10.852	6.338	6.798
184	5.194	19.840	42.648	6.617	10.967	2.464	2.741	3.214
150	6.919	5.456	5.800	5.550	3.732	4.185	3.338	2.121

NUTSIII	$\widehat{\mu}_i^{dir}$	$\widehat{\mu}_i^{sinQ}$	$\widehat{\mu}_i^{sinR}$	$\widehat{\mu}_i^{com}$	$\widehat{\mu}_i^{FH}$	$\widehat{\mu}_i^{NS}$	$\widehat{\mu}_{it}^{RY}$	$\widehat{\mu}_{it}^{LP}$
				t=2				
111	19.222	1.680	14.166	13.739	10.502	8.823	4.701	4.558
112	4.838	1.820	36.366	7.056	8.697	8.195	3.661	1.473
113	2.741	11.616	2.656	2.439	272	334	854	525
114	2.673	16.038	920	1.808	335	576	900	430
115	5.840	16.695	1.188	4.574	1.230	2.503	2.409	1.765
116	4.363	14.516	1.435	2.763	552	1.214	662	917
117	2.049	11.336	2.997	1.813	587	1.016	361	256
118	6.617	2.953	42.602	7.868	9.615	6.082	6.296	4.426
161	12.016	7.956	10.163	9.174	6.339	4.719	4.595	3.715
162	5.526	4.312	6.363	4.922	522	1.095	1.414	1.452
163	2.223	12.517	9.056	2.564	363	326	348	296
164	4.337	4.528	896	2.819	1.828	1.429	4.449	871
165	9.975	1.617	373	5.397	1.165	2.410	1.632	1.878
166	1.447	72.697	48.012	1.483	1.005	250	1.104	57
167	1.483	36.558	22.362	1.601	951	7	613	27
168	25.671	6.232	1.126	23.064	4.007	10.539	8.675	9.049
169	12.000	5.974	2.334	6.568	2.672	10.875	6.280	2.508
16A	10.554	51.443	36.051	14.096	4.264	5.083	2.240	5.811
16B	5.985	63.424	934	2.933	930	1.504	894	1.323
171	9.938	7.903	3.694	5.053	8.236	2.230	3.905	2.453
172	9.377	7.896	1.319	3.852	1.975	4.233	1.775	2.228
16C	4.510	16.332	6.194	3.925	797	486	572	774
185	10.587	16.602	7.667	9.656	1.751	2.748	4.765	3.743
181	36.440	15.761	16.937	26.237	13.188	27.673	10.393	12.423
182	37.604	3.486	2.587	28.155	12.416	4.242	11.648	3.891
183	12.144	19.770	29.521	16.566	15.090	6.025	13.961	6.526
184	8.539	3.362	6.864	6.569	752	6.394	2.027	2.192
150	8.715	6.947	1.546	3.493	847	1.572	1.282	1.685
NUTSIII				*t*=3				
111	22.438	1.475	10.895	15.342	10.193	8.996	5.213	5.244
112	6.367	15.656	419	3.419	2.234	3.507	2.186	1.354
113	4.346	7.498	2.600	3.152	267	436	529	630
114	2.417	8.793	2.057	1.976	497	277	346	346
115	6.312	14.573	628	5.120	1.410	1.869	2.371	2.320
116	2.501	12.147	790	1.815	440	971	267	295
117	1.857	19.840	447	1.284	184	103	181	143
118	2.520	1.041	23.749	2.927	2.692	1.687	543	459
161	7.318	3.076	1.908	4.234	1.429	1.676	2.034	2.162
162	7.991	6.692	14.612	9.463	1.568	2.270	2.818	2.505
163	3.187	4.968	4.629	3.262	402	600	728	754
164	5.088	2.240	3.703	4.801	8.341	1.981	1.814	1.186
165	4.458	4.289	2.461	3.643	942	4.521	743	804
166	61.932	7.566	13.751	60.876	62.921	74.799	34.514	34.685
167	1.559	21.478	14.544	1.753	145	87	46	14
168	1.466	3.032	5.187	1.595	2.105	1.298	77	23
169	28.722	1.170	823	17.733	10.533	4.268	6.810	6.152
16A	17.016	28.409	25.226	14.447	1.327	6.203	3.045	3.904
16B	8.899	81.701	1.013	4.189	1.840	1.809	2.121	2.205
171	6.329	3.808	4.941	4.709	5.700	1.638	2.741	1.875
172	4.718	10.635	8.486	4.916	2.090	1.510	1.745	1.378
16C	4.461	22.164	5.727	4.278	1.174	719	869	900
185	13.537	27.974	25.839	14.618	8.172	4.504	5.320	5.876
181	225.773	301.551	431.541	234.901	365.554	277.894	215.305	228.273
182	10.692	109.368	9.573	11.313	1.347	7.670	3.559	3.672
183	13.396	47.265	14.517	12.060	6.115	2.603	2.440	2.204
184	6.940	36.666	39.004	10.158	11.386	3.056	1.936	2.735
150	20.342	18.682	5.467	9.601	8.363	12.631	5.835	4.511

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=4**							
111	18.869	11.881	3.358	8.764	2.264	3.037	4.012	3.397
112	4.610	24.041	483	2.609	2.514	3.158	922	901
113	4.367	27.159	259	2.433	2.181	1.465	832	909
114	3.001	9.706	3.574	2.498	1.058	517	578	552
115	4.625	28.072	314	3.409	1.802	1.266	1.839	1.493
116	4.265	42.627	2.027	2.931	2.414	1.269	1.452	1.226
117	1.480	11.133	5.197	1.615	563	1.131	4	4
118	2.057	5.391	12.166	2.432	373	738	376	367
161	8.717	5.356	8.604	7.675	4.285	3.140	3.018	2.777
162	5.677	3.207	5.780	4.632	699	1.433	1.750	1.456
163	4.492	13.420	6.954	4.613	899	862	1.294	1.203
164	4.342	1.703	7.578	4.523	8.229	1.938	1.731	662
165	11.487	1.122	1.777	5.482	2.236	2.613	2.870	2.677
166	1.468	93.114	58.526	1.486	2.645	585	110	45
167	1.547	21.328	5.289	1.692	36	224	23	4
168	7.892	932	3.719	7.574	2.287	2.953	4.594	4.069
169	14.495	1.146	1.530	7.566	9.174	4.343	11.350	3.249
16A	15.892	43.772	26.955	15.443	2.213	5.029	4.175	4.770
16B	9.036	59.634	1.961	4.697	702	1.559	2.241	2.079
171	7.630	4.838	4.728	5.238	6.320	1.889	3.161	2.684
172	6.588	11.519	5.176	5.166	1.240	1.390	3.581	2.094
16C	6.879	25.864	774	5.264	3.903	2.817	2.410	2.305
185	10.995	68.927	12.833	10.028	3.408	2.079	3.646	2.706
181	300.460	275.746	488.816	285.382	394.089	302.008	247.472	262.675
182	32.216	239.777	8.899	26.508	1.997	6.423	4.817	7.632
183	9.081	103.570	42.578	13.806	21.363	11.096	4.718	4.346
184	5.725	120.182	9.138	5.959	761	3.917	1.080	1.569
150	6.454	4.814	11.089	6.601	3.035	2.156	3.158	2.196
NUTSIII	***t*=5**							
111	7.366	31.501	363	3.433	815	2.093	1.311	1.409
112	17.704	1.882	29.346	17.167	5.644	6.862	4.845	6.309
113	4.155	22.178	2.686	3.491	1.051	694	861	804
114	3.041	13.302	3.718	2.625	813	520	592	595
115	2.503	30.324	1.105	2.030	797	570	502	445
116	3.437	41.117	322	1.740	1.312	564	564	613
117	1.631	22.012	3.654	1.681	46	118	8	3
118	1.786	13.518	9.236	1.954	808	1.505	212	221
161	5.909	2.475	963	2.699	2.482	4.130	1.002	1.087
162	5.853	9.606	22.437	8.273	4.458	3.559	2.463	2.272
163	5.266	2.799	839	2.923	805	1.774	1.266	1.288
164	14.148	4.945	301	8.714	1.209	2.420	8.942	2.597
165	11.809	1.483	693	7.884	1.042	2.793	2.198	2.110
166	1.937	58.937	29.954	1.881	169	313	184	208
167	1.562	36.251	18.130	1.735	1.844	95	54	16
168	6.288	3.019	11.329	7.146	2.812	2.266	4.065	3.014
169	22.713	3.199	5.085	15.412	12.394	4.122	9.256	5.143
16A	5.578	68.441	48.594	4.846	9.698	5.826	1.317	1.332
16B	9.563	62.212	1.160	4.815	1.030	1.707	1.652	2.104
171	5.522	5.166	14.374	7.222	19.674	5.034	3.579	2.200
172	10.348	24.021	5.578	7.808	1.745	1.641	4.585	3.331
16C	8.303	18.648	2.836	6.354	1.707	2.077	1.901	2.178
185	7.278	28.261	6.146	6.408	2.680	1.935	2.170	1.562
181	168.692	139.888	198.390	158.499	143.669	112.195	123.571	128.267
182	2.837	58.287	5.721	2.965	6.223	1.834	789	422
183	11.016	95.669	984	7.038	7.528	21.936	5.489	3.024
184	6.693	42.914	36.069	10.307	5.390	2.080	4.571	3.443
150	9.242	7.759	5.725	5.986	1.239	1.590	4.771	2.428

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	t=6							
111	10.347	21.458	1.381	3.586	526	1.766	1.340	1.200
112	9.134	1.872	21.389	10.415	3.897	5.120	1.346	1.925
113	5.036	14.689	3.857	3.993	417	420	1.056	1.175
114	2.631	20.586	2.103	2.185	355	448	404	419
115	2.620	24.242	960	2.104	501	551	291	314
116	3.223	38.734	503	2.227	1.194	487	570	584
117	1.566	36.990	497	1.310	572	269	66	19
118	5.800	746	34.823	5.235	2.841	2.142	1.392	1.264
161	6.689	1.586	3.350	4.655	648	1.365	1.230	1.501
162	8.345	1.493	802	3.511	1.502	2.936	1.766	1.914
163	5.124	21.739	16.523	7.241	4.529	1.392	2.116	1.807
164	6.458	16.980	6.106	5.847	922	6.364	4.087	1.008
165	16.597	1.464	407	8.673	1.209	3.206	4.312	3.857
166	1.520	109.313	78.428	1.540	3.879	623	193	60
167	1.549	59.309	39.718	1.609	4.991	687	165	31
168	2.837	9.433	23.951	4.607	7.855	3.813	6.021	1.613
169	18.544	5.066	12.602	15.587	19.906	4.194	17.987	6.780
16A	10.399	79.752	56.016	21.079	15.179	9.775	6.417	11.231
16B	5.322	135.988	8.809	6.055	6.145	5.938	1.944	1.483
171	6.375	4.646	4.541	4.720	3.944	2.332	2.110	2.075
172	7.701	9.302	2.478	4.556	2.536	1.588	2.441	2.054
16C	3.955	11.551	18.475	5.712	1.484	926	504	1.076
185	5.968	29.102	2.378	4.697	1.147	1.740	1.707	1.402
181	30.017	9.303	1.867	18.887	17.783	56.449	9.707	6.988
182	10.652	4.110	1.771	8.390	3.765	1.724	4.305	1.430
183	9.618	5.753	16.714	9.177	3.811	2.789	2.088	2.174
184	7.106	12.167	30.216	7.213	4.085	2.561	1.735	1.764
150	19.700	17.841	14.337	13.696	9.184	16.404	4.843	5.026
NUTSIII	*t*=7							
111	15.333	42.036	766	6.912	3.493	3.579	2.369	2.489
112	6.988	55.793	3.570	5.328	13.929	10.956	2.486	1.434
113	3.609	18.120	2.789	3.232	866	508	799	878
114	8.903	14.909	7.749	7.819	1.159	1.115	3.705	3.124
115	4.348	47.555	1.823	3.735	6.001	1.843	2.535	1.343
116	3.702	48.657	1.901	2.808	3.452	1.304	776	776
117	1.596	29.331	315	1.162	419	85	21	0
118	1.453	3.832	22.967	1.524	75	55	19	8
161	8.461	1.437	788	3.341	931	1.841	1.709	2.028
162	4.658	1.281	1.715	2.579	816	1.122	1.286	1.147
163	3.824	6.917	5.478	3.889	1.383	667	787	886
164	5.033	3.776	1.090	3.556	917	1.297	1.275	635
165	6.199	11.715	15.143	7.892	7.070	6.995	2.667	1.986
166	1.476	123.702	108.555	1.497	9.664	1.735	158	83
167	1.426	20.564	10.795	1.607	620	14	40	8
168	2.150	3.207	7.125	2.290	1.345	959	335	346
169	12.398	5.300	8.689	10.915	13.370	3.541	7.317	2.895
16A	6.217	11.750	8.194	5.048	731	10.511	644	691
16B	6.596	83.511	3.130	4.857	3.210	2.830	1.474	1.795
171	4.562	2.670	3.924	3.842	1.776	1.996	1.251	1.314
172	6.801	3.940	3.532	5.135	4.801	1.638	2.503	2.001
16C	5.141	46.734	7.363	5.798	515	2.321	1.469	1.476
185	5.296	19.395	6.087	4.678	2.868	1.430	724	677
181	55.124	59.346	106.787	60.005	55.090	55.972	36.557	45.917
182	17.374	31.947	1.185	10.722	3.478	2.224	9.305	3.524
183	9.664	34.757	2.440	6.143	1.232	4.345	2.887	2.234
184	11.348	45.036	104.036	12.895	35.709	19.966	13.508	14.727
150	11.686	10.081	2.015	4.994	1.528	4.215	5.399	2.161

A15.3 Variância

NUTSIII	$\hat{\boldsymbol{\mu}}_i^{dir}$	$\hat{\boldsymbol{\mu}}_i^{sinQ}$	$\hat{\boldsymbol{\mu}}_i^{sinR}$	$\hat{\boldsymbol{\mu}}_i^{com}$	$\hat{\boldsymbol{\mu}}_i^{FH}$	$\hat{\boldsymbol{\mu}}_i^{NS}$	$\hat{\boldsymbol{\mu}}_{it}^{RY}$	$\hat{\boldsymbol{\mu}}_{it}^{LP}$
	***t*=1**							
111	14.498	2.602	234	5.665	643	2.089	2.392	2.340
112	4.874	2.187	328	4.538	642	1.455	973	1.089
113	2.308	2.440	261	2.276	140	201	171	172
114	4.390	4.303	480	4.594	1.112	726	1.197	1.207
115	4.654	2.391	272	4.474	1.711	1.432	1.442	1.216
116	5.968	2.731	221	3.683	449	706	1.019	1.056
117	10.707	2.123	350	6.642	401	457	1.642	2.326
118	6.103	1.964	416	5.904	1.567	2.311	2.657	2.751
161	5.250	1.179	220	3.671	872	1.528	1.103	1.209
162	10.793	1.769	644	5.121	748	1.360	2.627	2.717
163	6.065	1.150	216	3.919	348	555	1.351	1.596
164	8.825	829	321	6.324	2.413	803	1.401	897
165	16.266	943	248	12.370	1.474	2.882	3.962	4.203
166	1.592	903	269	1.615	625	115	11	2
167	1.432	863	294	1.533	185	3	5	1
168	2.669	637	565	2.605	1.227	3.846	754	773
169	11.301	940	249	10.047	4.487	3.728	7.357	3.462
16A	17.363	926	256	20.646	2.284	5.489	6.481	9.049
16B	7.181	885	541	6.239	510	1.606	1.819	2.097
171	7.055	3.728	2.097	3.813	1.866	1.564	1.922	1.989
172	4.068	2.788	942	2.704	763	1.093	1.015	953
16C	3.458	594	214	3.964	373	696	692	771
185	7.607	2.316	321	6.722	1.025	2.004	2.476	2.577
181	45.737	4.822	469	22.405	1.927	26.823	3.613	3.043
182	14.946	3.477	216	9.168	1.284	2.611	3.119	4.059
183	13.607	4.268	322	12.053	1.373	2.280	1.820	1.814
184	4.928	2.777	257	5.320	1.735	2.023	1.794	1.916
150	6.811	5.321	834	4.220	844	1.623	2.056	1.759

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=2**							
111	19.193	681	322	8.638	687	1.948	1.470	1.469
112	4.687	543	528	6.142	1.863	2.479	2.130	1.089
113	2.733	588	436	2.371	261	329	640	517
114	2.658	1.050	630	1.805	334	371	679	430
115	5.815	554	503	4.573	1.206	1.593	1.579	1.760
116	4.326	638	362	2.489	383	635	656	863
117	2.049	543	528	1.669	171	276	326	242
118	6.391	495	661	6.881	1.733	1.736	2.525	2.615
161	11.132	1.039	300	6.692	1.132	1.904	1.704	2.702
162	5.524	1.539	853	4.223	522	1.091	979	1.271
163	2.223	1.071	307	2.459	231	322	263	284
164	4.291	767	467	2.775	1.157	916	1.330	867
165	9.831	863	356	5.391	969	2.258	1.495	1.877
166	1.446	666	659	1.482	228	226	433	2
167	1.483	760	478	1.585	244	0	290	1
168	25.575	571	924	23.064	3.738	5.906	6.539	9.025
169	12.000	922	318	4.770	2.441	3.155	3.364	1.663
16A	10.046	812	407	13.687	1.436	4.816	1.816	3.277
16B	5.896	687	541	2.873	560	1.504	811	1.317
171	9.907	7.866	3.251	4.781	2.207	1.704	1.440	2.389
172	9.314	5.732	1.299	3.794	872	1.263	1.763	2.224
16C	4.461	500	300	3.332	391	371	460	692
185	10.378	2.310	402	9.554	1.469	2.237	3.225	3.438
181	36.266	5.350	811	22.423	2.801	27.570	2.967	3.698
182	33.963	3.352	307	25.621	1.376	2.624	3.450	3.730
183	12.138	4.094	350	13.704	1.826	2.717	4.328	3.073
184	8.124	3.095	345	4.806	751	2.414	1.689	1.649
150	8.572	6.813	1.188	3.403	800	1.382	1.148	1.654
NUTSIII	***t*=3**							
111	22.247	914	310	9.012	662	1.947	2.785	2.882
112	6.358	749	344	3.295	858	1.759	1.851	1.353
113	4.340	789	314	3.063	223	331	521	621
114	2.413	1.366	1.124	1.947	325	273	340	345
115	5.164	690	417	4.479	1.355	1.762	1.946	2.041
116	2.478	834	298	1.768	281	529	266	290
117	1.856	695	410	1.283	94	72	173	143
118	2.503	663	464	2.801	1.142	947	472	421
161	7.242	1.066	363	4.049	882	1.673	1.954	2.123
162	7.913	1.573	1.401	7.377	498	1.067	2.022	1.975
163	3.148	1.034	337	3.100	340	503	723	742
164	4.761	784	368	4.186	2.674	1.036	1.377	1.134
165	4.404	863	309	3.247	623	2.329	710	773
166	31.116	663	569	29.645	7.897	8.769	21.919	23.463
167	1.559	726	447	1.727	59	27	16	3
168	1.466	763	392	1.559	1.334	999	50	19
169	24.229	919	296	14.954	2.619	3.719	5.515	5.739
16A	11.948	910	297	10.848	1.314	4.904	2.998	3.606
16B	8.669	726	983	4.158	629	1.560	1.917	2.191
171	6.327	3.329	4.868	4.700	2.234	1.638	2.097	1.873
172	4.715	2.501	2.246	4.459	959	1.269	1.361	1.307
16C	4.442	523	331	4.083	661	571	865	884
185	12.764	25.912	606	11.622	1.689	2.407	3.927	4.243
181	165.470	58.708	1.202	157.452	8.725	34.335	33.679	40.026
182	10.601	41.563	362	10.852	1.100	2.750	3.303	3.323
183	13.264	45.184	467	10.004	1.310	2.549	1.735	1.418
184	6.894	35.652	297	8.566	1.826	1.872	1.808	1.888
150	20.255	18.587	2.597	8.119	934	1.429	5.794	3.548

NUTSIII	$\widehat{\mu}_i^{dir}$	$\widehat{\mu}_i^{sinQ}$	$\widehat{\mu}_i^{sinR}$	$\widehat{\mu}_i^{com}$	$\widehat{\mu}_i^{FH}$	$\widehat{\mu}_i^{NS}$	$\widehat{\mu}_{it}^{RY}$	$\widehat{\mu}_{it}^{LP}$
	***t*=4**							
111	18.860	1.500	318	5.058	605	1.879	2.153	2.478
112	4.599	1.241	248	2.601	723	1.524	869	901
113	4.365	1.301	248	2.420	398	416	793	909
114	2.987	2.207	1.221	2.248	464	358	521	529
115	4.622	1.138	276	3.310	1.481	1.257	1.788	1.491
116	4.192	1.383	265	2.738	609	554	1.291	1.171
117	1.480	1.142	274	1.568	150	316	3	1
118	2.054	1.027	347	2.411	369	530	364	359
161	8.404	1.265	359	5.398	666	1.564	2.142	2.212
162	5.676	1.946	1.600	4.277	483	976	1.539	1.395
163	4.423	1.242	338	4.175	453	643	1.158	1.152
164	4.242	949	263	3.975	3.060	992	1.343	604
165	11.467	1.050	248	4.164	736	2.533	2.844	2.294
166	1.459	909	283	1.486	401	184	45	11
167	1.546	813	369	1.655	36	49	11	1
168	7.892	877	306	7.414	2.165	2.649	4.217	3.985
169	14.427	1.122	265	7.543	3.147	3.530	6.847	3.230
16A	12.829	1.094	256	14.653	1.397	4.728	4.161	4.698
16B	8.811	935	1.151	4.648	638	1.557	2.114	2.073
171	7.624	4.838	4.715	5.230	2.650	1.825	2.632	2.681
172	6.565	3.560	2.125	4.954	1.088	1.339	2.735	1.998
16C	6.877	643	313	5.262	1.219	1.108	2.150	2.302
185	10.995	55.904	660	8.377	1.005	2.026	2.222	2.068
181	204.673	113.976	1.076	182.566	11.594	37.678	46.700	58.030
182	31.524	92.199	503	22.698	1.078	2.112	4.306	4.834
183	8.764	95.787	578	11.269	2.375	3.190	3.197	2.809
184	5.649	81.481	330	4.845	750	1.849	1.074	1.414
150	6.382	4.757	2.453	5.439	1.028	1.474	2.337	1.938
NUTSIII	***t*=5**							
111	7.208	1.463	252	3.374	389	1.534	1.257	1.406
112	17.547	1.194	215	13.152	487	1.472	1.791	1.859
113	3.961	1.240	201	3.447	392	414	784	804
114	3.027	2.199	1.431	2.386	428	387	559	565
115	2.415	1.132	249	1.815	642	522	501	425
116	3.399	1.361	206	1.726	445	429	546	613
117	1.630	1.084	288	1.619	45	62	7	0
118	1.784	1.023	355	1.913	431	732	209	214
161	5.812	1.654	418	2.641	459	1.388	999	1.075
162	5.818	2.393	1.896	7.468	756	1.215	1.769	1.728
163	5.188	1.578	338	2.915	382	626	1.065	1.286
164	13.250	1.198	216	8.090	1.017	988	3.836	2.552
165	11.803	1.388	212	7.768	1.004	2.643	1.836	2.072
166	1.694	1.056	323	1.683	157	220	149	163
167	1.555	1.186	222	1.687	522	20	23	1
168	5.965	1.115	267	7.137	2.186	2.266	3.791	3.014
169	22.703	1.427	229	15.094	3.277	4.076	5.402	4.529
16A	5.082	1.317	197	4.811	1.530	3.827	719	719
16B	9.355	1.227	1.160	4.810	833	1.684	1.634	2.103
171	5.522	3.991	4.126	6.135	3.422	1.882	2.204	1.873
172	10.333	3.350	2.480	7.705	1.323	1.621	3.727	3.151
16C	8.036	962	401	6.353	639	772	1.760	2.175
185	6.998	27.973	434	6.099	933	1.804	1.464	1.475
181	121.392	65.820	1.434	105.982	11.755	35.801	34.100	40.703
182	2.837	42.159	223	2.727	1.477	1.230	465	387
183	11.013	49.797	441	7.036	1.947	4.779	4.269	3.022
184	6.637	42.794	234	8.442	1.054	1.964	2.757	1.921
150	9.188	7.689	3.041	5.452	1.001	1.449	3.129	2.256

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
				t=6				
111	10.321	1.058	221	2.765	466	1.594	1.334	1.060
112	9.033	808	279	9.521	498	1.487	1.197	1.187
113	5.018	854	246	3.652	299	418	732	1.143
114	2.627	1.538	720	2.062	320	360	402	417
115	2.600	748	339	2.037	423	483	290	310
116	3.175	954	211	2.217	624	476	566	573
117	1.566	740	349	1.308	331	159	53	18
118	5.123	696	410	4.956	771	763	1.202	1.233
161	6.547	970	219	3.814	449	1.365	1.082	1.354
162	8.297	1.466	801	3.505	625	1.205	1.483	1.912
163	5.122	1.167	353	6.456	617	720	1.447	1.538
164	6.295	734	292	5.652	586	1.363	2.070	985
165	16.395	885	211	8.166	1.197	2.871	2.410	3.620
166	1.519	748	279	1.539	977	192	48	1
167	1.548	825	227	1.595	1.247	133	34	1
168	2.837	644	411	4.306	2.501	1.997	3.400	1.320
169	18.359	794	243	13.173	3.167	3.990	6.412	4.788
16A	10.281	806	237	19.060	2.325	5.166	4.252	6.802
16B	5.266	637	537	4.856	1.244	2.015	1.245	1.335
171	6.375	4.520	3.729	4.365	2.467	2.061	1.702	2.063
172	7.694	3.127	1.382	4.503	1.311	1.581	1.956	1.982
16C	3.705	539	298	4.628	512	521	503	888
185	5.968	1.344	285	4.646	792	1.503	1.435	1.368
181	29.986	4.252	1.143	18.885	6.870	30.467	7.456	6.986
182	10.630	2.250	215	7.596	1.052	1.715	1.159	1.279
183	9.579	2.690	245	7.580	829	2.782	1.130	1.203
184	7.010	2.582	225	6.768	1.144	2.423	976	879
150	19.693	17.833	1.403	9.275	1.143	1.607	2.016	2.219
NUTSIII				*t*=7				
111	15.306	5.624	239	6.684	1.012	2.256	2.034	2.488
112	6.906	4.699	192	4.826	2.067	2.568	1.794	1.361
113	3.605	4.426	197	3.017	394	458	795	860
114	8.903	7.927	663	7.300	732	715	2.503	2.782
115	4.296	4.222	206	3.396	1.502	1.126	1.863	1.312
116	3.521	4.884	195	2.807	979	687	770	770
117	1.587	3.759	250	1.160	151	40	2	0
118	1.445	3.530	285	1.499	49	54	12	6
161	8.352	933	300	3.155	557	1.603	1.709	1.983
162	4.633	1.210	672	2.484	606	1.109	1.190	1.133
163	3.801	869	249	3.626	536	621	785	849
164	4.896	665	199	3.556	586	760	939	628
165	5.906	745	195	6.356	2.133	3.223	1.998	1.666
166	1.475	706	193	1.489	2.063	398	8	3
167	1.422	549	271	1.592	215	2	3	0
168	2.111	608	223	2.290	837	932	335	344
169	11.655	681	195	10.417	2.068	3.518	2.529	2.464
16A	6.019	767	199	3.915	704	5.544	642	471
16B	6.543	583	753	4.501	523	1.481	1.456	1.752
171	4.562	2.496	3.499	3.792	1.697	1.846	1.162	1.310
172	6.751	1.705	1.416	5.015	1.391	1.629	1.996	1.940
16C	5.139	498	459	5.194	511	972	1.204	1.400
185	4.987	10.495	411	4.643	634	1.404	719	663
181	45.544	28.297	900	46.379	11.089	30.009	19.111	22.217
182	17.287	16.981	211	10.656	1.511	2.224	5.146	3.470
183	9.657	23.327	498	5.940	812	2.400	2.183	2.122
184	11.287	16.637	205	12.575	2.916	3.706	7.082	6.960
150	11.643	10.021	1.612	4.771	1.417	1.920	3.724	2.094

A15.4 Erro absoluto

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=1**							
111	97	107	61	79	44	36	40	41
112	56	80	88	61	22	31	25	28
113	37	99	69	39	12	13	10	10
114	55	75	109	65	63	44	31	30
115	55	241	73	57	67	43	39	30
116	61	111	58	57	37	53	25	27
117	96	67	102	85	76	79	56	49
118	68	54	118	74	39	38	47	48
161	57	55	82	59	51	37	26	29
162	84	34	21	48	51	55	44	41
163	65	57	25	48	35	38	29	32
164	72	38	84	68	115	50	54	23
165	101	53	94	99	95	64	64	59
166	32	325	281	32	53	16	12	9
167	31	186	140	32	23	2	6	4
168	42	126	64	43	43	92	23	23
169	84	116	158	100	188	77	111	58
16A	118	152	109	134	32	62	59	84
16B	69	192	65	67	25	43	34	37
171	68	50	41	49	60	33	45	36
172	51	69	35	41	23	27	27	25
16C	46	37	147	51	31	42	21	24
185	72	111	74	71	31	39	44	43
181	186	91	134	142	101	142	68	97
182	108	53	38	80	98	50	72	54
183	99	147	203	101	146	95	72	75
184	56	131	206	65	96	38	41	45
150	65	58	71	65	54	54	46	37

NUTSIII	$\widehat{\mu}_i^{dir}$	$\widehat{\mu}_i^{sinQ}$	$\widehat{\mu}_i^{sinR}$	$\widehat{\mu}_i^{com}$	$\widehat{\mu}_i^{FH}$	$\widehat{\mu}_i^{NS}$	$\widehat{\mu}_{it}^{RY}$	$\widehat{\mu}_{it}^{LP}$
	***t*=2**							
111	107	34	118	100	99	83	59	57
112	53	37	189	64	83	77	44	28
113	42	105	47	41	13	15	23	18
114	41	122	25	33	15	20	23	16
115	63	127	29	54	27	44	42	32
116	53	118	33	42	19	29	20	24
117	36	104	50	35	21	28	15	13
118	62	50	205	68	90	68	67	52
161	84	83	99	83	73	57	58	47
162	59	56	74	60	18	26	30	30
163	38	107	94	41	16	14	15	13
164	53	62	25	41	32	33	57	23
165	75	33	15	44	27	39	31	33
166	31	268	218	31	28	12	26	7
167	30	189	148	31	27	3	18	5
168	138	75	27	121	49	91	75	68
169	88	71	45	64	43	92	71	37
16A	83	225	189	104	61	64	40	71
16B	62	250	25	39	26	31	24	29
171	81	73	49	57	80	38	53	40
172	78	71	29	46	37	56	33	38
16C	49	126	77	51	24	18	19	21
185	84	120	85	85	34	41	58	50
181	160	108	127	138	105	143	89	97
182	153	47	48	118	106	51	98	51
183	89	127	171	111	116	65	102	69
184	67	47	81	65	22	64	31	33
150	73	64	31	43	23	32	28	33
NUTSIII	***t*=3**							
111	123	31	103	108	98	84	58	57
112	63	122	17	39	42	50	38	29
113	51	82	48	45	12	15	16	17
114	39	86	38	36	18	13	15	15
115	65	118	21	55	28	36	35	34
116	40	106	24	33	17	25	13	13
117	34	138	17	27	11	7	9	9
118	39	26	153	42	40	29	15	13
161	69	48	40	52	32	33	37	38
162	71	73	115	85	34	39	43	40
163	45	63	66	48	15	19	22	23
164	56	41	58	56	78	36	35	28
165	51	59	46	47	24	52	20	21
166	203	83	115	201	235	257	123	119
167	32	144	119	34	9	8	5	3
168	31	49	69	33	28	19	5	3
169	131	27	24	83	94	53	59	55
16A	99	166	158	92	27	48	38	39
16B	74	285	26	44	38	33	36	37
171	64	49	57	55	65	33	42	35
172	56	92	80	58	38	32	34	30
16C	53	147	73	54	29	22	22	23
185	90	137	159	102	83	57	57	60
181	410	496	656	407	598	494	432	441
182	86	262	96	96	28	79	48	49
183	82	181	119	86	70	40	40	37
184	64	168	197	78	98	46	31	37
150	116	112	62	77	87	106	61	54

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=4**							
111	111	102	55	77	41	41	52	46
112	54	151	18	35	44	45	24	24
113	52	161	13	33	43	34	23	24
114	44	88	51	40	27	18	19	18
115	56	164	14	45	37	29	37	31
116	52	203	42	45	44	31	30	28
117	31	100	70	33	20	29	1	2
118	37	67	109	40	16	24	16	16
161	75	65	91	77	61	45	43	41
162	58	46	66	55	21	31	33	30
163	53	110	81	58	26	24	30	28
164	48	34	86	52	73	33	25	16
165	87	27	39	56	41	41	44	41
166	31	304	241	31	47	20	8	6
167	32	143	70	34	4	13	3	2
168	76	25	58	78	35	50	59	52
169	95	27	36	63	88	54	91	44
16A	99	207	163	105	43	48	53	59
16B	75	242	35	50	21	31	38	36
171	70	55	56	58	68	35	45	42
172	65	93	59	59	28	29	49	37
16C	69	159	23	56	57	48	41	39
185	83	220	110	86	52	37	49	42
181	457	428	698	442	618	515	452	458
182	148	385	92	136	35	72	50	71
183	73	291	205	92	138	91	53	51
184	55	265	94	64	22	48	25	30
150	63	53	93	67	47	37	44	37
NUTSIII	***t*=5**							
111	67	173	15	36	24	39	28	30
112	111	36	171	112	72	74	59	69
113	50	145	50	48	28	22	23	22
114	44	106	52	41	24	18	19	19
115	41	171	30	37	23	19	18	17
116	46	199	15	29	32	19	19	20
117	32	145	58	34	5	8	2	2
118	33	112	94	35	24	32	12	12
161	60	41	25	37	46	55	25	26
162	61	87	143	76	62	51	40	38
163	57	44	24	40	23	36	29	30
164	93	62	14	60	25	44	77	38
165	87	32	23	63	24	41	35	33
166	35	241	172	35	9	11	9	9
167	31	187	134	33	36	9	5	4
168	66	46	105	73	45	35	58	45
169	120	47	70	101	104	50	86	58
16A	57	259	220	54	91	71	32	33
16B	75	247	28	47	27	32	32	36
171	59	56	104	67	128	59	47	37
172	84	144	61	73	34	32	55	47
16C	75	133	49	65	36	40	33	37
185	67	159	76	68	46	36	39	31
181	344	304	444	335	365	290	314	313
182	42	189	74	45	69	32	22	16
183	86	224	26	61	80	135	63	44
184	64	198	189	79	67	34	51	45
150	76	69	61	61	27	32	55	39

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
				t=6				
111	74	143	34	45	17	32	30	27
112	76	36	145	85	59	62	29	36
113	55	118	60	54	17	16	25	27
114	41	138	40	38	15	17	16	16
115	41	153	27	36	17	17	12	12
116	45	194	19	35	30	17	19	19
117	31	190	18	28	16	11	5	3
118	57	22	186	53	49	41	27	24
161	63	32	56	55	20	30	27	30
162	73	31	23	42	33	47	34	35
163	57	143	127	71	63	31	39	35
164	60	127	76	62	26	71	50	24
165	105	31	17	56	26	45	53	47
166	31	329	280	31	54	21	12	8
167	31	242	199	32	61	24	11	5
168	43	94	153	55	77	49	64	33
169	108	66	111	113	134	51	122	71
16A	85	281	236	128	117	94	75	98
16B	58	368	91	66	71	65	35	30
171	63	54	55	56	51	38	36	37
172	71	83	40	53	42	32	41	37
16C	48	105	135	60	33	24	17	24
185	63	167	46	57	28	33	33	30
181	142	77	34	100	108	188	90	67
182	81	50	40	68	54	30	57	30
183	78	65	128	81	55	42	38	39
184	66	100	173	68	58	39	35	37
150	113	107	114	102	90	122	57	58
NUTSIII				t=7				
111	96	191	24	59	55	52	37	39
112	63	226	58	58	110	93	39	29
113	48	117	51	48	26	18	23	24
114	78	90	84	77	28	27	52	47
115	51	208	40	50	71	35	40	29
116	48	209	41	44	53	31	22	22
117	32	160	14	26	16	7	4	0
118	30	49	151	31	5	4	3	2
161	74	31	24	42	25	36	33	36
162	54	29	35	40	23	26	28	27
163	50	78	72	54	33	21	22	24
164	52	56	30	42	22	29	24	17
165	58	105	122	71	73	67	38	33
166	30	351	329	31	87	37	12	9
167	30	141	103	32	20	4	6	3
168	37	51	83	39	28	21	14	14
169	89	68	92	92	108	47	78	45
16A	57	105	89	58	20	74	20	20
16B	65	288	49	57	52	46	31	34
171	54	41	51	50	34	36	28	29
172	66	53	50	58	61	32	41	36
16C	58	215	83	67	17	41	30	31
185	56	113	75	55	48	30	22	21
181	188	199	325	197	220	197	157	179
182	107	132	31	74	52	34	84	47
183	78	141	44	61	28	53	43	38
184	90	177	322	99	183	134	106	113
150	84	78	36	51	30	55	59	37

A15.5 Enviesamento relativo absoluto

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=1**							
111	2,5%	14,3%	8,1%	7,8%	5,7%	2,2%	3,2%	3,7%
112	0,8%	11,2%	12,5%	3,6%	1,5%	0,2%	0,1%	1,5%
113	0,6%	13,6%	9,4%	0,7%	1,2%	1,3%	0,0%	0,2%
114	1,8%	6,1%	10,5%	3,6%	6,0%	4,0%	1,9%	1,6%
115	0,2%	41,5%	12,6%	1,9%	10,6%	6,0%	4,2%	1,4%
116	0,2%	14,4%	7,5%	3,4%	4,6%	6,8%	0,5%	1,2%
117	3,7%	9,1%	14,4%	7,1%	10,7%	11,1%	7,5%	5,8%
118	3,4%	7,0%	16,9%	5,7%	4,5%	0,2%	4,3%	4,1%
161	1,0%	6,3%	9,6%	3,9%	5,8%	3,2%	0,6%	1,8%
162	0,9%	0,3%	0,2%	0,3%	5,0%	5,1%	1,9%	0,4%
163	1,5%	7,5%	3,4%	0,9%	4,7%	5,0%	0,7%	0,0%
164	0,3%	5,0%	11,8%	3,9%	16,2%	7,0%	7,5%	1,9%
165	0,4%	6,7%	12,2%	4,7%	12,0%	7,0%	5,2%	3,7%
166	0,2%	86,3%	74,6%	0,7%	14,2%	4,2%	3,2%	2,3%
167	0,3%	37,2%	27,9%	1,1%	4,6%	0,3%	1,2%	0,9%
168	0,1%	27,1%	13,8%	1,9%	8,3%	19,4%	1,2%	1,0%
169	4,8%	13,9%	18,9%	8,1%	22,6%	8,6%	12,8%	5,5%
16A	9,5%	27,2%	19,5%	3,6%	3,2%	9,6%	4,7%	1,8%
16B	2,3%	21,7%	7,3%	0,7%	2,3%	3,5%	0,2%	0,5%
171	0,2%	0,6%	1,8%	0,9%	4,3%	0,8%	2,7%	0,2%
172	0,7%	6,1%	2,9%	0,4%	1,0%	0,6%	0,9%	0,1%
16C	0,5%	6,0%	24,2%	2,7%	4,8%	6,7%	1,0%	2,0%
185	1,0%	14,1%	9,3%	1,1%	2,3%	2,7%	2,1%	1,7%
181	4,0%	7,9%	12,7%	7,6%	9,3%	1,1%	6,0%	9,0%
182	2,9%	3,6%	4,7%	4,0%	12,0%	4,8%	7,6%	3,4%
183	0,7%	13,7%	19,1%	3,6%	13,7%	8,7%	6,3%	6,6%
184	1,9%	15,0%	23,6%	4,1%	11,0%	2,4%	3,5%	4,1%
150	1,1%	1,2%	7,3%	3,8%	5,6%	5,3%	3,7%	2,0%

NUTSIII	$\widehat{\mu}_i^{dir}$	$\widehat{\mu}_i^{sinQ}$	$\widehat{\mu}_i^{sinR}$	$\widehat{\mu}_i^{com}$	$\widehat{\mu}_i^{FH}$	$\widehat{\mu}_i^{NS}$	$\widehat{\mu}_{it}^{RY}$	$\widehat{\mu}_{it}^{LP}$
				***t*=2**				
111	0,6%	3,7%	13,9%	8,4%	11,7%	9,8%	6,7%	6,6%
112	1,5%	4,4%	23,1%	3,7%	10,1%	9,2%	4,8%	2,4%
113	0,4%	14,8%	6,6%	1,2%	0,5%	0,3%	2,1%	0,4%
114	0,4%	12,6%	1,8%	0,2%	0,1%	1,5%	1,5%	0,0%
115	0,7%	19,1%	3,9%	0,1%	0,7%	4,5%	4,3%	0,3%
116	0,8%	16,1%	4,5%	2,3%	1,8%	3,3%	0,3%	1,0%
117	0,1%	15,3%	7,3%	1,8%	3,0%	4,0%	0,9%	0,5%
118	1,9%	6,2%	25,6%	3,9%	11,1%	8,3%	7,7%	5,3%
161	3,3%	9,4%	11,2%	5,6%	8,1%	6,0%	6,0%	3,6%
162	0,2%	5,7%	8,0%	2,8%	0,0%	0,2%	2,2%	1,5%
163	0,1%	15,0%	13,1%	1,4%	1,6%	0,3%	1,3%	0,5%
164	1,1%	9,7%	3,3%	1,1%	4,1%	3,6%	8,9%	0,3%
165	1,7%	3,9%	0,6%	0,4%	2,0%	1,7%	1,7%	0,2%
166	0,3%	71,3%	57,8%	0,4%	7,4%	1,3%	6,9%	2,0%
167	0,0%	37,8%	29,6%	0,8%	5,3%	0,5%	3,6%	1,0%
168	1,9%	14,4%	2,7%	0,0%	3,1%	13,0%	8,8%	0,9%
169	0,0%	10,3%	6,5%	6,2%	2,2%	12,8%	7,8%	4,2%
16A	4,6%	46,1%	38,7%	4,1%	10,9%	3,3%	4,2%	10,3%
16B	1,0%	27,6%	2,2%	0,9%	2,1%	0,0%	1,0%	0,3%
171	0,4%	0,5%	1,6%	1,3%	5,9%	1,7%	3,8%	0,6%
172	0,7%	4,3%	0,4%	0,7%	3,1%	5,0%	0,3%	0,2%
16C	1,0%	18,4%	11,2%	3,6%	2,9%	1,6%	1,5%	1,3%
185	1,8%	15,2%	10,8%	1,3%	2,1%	2,9%	5,0%	2,2%
181	1,2%	9,1%	11,3%	5,5%	9,1%	0,9%	7,7%	8,3%
182	7,6%	1,5%	6,0%	6,3%	13,2%	5,1%	11,4%	1,6%
183	0,2%	12,3%	16,8%	5,3%	11,3%	5,7%	9,7%	5,8%
184	2,6%	2,1%	10,2%	5,3%	0,2%	8,0%	2,3%	2,9%
150	1,1%	1,1%	1,8%	0,9%	0,7%	1,3%	1,1%	0,5%
NUTSIII				***t*=3**				
111	1,6%	2,8%	12,1%	9,3%	11,4%	9,8%	5,8%	5,7%
112	0,4%	18,2%	1,3%	1,7%	5,5%	6,2%	2,7%	0,1%
113	0,3%	11,2%	6,5%	1,3%	0,9%	1,4%	0,4%	0,4%
114	0,2%	8,7%	3,1%	0,5%	1,3%	0,2%	0,2%	0,1%
115	5,3%	18,3%	2,3%	3,9%	1,1%	1,6%	3,2%	2,6%
116	0,7%	14,5%	3,0%	0,9%	1,7%	2,9%	0,1%	0,3%
117	0,1%	22,1%	1,0%	0,1%	1,5%	0,9%	0,5%	0,0%
118	0,5%	2,5%	19,9%	1,5%	5,1%	3,6%	1,1%	0,8%
161	1,1%	5,4%	4,8%	1,6%	2,8%	0,2%	1,1%	0,8%
162	1,0%	8,2%	13,1%	5,2%	3,7%	4,0%	3,2%	2,6%
163	0,9%	8,9%	9,3%	1,8%	1,1%	1,4%	0,3%	0,5%
164	2,6%	5,4%	8,2%	3,5%	10,6%	4,3%	3,0%	1,0%
165	1,1%	9,1%	7,2%	3,1%	2,8%	7,3%	0,9%	0,9%
166	25,1%	11,9%	16,4%	25,3%	33,6%	36,8%	16,1%	15,2%
167	0,0%	28,8%	23,7%	1,0%	1,9%	1,5%	1,1%	0,6%
168	0,1%	6,7%	9,8%	0,9%	3,9%	2,4%	0,7%	0,3%
169	9,1%	2,1%	3,1%	7,1%	12,0%	3,2%	4,9%	2,7%
16A	12,8%	29,9%	28,4%	10,8%	0,6%	6,5%	1,2%	3,1%
16B	1,6%	30,6%	0,6%	0,6%	3,7%	1,7%	1,5%	0,4%
171	0,1%	1,7%	0,7%	0,2%	4,6%	0,0%	2,0%	0,1%
172	0,2%	9,0%	7,9%	2,1%	3,4%	1,5%	2,0%	0,8%
16C	0,6%	21,2%	10,6%	2,0%	3,3%	1,8%	0,3%	0,6%
185	3,9%	6,4%	22,5%	7,8%	11,4%	6,5%	5,3%	5,7%
181	15,1%	30,4%	40,4%	17,2%	36,8%	30,4%	26,3%	26,8%
182	1,4%	37,8%	13,9%	3,1%	2,3%	10,2%	2,3%	2,7%
183	1,2%	4,8%	12,5%	4,8%	7,3%	0,8%	2,8%	3,0%
184	0,7%	3,5%	21,6%	4,4%	10,7%	3,8%	1,2%	3,2%
150	0,8%	0,8%	4,6%	3,3%	7,4%	9,1%	0,6%	2,7%

NUTSIII	$\widehat{\mu}_i^{dir}$	$\widehat{\mu}_i^{sinQ}$	$\widehat{\mu}_i^{sinR}$	$\widehat{\mu}_i^{com}$	$\widehat{\mu}_i^{FH}$	$\widehat{\mu}_i^{NS}$	$\widehat{\mu}_{it}^{RY}$	$\widehat{\mu}_{it}^{LP}$
				***t*=4**				
111	0,4%	12,6%	6,8%	7,5%	5,0%	4,2%	5,3%	3,7%
112	0,5%	22,2%	2,3%	0,4%	6,2%	6,0%	1,1%	0,0%
113	0,2%	23,3%	0,5%	0,5%	6,1%	4,7%	0,9%	0,0%
114	0,4%	8,5%	4,8%	1,5%	2,4%	1,2%	0,7%	0,5%
115	0,3%	26,0%	1,0%	1,6%	2,8%	0,5%	1,1%	0,2%
116	1,3%	30,2%	6,2%	2,1%	6,3%	4,0%	1,9%	1,1%
117	0,0%	14,4%	10,1%	1,0%	2,9%	4,1%	0,1%	0,2%
118	0,3%	9,6%	15,8%	0,7%	0,3%	2,1%	0,5%	0,4%
161	2,0%	7,4%	10,5%	5,5%	6,9%	4,6%	3,4%	2,7%
162	0,1%	3,7%	6,7%	2,0%	1,5%	2,2%	1,5%	0,8%
163	1,2%	16,1%	11,9%	3,1%	3,1%	2,2%	1,7%	1,0%
164	1,4%	3,8%	11,8%	3,2%	10,0%	4,3%	2,7%	1,1%
165	0,6%	1,2%	5,4%	5,0%	5,4%	1,2%	0,7%	2,7%
166	0,8%	80,6%	64,1%	0,2%	12,6%	5,3%	2,1%	1,5%
167	0,2%	28,6%	14,0%	1,2%	0,1%	2,6%	0,7%	0,4%
168	0,0%	1,1%	8,8%	1,9%	1,7%	2,6%	2,9%	1,4%
169	1,1%	0,6%	4,7%	0,6%	10,3%	3,8%	8,9%	0,6%
16A	10,3%	38,3%	30,3%	5,2%	5,3%	3,2%	0,7%	1,6%
16B	1,6%	26,0%	3,1%	0,7%	0,9%	0,2%	1,2%	0,3%
171	0,2%	0,0%	0,3%	0,2%	4,6%	0,6%	1,8%	0,1%
172	0,5%	8,7%	5,4%	1,4%	1,2%	0,7%	2,8%	1,0%
16C	0,2%	21,7%	2,9%	0,2%	7,1%	5,7%	2,2%	0,2%
185	0,0%	15,1%	14,6%	5,4%	6,5%	1,0%	5,0%	3,3%
181	18,8%	24,4%	42,4%	19,5%	37,6%	31,2%	27,2%	27,5%
182	3,6%	52,2%	12,5%	8,4%	4,1%	8,9%	3,1%	7,2%
183	1,7%	8,4%	19,5%	4,8%	13,1%	8,4%	3,7%	3,7%
184	1,0%	23,0%	11,0%	3,9%	0,4%	5,3%	0,3%	1,5%
150	0,8%	0,7%	9,0%	3,3%	4,4%	2,5%	2,8%	1,6%
NUTSIII				***t*=5**				
111	1,7%	23,0%	1,4%	1,0%	2,7%	3,1%	1,0%	0,2%
112	1,5%	3,2%	21,1%	7,8%	8,9%	9,1%	6,8%	8,2%
113	2,0%	20,4%	7,0%	0,9%	3,6%	2,4%	1,2%	0,1%
114	0,4%	10,2%	4,6%	1,5%	1,9%	1,1%	0,6%	0,5%
115	1,5%	26,5%	4,5%	2,3%	1,9%	1,1%	0,1%	0,7%
116	0,9%	28,7%	1,6%	0,5%	4,2%	1,7%	0,6%	0,1%
117	0,1%	22,2%	8,9%	1,2%	0,2%	1,1%	0,2%	0,2%
118	0,2%	16,9%	14,2%	1,0%	2,9%	4,2%	0,3%	0,4%
161	1,3%	3,7%	3,0%	1,0%	5,7%	6,7%	0,2%	0,4%
162	0,7%	9,5%	16,1%	3,2%	6,8%	5,4%	3,0%	2,6%
163	1,2%	4,6%	3,0%	0,4%	2,7%	4,5%	1,9%	0,2%
164	4,8%	9,7%	1,5%	4,0%	2,2%	6,0%	11,3%	1,1%
165	0,3%	1,3%	3,0%	1,5%	0,8%	1,7%	2,6%	0,8%
166	3,8%	59,0%	42,2%	3,4%	0,8%	2,4%	1,4%	1,6%
167	0,5%	37,5%	26,8%	1,4%	7,3%	1,7%	1,1%	0,8%
168	2,5%	6,1%	14,8%	0,4%	3,5%	0,0%	2,3%	0,1%
169	0,4%	5,3%	8,8%	2,2%	12,0%	0,9%	7,8%	3,1%
16A	4,8%	55,7%	47,3%	1,3%	19,4%	9,6%	5,3%	5,3%
16B	1,5%	26,1%	0,0%	0,2%	1,5%	0,5%	0,4%	0,1%
171	0,0%	2,6%	7,7%	2,5%	9,6%	4,2%	2,8%	1,4%
172	0,4%	13,9%	5,4%	1,0%	2,0%	0,4%	2,8%	1,3%
16C	2,2%	17,7%	6,6%	0,2%	4,3%	4,8%	1,6%	0,2%
185	2,3%	2,3%	10,3%	2,4%	5,7%	1,6%	3,6%	1,3%
181	15,3%	19,1%	31,1%	16,1%	25,5%	19,4%	21,0%	20,8%
182	0,1%	15,9%	9,3%	1,9%	8,6%	3,1%	2,3%	0,7%
183	0,2%	27,1%	3,0%	0,2%	9,5%	16,6%	4,4%	0,2%
184	0,8%	1,2%	20,6%	4,7%	7,2%	1,2%	4,6%	4,2%
150	0,7%	0,8%	4,8%	2,1%	1,4%	1,1%	3,7%	1,2%

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
				***t*=6**				
111	0,6%	17,6%	4,2%	3,5%	1,0%	1,6%	0,3%	1,5%
112	1,3%	4,1%	18,1%	3,7%	7,3%	7,5%	1,5%	3,4%
113	0,6%	15,9%	8,1%	2,5%	1,5%	0,2%	2,4%	0,8%
114	0,2%	13,6%	3,7%	1,1%	0,6%	0,9%	0,1%	0,2%
115	0,7%	23,6%	3,8%	1,3%	1,4%	1,3%	0,2%	0,3%
116	1,0%	27,3%	2,4%	0,5%	3,4%	0,5%	0,3%	0,5%
117	0,0%	31,3%	2,0%	0,3%	2,6%	1,7%	0,6%	0,1%
118	3,3%	0,9%	23,7%	2,1%	5,8%	4,8%	1,8%	0,7%
161	1,4%	3,0%	6,7%	3,5%	1,7%	0,0%	1,5%	1,5%
162	0,7%	0,5%	0,1%	0,2%	3,0%	4,2%	1,7%	0,1%
163	0,2%	19,4%	17,2%	3,8%	8,5%	3,5%	3,5%	2,2%
164	2,2%	22,3%	13,3%	2,4%	3,2%	12,4%	7,9%	0,8%
165	1,9%	3,2%	1,9%	3,0%	0,5%	2,5%	5,9%	2,1%
166	0,3%	87,5%	74,2%	0,2%	14,3%	5,5%	3,2%	2,0%
167	0,2%	48,4%	39,7%	0,8%	12,2%	4,7%	2,3%	1,1%
168	0,0%	12,5%	20,5%	2,3%	9,8%	5,7%	6,8%	2,3%
169	1,7%	8,2%	14,0%	6,2%	16,3%	1,8%	13,6%	5,6%
16A	2,4%	62,2%	52,2%	9,9%	25,1%	15,0%	10,3%	14,7%
16B	0,7%	36,1%	8,9%	3,4%	6,9%	6,1%	2,6%	1,2%
171	0,0%	0,8%	2,1%	1,4%	2,8%	1,2%	1,5%	0,3%
172	0,2%	7,4%	3,1%	0,7%	3,3%	0,3%	2,1%	0,8%
16C	2,2%	14,9%	19,1%	4,7%	4,4%	2,9%	0,2%	1,9%
185	0,0%	21,2%	5,8%	0,9%	2,4%	2,0%	2,1%	0,7%
181	0,5%	6,9%	2,6%	0,1%	10,1%	15,6%	4,6%	0,1%
182	0,6%	5,7%	5,2%	3,7%	6,9%	0,4%	7,4%	1,6%
183	0,7%	5,9%	13,8%	4,3%	5,9%	0,3%	3,3%	3,3%
184	1,0%	10,2%	18,1%	2,2%	5,7%	1,2%	2,9%	3,1%
150	0,2%	0,2%	9,4%	5,5%	7,4%	10,0%	4,4%	4,4%
NUTSIII				***t*=7**				
111	0,7%	24,8%	3,0%	2,0%	6,5%	4,7%	2,4%	0,1%
112	1,4%	34,6%	8,9%	3,4%	16,7%	14,0%	4,0%	1,3%
113	0,3%	15,9%	6,9%	2,0%	3,0%	1,0%	0,3%	0,6%
114	0,0%	7,9%	8,0%	2,2%	2,0%	1,9%	3,3%	1,7%
115	1,1%	33,3%	6,4%	2,9%	10,7%	4,3%	4,1%	0,9%
116	2,0%	30,5%	6,0%	0,2%	7,2%	3,6%	0,4%	0,3%
117	0,5%	25,5%	1,3%	0,2%	2,6%	1,1%	0,7%	0,0%
118	0,4%	2,3%	20,2%	0,7%	0,7%	0,1%	0,4%	0,2%
161	1,2%	2,6%	2,6%	1,6%	2,3%	1,8%	0,1%	0,8%
162	0,5%	0,9%	3,4%	1,0%	1,5%	0,4%	1,0%	0,4%
163	0,7%	10,7%	9,9%	2,2%	4,0%	0,9%	0,2%	0,8%
164	1,8%	8,6%	4,6%	0,1%	2,8%	3,6%	2,8%	0,4%
165	2,0%	12,3%	14,4%	4,6%	8,2%	7,2%	3,0%	2,1%
166	0,3%	93,1%	87,4%	0,7%	23,1%	9,7%	3,3%	2,4%
167	0,4%	28,3%	20,5%	0,8%	4,0%	0,7%	1,2%	0,6%
168	0,9%	7,0%	11,4%	0,0%	3,1%	0,7%	0,1%	0,2%
169	3,5%	8,7%	11,8%	2,9%	13,6%	0,6%	8,8%	2,7%
16A	2,2%	16,1%	13,7%	5,2%	0,8%	10,8%	0,2%	2,3%
16B	0,8%	30,3%	5,1%	2,0%	5,5%	3,9%	0,4%	0,7%
171	0,0%	1,0%	1,5%	0,5%	0,6%	0,9%	0,7%	0,1%
172	0,7%	4,3%	4,2%	1,0%	5,4%	0,3%	2,1%	0,7%
16C	0,2%	26,0%	10,1%	3,0%	0,2%	4,4%	2,0%	1,1%
185	2,2%	11,6%	9,3%	0,7%	5,8%	0,6%	0,3%	0,4%
181	7,2%	13,0%	24,0%	8,6%	15,4%	11,9%	9,7%	11,3%
182	1,2%	15,4%	3,9%	1,0%	5,6%	0,0%	8,1%	0,9%
183	0,3%	11,1%	4,6%	1,5%	2,1%	4,6%	2,7%	1,1%
184	0,7%	15,7%	30,0%	1,7%	16,8%	11,9%	7,5%	8,2%
150	0,6%	0,7%	1,7%	1,3%	0,9%	4,0%	3,4%	0,7%

A15.6 Rácio de enviesamento absoluto

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=1**							
111	0,15	2,11	4,00	0,78	1,69	0,36	0,50	0,58
112	0,08	1,69	4,87	0,38	0,42	0,04	0,02	0,32
113	0,10	2,01	4,25	0,11	0,71	0,65	0,02	0,09
114	0,28	0,97	4,98	0,55	1,86	1,56	0,57	0,49
115	0,02	4,94	4,45	0,17	1,49	0,93	0,64	0,24
116	0,02	2,12	3,90	0,43	1,67	1,95	0,13	0,27
117	0,26	1,41	5,47	0,62	3,81	3,69	1,32	0,86
118	0,30	1,10	5,78	0,51	0,79	0,03	0,58	0,55
161	0,12	1,56	5,55	0,55	1,67	0,71	0,14	0,45
162	0,08	0,07	0,09	0,04	1,79	1,36	0,37	0,07
163	0,14	1,62	1,69	0,11	1,84	1,57	0,13	0,00
164	0,03	1,22	4,68	0,35	2,34	1,76	1,43	0,45
165	0,02	1,69	5,95	0,33	2,40	1,00	0,64	0,44
166	0,02	10,82	17,13	0,07	2,13	1,47	3,67	6,25
167	0,03	6,34	8,14	0,14	1,70	0,97	2,59	5,86
168	0,01	4,98	2,70	0,17	1,10	1,45	0,21	0,16
169	0,38	3,77	9,98	0,67	2,80	1,17	1,24	0,78
16A	0,40	4,99	6,82	0,14	0,37	0,72	0,32	0,10
16B	0,24	6,46	2,77	0,08	0,89	0,77	0,05	0,10
171	0,03	0,12	0,49	0,19	1,27	0,24	0,78	0,06
172	0,12	1,19	0,95	0,09	0,37	0,18	0,30	0,05
16C	0,05	1,48	10,03	0,26	1,49	1,53	0,23	0,43
185	0,09	2,31	4,11	0,11	0,57	0,47	0,34	0,26
181	0,20	1,20	6,20	0,54	2,26	0,07	1,05	1,74
182	0,19	0,48	2,55	0,33	2,69	0,75	1,08	0,43
183	0,07	2,24	11,32	0,35	3,94	1,94	1,58	1,66
184	0,23	2,48	12,84	0,49	2,31	0,47	0,73	0,82
150	0,13	0,16	2,44	0,56	1,85	1,26	0,79	0,45

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
				***t*=2**				
111	0,04	1,21	6,55	0,77	3,78	1,88	1,48	1,45
112	0,18	1,53	8,24	0,39	1,92	1,52	0,85	0,59
113	0,05	4,33	2,26	0,17	0,20	0,13	0,58	0,12
114	0,08	3,78	0,68	0,04	0,05	0,74	0,57	0,00
115	0,07	5,40	1,17	0,01	0,14	0,76	0,73	0,05
116	0,09	4,66	1,72	0,33	0,66	0,96	0,09	0,25
117	0,01	4,46	2,16	0,29	1,56	1,64	0,33	0,24
118	0,19	2,23	7,97	0,38	2,13	1,58	1,22	0,83
161	0,28	2,58	5,74	0,61	2,14	1,22	1,30	0,61
162	0,02	1,34	2,54	0,41	0,01	0,06	0,67	0,38
163	0,01	3,27	5,34	0,21	0,75	0,10	0,57	0,20
164	0,10	2,22	0,96	0,13	0,76	0,75	1,53	0,07
165	0,12	0,93	0,21	0,03	0,45	0,26	0,30	0,03
166	0,03	10,40	8,48	0,03	1,85	0,32	1,24	5,12
167	0,01	6,86	6,77	0,10	1,70	4,13	1,06	5,05
168	0,06	3,15	0,47	0,00	0,27	0,89	0,57	0,05
169	0,00	2,34	2,52	0,61	0,31	1,56	0,93	0,71
16A	0,22	7,89	9,36	0,17	1,40	0,24	0,48	0,88
16B	0,12	9,55	0,85	0,14	0,81	0,00	0,32	0,07
171	0,06	0,07	0,37	0,24	1,65	0,56	1,31	0,16
172	0,08	0,61	0,12	0,12	1,12	1,53	0,08	0,04
16C	0,10	5,63	4,43	0,42	1,02	0,56	0,49	0,34
185	0,14	2,49	4,25	0,10	0,44	0,48	0,69	0,30
181	0,07	1,39	4,46	0,41	1,93	0,06	1,58	1,54
182	0,33	0,20	2,72	0,31	2,83	0,79	1,54	0,21
183	0,02	1,96	9,13	0,46	2,70	1,10	1,49	1,06
184	0,23	0,29	4,34	0,61	0,04	1,28	0,45	0,57
150	0,13	0,14	0,55	0,16	0,24	0,37	0,34	0,14
NUTSIII				***t*=3**				
111	0,09	0,78	5,84	0,84	3,79	1,90	0,93	0,91
112	0,04	4,46	0,47	0,19	1,27	1,00	0,43	0,02
113	0,04	2,92	2,70	0,17	0,44	0,56	0,12	0,12
114	0,04	2,33	0,91	0,12	0,73	0,12	0,13	0,06
115	0,47	4,49	0,71	0,38	0,20	0,25	0,47	0,37
116	0,10	3,68	1,29	0,16	0,75	0,91	0,06	0,14
117	0,01	5,25	0,30	0,02	0,98	0,66	0,23	0,02
118	0,08	0,76	7,09	0,21	1,16	0,88	0,39	0,30
161	0,10	1,37	2,06	0,21	0,79	0,04	0,20	0,13
162	0,10	1,80	3,07	0,53	1,47	1,06	0,63	0,52
163	0,11	1,95	3,57	0,23	0,43	0,44	0,08	0,12
164	0,26	1,36	3,01	0,38	1,46	0,95	0,56	0,21
165	0,11	1,99	2,64	0,35	0,72	0,97	0,21	0,20
166	1,00	3,23	4,81	1,03	2,64	2,74	0,76	0,69
167	0,00	5,35	5,62	0,12	1,20	1,48	1,36	1,72
168	0,02	1,72	3,50	0,15	0,76	0,55	0,73	0,49
169	0,43	0,52	1,33	0,43	1,74	0,38	0,48	0,27
16A	0,65	5,50	9,16	0,58	0,10	0,51	0,13	0,29
16B	0,16	10,56	0,18	0,09	1,39	0,40	0,33	0,08
171	0,02	0,38	0,12	0,04	1,25	0,01	0,55	0,03
172	0,03	1,80	1,67	0,32	1,09	0,44	0,53	0,23
16C	0,07	6,43	4,04	0,22	0,88	0,51	0,06	0,13
185	0,25	0,28	6,45	0,51	1,96	0,93	0,60	0,62
181	0,60	2,03	18,92	0,70	6,40	2,66	2,32	2,17
182	0,09	1,28	5,04	0,21	0,47	1,34	0,28	0,32
183	0,10	0,21	5,48	0,45	1,91	0,15	0,64	0,74
184	0,08	0,17	11,42	0,43	2,29	0,80	0,27	0,67
150	0,07	0,07	1,05	0,43	2,82	2,80	0,08	0,52

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=4**							
111	0,02	2,63	3,09	0,86	1,66	0,78	0,93	0,61
112	0,05	4,29	0,97	0,06	1,57	1,04	0,25	0,00
113	0,02	4,46	0,21	0,07	2,12	1,59	0,22	0,01
114	0,07	1,84	1,39	0,33	1,13	0,67	0,33	0,21
115	0,02	4,87	0,37	0,17	0,47	0,08	0,17	0,04
116	0,13	5,46	2,58	0,27	1,72	1,14	0,35	0,22
117	0,01	2,96	4,24	0,17	1,66	1,61	0,34	1,66
118	0,04	2,06	5,83	0,09	0,11	0,63	0,18	0,15
161	0,19	1,80	4,79	0,65	2,33	1,00	0,64	0,51
162	0,01	0,81	1,62	0,29	0,67	0,68	0,37	0,21
163	0,13	3,13	4,42	0,32	0,99	0,58	0,34	0,21
164	0,15	0,89	5,27	0,37	1,30	0,98	0,54	0,31
165	0,04	0,26	2,48	0,56	1,43	0,18	0,10	0,41
166	0,08	10,07	14,34	0,02	2,36	1,48	1,21	1,79
167	0,03	5,02	3,65	0,15	0,05	1,89	1,02	1,62
168	0,00	0,25	3,34	0,15	0,24	0,34	0,30	0,15
169	0,07	0,14	2,18	0,06	1,38	0,48	0,81	0,08
16A	0,49	6,25	10,22	0,23	0,76	0,25	0,06	0,12
16B	0,16	7,92	0,84	0,10	0,32	0,04	0,24	0,05
171	0,03	0,00	0,05	0,04	1,18	0,19	0,45	0,03
172	0,06	1,50	1,20	0,21	0,37	0,20	0,56	0,22
16C	0,02	6,26	1,21	0,02	1,48	1,24	0,35	0,03
185	0,00	0,48	4,30	0,44	1,55	0,16	0,80	0,56
181	0,68	1,19	21,29	0,75	5,74	2,65	2,07	1,88
182	0,15	1,27	4,09	0,41	0,92	1,43	0,34	0,76
183	0,19	0,29	8,52	0,47	2,83	1,57	0,69	0,74
184	0,12	0,69	5,16	0,48	0,12	1,06	0,08	0,33
150	0,11	0,11	1,88	0,46	1,40	0,68	0,59	0,36
NUTSIII	***t*=5**							
111	0,15	4,53	0,66	0,13	1,05	0,60	0,21	0,04
112	0,09	0,76	11,63	0,55	3,25	1,91	1,31	1,55
113	0,22	4,11	3,51	0,11	1,30	0,82	0,31	0,01
114	0,07	2,25	1,26	0,32	0,95	0,58	0,24	0,23
115	0,19	5,08	1,85	0,34	0,49	0,30	0,04	0,22
116	0,10	5,40	0,75	0,09	1,40	0,56	0,18	0,02
117	0,02	4,39	3,42	0,20	0,15	0,95	0,45	2,67
118	0,04	3,50	5,00	0,15	0,93	1,03	0,12	0,18
161	0,13	0,70	1,14	0,15	2,10	1,41	0,06	0,10
162	0,08	1,74	3,29	0,33	2,21	1,39	0,63	0,56
163	0,12	0,88	1,22	0,05	1,05	1,35	0,43	0,04
164	0,26	1,77	0,63	0,28	0,44	1,20	1,15	0,13
165	0,02	0,26	1,51	0,12	0,19	0,24	0,44	0,14
166	0,38	7,40	9,58	0,34	0,28	0,65	0,48	0,53
167	0,06	5,44	8,99	0,17	1,59	1,96	1,14	3,21
168	0,23	1,31	6,44	0,04	0,54	0,00	0,27	0,01
169	0,02	1,11	4,61	0,15	1,67	0,11	0,84	0,37
16A	0,31	7,14	15,66	0,09	2,31	0,72	0,91	0,92
16B	0,15	7,05	0,01	0,03	0,49	0,12	0,10	0,01
171	0,01	0,54	1,58	0,42	2,18	1,29	0,79	0,42
172	0,04	2,48	1,12	0,12	0,56	0,11	0,48	0,24
16C	0,18	4,29	2,47	0,02	1,29	1,30	0,28	0,04
185	0,20	0,10	3,63	0,23	1,37	0,27	0,69	0,24
181	0,62	1,06	11,72	0,70	3,35	1,46	1,62	1,47
182	0,01	0,62	4,96	0,30	1,79	0,70	0,83	0,30
183	0,02	0,96	1,11	0,02	1,69	1,89	0,53	0,02
184	0,09	0,05	12,39	0,47	2,03	0,24	0,81	0,89
150	0,08	0,10	0,94	0,31	0,49	0,31	0,72	0,28

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
				***t*=6**				
111	0,05	4,39	2,29	0,55	0,36	0,33	0,07	0,36
112	0,11	1,15	8,70	0,31	2,61	1,56	0,35	0,79
113	0,06	4,02	3,83	0,31	0,63	0,08	0,67	0,17
114	0,04	3,52	1,39	0,24	0,33	0,49	0,06	0,08
115	0,09	5,60	1,35	0,18	0,43	0,38	0,06	0,11
116	0,12	6,29	1,18	0,07	0,96	0,15	0,08	0,14
117	0,01	7,00	0,65	0,05	0,85	0,83	0,49	0,10
118	0,36	0,27	9,16	0,24	1,64	1,34	0,40	0,16
161	0,15	0,80	3,78	0,47	0,67	0,00	0,37	0,33
162	0,08	0,14	0,04	0,04	1,19	1,20	0,44	0,03
163	0,02	4,20	6,77	0,35	2,52	0,97	0,68	0,42
164	0,16	4,71	4,46	0,19	0,76	1,92	0,99	0,15
165	0,11	0,81	0,97	0,25	0,10	0,34	0,89	0,26
166	0,03	12,05	16,74	0,02	1,72	1,50	1,75	7,06
167	0,03	8,42	13,18	0,10	1,73	2,04	1,95	6,92
168	0,00	3,70	7,57	0,26	1,46	0,95	0,88	0,47
169	0,10	2,32	7,12	0,43	2,30	0,23	1,34	0,65
16A	0,11	9,90	15,35	0,33	2,35	0,94	0,71	0,81
16B	0,10	14,57	3,92	0,50	1,99	1,40	0,75	0,33
171	0,01	0,17	0,47	0,29	0,77	0,36	0,49	0,08
172	0,03	1,41	0,89	0,11	0,97	0,07	0,50	0,19
16C	0,26	4,52	7,81	0,48	1,38	0,88	0,06	0,46
185	0,00	4,55	2,71	0,11	0,67	0,40	0,44	0,16
181	0,03	1,09	0,80	0,01	1,26	0,92	0,55	0,02
182	0,04	0,91	2,69	0,32	1,61	0,08	1,65	0,34
183	0,06	1,07	8,20	0,46	1,90	0,05	0,92	0,90
184	0,12	1,93	11,54	0,26	1,60	0,24	0,88	1,00
150	0,02	0,02	3,04	0,69	2,65	3,03	1,18	1,12
NUTSIII				***t*=7**				
111	0,04	2,54	1,48	0,18	1,57	0,77	0,41	0,02
112	0,11	3,30	4,19	0,32	2,40	1,81	0,62	0,23
113	0,03	1,76	3,63	0,27	1,10	0,33	0,07	0,15
114	0,00	0,94	3,27	0,27	0,76	0,75	0,69	0,35
115	0,11	3,20	2,80	0,32	1,73	0,80	0,60	0,16
116	0,23	2,99	2,96	0,02	1,59	0,95	0,09	0,09
117	0,08	2,61	0,51	0,04	1,33	1,05	2,87	0,80
118	0,07	0,29	8,93	0,13	0,71	0,06	0,78	0,68
161	0,11	0,74	1,28	0,24	0,82	0,39	0,01	0,15
162	0,07	0,24	1,25	0,20	0,59	0,11	0,28	0,11
163	0,08	2,64	4,58	0,27	1,26	0,27	0,05	0,21
164	0,17	2,16	2,12	0,01	0,75	0,84	0,60	0,11
165	0,22	3,84	8,76	0,49	1,52	1,08	0,58	0,44
166	0,03	13,20	23,72	0,07	1,92	1,83	4,36	5,62
167	0,05	6,04	6,23	0,10	1,37	2,81	3,61	4,87
168	0,14	2,07	5,56	0,00	0,78	0,17	0,03	0,08
169	0,25	2,60	6,60	0,22	2,34	0,08	1,38	0,42
16A	0,18	3,79	6,33	0,54	0,20	0,95	0,05	0,68
16B	0,09	11,92	1,78	0,28	2,27	0,95	0,11	0,16
171	0,00	0,26	0,35	0,11	0,21	0,28	0,28	0,06
172	0,09	1,14	1,22	0,15	1,57	0,08	0,50	0,18
16C	0,02	9,64	3,88	0,34	0,08	1,18	0,47	0,23
185	0,25	0,92	3,72	0,09	1,88	0,13	0,09	0,14
181	0,46	1,05	10,85	0,54	1,99	0,93	0,96	1,03
182	0,07	0,94	2,15	0,08	1,14	0,00	0,90	0,13
183	0,03	0,70	1,97	0,18	0,72	0,90	0,57	0,23
184	0,07	1,31	22,52	0,16	3,35	2,09	0,95	1,06
150	0,06	0,08	0,50	0,22	0,28	1,09	0,67	0,18

A15.7 Erro relativo absoluto

NUTSIII	$\hat{\boldsymbol{\mu}}_i^{dir}$	$\hat{\boldsymbol{\mu}}_i^{sinQ}$	$\hat{\boldsymbol{\mu}}_i^{sinR}$	$\hat{\boldsymbol{\mu}}_i^{com}$	$\hat{\boldsymbol{\mu}}_i^{FH}$	$\hat{\boldsymbol{\mu}}_i^{NS}$	$\hat{\boldsymbol{\mu}}_{it}^{RY}$	$\hat{\boldsymbol{\mu}}_{it}^{LP}$
	***t*=1**							
111	12,9%	14,3%	8,1%	10,5%	5,8%	4,8%	5,3%	5,5%
112	7,9%	11,2%	12,5%	8,6%	3,1%	4,4%	3,5%	3,9%
113	5,1%	13,6%	9,4%	5,3%	1,6%	1,8%	1,4%	1,4%
114	5,3%	7,2%	10,5%	6,3%	6,1%	4,2%	3,0%	2,9%
115	9,5%	41,5%	12,6%	9,8%	11,5%	7,4%	6,7%	5,1%
116	8,0%	14,4%	7,5%	7,4%	4,8%	6,9%	3,3%	3,5%
117	13,6%	9,4%	14,4%	12,0%	10,7%	11,1%	7,9%	6,9%
118	9,8%	7,7%	16,9%	10,7%	5,6%	5,4%	6,8%	6,8%
161	6,7%	6,4%	9,6%	6,9%	6,0%	4,3%	3,0%	3,4%
162	8,6%	3,5%	2,1%	4,9%	5,2%	5,6%	4,5%	4,2%
163	8,8%	7,7%	3,5%	6,6%	4,7%	5,2%	3,9%	4,3%
164	10,2%	5,3%	11,8%	9,5%	16,2%	7,1%	7,6%	3,3%
165	13,1%	6,8%	12,2%	12,9%	12,3%	8,4%	8,3%	7,6%
166	8,4%	86,3%	74,6%	8,5%	14,2%	4,2%	3,2%	2,3%
167	6,1%	37,2%	27,9%	6,4%	4,6%	0,3%	1,2%	0,9%
168	9,1%	27,1%	13,8%	9,4%	9,3%	19,7%	5,0%	5,0%
169	10,1%	13,9%	18,9%	12,0%	22,6%	9,2%	13,4%	7,0%
16A	21,2%	27,2%	19,5%	23,9%	5,7%	11,0%	10,5%	15,0%
16B	7,8%	21,7%	7,3%	7,5%	2,9%	4,8%	3,8%	4,2%
171	5,3%	3,9%	3,2%	3,9%	4,7%	2,6%	3,6%	2,8%
172	5,0%	6,7%	3,4%	4,0%	2,3%	2,6%	2,6%	2,4%
16C	7,6%	6,2%	24,2%	8,4%	5,1%	6,9%	3,5%	4,0%
185	9,1%	14,1%	9,3%	9,1%	3,9%	4,9%	5,5%	5,5%
181	17,5%	8,6%	12,7%	13,4%	9,5%	13,3%	6,4%	9,1%
182	13,4%	6,6%	4,7%	9,9%	12,2%	6,2%	8,9%	6,8%
183	9,3%	13,8%	19,1%	9,4%	13,7%	8,9%	6,7%	7,1%
184	6,4%	15,0%	23,6%	7,5%	11,1%	4,4%	4,7%	5,2%
150	6,8%	6,1%	7,4%	6,7%	5,7%	5,7%	4,8%	3,9%

NUTSIII	$\widehat{\mu}_i^{dir}$	$\widehat{\mu}_i^{sinQ}$	$\widehat{\mu}_i^{sinR}$	$\widehat{\mu}_i^{com}$	$\widehat{\mu}_i^{FH}$	$\widehat{\mu}_i^{NS}$	$\widehat{\mu}_{it}^{RY}$	$\widehat{\mu}_{it}^{LP}$
	t=2							
111	12,6%	4,1%	13,9%	11,8%	11,7%	9,8%	6,9%	6,7%
112	6,5%	4,5%	23,1%	7,8%	10,2%	9,4%	5,4%	3,4%
113	5,9%	14,8%	6,6%	5,8%	1,9%	2,1%	3,3%	2,6%
114	4,3%	12,6%	2,6%	3,4%	1,5%	2,0%	2,4%	1,7%
115	9,5%	19,1%	4,4%	8,0%	4,0%	6,5%	6,3%	4,8%
116	7,2%	16,1%	4,6%	5,7%	2,6%	3,9%	2,8%	3,3%
117	5,3%	15,3%	7,3%	5,2%	3,1%	4,1%	2,1%	1,9%
118	7,7%	6,2%	25,6%	8,5%	11,3%	8,5%	8,3%	6,5%
161	9,5%	9,4%	11,2%	9,4%	8,2%	6,4%	6,5%	5,3%
162	6,3%	6,0%	8,0%	6,4%	1,9%	2,8%	3,3%	3,2%
163	5,3%	15,0%	13,1%	5,7%	2,2%	2,0%	2,1%	1,9%
164	8,3%	9,7%	3,9%	6,4%	5,1%	5,2%	9,1%	3,6%
165	10,7%	4,7%	2,2%	6,2%	3,9%	5,5%	4,4%	4,6%
166	8,1%	71,3%	57,8%	8,2%	7,4%	3,1%	6,9%	2,0%
167	6,0%	37,8%	29,6%	6,2%	5,3%	0,5%	3,6%	1,0%
168	26,4%	14,4%	5,2%	23,1%	9,3%	17,5%	14,4%	13,1%
169	12,8%	10,3%	6,5%	9,3%	6,3%	13,4%	10,4%	5,4%
16A	17,0%	46,1%	38,7%	21,3%	12,5%	13,1%	8,2%	14,6%
16B	6,8%	27,6%	2,7%	4,3%	2,8%	3,4%	2,6%	3,2%
171	6,2%	5,5%	3,7%	4,3%	6,1%	2,9%	4,0%	3,0%
172	7,2%	6,5%	2,7%	4,2%	3,4%	5,2%	3,1%	3,5%
16C	7,2%	18,4%	11,2%	7,5%	3,5%	2,6%	2,8%	3,0%
185	10,7%	15,2%	10,8%	10,8%	4,2%	5,2%	7,4%	6,3%
181	14,3%	9,7%	11,3%	12,3%	9,4%	12,8%	7,9%	8,7%
182	19,2%	6,0%	6,0%	14,8%	13,3%	6,4%	12,4%	6,4%
183	8,7%	12,5%	16,8%	10,9%	11,4%	6,4%	10,1%	6,8%
184	8,4%	6,0%	10,2%	8,2%	2,8%	8,1%	4,0%	4,2%
150	7,0%	6,1%	3,0%	4,1%	2,2%	3,1%	2,7%	3,1%
NUTSIII	*t*=3							
111	14,4%	3,6%	12,1%	12,7%	11,4%	9,9%	6,8%	6,7%
112	9,4%	18,2%	2,5%	5,8%	6,2%	7,4%	5,6%	4,3%
113	7,0%	11,2%	6,5%	6,2%	1,6%	2,1%	2,2%	2,4%
114	4,0%	8,8%	3,9%	3,7%	1,9%	1,3%	1,5%	1,5%
115	10,2%	18,3%	3,2%	8,6%	4,3%	5,6%	5,5%	5,3%
116	5,4%	14,5%	3,2%	4,5%	2,3%	3,4%	1,8%	1,8%
117	5,5%	22,1%	2,7%	4,3%	1,7%	1,2%	1,5%	1,4%
118	5,1%	3,4%	19,9%	5,5%	5,3%	3,8%	1,9%	1,8%
161	8,3%	5,8%	4,8%	6,4%	3,9%	4,0%	4,4%	4,6%
162	8,1%	8,3%	13,1%	9,6%	3,9%	4,4%	4,9%	4,6%
163	6,4%	9,0%	9,3%	6,8%	2,2%	2,7%	3,1%	3,2%
164	8,0%	5,7%	8,2%	8,0%	11,0%	5,2%	4,9%	3,9%
165	7,9%	9,2%	7,2%	7,4%	3,7%	8,1%	3,1%	3,3%
166	29,1%	11,9%	16,4%	28,8%	33,6%	36,8%	17,6%	17,1%
167	6,4%	28,8%	23,7%	6,7%	1,9%	1,5%	1,1%	0,6%
168	4,4%	6,9%	9,8%	4,6%	3,9%	2,6%	0,8%	0,4%
169	17,7%	3,7%	3,3%	11,2%	12,7%	7,2%	8,0%	7,4%
16A	17,8%	29,9%	28,4%	16,6%	4,9%	8,6%	6,9%	7,1%
16B	8,0%	30,6%	2,7%	4,7%	4,0%	3,5%	3,9%	4,0%
171	5,0%	3,8%	4,4%	4,3%	5,0%	2,5%	3,2%	2,7%
172	5,6%	9,2%	8,0%	5,8%	3,8%	3,2%	3,4%	3,0%
16C	7,6%	21,2%	10,6%	7,8%	4,2%	3,1%	3,2%	3,3%
185	12,7%	19,4%	22,5%	14,4%	11,8%	8,0%	8,1%	8,5%
181	25,3%	30,6%	40,4%	25,1%	36,9%	30,5%	26,6%	27,2%
182	12,4%	38,0%	13,9%	14,0%	4,0%	11,5%	7,0%	7,2%
183	8,6%	19,1%	12,5%	9,1%	7,4%	4,2%	4,2%	4,0%
184	7,0%	18,4%	21,6%	8,6%	10,7%	5,0%	3,4%	4,0%
150	10,0%	9,6%	5,3%	6,6%	7,4%	9,1%	5,2%	4,6%

NUTSIII	$\widehat{\mu}_i^{dir}$	$\widehat{\mu}_i^{sinQ}$	$\widehat{\mu}_i^{sinR}$	$\widehat{\mu}_i^{com}$	$\widehat{\mu}_i^{FH}$	$\widehat{\mu}_i^{NS}$	$\widehat{\mu}_{it}^{RY}$	$\widehat{\mu}_{it}^{LP}$
	t=4							
111	13,7%	12,6%	6,8%	9,5%	5,1%	5,1%	6,4%	5,6%
112	7,9%	22,2%	2,7%	5,2%	6,5%	6,7%	3,6%	3,6%
113	7,6%	23,3%	1,9%	4,8%	6,2%	5,0%	3,4%	3,5%
114	4,3%	8,6%	5,0%	4,0%	2,7%	1,8%	1,8%	1,8%
115	8,9%	26,0%	2,3%	7,1%	5,9%	4,5%	5,8%	5,0%
116	7,8%	30,2%	6,2%	6,7%	6,6%	4,5%	4,4%	4,1%
117	4,4%	14,4%	10,1%	4,7%	2,9%	4,1%	0,2%	0,2%
118	5,3%	9,7%	15,8%	5,8%	2,3%	3,4%	2,4%	2,3%
161	8,6%	7,5%	10,5%	8,9%	7,0%	5,1%	5,0%	4,7%
162	6,1%	4,8%	6,8%	5,7%	2,2%	3,2%	3,4%	3,1%
163	7,8%	16,1%	11,9%	8,4%	3,8%	3,5%	4,3%	4,1%
164	6,7%	4,7%	11,8%	7,2%	10,1%	4,5%	3,4%	2,3%
165	12,0%	3,7%	5,4%	7,8%	5,7%	5,7%	6,1%	5,7%
166	8,2%	80,6%	64,1%	8,2%	12,6%	5,3%	2,1%	1,5%
167	6,3%	28,6%	14,0%	6,7%	0,9%	2,6%	0,7%	0,4%
168	11,6%	3,7%	8,8%	11,9%	5,3%	7,5%	9,0%	7,8%
169	12,7%	3,6%	4,8%	8,5%	11,8%	7,2%	12,2%	5,9%
16A	18,3%	38,3%	30,3%	19,4%	8,0%	8,8%	9,8%	11,0%
16B	8,1%	26,0%	3,8%	5,4%	2,3%	3,4%	4,1%	3,9%
171	5,4%	4,2%	4,3%	4,4%	5,2%	2,7%	3,4%	3,2%
172	6,3%	9,0%	5,7%	5,7%	2,7%	2,9%	4,7%	3,6%
16C	9,5%	21,7%	3,2%	7,7%	7,8%	6,6%	5,6%	5,3%
185	11,0%	29,0%	14,6%	11,3%	6,9%	4,9%	6,5%	5,5%
181	27,8%	26,0%	42,4%	26,8%	37,6%	31,2%	27,5%	27,8%
182	20,1%	52,4%	12,5%	18,5%	4,8%	9,8%	6,8%	9,6%
183	6,9%	27,7%	19,5%	8,7%	13,1%	8,7%	5,0%	4,8%
184	6,5%	31,0%	11,0%	7,5%	2,5%	5,6%	2,9%	3,5%
150	6,2%	5,2%	9,1%	6,5%	4,6%	3,6%	4,3%	3,6%
NUTSIII	*t*=5							
111	8,9%	23,0%	2,1%	4,8%	3,2%	5,2%	3,8%	4,0%
112	13,7%	4,5%	21,1%	13,9%	8,9%	9,1%	7,3%	8,5%
113	7,1%	20,4%	7,0%	6,8%	4,0%	3,1%	3,2%	3,1%
114	4,3%	10,3%	5,0%	4,0%	2,3%	1,7%	1,8%	1,8%
115	6,3%	26,5%	4,6%	5,7%	3,5%	3,0%	2,8%	2,6%
116	6,7%	28,7%	2,1%	4,2%	4,5%	2,8%	2,7%	2,8%
117	4,9%	22,2%	8,9%	5,2%	0,7%	1,2%	0,2%	0,2%
118	5,0%	16,9%	14,2%	5,3%	3,6%	4,8%	1,8%	1,9%
161	7,7%	5,2%	3,3%	4,8%	5,8%	7,1%	3,2%	3,3%
162	6,9%	9,7%	16,1%	8,5%	6,9%	5,7%	4,5%	4,3%
163	7,6%	5,8%	3,2%	5,3%	3,0%	4,7%	3,8%	3,9%
164	14,8%	9,9%	2,2%	9,6%	4,0%	7,0%	12,3%	6,0%
165	11,8%	4,3%	3,1%	8,6%	3,2%	5,6%	4,7%	4,5%
166	8,6%	59,0%	42,2%	8,5%	2,2%	2,8%	2,1%	2,3%
167	6,3%	37,5%	26,8%	6,6%	7,3%	1,7%	1,1%	0,8%
168	9,3%	6,5%	14,8%	10,3%	6,3%	5,0%	8,2%	6,3%
169	15,1%	5,9%	8,8%	12,7%	13,1%	6,3%	10,8%	7,3%
16A	12,2%	55,7%	47,3%	11,5%	19,6%	15,3%	6,9%	7,0%
16B	7,9%	26,1%	3,0%	4,9%	2,8%	3,4%	3,4%	3,9%
171	4,5%	4,2%	7,9%	5,0%	9,7%	4,5%	3,5%	2,8%
172	8,1%	13,9%	5,9%	7,0%	3,3%	3,1%	5,3%	4,5%
16C	10,0%	17,7%	6,6%	8,7%	4,8%	5,3%	4,4%	5,0%
185	9,2%	21,7%	10,3%	9,3%	6,2%	4,9%	5,3%	4,3%
181	24,1%	21,3%	31,1%	23,5%	25,6%	20,3%	22,0%	22,0%
182	5,3%	23,7%	9,3%	5,7%	8,7%	4,0%	2,7%	2,0%
183	10,9%	28,3%	3,3%	7,8%	10,1%	17,1%	8,0%	5,5%
184	7,0%	21,5%	20,6%	8,6%	7,3%	3,7%	5,6%	4,9%
150	7,0%	6,4%	5,6%	5,6%	2,5%	2,9%	5,1%	3,6%

NUTSIII	$\widehat{\mu}_i^{dir}$	$\widehat{\mu}_i^{sinQ}$	$\widehat{\mu}_i^{sinR}$	$\widehat{\mu}_i^{com}$	$\widehat{\mu}_i^{FH}$	$\widehat{\mu}_i^{NS}$	$\widehat{\mu}_{it}^{RY}$	$\widehat{\mu}_{it}^{LP}$
	***t*=6**							
111	9,1%	17,6%	4,2%	5,5%	2,1%	4,0%	3,6%	3,4%
112	9,5%	4,5%	18,1%	10,6%	7,3%	7,8%	3,6%	4,5%
113	7,4%	15,9%	8,1%	7,3%	2,3%	2,1%	3,4%	3,7%
114	4,1%	13,6%	3,9%	3,7%	1,5%	1,7%	1,6%	1,6%
115	6,3%	23,6%	4,1%	5,6%	2,6%	2,7%	1,8%	1,9%
116	6,4%	27,3%	2,6%	5,0%	4,2%	2,4%	2,6%	2,6%
117	5,2%	31,3%	2,9%	4,6%	2,6%	1,8%	0,9%	0,6%
118	7,3%	2,8%	23,7%	6,8%	6,3%	5,3%	3,5%	3,1%
161	7,6%	3,9%	6,7%	6,6%	2,4%	3,6%	3,3%	3,6%
162	7,4%	3,1%	2,3%	4,2%	3,3%	4,7%	3,4%	3,5%
163	7,7%	19,4%	17,2%	9,6%	8,5%	4,3%	5,2%	4,8%
164	10,6%	22,3%	13,3%	10,8%	4,5%	12,5%	8,7%	4,2%
165	14,1%	4,2%	2,2%	7,5%	3,5%	6,1%	7,1%	6,3%
166	8,2%	87,5%	74,2%	8,3%	14,3%	5,5%	3,2%	2,0%
167	6,2%	48,4%	39,7%	6,3%	12,2%	4,7%	2,3%	1,1%
168	5,8%	12,5%	20,5%	7,3%	10,3%	6,5%	8,6%	4,4%
169	13,6%	8,3%	14,0%	14,2%	16,9%	6,4%	15,4%	8,9%
16A	18,8%	62,2%	52,2%	28,2%	26,0%	20,7%	16,5%	21,6%
16B	5,7%	36,1%	8,9%	6,5%	7,0%	6,4%	3,5%	3,0%
171	4,5%	3,9%	4,0%	4,0%	3,7%	2,8%	2,6%	2,7%
172	6,7%	7,9%	3,8%	5,0%	4,0%	3,0%	3,8%	3,5%
16C	6,9%	14,9%	19,1%	8,6%	4,8%	3,4%	2,4%	3,5%
185	8,0%	21,2%	5,8%	7,2%	3,6%	4,1%	4,2%	3,8%
181	13,8%	7,5%	3,3%	9,7%	10,4%	18,2%	8,7%	6,5%
182	10,6%	6,6%	5,2%	8,9%	7,1%	3,9%	7,6%	3,9%
183	8,4%	7,0%	13,8%	8,6%	5,9%	4,5%	4,0%	4,2%
184	6,9%	10,4%	18,1%	7,1%	6,0%	4,1%	3,7%	3,9%
150	9,3%	8,8%	9,4%	8,4%	7,5%	10,0%	4,7%	4,8%
NUTSIII	***t*=7**							
111	12,4%	24,8%	3,1%	7,7%	7,2%	6,7%	4,8%	5,0%
112	9,6%	34,6%	8,9%	8,9%	16,8%	14,3%	6,0%	4,5%
113	6,5%	15,9%	6,9%	6,6%	3,6%	2,5%	3,1%	3,3%
114	7,4%	8,5%	8,0%	7,2%	2,7%	2,6%	5,0%	4,4%
115	8,2%	33,3%	6,4%	7,9%	11,4%	5,6%	6,3%	4,7%
116	7,0%	30,5%	6,0%	6,5%	7,7%	4,5%	3,3%	3,2%
117	5,1%	25,5%	2,3%	4,1%	2,6%	1,1%	0,7%	0,0%
118	4,1%	6,6%	20,2%	4,2%	0,7%	0,5%	0,4%	0,2%
161	8,6%	3,6%	2,8%	4,9%	3,0%	4,2%	3,8%	4,2%
162	5,7%	3,0%	3,7%	4,3%	2,4%	2,8%	3,0%	2,8%
163	6,8%	10,7%	9,9%	7,4%	4,5%	2,9%	3,1%	3,3%
164	7,9%	8,6%	4,6%	6,5%	3,4%	4,5%	3,6%	2,6%
165	6,8%	12,3%	14,4%	8,3%	8,6%	7,9%	4,5%	3,8%
166	8,1%	93,1%	87,4%	8,1%	23,1%	9,7%	3,3%	2,4%
167	6,1%	28,3%	20,5%	6,4%	4,0%	0,7%	1,2%	0,6%
168	5,1%	7,1%	11,4%	5,4%	3,9%	3,0%	2,0%	2,0%
169	11,4%	8,7%	11,8%	11,7%	13,8%	6,0%	10,0%	5,8%
16A	8,7%	16,1%	13,7%	8,9%	3,1%	11,3%	3,1%	3,1%
16B	6,9%	30,3%	5,2%	6,1%	5,5%	4,8%	3,2%	3,6%
171	3,9%	3,0%	3,7%	3,6%	2,4%	2,6%	2,0%	2,1%
172	6,1%	4,8%	4,6%	5,3%	5,6%	3,0%	3,8%	3,3%
16C	7,1%	26,0%	10,1%	8,1%	2,1%	5,0%	3,7%	3,8%
185	6,9%	13,9%	9,3%	6,7%	5,9%	3,7%	2,6%	2,5%
181	13,9%	14,6%	24,0%	14,5%	16,2%	14,5%	11,6%	13,2%
182	13,5%	16,6%	3,9%	9,4%	6,6%	4,3%	10,6%	5,9%
183	8,1%	14,6%	4,6%	6,3%	2,9%	5,5%	4,4%	4,0%
184	8,4%	16,5%	30,0%	9,2%	17,1%	12,5%	9,8%	10,5%
150	7,1%	6,6%	3,1%	4,3%	2,6%	4,6%	4,9%	3,1%

A15.8 Erro padrão relativo

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=1**							
111	16,2%	15,8%	8,4%	12,7%	6,6%	6,5%	7,3%	7,4%
112	9,9%	13,0%	12,7%	10,2%	3,9%	5,4%	4,4%	4,9%
113	6,6%	15,2%	9,6%	6,6%	2,0%	2,3%	1,8%	1,8%
114	6,6%	8,8%	10,7%	7,4%	6,8%	4,8%	3,8%	3,7%
115	11,7%	42,3%	12,9%	11,7%	12,8%	8,9%	7,7%	6,2%
116	10,1%	16,0%	7,8%	8,6%	5,4%	7,6%	4,2%	4,4%
117	15,0%	11,2%	14,6%	13,5%	11,1%	11,5%	9,5%	8,9%
118	11,7%	9,4%	17,2%	12,4%	7,2%	6,9%	8,5%	8,6%
161	8,5%	7,4%	9,8%	8,1%	6,7%	5,6%	3,9%	4,5%
162	10,6%	4,3%	2,6%	7,3%	5,7%	6,4%	5,6%	5,3%
163	10,7%	8,8%	3,9%	8,5%	5,3%	5,9%	5,0%	5,4%
164	13,3%	6,4%	12,1%	11,9%	17,6%	8,1%	9,2%	4,6%
165	16,6%	7,8%	12,3%	15,2%	13,0%	9,9%	9,7%	9,2%
166	10,6%	86,7%	74,7%	10,7%	15,6%	5,1%	3,4%	2,3%
167	7,6%	37,7%	28,1%	7,9%	5,4%	0,4%	1,3%	0,9%
168	11,1%	27,6%	14,7%	11,2%	11,2%	23,5%	6,1%	6,1%
169	13,7%	14,4%	19,0%	14,5%	24,0%	11,3%	16,4%	9,0%
16A	25,4%	27,7%	19,7%	26,0%	9,1%	16,4%	15,1%	17,1%
16B	9,8%	21,9%	7,7%	8,9%	3,4%	5,7%	4,8%	5,2%
171	6,6%	4,9%	4,0%	5,0%	5,5%	3,2%	4,4%	3,5%
172	6,3%	8,0%	4,1%	5,1%	2,9%	3,3%	3,3%	3,0%
16C	9,7%	7,2%	24,3%	10,8%	5,7%	8,0%	4,5%	5,0%
185	11,1%	15,3%	9,6%	10,5%	4,7%	6,3%	6,7%	6,6%
181	20,5%	10,2%	12,8%	16,0%	10,2%	15,5%	8,2%	10,4%
182	15,5%	8,2%	5,0%	12,6%	12,8%	8,0%	10,3%	8,6%
183	11,0%	15,0%	19,1%	10,9%	14,1%	9,8%	7,5%	7,7%
184	8,3%	16,1%	23,7%	9,3%	12,0%	5,7%	6,0%	6,5%
150	8,7%	7,7%	7,9%	7,8%	6,4%	6,7%	6,0%	4,8%

NUTSIII	$\widehat{\mu}_i^{dir}$	$\widehat{\mu}_i^{sinQ}$	$\widehat{\mu}_i^{sinR}$	$\widehat{\mu}_i^{com}$	$\widehat{\mu}_i^{FH}$	$\widehat{\mu}_i^{NS}$	$\widehat{\mu}_{it}^{RY}$	$\widehat{\mu}_{it}^{LP}$
	***t*=2**							
111	16,4%	4,8%	14,1%	13,8%	12,1%	11,1%	8,1%	8,0%
112	8,5%	5,2%	23,2%	10,2%	11,4%	11,0%	7,4%	4,7%
113	7,4%	15,2%	7,2%	6,9%	2,3%	2,6%	4,1%	3,2%
114	5,3%	13,1%	3,1%	4,4%	1,9%	2,5%	3,1%	2,1%
115	11,5%	19,4%	5,2%	10,2%	5,3%	7,5%	7,4%	6,3%
116	9,0%	16,4%	5,2%	7,2%	3,2%	4,8%	3,5%	4,1%
117	6,6%	15,6%	8,0%	6,3%	3,6%	4,7%	2,8%	2,4%
118	10,2%	6,8%	25,8%	11,1%	12,3%	9,8%	9,9%	8,3%
161	12,3%	10,0%	11,3%	10,8%	9,0%	7,7%	7,6%	6,9%
162	8,0%	7,1%	8,6%	7,6%	2,5%	3,6%	4,1%	4,1%
163	6,6%	15,7%	13,4%	7,1%	2,7%	2,5%	2,6%	2,4%
164	10,4%	10,7%	4,7%	8,4%	6,8%	6,0%	10,6%	4,7%
165	14,1%	5,7%	2,7%	10,4%	4,8%	6,9%	5,7%	6,1%
166	10,1%	71,6%	58,2%	10,2%	8,4%	4,2%	8,8%	2,0%
167	7,7%	38,2%	29,9%	8,0%	6,2%	0,5%	5,0%	1,0%
168	30,7%	15,1%	6,4%	29,1%	12,1%	19,7%	17,8%	18,2%
169	15,9%	11,2%	7,0%	11,8%	7,5%	15,2%	11,5%	7,3%
16A	21,1%	46,5%	38,9%	24,3%	13,4%	14,6%	9,7%	15,6%
16B	8,5%	27,8%	3,4%	6,0%	3,4%	4,3%	3,3%	4,0%
171	7,6%	6,7%	4,6%	5,4%	6,9%	3,6%	4,7%	3,8%
172	8,9%	8,2%	3,3%	5,7%	4,1%	6,0%	3,9%	4,4%
16C	9,8%	18,7%	11,5%	9,1%	4,1%	3,2%	3,5%	4,1%
185	13,0%	16,3%	11,1%	12,5%	5,3%	6,6%	8,7%	7,8%
181	17,0%	11,2%	11,6%	14,5%	10,2%	14,8%	9,1%	9,9%
182	24,4%	7,4%	6,4%	21,1%	14,0%	8,2%	13,6%	7,8%
183	10,8%	13,8%	16,9%	12,7%	12,1%	7,6%	11,6%	7,9%
184	11,7%	7,3%	10,5%	10,2%	3,5%	10,1%	5,7%	5,9%
150	8,9%	8,0%	3,8%	5,7%	2,8%	3,8%	3,4%	3,9%
NUTSIII	***t*=3**							
111	17,6%	4,5%	12,2%	14,5%	11,8%	11,1%	8,5%	8,5%
112	11,9%	18,6%	3,0%	8,7%	7,0%	8,8%	7,0%	5,5%
113	9,0%	11,8%	7,0%	7,7%	2,2%	2,9%	3,1%	3,4%
114	5,0%	9,5%	4,6%	4,5%	2,3%	1,7%	1,9%	1,9%
115	12,3%	18,8%	3,9%	11,1%	5,8%	6,7%	7,6%	7,5%
116	6,8%	15,1%	3,8%	5,8%	2,9%	4,3%	2,2%	2,4%
117	6,9%	22,5%	3,4%	5,7%	2,2%	1,6%	2,2%	1,9%
118	6,6%	4,2%	20,1%	7,1%	6,8%	5,4%	3,0%	2,8%
161	10,4%	6,7%	5,3%	7,9%	4,6%	5,0%	5,5%	5,6%
162	10,2%	9,3%	13,8%	11,1%	4,5%	5,4%	6,1%	5,7%
163	8,0%	10,0%	9,6%	8,1%	2,8%	3,5%	3,8%	3,9%
164	10,1%	6,7%	8,6%	9,8%	12,9%	6,3%	6,0%	4,9%
165	10,4%	10,2%	7,7%	9,4%	4,8%	10,4%	4,2%	4,4%
166	35,6%	12,4%	16,8%	35,3%	35,9%	39,1%	26,6%	26,7%
167	7,9%	29,3%	24,1%	8,4%	2,4%	1,9%	1,4%	0,7%
168	5,4%	7,8%	10,2%	5,6%	6,5%	5,1%	1,2%	0,7%
169	22,9%	4,6%	3,9%	18,0%	13,9%	8,8%	11,1%	10,6%
16A	23,5%	30,4%	28,6%	21,6%	6,6%	14,2%	9,9%	11,3%
16B	10,2%	30,8%	3,4%	7,0%	4,6%	4,6%	5,0%	5,1%
171	6,2%	4,8%	5,5%	5,4%	5,9%	3,2%	4,1%	3,4%
172	6,9%	10,3%	9,2%	7,0%	4,6%	3,9%	4,2%	3,7%
16C	9,6%	21,5%	10,9%	9,4%	4,9%	3,9%	4,2%	4,3%
185	16,5%	23,7%	22,8%	17,2%	12,8%	9,5%	10,4%	10,9%
181	29,3%	33,9%	40,5%	29,9%	37,3%	32,5%	28,6%	29,5%
182	15,0%	48,0%	14,2%	15,4%	5,3%	12,7%	8,7%	8,8%
183	12,2%	23,0%	12,8%	11,6%	8,3%	5,4%	5,2%	5,0%
184	9,1%	21,0%	21,7%	11,1%	11,7%	6,1%	4,8%	5,7%
150	12,2%	11,7%	6,3%	8,4%	7,8%	9,6%	6,6%	5,8%

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=4**							
111	17,0%	13,5%	7,2%	11,6%	5,9%	6,8%	7,8%	7,2%
112	10,0%	22,8%	3,2%	7,5%	7,4%	8,3%	4,5%	4,4%
113	9,6%	23,9%	2,3%	7,2%	6,8%	5,6%	4,2%	4,4%
114	5,4%	9,7%	5,9%	4,9%	3,2%	2,2%	2,4%	2,3%
115	10,8%	26,6%	2,8%	9,3%	6,7%	5,6%	6,8%	6,1%
116	9,7%	30,7%	6,7%	8,0%	7,3%	5,3%	5,7%	5,2%
117	5,5%	15,2%	10,4%	5,8%	3,4%	4,8%	0,3%	0,3%
118	6,6%	10,7%	16,0%	7,2%	2,8%	3,9%	2,8%	2,8%
161	10,8%	8,4%	10,7%	10,1%	7,6%	6,5%	6,3%	6,1%
162	7,9%	5,9%	7,9%	7,1%	2,8%	3,9%	4,4%	4,0%
163	9,8%	16,9%	12,2%	9,9%	4,4%	4,3%	5,3%	5,1%
164	9,1%	5,7%	12,1%	9,3%	12,6%	6,1%	5,8%	3,6%
165	14,8%	4,6%	5,8%	10,2%	6,5%	7,1%	7,4%	7,2%
166	10,2%	81,0%	64,2%	10,2%	13,7%	6,4%	2,8%	1,8%
167	7,9%	29,2%	14,5%	8,2%	1,2%	3,0%	1,0%	0,4%
168	13,4%	4,6%	9,2%	13,2%	7,2%	8,2%	10,3%	9,7%
169	16,0%	4,5%	5,2%	11,6%	12,8%	8,8%	14,2%	7,6%
16A	23,4%	38,8%	30,4%	23,0%	8,7%	13,1%	12,0%	12,8%
16B	10,2%	26,2%	4,8%	7,4%	2,8%	4,2%	5,1%	4,9%
171	6,7%	5,3%	5,3%	5,5%	6,1%	3,3%	4,3%	4,0%
172	7,9%	10,4%	7,0%	7,0%	3,4%	3,6%	5,8%	4,4%
16C	11,3%	22,0%	3,8%	9,9%	8,5%	7,3%	6,7%	6,6%
185	13,8%	34,7%	15,0%	13,2%	7,7%	6,0%	8,0%	6,9%
181	33,3%	31,9%	42,5%	32,4%	38,1%	33,4%	30,2%	31,1%
182	24,4%	66,6%	12,8%	22,1%	6,1%	10,9%	9,4%	11,9%
183	9,1%	30,6%	19,6%	11,2%	13,9%	10,0%	6,5%	6,3%
184	8,8%	40,5%	11,2%	9,0%	3,2%	7,3%	3,8%	4,6%
150	7,8%	6,7%	10,2%	7,9%	5,4%	4,5%	5,5%	4,6%
NUTSIII	***t*=5**							
111	11,4%	23,6%	2,5%	7,8%	3,8%	6,1%	4,8%	5,0%
112	16,4%	5,4%	21,1%	16,2%	9,3%	10,2%	8,6%	9,8%
113	9,1%	21,0%	7,3%	8,3%	4,6%	3,7%	4,1%	4,0%
114	5,4%	11,2%	5,9%	5,0%	2,8%	2,2%	2,4%	2,4%
115	7,8%	27,0%	5,2%	7,0%	4,4%	3,7%	3,5%	3,3%
116	8,5%	29,2%	2,6%	6,0%	5,2%	3,4%	3,4%	3,6%
117	6,2%	22,7%	9,3%	6,3%	1,0%	1,7%	0,4%	0,3%
118	6,4%	17,6%	14,5%	6,7%	4,3%	5,9%	2,2%	2,2%
161	9,8%	6,4%	4,0%	6,6%	6,4%	8,2%	4,0%	4,2%
162	8,6%	11,0%	16,8%	10,2%	7,5%	6,7%	5,6%	5,3%
163	9,6%	7,0%	3,8%	7,1%	3,7%	5,6%	4,7%	4,7%
164	18,9%	11,2%	2,8%	14,8%	5,5%	7,8%	15,0%	8,1%
165	14,8%	5,2%	3,6%	12,1%	4,4%	7,2%	6,4%	6,3%
166	10,8%	59,5%	42,4%	10,6%	3,2%	4,3%	3,3%	3,5%
167	7,9%	38,1%	26,9%	8,3%	8,6%	2,0%	1,5%	0,8%
168	11,2%	7,7%	15,0%	11,9%	7,5%	6,7%	9,0%	7,7%
169	18,9%	7,1%	9,0%	15,6%	14,0%	8,1%	12,1%	9,0%
16A	16,1%	56,2%	47,4%	15,0%	21,2%	16,4%	7,8%	7,8%
16B	10,3%	26,4%	3,6%	7,3%	3,4%	4,4%	4,3%	4,8%
171	5,6%	5,4%	9,1%	6,4%	10,6%	5,4%	4,5%	3,5%
172	9,8%	15,0%	7,2%	8,5%	4,0%	3,9%	6,5%	5,6%
16C	12,1%	18,2%	7,1%	10,6%	5,5%	6,1%	5,8%	6,2%
185	11,6%	22,9%	10,7%	10,9%	7,0%	6,0%	6,3%	5,4%
181	28,8%	26,2%	31,2%	27,9%	26,6%	23,5%	24,7%	25,1%
182	6,7%	30,3%	9,5%	6,8%	9,9%	5,4%	3,5%	2,6%
183	13,3%	39,2%	4,0%	10,6%	11,0%	18,8%	9,4%	7,0%
184	8,9%	22,5%	20,7%	11,0%	8,0%	5,0%	7,4%	6,4%
150	8,8%	8,1%	7,0%	7,1%	3,2%	3,7%	6,3%	4,5%

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=6**							
111	12,5%	18,0%	4,6%	7,4%	2,8%	5,2%	4,5%	4,3%
112	11,9%	5,4%	18,2%	12,7%	7,8%	8,9%	4,6%	5,5%
113	9,6%	16,4%	8,4%	8,5%	2,8%	2,8%	4,4%	4,6%
114	5,1%	14,2%	4,5%	4,6%	1,9%	2,1%	2,0%	2,0%
115	7,9%	24,0%	4,8%	7,1%	3,4%	3,6%	2,6%	2,7%
116	8,0%	27,6%	3,1%	6,6%	4,9%	3,1%	3,4%	3,4%
117	6,5%	31,6%	3,7%	6,0%	3,9%	2,7%	1,3%	0,7%
118	9,7%	3,5%	23,9%	9,3%	6,8%	5,9%	4,8%	4,5%
161	9,9%	4,8%	7,0%	8,2%	3,1%	4,5%	4,2%	4,7%
162	9,2%	3,9%	2,8%	6,0%	3,9%	5,5%	4,2%	4,4%
163	9,7%	20,0%	17,4%	11,5%	9,1%	5,1%	6,2%	5,8%
164	14,0%	22,8%	13,7%	13,4%	5,3%	13,9%	11,2%	5,5%
165	17,3%	5,1%	2,7%	12,5%	4,7%	7,6%	8,8%	8,3%
166	10,4%	87,8%	74,3%	10,4%	16,5%	6,6%	3,7%	2,1%
167	7,9%	48,7%	39,9%	8,0%	14,1%	5,2%	2,6%	1,1%
168	7,1%	13,0%	20,7%	9,1%	11,8%	8,2%	10,4%	5,4%
169	17,2%	9,0%	14,2%	15,7%	17,8%	8,2%	16,9%	10,4%
16A	22,6%	62,5%	52,4%	32,1%	27,3%	21,9%	17,7%	23,4%
16B	7,2%	36,2%	9,2%	7,6%	7,7%	7,6%	4,3%	3,8%
171	5,8%	4,9%	4,9%	5,0%	4,5%	3,5%	3,3%	3,3%
172	8,3%	9,1%	4,7%	6,4%	4,7%	3,8%	4,6%	4,3%
16C	8,9%	15,3%	19,3%	10,7%	5,5%	4,3%	3,2%	4,7%
185	9,8%	21,7%	6,2%	8,7%	4,3%	5,3%	5,3%	4,8%
181	16,8%	9,4%	4,2%	13,3%	12,9%	23,1%	9,6%	8,1%
182	13,6%	8,5%	5,5%	12,1%	8,1%	5,5%	8,7%	5,0%
183	10,5%	8,1%	13,9%	10,3%	6,6%	5,7%	4,9%	5,0%
184	8,8%	11,5%	18,2%	8,9%	6,7%	5,3%	4,4%	4,4%
150	11,6%	11,0%	9,9%	9,7%	7,9%	10,6%	5,7%	5,8%
NUTSIII	***t*=7**							
111	16,1%	26,6%	3,6%	10,8%	7,7%	7,8%	6,3%	6,5%
112	12,8%	36,2%	9,2%	11,2%	18,1%	16,0%	7,6%	5,8%
113	8,2%	18,3%	7,2%	7,7%	4,0%	3,1%	3,8%	4,0%
114	8,9%	11,5%	8,3%	8,4%	3,2%	3,2%	5,8%	5,3%
115	10,6%	34,9%	6,8%	9,8%	12,4%	6,9%	8,1%	5,9%
116	8,9%	32,1%	6,3%	7,7%	8,6%	5,3%	4,1%	4,1%
117	6,4%	27,4%	2,8%	5,4%	3,3%	1,5%	0,7%	0,1%
118	5,1%	8,3%	20,4%	5,2%	1,2%	1,0%	0,6%	0,4%
161	10,7%	4,4%	3,3%	6,7%	3,6%	5,0%	4,8%	5,2%
162	7,2%	3,8%	4,4%	5,4%	3,0%	3,5%	3,8%	3,6%
163	8,5%	11,4%	10,2%	8,6%	5,1%	3,5%	3,8%	4,1%
164	10,9%	9,5%	5,1%	9,2%	4,7%	5,5%	5,5%	3,9%
165	9,2%	12,7%	14,4%	10,4%	9,9%	9,8%	6,1%	5,2%
166	10,2%	93,4%	87,5%	10,3%	26,1%	11,1%	3,3%	2,4%
167	7,6%	28,7%	20,8%	8,0%	5,0%	0,8%	1,3%	0,6%
168	6,4%	7,8%	11,6%	6,6%	5,1%	4,3%	2,5%	2,6%
169	14,2%	9,3%	11,9%	13,4%	14,8%	7,6%	10,9%	6,9%
16A	12,1%	16,6%	13,9%	10,9%	4,1%	15,7%	3,9%	4,0%
16B	8,6%	30,5%	5,9%	7,3%	6,0%	5,6%	4,0%	4,5%
171	4,9%	3,7%	4,5%	4,5%	3,0%	3,2%	2,5%	2,6%
172	7,6%	5,8%	5,5%	6,6%	6,4%	3,7%	4,6%	4,1%
16C	8,7%	26,2%	10,4%	9,2%	2,7%	5,8%	4,6%	4,7%
185	8,9%	17,1%	9,6%	8,4%	6,6%	4,6%	3,3%	3,2%
181	17,3%	17,9%	24,1%	18,0%	17,3%	17,4%	14,1%	15,8%
182	16,6%	22,5%	4,3%	13,0%	7,4%	5,9%	12,2%	7,5%
183	10,2%	19,3%	5,1%	8,1%	3,6%	6,8%	5,6%	4,9%
184	9,9%	19,7%	30,0%	10,6%	17,6%	13,1%	10,8%	11,3%
150	9,1%	8,5%	3,8%	6,0%	3,3%	5,5%	6,2%	3,9%

A15.9 Coeficiente de variação

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=1**							
111	16,0%	6,8%	2,0%	10,0%	3,4%	6,1%	6,5%	6,4%
112	9,9%	6,6%	2,6%	9,5%	3,6%	5,4%	4,4%	4,7%
113	6,6%	6,7%	2,2%	6,5%	1,6%	1,9%	1,8%	1,8%
114	6,4%	6,3%	2,1%	6,5%	3,2%	2,6%	3,3%	3,3%
115	11,7%	8,4%	2,8%	11,5%	7,1%	6,5%	6,5%	6,0%
116	10,1%	6,8%	1,9%	7,9%	2,8%	3,5%	4,2%	4,2%
117	14,6%	6,5%	2,6%	11,5%	2,8%	3,0%	5,7%	6,8%
118	11,2%	6,4%	2,9%	11,0%	5,7%	6,9%	7,4%	7,5%
161	8,5%	4,0%	1,7%	7,1%	3,5%	4,6%	3,9%	4,1%
162	10,6%	4,3%	2,6%	7,3%	2,8%	3,8%	5,2%	5,3%
163	10,6%	4,6%	2,0%	8,5%	2,5%	3,2%	5,0%	5,4%
164	13,3%	4,1%	2,5%	11,2%	6,9%	4,0%	5,3%	4,2%
165	16,6%	4,0%	2,0%	14,5%	5,0%	7,0%	8,2%	8,4%
166	10,6%	8,0%	4,4%	10,7%	6,6%	2,9%	0,9%	0,4%
167	7,6%	5,9%	3,4%	7,8%	2,7%	0,3%	0,5%	0,2%
168	11,1%	5,4%	5,1%	11,0%	7,6%	13,4%	5,9%	6,0%
169	12,8%	3,7%	1,9%	12,1%	8,1%	7,3%	10,3%	7,1%
16A	23,6%	5,4%	2,9%	25,7%	8,6%	13,3%	14,4%	17,0%
16B	9,6%	3,4%	2,6%	8,9%	2,5%	4,5%	4,8%	5,2%
171	6,6%	4,8%	3,6%	4,9%	3,4%	3,1%	3,5%	3,5%
172	6,2%	5,2%	3,0%	5,1%	2,7%	3,2%	3,1%	3,0%
16C	9,7%	4,0%	2,4%	10,4%	3,2%	4,4%	4,3%	4,6%
185	11,1%	6,1%	2,3%	10,4%	4,1%	5,7%	6,3%	6,4%
181	20,1%	6,5%	2,0%	14,1%	4,1%	15,4%	5,7%	5,2%
182	15,2%	7,4%	1,8%	11,9%	4,5%	6,4%	7,0%	7,9%
183	10,9%	6,1%	1,7%	10,3%	3,5%	4,5%	4,0%	4,0%
184	8,0%	6,0%	1,8%	8,4%	4,8%	5,2%	4,9%	5,0%
150	8,6%	7,6%	3,0%	6,8%	3,0%	4,2%	4,7%	4,4%

NUTSIII	$\widehat{\mu}_i^{dir}$	$\widehat{\mu}_i^{sinQ}$	$\widehat{\mu}_i^{sinR}$	$\widehat{\mu}_i^{com}$	$\widehat{\mu}_i^{FH}$	$\widehat{\mu}_i^{NS}$	$\widehat{\mu}_{it}^{RY}$	$\widehat{\mu}_{it}^{LP}$
	t=2							
111	16,4%	3,1%	2,1%	11,0%	3,1%	5,2%	4,5%	4,5%
112	8,3%	2,8%	2,8%	9,6%	5,3%	6,1%	5,6%	4,0%
113	7,4%	3,4%	2,9%	6,8%	2,3%	2,5%	3,6%	3,2%
114	5,3%	3,3%	2,6%	4,4%	1,9%	2,0%	2,7%	2,1%
115	11,5%	3,5%	3,4%	10,2%	5,2%	6,0%	6,0%	6,3%
116	9,0%	3,4%	2,6%	6,8%	2,7%	3,4%	3,5%	4,0%
117	6,6%	3,4%	3,4%	6,0%	1,9%	2,4%	2,7%	2,3%
118	10,0%	2,8%	3,2%	10,4%	5,2%	5,2%	6,3%	6,4%
161	11,9%	3,6%	1,9%	9,2%	3,8%	4,9%	4,6%	5,8%
162	8,0%	4,2%	3,1%	7,0%	2,5%	3,6%	3,4%	3,8%
163	6,6%	4,6%	2,5%	7,0%	2,1%	2,5%	2,3%	2,4%
164	10,4%	4,4%	3,4%	8,3%	5,4%	4,8%	5,8%	4,7%
165	14,0%	4,2%	2,7%	10,4%	4,4%	6,7%	5,5%	6,1%
166	10,1%	6,8%	6,8%	10,2%	4,0%	4,0%	5,5%	0,4%
167	7,7%	5,5%	4,4%	8,0%	3,1%	0,1%	3,4%	0,2%
168	30,6%	4,6%	5,8%	29,1%	11,7%	14,7%	15,5%	18,2%
169	15,9%	4,4%	2,6%	10,0%	7,2%	8,2%	8,4%	5,9%
16A	20,6%	5,8%	4,1%	24,0%	7,8%	14,2%	8,7%	11,7%
16B	8,5%	2,9%	2,6%	5,9%	2,6%	4,3%	3,1%	4,0%
171	7,5%	6,7%	4,3%	5,2%	3,6%	3,1%	2,9%	3,7%
172	8,9%	7,0%	3,3%	5,7%	2,7%	3,3%	3,9%	4,3%
16C	9,8%	3,3%	2,5%	8,4%	2,9%	2,8%	3,1%	3,8%
185	12,9%	6,1%	2,5%	12,4%	4,9%	6,0%	7,2%	7,4%
181	17,0%	6,5%	2,5%	13,4%	4,7%	14,8%	4,9%	5,4%
182	23,2%	7,3%	2,2%	20,1%	4,7%	6,4%	7,4%	7,7%
183	10,8%	6,3%	1,8%	11,5%	4,2%	5,1%	6,5%	5,5%
184	11,4%	7,0%	2,3%	8,8%	3,5%	6,2%	5,2%	5,1%
150	8,9%	7,9%	3,3%	5,6%	2,7%	3,6%	3,2%	3,9%
NUTSIII	*t*=3							
111	17,5%	3,5%	2,1%	11,1%	3,0%	5,2%	6,2%	6,3%
112	11,9%	4,1%	2,8%	8,5%	4,4%	6,2%	6,4%	5,5%
113	9,0%	3,8%	2,4%	7,6%	2,0%	2,5%	3,1%	3,4%
114	5,0%	3,8%	3,4%	4,5%	1,8%	1,7%	1,9%	1,9%
115	11,2%	4,1%	3,2%	10,4%	5,7%	6,5%	6,9%	7,0%
116	6,8%	4,0%	2,4%	5,8%	2,3%	3,1%	2,2%	2,3%
117	6,9%	4,2%	3,2%	5,7%	1,6%	1,4%	2,1%	1,9%
118	6,5%	3,4%	2,8%	6,9%	4,4%	4,0%	2,8%	2,7%
161	10,3%	4,0%	2,3%	7,7%	3,6%	5,0%	5,4%	5,6%
162	10,1%	4,5%	4,3%	9,8%	2,5%	3,7%	5,1%	5,1%
163	7,9%	4,6%	2,6%	7,9%	2,6%	3,2%	3,8%	3,9%
164	9,8%	4,0%	2,7%	9,1%	7,3%	4,5%	5,2%	4,8%
165	10,3%	4,6%	2,7%	8,8%	3,9%	7,5%	4,1%	4,3%
166	25,2%	3,7%	3,4%	24,6%	12,7%	13,4%	21,2%	21,9%
167	7,9%	5,4%	4,2%	8,3%	1,5%	1,0%	0,8%	0,4%
168	5,4%	3,9%	2,8%	5,6%	5,2%	4,5%	1,0%	0,6%
169	21,0%	4,1%	2,3%	16,5%	6,9%	8,2%	10,0%	10,2%
16A	19,7%	5,4%	3,1%	18,8%	6,5%	12,6%	9,9%	10,8%
16B	10,0%	2,9%	3,4%	6,9%	2,7%	4,3%	4,7%	5,0%
171	6,2%	4,5%	5,4%	5,3%	3,7%	3,2%	3,6%	3,4%
172	6,9%	5,0%	4,7%	6,7%	3,1%	3,6%	3,7%	3,6%
16C	9,6%	3,3%	2,6%	9,2%	3,7%	3,4%	4,2%	4,3%
185	16,0%	22,8%	3,5%	15,3%	5,8%	7,0%	8,9%	9,2%
181	25,1%	14,9%	2,1%	24,5%	5,8%	11,4%	11,3%	12,3%
182	14,9%	29,6%	2,8%	15,1%	4,8%	7,6%	8,3%	8,4%
183	12,2%	22,5%	2,3%	10,6%	3,8%	5,3%	4,4%	4,0%
184	9,1%	20,7%	1,9%	10,2%	4,7%	4,7%	4,7%	4,8%
150	12,2%	11,7%	4,4%	7,7%	2,6%	3,2%	6,5%	5,1%

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
				***t*=4**				
111	16,9%	4,8%	2,2%	8,8%	3,0%	5,3%	5,7%	6,1%
112	10,0%	5,2%	2,3%	7,5%	4,0%	5,8%	4,3%	4,4%
113	9,6%	5,2%	2,3%	7,1%	2,9%	3,0%	4,1%	4,4%
114	5,4%	4,6%	3,4%	4,6%	2,1%	1,9%	2,2%	2,3%
115	10,8%	5,4%	2,6%	9,1%	6,1%	5,6%	6,7%	6,1%
116	9,6%	5,5%	2,4%	7,8%	3,7%	3,5%	5,3%	5,1%
117	5,5%	4,9%	2,4%	5,7%	1,8%	2,6%	0,3%	0,1%
118	6,6%	4,7%	2,7%	7,1%	2,8%	3,3%	2,8%	2,8%
161	10,6%	4,1%	2,2%	8,5%	3,0%	4,6%	5,3%	5,4%
162	7,9%	4,6%	4,2%	6,8%	2,3%	3,3%	4,1%	3,9%
163	9,7%	5,1%	2,7%	9,4%	3,1%	3,7%	5,0%	5,0%
164	9,0%	4,3%	2,2%	8,7%	7,7%	4,4%	5,1%	3,4%
165	14,8%	4,5%	2,2%	8,9%	3,8%	7,0%	7,4%	6,6%
166	10,1%	8,0%	4,5%	10,2%	5,3%	3,6%	1,8%	0,9%
167	7,9%	5,7%	3,8%	8,1%	1,2%	1,4%	0,7%	0,2%
168	13,4%	4,5%	2,6%	13,0%	7,0%	7,8%	9,8%	9,6%
169	16,0%	4,5%	2,2%	11,6%	7,5%	7,9%	11,0%	7,6%
16A	21,0%	6,1%	3,0%	22,4%	6,9%	12,7%	12,0%	12,7%
16B	10,1%	3,3%	3,6%	7,3%	2,7%	4,2%	4,9%	4,9%
171	6,7%	5,3%	5,3%	5,5%	3,9%	3,3%	3,9%	4,0%
172	7,9%	5,8%	4,5%	6,8%	3,2%	3,5%	5,1%	4,3%
16C	11,3%	3,5%	2,4%	9,9%	4,8%	4,6%	6,3%	6,6%
185	13,8%	31,2%	3,4%	12,1%	4,2%	5,9%	6,2%	6,0%
181	27,5%	20,5%	2,0%	25,9%	6,5%	11,8%	13,1%	14,6%
182	24,1%	41,3%	3,0%	20,5%	4,5%	6,3%	8,9%	9,5%
183	8,9%	29,4%	2,3%	10,1%	4,6%	5,4%	5,4%	5,0%
184	8,8%	33,4%	2,1%	8,1%	3,2%	5,0%	3,8%	4,4%
150	7,8%	6,7%	4,8%	7,2%	3,1%	3,7%	4,7%	4,3%
NUTSIII				***t*=5**				
111	11,3%	5,1%	2,1%	7,7%	2,6%	5,2%	4,7%	5,0%
112	16,3%	4,3%	1,8%	14,2%	2,7%	4,7%	5,2%	5,3%
113	8,9%	5,0%	2,0%	8,3%	2,8%	2,9%	4,0%	4,0%
114	5,3%	4,6%	3,7%	4,7%	2,0%	1,9%	2,3%	2,3%
115	7,6%	5,2%	2,5%	6,6%	3,9%	3,5%	3,5%	3,2%
116	8,4%	5,3%	2,1%	6,0%	3,0%	3,0%	3,4%	3,6%
117	6,2%	5,0%	2,6%	6,2%	1,0%	1,2%	0,4%	0,1%
118	6,4%	4,8%	2,8%	6,6%	3,1%	4,1%	2,2%	2,2%
161	9,7%	5,2%	2,6%	6,6%	2,7%	4,8%	4,0%	4,2%
162	8,6%	5,5%	4,9%	9,7%	3,1%	3,9%	4,7%	4,7%
163	9,5%	5,2%	2,4%	7,1%	2,6%	3,3%	4,3%	4,7%
164	18,3%	5,5%	2,3%	14,3%	5,1%	5,0%	9,8%	8,0%
165	14,8%	5,1%	2,0%	12,0%	4,3%	7,0%	5,8%	6,2%
166	10,1%	8,0%	4,4%	10,1%	3,1%	3,6%	3,0%	3,1%
167	7,9%	6,9%	3,0%	8,2%	4,6%	0,9%	1,0%	0,2%
168	10,9%	4,7%	2,3%	11,9%	6,6%	6,7%	8,7%	7,7%
169	18,9%	4,7%	1,9%	15,4%	7,2%	8,0%	9,2%	8,5%
16A	15,3%	7,8%	3,0%	14,9%	8,4%	13,3%	5,8%	5,8%
16B	10,2%	3,7%	3,6%	7,3%	3,1%	4,3%	4,3%	4,8%
171	5,6%	4,8%	4,9%	5,9%	4,4%	3,3%	3,6%	3,3%
172	9,8%	5,6%	4,8%	8,5%	3,5%	3,9%	5,9%	5,4%
16C	11,9%	4,1%	2,7%	10,6%	3,4%	3,7%	5,6%	6,2%
185	11,4%	22,7%	2,8%	10,6%	4,2%	5,8%	5,2%	5,2%
181	24,4%	18,0%	2,7%	22,8%	7,6%	13,3%	12,9%	14,1%
182	6,7%	25,8%	1,9%	6,6%	4,8%	4,4%	2,7%	2,5%
183	13,3%	28,3%	2,7%	10,6%	5,6%	8,8%	8,3%	7,0%
184	8,9%	22,5%	1,7%	10,0%	3,5%	4,8%	5,7%	4,8%
150	8,8%	8,1%	5,1%	6,8%	2,9%	3,5%	5,1%	4,4%

NUTSIII	$\widehat{\mu}_i^{dir}$	$\widehat{\mu}_i^{sinQ}$	$\widehat{\mu}_i^{sinR}$	$\widehat{\mu}_i^{com}$	$\widehat{\mu}_i^{FH}$	$\widehat{\mu}_i^{NS}$	$\widehat{\mu}_{it}^{RY}$	$\widehat{\mu}_{it}^{LP}$
				***t*=6**				
111	12,5%	4,0%	1,8%	6,5%	2,7%	4,9%	4,5%	4,0%
112	11,9%	3,5%	2,1%	12,2%	2,8%	4,8%	4,3%	4,3%
113	9,6%	3,9%	2,1%	8,2%	2,3%	2,8%	3,7%	4,6%
114	5,1%	3,9%	2,7%	4,5%	1,8%	1,9%	2,0%	2,0%
115	7,9%	4,2%	2,8%	6,9%	3,2%	3,4%	2,6%	2,7%
116	7,9%	4,3%	2,0%	6,6%	3,5%	3,1%	3,3%	3,4%
117	6,5%	4,5%	3,1%	5,9%	3,0%	2,1%	1,2%	0,7%
118	9,2%	3,4%	2,6%	9,0%	3,6%	3,5%	4,4%	4,5%
161	9,8%	3,8%	1,8%	7,4%	2,6%	4,5%	4,0%	4,4%
162	9,2%	3,9%	2,8%	6,0%	2,5%	3,5%	3,9%	4,4%
163	9,7%	4,6%	2,5%	10,9%	3,4%	3,6%	5,2%	5,3%
164	13,9%	4,7%	3,0%	13,1%	4,2%	6,5%	8,0%	5,5%
165	17,2%	4,0%	2,0%	12,1%	4,7%	7,2%	6,6%	8,1%
166	10,3%	7,3%	4,4%	10,4%	8,3%	3,7%	1,8%	0,3%
167	7,9%	5,7%	3,0%	8,0%	7,1%	2,3%	1,2%	0,2%
168	7,1%	3,4%	2,7%	8,8%	6,7%	6,0%	7,8%	4,9%
169	17,1%	3,6%	2,0%	14,5%	7,1%	8,0%	10,1%	8,7%
16A	22,4%	6,3%	3,4%	30,5%	10,7%	15,9%	14,4%	18,2%
16B	7,1%	2,5%	2,3%	6,8%	3,5%	4,4%	3,5%	3,6%
171	5,8%	4,9%	4,4%	4,8%	3,6%	3,3%	3,0%	3,3%
172	8,3%	5,3%	3,5%	6,3%	3,4%	3,7%	4,2%	4,2%
16C	8,6%	3,3%	2,5%	9,7%	3,2%	3,2%	3,2%	4,2%
185	9,8%	4,7%	2,1%	8,7%	3,6%	4,9%	4,8%	4,7%
181	16,8%	6,3%	3,3%	13,3%	8,0%	16,9%	8,4%	8,1%
182	13,6%	6,3%	1,9%	11,5%	4,3%	5,5%	4,5%	4,7%
183	10,5%	5,6%	1,7%	9,3%	3,1%	5,7%	3,6%	3,7%
184	8,8%	5,3%	1,6%	8,6%	3,5%	5,1%	3,3%	3,1%
150	11,6%	11,0%	3,1%	7,9%	2,8%	3,3%	3,7%	3,9%
NUTSIII				***t*=7**				
111	16,1%	9,7%	2,0%	10,6%	4,1%	6,2%	5,9%	6,5%
112	12,7%	10,5%	2,1%	10,6%	7,0%	7,8%	6,5%	5,7%
113	8,2%	9,0%	1,9%	7,5%	2,7%	2,9%	3,8%	4,0%
114	8,9%	8,4%	2,4%	8,1%	2,6%	2,5%	4,7%	5,0%
115	10,5%	10,4%	2,3%	9,3%	6,2%	5,4%	6,9%	5,8%
116	8,6%	10,2%	2,0%	7,7%	4,6%	3,8%	4,0%	4,0%
117	6,4%	9,8%	2,5%	5,4%	2,0%	1,0%	0,2%	0,0%
118	5,1%	8,0%	2,3%	5,2%	0,9%	1,0%	0,5%	0,3%
161	10,6%	3,6%	2,0%	6,5%	2,8%	4,7%	4,8%	5,2%
162	7,2%	3,7%	2,7%	5,3%	2,6%	3,5%	3,7%	3,6%
163	8,5%	4,0%	2,2%	8,3%	3,2%	3,4%	3,8%	4,0%
164	10,8%	4,0%	2,2%	9,2%	3,7%	4,2%	4,7%	3,9%
165	9,0%	3,2%	1,6%	9,4%	5,4%	6,7%	5,2%	4,8%
166	10,2%	7,1%	3,7%	10,2%	12,1%	5,3%	0,7%	0,4%
167	7,5%	4,7%	3,3%	8,0%	2,9%	0,3%	0,3%	0,1%
168	6,3%	3,4%	2,1%	6,6%	4,0%	4,2%	2,5%	2,6%
169	13,8%	3,3%	1,8%	13,0%	5,8%	7,6%	6,4%	6,3%
16A	11,9%	4,2%	2,2%	9,6%	4,1%	11,4%	3,9%	3,3%
16B	8,5%	2,5%	2,9%	7,1%	2,4%	4,1%	4,0%	4,4%
171	4,9%	3,6%	4,3%	4,4%	3,0%	3,1%	2,5%	2,6%
172	7,5%	3,8%	3,5%	6,5%	3,4%	3,7%	4,1%	4,0%
16C	8,7%	2,7%	2,6%	8,7%	2,7%	3,8%	4,2%	4,5%
185	8,7%	12,6%	2,5%	8,4%	3,1%	4,6%	3,3%	3,2%
181	15,7%	12,4%	2,2%	15,9%	7,8%	12,8%	10,2%	11,0%
182	16,6%	16,4%	1,8%	13,0%	4,9%	5,9%	9,0%	7,4%
183	10,2%	15,8%	2,3%	8,0%	2,9%	5,1%	4,8%	4,8%
184	9,9%	12,0%	1,3%	10,4%	5,0%	5,7%	7,8%	7,8%
150	9,1%	8,4%	3,4%	5,8%	3,2%	3,7%	5,1%	3,9%

A15.10 Eficiência relativa

NUTSIII	$\widehat{\mu}_i^{dir}$	$\widehat{\mu}_i^{sinQ}$	$\widehat{\mu}_i^{sinR}$	$\widehat{\mu}_i^{com}$	$\widehat{\mu}_i^{FH}$	$\widehat{\mu}_i^{NS}$	$\widehat{\mu}_{it}^{RY}$	$\widehat{\mu}_{it}^{LP}$
	***t*=1**							
111	1,00	1,02	1,93	1,28	2,45	2,51	2,23	2,18
112	1,00	0,76	0,78	0,97	2,55	1,83	2,24	2,02
113	1,00	0,43	0,68	1,01	3,32	2,85	3,69	3,67
114	1,00	0,75	0,62	0,89	0,98	1,38	1,73	1,78
115	1,00	0,28	0,91	1,01	0,92	1,32	1,52	1,90
116	1,00	0,63	1,29	1,17	1,87	1,33	2,40	2,29
117	1,00	1,34	1,03	1,11	1,35	1,31	1,59	1,68
118	1,00	1,24	0,68	0,95	1,62	1,70	1,37	1,37
161	1,00	1,15	0,87	1,05	1,27	1,52	2,18	1,91
162	1,00	2,47	4,09	1,46	1,86	1,67	1,91	1,99
163	1,00	1,22	2,73	1,25	2,01	1,80	2,12	1,97
164	1,00	2,07	1,10	1,12	0,75	1,64	1,44	2,86
165	1,00	2,12	1,34	1,09	1,28	1,68	1,71	1,80
166	1,00	0,12	0,14	0,99	0,68	2,09	3,16	4,54
167	1,00	0,20	0,27	0,96	1,41	16,94	5,86	8,43
168	1,00	0,40	0,76	1,00	0,99	0,47	1,84	1,83
169	1,00	0,95	0,72	0,94	0,57	1,21	0,83	1,52
16A	1,00	0,92	1,29	0,98	2,79	1,55	1,68	1,49
16B	1,00	0,45	1,27	1,10	2,89	1,72	2,04	1,89
171	1,00	1,37	1,65	1,34	1,21	2,07	1,51	1,88
172	1,00	0,78	1,51	1,23	2,18	1,91	1,93	2,08
16C	1,00	1,35	0,40	0,90	1,70	1,22	2,18	1,95
185	1,00	0,72	1,16	1,06	2,38	1,77	1,67	1,67
181	1,00	2,01	1,60	1,28	2,01	1,33	2,50	1,97
182	1,00	1,90	3,09	1,23	1,21	1,95	1,51	1,79
183	1,00	0,73	0,57	1,00	0,78	1,12	1,47	1,42
184	1,00	0,51	0,35	0,89	0,69	1,45	1,38	1,27
150	1,00	1,13	1,09	1,12	1,36	1,29	1,44	1,81

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=2**							
111	1,00	3,38	1,16	1,18	1,35	1,48	2,02	2,05
112	1,00	1,63	0,36	0,83	0,75	0,77	1,15	1,81
113	1,00	0,49	1,02	1,06	3,18	2,86	1,79	2,29
114	1,00	0,41	1,70	1,22	2,82	2,15	1,72	2,49
115	1,00	0,59	2,22	1,13	2,18	1,53	1,56	1,82
116	1,00	0,55	1,74	1,26	2,81	1,90	2,57	2,18
117	1,00	0,43	0,83	1,06	1,87	1,42	2,38	2,83
118	1,00	1,50	0,39	0,92	0,83	1,04	1,03	1,22
161	1,00	1,23	1,09	1,14	1,38	1,60	1,62	1,80
162	1,00	1,13	0,93	1,06	3,25	2,25	1,98	1,95
163	1,00	0,42	0,50	0,93	2,48	2,61	2,53	2,74
164	1,00	0,98	2,20	1,24	1,54	1,74	0,99	2,23
165	1,00	2,48	5,17	1,36	2,93	2,03	2,47	2,30
166	1,00	0,14	0,17	0,99	1,20	2,41	1,14	5,02
167	1,00	0,20	0,26	0,96	1,25	14,91	1,56	7,47
168	1,00	2,03	4,77	1,05	2,53	1,56	1,72	1,68
169	1,00	1,42	2,27	1,35	2,12	1,05	1,38	2,19
16A	1,00	0,45	0,54	0,87	1,57	1,44	2,17	1,35
16B	1,00	0,31	2,53	1,43	2,54	1,99	2,59	2,13
171	1,00	1,12	1,64	1,40	1,10	2,11	1,60	2,01
172	1,00	1,09	2,67	1,56	2,18	1,49	2,30	2,05
16C	1,00	0,53	0,85	1,07	2,38	3,05	2,81	2,41
185	1,00	0,80	1,18	1,05	2,46	1,96	1,49	1,68
181	1,00	1,52	1,47	1,18	1,66	1,15	1,87	1,71
182	1,00	3,28	3,81	1,16	1,74	2,98	1,80	3,11
183	1,00	0,78	0,64	0,86	0,90	1,42	0,93	1,36
184	1,00	1,59	1,12	1,14	3,37	1,16	2,05	1,97
150	1,00	1,12	2,37	1,58	3,21	2,35	2,61	2,27
NUTSIII	***t*=3**							
111	1,00	3,90	1,44	1,21	1,48	1,58	2,07	2,07
112	1,00	0,64	3,90	1,36	1,69	1,35	1,71	2,17
113	1,00	0,76	1,29	1,17	4,04	3,16	2,87	2,63
114	1,00	0,52	1,08	1,11	2,20	2,96	2,64	2,64
115	1,00	0,66	3,17	1,11	2,12	1,84	1,63	1,65
116	1,00	0,45	1,78	1,17	2,38	1,61	3,06	2,91
117	1,00	0,31	2,04	1,20	3,18	4,25	3,20	3,60
118	1,00	1,56	0,33	0,93	0,97	1,22	2,16	2,34
161	1,00	1,54	1,96	1,31	2,26	2,09	1,90	1,84
162	1,00	1,09	0,74	0,92	2,26	1,88	1,68	1,79
163	1,00	0,80	0,83	0,99	2,81	2,30	2,09	2,06
164	1,00	1,51	1,17	1,03	0,78	1,60	1,67	2,07
165	1,00	1,02	1,35	1,11	2,18	0,99	2,45	2,35
166	1,00	2,86	2,12	1,01	0,99	0,91	1,34	1,34
167	1,00	0,27	0,33	0,94	3,28	4,22	5,80	10,69
168	1,00	0,70	0,53	0,96	0,83	1,06	4,37	7,98
169	1,00	4,95	5,91	1,27	1,65	2,59	2,05	2,16
16A	1,00	0,77	0,82	1,09	3,58	1,66	2,36	2,09
16B	1,00	0,33	2,96	1,46	2,20	2,22	2,05	2,01
171	1,00	1,29	1,13	1,16	1,05	1,97	1,52	1,84
172	1,00	0,67	0,75	0,98	1,50	1,77	1,64	1,85
16C	1,00	0,45	0,88	1,02	1,95	2,49	2,27	2,23
185	1,00	0,70	0,72	0,96	1,29	1,73	1,60	1,52
181	1,00	0,87	0,72	0,98	0,79	0,90	1,02	0,99
182	1,00	0,31	1,06	0,97	2,82	1,18	1,73	1,71
183	1,00	0,53	0,96	1,05	1,48	2,27	2,34	2,47
184	1,00	0,44	0,42	0,83	0,78	1,51	1,89	1,59
150	1,00	1,04	1,93	1,46	1,56	1,27	1,87	2,12

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=4**							
111	1,00	1,26	2,37	1,47	2,89	2,49	2,17	2,36
112	1,00	0,44	3,09	1,33	1,35	1,21	2,24	2,26
113	1,00	0,40	4,11	1,34	1,42	1,73	2,29	2,19
114	1,00	0,56	0,92	1,10	1,68	2,41	2,28	2,33
115	1,00	0,41	3,84	1,16	1,60	1,91	1,59	1,76
116	1,00	0,32	1,45	1,21	1,33	1,83	1,71	1,86
117	1,00	0,36	0,53	0,96	1,62	1,14	20,08	20,56
118	1,00	0,62	0,41	0,92	2,35	1,67	2,34	2,37
161	1,00	1,28	1,01	1,07	1,43	1,67	1,70	1,77
162	1,00	1,33	0,99	1,11	2,85	1,99	1,80	1,97
163	1,00	0,58	0,80	0,99	2,24	2,28	1,86	1,93
164	1,00	1,60	0,76	0,98	0,73	1,50	1,58	2,56
165	1,00	3,20	2,54	1,45	2,27	2,10	2,00	2,07
166	1,00	0,13	0,16	0,99	0,74	1,58	3,65	5,74
167	1,00	0,27	0,54	0,96	6,55	2,63	8,19	18,82
168	1,00	2,91	1,46	1,02	1,86	1,63	1,31	1,39
169	1,00	3,56	3,08	1,38	1,26	1,83	1,13	2,11
16A	1,00	0,60	0,77	1,01	2,68	1,78	1,95	1,83
16B	1,00	0,39	2,15	1,39	3,59	2,41	2,01	2,08
171	1,00	1,26	1,27	1,21	1,10	2,01	1,55	1,69
172	1,00	0,76	1,13	1,13	2,31	2,18	1,36	1,77
16C	1,00	0,52	2,98	1,14	1,33	1,56	1,69	1,73
185	1,00	0,40	0,93	1,05	1,80	2,30	1,74	2,02
181	1,00	1,04	0,78	1,03	0,87	1,00	1,10	1,07
182	1,00	0,37	1,90	1,10	4,02	2,24	2,59	2,05
183	1,00	0,30	0,46	0,81	0,65	0,90	1,39	1,45
184	1,00	0,22	0,79	0,98	2,74	1,21	2,30	1,91
150	1,00	1,16	0,76	0,99	1,46	1,73	1,43	1,71
NUTSIII	***t*=5**							
111	1,00	0,48	4,51	1,46	3,01	1,88	2,37	2,29
112	1,00	3,07	0,78	1,02	1,77	1,61	1,91	1,68
113	1,00	0,43	1,24	1,09	1,99	2,45	2,20	2,27
114	1,00	0,48	0,90	1,08	1,93	2,42	2,27	2,26
115	1,00	0,29	1,51	1,11	1,77	2,09	2,23	2,37
116	1,00	0,29	3,27	1,41	1,62	2,47	2,47	2,37
117	1,00	0,27	0,67	0,99	5,94	3,71	14,44	24,09
118	1,00	0,36	0,44	0,96	1,49	1,09	2,91	2,84
161	1,00	1,55	2,48	1,48	1,54	1,20	2,43	2,33
162	1,00	0,78	0,51	0,84	1,15	1,28	1,54	1,60
163	1,00	1,37	2,51	1,34	2,56	1,72	2,04	2,02
164	1,00	1,69	6,86	1,27	3,42	2,42	1,26	2,33
165	1,00	2,82	4,13	1,22	3,37	2,06	2,32	2,37
166	1,00	0,18	0,25	1,01	3,39	2,49	3,25	3,05
167	1,00	0,21	0,29	0,95	0,92	4,05	5,40	9,73
168	1,00	1,44	0,75	0,94	1,50	1,67	1,24	1,44
169	1,00	2,66	2,11	1,21	1,35	2,35	1,57	2,10
16A	1,00	0,29	0,34	1,07	0,76	0,98	2,06	2,05
16B	1,00	0,39	2,87	1,41	3,05	2,37	2,41	2,13
171	1,00	1,03	0,62	0,87	0,53	1,05	1,24	1,58
172	1,00	0,66	1,36	1,15	2,44	2,51	1,50	1,76
16C	1,00	0,67	1,71	1,14	2,21	2,00	2,09	1,95
185	1,00	0,51	1,09	1,07	1,65	1,94	1,83	2,16
181	1,00	1,10	0,92	1,03	1,08	1,23	1,17	1,15
182	1,00	0,22	0,70	0,98	0,68	1,24	1,90	2,59
183	1,00	0,34	3,35	1,25	1,21	0,71	1,42	1,91
184	1,00	0,39	0,43	0,81	1,11	1,79	1,21	1,39
150	1,00	1,09	1,27	1,24	2,73	2,41	1,39	1,95

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
				***t*=6**				
111	1,00	0,69	2,74	1,70	4,43	2,42	2,78	2,94
112	1,00	2,21	0,65	0,94	1,53	1,34	2,60	2,18
113	1,00	0,59	1,14	1,12	3,48	3,46	2,18	2,07
114	1,00	0,36	1,12	1,10	2,72	2,42	2,55	2,51
115	1,00	0,33	1,65	1,12	2,29	2,18	3,00	2,89
116	1,00	0,29	2,53	1,20	1,64	2,57	2,38	2,35
117	1,00	0,21	1,77	1,09	1,65	2,41	4,88	9,16
118	1,00	2,79	0,41	1,05	1,43	1,65	2,04	2,14
161	1,00	2,05	1,41	1,20	3,21	2,21	2,33	2,11
162	1,00	2,36	3,23	1,54	2,36	1,69	2,17	2,09
163	1,00	0,49	0,56	0,84	1,06	1,92	1,56	1,68
164	1,00	0,62	1,03	1,05	2,65	1,01	1,26	2,53
165	1,00	3,37	6,38	1,38	3,71	2,28	1,96	2,07
166	1,00	0,12	0,14	0,99	0,63	1,56	2,80	5,03
167	1,00	0,16	0,20	0,98	0,56	1,50	3,07	7,12
168	1,00	0,55	0,34	0,78	0,60	0,86	0,69	1,33
169	1,00	1,91	1,21	1,09	0,97	2,10	1,02	1,65
16A	1,00	0,36	0,43	0,70	0,83	1,03	1,27	0,96
16B	1,00	0,20	0,78	0,94	0,93	0,95	1,65	1,89
171	1,00	1,17	1,18	1,16	1,27	1,65	1,74	1,75
172	1,00	0,91	1,76	1,30	1,74	2,20	1,78	1,94
16C	1,00	0,59	0,46	0,83	1,63	2,07	2,80	1,92
185	1,00	0,45	1,58	1,13	2,28	1,85	1,87	2,06
181	1,00	1,80	4,01	1,26	1,30	0,73	1,76	2,07
182	1,00	1,61	2,45	1,13	1,68	2,49	1,57	2,73
183	1,00	1,29	0,76	1,02	1,59	1,86	2,15	2,10
184	1,00	0,76	0,48	0,99	1,32	1,67	2,02	2,01
150	1,00	1,05	1,17	1,20	1,46	1,10	2,02	1,98
NUTSIII				***t*=7**				
111	1,00	0,60	4,47	1,49	2,10	2,07	2,54	2,48
112	1,00	0,35	1,40	1,15	0,71	0,80	1,68	2,21
113	1,00	0,45	1,14	1,06	2,04	2,67	2,13	2,03
114	1,00	0,77	1,07	1,07	2,77	2,83	1,55	1,69
115	1,00	0,30	1,54	1,08	0,85	1,54	1,31	1,80
116	1,00	0,28	1,40	1,15	1,04	1,68	2,18	2,18
117	1,00	0,23	2,25	1,17	1,95	4,34	8,72	107,31
118	1,00	0,62	0,25	0,98	4,41	5,16	8,84	13,38
161	1,00	2,43	3,28	1,59	3,01	2,14	2,23	2,04
162	1,00	1,91	1,65	1,34	2,39	2,04	1,90	2,02
163	1,00	0,74	0,84	0,99	1,66	2,39	2,20	2,08
164	1,00	1,15	2,15	1,19	2,34	1,97	1,99	2,81
165	1,00	0,73	0,64	0,89	0,94	0,94	1,52	1,77
166	1,00	0,11	0,12	0,99	0,39	0,92	3,05	4,22
167	1,00	0,26	0,36	0,94	1,52	10,04	5,94	13,26
168	1,00	0,82	0,55	0,97	1,26	1,50	2,53	2,49
169	1,00	1,53	1,19	1,07	0,96	1,87	1,30	2,07
16A	1,00	0,73	0,87	1,11	2,92	0,77	3,11	3,00
16B	1,00	0,28	1,45	1,17	1,43	1,53	2,12	1,92
171	1,00	1,31	1,08	1,09	1,60	1,51	1,91	1,86
172	1,00	1,31	1,39	1,15	1,19	2,04	1,65	1,84
16C	1,00	0,33	0,84	0,94	3,16	1,49	1,87	1,87
185	1,00	0,52	0,93	1,06	1,36	1,92	2,70	2,80
181	1,00	0,96	0,72	0,96	1,00	0,99	1,23	1,10
182	1,00	0,74	3,83	1,27	2,23	2,79	1,37	2,22
183	1,00	0,53	1,99	1,25	2,80	1,49	1,83	2,08
184	1,00	0,50	0,33	0,94	0,56	0,75	0,92	0,88
150	1,00	1,08	2,41	1,53	2,77	1,67	1,47	2,33

A15.11 Taxa de cobertura do IC *design-based*

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=1**							
111	96%	47%	3%	88%	67%	91%	90%	90%
112	95%	57%	0%	97%	93%	96%	95%	94%
113	94%	49%	1%	96%	89%	90%	95%	95%
114	94%	80%	0%	94%	46%	62%	90%	91%
115	95%	0%	1%	96%	65%	90%	97%	98%
116	95%	46%	3%	95%	60%	50%	95%	94%
117	99%	64%	0%	96%	4%	7%	65%	75%
118	96%	75%	0%	96%	82%	92%	83%	89%
161	95%	66%	0%	96%	56%	86%	94%	91%
162	96%	95%	95%	91%	53%	70%	94%	95%
163	96%	62%	61%	93%	59%	66%	93%	95%
164	93%	77%	0%	94%	36%	63%	77%	91%
165	97%	61%	0%	97%	23%	85%	93%	94%
166	95%	0%	0%	95%	44%	76%	2%	0%
167	96%	0%	0%	96%	59%	79%	25%	0%
168	95%	0%	24%	96%	75%	65%	100%	100%
169	93%	3%	0%	98%	21%	79%	75%	86%
16A	96%	0%	0%	99%	91%	80%	87%	96%
16B	94%	0%	21%	95%	86%	89%	95%	95%
171	95%	95%	93%	94%	74%	94%	88%	95%
172	95%	77%	84%	96%	94%	95%	94%	95%
16C	95%	69%	0%	93%	68%	66%	95%	92%
185	96%	37%	1%	96%	92%	91%	95%	95%
181	98%	76%	0%	96%	36%	96%	75%	68%
182	97%	93%	28%	92%	19%	89%	79%	95%
183	97%	40%	0%	97%	2%	48%	59%	54%
184	93%	32%	0%	92%	33%	92%	87%	83%
150	95%	95%	33%	96%	54%	76%	88%	93%

NUTSIII	$\widehat{\mu}_i^{dir}$	$\widehat{\mu}_i^{sinQ}$	$\widehat{\mu}_i^{sinR}$	$\widehat{\mu}_i^{com}$	$\widehat{\mu}_i^{FH}$	$\widehat{\mu}_i^{NS}$	$\widehat{\mu}_{it}^{RY}$	$\widehat{\mu}_{it}^{LP}$
				***t*=2**				
111	94%	76%	0%	95%	5%	58%	69%	73%
112	94%	65%	0%	90%	47%	67%	86%	90%
113	95%	1%	37%	95%	95%	96%	91%	95%
114	95%	3%	89%	94%	95%	88%	90%	96%
115	97%	0%	80%	94%	93%	91%	90%	94%
116	96%	0%	59%	93%	92%	86%	95%	95%
117	95%	0%	41%	95%	60%	57%	95%	99%
118	94%	42%	0%	92%	40%	79%	78%	84%
161	93%	27%	0%	96%	35%	78%	72%	91%
162	95%	73%	29%	96%	95%	95%	90%	92%
163	94%	10%	0%	95%	89%	95%	91%	94%
164	95%	39%	83%	94%	89%	91%	64%	93%
165	95%	84%	95%	95%	96%	94%	97%	96%
166	96%	0%	0%	96%	61%	96%	84%	0%
167	94%	0%	0%	95%	57%	0%	86%	0%
168	100%	13%	92%	99%	93%	92%	87%	86%
169	95%	35%	29%	89%	96%	62%	89%	86%
16A	95%	0%	0%	98%	69%	96%	94%	97%
16B	95%	0%	87%	92%	88%	95%	93%	95%
171	96%	96%	93%	95%	59%	91%	74%	95%
172	96%	91%	95%	92%	79%	68%	95%	96%
16C	95%	0%	1%	97%	84%	95%	93%	94%
185	96%	31%	1%	96%	95%	91%	91%	94%
181	98%	69%	1%	95%	46%	97%	65%	72%
182	93%	95%	22%	93%	16%	90%	60%	95%
183	96%	49%	0%	98%	19%	82%	68%	81%
184	94%	94%	1%	96%	96%	82%	90%	89%
150	95%	95%	91%	95%	95%	93%	93%	95%
NUTSIII				***t*=3**				
111	96%	87%	0%	90%	4%	57%	85%	86%
112	95%	1%	93%	93%	74%	81%	93%	95%
113	94%	17%	26%	95%	92%	90%	93%	92%
114	96%	37%	85%	95%	89%	95%	95%	95%
115	94%	1%	89%	92%	91%	94%	86%	88%
116	95%	4%	75%	95%	91%	87%	95%	94%
117	95%	0%	94%	93%	83%	87%	93%	94%
118	95%	89%	0%	94%	78%	84%	92%	94%
161	95%	72%	44%	94%	90%	95%	95%	96%
162	95%	56%	13%	96%	69%	83%	92%	92%
163	95%	50%	5%	97%	93%	93%	96%	96%
164	93%	73%	18%	93%	71%	90%	94%	97%
165	94%	48%	25%	94%	89%	84%	95%	94%
166	86%	10%	0%	83%	27%	23%	66%	66%
167	95%	0%	0%	95%	75%	63%	77%	69%
168	95%	60%	7%	96%	86%	86%	86%	75%
169	91%	92%	73%	89%	49%	94%	91%	92%
16A	86%	0%	0%	89%	93%	83%	92%	89%
16B	94%	0%	95%	93%	71%	93%	93%	95%
171	95%	93%	95%	96%	73%	95%	91%	95%
172	96%	56%	63%	94%	82%	94%	92%	94%
16C	95%	0%	1%	95%	88%	94%	95%	95%
185	95%	94%	0%	97%	41%	84%	91%	90%
181	99%	44%	0%	98%	1%	28%	31%	36%
182	97%	71%	0%	98%	91%	75%	93%	93%
183	95%	94%	0%	96%	52%	94%	90%	88%
184	94%	98%	0%	91%	37%	91%	91%	87%
150	95%	95%	82%	94%	19%	19%	95%	91%

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=4**							
111	96%	26%	13%	87%	69%	86%	86%	90%
112	95%	2%	83%	93%	62%	80%	95%	95%
113	95%	1%	95%	92%	38%	61%	94%	95%
114	95%	52%	68%	95%	79%	90%	94%	95%
115	96%	0%	94%	93%	95%	96%	97%	96%
116	95%	0%	27%	95%	52%	79%	93%	94%
117	95%	16%	1%	95%	62%	57%	93%	68%
118	96%	44%	0%	95%	96%	93%	98%	98%
161	95%	56%	0%	95%	30%	82%	90%	92%
162	95%	87%	64%	95%	91%	90%	93%	94%
163	95%	11%	1%	97%	85%	93%	95%	95%
164	93%	86%	0%	95%	75%	87%	92%	94%
165	95%	94%	30%	93%	71%	96%	96%	93%
166	95%	0%	0%	95%	37%	77%	80%	67%
167	95%	0%	4%	95%	93%	53%	86%	70%
168	99%	94%	8%	99%	100%	100%	100%	100%
169	96%	95%	40%	93%	67%	92%	89%	94%
16A	91%	0%	0%	96%	95%	87%	93%	95%
16B	95%	0%	86%	95%	94%	95%	95%	96%
171	95%	95%	95%	95%	76%	95%	92%	96%
172	95%	68%	78%	96%	95%	94%	92%	95%
16C	97%	0%	78%	94%	63%	80%	93%	95%
185	95%	94%	1%	96%	62%	96%	87%	91%
181	93%	59%	0%	97%	1%	28%	40%	50%
182	96%	60%	1%	92%	87%	73%	94%	90%
183	95%	97%	0%	92%	21%	67%	89%	89%
184	93%	90%	0%	98%	95%	88%	94%	94%
150	94%	95%	55%	93%	72%	89%	91%	93%
NUTSIII	***t*=5**							
111	94%	0%	89%	92%	83%	93%	93%	95%
112	96%	88%	0%	97%	8%	53%	76%	66%
113	94%	2%	6%	95%	75%	90%	94%	96%
114	95%	37%	72%	95%	84%	91%	94%	94%
115	96%	0%	54%	94%	93%	94%	96%	95%
116	95%	0%	90%	93%	68%	92%	95%	95%
117	94%	1%	7%	95%	93%	74%	97%	25%
118	96%	8%	0%	94%	87%	77%	98%	98%
161	94%	90%	77%	94%	41%	67%	95%	95%
162	95%	57%	9%	96%	37%	73%	89%	92%
163	94%	86%	75%	94%	82%	74%	92%	96%
164	92%	56%	91%	91%	91%	80%	75%	94%
165	96%	95%	66%	92%	96%	94%	93%	93%
166	94%	0%	0%	94%	92%	75%	82%	79%
167	94%	0%	0%	94%	57%	51%	92%	4%
168	96%	73%	0%	98%	92%	97%	99%	100%
169	96%	79%	0%	92%	50%	94%	91%	94%
16A	93%	0%	0%	95%	34%	99%	91%	92%
16B	95%	0%	97%	94%	93%	94%	96%	95%
171	95%	91%	62%	91%	40%	74%	88%	93%
172	96%	27%	77%	97%	94%	95%	93%	94%
16C	95%	1%	33%	95%	76%	76%	93%	94%
185	94%	99%	4%	95%	71%	97%	91%	95%
181	94%	75%	0%	94%	9%	62%	61%	66%
182	95%	91%	0%	96%	55%	94%	90%	92%
183	96%	81%	78%	94%	56%	46%	94%	96%
184	95%	99%	0%	88%	48%	95%	86%	86%
150	95%	96%	83%	94%	94%	95%	89%	95%

NUTSIII	$\widehat{\mu}_i^{dir}$	$\widehat{\mu}_i^{sinQ}$	$\widehat{\mu}_i^{sinR}$	$\widehat{\mu}_i^{com}$	$\widehat{\mu}_i^{FH}$	$\widehat{\mu}_i^{NS}$	$\widehat{\mu}_{it}^{RY}$	$\widehat{\mu}_{it}^{LP}$
				***t*=6**				
111	96%	1%	35%	94%	92%	92%	96%	93%
112	94%	78%	0%	97%	25%	64%	94%	89%
113	95%	2%	3%	96%	93%	94%	89%	95%
114	96%	6%	71%	95%	93%	92%	95%	96%
115	95%	0%	72%	94%	91%	92%	92%	92%
116	95%	0%	78%	93%	86%	95%	95%	94%
117	95%	0%	90%	94%	83%	83%	88%	98%
118	91%	95%	0%	91%	57%	82%	96%	97%
161	95%	88%	4%	95%	90%	95%	93%	93%
162	95%	95%	95%	92%	78%	77%	93%	95%
163	95%	1%	0%	97%	24%	84%	90%	94%
164	94%	0%	1%	95%	90%	49%	81%	94%
165	95%	87%	84%	90%	95%	93%	88%	94%
166	95%	0%	0%	95%	53%	67%	72%	0%
167	95%	0%	0%	95%	54%	50%	59%	0%
168	95%	4%	0%	94%	62%	81%	85%	88%
169	95%	37%	0%	96%	26%	93%	73%	92%
16A	97%	0%	0%	99%	20%	98%	95%	92%
16B	95%	0%	3%	97%	44%	73%	89%	93%
171	95%	94%	92%	94%	88%	94%	91%	95%
172	96%	72%	84%	95%	85%	95%	92%	95%
16C	93%	0%	0%	91%	71%	87%	95%	91%
185	95%	1%	22%	95%	91%	92%	92%	95%
181	95%	81%	87%	94%	82%	81%	97%	96%
182	95%	85%	23%	93%	61%	93%	62%	93%
183	96%	80%	0%	97%	52%	96%	86%	86%
184	95%	48%	0%	95%	65%	95%	90%	89%
150	96%	95%	16%	94%	23%	14%	78%	80%
NUTSIII				***t*=7**				
111	96%	39%	68%	93%	63%	91%	93%	93%
112	93%	2%	1%	95%	28%	56%	90%	95%
113	95%	62%	5%	96%	82%	94%	96%	95%
114	97%	79%	10%	97%	89%	89%	90%	95%
115	94%	4%	20%	94%	53%	89%	91%	96%
116	95%	11%	16%	96%	57%	85%	94%	94%
117	96%	36%	92%	94%	77%	83%	16%	89%
118	95%	94%	0%	95%	89%	93%	92%	96%
161	95%	89%	75%	92%	88%	95%	95%	94%
162	95%	94%	75%	95%	91%	94%	94%	95%
163	95%	25%	1%	97%	75%	95%	95%	95%
164	93%	42%	44%	94%	89%	88%	92%	96%
165	95%	3%	0%	97%	61%	80%	90%	92%
166	95%	0%	0%	95%	49%	54%	0%	0%
167	95%	0%	0%	95%	66%	22%	1%	0%
168	95%	45%	0%	97%	87%	89%	100%	100%
169	94%	26%	0%	95%	30%	95%	74%	95%
16A	94%	3%	0%	98%	92%	80%	95%	89%
16B	95%	0%	56%	95%	38%	85%	95%	95%
171	94%	94%	94%	95%	95%	94%	94%	95%
172	95%	80%	76%	95%	65%	95%	93%	94%
16C	96%	0%	3%	97%	94%	77%	91%	95%
185	94%	88%	4%	95%	53%	94%	97%	96%
181	95%	77%	0%	97%	34%	75%	79%	70%
182	96%	82%	44%	93%	80%	93%	88%	93%
183	95%	89%	50%	94%	90%	85%	91%	96%
184	98%	54%	0%	99%	6%	30%	91%	93%
150	94%	94%	92%	94%	95%	80%	88%	95%

A15.12 Taxa de cobertura do IC *model-based*

NUTSIII	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=1**				***t*=2**			
111	85%	91%	90%	90%	24%	60%	98%	95%
112	98%	95%	95%	94%	51%	50%	99%	97%
113	100%	100%	100%	100%	99%	98%	97%	95%
114	72%	92%	93%	93%	99%	99%	99%	99%
115	67%	82%	73%	76%	91%	86%	89%	84%
116	96%	87%	96%	95%	100%	99%	99%	99%
117	61%	63%	65%	65%	100%	100%	100%	100%
118	79%	78%	59%	59%	54%	77%	88%	84%
161	84%	89%	96%	95%	51%	73%	96%	96%
162	81%	80%	96%	96%	100%	98%	100%	99%
163	95%	92%	92%	92%	99%	99%	99%	98%
164	26%	81%	77%	87%	89%	92%	86%	91%
165	33%	69%	88%	88%	95%	88%	94%	92%
166	94%	100%	100%	100%	99%	100%	100%	100%
167	100%	100%	100%	100%	100%	100%	100%	100%
168	98%	61%	94%	100%	78%	40%	73%	69%
169	8%	65%	59%	95%	86%	49%	94%	90%
16A	88%	75%	76%	72%	74%	70%	99%	95%
16B	100%	90%	97%	96%	98%	96%	99%	98%
171	66%	96%	98%	99%	50%	90%	95%	96%
172	99%	96%	99%	98%	92%	71%	98%	95%
16C	97%	95%	95%	95%	100%	100%	100%	98%
185	98%	88%	93%	92%	95%	88%	90%	91%
181	27%	33%	78%	75%	27%	36%	84%	81%
182	31%	79%	70%	72%	27%	84%	82%	99%
183	10%	43%	68%	68%	24%	66%	85%	94%
184	36%	92%	92%	92%	99%	68%	99%	98%
150	83%	81%	99%	100%	99%	95%	100%	99%
NUTSIII	***t*=3**				***t*=4**			
111	23%	57%	90%	89%	87%	84%	90%	90%
112	89%	73%	94%	94%	92%	85%	98%	98%
113	100%	100%	99%	99%	94%	96%	93%	93%
114	100%	100%	99%	99%	99%	99%	99%	99%
115	88%	86%	79%	79%	80%	79%	69%	68%
116	98%	97%	96%	96%	90%	97%	93%	93%
117	100%	100%	100%	100%	100%	100%	100%	100%
118	90%	93%	98%	98%	88%	90%	77%	77%
161	95%	93%	94%	94%	72%	82%	91%	91%
162	98%	90%	99%	99%	99%	96%	99%	99%
163	99%	98%	97%	97%	98%	98%	95%	96%
164	48%	78%	69%	72%	66%	93%	95%	99%
165	98%	77%	99%	99%	85%	81%	82%	78%
166	5%	4%	65%	65%	91%	100%	100%	100%
167	100%	100%	100%	100%	100%	100%	100%	100%
168	89%	92%	100%	100%	75%	73%	52%	52%
169	36%	78%	93%	95%	41%	78%	86%	93%
16A	92%	74%	87%	83%	93%	76%	88%	87%
16B	90%	92%	99%	99%	98%	94%	97%	97%
171	64%	94%	97%	99%	59%	92%	98%	99%
172	95%	96%	97%	98%	96%	93%	95%	94%
16C	98%	98%	97%	97%	75%	85%	84%	83%
185	47%	76%	87%	86%	79%	89%	89%	90%
181	0%	3%	16%	21%	0%	3%	19%	24%
182	96%	60%	92%	92%	95%	60%	95%	95%
183	59%	89%	96%	97%	14%	45%	91%	95%
184	35%	88%	98%	95%	98%	83%	99%	100%
150	38%	29%	93%	91%	89%	93%	99%	99%

NUTSIII	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	***t*=5**				***t*=6**			
111	99%	93%	98%	98%	97%	90%	96%	94%
112	60%	58%	84%	82%	79%	69%	94%	92%
113	99%	99%	98%	97%	100%	99%	98%	99%
114	99%	99%	97%	97%	99%	98%	97%	97%
115	93%	91%	82%	80%	98%	98%	96%	96%
116	98%	98%	97%	97%	98%	98%	96%	95%
117	100%	100%	100%	100%	100%	100%	100%	100%
118	90%	88%	78%	78%	70%	73%	77%	77%
161	92%	72%	99%	98%	100%	96%	99%	99%
162	76%	80%	98%	99%	95%	82%	98%	98%
163	98%	94%	95%	96%	78%	97%	97%	97%
164	93%	90%	83%	95%	99%	71%	89%	97%
165	97%	88%	95%	95%	97%	85%	95%	96%
166	100%	100%	100%	100%	92%	100%	100%	100%
167	100%	100%	100%	100%	85%	100%	100%	100%
168	74%	75%	52%	52%	56%	76%	67%	89%
169	31%	84%	93%	96%	19%	85%	70%	95%
16A	36%	59%	93%	93%	20%	36%	89%	86%
16B	95%	91%	98%	97%	59%	64%	97%	97%
171	24%	68%	95%	98%	77%	89%	98%	99%
172	95%	94%	94%	94%	91%	94%	98%	98%
16C	94%	93%	91%	91%	97%	98%	95%	94%
185	87%	92%	90%	91%	98%	92%	97%	97%
181	4%	21%	27%	31%	40%	40%	88%	86%
182	65%	85%	87%	88%	73%	92%	95%	99%
183	45%	21%	88%	88%	74%	84%	91%	90%
184	65%	94%	93%	96%	65%	85%	82%	81%
150	98%	95%	98%	99%	37%	25%	92%	92%
NUTSIII	***t*=7**							
111	72%	87%	93%	93%				
112	28%	48%	97%	96%				
113	96%	96%	92%	91%				
114	97%	97%	86%	89%				
115	57%	91%	91%	94%				
116	89%	97%	93%	93%				
117	100%	100%	100%	100%				
118	100%	100%	100%	100%				
161	99%	95%	96%	95%				
162	99%	97%	98%	97%				
163	99%	99%	97%	96%				
164	97%	99%	99%	99%				
165	63%	70%	93%	93%				
166	52%	100%	100%	100%				
167	100%	100%	100%	100%				
168	100%	100%	100%	100%				
169	30%	84%	88%	93%				
16A	97%	75%	98%	97%				
16B	89%	90%	99%	99%				
171	95%	94%	98%	99%				
172	74%	96%	96%	96%				
16C	98%	93%	93%	95%				
185	87%	91%	91%	91%				
181	10%	34%	56%	54%				
182	70%	91%	90%	93%				
183	98%	82%	96%	98%				
184	6%	19%	69%	69%				
150	96%	78%	98%	98%				

Apêndice 16 – Geração de uma pseudo-população e cálculo do verdadeiro valor dos parâmetros de interesse

FICHEIROS DE DADOS UTILIZADOS:

Amos_trans.sas7bdat – ficheiro que contém informação sobre as amostras reais de transacções de habitações usadas no estudo empírico: data da realização da transacção (ano, trimestre e mes), identificação da empresa que realizou a transacção (NPC), preço de transacção por metro quadrado (preco), NUT I (N1), NUT II (N2) e NUT III (N3) onde se localiza o imóvel e os códigos de NUT II e de NUT III (cod_nut_ii_trans e cod_nut_iii_trans, respectivamente).

CodigoDominios.sas7bdat – ficheiro que contém informação sobre as NUTSIII. Neste ficheiro encontra-se informação sobre os códigos associados a cada NUTSIII.

Ppopul.sas7bdat – ficheiro que contém a pseudo-população de empresas de mediação imobiliária. Neste ficheiro encontra-se informação sobre a identificação de cada empresa (NPC) e sobre o estrato a que pertence (estrato).

Pdims.sas7bdat – ficheiro que contém informação sobre as dimensões amostral (m_h) e pseudo-populacional (Mh) em cada cada estrato (estrato). Neste ficheiro encontra-se também informação sobre coeficiente de extrapolação (coef) associado a cada estrato.

Este código SAS serve para criar os seguintes ficheiros de dados:

- **Pseudo_populacao_transaccoes.sas7bdat**: ficheiro formado por todas as transacções efectuadas pelas empresas que pertencem à pseudo-população. Este ficheiro é utilizado para se calcularem os verdadeiros valores dos parâmetros de interesse;

- **Pseudo_amostra_transaccoes.sas7bdat**: ficheiro formado por todas as transacções que pertencem às 1000 amostras extraídas da pseudo-população. Este ficheiro é utilizado para se calcularem as estimativas directas por NUTSI, NUTSII e NUTSIII;

- **Verdadeiro_parametro.sas7bdat:** ficheiro com os verdadeiros valores dos parâmetros de interesse.

```
/**********************************************
*********************************************/
```

<u>CÓDIGO SAS</u>

```
/*******************************************/
/*Criação das pseudo-amostras de transacções*/
/*******************************************/

Data amos_trans;
set hb.amos_trans;
rename trimestre=t;
run;

Proc sort data=amos_trans out=amos_trans;
by cod_nut_iii_trans;
run;

Data amos_trans;
merge amos_trans hb.CodigoDominios;
by cod_nut_iii_trans;
run;

Data amos_trans;
set amos_trans;
drop cod_nut_iii_trans;
run;

Proc sort data=amos_trans out=amos_trans;
by ano t npc;
run;

Data trans_q1;
set amos_trans;
where ano=2002 and t=1;
Data trans_q2;
set amos_trans;
where ano=2002 and t=2;
Data trans_q3;
set amos_trans;
where ano=2002 and t=3;
Data trans_q4;
set amos_trans;
where ano=2002 and t=4;
Data trans_q5;
set amos_trans;
where ano=2003 and t=5;
Data trans_q6;
set amos_trans;
where ano=2003 and t=6;
Data trans_q7;
set amos_trans;
where ano=2003 and t=7;
run;

Data trans_q1;
set trans_q1;
keep npc preco t cod_nut_ii_trans i ano;
Data trans_q2;
set trans_q2;
keep npc preco t cod_nut_ii_trans i ano;
Data trans_q3;
set trans_q3;
keep npc preco t cod_nut_ii_trans i ano;
Data trans_q4;
set trans_q4;
keep npc preco t cod_nut_ii_trans i ano;
Data trans_q5;
set trans_q5;
keep npc preco t cod_nut_ii_trans i ano;
Data trans_q6;
set trans_q6;
keep npc preco t cod_nut_ii_trans i ano;
Data trans_q7;
set trans_q7;
keep npc preco t cod_nut_ii_trans i ano;
run;

Data null;
file 'C:\Documents and Settings\Luis Pereira
\My Documents\EstudoEmpirico\chamamacro.txt';
   do i= 1 to 1000; k=i;
var='%ext_am('||i||','||k||');'; put var; end;
run;

%macro ext_am(i,k);

/* Extracção das amostras */

Data amostra;
merge hb.ppopul hb.pdims;
by estrato;
if ranuni(&i) <= m_h/npop then do;
output;
m_h=m_h - 1;
end;
npop=npop - 1;
drop mh m_h npop coef;
```

```
run;

/* Fim da extracção das amostras */
/*Dimensões amostrais efectivas por estrato */

proc means data=amostra nway noprint;
   var npc;
   class estrato;
   output out=resultn n=m;
run;

data resultn;
merge resultn hb.pdims;
by estrato;
if m=. then m=0;
run;

data resultn;
   set resultn;
   coef=npop/m; /* Este é o novo coeficiente
de extrapolação calculado com as dimensões
populacionais da pseudo-população e da amostra
dela extraída */
   drop _type_ _freq_ m npop m_h mh;
run;

/* É necessário atribuir o novo coef à amostra
de empresas */

Proc sort data=amostra;
by estrato;
Proc sort data=resultn;
by estrato;
run;

Data amostra;
merge amostra (in=a) resultn (in=b);
by estrato;
if a=1 and b=1 then output;
run;

/* Fim das dimensões amostrais efectivas por
estrato */

/* Dimensões amostrais efectivas de cada
empresa */

proc sort data=amostra;
by npc;
run;

Data amostra_aux;
set amostra;
var1=1;
run;

proc means data=amostra_aux nway noprint;
class npc;
output out=resultemp n=n;
run;

data resultemp;
   set resultemp;
   drop _type_ _freq_;
run;

/* Fim das dimensões amostrais efectivas de
cada empresa */

/*********************************************
*********************************************/
/* Criação das pseudo-amostras de transacções
para cada trimestre */
/*********************************************
*********************************************/

Data amos_first;
set amostra;
by npc;
if first.npc;
run;

Data resultemp;
merge resultemp amos_first;
by npc;
run;

Data amos_transq1;
merge resultemp (in=a) trans_q1 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq2;
merge resultemp (in=a) trans_q2 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq3;
merge resultemp (in=a) trans_q3 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq4;
merge resultemp (in=a) trans_q4 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq5;
merge resultemp (in=a) trans_q5 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq6;
merge resultemp (in=a) trans_q6 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq7;
merge resultemp (in=a) trans_q7 (in=b);
by npc;
if a=1 and b=1 then output;
run;

Data s_transq1;
set amos_transq1;
do j=1 to n;
by npc;
output;
end;
Data s_transq2;
set amos_transq2;
do j=1 to n;
by npc;
output;
end;
Data s_transq3;
set amos_transq3;
do j=1 to n;
by npc;
output;
end;
Data s_transq4;
set amos_transq4;
do j=1 to n;
by npc;
output;
end;
Data s_transq5;
set amos_transq5;
do j=1 to n;
by npc;
output;
end;
Data s_transq6;
set amos_transq6;
do j=1 to n;
by npc;
output;
end;
Data s_transq7;
set amos_transq7;
do j=1 to n;
by npc;
output;
end;
run;

Data s_transq1;   /* Ficheiro com as
transacções do 1º trimestre */
set s_transq1;
drop n j;
Data s_transq2;
set s_transq2;
drop n j;
Data s_transq3;
set s_transq3;
drop n j;
Data s_transq4;
set s_transq4;
drop n j;
Data s_transq5;
set s_transq5;
```

```
drop n j;
Data s_transq6;
set s_transq6;
drop n j;
Data s_transq7;
set s_transq7;
drop n j;
run;

Data stotal_trans;
set s_transq1 s_transq2 s_transq3 s_transq4
s_transq5
s_transq6 s_transq7;
run;

Data stotal_trans;
set stotal_trans;
simul=&k;
run;

Proc append base=Hb.Pseudo_amostra_trans
data=stotal_trans;
run;

DM 'CLEAR OUTPUT';
DM 'CLEAR LOG';
%mend;
%inc 'C:\Documents and Settings\Luis Pereira
\My Documents\EstudoEmpirico\chamamacro.txt';

/*********************************************
*********************************************/
/*Criação das pseudo-populacoes transacções*/
/*********************************************
*********************************************/

/* Dimensões populacionais efectivas de cada
empresa */

proc sort data=Hb.ppopul out=ppopul;
by npc;
run;

Data ppopul_aux;
set ppopul;
var1=1;
run;

proc means data=ppopul_aux nway noprint;
class npc;
output out=resultemp n=n;
run;

data resultemp;
   set resultemp;
   drop _type_ _freq_;
run;

/* Fim das dimensões populacionais efectivas
de cada empresa */

/*********************************************
*********************************************/
/* Criação das pseudo-populacções de
transacções para cada trimestre */
/*********************************************
*********************************************/

Data amos_first;
set ppopul;
by npc;
if first.npc;
run;

Data resultemp;
merge resultemp amos_first;
by npc;
run;

Data ppopul_transq1;
merge resultemp (in=a) trans_q1 (in=b);
by npc;
if a=1 and b=1 then output;
Data ppopul_transq2;
merge resultemp (in=a) trans_q2 (in=b);
by npc;
if a=1 and b=1 then output;
Data ppopul_transq3;
merge resultemp (in=a) trans_q3 (in=b);
by npc;
if a=1 and b=1 then output;
Data ppopul_transq4;
merge resultemp (in=a) trans_q4 (in=b);
by npc;
if a=1 and b=1 then output;
Data ppopul_transq5;
merge resultemp (in=a) trans_q5 (in=b);
by npc;
if a=1 and b=1 then output;
Data ppopul_transq6;
merge resultemp (in=a) trans_q6 (in=b);
by npc;
if a=1 and b=1 then output;
Data ppopul_transq7;
merge resultemp (in=a) trans_q7 (in=b);
by npc;
if a=1 and b=1 then output;
run;

Data s_transq1;
set ppopul_transq1;
do j=1 to n;
by npc;
output;
end;
Data s_transq2;
set ppopul_transq2;
do j=1 to n;
by npc;
output;
end;
Data s_transq3;
set ppopul_transq3;
do j=1 to n;
by npc;
output;
end;
Data s_transq4;
set ppopul_transq4;
do j=1 to n;
by npc;
output;
end;
Data s_transq5;
set ppopul_transq5;
do j=1 to n;
by npc;
output;
end;
Data s_transq6;
set ppopul_transq6;
do j=1 to n;
by npc;
output;
end;
Data s_transq7;
set ppopul_transq7;
do j=1 to n;
by npc;
output;
end;
run;

Data s_transq1;   /* Ficheiro com as
transacções do 1º trimestre */
set s_transq1;
drop n j;
Data s_transq2;
set s_transq2;
drop n j;
Data s_transq3;
set s_transq3;
drop n j;
Data s_transq4;
set s_transq4;
drop n j;
Data s_transq5;
set s_transq5;
drop n j;
Data s_transq6;
set s_transq6;
drop n j;
Data s_transq7;
set s_transq7;
drop n j;
run;
```

```
Data stotal_trans;
set s_transq1 s_transq2 s_transq3 s_transq4
s_transq5
s_transq6 s_transq7;
run;

Proc append base=Hb.Pseudo_populacao_trans
data=stotal_trans;
run;

/*** Cálculo do verdadeiro valor do parâmetro
de interesse na pseudo-população de
transacções gerada a partir de uma amostra de
dados reais ***/

Proc sort data=Hb.Pseudo_populacao_trans
out=Pseudo_populacao_trans;
by t i;
run;

proc means data=Pseudo_populacao_trans;
by t i;
var preco;
output out=hb.verdadeiro_parametro
mean=parametro var=variancia;
run;

proc means data=Pseudo_populacao_trans;
by t i;
var preco;
output out=hb.verdadeiro_parametro1
mean=parametro var=variancia min=minimo
max=maximo median=mediana;
run;

Proc sort data=Pseudo_populacao_trans
out=Pseudo_populacao_trans;
by i;
run;

proc means data=Pseudo_populacao_trans;
by i;
var preco;
output out=hb.verdadeiro_parametro2
mean=parametro var=variancia min=minimo
max=maximo median=mediana;
run;

proc means data=Pseudo_populacao_trans;
var preco;
output out=hb.verdadeiro_parametro3
mean=parametro var=variancia min=minimo
max=maximo median=mediana;
run;

Data hb.verdadeiro_parametro1;
set hb.verdadeiro_parametro1;
cv=sqrt(variancia)/parametro;
run;

Data hb.verdadeiro_parametro2;
set hb.verdadeiro_parametro2;
cv=sqrt(variancia)/parametro;
run;

Data hb.verdadeiro_parametro3;
set hb.verdadeiro_parametro3;
cv=sqrt(variancia)/parametro;
run;
```

Apêndice 17 – Estimação dos parâmetros de interesse com base em estimadores tradicionais no âmbito do estudo empírico por simulação *design-based*

FICHEIROS DE DADOS UTILIZADOS:

Amos_trans.sas7bdat; CodigoDominios.sas7bdat; Ppopul.sas7bdat;

Pdims_229 – ficheiro que contém informação sobre as dimensões amostral (m_h) e pseudo-populacional (Mh) em cada cada estrato (estrato), bem como o coeficiente de extrapolação (coef) associado a cada estrato para amostras de dimensão 229.

Iabh_total_nutiii_ii.sas7bdat – ficheiro que contém informação sobre os preços médios de avaliação bancária da habitação ao nível de NUTSIII e de NUTSII em cada trimestre (preco_01q4, preco_02q1, preco_02q2, preco_02q3, preco_02q4, preco_03q1, preco_03q2).

Est_fh_simulacao.sas7bdat – ficheiro que contém as estimativas dos parâmetros de interesse produzidas pelo modelo de Fay-Herriot, bem como as respectivas estimativas dos efeitos fixos e dos efeitos aleatórios.

O parâmetro de interesse é estimado utilizando os seguintes estimadores tradicionais:
- Estimador directo;
- Estimador sintético pelo quociente;
- Estimador sintético pela regressão;
- Estimador combinado com pesos dependentes dos dados.

Os resultados estão guardados nos seguintes ficheiros:
- Estimativas directas ao nível de NUTSIII: **Estimat_directas_pseudo.sas7bdat**
- Estimativas directas ao nível de NUTSII: **Estimat_directasNUTSII_pseudo.sas7bdat**
- Estimativas directas ao nível de NUTSI: **Estimat_directasNUTSI_pseudo.sas7bdat**
- Estimativas sintéticas pelo quociente: **Est_sintQuoc_simulacao.sas7bdat**
- Estimativas sintéticas pela regressão: **Est_sintReg_simulacao.sas7bdat**
- Estimativas combinadas com pesos dependentes dos dados: **Est_comb_simulacao.sas7bdat**

```
/*********************************************
*********************************************/
```

CÓDIGO SAS

```
Data amos_trans;
set hb.amos_trans;
rename trimestre=t;
run;

Proc sort data=amos_trans out=amos_trans;
by cod_nut_iii_trans;
run;

Data amos_trans;
merge amos_trans hb.CodigoDominios;
by cod_nut_iii_trans;
run;

Data amos_trans;
set amos_trans;
drop cod_nut_iii_trans;
run;

Proc sort data=amos_trans out=amos_trans;
by ano t npc;
run;

Data trans_q1;
set amos_trans;
where ano=2002 and t=1;
Data trans_q2;
set amos_trans;
where ano=2002 and t=2;
Data trans_q3;
set amos_trans;
where ano=2002 and t=3;
Data trans_q4;
set amos_trans;
where ano=2002 and t=4;
Data trans_q5;
set amos_trans;
where ano=2003 and t=5;
Data trans_q6;
set amos_trans;
where ano=2003 and t=6;
Data trans_q7;
set amos_trans;
where ano=2003 and t=7;
run;

Data trans_q1;
set trans_q1;
keep npc preco t cod_nut_ii_trans i ano;
Data trans_q2;
set trans_q2;
keep npc preco t cod_nut_ii_trans i ano;
Data trans_q3;
set trans_q3;
keep npc preco t cod_nut_ii_trans i ano;
Data trans_q4;
set trans_q4;
keep npc preco t cod_nut_ii_trans i ano;
Data trans_q5;
set trans_q5;
keep npc preco t cod_nut_ii_trans i ano;
Data trans_q6;
set trans_q6;
keep npc preco t cod_nut_ii_trans i ano;
Data trans_q7;
set trans_q7;
keep npc preco t cod_nut_ii_trans i ano;
run;

Data null;
file 'C:\Documents and Settings\Luis 
Pereira\My 
Documents\EstudoEmpirico\chamamacro.txt';
   do i= 1 to 1000; k=i;
var='%ext_am('||i||','||k||');'; put var; end;
run;

/*********************************************
*********************************************/
/** Estimação directa ao nível de NUTSIII ***/
/*********************************************
*********************************************/

%macro ext_am(i,k);

/* Extracção das amostras */

Data amostra;
merge hb.ppopul hb.pdims_229;
by estrato;
if ranuni(&i) <= m_h/npop then do;
output;
m_h=m_h - 1;
end;
npop=npop - 1;
```

```
drop mh m_h npop coef;
run;

/* Fim da extracção das amostras */
/* Dimensões amostrais efectivas por estrato
*/

proc means data=amostra nway noprint;
   var npc;
   class estrato;
   output out=resultn n=m;
run;

data resultn;
merge resultn hb.pdims_229;
by estrato;
if m=. then m=0;
run;

data resultn;
   set resultn;
   coef=npop/m; /* Este é o novo coeficiente
de extrapolação calculado
   com as dimensões populacionais da pseudo-
população e da amostra dela
   extraída */
   drop _type_ _freq_ m npop m_h mh;
run;

Proc sort data=amostra;
by estrato;
Proc sort data=resultn;
by estrato;
run;

Data amostra;
merge amostra (in=a) resultn (in=b);
by estrato;
if a=1 and b=1 then output;
run;

/* Fim das dimensões amostrais efectivas por
estrato */
/* Dimensões amostrais efectivas de cada
empresa */

proc sort data=amostra;
by npc;
run;

Data amostra_aux;
set amostra;
var1=1;
run;

proc means data=amostra_aux nway noprint;
class npc;
output out=resultemp n=n;
run;

data resultemp;
   set resultemp;
   drop _type_ _freq_;
run;

/* Fim das dimensões amostrais efectivas de
cada empresa */

/*******************************************/
/* Criação das pseudo-amostras de transacções
para cada trimestre */
/*******************************************/

Data amos_first;
set amostra;
by npc;
if first.npc;
run;

Data resultemp;
merge resultemp amos_first;
by npc;
run;

Data amos_transq1;
merge resultemp (in=a) trans_q1 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq2;
merge resultemp (in=a) trans_q2 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq3;
merge resultemp (in=a) trans_q3 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq4;
merge resultemp (in=a) trans_q4 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq5;
merge resultemp (in=a) trans_q5 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq6;
merge resultemp (in=a) trans_q6 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq7;
merge resultemp (in=a) trans_q7 (in=b);
by npc;
if a=1 and b=1 then output;
run;

Data s_transq1;
set amos_transq1;
do j=1 to n;
by npc;
output;
end;
Data s_transq2;
set amos_transq2;
do j=1 to n;
by npc;
output;
end;
Data s_transq3;
set amos_transq3;
do j=1 to n;
by npc;
output;
end;
Data s_transq4;
set amos_transq4;
do j=1 to n;
by npc;
output;
end;
Data s_transq5;
set amos_transq5;
do j=1 to n;
by npc;
output;
end;
Data s_transq6;
set amos_transq6;
do j=1 to n;
by npc;
output;
end;
Data s_transq7;
set amos_transq7;
do j=1 to n;
by npc;
output;
end;
run;

Data s_transq1;   /* Ficheiro com as
transacções do 1º trimestre */
set s_transq1;
drop n j;
Data s_transq2;
set s_transq2;
drop n j;
Data s_transq3;
set s_transq3;
drop n j;
Data s_transq4;
set s_transq4;
drop n j;
Data s_transq5;
set s_transq5;
drop n j;
Data s_transq6;
set s_transq6;
```

```
drop n j;
Data s_transq7;
set s_transq7;
drop n j;
run;

Data stotal_trans;
set s_transq1 s_transq2 s_transq3 s_transq4
s_transq5
s_transq6 s_transq7;
run;

/* Cálculo de estimativas directas por NUTS
III */

Proc sort data=stotal_trans out=stotal_trans;
by t estrato NPC i;
run;

ods trace on;
ods output domain analysis=state_stats;

Proc surveymeans data=stotal_trans mean cv var
nobs ncluster;
by t;   /* Para calcular por trimestre */
class i;   /* Define que esta variável é
categórica */
strata estrato;
cluster NPC;
weight coef;
var preco;
domain i;
run;

ods trace off;

Data est_dir_pseudo;
set Data&k;
keep i t mean var;

Data est_dir_pseudo;
set est_dir_pseudo;
simul=&k;
run;

Proc append
base=hb.estimat_directas_pseudo_229
data=est_dir_pseudo;
run;

DM 'CLEAR OUTPUT';
DM 'CLEAR LOG';
%mend;
%inc 'C:\Documents and Settings\Luis
Pereira\My
Documents\EstudoEmpirico\chamamacro.txt';

/*********************************************
*********************************************/
/* Estimação directa ao nível de NUTSII e
estimação sintética pelo Quociente */
/*********************************************
*********************************************/

%macro ext_am(i,k);

/* Extracção das amostras */

Data amostra;
merge hb.ppopul hb.pdims;
by estrato;
if ranuni(&i) <= m_h/npop then do;
output;
m_h=m_h - 1;
end;
npop=npop - 1;
drop mh m_h npop coef;
run;

/* Fim da extracção das amostras */
/* Dimensões amostrais efectivas por estrato
*/

proc means data=amostra nway noprint;
   var npc;
   class estrato;
   output out=resultn n=m;
run;
```

```
data resultn;
merge resultn hb.pdims;
by estrato;
if m=. then m=0;
run;

data resultn;
   set resultn;
   coef=npop/m; /* Este é o novo coeficiente
de extrapolação calculado
   com as dimensões populacionais da pseudo-
população e da amostra dela
   extraída */
   drop _type_ _freq_ m npop m_h mh;
run;

Proc sort data=amostra;
by estrato;
Proc sort data=resultn;
by estrato;
run;

Data amostra;
merge amostra (in=a) resultn (in=b);
by estrato;
if a=1 and b=1 then output;
run;

/* Fim das dimensões amostrais efectivas por
estrato */
/* Dimensões amostrais efectivas de cada
empresa */

proc sort data=amostra;
by npc;
run;

Data amostra_aux;
set amostra;
var1=1;
run;

proc means data=amostra_aux nway noprint;
class npc;
output out=resultemp n=n;
run;

data resultemp;
   set resultemp;
   drop _type_ _freq_;
run;

/* Fim das dimensões amostrais efectivas de
cada empresa */

/*******************************************/
/* Criação das pseudo-amostras de transacções
para cada trimestre */
/*******************************************/

Data amos_first;
set amostra;
by npc;
if first.npc;
run;

Data resultemp;
merge resultemp amos_first;
by npc;
run;

Data amos_transq1;
merge resultemp (in=a) trans_q1 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq2;
merge resultemp (in=a) trans_q2 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq3;
merge resultemp (in=a) trans_q3 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq4;
merge resultemp (in=a) trans_q4 (in=b);
by npc;
if a=1 and b=1 then output;
```

```
Data amos_transq5;
merge resultemp (in=a) trans_q5 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq6;
merge resultemp (in=a) trans_q6 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq7;
merge resultemp (in=a) trans_q7 (in=b);
by npc;
if a=1 and b=1 then output;
run;

Data s_transq1;
set amos_transq1;
do j=1 to n;
by npc;
output;
end;
Data s_transq2;
set amos_transq2;
do j=1 to n;
by npc;
output;
end;
Data s_transq3;
set amos_transq3;
do j=1 to n;
by npc;
output;
end;
Data s_transq4;
set amos_transq4;
do j=1 to n;
by npc;
output;
end;
Data s_transq5;
set amos_transq5;
do j=1 to n;
by npc;
output;
end;
Data s_transq6;
set amos_transq6;
do j=1 to n;
by npc;
output;
end;
Data s_transq7;
set amos_transq7;
do j=1 to n;
by npc;
output;
end;
run;

Data s_transq1;   /* Ficheiro com as
transacções do 1º trimestre */
set s_transq1;
drop n j;
Data s_transq2;
set s_transq2;
drop n j;
Data s_transq3;
set s_transq3;
drop n j;
Data s_transq4;
set s_transq4;
drop n j;
Data s_transq5;
set s_transq5;
drop n j;
Data s_transq6;
set s_transq6;
drop n j;
Data s_transq7;
set s_transq7;
drop n j;
run;

Data stotal_trans;
set s_transq1 s_transq2 s_transq3 s_transq4
s_transq5
s_transq6 s_transq7;
run;

/* Cálculo de estimativas directas por NUTS II
*/

Proc sort data=stotal_trans out=stotal_trans;
by t estrato NPC cod_nut_ii_trans;
run;

ods trace on;
ods output domain analysis=state_stats;

Proc surveymeans data=stotal_trans mean cv var
nobs ncluster;
by t;   /* Para calcular por trimestre */
class cod_nut_ii_trans;   /* Define que esta
variável é categórica */
strata estrato;
cluster NPC;
weight coef;
var preco;
domain cod_nut_ii_trans;
run;

ods trace off;

Data est_dir_pseudo;
set Data&k;
keep cod_nut_ii_trans t mean var n;

Data est_dir_pseudo;
set est_dir_pseudo;
simul=&k;
run;

Proc append
base=Hb.estimat_directasNUTSII_pseudo
data=est_dir_pseudo;
run;

/* Cálculo de estimativas sintéticas pelo
quociente */

Data est_sint_q;
merge est_dir_pseudo Hb.iabh_total_nutiii_ii;
by t cod_nut_ii_trans;
run;

Data est_sint_quoc;
set est_sint_q;
estQ=(x/x_nutii)*mean;
keep t i x x_nutii mean estQ var;
simul=&k;
run;

Proc append base=Hb.Est_sintQuoc_simulacao
data=est_sint_quoc;
run;

DM 'CLEAR OUTPUT';
DM 'CLEAR LOG';
%mend;
%inc 'C:\Documents and Settings\Luis
Pereira\My
Documents\EstudoEmpirico\chamamacro.txt';


/*********************************************
*********************************************/
/* Estimação directa ao nível de NUTSI **/
/*********************************************
*********************************************/

%macro ext_am(i,k);

/* Extracção das amostras */

Data amostra;
merge hb.ppopul hb.pdims;
by estrato;
if ranuni(&i) <= m_h/npop then do;
output;
m_h=m_h - 1;
end;
npop=npop - 1;
drop mh m_h npop coef;
run;

/* Fim da extracção das amostras */
```

```
/* Dimensões amostrais efectivas por estrato
*/

proc means data=amostra nway noprint;
   var npc;
   class estrato;
   output out=resultn n=m;
run;

data resultn;
merge resultn hb.pdims;
by estrato;
if m=. then m=0;
run;

data resultn;
   set resultn;
   coef=npop/m; /* Este é o novo coeficiente
de extrapolação calculado
   com as dimensões populacionais da pseudo-
população e da amostra dela
   extraída */
   drop _type_ _freq_ m npop m_h mh;
run;

Proc sort data=amostra;
by estrato;
Proc sort data=resultn;
by estrato;
run;

Data amostra;
merge amostra (in=a) resultn (in=b);
by estrato;
if a=1 and b=1 then output;
run;

/* Fim das dimensões amostrais efectivas por
estrato */
/* Dimensões amostrais efectivas de cada
empresa */

proc sort data=amostra;
by npc;
run;

Data amostra_aux;
set amostra;
var1=1;
run;

proc means data=amostra_aux nway noprint;
class npc;
output out=resultemp n=n;
run;

data resultemp;
   set resultemp;
   drop _type_ _freq_;
run;

/* Fim das dimensões amostrais efectivas de
cada empresa */

/*********************************************/
/* Criação das pseudo-amostras de transacções
para cada trimestre */
/*********************************************/

Data amos_first;
set amostra;
by npc;
if first.npc;
run;

Data resultemp;
merge resultemp amos_first;
by npc;
run;

Data amos_transq1;
merge resultemp (in=a) trans_q1 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq2;
merge resultemp (in=a) trans_q2 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq3;
merge resultemp (in=a) trans_q3 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq4;
merge resultemp (in=a) trans_q4 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq5;
merge resultemp (in=a) trans_q5 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq6;
merge resultemp (in=a) trans_q6 (in=b);
by npc;
if a=1 and b=1 then output;
Data amos_transq7;
merge resultemp (in=a) trans_q7 (in=b);
by npc;
if a=1 and b=1 then output;
run;

Data s_transq1;
set amos_transq1;
do j=1 to n;
by npc;
output;
end;
Data s_transq2;
set amos_transq2;
do j=1 to n;
by npc;
output;
end;
Data s_transq3;
set amos_transq3;
do j=1 to n;
by npc;
output;
end;
Data s_transq4;
set amos_transq4;
do j=1 to n;
by npc;
output;
end;
Data s_transq5;
set amos_transq5;
do j=1 to n;
by npc;
output;
end;
Data s_transq6;
set amos_transq6;
do j=1 to n;
by npc;
output;
end;
Data s_transq7;
set amos_transq7;
do j=1 to n;
by npc;
output;
end;
run;

Data s_transq1;   /* Ficheiro com as
transacções do 1º trimestre */
set s_transq1;
drop n j;
Data s_transq2;
set s_transq2;
drop n j;
Data s_transq3;
set s_transq3;
drop n j;
Data s_transq4;
set s_transq4;
drop n j;
Data s_transq5;
set s_transq5;
drop n j;
Data s_transq6;
set s_transq6;
drop n j;
Data s_transq7;
set s_transq7;
drop n j;
```

```
run;

Data stotal_trans;
set s_transq1 s_transq2 s_transq3 s_transq4
s_transq5
s_transq6 s_transq7;
run;

/* Cálculo de estimativas directas por NUTS I
*/

Proc sort data=stotal_trans out=stotal_trans;
by t estrato NPC cod_nut_ii_trans;
run;

Data stotal_trans1;
set stotal_trans;
cod_nut_i_trans='1';
run;

ods trace on;
ods output domain analysis=state_stats;

Proc surveymeans data=stotal_trans1 mean cv
var nobs ncluster;
by t;   /* Para calcular por trimestre */
class cod_nut_i_trans;   /* Define que esta
variável é categórica */
strata estrato;
cluster NPC;
weight coef;
var preco;
domain cod_nut_i_trans;
run;

ods trace off;

Data est_dir_pseudo1;
set Data&k;
keep cod_nut_i_trans t mean var n;

Data est_dir_pseudo1;
set est_dir_pseudo1;
simul=&k;
run;

Proc append
base=Hb.estimat_directasNUTSI_pseudo
data=est_dir_pseudo1;
run;

DM 'CLEAR OUTPUT';
DM 'CLEAR LOG';
%mend;
%inc 'C:\Documents and Settings\Luis
Pereira\My Documents\
EstudoEmpirico\chamamacro.txt';

/********************************************
********************************************/
/* Estimação sintética pela regressão
********/
/********************************************
********************************************/

Data Hb.Est_sintReg_simulacao;
set Hb.Est_fh_simulacao;
estR=e-v;
keep i y x b0 b1 estR t simul;
run;

/********************************************
********************************************/
/* Estimação combinada com pesos dependentes
dos dados ***********************************/
/********************************************
********************************************/

Proc sort data=Hb.Est_SintReg_simulacao
out=Est_SintReg_simulacao;
by simul t i;
Proc sort data=Hb.Estimat_directas_pseudo1
out=Estimat_directas_pseudo1;
by simul t i;
run;

Data Est_comb;
merge Est_SintReg_simulacao
Estimat_directas_pseudo1;
by simul t i;
keep simul t i y var estR;
run;

Data Est_comb;
set Est_comb;
pesos=1-var/((estR-y)*(estR-y));
run;

Data Hb.Est_comb_simulacao;
set Est_comb;
if pesos <0 then pesos1=0;
if pesos >1 then pesos1=1;
if 0<= pesos <= 1 then pesos1=pesos;
EstC=pesos1*y+(1-pesos1)*estR;
run;
```

Apêndice 18 – Estimação dos parâmetros de interesse com base em estimadores EBLUP no âmbito do estudo empírico por simulação *design-based*

FICHEIROS DE DADOS UTILIZADOS:

Estimat_directas_pseudo.sas7bdat – ficheiro com as estimativas directas dos parâmetros de interesse produzidas a partir de cada uma das 1000 amostras aleatórias independentes extraídas da pseudo-população.

W1.sas7bdat – ficheiro com os pesos espaciais.

O parâmetro de interesse é estimado utilizando os seguintes estimadores EBLUP:
- Estimador de Fay-Herriot;
- Estimador de Salvati;
- Estimador de Rao-Yu;
- Estimador de Luis Pereira.

Os resultados estão guardados nos seguintes ficheiros:
- Estimativas EBLUP de Fay-Herriot: **Est_FH_simulacao.sas7bdat**
- Estimativas EBLUP de Salvati: **Est_NS_simulacao.sas7bdat**
- Estimativas EBLUP de Rao-Yu: **Est_RY_simulacao.sas7bdat**
- Estimativas EBLUP de Luis Pereira: **Est_LP_simulacao.sas7bdat**

CÓDIGO SAS

```
/*********************************************
*********************************************/
/*********** MODELO DE FAY-HERRIOT **********/
/*********************************************
*********************************************/

/*É utilizado o modelo de Fay-Herriot e o EQMP
do EBLUP é estimado pela aproximação analítica
- proposta por Prasad-Rao. */

/* Dados: Hb.Estimat_directas_pseudo */
/* Os resultados estão guardados no ficheiro:
Hb.Est_FH_simulacao */

Proc sort data=Hb.Estimat_directas_pseudo
out=A5;
by simul t i;
run;

Data null;
file 'C:\Documents and Settings\Luis Pereira
\My Documents\EstudoEmpirico\chamamacro.txt';
   do i= 1 to 1000; k=i;
var='%ext_am('||i||','||k||');'; put var; end;
run;

%macro ext_am(i,k);

/*** t=1 ***/
Data A6;
set A5;
if simul=&i;
if t=1;
run;

Data iabh;
set a6;
keep i c x;
run;

Data ipth;
set a6;
keep i y;
run;

Data iabhipth1;
set a6;
keep i y x;
run;

/** Regressão de X (iabh) sobre Y (ipth) **/

proc reg data=iabhipth1 noprint;
        model y=x;
        output out=rg1  residual=r1;
run; quit;

Data rg1;
set rg1;
qr1=r1*r1;
Keep qr1;run;

Proc means data=rg1 sum noprint;
var qr1;
output out=rg1_ sum=sqr1; run;

Data rg1_;
set rg1_;
keep sqr1; run;

/** Cálculo da variância sigma2_u **/

Proc sql; create table iabh0 as select c, x, i
from iabh;

Data X;
set iabh0;
keep c x;
run;
Data Y;
set ipth;
keep y; run;

Proc iml;
use X; read all into X;
Ident=I(28);
IX=Ident - X*inv(t(X)*X)*t(X);
create IX from IX; append from IX; quit;

Proc iml;
use IX; read all into IX;
DiagIX=vecdiag(IX);
create DiagIX from DiagIX; append from DiagIX;
quit;

Proc means data=DiagIX sum noprint;
var col1;
output out=DiagIX_ sum=scol1;
run;

Data DiagIX_;
set DiagIX_;
keep scol1; run;

Proc iml;
use rg1_; read all into rg1_;
use DiagIX_; read all into DiagIX_;
varv=rg1_||DiagIX_;
create varv from varv; append from varv; quit;

Data varv;
set varv;
varv=(col1-col2)/(28-2);
keep varv; run;

/* Truncagem a zero das variâncias sigma2 */

Data varv;
```

```
set varv;
varv= max( varv , 0);
run;

/*Termina aqui o cálculo da variância sigma2*/
/* Construção da matriz de covariâncias (SIGMA
= MatV) */

Proc iml;
use varv; read all into varv;
Ident=I(28);
Sigma=varv*Ident;
create Sigma from Sigma; append from Sigma;
quit;

Data A4;
set A6;
keep var; run;

Proc iml;
use varv; read all into varv;
use A4; read all into A4;
par =varv//A4;
create par from par; append from par; quit;

Data par;
set par;
rename col1=est;
run;

/*********************************************
*********************************************/
/*** ESTIMAÇÃO DO MODELO DE FAY-HERRIOT *****/
/*********************************************
*********************************************/

Proc sort data=iabhipth1 out= cron;
by i; run;

Proc mixed data=cron MMEqSol noinfo noitprint
noprofile noclprint;
        class i;
        model y=x / solution;
        random i / solution G V;
        repeated /group=i;
        parms /parmsdata=par noiter;
make 'solutionf' out=beta;
make 'solutionr' out=u;
make 'Fitstatistics' out=aic;
make 'G' out=Mat_Gaux;
make 'V' out=Mat_V;
run; quit;

/*********************************************/
/* CÁLCULO DAS ESTIMATIVAS DO PREÇO MÉDIO DE
TRANSACÇÃO - NUT III */
/*********************************************/

Data id;
do id=1 to 2;
output; end; run;

Data beta;
merge id beta;
run;

Data beta0;
set beta;
if id=1;
rename estimate=b0;
keep estimate;
run;

Data beta1;
set beta;
if id=2;
rename estimate=b1;
keep estimate;
run;

Data betan;
merge beta0 beta1;
_type_=0;
run;

/****** Efeitos Aleatórios: v ******/

Data u;
```

```
set u;
if estimate='.' then estimate=0;
run;

Data v;
set u;
if Effect='i';
_type_=0;
rename estimate=v;
keep i estimate _type_;
run;

Data beta_vu;
merge betan v;
by _type_;
run;

/** Estimação do preço médio de transacção em
cada NUTIII **/

Data cron;
merge cron beta_vu;
by i; run;

Data crono_resf;
set cron;
e = b0+x*b1 + v;
t=1;
keep i _type_ y x b0 b1 v e t;
run;

/*********************************************/
/* Cálculo das estimativas do EQM do EBLUP **/
/*********************************************/

/*** MATRIZ G ***/

Data MatG;
set Mat_Gaux;
drop row effect i;
run;

/*** MATRIZ V ***/

Data MatV;
set Mat_v;
drop Index row;
run;

/**************************/
/****** Cálculo do g1 ******/
/**************************/

Proc iml;
use MatV; read all into MatV;
use MatG; read all into MatG;
G1_aux=MatG - MatG*inv(MatV)*MatG;
create G1_aux from G1_aux;
append from G1_aux;
quit;

Proc iml;
use G1_aux;
read all into G1_aux;
G1=vecdiag(G1_aux);
create G1 from G1;
append from G1;
quit;

Data g1;
set g1;
rename col1=g1;
run;

/* Verificação de G1 - pode ser feito com uma
calculadora*/

/**************************/
/****** Cálculo do g2 ******/
/**************************/

Proc iml;
use X; read all into X;
use MatV; read all into MatV;
use MatG; read all into MatG;
A=I(28);
```

```
G2_aux=t(t(X) -
t(X)*inv(MatV)*MatG*A)*inv(t(X)*inv(MatV)*X)*(
t(X) - t(X)*inv(MatV)*MatG*A);
create G2_aux from G2_aux;
append from G2_aux;
quit;

Proc iml;
use G2_aux;
read all into G2_aux;
G2=vecdiag(G2_aux);
create G2 from G2;
append from G2;
quit;

Data g2;
set g2;
rename col1=g2;
run;

/***************************/
/****** Cálculo do g3 ******/
/***************************/

/* Cálculo da variância assimptótica do
estimador da comp. de variância */

Proc iml;
use Matv; read all into Matv;
diagV=vecdiag(Matv);
create diagV from diagV; append from diagV;
quit;

Proc iml;
use Matv; read all into Matv;
MatV2=Matv*Matv;
diagV2=vecdiag(Matv2);  /* Quadrado da
diagonal da matriz V */
Vect1=J(1,28,1);      /* Vector 1*28 de uns */
varassimp=2*(Vect1*diagV2)/(28*28);  /*
Variância assimptótica */
create varassimp from varassimp; append from
varassimp; quit;

Proc iml;
use Matv; read all into Matv;
use Matg; read all into Matg;
Matr=Matv-Matg;    /* Matriz R */
create Matr from Matr; append from Matr; quit;

Proc iml;
use Matv; read all into Matv;
use Matr; read all into Matr;
use varassimp; read all into varassimp;
MatV3=Matv*Matv*Matv;
MatR2=Matr*Matr;
diagR2=vecdiag(MatR2);  /* Quadrado da
diagonal da matriz R */
g3=inv(MatV3)*diagR2*varassimp;
create g3 from g3; append from g3; quit;

Data g3;
set g3;
rename col1=g3;
run;

/* **** Estimativa do EQMP do EBLUP *****/

Data Estimat_FH;
merge crono_resf g1 g2 g3;
EQM_EBLUP= G1 + G2 + 2*G3;
CV=sqrt(EQM_EBLUP)/e;
simul=&i;
run;

Proc append base=HB.Est_FH_simulacao
data=Estimat_FH;
run;

/*** t=2 ***/ (…)
/*** t=3 ***/ (…)
/*** t=4 ***/ (…)
/*** t=5 ***/ (…)
/*** t=6 ***/ (…)
/*** t=7 ***/ (…)

%mend;

%inc 'C:\Documents and Settings\Luis Pereira
\My Documents\EstudoEmpirico\chamamacro.txt';

/*******************************************/
/*FIM DA SIMULAÇÃO COM ESTIMADOR EBLUP DE FH*/
/*******************************************/

/********************************************
********************************************/
/************* MODELO DE SALVATI ************/
/********************************************
********************************************/

/*É utilizado o modelo espacial de Salvati e o
EQMP do EBLUP é estimado pela aproximação
analítica.
Contudo, admite-se que o coeficiente de
associação espacial é conhecido e que a
componente de variância é estimada pelo método
dos momentos. */

/* Dados: Hb.Estimat_directas_pseudo */
/* Os resultados estão guardados no ficheiro:
Hb.Est_NS_simulacao */

Proc sort data=Hb.Estimat_directas_pseudo
out=A5;
by simul t i;
run;

Data null;
file 'C:\Documents and Settings\Luis Pereira
\My Documents\EstudoEmpirico\chamamacro.txt';
   do i= 1 to 1000; k=i;
var='%ext_am('||i||','||k||');'; put var; end;
run;

%macro ext_am(i,k);

/*** t=1 ***/

Data A6;
set A5;
if simul=&i;
if t=1; run;

/* Dependência espacial */

Data dataphi;
phi=0.293;
run;

/* Matriz B=(I-phi*W)'*(I-phi*W) */

Data W;
set Hb.W1;   /* Matriz de distâncias */
run;

Proc iml;
use W; read all into W;
use dataphi; read all into dataphi;
Ident=I(28);
B = t(Ident - dataphi*W)*(Ident - dataphi*W);
create B from B; append from B; quit;

Proc iml;
use B; read all into B;
Binv=inv(B);   /* Inversa de B */
create Binv from Binv; append from Binv; quit;

Data iabh;
set a6;
keep i c x; run;

Data ipth;
set a6;
keep i y; run;

Data iabhipth1;
set a6;
keep i y x; run;

/** Cálculo da matriz Z(1) **/

Data ipth1;
set ipth;
keep y;
run;
```

```
Proc iml;
use ipth1; read all into ipth1;
use Dataphi; read all into Dataphi;
use W; read all into W;
Ident=I(28);
z1=(Ident - Dataphi*W)*ipth1;
create z1 from z1; append from z1; quit;

/** Cálculo da matriz H(1) **/

Proc sql; create table iabh1 as select c, x
from iabh;

Proc iml;
use iabh1; read all into iabh1;
use Dataphi; read all into Dataphi;
use W; read all into W;
Ident=I(28);
H1=(Ident - Dataphi*W)*iabh1;
create H1 from H1; append from H1; quit;

/** Regressão de Z(1) sobre H(1) **/

Data Z1;
set Z1;
rename col1=col0;
run;

Data reg;
merge Z1 H1;
run;

proc reg data=reg noprint;
        model col0=col1 col2 / noint;
        output out=rg1  residual=r1;
run; quit;

Data rg1;
set rg1;
qr1=r1*r1;
Keep qr1;
run;

Proc means data=rg1 sum noprint;
var qr1;
output out=rg1_ sum=sqr1;
run;

Data rg1_;
set rg1_;
keep sqr1; run;

/** Cálculo da variância sigma2_u **/

Proc iml;
use H1; read all into H1;
Ident=I(28);
PH1=Ident - H1*ginv(t(H1)*H1)*t(H1); /*
ginv(A) is the Moore-Penrose generalized
inverse matrix of A */
create PH1 from PH1; append from PH1; quit; /*
Hs1 é a matriz H*(1) */

Data s1;
set a6;
keep var i;
run;

Proc sort data=s1;
by i; run;

Data s1;
set s1;
keep var;
run;
Proc iml;
use s1; read all into s1;
R=diag(s1);
create R from R; append from R; quit;

Proc iml;
use H1; read all into H1;
HH=H1*t(H1);
eval=eigval(HH);
create eval from eval; append from eval; quit;

Data id;
do id=1 to 196;
output; end;
run;

Data eval;
merge id eval; run;

Data new;
set eval;
by id;
retain count 0;
if first.id then count=0;
if col1 > 0.000001 then count = count + 1;
run;

Proc means data=new sum noprint;
var count;
output out=caracH1 sum=caracH1;
run;

Data caracH1;
set caracH1;
keep caracH1; run;

Proc iml;
use Dataphi; read all into Dataphi;
use W; read all into W;
use rg1_; read all into rg1_;
use PH1; read all into PH1;
use R; read all into R;
use caracH1; read all into caracH1;
Ident=I(28);
varu= (rg1_ -  trace(PH1*(Ident -
Dataphi*W)*R*t(Ident - Dataphi*W) ))*(1/(28 -
caracH1));
create varu from varu; append from varu; quit;

/* Truncagem a zero das variâncias sigma2_u e
sigma2_v */

Data varu;
set varu;
sigma2u= max( col1 , 0);
keep sigma2u;
run;

/*Termina aqui o cálculo da variância sigma2*/
/*** Matriz G ***/

Proc iml;
use varu; read all into varu;
use Binv; read all into Binv;
Mat_GG=varu*Binv;
create Mat_GG from Mat_GG; append from Mat_GG;
quit;

Proc iml;
row=t(do(1,28,1));
create row from row; append from row; quit;

Data row;
set row;
rename col1=row;
run;

Data Mat_GG;
merge row Mat_GG;
run;

Data par;
set s1;
rename var=est;
run;

/*********************************************
*********************************************/
/** ESTIMAÇÃO DO MODELO DE NICOLA SALVATI ***/
/*********************************************
*********************************************/

Proc sort data=iabhipth1 out= cron;
by i; run;

/* O Proc mixed só permite incluir uma matriz
G, independentemente no número de RANDOM
definidos. A matriz G definida tem que ser a
matriz completa para todo o modelo */
```

```
Proc mixed data=cron MMEqSol noprofile
noclprint;
        class i;
        model y=x / solution;
        random i /type=un Gdata=Mat_GG G V S;
        repeated /group=i;
        parms /parmsdata=par noiter;
make 'solutionf' out=beta;
make 'solutionr' out=u;
make 'Fitstatistics' out=aic;
make 'G' out=Mat_Gaux;
make 'V' out=Mat_V;
run; quit;

/*********************************************/
/* CÁLCULO DAS ESTIMATIVAS DO PREÇO MÉDIO DE
TRANSACÇÃO - NUT III */
/*********************************************/

Data id;
do id=1 to 2;
output; end;
run;

Data beta;
merge id beta;
run;

Data beta0;
set beta;
if id=1;
rename estimate=b0;
keep estimate;
run;

Data beta1;
set beta;
if id=2;
rename estimate=b1;
keep estimate;
run;

Data betan;
merge beta0 beta1;
_type_=0;
run;

/****** Efeitos Aleatórios: v ******/

Data u;
set u;
if estimate='.' then estimate=0;
run;

Data v;
set u;
if Effect='i';
_type_=0;
rename estimate=v;
keep i estimate _type_;
run;

Data beta_vu;
merge betan v;
by _type_;
run;

/** Estimação do preço médio de transacção em
cada NUTIII **/

Data cron;
merge cron beta_vu;
by i;
run;

Data crono_resf;
set cron;
e = b0+x*b1 + v;
t=1;
keep i _type_ y x b0 b1 v e t;
run;

/*********************************************/
/** Cálculo das estimativas do EQM do EBLUP */
/*********************************************/

/*** MATRIZ G ***/

Data MatG;
set Mat_Gaux;
drop row effect i;
run;

/*** MATRIZ V ***/

Data MatV;
set Mat_v;
drop Index row;
run;

/***************************/
/****** Cálculo do g1 ******/
/***************************/

Proc iml;
use MatV; read all into MatV;
use MatG; read all into MatG;
G1_aux=MatG - MatG*inv(MatV)*MatG;
create G1_aux from G1_aux;
append from G1_aux;
quit;

Proc iml;
use G1_aux;
read all into G1_aux;
G1=vecdiag(G1_aux);
create G1 from G1;
append from G1;
quit;

Data g1;
set g1;
rename col1=g1;
run;

/***************************/
/****** Cálculo do g2 ******/
/***************************/

Data X;
set iabh1;
run;

Proc iml;
use X; read all into X;
use MatV; read all into MatV;
use MatG; read all into MatG;
A=I(28);
G2_aux=t(t(X) -
t(X)*inv(MatV)*MatG*A)*inv(t(X)*inv(MatV)*X)*(
t(X) - t(X)*inv(MatV)*MatG*A);
create G2_aux from G2_aux;
append from G2_aux;
quit;

Proc iml;
use G2_aux;
read all into G2_aux;
G2=vecdiag(G2_aux);
create G2 from G2;
append from G2;
quit;

Data g2;
set g2;
rename col1=g2;
run;

/***************************/
/****** Cálculo do g3 ******/
/***************************/

/* Cálculo da variância assimptótica do
estimador da comp. de variância */

Proc iml;
use Matv; read all into Matv;
use Binv; read all into Binv;
use varu; read all into varu;
A=I(28);
g3_=A*(Binv -
varu*Binv*inv(MatV)*Binv)*inv(MatV)*(Binv -
varu*Binv*inv(MatV)*Binv)*A;
g3_aux=vecdiag(g3_);
create g3_aux from g3_aux; append from g3_aux;
quit;
```

```
/* MatC_1 */

Proc iml;
use dataphi; read all into dataphi;
use W; read all into W;
Ident=I(28);
C_1=(Ident - dataphi*W);
create C_1 from C_1; append from C_1; quit;

/* MatC1 */

Proc iml;
use C_1; read all into C_1;
use X; read all into X;
Ident=I(28);
MatC1=t(C_1)* (Ident - C_1*X*
ginv(t(X)*t(C_1)*C_1*X) *t(X)*t(C_1)) *C_1;
create MatC1 from MatC1; append from MatC1;
quit;

/* varu = k1* a'C1a + k2 */

Proc iml;
use caracH1; read all into caracH1;
k1= 1/(28 - caracH1);
create k1 from k1; append from k1; quit;

Proc iml;
use MatC1; read all into MatC1;
use MatV; read all into MatV;
use k1; read all into k1;
Sigma11=2*k1*k1* trace(MatC1 * MatV * MatC1 *
MatV);
create Sigma11 from Sigma11; append from
Sigma11; quit;

/********/

Proc iml;
use g3_aux; read all into g3_aux;
use Sigma11; read all into Sigma11;
g3=Sigma11*g3_aux;
create g3 from g3; append from g3; quit;

Data g3;
set g3;
rename col1=g3;
run;

/* **** Estimativa do EQMP do EBLUP *****/

Data Estimat_NS;
merge crono_resf g1 g2 g3;
EQM_EBLUP= G1 + G2 + 2*G3;
CV=sqrt(EQM_EBLUP)/e;
simul=&i;
run;

Proc append base=HB.Est_NS_simulacao
data=Estimat_NS;
run;

/*** t=2 ***/ (…)
/*** t=3 ***/ (…)
/*** t=4 ***/ (…)
/*** t=5 ***/ (…)
/*** t=6 ***/ (…)
/*** t=7 ***/ (…)

%mend;
%inc 'C:\Documents and Settings\Luis Pereira
\My Documents\EstudoEmpirico\chamamacro.txt';

/*******************************************/
/*FIM DA SIMULAÇÃO COM ESTIMADOR EBLUP DE NS*/
/*******************************************/

/*******************************************
*******************************************/
/************* MODELO DE RAO-YU *************/
/*******************************************
*******************************************/

/*É utilizado o modelo de Rao-Yu e o EQMP do
EBLUP é estimado pela aproximação analítica.*/

/* Dados: Hb.Estimat_directas_pseudo */

/* Os resultados estão guardados no ficheiro:
Hb.Est_RY_simulacao */

Proc sort data=Hb.Estimat_directas_pseudo
out=A5;
by simul i t;
run;

Data null;
file 'C:\Documents and Settings\Luis Pereira
\My Documents\EstudoEmpirico\chamamacro.txt';
   do i= 1 to 1000; k=i;
var='%ext_am('||i||','||k||');'; put var; end;
run;

%macro ext_am(i,k);

Data A6;
set A5;
if simul=&i;
run;

/* *** */

Data dataro;
ro=0.37;
run;

Data dataro2;
set dataro;
ro2= sqrt(1-ro*ro);
keep ro2;
run;

Proc iml;
use dataro;
read all into dataro;
use dataro2;
read all into dataro2;
I1={1 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I2={0 0 0 0 0 0 0, 0 1 0 0 0 0 0, 0 0 1 0 0 0
0, 0 0 0 1 0 0 0, 0 0 0 0 1 0 0, 0 0 0 0 0 1
0, 0 0 0 0 0 0 1};
I4={0 0 0 0 0 0 0, 1 0 0 0 0 0 0, 0 1 0 0 0 0
0, 0 0 1 0 0 0 0, 0 0 0 1 0 0 0, 0 0 0 0 1 0
0, 0 0 0 0 0 1 0};
P = dataro2*I1 + I2 - dataro*I4;
create P from P;
append from P;
quit;

Proc iml;
use dataro;
read all into dataro;
use dataro2;
read all into dataro2;
C1={1, 0, 0, 0, 0, 0, 0};
C2={0, 1, 1, 1, 1, 1, 1};
f = dataro2*C1 + C2 - dataro*C2;
create f from f;
append from f;
quit;

Proc iml;
use f;
read all into f;
c=f`*f;
create c from c;
append from c;
quit;

Proc iml;
use f;
read all into f;
use c;
read all into c;
D=(1/c)*f*f`;
create D from D;
append from D;
quit;

/* *** */

Data iabh;
set a6;
keep i t c x; run;
```

```
Data ipth;
set a6;
keep i t y; run;

Data iabhipth1;
set a6;
keep i t y x; run;

/** Preparação do ficheiro das observações da
variável de interesse **/

Proc sort data=ipth;
by i t;
run;

Data tetad1; set ipth; where i= 1; keep y;
Data tetad2; set ipth; where i= 2; keep y;
Data tetad3; set ipth; where i= 3; keep y;
Data tetad4; set ipth; where i= 4; keep y;
Data tetad5; set ipth; where i= 5; keep y;
Data tetad6; set ipth; where i= 6; keep y;
Data tetad7; set ipth; where i= 7; keep y;
Data tetad8; set ipth; where i= 8; keep y;
Data tetad9; set ipth; where i= 9; keep y;
Data tetad10; set ipth; where i= 10; keep y;
Data tetad11; set ipth; where i= 11; keep y;
Data tetad12; set ipth; where i= 12; keep y;
Data tetad13; set ipth; where i= 13; keep y;
Data tetad14; set ipth; where i= 14; keep y;
Data tetad15; set ipth; where i= 15; keep y;
Data tetad16; set ipth; where i= 16; keep y;
Data tetad17; set ipth; where i= 17; keep y;
Data tetad18; set ipth; where i= 18; keep y;
Data tetad19; set ipth; where i= 19; keep y;
Data tetad20; set ipth; where i= 20; keep y;
Data tetad21; set ipth; where i= 21; keep y;
Data tetad22; set ipth; where i= 22; keep y;
Data tetad23; set ipth; where i= 23; keep y;
Data tetad24; set ipth; where i= 24; keep y;
Data tetad25; set ipth; where i= 25; keep y;
Data tetad26; set ipth; where i= 26; keep y;
Data tetad27; set ipth; where i= 27; keep y;
Data tetad28; set ipth; where i= 28; keep y;
run;
/** Preparação do ficheiro das observações da
variável auxiliar **/

Proc sort data=iabh;
by i t; run;

Proc sql; create table iabh0 as select c, x,
i, t from iabh;

Data xd1; set iabh0; where i= 1; keep c x;
Data xd2; set iabh0; where i= 2; keep c x;
Data xd3; set iabh0; where i= 3; keep c x;
Data xd4; set iabh0; where i= 4; keep c x;
Data xd5; set iabh0; where i= 5; keep c x;
Data xd6; set iabh0; where i= 6; keep c x;
Data xd7; set iabh0; where i= 7; keep c x;
Data xd8; set iabh0; where i= 8; keep c x;
Data xd9; set iabh0; where i= 9; keep c x;
Data xd10; set iabh0; where i= 10; keep c x;
Data xd11; set iabh0; where i= 11; keep c x;
Data xd12; set iabh0; where i= 12; keep c x;
Data xd13; set iabh0; where i= 13; keep c x;
Data xd14; set iabh0; where i= 14; keep c x;
Data xd15; set iabh0; where i= 15; keep c x;
Data xd16; set iabh0; where i= 16; keep c x;
Data xd17; set iabh0; where i= 17; keep c x;
Data xd18; set iabh0; where i= 18; keep c x;
Data xd19; set iabh0; where i= 19; keep c x;
Data xd20; set iabh0; where i= 20; keep c x;
Data xd21; set iabh0; where i= 21; keep c x;
Data xd22; set iabh0; where i= 22; keep c x;
Data xd23; set iabh0; where i= 23; keep c x;
Data xd24; set iabh0; where i= 24; keep c x;
Data xd25; set iabh0; where i= 25; keep c x;
Data xd26; set iabh0; where i= 26; keep c x;
Data xd27; set iabh0; where i= 27; keep c x;
Data xd28; set iabh0; where i= 28; keep c x;
run;

/** Cálculo da matriz Z(1) **/

Proc iml;
use tetad1; read all into tetad1;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd1=(Ident - D)*P*tetad1;
create zd1 from zd1; append from zd1; quit;
Proc iml;
use tetad2; read all into tetad2;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd2=(Ident - D)*P*tetad2;
create zd2 from zd2; append from zd2; quit;
Proc iml;
use tetad3; read all into tetad3;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd3=(Ident - D)*P*tetad3;
create zd3 from zd3; append from zd3; quit;
Proc iml;
use tetad4; read all into tetad4;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd4=(Ident - D)*P*tetad4;
create zd4 from zd4; append from zd4; quit;
Proc iml;
use tetad5; read all into tetad5;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd5=(Ident - D)*P*tetad5;
create zd5 from zd5; append from zd5; quit;
Proc iml;
use tetad6; read all into tetad6;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd6=(Ident - D)*P*tetad6;
create zd6 from zd6; append from zd6; quit;
Proc iml;
use tetad7; read all into tetad7;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd7=(Ident - D)*P*tetad7;
create zd7 from zd7; append from zd7; quit;
Proc iml;
use tetad8; read all into tetad8;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd8=(Ident - D)*P*tetad8;
create zd8 from zd8; append from zd8; quit;
Proc iml;
use tetad9; read all into tetad9;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd9=(Ident - D)*P*tetad9;
create zd9 from zd9; append from zd9; quit;
Proc iml;
use tetad10; read all into tetad10;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd10=(Ident - D)*P*tetad10;
create zd10 from zd10; append from zd10; quit;
Proc iml;
use tetad11; read all into tetad11;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd11=(Ident - D)*P*tetad11;
create zd11 from zd11; append from zd11; quit;
Proc iml;
use tetad12; read all into tetad12;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd12=(Ident - D)*P*tetad12;
create zd12 from zd12; append from zd12; quit;
Proc iml;
use tetad13; read all into tetad13;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd13=(Ident - D)*P*tetad13;
```

```
create zd13 from zd13; append from zd13; quit;
Proc iml;
use tetad14; read all into tetad14;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd14=(Ident - D)*P*tetad14;
create zd14 from zd14; append from zd14; quit;
Proc iml;
use tetad15; read all into tetad15;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd15=(Ident - D)*P*tetad15;
create zd15 from zd15; append from zd15; quit;
Proc iml;
use tetad16; read all into tetad16;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd16=(Ident - D)*P*tetad16;
create zd16 from zd16; append from zd16; quit;
Proc iml;
use tetad17; read all into tetad17;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd17=(Ident - D)*P*tetad17;
create zd17 from zd17; append from zd17; quit;
Proc iml;
use tetad18; read all into tetad18;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd18=(Ident - D)*P*tetad18;
create zd18 from zd18; append from zd18; quit;
Proc iml;
use tetad19; read all into tetad19;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd19=(Ident - D)*P*tetad19;
create zd19 from zd19; append from zd19; quit;
Proc iml;
use tetad20; read all into tetad20;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd20=(Ident - D)*P*tetad20;
create zd20 from zd20; append from zd20; quit;
Proc iml;
use tetad21; read all into tetad21;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd21=(Ident - D)*P*tetad21;
create zd21 from zd21; append from zd21; quit;
Proc iml;
use tetad22; read all into tetad22;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd22=(Ident - D)*P*tetad22;
create zd22 from zd22; append from zd22; quit;
Proc iml;
use tetad23; read all into tetad23;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd23=(Ident - D)*P*tetad23;
create zd23 from zd23; append from zd23; quit;
Proc iml;
use tetad24; read all into tetad24;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd24=(Ident - D)*P*tetad24;
create zd24 from zd24; append from zd24; quit;
Proc iml;
use tetad25; read all into tetad25;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd25=(Ident - D)*P*tetad25;
create zd25 from zd25; append from zd25; quit;
Proc iml;
use tetad26; read all into tetad26;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd26=(Ident - D)*P*tetad26;
create zd26 from zd26; append from zd26; quit;
Proc iml;
use tetad27; read all into tetad27;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd27=(Ident - D)*P*tetad27;
create zd27 from zd27; append from zd27; quit;
Proc iml;
use tetad28; read all into tetad28;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd28=(Ident - D)*P*tetad28;
create zd28 from zd28; append from zd28; quit;

Data Z1;
set zd1 zd2 zd3 zd4 zd5 zd6 zd7 zd8 zd9 zd10
zd11 zd12 zd13 zd14 zd15
zd16 zd17 zd18 zd19 zd20 zd21 zd22 zd23 zd24
zd25 zd26 zd27 zd28;
run;

/** Cálculo da matriz H(1) **/

Proc iml;
use xd1; read all into xd1;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd1=(Ident - D)*P*xd1;
create hd1 from hd1; append from hd1; quit;
Proc iml;
use xd2; read all into xd2;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd2=(Ident - D)*P*xd2;
create hd2 from hd2; append from hd2; quit;
Proc iml;
use xd3; read all into xd3;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd3=(Ident - D)*P*xd3;
create hd3 from hd3; append from hd3; quit;
Proc iml;
use xd4; read all into xd4;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd4=(Ident - D)*P*xd4;
create hd4 from hd4; append from hd4; quit;
Proc iml;
use xd5; read all into xd5;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd5=(Ident - D)*P*xd5;
create hd5 from hd5; append from hd5; quit;
Proc iml;
use xd6; read all into xd6;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd6=(Ident - D)*P*xd6;
create hd6 from hd6; append from hd6; quit;
Proc iml;
use xd7; read all into xd7;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd7=(Ident - D)*P*xd7;
create hd7 from hd7; append from hd7; quit;
Proc iml;
use xd8; read all into xd8;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd8=(Ident - D)*P*xd8;
create hd8 from hd8; append from hd8; quit;
Proc iml;
use xd9; read all into xd9;
use P; read all into P;
use D; read all into D;
```

```
Ident=I(7);
hd9=(Ident - D)*P*xd9;
create hd9 from hd9; append from hd9; quit;
Proc iml;
use xd10; read all into xd10;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd10=(Ident - D)*P*xd10;
create hd10 from hd10; append from hd10; quit;
Proc iml;
use xd11; read all into xd11;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd11=(Ident - D)*P*xd11;
create hd11 from hd11; append from hd11; quit;
Proc iml;
use xd12; read all into xd12;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd12=(Ident - D)*P*xd12;
create hd12 from hd12; append from hd12; quit;
Proc iml;
use xd13; read all into xd13;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd13=(Ident - D)*P*xd13;
create hd13 from hd13; append from hd13; quit;
Proc iml;
use xd14; read all into xd14;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd14=(Ident - D)*P*xd14;
create hd14 from hd14; append from hd14; quit;
Proc iml;
use xd15; read all into xd15;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd15=(Ident - D)*P*xd15;
create hd15 from hd15; append from hd15; quit;
Proc iml;
use xd16; read all into xd16;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd16=(Ident - D)*P*xd16;
create hd16 from hd16; append from hd16; quit;
Proc iml;
use xd17; read all into xd17;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd17=(Ident - D)*P*xd17;
create hd17 from hd17; append from hd17; quit;
Proc iml;
use xd18; read all into xd18;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd18=(Ident - D)*P*xd18;
create hd18 from hd18; append from hd18; quit;
Proc iml;
use xd19; read all into xd19;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd19=(Ident - D)*P*xd19;
create hd19 from hd19; append from hd19; quit;
Proc iml;
use xd20; read all into xd20;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd20=(Ident - D)*P*xd20;
create hd20 from hd20; append from hd20; quit;
Proc iml;
use xd21; read all into xd21;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd21=(Ident - D)*P*xd21;
create hd21 from hd21; append from hd21; quit;
Proc iml;
use xd22; read all into xd22;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd22=(Ident - D)*P*xd22;
create hd22 from hd22; append from hd22; quit;
Proc iml;
use xd23; read all into xd23;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd23=(Ident - D)*P*xd23;
create hd23 from hd23; append from hd23; quit;
Proc iml;
use xd24; read all into xd24;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd24=(Ident - D)*P*xd24;
create hd24 from hd24; append from hd24; quit;
Proc iml;
use xd25; read all into xd25;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd25=(Ident - D)*P*xd25;
create hd25 from hd25; append from hd25; quit;
Proc iml;
use xd26; read all into xd26;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd26=(Ident - D)*P*xd26;
create hd26 from hd26; append from hd26; quit;
Proc iml;
use xd27; read all into xd27;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd27=(Ident - D)*P*xd27;
create hd27 from hd27; append from hd27; quit;
Proc iml;
use xd28; read all into xd28;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd28=(Ident - D)*P*xd28;
create hd28 from hd28; append from hd28; quit;

Data H1;
set hd1 hd2 hd3 hd4 hd5 hd6 hd7 hd8 hd9 hd10
hd11 hd12 hd13 hd14 hd15
hd16 hd17 hd18 hd19 hd20 hd21 hd22 hd23 hd24
hd25 hd26 hd27 hd28;
run;

/** Regressão de Z(1) sobre H(1) **/

Data Z1;
set Z1;
rename col1=col0;
run;

Data reg;
merge Z1 H1; run;

proc reg data=reg noprint;
        model col0=col1 col2 / noint;
        output out=rg1  residual=r1;
run; quit;

Data rg1;
set rg1;
qr1=r1*r1;
Keep qr1; run;

Proc means data=rg1 sum noprint;
var qr1;
output out=rg1_ sum=sqr1;
run;

Data rg1_;
set rg1_;
keep sqr1; run;

/** Cálculo da variância sigma2_u **/

Proc iml;
```

```
use H1; read all into H1;
Hs1=H1*ginv(t(H1)*H1)*t(H1); /* ginv(A) is the
Moore-Penrose generalized inverse matrix */
create Hs1 from Hs1; append from Hs1; quit; /*
Hs1 é a matriz H*(1) */

Proc iml;
use D; read all into D;
Ident=I(7);
ID=Ident - D; /* Matriz I - D */
create ID from ID; append from ID; quit;

Proc iml;
use ID; read all into ID;
blockID1=BLOCK(ID,ID,ID,ID,ID,ID,ID,ID,ID,ID,I
D,ID,ID,ID);
create blockID1 from blockID1; append from
blockID1; quit;
Proc iml;
use blockID1; read all into blockID1;
blockID=BLOCK(blockID1,blockID1);
create blockID from blockID; append from
blockID; quit;

Data s1;
set a6;
keep var t i;
run;

Proc sort data=s1;
by i t; run;

Data Sigma_1;
set s1;
if i=1;
keep var;
run;
Proc iml;
use Sigma_1; read all into Sigma_1;
Sigma1=diag(Sigma_1);
create Sigma1 from Sigma1; append from Sigma1;
quit;
Proc iml;
use P; read all into P;
use Sigma1; read all into Sigma1;
P1=P*Sigma1*t(P);
create P1 from P1; append from P1; quit;

Data Sigma_2;
set s1;
if i=2;
keep var;
run;
Proc iml;
use Sigma_2; read all into Sigma_2;
Sigma2=diag(Sigma_2);
create Sigma2 from Sigma2; append from Sigma2;
quit;
Proc iml;
use P; read all into P;
use Sigma2; read all into Sigma2;
P2=P*Sigma2*t(P);
create P2 from P2; append from P2; quit;

Data Sigma_3;
set s1;
if i=3;
keep var;
run;
Proc iml;
use Sigma_3; read all into Sigma_3;
Sigma3=diag(Sigma_3);
create Sigma3 from Sigma3; append from Sigma3;
quit;
Proc iml;
use P; read all into P;
use Sigma3; read all into Sigma3;
P3=P*Sigma3*t(P);
create P3 from P3; append from P3; quit;

Data Sigma_4;
set s1;
if i=4;
keep var;
run;
Proc iml;
use Sigma_4; read all into Sigma_4;
Sigma4=diag(Sigma_4);
create Sigma4 from Sigma4; append from Sigma4;
quit;
Proc iml;
use P; read all into P;
use Sigma4; read all into Sigma4;
P4=P*Sigma4*t(P);
create P4 from P4; append from P4; quit;

Data Sigma_5;
set s1;
if i=5;
keep var;
run;
Proc iml;
use Sigma_5; read all into Sigma_5;
Sigma5=diag(Sigma_5);
create Sigma5 from Sigma5; append from Sigma5;
quit;
Proc iml;
use P; read all into P;
use Sigma5; read all into Sigma5;
P5=P*Sigma5*t(P);
create P5 from P5; append from P5; quit;

Data Sigma_6;
set s1;
if i=6;
keep var;
run;
Proc iml;
use Sigma_6; read all into Sigma_6;
Sigma6=diag(Sigma_6);
create Sigma6 from Sigma6; append from Sigma6;
quit;
Proc iml;
use P; read all into P;
use Sigma6; read all into Sigma6;
P6=P*Sigma6*t(P);
create P6 from P6; append from P6; quit;

Data Sigma_7;
set s1;
if i=7;
keep var;
run;
Proc iml;
use Sigma_7; read all into Sigma_7;
Sigma7=diag(Sigma_7);
create Sigma7 from Sigma7; append from Sigma7;
quit;
Proc iml;
use P; read all into P;
use Sigma7; read all into Sigma7;
P7=P*Sigma7*t(P);
create P7 from P7; append from P7; quit;

Data Sigma_8;
set s1;
if i=8;
keep var;
run;
Proc iml;
use Sigma_8; read all into Sigma_8;
Sigma8=diag(Sigma_8);
create Sigma8 from Sigma8; append from Sigma8;
quit;
Proc iml;
use P; read all into P;
use Sigma8; read all into Sigma8;
P8=P*Sigma8*t(P);
create P8 from P8; append from P8; quit;

Data Sigma_9;
set s1;
if i=9;
keep var;
run;
Proc iml;
use Sigma_9; read all into Sigma_9;
Sigma9=diag(Sigma_9);
create Sigma9 from Sigma9; append from Sigma9;
quit;
Proc iml;
use P; read all into P;
use Sigma9; read all into Sigma9;
P9=P*Sigma9*t(P);
create P9 from P9; append from P9; quit;
```

```
Data Sigma_10;
set s1;
if i=10;
keep var;
run;
Proc iml;
use Sigma_10; read all into Sigma_10;
Sigma10=diag(Sigma_10);
create Sigma10 from Sigma10; append from
Sigma10; quit;
Proc iml;
use P; read all into P;
use Sigma10; read all into Sigma10;
P10=P*Sigma10*t(P);
create P10 from P10; append from P10; quit;

Data Sigma_11;
set s1;
if i=11;
keep var;
run;
Proc iml;
use Sigma_11; read all into Sigma_11;
Sigma11=diag(Sigma_11);
create Sigma11 from Sigma11; append from
Sigma11; quit;
Proc iml;
use P; read all into P;
use Sigma11; read all into Sigma11;
P11=P*Sigma11*t(P);
create P11 from P11; append from P11; quit;

Data Sigma_12;
set s1;
if i=12;
keep var;
run;
Proc iml;
use Sigma_12; read all into Sigma_12;
Sigma12=diag(Sigma_12);
create Sigma12 from Sigma12; append from
Sigma12; quit;
Proc iml;
use P; read all into P;
use Sigma12; read all into Sigma12;
P12=P*Sigma12*t(P);
create P12 from P12; append from P12; quit;

Data Sigma_13;
set s1;
if i=13;
keep var;
run;
Proc iml;
use Sigma_13; read all into Sigma_13;
Sigma13=diag(Sigma_13);
create Sigma13 from Sigma13; append from
Sigma13; quit;
Proc iml;
use P; read all into P;
use Sigma13; read all into Sigma13;
P13=P*Sigma13*t(P);
create P13 from P13; append from P13; quit;

Data Sigma_14;
set s1;
if i=14;
keep var;
run;
Proc iml;
use Sigma_14; read all into Sigma_14;
Sigma14=diag(Sigma_14);
create Sigma14 from Sigma14; append from
Sigma14; quit;
Proc iml;
use P; read all into P;
use Sigma14; read all into Sigma14;
P14=P*Sigma14*t(P);
create P14 from P14; append from P14; quit;

Data Sigma_15;
set s1;
if i=15;
keep var;
run;
Proc iml;
use Sigma_15; read all into Sigma_15;
Sigma15=diag(Sigma_15);
create Sigma15 from Sigma15; append from
Sigma15; quit;
Proc iml;
use P; read all into P;
use Sigma15; read all into Sigma15;
P15=P*Sigma15*t(P);
create P15 from P15; append from P15; quit;

Data Sigma_16;
set s1;
if i=16;
keep var;
run;
Proc iml;
use Sigma_16; read all into Sigma_16;
Sigma16=diag(Sigma_16);
create Sigma16 from Sigma16; append from
Sigma16; quit;
Proc iml;
use P; read all into P;
use Sigma16; read all into Sigma16;
P16=P*Sigma16*t(P);
create P16 from P16; append from P16; quit;

Data Sigma_17;
set s1;
if i=17;
keep var;
run;
Proc iml;
use Sigma_17; read all into Sigma_17;
Sigma17=diag(Sigma_17);
create Sigma17 from Sigma17; append from
Sigma17; quit;
Proc iml;
use P; read all into P;
use Sigma17; read all into Sigma17;
P17=P*Sigma17*t(P);
create P17 from P17; append from P17; quit;

Data Sigma_18;
set s1;
if i=18;
keep var;
run;
Proc iml;
use Sigma_18; read all into Sigma_18;
Sigma18=diag(Sigma_18);
create Sigma18 from Sigma18; append from
Sigma18; quit;
Proc iml;
use P; read all into P;
use Sigma18; read all into Sigma18;
P18=P*Sigma18*t(P);
create P18 from P18; append from P18; quit;

Data Sigma_19;
set s1;
if i=19;
keep var;
run;
Proc iml;
use Sigma_19; read all into Sigma_19;
Sigma19=diag(Sigma_19);
create Sigma19 from Sigma19; append from
Sigma19; quit;
Proc iml;
use P; read all into P;
use Sigma19; read all into Sigma19;
P19=P*Sigma19*t(P);
create P19 from P19; append from P19; quit;

Data Sigma_20;
set s1;
if i=20;
keep var;
run;
Proc iml;
use Sigma_20; read all into Sigma_20;
Sigma20=diag(Sigma_20);
create Sigma20 from Sigma20; append from
Sigma20; quit;
Proc iml;
use P; read all into P;
use Sigma20; read all into Sigma20;
P20=P*Sigma20*t(P);
create P20 from P20; append from P20; quit;
```

```
Data Sigma_21;
set s1;
if i=21;
keep var;
run;
Proc iml;
use Sigma_21; read all into Sigma_21;
Sigma21=diag(Sigma_21);
create Sigma21 from Sigma21; append from
Sigma21; quit;
Proc iml;
use P; read all into P;
use Sigma21; read all into Sigma21;
P21=P*Sigma21*t(P);
create P21 from P21; append from P21; quit;

Data Sigma_22;
set s1;
if i=22;
keep var;
run;
Proc iml;
use Sigma_22; read all into Sigma_22;
Sigma22=diag(Sigma_22);
create Sigma22 from Sigma22; append from
Sigma22; quit;
Proc iml;
use P; read all into P;
use Sigma22; read all into Sigma22;
P22=P*Sigma22*t(P);
create P22 from P22; append from P22; quit;

Data Sigma_23;
set s1;
if i=23;
keep var;
run;
Proc iml;
use Sigma_23; read all into Sigma_23;
Sigma23=diag(Sigma_23);
create Sigma23 from Sigma23; append from
Sigma23; quit;
Proc iml;
use P; read all into P;
use Sigma23; read all into Sigma23;
P23=P*Sigma23*t(P);
create P23 from P23; append from P23; quit;

Data Sigma_24;
set s1;
if i=24;
keep var;
run;
Proc iml;
use Sigma_24; read all into Sigma_24;
Sigma24=diag(Sigma_24);
create Sigma24 from Sigma24; append from
Sigma24; quit;
Proc iml;
use P; read all into P;
use Sigma24; read all into Sigma24;
P24=P*Sigma24*t(P);
create P24 from P24; append from P24; quit;

Data Sigma_25;
set s1;
if i=25;
keep var;
run;
Proc iml;
use Sigma_25; read all into Sigma_25;
Sigma25=diag(Sigma_25);
create Sigma25 from Sigma25; append from
Sigma25; quit;
Proc iml;
use P; read all into P;
use Sigma25; read all into Sigma25;
P25=P*Sigma25*t(P);
create P25 from P25; append from P25; quit;

Data Sigma_26;
set s1;
if i=26;
keep var;
run;
Proc iml;
use Sigma_26; read all into Sigma_26;
Sigma26=diag(Sigma_26);
create Sigma26 from Sigma26; append from
Sigma26; quit;
Proc iml;
use P; read all into P;
use Sigma26; read all into Sigma26;
P26=P*Sigma26*t(P);
create P26 from P26; append from P26; quit;

Data Sigma_27;
set s1;
if i=27;
keep var; run;

Proc iml;
use Sigma_27; read all into Sigma_27;
Sigma27=diag(Sigma_27);
create Sigma27 from Sigma27; append from
Sigma27; quit;
Proc iml;
use P; read all into P;
use Sigma27; read all into Sigma27;
P27=P*Sigma27*t(P);
create P27 from P27; append from P27; quit;

Data Sigma_28;
set s1;
if i=28;
keep var;
run;
Proc iml;
use Sigma_28; read all into Sigma_28;
Sigma28=diag(Sigma_28);
create Sigma28 from Sigma28; append from
Sigma28; quit;
Proc iml;
use P; read all into P;
use Sigma28; read all into Sigma28;
P28=P*Sigma28*t(P);
create P28 from P28; append from P28; quit;

Proc iml;
use P1; read all into P1;
use P2; read all into P2;
use P3; read all into P3;
use P4; read all into P4;
use P5; read all into P5;
use P6; read all into P6;
use P7; read all into P7;
use P8; read all into P8;
use P9; read all into P9;
use P10; read all into P10;
use P11; read all into P11;
use P12; read all into P12;
use P13; read all into P13;
use P14; read all into P14;
blockPP1=BLOCK(P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,
P11,P12,P13,P14);
create blockPP1 from blockPP1; append from
blockPP1; quit;

Proc iml;
use P15; read all into P15;
use P16; read all into P16;
use P17; read all into P17;
use P18; read all into P18;
use P19; read all into P19;
use P20; read all into P20;
use P21; read all into P21;
use P22; read all into P22;
use P23; read all into P23;
use P24; read all into P24;
use P25; read all into P25;
use P26; read all into P26;
use P27; read all into P27;
use P28; read all into P28;
blockPP2=BLOCK(P15,P16,P17,P18,P19,P20,P21,P22
,P23,P24,P25,P26,P27,P28);
create blockPP2 from blockPP2; append from
blockPP2; quit;

Proc iml;
use blockPP1; read all into blockPP1;
use blockPP2; read all into blockPP2;
blockPP=BLOCK(blockPP1,blockPP2);
create blockPP from blockPP; append from
blockPP; quit;

/* *** Fim *** */
```

```
Proc iml;
use H1; read all into H1;
HH=H1*t(H1);
eval=eigval(HH);
create eval from eval; append from eval; quit;

Data id;
do id=1 to 196;
output; end; run;

Data eval;
merge id eval;
run;

Data new;
set eval;
by id;
retain count 0;
if first.id then count=0;
if col1 > 0.000001 then count = count + 1;
run;

Proc means data=new sum noprint;
var count;
output out=caracH1 sum=caracH1;
run;

Data caracH1;
set caracH1;
keep caracH1; run;

Proc iml;
use blockPP; read all into blockPP;
use blockID; read all into blockID;
use Hs1; read all into Hs1;
use rg1_; read all into rg1_;
use caracH1; read all into caracH1;
varu= (rg1_ -  trace((blockID -
Hs1)*blockPP))*(1/(28*6 - caracH1));
create varu from varu; append from varu; quit;

/* Termina aqui o cálculo da variância
sigma2_u */

/** Cálculo da matriz: c^(-0.5)*f'*P*Teta -
var. dependente do modelo transformado **/

Proc iml;
use tetad1; read all into tetad1;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y1=(1/sqrt(c))*t(f)*P*tetad1;
create y1 from y1; append from y1; quit;
Proc iml;
use tetad2; read all into tetad2;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y2=(1/sqrt(c))*t(f)*P*tetad2;
create y2 from y2; append from y2; quit;
Proc iml;
use tetad3; read all into tetad3;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y3=(1/sqrt(c))*t(f)*P*tetad3;
create y3 from y3; append from y3; quit;
Proc iml;
use tetad4; read all into tetad4;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y4=(1/sqrt(c))*t(f)*P*tetad4;
create y4 from y4; append from y4; quit;
Proc iml;
use tetad5; read all into tetad5;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y5=(1/sqrt(c))*t(f)*P*tetad5;
create y5 from y5; append from y5; quit;
Proc iml;
use tetad6; read all into tetad6;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y6=(1/sqrt(c))*t(f)*P*tetad6;
create y6 from y6; append from y6; quit;
Proc iml;
use tetad7; read all into tetad7;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y7=(1/sqrt(c))*t(f)*P*tetad7;
create y7 from y7; append from y7; quit;
Proc iml;
use tetad8; read all into tetad8;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y8=(1/sqrt(c))*t(f)*P*tetad8;
create y8 from y8; append from y8; quit;
Proc iml;
use tetad9; read all into tetad9;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y9=(1/sqrt(c))*t(f)*P*tetad9;
create y9 from y9; append from y9; quit;
Proc iml;
use tetad10; read all into tetad10;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y10=(1/sqrt(c))*t(f)*P*tetad10;
create y10 from y10; append from y10; quit;
Proc iml;
use tetad11; read all into tetad11;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y11=(1/sqrt(c))*t(f)*P*tetad11;
create y11 from y11; append from y11; quit;
Proc iml;
use tetad12; read all into tetad12;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y12=(1/sqrt(c))*t(f)*P*tetad12;
create y12 from y12; append from y12; quit;
Proc iml;
use tetad13; read all into tetad13;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y13=(1/sqrt(c))*t(f)*P*tetad13;
create y13 from y13; append from y13; quit;
Proc iml;
use tetad14; read all into tetad14;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y14=(1/sqrt(c))*t(f)*P*tetad14;
create y14 from y14; append from y14; quit;
Proc iml;
use tetad15; read all into tetad15;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y15=(1/sqrt(c))*t(f)*P*tetad15;
create y15 from y15; append from y15; quit;
Proc iml;
use tetad16; read all into tetad16;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y16=(1/sqrt(c))*t(f)*P*tetad16;
create y16 from y16; append from y16; quit;
Proc iml;
use tetad17; read all into tetad17;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y17=(1/sqrt(c))*t(f)*P*tetad17;
create y17 from y17; append from y17; quit;
Proc iml;
use tetad18; read all into tetad18;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y18=(1/sqrt(c))*t(f)*P*tetad18;
create y18 from y18; append from y18; quit;
Proc iml;
use tetad19; read all into tetad19;
```

```
use P; read all into P;
use f; read all into f;
use c; read all into c;
y19=(1/sqrt(c))*t(f)*P*tetad19;
create y19 from y19; append from y19; quit;
Proc iml;
use tetad20; read all into tetad20;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y20=(1/sqrt(c))*t(f)*P*tetad20;
create y20 from y20; append from y20; quit;
Proc iml;
use tetad21; read all into tetad21;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y21=(1/sqrt(c))*t(f)*P*tetad21;
create y21 from y21; append from y21; quit;
Proc iml;
use tetad22; read all into tetad22;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y22=(1/sqrt(c))*t(f)*P*tetad22;
create y22 from y22; append from y22; quit;
Proc iml;
use tetad23; read all into tetad23;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y23=(1/sqrt(c))*t(f)*P*tetad23;
create y23 from y23; append from y23; quit;
Proc iml;
use tetad24; read all into tetad24;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y24=(1/sqrt(c))*t(f)*P*tetad24;
create y24 from y24; append from y24; quit;
Proc iml;
use tetad25; read all into tetad25;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y25=(1/sqrt(c))*t(f)*P*tetad25;
create y25 from y25; append from y25; quit;
Proc iml;
use tetad26; read all into tetad26;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y26=(1/sqrt(c))*t(f)*P*tetad26;
create y26 from y26; append from y26; quit;
Proc iml;
use tetad27; read all into tetad27;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y27=(1/sqrt(c))*t(f)*P*tetad27;
create y27 from y27; append from y27; quit;
Proc iml;
use tetad28; read all into tetad28;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y28=(1/sqrt(c))*t(f)*P*tetad28;
create y28 from y28; append from y28; quit;
run;

Data Z2;
set y1 y2 y3 y4 y5 y6 y7 y8 y9 y10 y11 y12 y13
y14 y15
y16 y17 y18 y19 y20 y21 y22 y23 y24 y25 y26
y27 y28;
run;

/** Cálculo da matriz: c^(-0.5)*f'*P*X - var.
independentes do modelo transformado **/

Proc iml;
use xd1; read all into xd1;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x1=(1/sqrt(c))*t(f)*P*xd1;
create x1 from x1; append from x1; quit;
Proc iml;
use xd2; read all into xd2;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x2=(1/sqrt(c))*t(f)*P*xd2;
create x2 from x2; append from x2; quit;
Proc iml;
use xd3; read all into xd3;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x3=(1/sqrt(c))*t(f)*P*xd3;
create x3 from x3; append from x3; quit;
Proc iml;
use xd4; read all into xd4;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x4=(1/sqrt(c))*t(f)*P*xd4;
create x4 from x4; append from x4; quit;
Proc iml;
use xd5; read all into xd5;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x5=(1/sqrt(c))*t(f)*P*xd5;
create x5 from x5; append from x5; quit;
Proc iml;
use xd6; read all into xd6;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x6=(1/sqrt(c))*t(f)*P*xd6;
create x6 from x6; append from x6; quit;
Proc iml;
use xd7; read all into xd7;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x7=(1/sqrt(c))*t(f)*P*xd7;
create x7 from x7; append from x7; quit;
Proc iml;
use xd8; read all into xd8;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x8=(1/sqrt(c))*t(f)*P*xd8;
create x8 from x8; append from x8; quit;
Proc iml;
use xd9; read all into xd9;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x9=(1/sqrt(c))*t(f)*P*xd9;
create x9 from x9; append from x9; quit;
Proc iml;
use xd10; read all into xd10;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x10=(1/sqrt(c))*t(f)*P*xd10;
create x10 from x10; append from x10; quit;
Proc iml;
use xd11; read all into xd11;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x11=(1/sqrt(c))*t(f)*P*xd11;
create x11 from x11; append from x11; quit;
Proc iml;
use xd12; read all into xd12;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x12=(1/sqrt(c))*t(f)*P*xd12;
create x12 from x12; append from x12; quit;
Proc iml;
use xd13; read all into xd13;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x13=(1/sqrt(c))*t(f)*P*xd13;
create x13 from x13; append from x13; quit;
Proc iml;
use xd14; read all into xd14;
use P; read all into P;
use f; read all into f;
use c; read all into c;
```

```
x14=(1/sqrt(c))*t(f)*P*xd14;
create x14 from x14; append from x14; quit;
Proc iml;
use xd15; read all into xd15;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x15=(1/sqrt(c))*t(f)*P*xd15;
create x15 from x15; append from x15; quit;
Proc iml;
use xd16; read all into xd16;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x16=(1/sqrt(c))*t(f)*P*xd16;
create x16 from x16; append from x16; quit;
Proc iml;
use xd17; read all into xd17;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x17=(1/sqrt(c))*t(f)*P*xd17;
create x17 from x17; append from x17; quit;
Proc iml;
use xd18; read all into xd18;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x18=(1/sqrt(c))*t(f)*P*xd18;
create x18 from x18; append from x18; quit;
Proc iml;
use xd19; read all into xd19;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x19=(1/sqrt(c))*t(f)*P*xd19;
create x19 from x19; append from x19; quit;
Proc iml;
use xd20; read all into xd20;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x20=(1/sqrt(c))*t(f)*P*xd20;
create x20 from x20; append from x20; quit;
Proc iml;
use xd21; read all into xd21;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x21=(1/sqrt(c))*t(f)*P*xd21;
create x21 from x21; append from x21; quit;
Proc iml;
use xd22; read all into xd22;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x22=(1/sqrt(c))*t(f)*P*xd22;
create x22 from x22; append from x22; quit;
Proc iml;
use xd23; read all into xd23;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x23=(1/sqrt(c))*t(f)*P*xd23;
create x23 from x23; append from x23; quit;
Proc iml;
use xd24; read all into xd24;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x24=(1/sqrt(c))*t(f)*P*xd24;
create x24 from x24; append from x24; quit;
Proc iml;
use xd25; read all into xd25;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x25=(1/sqrt(c))*t(f)*P*xd25;
create x25 from x25; append from x25; quit;
Proc iml;
use xd26; read all into xd26;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x26=(1/sqrt(c))*t(f)*P*xd26;
create x26 from x26; append from x26; quit;
Proc iml;
use xd27; read all into xd27;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x27=(1/sqrt(c))*t(f)*P*xd27;
create x27 from x27; append from x27; quit;
Proc iml;
use xd28; read all into xd28;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x28=(1/sqrt(c))*t(f)*P*xd28;
create x28 from x28; append from x28; quit;
run;

Data H2;
set x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13
x14 x15
x16 x17 x18 x19 x20 x21 x22 x23 x24 x25 x26
x27 x28;
run;

/** Regressão de Z2 (Y) sobre H2 (X) **/

Data Z2;
set Z2;
rename col1=col0; run;

Data reg2;
merge Z2 H2; run;

proc reg data=reg2 noprint;
        model col0=col1 col2 / noint;
        output out=rg2  residual=r2;
run; quit;

Data rg2;
set rg2;
qr2=r2*r2;
Keep qr2; run;

Proc means data=rg2 sum noprint;
var qr2;
output out=rg2_ sum=sqr2;
run;

Data rg2_;
set rg2_;
keep sqr2; run;

/** Cálculo da variância sigma2_v **/

Proc iml;
use H2; read all into H2;
Ident=I(28);
IF=Ident - H2*ginv(t(H2)*H2)*t(H2);
create IF from IF; append from IF; quit;

Proc iml;
use f; read all into f;
use P1; read all into P1;
use P2; read all into P2;
use P3; read all into P3;
use P4; read all into P4;
use P5; read all into P5;
use P6; read all into P6;
use P7; read all into P7;
use P8; read all into P8;
use P9; read all into P9;
use P10; read all into P10;
use P11; read all into P11;
use P12; read all into P12;
use P13; read all into P13;
use P14; read all into P14;
use P15; read all into P15;
use P16; read all into P16;
use P17; read all into P17;
use P18; read all into P18;
use P19; read all into P19;
use P20; read all into P20;
use P21; read all into P21;
use P22; read all into P22;
use P23; read all into P23;
use P24; read all into P24;
use P25; read all into P25;
use P26; read all into P26;
use P27; read all into P27;
use P28; read all into P28;
use c; read all into c;
```

```
cfp1=t(f)*P1*f*(1 / c);
cfp2=t(f)*P2*f*(1 / c);
cfp3=t(f)*P3*f*(1 / c);
cfp4=t(f)*P4*f*(1 / c);
cfp5=t(f)*P5*f*(1 / c);
cfp6=t(f)*P6*f*(1 / c);
cfp7=t(f)*P7*f*(1 / c);
cfp8=t(f)*P8*f*(1 / c);
cfp9=t(f)*P9*f*(1 / c);
cfp10=t(f)*P10*f*(1 / c);
cfp11=t(f)*P11*f*(1 / c);
cfp12=t(f)*P12*f*(1 / c);
cfp13=t(f)*P13*f*(1 / c);
cfp14=t(f)*P14*f*(1 / c);
cfp15=t(f)*P15*f*(1 / c);
cfp16=t(f)*P16*f*(1 / c);
cfp17=t(f)*P17*f*(1 / c);
cfp18=t(f)*P18*f*(1 / c);
cfp19=t(f)*P19*f*(1 / c);
cfp20=t(f)*P20*f*(1 / c);
cfp21=t(f)*P21*f*(1 / c);
cfp22=t(f)*P22*f*(1 / c);
cfp23=t(f)*P23*f*(1 / c);
cfp24=t(f)*P24*f*(1 / c);
cfp25=t(f)*P25*f*(1 / c);
cfp26=t(f)*P26*f*(1 / c);
cfp27=t(f)*P27*f*(1 / c);
cfp28=t(f)*P28*f*(1 / c);
cfp_=cfp1//cfp2//cfp3//cfp4//cfp5//cfp6//cfp7/
/cfp8//cfp9//cfp10//cfp11//cfp12//cfp13//cfp14
//cfp15//cfp16//cfp17//cfp18//cfp19//cfp20//cf
p21//cfp22//cfp23//cfp24//cfp25//cfp26//cfp27/
/cfp28;
cfp=diag(cfp_);
create cfp from cfp; append from cfp; quit;

/* *** Fim *** */

Proc iml;
use H2; read all into H2;
FF=H2*t(H2);
eval1=eigval(FF);
create eval1 from eval1; append from eval1;
quit;

Data id1;
do id=1 to 28;
output; end;
run;

Data eval1;
merge id1 eval1;
run;

Data new1;
set eval1;
by id;
retain count 0;
if first.id then count=0;
if col1 > 0.000001 then count = count + 1;
run;

Proc means data=new1 sum noprint;
var count;
output out=caracH2 sum=caracH2;
run;

Data caracH2;
set caracH2;
keep caracH2;  /* Caracteristica de F */
run;

Proc iml;
use IF; read all into IF;
use cfp; read all into cfp;
use c; read all into c;
use rg2_; read all into rg2_;
use caracH2; read all into caracH2;
use varu; read all into varu;
varv= (rg2_ -  trace(IF*cfp))*(1/(c*(28 -
caracH2))) - varu/c;
create varv from varv; append from varv; quit;

/* Termina aqui o cálculo da variância
sigma2_v */
/* Truncagem a zero das variâncias sigma2_u e
sigma2_v */

Data varu;
set varu;
sigma2u= max( col1 , 0);
keep sigma2u;
run;
Data varv;
set varv;
sigma2v= max( col1 , 0);
keep sigma2v;
run;

Proc sort data=s1;
by i t; run;

Data s2;
set s1;
keep var; run;

Proc iml;
use dataro; read all into dataro;
use varu; read all into varu;
use varv; read all into varv;
use s2; read all into s2;
C1={1, 0, 0};
C2={0, 1, 0};
C3={0, 0, 1};
par0 = varv*C1 + (varu/(1 - dataro*dataro))*C2
+ dataro*C3;
par1 = par0//s2;
create par1 from par1; append from par1; quit;

/* *** Fim *** */

Data par;
set par1;
rename col1=est;
run;

/*********************************************
*********************************************/
/***** ESTIMAÇÃO DO MODELO DE RAO-YU ********/
/*********************************************
*********************************************/

Proc sort data=iabhipth1 out= cron;
by i t;
run;

Proc mixed data=cron MMEqSol noprofile
noclprint;
        class i t;
        model y=x / solution;
        random i / solution G V;
        random t / subject=i type=ar(1);
        repeated /group=i*t;
        parms /parmsdata=par noiter;
make 'solutionf' out=beta;
make 'solutionr' out=u;
make 'Fitstatistics' out=aic;
make 'G' out=Mat_Gaux;
make 'V' out=Mat_V;
run; quit;

/********************************************/
/* CÁLCULO DAS ESTIMATIVAS DO PREÇO MÉDIO DE
TRANSACÇÃO - NUT III */
/********************************************/

Data id;
do id=1 to 2;
output; end;
run;

Data beta;
merge id beta;
run;

Data beta0;
set beta;
if id=1;
rename estimate=b0;
keep estimate;
run;

Data beta1;
set beta;
if id=2;
rename estimate=b1;
```

```
keep estimate;
run;

Data betan;
merge beta0 beta1;
_type_=0; run;

/****** Efeitos Aleatórios: v ******/

Data u;
set u;
if estimate='.' then estimate=0;
run;

Data v;
set u;
if Effect='i';
keep i estimate;
run;

Proc sort data=v;
by i; run;

Proc iml;
use v; read all into v;
one=J(7,1,1);
v1=v@one;
create v1 from v1; append from v1; quit;

Data v1;
set v1;
_type_=0;
rename col2=v; run;

/****** Efeitos Aleatórios: u ******/

Data u1;
set u;
if Effect='t';
_type_=0;
keep i t estimate _type_;
rename estimate=u; run;

Proc sort data=u1;
by i t; run;

Data beta_vu;
merge betan v1 u1;
by _type_; run;

/** Estimação do preço médio de transacção em
cada NUTIII **/

Proc sort data=cron;
by i t; run;

Data cron;
merge cron beta_vu;
by i t; run;

Data crono_resf;
set cron;
e = b0 + x*b1 + v + u;
keep i y t _type_ x b0 b1 u v e;
run;

/*********************************************/
/* Cálculo das estimativas do EQM do EBLUP **/
/*********************************************/

/*** MATRIZ X ***/

Proc sort data=cron out=cronologico;
by i t;
run;

Data XX1;
set cronologico;
keep x;
run;

Proc iml;
J=J(196,1,1);
create J from J;
append from J;
quit;

Proc iml;
use J;
read all into J;
use XX1;
read all into XX1;
X=insert(J,XX1,0,2);
create X from X;
append from X;
quit;

/*** MATRIZ G ***/

Proc iml;
J7=J(7,1,1);
create J7 from J7;
append from J7;
quit;

Proc iml;
use J7;
read all into J7;
use varv;
read all into varv;
J77=J7*J7`;
BlockMat_Gv=varv*J77;
create BlockMat_Gv from BlockMat_Gv;
append from BlockMat_Gv;
quit;

Proc iml;
use BlockMat_Gv;
read all into BlockMat_Gv;
Mat_Gv1=block (BlockMat_Gv, BlockMat_Gv,
BlockMat_Gv, BlockMat_Gv, BlockMat_Gv,
BlockMat_Gv, BlockMat_Gv,
BlockMat_Gv, BlockMat_Gv, BlockMat_Gv,
BlockMat_Gv, BlockMat_Gv, BlockMat_Gv,
BlockMat_Gv);
create Mat_Gv1 from Mat_Gv1;
append from Mat_Gv1; quit;

Proc iml;
use Mat_Gv1;
read all into Mat_Gv1;
Mat_Gv=block (Mat_Gv1, Mat_Gv1);
create Mat_Gv from Mat_Gv;
append from Mat_Gv; quit;

/***/

Data Mat_Gu;
set Mat_Gaux;
where row>28;
drop row effect i t col1-col28;
run;

/***/

Proc iml;
use Mat_Gv;
read all into Mat_Gv;
use Mat_Gu;
read all into Mat_Gu;
MatG=Mat_Gv+Mat_Gu;
create MatG from MatG;
append from MatG;
quit;

/*** Matriz GAMA e suas colunas ***/

Data ro_1;
set dataro;
ro_1= 1-ro*ro;
keep ro_1;
run;
Data ro_2;
set dataro;
ro_2= ro*ro;
keep ro_2;
run;
Data ro_3;
set dataro;
ro_3= ro*ro*ro;
keep ro_3;
run;
Data ro_4;
set dataro;
ro_4= ro*ro*ro*ro;
keep ro_4;
```

```
run;
Data ro_5;
set dataro;
ro_5= ro*ro*ro*ro*ro;
keep ro_5;
run;
Data ro_6;
set dataro;
ro_6= ro*ro*ro*ro*ro*ro;
keep ro_6;
run;

Proc iml;
use ro_1; read all into ro_1;
use dataro; read all into dataro;
use ro_2; read all into ro_2;
use ro_3; read all into ro_3;
use ro_4; read all into ro_4;
use ro_5; read all into ro_5;
use ro_6; read all into ro_6;
I=I(7);
I1={0 1 0 0 0 0 0, 0 0 1 0 0 0 0, 0 0 0 1 0 0
0, 0 0 0 0 1 0 0, 0 0 0 0 0 1 0, 0 0 0 0 0 0
1, 0 0 0 0 0 0 0};
I2={0 0 1 0 0 0 0, 0 0 0 1 0 0 0, 0 0 0 0 1 0
0, 0 0 0 0 0 1 0, 0 0 0 0 0 0 1, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I3={0 0 0 1 0 0 0, 0 0 0 0 1 0 0, 0 0 0 0 0 1
0, 0 0 0 0 0 0 1, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I4={0 0 0 0 1 0 0, 0 0 0 0 0 1 0, 0 0 0 0 0 0
1, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I5={0 0 0 0 0 1 0, 0 0 0 0 0 0 1, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I6={0 0 0 0 0 0 1, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
Gama = I/ro_1 + dataro*I1/ro_1 +
t(dataro*I1/ro_1) + ro_2*I2/ro_1 +
t(ro_2*I2/ro_1) + ro_3*I3/ro_1 +
t(ro_3*I3/ro_1) + ro_4*I4/ro_1 +
t(ro_4*I4/ro_1) + ro_5*I5/ro_1 +
t(ro_5*I5/ro_1) + ro_6*I6/ro_1 +
t(ro_6*I6/ro_1);
create Gama from Gama;
append from Gama;
quit;

Proc iml;
use Gama; read all into Gama;
Gama1=Gama[,1];
create Gama1 from Gama1;
append from Gama1;
quit;
Proc iml;
use Gama; read all into Gama;
Gama2=Gama[,2];
create Gama2 from Gama2;
append from Gama2;
quit;
Proc iml;
use Gama; read all into Gama;
Gama3=Gama[,3];
create Gama3 from Gama3;
append from Gama3;
quit;
Proc iml;
use Gama; read all into Gama;
Gama4=Gama[,4];
create Gama4 from Gama4;
append from Gama4;
quit;
Proc iml;
use Gama; read all into Gama;
Gama5=Gama[,5];
create Gama5 from Gama5;
append from Gama5;
quit;
Proc iml;
use Gama; read all into Gama;
Gama6=Gama[,6];
create Gama6 from Gama6;
append from Gama6;
quit;
Proc iml;
use Gama; read all into Gama;
Gama7=Gama[,7];
create Gama7 from Gama7;
append from Gama7;
quit;

/*** MATRIZ V e seus Blocos ***/

Data MatV;
set Mat_v;
drop Index row;
run;

Proc iml;
use MatV; read all into MatV;
MatV1=MatV[1:7,1:7];
create MatV1 from MatV1; append from MatV1;
quit;
Proc iml;
use MatV; read all into MatV;
MatV2=MatV[8:14,8:14];
create MatV2 from MatV2; append from MatV2;
quit;
Proc iml;
use MatV; read all into MatV;
MatV3=MatV[15:21,15:21];
create MatV3 from MatV3; append from MatV3;
quit;
Proc iml;
use MatV; read all into MatV;
MatV4=MatV[22:28,22:28];
create MatV4 from MatV4; append from MatV4;
quit;
Proc iml;
use MatV; read all into MatV;
MatV5=MatV[29:35,29:35];
create MatV5 from MatV5; append from MatV5;
quit;
Proc iml;
use MatV; read all into MatV;
MatV6=MatV[36:42,36:42];
create MatV6 from MatV6; append from MatV6;
quit;
Proc iml;
use MatV; read all into MatV;
MatV7=MatV[43:49,43:49];
create MatV7 from MatV7; append from MatV7;
quit;
Proc iml;
use MatV; read all into MatV;
MatV8=MatV[50:56,50:56];
create MatV8 from MatV8; append from MatV8;
quit;
Proc iml;
use MatV; read all into MatV;
MatV9=MatV[57:63,57:63];
create MatV9 from MatV9; append from MatV9;
quit;
Proc iml;
use MatV; read all into MatV;
MatV10=MatV[64:70,64:70];
create MatV10 from MatV10; append from MatV10;
quit;
Proc iml;
use MatV; read all into MatV;
MatV11=MatV[71:77,71:77];
create MatV11 from MatV11; append from MatV11;
quit;
Proc iml;
use MatV; read all into MatV;
MatV12=MatV[78:84,78:84];
create MatV12 from MatV12; append from MatV12;
quit;
Proc iml;
use MatV; read all into MatV;
MatV13=MatV[85:91,85:91];
create MatV13 from MatV13; append from MatV13;
quit;
Proc iml;
use MatV; read all into MatV;
MatV14=MatV[92:98,92:98];
create MatV14 from MatV14; append from MatV14;
quit;
Proc iml;
use MatV; read all into MatV;
MatV15=MatV[99:105,99:105];
create MatV15 from MatV15; append from MatV15;
quit;
Proc iml;
```

```
use MatV; read all into MatV;
MatV16=MatV[106:112,106:112];
create MatV16 from MatV16; append from MatV16;
quit;
Proc iml;
use MatV; read all into MatV;
MatV17=MatV[113:119,113:119];
create MatV17 from MatV17; append from MatV17;
quit;
Proc iml;
use MatV; read all into MatV;
MatV18=MatV[120:126,120:126];
create MatV18 from MatV18; append from MatV18;
quit;
Proc iml;
use MatV; read all into MatV;
MatV19=MatV[127:133,127:133];
create MatV19 from MatV19; append from MatV19;
quit;
Proc iml;
use MatV; read all into MatV;
MatV20=MatV[134:140,134:140];
create MatV20 from MatV20; append from MatV20;
quit;
Proc iml;
use MatV; read all into MatV;
MatV21=MatV[141:147,141:147];
create MatV21 from MatV21; append from MatV21;
quit;
Proc iml;
use MatV; read all into MatV;
MatV22=MatV[148:154,148:154];
create MatV22 from MatV22; append from MatV22;
quit;
Proc iml;
use MatV; read all into MatV;
MatV23=MatV[155:161,155:161];
create MatV23 from MatV23; append from MatV23;
quit;
Proc iml;
use MatV; read all into MatV;
MatV24=MatV[162:168,162:168];
create MatV24 from MatV24; append from MatV24;
quit;
Proc iml;
use MatV; read all into MatV;
MatV25=MatV[169:175,169:175];
create MatV25 from MatV25; append from MatV25;
quit;
Proc iml;
use MatV; read all into MatV;
MatV26=MatV[176:182,176:182];
create MatV26 from MatV26; append from MatV26;
quit;
Proc iml;
use MatV; read all into MatV;
MatV27=MatV[183:189,183:189];
create MatV27 from MatV27; append from MatV27;
quit;
Proc iml;
use MatV; read all into MatV;
MatV28=MatV[190:196,190:196];
create MatV28 from MatV28; append from MatV28;
quit;

/* *** Fim *** */

/* MatVi - verificação para o 2º bloco */
/*
Proc iml;
use BlockMat_Gv; read all into BlockMat_Gv;
use Gama; read all into Gama;
use varu; read all into varu;
use Sigma2; read all into Sigma2;
MatV2=Sigma2 + varu*Gama + BlockMat_Gv;
create MatV2 from MatV2;
append from MatV2; quit;*/

/***************************/
/****** Cálculo do g1 ******/
/***************************/

Proc iml;
use MatV; read all into MatV;
use MatG; read all into MatG;
G1_aux=MatG - MatG*inv(MatV)*MatG;
create G1_aux from G1_aux;
append from G1_aux;
quit;

Proc iml;
use G1_aux;
read all into G1_aux;
G1=vecdiag(G1_aux);
create G1 from G1;
append from G1;
quit;

Data g1;
set g1;
rename col1=g1;
run;

/* Verificação de G1 - cálculo de apenas um
valor: g1,13*/
/*
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use dataro; read all into dataro;
use Gama3; read all into Gama3;
use MatV1; read all into MatV1;
J7=J(7,1,1);
G1_13=varv + varu/(1-dataro*dataro) -
t(varv*J7 + varu*Gama3)*inv(MatV1)*(varv*J7 +
varu*Gama3);
create G1_13 from G1_13;
append from G1_13;
quit;
*/
/* Fim da verificação */

/***************************/
/****** Cálculo do g2 ******/
/***************************/

Proc iml;
use X; read all into X;
use MatV; read all into MatV;
use MatG; read all into MatG;
A=I(196);
G2_aux=t(t(X) -
t(X)*inv(MatV)*MatG*A)*inv(t(X)*inv(MatV)*X)*(
t(X) - t(X)*inv(MatV)*MatG*A);
create G2_aux from G2_aux;
append from G2_aux;
quit;

Proc iml;
use G2_aux;
read all into G2_aux;
G2=vecdiag(G2_aux);
create G2 from G2;
append from G2;
quit;

Data g2;
set g2;
rename col1=g2;
run;

/* Verificação de G2 - cálculo de apenas um
valor: g2,13*/
/*
Data lixo;
merge new X;
run;

Data X_13;
set lixo;
if id=3;
drop id count;
run;

Data X_1;
set lixo;
if id<8;
drop id count;
run;

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use dataro; read all into dataro;
use Gama3; read all into Gama3;
use MatV1; read all into MatV1;
```

```
use Matv; read all into Matv;
use X; read all into X;
use X_13; read all into X_13;
use X_1; read all into X_1;
J7=J(7,1,1);
G2_13=t(t(X_13)-
t(X_1)*inv(MatV1)*(varv*J7+varu*Gama3))*inv(t(
X)*inv(MatV)*X)*(t(X_13)-
t(X_1)*inv(MatV1)*(varv*J7+varu*Gama3));
create G2_13 from G2_13;
append from G2_13;
quit;*/
/* Fim da verificação */

/***************************/
/****** Cálculo do g3 ******/
/***************************/

/* O Cálculo do g3 tem que ser feito
individualmente para cada trimestre EM CADA
DOMÍNIO. Note-se que o vector coluna "gama"
varia em função do trimeste e que a matriz
"Vi" É DIFERENTE para diferentes domínios.
Basta calcular a matriz A para cada trimestre
de um determinado domínio, e depois replicar
para os restantes domínios, que é sempre igual
*/

/* DOMÍNIO 1 */
/* 1º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama1; read all into Gama1;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a11=t(Gama1 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama1))*inv(MatV1)*(Gama1 -
Gama*inv(MatV1)*(varv*J7 + varu*Gama1));
create a11 from a11; append from a11; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama1; read all into Gama1;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a22=t(J7 - J77*inv(MatV1)*(varv*J7 +
varu*Gama1))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama1));
create a22 from a22; append from a22; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama1; read all into Gama1;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a21=t(Gama1 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama1))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama1));
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A11_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A11_ from A11_; append from A11_; quit;

/* 2º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama2; read all into Gama2;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a11=t(Gama2 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama2))*inv(MatV1)*(Gama2 -
Gama*inv(MatV1)*(varv*J7 + varu*Gama2));
create a11 from a11; append from a11; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama2; read all into Gama2;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a22=t(J7 - J77*inv(MatV1)*(varv*J7 +
varu*Gama2))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama2));
create a22 from a22; append from a22; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama2; read all into Gama2;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a21=t(Gama2 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama2))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama2));
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A12_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A12_ from A12_; append from A12_; quit;

/* 3º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama3; read all into Gama3;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a11=t(Gama3 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama3))*inv(MatV1)*(Gama3 -
Gama*inv(MatV1)*(varv*J7 + varu*Gama3));
create a11 from a11; append from a11; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama3; read all into Gama3;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a22=t(J7 - J77*inv(MatV1)*(varv*J7 +
varu*Gama3))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama3));
create a22 from a22; append from a22; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama3; read all into Gama3;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a21=t(Gama3 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama3))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama3));
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A13_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A13_ from A13_; append from A13_; quit;
```

```
/* 4º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama4; read all into Gama4;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a11=t(Gama4 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama4))*inv(MatV1)*(Gama4 -
Gama*inv(MatV1)*(varv*J7 + varu*Gama4));
create a11 from a11; append from a11; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama4; read all into Gama4;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a22=t(J7 - J77*inv(MatV1)*(varv*J7 +
varu*Gama4))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama4));
create a22 from a22; append from a22; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama4; read all into Gama4;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a21=t(Gama4 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama4))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama4));
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A14_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A14_ from A14_; append from A14_; quit;

/* 5º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama5; read all into Gama5;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a11=t(Gama5 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama5))*inv(MatV1)*(Gama5 -
Gama*inv(MatV1)*(varv*J7 + varu*Gama5));
create a11 from a11; append from a11; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama5; read all into Gama5;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a22=t(J7 - J77*inv(MatV1)*(varv*J7 +
varu*Gama5))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama5));
create a22 from a22; append from a22; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama5; read all into Gama5;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a21=t(Gama5 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama5))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama5));
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A15_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A15_ from A15_; append from A15_; quit;

/* 6º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama6; read all into Gama6;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a11=t(Gama6 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama6))*inv(MatV1)*(Gama6 -
Gama*inv(MatV1)*(varv*J7 + varu*Gama6));
create a11 from a11; append from a11; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama6; read all into Gama6;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a22=t(J7 - J77*inv(MatV1)*(varv*J7 +
varu*Gama6))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama6));
create a22 from a22; append from a22; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama6; read all into Gama6;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a21=t(Gama6 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama6))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama6));
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A16_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A16_ from A16_; append from A16_; quit;

/* 7º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama7; read all into Gama7;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a11=t(Gama7 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama7))*inv(MatV1)*(Gama7 -
Gama*inv(MatV1)*(varv*J7 + varu*Gama7));
create a11 from a11; append from a11; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama7; read all into Gama7;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a22=t(J7 - J77*inv(MatV1)*(varv*J7 +
varu*Gama7))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama7));
create a22 from a22; append from a22; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
```

```
use Gama; read all into Gama;
use Gama7; read all into Gama7;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a21=t(Gama7 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama7))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama7));
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A17_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A17_ from A17_; append from A17_; quit;


/* DOMÍNIO 3 */  (…)
/* DOMÍNIO 4 */  (…)
/* DOMÍNIO 5 */  (…)
/* DOMÍNIO 6 */  (…)
(…)
/* DOMÍNIO 25 */  (…)
/* DOMÍNIO 26 */  (…)
/* DOMÍNIO 27 */  (…)
/* DOMÍNIO 28 */  (…)

/* *** Fim *** */

/**********************************/
/* Cálculo da matriz SIGMA* (2*2) */
/**********************************/

/* OBS.: Antes de calcular as covariâncias da
matriz SIGMA*, é necessário calcular as
matrizes C, C*, C1 e C2. Note-se que a matriz
V (que depois funciona como a matriz OMEGA já
está calculada, pois é a matriz Cov(Y) e note-
se também que não é necessário calcular a */

/* MatC */

Proc iml;
use D; read all into D;
use P; read all into P;
Ident=I(7);
IDP=(Ident - D)*P; /* Matriz (I - D)*P */
create IDP from IDP; append from IDP; quit;

Proc iml;
use IDP; read all into IDP;
blockIDP1=BLOCK(IDP,IDP,IDP,IDP,IDP,IDP,IDP,ID
P,IDP,IDP,IDP,IDP,IDP,IDP);
create blockIDP1 from blockIDP1; append from
blockIDP1; quit;
Proc iml;
use blockIDP1; read all into blockIDP1;
MatC=BLOCK(blockIDP1,blockIDP1);
create MatC from MatC; append from MatC; quit;

/* MatC* */

Proc iml;
use f; read all into f;
use P; read all into P;
use c; read all into c;
cfP1=(1/sqrt(c))*t(f)*P; /* Matriz c(-
1/2)*fT*P */
create cfP1 from cfP1; append from cfP1; quit;

Proc iml;
use cfP1; read all into cfP1;
blockcfP1=BLOCK(cfP1,cfP1,cfP1,cfP1,cfP1,cfP1,
cfP1,cfP1,cfP1,cfP1,cfP1,cfP1,cfP1,cfP1);
create blockcfP1 from blockcfP1; append from
blockcfP1; quit;
Proc iml;
use blockcfP1; read all into blockcfP1;
MatC_ast=BLOCK(blockcfP1,blockcfP1);
create MatC_ast from MatC_ast; append from
MatC_ast; quit;

/* MatC1 */

Proc iml;
use MatC; read all into MatC;
use X; read all into X;
Ident=I(196);
MatC1=t(MatC)* (Ident - MatC*X*
ginv(t(X)*t(MatC)*MatC*X) *t(X)*t(MatC))
*MatC;
create MatC1 from MatC1; append from MatC1;
quit;

/* MatC2 */

Proc iml;
use MatC_ast; read all into MatC_ast;
use X; read all into X;
Ident=I(28);
MatC2=t(MatC_ast)* (Ident - MatC_ast*X*
ginv(t(X)*t(MatC_ast)*MatC_ast*X)
*t(X)*t(MatC_ast)) *MatC_ast;
create MatC2 from MatC2; append from MatC2;
quit;

/* varu = k1* a'C1a + k2 */

Proc iml;
use caracH1; read all into caracH1;
k1= 1/(28*6 - caracH1);
create k1 from k1; append from k1; quit;

Proc iml;
use blockPP; read all into blockPP;
use blockID; read all into blockID;
use Hs1; read all into Hs1;
use k1; read all into k1;
k2= -trace((blockID - Hs1)*blockPP)*k1;
create k2 from k2; append from k2; quit;

/* varv = k3* a'C2a + k4* a'C1a + k5 */

Proc iml;
use caracH2; read all into caracH2;
use c; read all into c;
k3= 1/(c*(28 - caracH2));
create k3 from k3; append from k3; quit;

Proc iml;
use k1; read all into k1;
use c; read all into c;
k4= -(1/c)*k1;
create k4 from k4; append from k4; quit;

Proc iml;
use IF; read all into IF;
use cfp; read all into cfp;
use c; read all into c;
use k2; read all into k2;
use k3; read all into k3;
k5= -(1/c)*k2 - trace(IF*cfp)*k3;
create k5 from k5; append from k5; quit;

/* SIGMA11 = Var(varu) = 2* k1^2 * tr(C1 * V *
C1 * V) */
/* SIGMA11 */

Proc iml;
use MatC1; read all into MatC1;
use MatV; read all into MatV;
use k1; read all into k1;
Sigma11_=2*k1*k1* trace(MatC1 * MatV * MatC1 *
MatV);
create Sigma11_ from Sigma11_; append from
Sigma11_; quit;

/* SIGMA22 = Var(varv)= 2*k3^2 * tr(C2*V*C2*V)
+ 4* k3*k4* tr(C1*V*C2*V) + 2* k4^2 *
tr(C1*V*C1*V)*/
/* SIGMA22 */

Proc iml;
use MatC1; read all into MatC1;
use MatC2; read all into MatC2;
use MatV; read all into MatV;
use k3; read all into k3;
use k4; read all into k4;
Sigma22_=2*k3*k3*trace(MatC2 * MatV * MatC2 *
MatV) + 4*k3*k4*trace(MatC1 * MatV * MatC2 *
MatV) +
2*k4*k4*trace(MatC1 * MatV * MatC1 * MatV);
```

```
create Sigma22_ from Sigma22_; append from
Sigma22_; quit;

/* SIGMA21 = Cov(varu,varv)= 2*k1*k3 *
tr(C1*V*C2*V) + 2* k1*k4* tr(C1*V*C1*V) */
/* SIGMA21 */

Proc iml;
use MatC1; read all into MatC1;
use MatC2; read all into MatC2;
use MatV; read all into MatV;
use k1; read all into k1;
use k3; read all into k3;
use k4; read all into k4;
Sigma21_=2*k1*k3* trace(MatC1 * MatV * MatC2 *
MatV) + 2*k1*k4* trace(MatC1 * MatV * MatC1 *
MatV);
create Sigma21_ from Sigma21_; append from
Sigma21_; quit;

Proc iml;
use Sigma11_; read all into Sigma11_;
use Sigma22_; read all into Sigma22_;
use Sigma21_; read all into Sigma21_;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
SIGMA=Sigma11_*I1 + Sigma21_*I2 + Sigma21_*I3
+ Sigma22_*I4;
create SIGMA from SIGMA;
append from SIGMA; quit;

/* *** */

/* Cálculo do trace(Ait * SIGMA) */

Proc iml;
use SIGMA; read all into SIGMA;
use A11_; read all into A11_;
use A12_; read all into A12_;
use A13_; read all into A13_;
use A14_; read all into A14_;
use A15_; read all into A15_;
use A16_; read all into A16_;
use A17_; read all into A17_;
use A21_; read all into A21_;
use A22_; read all into A22_;
use A23_; read all into A23_;
use A24_; read all into A24_;
use A25_; read all into A25_;
use A26_; read all into A26_;
use A27_; read all into A27_;
use A31_; read all into A31_;
use A32_; read all into A32_;
use A33_; read all into A33_;
use A34_; read all into A34_;
use A35_; read all into A35_;
use A36_; read all into A36_;
use A37_; read all into A37_;
use A41_; read all into A41_;
use A42_; read all into A42_;
use A43_; read all into A43_;
use A44_; read all into A44_;
use A45_; read all into A45_;
use A46_; read all into A46_;
use A47_; read all into A47_;
use A51_; read all into A51_;
use A52_; read all into A52_;
use A53_; read all into A53_;
use A54_; read all into A54_;
use A55_; read all into A55_;
use A56_; read all into A56_;
use A57_; read all into A57_;
use A61_; read all into A61_;
use A62_; read all into A62_;
use A63_; read all into A63_;
use A64_; read all into A64_;
use A65_; read all into A65_;
use A66_; read all into A66_;
use A67_; read all into A67_;
use A71_; read all into A71_;
use A72_; read all into A72_;
use A73_; read all into A73_;
use A74_; read all into A74_;
use A75_; read all into A75_;
use A76_; read all into A76_;
use A77_; read all into A77_;
use A81_; read all into A81_;
use A82_; read all into A82_;
use A83_; read all into A83_;
use A84_; read all into A84_;
use A85_; read all into A85_;
use A86_; read all into A86_;
use A87_; read all into A87_;
use A91_; read all into A91_;
use A92_; read all into A92_;
use A93_; read all into A93_;
use A94_; read all into A94_;
use A95_; read all into A95_;
use A96_; read all into A96_;
use A97_; read all into A97_;
use A101_; read all into A101_;
use A102_; read all into A102_;
use A103_; read all into A103_;
use A104_; read all into A104_;
use A105_; read all into A105_;
use A106_; read all into A106_;
use A107_; read all into A107_;
G3_11=trace(A11_ * SIGMA);
G3_12=trace(A12_ * SIGMA);
G3_13=trace(A13_ * SIGMA);
G3_14=trace(A14_ * SIGMA);
G3_15=trace(A15_ * SIGMA);
G3_16=trace(A16_ * SIGMA);
G3_17=trace(A17_ * SIGMA);
G3_21=trace(A21_ * SIGMA);
G3_22=trace(A22_ * SIGMA);
G3_23=trace(A23_ * SIGMA);
G3_24=trace(A24_ * SIGMA);
G3_25=trace(A25_ * SIGMA);
G3_26=trace(A26_ * SIGMA);
G3_27=trace(A27_ * SIGMA);
G3_31=trace(A31_ * SIGMA);
G3_32=trace(A32_ * SIGMA);
G3_33=trace(A33_ * SIGMA);
G3_34=trace(A34_ * SIGMA);
G3_35=trace(A35_ * SIGMA);
G3_36=trace(A36_ * SIGMA);
G3_37=trace(A37_ * SIGMA);
G3_41=trace(A41_ * SIGMA);
G3_42=trace(A42_ * SIGMA);
G3_43=trace(A43_ * SIGMA);
G3_44=trace(A44_ * SIGMA);
G3_45=trace(A45_ * SIGMA);
G3_46=trace(A46_ * SIGMA);
G3_47=trace(A47_ * SIGMA);
G3_51=trace(A51_ * SIGMA);
G3_52=trace(A52_ * SIGMA);
G3_53=trace(A53_ * SIGMA);
G3_54=trace(A54_ * SIGMA);
G3_55=trace(A55_ * SIGMA);
G3_56=trace(A56_ * SIGMA);
G3_57=trace(A57_ * SIGMA);
G3_61=trace(A61_ * SIGMA);
G3_62=trace(A62_ * SIGMA);
G3_63=trace(A63_ * SIGMA);
G3_64=trace(A64_ * SIGMA);
G3_65=trace(A65_ * SIGMA);
G3_66=trace(A66_ * SIGMA);
G3_67=trace(A67_ * SIGMA);
G3_71=trace(A71_ * SIGMA);
G3_72=trace(A72_ * SIGMA);
G3_73=trace(A73_ * SIGMA);
G3_74=trace(A74_ * SIGMA);
G3_75=trace(A75_ * SIGMA);
G3_76=trace(A76_ * SIGMA);
G3_77=trace(A77_ * SIGMA);
G3_81=trace(A81_ * SIGMA);
G3_82=trace(A82_ * SIGMA);
G3_83=trace(A83_ * SIGMA);
G3_84=trace(A84_ * SIGMA);
G3_85=trace(A85_ * SIGMA);
G3_86=trace(A86_ * SIGMA);
G3_87=trace(A87_ * SIGMA);
G3_91=trace(A91_ * SIGMA);
G3_92=trace(A92_ * SIGMA);
G3_93=trace(A93_ * SIGMA);
G3_94=trace(A94_ * SIGMA);
G3_95=trace(A95_ * SIGMA);
G3_96=trace(A96_ * SIGMA);
G3_97=trace(A97_ * SIGMA);
G3_101=trace(A101_ * SIGMA);
G3_102=trace(A102_ * SIGMA);
G3_103=trace(A103_ * SIGMA);
G3_104=trace(A104_ * SIGMA);
```

```
G3_105=trace(A105_ * SIGMA);
G3_106=trace(A106_ * SIGMA);
G3_107=trace(A107_ * SIGMA);
G3_aux1=G3_11//G3_12//G3_13//G3_14//G3_15//G3_
16//G3_17//G3_21//G3_22//G3_23//G3_24//G3_25//
G3_26//G3_27//G3_31//G3_32//G3_33//G3_34//G3_3
5//G3_36//G3_37//G3_41//G3_42//G3_43//G3_44//G
3_45//G3_46//G3_47//G3_51//G3_52//G3_53//G3_54
//G3_55//G3_56//G3_57//G3_61//G3_62//G3_63//G3
_64//G3_65//G3_66//G3_67//G3_71//G3_72//G3_73/
/G3_74//G3_75//G3_76//G3_77//G3_81//G3_82//G3_
83//G3_84//G3_85//G3_86//G3_87//G3_91//G3_92//
G3_93//G3_94//G3_95//G3_96//G3_97//G3_101//G3_
102//G3_103//G3_104//G3_105//G3_106//G3_107;
create G3_aux1 from G3_aux1; append from
G3_aux1; quit;

Proc iml;
use SIGMA; read all into SIGMA;
use A111_; read all into A111_;
use A112_; read all into A112_;
use A113_; read all into A113_;
use A114_; read all into A114_;
use A115_; read all into A115_;
use A116_; read all into A116_;
use A117_; read all into A117_;
use A121_; read all into A121_;
use A122_; read all into A122_;
use A123_; read all into A123_;
use A124_; read all into A124_;
use A125_; read all into A125_;
use A126_; read all into A126_;
use A127_; read all into A127_;
use A131_; read all into A131_;
use A132_; read all into A132_;
use A133_; read all into A133_;
use A134_; read all into A134_;
use A135_; read all into A135_;
use A136_; read all into A136_;
use A137_; read all into A137_;
use A141_; read all into A141_;
use A142_; read all into A142_;
use A143_; read all into A143_;
use A144_; read all into A144_;
use A145_; read all into A145_;
use A146_; read all into A146_;
use A147_; read all into A147_;
use A151_; read all into A151_;
use A152_; read all into A152_;
use A153_; read all into A153_;
use A154_; read all into A154_;
use A155_; read all into A155_;
use A156_; read all into A156_;
use A157_; read all into A157_;
use A161_; read all into A161_;
use A162_; read all into A162_;
use A163_; read all into A163_;
use A164_; read all into A164_;
use A165_; read all into A165_;
use A166_; read all into A166_;
use A167_; read all into A167_;
use A171_; read all into A171_;
use A172_; read all into A172_;
use A173_; read all into A173_;
use A174_; read all into A174_;
use A175_; read all into A175_;
use A176_; read all into A176_;
use A177_; read all into A177_;
use A181_; read all into A181_;
use A182_; read all into A182_;
use A183_; read all into A183_;
use A184_; read all into A184_;
use A185_; read all into A185_;
use A186_; read all into A186_;
use A187_; read all into A187_;
use A191_; read all into A191_;
use A192_; read all into A192_;
use A193_; read all into A193_;
use A194_; read all into A194_;
use A195_; read all into A195_;
use A196_; read all into A196_;
use A197_; read all into A197_;
use A201_; read all into A201_;
use A202_; read all into A202_;
use A203_; read all into A203_;
use A204_; read all into A204_;
use A205_; read all into A205_;
use A206_; read all into A206_;
use A207_; read all into A207_;
G3_111=trace(A111_ * SIGMA);
G3_112=trace(A112_ * SIGMA);
G3_113=trace(A113_ * SIGMA);
G3_114=trace(A114_ * SIGMA);
G3_115=trace(A115_ * SIGMA);
G3_116=trace(A116_ * SIGMA);
G3_117=trace(A117_ * SIGMA);
G3_121=trace(A121_ * SIGMA);
G3_122=trace(A122_ * SIGMA);
G3_123=trace(A123_ * SIGMA);
G3_124=trace(A124_ * SIGMA);
G3_125=trace(A125_ * SIGMA);
G3_126=trace(A126_ * SIGMA);
G3_127=trace(A127_ * SIGMA);
G3_131=trace(A131_ * SIGMA);
G3_132=trace(A132_ * SIGMA);
G3_133=trace(A133_ * SIGMA);
G3_134=trace(A134_ * SIGMA);
G3_135=trace(A135_ * SIGMA);
G3_136=trace(A136_ * SIGMA);
G3_137=trace(A137_ * SIGMA);
G3_141=trace(A141_ * SIGMA);
G3_142=trace(A142_ * SIGMA);
G3_143=trace(A143_ * SIGMA);
G3_144=trace(A144_ * SIGMA);
G3_145=trace(A145_ * SIGMA);
G3_146=trace(A146_ * SIGMA);
G3_147=trace(A147_ * SIGMA);
G3_151=trace(A151_ * SIGMA);
G3_152=trace(A152_ * SIGMA);
G3_153=trace(A153_ * SIGMA);
G3_154=trace(A154_ * SIGMA);
G3_155=trace(A155_ * SIGMA);
G3_156=trace(A156_ * SIGMA);
G3_157=trace(A157_ * SIGMA);
G3_161=trace(A161_ * SIGMA);
G3_162=trace(A162_ * SIGMA);
G3_163=trace(A163_ * SIGMA);
G3_164=trace(A164_ * SIGMA);
G3_165=trace(A165_ * SIGMA);
G3_166=trace(A166_ * SIGMA);
G3_167=trace(A167_ * SIGMA);
G3_171=trace(A171_ * SIGMA);
G3_172=trace(A172_ * SIGMA);
G3_173=trace(A173_ * SIGMA);
G3_174=trace(A174_ * SIGMA);
G3_175=trace(A175_ * SIGMA);
G3_176=trace(A176_ * SIGMA);
G3_177=trace(A177_ * SIGMA);
G3_181=trace(A181_ * SIGMA);
G3_182=trace(A182_ * SIGMA);
G3_183=trace(A183_ * SIGMA);
G3_184=trace(A184_ * SIGMA);
G3_185=trace(A185_ * SIGMA);
G3_186=trace(A186_ * SIGMA);
G3_187=trace(A187_ * SIGMA);
G3_191=trace(A191_ * SIGMA);
G3_192=trace(A192_ * SIGMA);
G3_193=trace(A193_ * SIGMA);
G3_194=trace(A194_ * SIGMA);
G3_195=trace(A195_ * SIGMA);
G3_196=trace(A196_ * SIGMA);
G3_197=trace(A197_ * SIGMA);
G3_201=trace(A201_ * SIGMA);
G3_202=trace(A202_ * SIGMA);
G3_203=trace(A203_ * SIGMA);
G3_204=trace(A204_ * SIGMA);
G3_205=trace(A205_ * SIGMA);
G3_206=trace(A206_ * SIGMA);
G3_207=trace(A207_ * SIGMA);
G3_aux2=G3_111//G3_112//G3_113//G3_114//G3_115
//G3_116//G3_117//G3_121//G3_122//G3_123//G3_1
24//G3_125//G3_126//G3_127//G3_131//G3_132//G3
_133//G3_134//G3_135//G3_136//G3_137//G3_141//
G3_142//G3_143//G3_144//G3_145//G3_146//G3_147
//G3_151//G3_152//G3_153//G3_154//G3_155//G3_1
56//G3_157//G3_161//G3_162//G3_163//G3_164//G3
_165//G3_166//G3_167//G3_171//G3_172//G3_173//
G3_174//G3_175//G3_176//G3_177//G3_181//G3_182
//G3_183//G3_184//G3_185//G3_186//G3_187//G3_1
91//G3_192//G3_193//G3_194//G3_195//G3_196//G3
_197//G3_201//G3_202//G3_203//G3_204//G3_205//
G3_206//G3_207;
create G3_aux2 from G3_aux2; append from
G3_aux2; quit;

Proc iml;
```

```
use SIGMA; read all into SIGMA;
use A211_; read all into A211_;
use A212_; read all into A212_;
use A213_; read all into A213_;
use A214_; read all into A214_;
use A215_; read all into A215_;
use A216_; read all into A216_;
use A217_; read all into A217_;
use A221_; read all into A221_;
use A222_; read all into A222_;
use A223_; read all into A223_;
use A224_; read all into A224_;
use A225_; read all into A225_;
use A226_; read all into A226_;
use A227_; read all into A227_;
use A231_; read all into A231_;
use A232_; read all into A232_;
use A233_; read all into A233_;
use A234_; read all into A234_;
use A235_; read all into A235_;
use A236_; read all into A236_;
use A237_; read all into A237_;
use A241_; read all into A241_;
use A242_; read all into A242_;
use A243_; read all into A243_;
use A244_; read all into A244_;
use A245_; read all into A245_;
use A246_; read all into A246_;
use A247_; read all into A247_;
use A251_; read all into A251_;
use A252_; read all into A252_;
use A253_; read all into A253_;
use A254_; read all into A254_;
use A255_; read all into A255_;
use A256_; read all into A256_;
use A257_; read all into A257_;
use A261_; read all into A261_;
use A262_; read all into A262_;
use A263_; read all into A263_;
use A264_; read all into A264_;
use A265_; read all into A265_;
use A266_; read all into A266_;
use A267_; read all into A267_;
use A271_; read all into A271_;
use A272_; read all into A272_;
use A273_; read all into A273_;
use A274_; read all into A274_;
use A275_; read all into A275_;
use A276_; read all into A276_;
use A277_; read all into A277_;
use A281_; read all into A281_;
use A282_; read all into A282_;
use A283_; read all into A283_;
use A284_; read all into A284_;
use A285_; read all into A285_;
use A286_; read all into A286_;
use A287_; read all into A287_;
G3_211=trace(A211_ * SIGMA);
G3_212=trace(A212_ * SIGMA);
G3_213=trace(A213_ * SIGMA);
G3_214=trace(A214_ * SIGMA);
G3_215=trace(A215_ * SIGMA);
G3_216=trace(A216_ * SIGMA);
G3_217=trace(A217_ * SIGMA);
G3_221=trace(A221_ * SIGMA);
G3_222=trace(A222_ * SIGMA);
G3_223=trace(A223_ * SIGMA);
G3_224=trace(A224_ * SIGMA);
G3_225=trace(A225_ * SIGMA);
G3_226=trace(A226_ * SIGMA);
G3_227=trace(A227_ * SIGMA);
G3_231=trace(A231_ * SIGMA);
G3_232=trace(A232_ * SIGMA);
G3_233=trace(A233_ * SIGMA);
G3_234=trace(A234_ * SIGMA);
G3_235=trace(A235_ * SIGMA);
G3_236=trace(A236_ * SIGMA);
G3_237=trace(A237_ * SIGMA);
G3_241=trace(A241_ * SIGMA);
G3_242=trace(A242_ * SIGMA);
G3_243=trace(A243_ * SIGMA);
G3_244=trace(A244_ * SIGMA);
G3_245=trace(A245_ * SIGMA);
G3_246=trace(A246_ * SIGMA);
G3_247=trace(A247_ * SIGMA);
G3_251=trace(A251_ * SIGMA);
G3_252=trace(A252_ * SIGMA);
G3_253=trace(A253_ * SIGMA);
G3_254=trace(A254_ * SIGMA);
G3_255=trace(A255_ * SIGMA);
G3_256=trace(A256_ * SIGMA);
G3_257=trace(A257_ * SIGMA);
G3_261=trace(A261_ * SIGMA);
G3_262=trace(A262_ * SIGMA);
G3_263=trace(A263_ * SIGMA);
G3_264=trace(A264_ * SIGMA);
G3_265=trace(A265_ * SIGMA);
G3_266=trace(A266_ * SIGMA);
G3_267=trace(A267_ * SIGMA);
G3_271=trace(A271_ * SIGMA);
G3_272=trace(A272_ * SIGMA);
G3_273=trace(A273_ * SIGMA);
G3_274=trace(A274_ * SIGMA);
G3_275=trace(A275_ * SIGMA);
G3_276=trace(A276_ * SIGMA);
G3_277=trace(A277_ * SIGMA);
G3_281=trace(A281_ * SIGMA);
G3_282=trace(A282_ * SIGMA);
G3_283=trace(A283_ * SIGMA);
G3_284=trace(A284_ * SIGMA);
G3_285=trace(A285_ * SIGMA);
G3_286=trace(A286_ * SIGMA);
G3_287=trace(A287_ * SIGMA);
G3_aux3=G3_211//G3_212//G3_213//G3_214//G3_215
//G3_216//G3_217//G3_221//G3_222//G3_223//G3_2
24//G3_225//G3_226//G3_227//G3_231//G3_232//G3
_233//G3_234//G3_235//G3_236//G3_237//G3_241//
G3_242//G3_243//G3_244//G3_245//G3_246//G3_247
//G3_251//G3_252//G3_253//G3_254//G3_255//G3_2
56//G3_257//G3_261//G3_262//G3_263//G3_264//G3
_265//G3_266//G3_267//G3_271//G3_272//G3_273//
G3_274//G3_275//G3_276//G3_277//G3_281//G3_282
//G3_283//G3_284//G3_285//G3_286//G3_287;
create G3_aux3 from G3_aux3; append from
G3_aux3; quit;

Data G3_aux;
set G3_aux1 G3_aux2 G3_aux3;
run;

Data g3;
set g3_aux;
rename col1=g3; run;

/* *** Fim *** */

/* **** Estimativa do EQMP do EBLUP *****/

Data Estimat_RY;
merge crono_resf g1 g2 g3;
EQM_EBLUP= G1 + G2 + 2*G3;
CV=sqrt(EQM_EBLUP)/e;
simul=&i;
run;

Proc sort data=Estimat_RY out=Estimat_RY;
by t i;
run;

Proc append base=HB.Est_RY_simulacao
data=Estimat_RY;
run;

DM 'CLEAR OUTPUT';
DM 'CLEAR LOG';

%mend;
%inc 'C:\Documents and Settings\Luis Pereira
\My Documents\EstudoEmpirico\chamamacro.txt';

/*******************************************/
/*FIM DA SIMULAÇÃO COM ESTIMADOR EBLUP DE RY*/
/*******************************************/

/*******************************************
*******************************************/
/********** MODELO DE LUIS PEREIRA **********/
/*******************************************
*******************************************/

/*É utilizado o modelo espaciotemoral de Luis
Pereira e o EQMP do EBLUP é estimado pela
aproximação analítica.
Contudo, admite-se que os coeficientes de
associação espacial e de autocorrelação
temporal são conhecidos. */
```

```
/* Dados: Hb.Estimat_directas_pseudo */
/* Os resultados estão guardados no ficheiro:
Hb.Est_LP_simulacao */

Proc sort data=Hb.Estimat_directas_pseudo
out=A5;
by simul i t;
run;

Proc sort
data=Hb.Estimat_directasnutsii_pseudo
out=Estimat_directasnutsii_pseudo;
by simul cod_nut_ii_trans t;
run;

Proc sort data=Hb.Estimat_directasnutsi_pseudo
out=Estimat_directasnutsi_pseudo;
by simul t; run;

Data null;
file 'C:\Documents and Settings\Luis Pereira
\My Documents \EstudoEmpirico\chamamacro.txt';
   do i= 1 to 1000; k=i;
var='%ext_am('||i||','||k||');'; put var; end;
run;

%macro ext_am(i,k);

Data A6;
set A5;
if simul=&i;
run;

/* Dependência espacial */

Data dataphi;
phi=0.293;
run;

/* Autocorrelação temporal */

Data dataro;
ro=0.37;
run;

/* Matriz B=(I-phi*W)'*(I-phi*W) */

Data W;
set Hb.W1;   /* Matriz de distâncias */
run;

Proc iml;
use W; read all into W;
use dataphi; read all into dataphi;
Ident=I(28);
B = t(Ident - dataphi*W)*(Ident - dataphi*W);
create B from B; append from B; quit;

Proc iml;
use B; read all into B;
Binv=inv(B);   /* Inversa de B */
create Binv from Binv; append from Binv; quit;

/* Matriz P */

Data dataro2;
set dataro;
ro2= sqrt(1-ro*ro);
keep ro2;
run;

Proc iml;
use dataro;
read all into dataro;
use dataro2;
read all into dataro2;
I1={1 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I2={0 0 0 0 0 0 0, 0 1 0 0 0 0 0, 0 0 1 0 0 0
0, 0 0 0 1 0 0 0, 0 0 0 0 1 0 0, 0 0 0 0 0 1
0, 0 0 0 0 0 0 1};
I4={0 0 0 0 0 0 0, 1 0 0 0 0 0 0, 0 1 0 0 0 0
0, 0 0 1 0 0 0 0, 0 0 0 1 0 0 0, 0 0 0 0 1 0
0, 0 0 0 0 0 1 0};
P = dataro2*I1 + I2 - dataro*I4;
create P from P;
```

```
append from P;
quit;

Proc iml;
use dataro;
read all into dataro;
use dataro2;
read all into dataro2;
C1={1, 0, 0, 0, 0, 0, 0};
C2={0, 1, 1, 1, 1, 1, 1};
f = dataro2*C1 + C2 - dataro*C2;
create f from f;
append from f;
quit;

Proc iml;
use f;
read all into f;
c=f`*f;
create c from c;
append from c;
quit;

Proc iml;
use f;
read all into f;
use c;
read all into c;
D=(1/c)*f*f`;
create D from D;
append from D;
quit;

/* *** */

Data iabh;
set a6;
keep i t c x;
run;

Data ipth;
set a6;
keep i t y;
run;

Data iabhipth1;
set a6;
keep i t y x;
run;

/** Preparação do ficheiro das observações da
variável de interesse **/

Proc sort data=ipth;
by i t; run;

Data tetad1; set ipth; where i= 1; keep y;
Data tetad2; set ipth; where i= 2; keep y;
Data tetad3; set ipth; where i= 3; keep y;
Data tetad4; set ipth; where i= 4; keep y;
Data tetad5; set ipth; where i= 5; keep y;
Data tetad6; set ipth; where i= 6; keep y;
Data tetad7; set ipth; where i= 7; keep y;
Data tetad8; set ipth; where i= 8; keep y;
Data tetad9; set ipth; where i= 9; keep y;
Data tetad10; set ipth; where i= 10; keep y;
Data tetad11; set ipth; where i= 11; keep y;
Data tetad12; set ipth; where i= 12; keep y;
Data tetad13; set ipth; where i= 13; keep y;
Data tetad14; set ipth; where i= 14; keep y;
Data tetad15; set ipth; where i= 15; keep y;
Data tetad16; set ipth; where i= 16; keep y;
Data tetad17; set ipth; where i= 17; keep y;
Data tetad18; set ipth; where i= 18; keep y;
Data tetad19; set ipth; where i= 19; keep y;
Data tetad20; set ipth; where i= 20; keep y;
Data tetad21; set ipth; where i= 21; keep y;
Data tetad22; set ipth; where i= 22; keep y;
Data tetad23; set ipth; where i= 23; keep y;
Data tetad24; set ipth; where i= 24; keep y;
Data tetad25; set ipth; where i= 25; keep y;
Data tetad26; set ipth; where i= 26; keep y;
Data tetad27; set ipth; where i= 27; keep y;
Data tetad28; set ipth; where i= 28; keep y;
run;
/** Preparação do ficheiro das observações da
variável auxiliar **/
```

```
Proc sort data=iabh;
by i t; run;

Proc sql; create table iabh0 as select c, x,
i, t from iabh;

Data xd1; set iabh0; where i= 1; keep c x;
Data xd2; set iabh0; where i= 2; keep c x;
Data xd3; set iabh0; where i= 3; keep c x;
Data xd4; set iabh0; where i= 4; keep c x;
Data xd5; set iabh0; where i= 5; keep c x;
Data xd6; set iabh0; where i= 6; keep c x;
Data xd7; set iabh0; where i= 7; keep c x;
Data xd8; set iabh0; where i= 8; keep c x;
Data xd9; set iabh0; where i= 9; keep c x;
Data xd10; set iabh0; where i= 10; keep c x;
Data xd11; set iabh0; where i= 11; keep c x;
Data xd12; set iabh0; where i= 12; keep c x;
Data xd13; set iabh0; where i= 13; keep c x;
Data xd14; set iabh0; where i= 14; keep c x;
Data xd15; set iabh0; where i= 15; keep c x;
Data xd16; set iabh0; where i= 16; keep c x;
Data xd17; set iabh0; where i= 17; keep c x;
Data xd18; set iabh0; where i= 18; keep c x;
Data xd19; set iabh0; where i= 19; keep c x;
Data xd20; set iabh0; where i= 20; keep c x;
Data xd21; set iabh0; where i= 21; keep c x;
Data xd22; set iabh0; where i= 22; keep c x;
Data xd23; set iabh0; where i= 23; keep c x;
Data xd24; set iabh0; where i= 24; keep c x;
Data xd25; set iabh0; where i= 25; keep c x;
Data xd26; set iabh0; where i= 26; keep c x;
Data xd27; set iabh0; where i= 27; keep c x;
Data xd28; set iabh0; where i= 28; keep c x;
run;

/** Cálculo da matriz Z(1) **/

Proc iml;
use tetad1; read all into tetad1;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd1=(Ident - D)*P*tetad1;
create zd1 from zd1; append from zd1; quit;
Proc iml;
use tetad2; read all into tetad2;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd2=(Ident - D)*P*tetad2;
create zd2 from zd2; append from zd2; quit;
Proc iml;
use tetad3; read all into tetad3;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd3=(Ident - D)*P*tetad3;
create zd3 from zd3; append from zd3; quit;
Proc iml;
use tetad4; read all into tetad4;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd4=(Ident - D)*P*tetad4;
create zd4 from zd4; append from zd4; quit;
Proc iml;
use tetad5; read all into tetad5;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd5=(Ident - D)*P*tetad5;
create zd5 from zd5; append from zd5; quit;
Proc iml;
use tetad6; read all into tetad6;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd6=(Ident - D)*P*tetad6;
create zd6 from zd6; append from zd6; quit;
Proc iml;
use tetad7; read all into tetad7;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd7=(Ident - D)*P*tetad7;
create zd7 from zd7; append from zd7; quit;
Proc iml;
use tetad8; read all into tetad8;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd8=(Ident - D)*P*tetad8;
create zd8 from zd8; append from zd8; quit;
Proc iml;
use tetad9; read all into tetad9;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd9=(Ident - D)*P*tetad9;
create zd9 from zd9; append from zd9; quit;
Proc iml;
use tetad10; read all into tetad10;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd10=(Ident - D)*P*tetad10;
create zd10 from zd10; append from zd10; quit;
Proc iml;
use tetad11; read all into tetad11;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd11=(Ident - D)*P*tetad11;
create zd11 from zd11; append from zd11; quit;
Proc iml;
use tetad12; read all into tetad12;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd12=(Ident - D)*P*tetad12;
create zd12 from zd12; append from zd12; quit;
Proc iml;
use tetad13; read all into tetad13;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd13=(Ident - D)*P*tetad13;
create zd13 from zd13; append from zd13; quit;
Proc iml;
use tetad14; read all into tetad14;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd14=(Ident - D)*P*tetad14;
create zd14 from zd14; append from zd14; quit;
Proc iml;
use tetad15; read all into tetad15;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd15=(Ident - D)*P*tetad15;
create zd15 from zd15; append from zd15; quit;
Proc iml;
use tetad16; read all into tetad16;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd16=(Ident - D)*P*tetad16;
create zd16 from zd16; append from zd16; quit;
Proc iml;
use tetad17; read all into tetad17;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd17=(Ident - D)*P*tetad17;
create zd17 from zd17; append from zd17; quit;
Proc iml;
use tetad18; read all into tetad18;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd18=(Ident - D)*P*tetad18;
create zd18 from zd18; append from zd18; quit;
Proc iml;
use tetad19; read all into tetad19;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd19=(Ident - D)*P*tetad19;
create zd19 from zd19; append from zd19; quit;
Proc iml;
use tetad20; read all into tetad20;
use P; read all into P;
use D; read all into D;
Ident=I(7);
```

```
zd20=(Ident - D)*P*tetad20;
create zd20 from zd20; append from zd20; quit;
Proc iml;
use tetad21; read all into tetad21;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd21=(Ident - D)*P*tetad21;
create zd21 from zd21; append from zd21; quit;
Proc iml;
use tetad22; read all into tetad22;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd22=(Ident - D)*P*tetad22;
create zd22 from zd22; append from zd22; quit;
Proc iml;
use tetad23; read all into tetad23;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd23=(Ident - D)*P*tetad23;
create zd23 from zd23; append from zd23; quit;
Proc iml;
use tetad24; read all into tetad24;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd24=(Ident - D)*P*tetad24;
create zd24 from zd24; append from zd24; quit;
Proc iml;
use tetad25; read all into tetad25;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd25=(Ident - D)*P*tetad25;
create zd25 from zd25; append from zd25; quit;
Proc iml;
use tetad26; read all into tetad26;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd26=(Ident - D)*P*tetad26;
create zd26 from zd26; append from zd26; quit;
Proc iml;
use tetad27; read all into tetad27;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd27=(Ident - D)*P*tetad27;
create zd27 from zd27; append from zd27; quit;
Proc iml;
use tetad28; read all into tetad28;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd28=(Ident - D)*P*tetad28;
create zd28 from zd28; append from zd28; quit;

Data Z1;
set zd1 zd2 zd3 zd4 zd5 zd6 zd7 zd8 zd9 zd10
zd11 zd12 zd13 zd14 zd15
zd16 zd17 zd18 zd19 zd20 zd21 zd22 zd23 zd24
zd25 zd26 zd27 zd28;
run;

/** Cálculo da matriz H(1) **/

Proc iml;
use xd1; read all into xd1;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd1=(Ident - D)*P*xd1;
create hd1 from hd1; append from hd1; quit;
Proc iml;
use xd2; read all into xd2;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd2=(Ident - D)*P*xd2;
create hd2 from hd2; append from hd2; quit;
Proc iml;
use xd3; read all into xd3;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd3=(Ident - D)*P*xd3;
```

```
create hd3 from hd3; append from hd3; quit;
Proc iml;
use xd4; read all into xd4;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd4=(Ident - D)*P*xd4;
create hd4 from hd4; append from hd4; quit;
Proc iml;
use xd5; read all into xd5;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd5=(Ident - D)*P*xd5;
create hd5 from hd5; append from hd5; quit;
Proc iml;
use xd6; read all into xd6;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd6=(Ident - D)*P*xd6;
create hd6 from hd6; append from hd6; quit;
Proc iml;
use xd7; read all into xd7;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd7=(Ident - D)*P*xd7;
create hd7 from hd7; append from hd7; quit;
Proc iml;
use xd8; read all into xd8;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd8=(Ident - D)*P*xd8;
create hd8 from hd8; append from hd8; quit;
Proc iml;
use xd9; read all into xd9;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd9=(Ident - D)*P*xd9;
create hd9 from hd9; append from hd9; quit;
Proc iml;
use xd10; read all into xd10;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd10=(Ident - D)*P*xd10;
create hd10 from hd10; append from hd10; quit;
Proc iml;
use xd11; read all into xd11;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd11=(Ident - D)*P*xd11;
create hd11 from hd11; append from hd11; quit;
Proc iml;
use xd12; read all into xd12;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd12=(Ident - D)*P*xd12;
create hd12 from hd12; append from hd12; quit;
Proc iml;
use xd13; read all into xd13;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd13=(Ident - D)*P*xd13;
create hd13 from hd13; append from hd13; quit;
Proc iml;
use xd14; read all into xd14;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd14=(Ident - D)*P*xd14;
create hd14 from hd14; append from hd14; quit;
Proc iml;
use xd15; read all into xd15;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd15=(Ident - D)*P*xd15;
create hd15 from hd15; append from hd15; quit;
Proc iml;
use xd16; read all into xd16;
use P; read all into P;
```

```
use D; read all into D;
Ident=I(7);
hd16=(Ident - D)*P*xd16;
create hd16 from hd16; append from hd16; quit;
Proc iml;
use xd17; read all into xd17;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd17=(Ident - D)*P*xd17;
create hd17 from hd17; append from hd17; quit;
Proc iml;
use xd18; read all into xd18;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd18=(Ident - D)*P*xd18;
create hd18 from hd18; append from hd18; quit;
Proc iml;
use xd19; read all into xd19;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd19=(Ident - D)*P*xd19;
create hd19 from hd19; append from hd19; quit;
Proc iml;
use xd20; read all into xd20;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd20=(Ident - D)*P*xd20;
create hd20 from hd20; append from hd20; quit;
Proc iml;
use xd21; read all into xd21;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd21=(Ident - D)*P*xd21;
create hd21 from hd21; append from hd21; quit;
Proc iml;
use xd22; read all into xd22;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd22=(Ident - D)*P*xd22;
create hd22 from hd22; append from hd22; quit;
Proc iml;
use xd23; read all into xd23;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd23=(Ident - D)*P*xd23;
create hd23 from hd23; append from hd23; quit;
Proc iml;
use xd24; read all into xd24;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd24=(Ident - D)*P*xd24;
create hd24 from hd24; append from hd24; quit;
Proc iml;
use xd25; read all into xd25;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd25=(Ident - D)*P*xd25;
create hd25 from hd25; append from hd25; quit;
Proc iml;
use xd26; read all into xd26;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd26=(Ident - D)*P*xd26;
create hd26 from hd26; append from hd26; quit;
Proc iml;
use xd27; read all into xd27;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd27=(Ident - D)*P*xd27;
create hd27 from hd27; append from hd27; quit;
Proc iml;
use xd28; read all into xd28;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd28=(Ident - D)*P*xd28;
create hd28 from hd28; append from hd28; quit;
```

```
Data H1;
set hd1 hd2 hd3 hd4 hd5 hd6 hd7 hd8 hd9 hd10
hd11 hd12 hd13 hd14 hd15
hd16 hd17 hd18 hd19 hd20 hd21 hd22 hd23 hd24
hd25 hd26 hd27 hd28;
run;

/** Regressão de Z(1) sobre H(1) **/

Data Z1;
set Z1;
rename col1=col0; run;

Data reg;
merge Z1 H1; run;

proc reg data=reg noprint;
        model col0=col1 col2 / noint;
        output out=rg1  residual=r1;
run; quit;

Data rg1;
set rg1;
qr1=r1*r1;
Keep qr1;
run;

Proc means data=rg1 sum noprint;
var qr1;
output out=rg1_ sum=sqr1;
run;

Data rg1_;
set rg1_;
keep sqr1;
run;

/** Cálculo da variância sigma2_u **/

Proc iml;
use H1; read all into H1;
Hs1=H1*ginv(t(H1)*H1)*t(H1); /* ginv(A) is the
Moore-Penrose generalized inverse matrix of A
*/
create Hs1 from Hs1; append from Hs1; quit; /*
Hs1 é a matriz H*(1) */

Proc iml;
use D; read all into D;
Ident=I(7);
ID=Ident - D; /* Matriz I - D */
create ID from ID; append from ID; quit;

Proc iml;
use ID; read all into ID;
blockID1=BLOCK(ID,ID,ID,ID,ID,ID,ID,ID,ID,ID,I
D,ID,ID,ID);
create blockID1 from blockID1; append from
blockID1; quit;
Proc iml;
use blockID1; read all into blockID1;
blockID=BLOCK(blockID1,blockID1);
create blockID from blockID; append from
blockID; quit;

Data s1;
set a6;
keep var t i; run;

Proc sort data=s1;
by i t; run;

Data Sigma_1;
set s1;
if i=1;
keep var;
run;
Proc iml;
use Sigma_1; read all into Sigma_1;
Sigma1=diag(Sigma_1);
create Sigma1 from Sigma1; append from Sigma1;
quit;
Proc iml;
use P; read all into P;
use Sigma1; read all into Sigma1;
P1=P*Sigma1*t(P);
create P1 from P1; append from P1; quit;
```

```
Data Sigma_2;
set s1;
if i=2;
keep var;
run;
Proc iml;
use Sigma_2; read all into Sigma_2;
Sigma2=diag(Sigma_2);
create Sigma2 from Sigma2; append from Sigma2;
quit;
Proc iml;
use P; read all into P;
use Sigma2; read all into Sigma2;
P2=P*Sigma2*t(P);
create P2 from P2; append from P2; quit;

Data Sigma_3;
set s1;
if i=3;
keep var;
run;
Proc iml;
use Sigma_3; read all into Sigma_3;
Sigma3=diag(Sigma_3);
create Sigma3 from Sigma3; append from Sigma3;
quit;
Proc iml;
use P; read all into P;
use Sigma3; read all into Sigma3;
P3=P*Sigma3*t(P);
create P3 from P3; append from P3; quit;

Data Sigma_4;
set s1;
if i=4;
keep var;
run;
Proc iml;
use Sigma_4; read all into Sigma_4;
Sigma4=diag(Sigma_4);
create Sigma4 from Sigma4; append from Sigma4;
quit;
Proc iml;
use P; read all into P;
use Sigma4; read all into Sigma4;
P4=P*Sigma4*t(P);
create P4 from P4; append from P4; quit;

Data Sigma_5;
set s1;
if i=5;
keep var;
run;
Proc iml;
use Sigma_5; read all into Sigma_5;
Sigma5=diag(Sigma_5);
create Sigma5 from Sigma5; append from Sigma5;
quit;
Proc iml;
use P; read all into P;
use Sigma5; read all into Sigma5;
P5=P*Sigma5*t(P);
create P5 from P5; append from P5; quit;

Data Sigma_6;
set s1;
if i=6;
keep var;
run;
Proc iml;
use Sigma_6; read all into Sigma_6;
Sigma6=diag(Sigma_6);
create Sigma6 from Sigma6; append from Sigma6;
quit;
Proc iml;
use P; read all into P;
use Sigma6; read all into Sigma6;
P6=P*Sigma6*t(P);
create P6 from P6; append from P6; quit;

Data Sigma_7;
set s1;
if i=7;
keep var;
run;
Proc iml;
use Sigma_7; read all into Sigma_7;
Sigma7=diag(Sigma_7);
create Sigma7 from Sigma7; append from Sigma7;
quit;
Proc iml;
use P; read all into P;
use Sigma7; read all into Sigma7;
P7=P*Sigma7*t(P);
create P7 from P7; append from P7; quit;

Data Sigma_8;
set s1;
if i=8;
keep var;
run;
Proc iml;
use Sigma_8; read all into Sigma_8;
Sigma8=diag(Sigma_8);
create Sigma8 from Sigma8; append from Sigma8;
quit;
Proc iml;
use P; read all into P;
use Sigma8; read all into Sigma8;
P8=P*Sigma8*t(P);
create P8 from P8; append from P8; quit;

Data Sigma_9;
set s1;
if i=9;
keep var;
run;
Proc iml;
use Sigma_9; read all into Sigma_9;
Sigma9=diag(Sigma_9);
create Sigma9 from Sigma9; append from Sigma9;
quit;
Proc iml;
use P; read all into P;
use Sigma9; read all into Sigma9;
P9=P*Sigma9*t(P);
create P9 from P9; append from P9; quit;

Data Sigma_10;
set s1;
if i=10;
keep var;
run;
Proc iml;
use Sigma_10; read all into Sigma_10;
Sigma10=diag(Sigma_10);
create Sigma10 from Sigma10; append from
Sigma10; quit;
Proc iml;
use P; read all into P;
use Sigma10; read all into Sigma10;
P10=P*Sigma10*t(P);
create P10 from P10; append from P10; quit;

Data Sigma_11;
set s1;
if i=11;
keep var;
run;
Proc iml;
use Sigma_11; read all into Sigma_11;
Sigma11=diag(Sigma_11);
create Sigma11 from Sigma11; append from
Sigma11; quit;
Proc iml;
use P; read all into P;
use Sigma11; read all into Sigma11;
P11=P*Sigma11*t(P);
create P11 from P11; append from P11; quit;

Data Sigma_12;
set s1;
if i=12;
keep var;
run;
Proc iml;
use Sigma_12; read all into Sigma_12;
Sigma12=diag(Sigma_12);
create Sigma12 from Sigma12; append from
Sigma12; quit;
Proc iml;
use P; read all into P;
use Sigma12; read all into Sigma12;
P12=P*Sigma12*t(P);
create P12 from P12; append from P12; quit;
```

```
Data Sigma_13;
set s1;
if i=13;
keep var;
run;
Proc iml;
use Sigma_13; read all into Sigma_13;
Sigma13=diag(Sigma_13);
create Sigma13 from Sigma13; append from
Sigma13; quit;
Proc iml;
use P; read all into P;
use Sigma13; read all into Sigma13;
P13=P*Sigma13*t(P);
create P13 from P13; append from P13; quit;

Data Sigma_14;
set s1;
if i=14;
keep var;
run;
Proc iml;
use Sigma_14; read all into Sigma_14;
Sigma14=diag(Sigma_14);
create Sigma14 from Sigma14; append from
Sigma14; quit;
Proc iml;
use P; read all into P;
use Sigma14; read all into Sigma14;
P14=P*Sigma14*t(P);
create P14 from P14; append from P14; quit;

Data Sigma_15;
set s1;
if i=15;
keep var;
run;
Proc iml;
use Sigma_15; read all into Sigma_15;
Sigma15=diag(Sigma_15);
create Sigma15 from Sigma15; append from
Sigma15; quit;
Proc iml;
use P; read all into P;
use Sigma15; read all into Sigma15;
P15=P*Sigma15*t(P);
create P15 from P15; append from P15; quit;

Data Sigma_16;
set s1;
if i=16;
keep var;
run;
Proc iml;
use Sigma_16; read all into Sigma_16;
Sigma16=diag(Sigma_16);
create Sigma16 from Sigma16; append from
Sigma16; quit;
Proc iml;
use P; read all into P;
use Sigma16; read all into Sigma16;
P16=P*Sigma16*t(P);
create P16 from P16; append from P16; quit;

Data Sigma_17;
set s1;
if i=17;
keep var;
run;
Proc iml;
use Sigma_17; read all into Sigma_17;
Sigma17=diag(Sigma_17);
create Sigma17 from Sigma17; append from
Sigma17; quit;
Proc iml;
use P; read all into P;
use Sigma17; read all into Sigma17;
P17=P*Sigma17*t(P);
create P17 from P17; append from P17; quit;

Data Sigma_18;
set s1;
if i=18;
keep var;
run;
Proc iml;
use Sigma_18; read all into Sigma_18;
Sigma18=diag(Sigma_18);
create Sigma18 from Sigma18; append from
Sigma18; quit;
Proc iml;
use P; read all into P;
use Sigma18; read all into Sigma18;
P18=P*Sigma18*t(P);
create P18 from P18; append from P18; quit;

Data Sigma_19;
set s1;
if i=19;
keep var;
run;
Proc iml;
use Sigma_19; read all into Sigma_19;
Sigma19=diag(Sigma_19);
create Sigma19 from Sigma19; append from
Sigma19; quit;
Proc iml;
use P; read all into P;
use Sigma19; read all into Sigma19;
P19=P*Sigma19*t(P);
create P19 from P19; append from P19; quit;

Data Sigma_20;
set s1;
if i=20;
keep var;
run;
Proc iml;
use Sigma_20; read all into Sigma_20;
Sigma20=diag(Sigma_20);
create Sigma20 from Sigma20; append from
Sigma20; quit;
Proc iml;
use P; read all into P;
use Sigma20; read all into Sigma20;
P20=P*Sigma20*t(P);
create P20 from P20; append from P20; quit;

Data Sigma_21;
set s1;
if i=21;
keep var;
run;
Proc iml;
use Sigma_21; read all into Sigma_21;
Sigma21=diag(Sigma_21);
create Sigma21 from Sigma21; append from
Sigma21; quit;
Proc iml;
use P; read all into P;
use Sigma21; read all into Sigma21;
P21=P*Sigma21*t(P);
create P21 from P21; append from P21; quit;

Data Sigma_22;
set s1;
if i=22;
keep var;
run;
Proc iml;
use Sigma_22; read all into Sigma_22;
Sigma22=diag(Sigma_22);
create Sigma22 from Sigma22; append from
Sigma22; quit;
Proc iml;
use P; read all into P;
use Sigma22; read all into Sigma22;
P22=P*Sigma22*t(P);
create P22 from P22; append from P22; quit;

Data Sigma_23;
set s1;
if i=23;
keep var;
run;
Proc iml;
use Sigma_23; read all into Sigma_23;
Sigma23=diag(Sigma_23);
create Sigma23 from Sigma23; append from
Sigma23; quit;
Proc iml;
use P; read all into P;
use Sigma23; read all into Sigma23;
P23=P*Sigma23*t(P);
create P23 from P23; append from P23; quit;
```

```
Data Sigma_24;
set s1;
if i=24;
keep var;
run;
Proc iml;
use Sigma_24; read all into Sigma_24;
Sigma24=diag(Sigma_24);
create Sigma24 from Sigma24; append from
Sigma24; quit;
Proc iml;
use P; read all into P;
use Sigma24; read all into Sigma24;
P24=P*Sigma24*t(P);
create P24 from P24; append from P24; quit;

Data Sigma_25;
set s1;
if i=25;
keep var;
run;
Proc iml;
use Sigma_25; read all into Sigma_25;
Sigma25=diag(Sigma_25);
create Sigma25 from Sigma25; append from
Sigma25; quit;
Proc iml;
use P; read all into P;
use Sigma25; read all into Sigma25;
P25=P*Sigma25*t(P);
create P25 from P25; append from P25; quit;

Data Sigma_26;
set s1;
if i=26;
keep var;
run;
Proc iml;
use Sigma_26; read all into Sigma_26;
Sigma26=diag(Sigma_26);
create Sigma26 from Sigma26; append from
Sigma26; quit;
Proc iml;
use P; read all into P;
use Sigma26; read all into Sigma26;
P26=P*Sigma26*t(P);
create P26 from P26; append from P26; quit;

Data Sigma_27;
set s1;
if i=27;
keep var;
run;
Proc iml;
use Sigma_27; read all into Sigma_27;
Sigma27=diag(Sigma_27);
create Sigma27 from Sigma27; append from
Sigma27; quit;
Proc iml;
use P; read all into P;
use Sigma27; read all into Sigma27;
P27=P*Sigma27*t(P);
create P27 from P27; append from P27; quit;

Data Sigma_28;
set s1;
if i=28;
keep var;
run;
Proc iml;
use Sigma_28; read all into Sigma_28;
Sigma28=diag(Sigma_28);
create Sigma28 from Sigma28; append from
Sigma28; quit;
Proc iml;
use P; read all into P;
use Sigma28; read all into Sigma28;
P28=P*Sigma28*t(P);
create P28 from P28; append from P28; quit;

Proc iml;
use P1; read all into P1;
use P2; read all into P2;
use P3; read all into P3;
use P4; read all into P4;
use P5; read all into P5;
use P6; read all into P6;
use P7; read all into P7;
use P8; read all into P8;
use P9; read all into P9;
use P10; read all into P10;
use P11; read all into P11;
use P12; read all into P12;
use P13; read all into P13;
use P14; read all into P14;
blockPP1=BLOCK(P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,
P11,P12,P13,P14);
create blockPP1 from blockPP1; append from
blockPP1; quit;

Proc iml;
use P15; read all into P15;
use P16; read all into P16;
use P17; read all into P17;
use P18; read all into P18;
use P19; read all into P19;
use P20; read all into P20;
use P21; read all into P21;
use P22; read all into P22;
use P23; read all into P23;
use P24; read all into P24;
use P25; read all into P25;
use P26; read all into P26;
use P27; read all into P27;
use P28; read all into P28;
blockPP2=BLOCK(P15,P16,P17,P18,P19,P20,P21,P22
,P23,P24,P25,P26,P27,P28);
create blockPP2 from blockPP2; append from
blockPP2; quit;

Proc iml;
use blockPP1; read all into blockPP1;
use blockPP2; read all into blockPP2;
blockPP=BLOCK(blockPP1,blockPP2);
create blockPP from blockPP; append from
blockPP; quit;

/* *** Fim *** */

Proc iml;
use H1; read all into H1;
HH=H1*t(H1);
eval=eigval(HH);
create eval from eval; append from eval; quit;

Data id;
do id=1 to 196;
output; end;
run;

Data eval;
merge id eval;
run;

Data new;
set eval;
by id;
retain count 0;
if first.id then count=0;
if col1 > 0.000001 then count = count + 1;
run;

Proc means data=new sum noprint;
var count;
output out=caracH1 sum=caracH1;
run;

Data caracH1;
set caracH1;
keep caracH1;
run;

Proc iml;
use blockPP; read all into blockPP;
use blockID; read all into blockID;
use Hs1; read all into Hs1;
use rg1_; read all into rg1_;
use caracH1; read all into caracH1;
varu= (rg1_ -  trace((blockID -
Hs1)*blockPP))*(1/(28*6 - caracH1));
create varu from varu; append from varu; quit;

/* Termina aqui o cálculo da variância
sigma2_u */
```

```
/** Cálculo da matriz: z2= c^(-0.5)*f'*P*Teta
- var. dependente do modelo transformado **/

Proc iml;
use tetad1; read all into tetad1;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y1=(1/sqrt(c))*t(f)*P*tetad1;
create y1 from y1; append from y1; quit;
Proc iml;
use tetad2; read all into tetad2;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y2=(1/sqrt(c))*t(f)*P*tetad2;
create y2 from y2; append from y2; quit;
Proc iml;
use tetad3; read all into tetad3;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y3=(1/sqrt(c))*t(f)*P*tetad3;
create y3 from y3; append from y3; quit;
Proc iml;
use tetad4; read all into tetad4;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y4=(1/sqrt(c))*t(f)*P*tetad4;
create y4 from y4; append from y4; quit;
Proc iml;
use tetad5; read all into tetad5;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y5=(1/sqrt(c))*t(f)*P*tetad5;
create y5 from y5; append from y5; quit;
Proc iml;
use tetad6; read all into tetad6;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y6=(1/sqrt(c))*t(f)*P*tetad6;
create y6 from y6; append from y6; quit;
Proc iml;
use tetad7; read all into tetad7;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y7=(1/sqrt(c))*t(f)*P*tetad7;
create y7 from y7; append from y7; quit;
Proc iml;
use tetad8; read all into tetad8;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y8=(1/sqrt(c))*t(f)*P*tetad8;
create y8 from y8; append from y8; quit;
Proc iml;
use tetad9; read all into tetad9;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y9=(1/sqrt(c))*t(f)*P*tetad9;
create y9 from y9; append from y9; quit;
Proc iml;
use tetad10; read all into tetad10;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y10=(1/sqrt(c))*t(f)*P*tetad10;
create y10 from y10; append from y10; quit;
Proc iml;
use tetad11; read all into tetad11;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y11=(1/sqrt(c))*t(f)*P*tetad11;
create y11 from y11; append from y11; quit;
Proc iml;
use tetad12; read all into tetad12;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y12=(1/sqrt(c))*t(f)*P*tetad12;
create y12 from y12; append from y12; quit;
Proc iml;
use tetad13; read all into tetad13;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y13=(1/sqrt(c))*t(f)*P*tetad13;
create y13 from y13; append from y13; quit;
Proc iml;
use tetad14; read all into tetad14;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y14=(1/sqrt(c))*t(f)*P*tetad14;
create y14 from y14; append from y14; quit;
Proc iml;
use tetad15; read all into tetad15;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y15=(1/sqrt(c))*t(f)*P*tetad15;
create y15 from y15; append from y15; quit;
Proc iml;
use tetad16; read all into tetad16;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y16=(1/sqrt(c))*t(f)*P*tetad16;
create y16 from y16; append from y16; quit;
Proc iml;
use tetad17; read all into tetad17;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y17=(1/sqrt(c))*t(f)*P*tetad17;
create y17 from y17; append from y17; quit;
Proc iml;
use tetad18; read all into tetad18;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y18=(1/sqrt(c))*t(f)*P*tetad18;
create y18 from y18; append from y18; quit;
Proc iml;
use tetad19; read all into tetad19;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y19=(1/sqrt(c))*t(f)*P*tetad19;
create y19 from y19; append from y19; quit;
Proc iml;
use tetad20; read all into tetad20;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y20=(1/sqrt(c))*t(f)*P*tetad20;
create y20 from y20; append from y20; quit;
Proc iml;
use tetad21; read all into tetad21;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y21=(1/sqrt(c))*t(f)*P*tetad21;
create y21 from y21; append from y21; quit;
Proc iml;
use tetad22; read all into tetad22;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y22=(1/sqrt(c))*t(f)*P*tetad22;
create y22 from y22; append from y22; quit;
Proc iml;
use tetad23; read all into tetad23;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y23=(1/sqrt(c))*t(f)*P*tetad23;
create y23 from y23; append from y23; quit;
Proc iml;
use tetad24; read all into tetad24;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y24=(1/sqrt(c))*t(f)*P*tetad24;
create y24 from y24; append from y24; quit;
Proc iml;
use tetad25; read all into tetad25;
use P; read all into P;
use f; read all into f;
use c; read all into c;
```

```
y25=(1/sqrt(c))*t(f)*P*tetad25;
create y25 from y25; append from y25; quit;
Proc iml;
use tetad26; read all into tetad26;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y26=(1/sqrt(c))*t(f)*P*tetad26;
create y26 from y26; append from y26; quit;
Proc iml;
use tetad27; read all into tetad27;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y27=(1/sqrt(c))*t(f)*P*tetad27;
create y27 from y27; append from y27; quit;
Proc iml;
use tetad28; read all into tetad28;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y28=(1/sqrt(c))*t(f)*P*tetad28;
create y28 from y28; append from y28; quit;
run;

Data z2;
set y1 y2 y3 y4 y5 y6 y7 y8 y9 y10 y11 y12 y13
y14 y15 y16 y17 y18 y19 y20 y21 y22 y23 y24
y25 y26 y27 y28;
run;

/** Cálculo da matriz: H2= c^(-0.5)*f'*P*X -
var. independentes do modelo transformado **/

Proc iml;
use xd1; read all into xd1;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x1=(1/sqrt(c))*t(f)*P*xd1;
create x1 from x1; append from x1; quit;
Proc iml;
use xd2; read all into xd2;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x2=(1/sqrt(c))*t(f)*P*xd2;
create x2 from x2; append from x2; quit;
Proc iml;
use xd3; read all into xd3;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x3=(1/sqrt(c))*t(f)*P*xd3;
create x3 from x3; append from x3; quit;
Proc iml;
use xd4; read all into xd4;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x4=(1/sqrt(c))*t(f)*P*xd4;
create x4 from x4; append from x4; quit;
Proc iml;
use xd5; read all into xd5;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x5=(1/sqrt(c))*t(f)*P*xd5;
create x5 from x5; append from x5; quit;
Proc iml;
use xd6; read all into xd6;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x6=(1/sqrt(c))*t(f)*P*xd6;
create x6 from x6; append from x6; quit;
Proc iml;
use xd7; read all into xd7;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x7=(1/sqrt(c))*t(f)*P*xd7;
create x7 from x7; append from x7; quit;
Proc iml;
use xd8; read all into xd8;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x8=(1/sqrt(c))*t(f)*P*xd8;
create x8 from x8; append from x8; quit;
Proc iml;
use xd9; read all into xd9;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x9=(1/sqrt(c))*t(f)*P*xd9;
create x9 from x9; append from x9; quit;
Proc iml;
use xd10; read all into xd10;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x10=(1/sqrt(c))*t(f)*P*xd10;
create x10 from x10; append from x10; quit;
Proc iml;
use xd11; read all into xd11;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x11=(1/sqrt(c))*t(f)*P*xd11;
create x11 from x11; append from x11; quit;
Proc iml;
use xd12; read all into xd12;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x12=(1/sqrt(c))*t(f)*P*xd12;
create x12 from x12; append from x12; quit;
Proc iml;
use xd13; read all into xd13;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x13=(1/sqrt(c))*t(f)*P*xd13;
create x13 from x13; append from x13; quit;
Proc iml;
use xd14; read all into xd14;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x14=(1/sqrt(c))*t(f)*P*xd14;
create x14 from x14; append from x14; quit;
Proc iml;
use xd15; read all into xd15;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x15=(1/sqrt(c))*t(f)*P*xd15;
create x15 from x15; append from x15; quit;
Proc iml;
use xd16; read all into xd16;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x16=(1/sqrt(c))*t(f)*P*xd16;
create x16 from x16; append from x16; quit;
Proc iml;
use xd17; read all into xd17;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x17=(1/sqrt(c))*t(f)*P*xd17;
create x17 from x17; append from x17; quit;
Proc iml;
use xd18; read all into xd18;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x18=(1/sqrt(c))*t(f)*P*xd18;
create x18 from x18; append from x18; quit;
Proc iml;
use xd19; read all into xd19;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x19=(1/sqrt(c))*t(f)*P*xd19;
create x19 from x19; append from x19; quit;
Proc iml;
use xd20; read all into xd20;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x20=(1/sqrt(c))*t(f)*P*xd20;
create x20 from x20; append from x20; quit;
Proc iml;
use xd21; read all into xd21;
```

```
use P; read all into P;
use f; read all into f;
use c; read all into c;
x21=(1/sqrt(c))*t(f)*P*xd21;
create x21 from x21; append from x21; quit;
Proc iml;
use xd22; read all into xd22;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x22=(1/sqrt(c))*t(f)*P*xd22;
create x22 from x22; append from x22; quit;
Proc iml;
use xd23; read all into xd23;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x23=(1/sqrt(c))*t(f)*P*xd23;
create x23 from x23; append from x23; quit;
Proc iml;
use xd24; read all into xd24;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x24=(1/sqrt(c))*t(f)*P*xd24;
create x24 from x24; append from x24; quit;
Proc iml;
use xd25; read all into xd25;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x25=(1/sqrt(c))*t(f)*P*xd25;
create x25 from x25; append from x25; quit;
Proc iml;
use xd26; read all into xd26;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x26=(1/sqrt(c))*t(f)*P*xd26;
create x26 from x26; append from x26; quit;
Proc iml;
use xd27; read all into xd27;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x27=(1/sqrt(c))*t(f)*P*xd27;
create x27 from x27; append from x27; quit;
Proc iml;
use xd28; read all into xd28;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x28=(1/sqrt(c))*t(f)*P*xd28;
create x28 from x28; append from x28; quit;
run;

Data H2;
set x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13
x14 x15
x16 x17 x18 x19 x20 x21 x22 x23 x24 x25 x26
x27 x28;
run;

/** Regressão de z2(Y) sobre H2(X) **/

Data z2;
set z2;
rename col1=col0;
run;

Data reg2;
merge z2 H2;
run;

proc reg data=reg2 noprint;
        model col0=col1 col2 / noint;
        output out=rg2  residual=r2;
run;
quit;

Data rg2;
set rg2;
qr2=r2*r2;
Keep qr2;
run;

Proc means data=rg2 sum noprint;
var qr2;
output out=rg2_ sum=sqr2;
run;

Data rg2_;
set rg2_;
keep sqr2;
run;

/** Cálculo da variância sigma2_v **/

Proc iml;
use H2; read all into H2;
Ident=I(28);
PH2=Ident - H2*ginv(t(H2)*H2)*t(H2);
create PH2 from PH2; append from PH2; quit;

Proc iml;
use f; read all into f;
use P1; read all into P1;
use P2; read all into P2;
use P3; read all into P3;
use P4; read all into P4;
use P5; read all into P5;
use P6; read all into P6;
use P7; read all into P7;
use P8; read all into P8;
use P9; read all into P9;
use P10; read all into P10;
use P11; read all into P11;
use P12; read all into P12;
use P13; read all into P13;
use P14; read all into P14;
use P15; read all into P15;
use P16; read all into P16;
use P17; read all into P17;
use P18; read all into P18;
use P19; read all into P19;
use P20; read all into P20;
use P21; read all into P21;
use P22; read all into P22;
use P23; read all into P23;
use P24; read all into P24;
use P25; read all into P25;
use P26; read all into P26;
use P27; read all into P27;
use P28; read all into P28;
use c; read all into c;
cfp1=t(f)*P1*f*(1 / c);
cfp2=t(f)*P2*f*(1 / c);
cfp3=t(f)*P3*f*(1 / c);
cfp4=t(f)*P4*f*(1 / c);
cfp5=t(f)*P5*f*(1 / c);
cfp6=t(f)*P6*f*(1 / c);
cfp7=t(f)*P7*f*(1 / c);
cfp8=t(f)*P8*f*(1 / c);
cfp9=t(f)*P9*f*(1 / c);
cfp10=t(f)*P10*f*(1 / c);
cfp11=t(f)*P11*f*(1 / c);
cfp12=t(f)*P12*f*(1 / c);
cfp13=t(f)*P13*f*(1 / c);
cfp14=t(f)*P14*f*(1 / c);
cfp15=t(f)*P15*f*(1 / c);
cfp16=t(f)*P16*f*(1 / c);
cfp17=t(f)*P17*f*(1 / c);
cfp18=t(f)*P18*f*(1 / c);
cfp19=t(f)*P19*f*(1 / c);
cfp20=t(f)*P20*f*(1 / c);
cfp21=t(f)*P21*f*(1 / c);
cfp22=t(f)*P22*f*(1 / c);
cfp23=t(f)*P23*f*(1 / c);
cfp24=t(f)*P24*f*(1 / c);
cfp25=t(f)*P25*f*(1 / c);
cfp26=t(f)*P26*f*(1 / c);
cfp27=t(f)*P27*f*(1 / c);
cfp28=t(f)*P28*f*(1 / c);
cfp_=cfp1//cfp2//cfp3//cfp4//cfp5//cfp6//cfp7/
/cfp8//cfp9//cfp10//cfp11//cfp12//cfp13//cfp14
//cfp15//cfp16//cfp17//cfp18//cfp19//cfp20//cf
p21//cfp22//cfp23//cfp24//cfp25//cfp26//cfp27/
/cfp28;
cfp=diag(cfp_);
create cfp from cfp; append from cfp; quit;

/* *** Fim *** */

Proc iml;
use H2; read all into H2;
FF=H2*t(H2);
```

```
eval1=eigval(FF);
create eval1 from eval1; append from eval1;
quit;

Data id1;
do id=1 to 28;
output; end;
run;

Data eval1;
merge id1 eval1;
run;

Data new1;
set eval1;
by id;
retain count 0;
if first.id then count=0;
if col1 > 0.000001 then count = count + 1;
run;

Proc means data=new1 sum noprint;
var count;
output out=caracH2 sum=caracH2;
run;

Data caracH2;
set caracH2;
keep caracH2;  /* Caracteristica de H2 */
run;

Proc iml;
use PH2; read all into PH2;
use cfp; read all into cfp;
use c; read all into c;
use rg2_; read all into rg2_;
use caracH2; read all into caracH2;
use varu; read all into varu;
use Binv; read all into Binv;
varv= (rg2_ -  trace(PH2*cfp) - varu*(28 -
caracH2))/(c*trace(PH2*Binv));
create varv from varv; append from varv; quit;

/* Termina aqui o cálculo da variância
sigma2_v */
/* Truncagem a zero das variâncias sigma2_u e
sigma2_v */

Data varu;
set varu;
sigma2u= max( col1 , 0);
keep sigma2u;
run;
Data varv;
set varv;
sigma2v= max( col1 , 0);
keep sigma2v;
run;

/* Construção da matriz G, sendo para tal
necessária a matriz Gama */

/*** Matriz GAMA  ***/

Data ro_1;
set dataro;
ro_1= 1-ro*ro;
keep ro_1;
run;
Data ro_2;
set dataro;
ro_2= ro*ro;
keep ro_2;
run;
Data ro_3;
set dataro;
ro_3= ro*ro*ro;
keep ro_3;
run;
Data ro_4;
set dataro;
ro_4= ro*ro*ro*ro;
keep ro_4;
run;
Data ro_5;
set dataro;
ro_5= ro*ro*ro*ro*ro;
keep ro_5;
run;
Data ro_6;
set dataro;
ro_6= ro*ro*ro*ro*ro*ro;
keep ro_6;
run;

Proc iml;
use ro_1; read all into ro_1;
use dataro; read all into dataro;
use ro_2; read all into ro_2;
use ro_3; read all into ro_3;
use ro_4; read all into ro_4;
use ro_5; read all into ro_5;
use ro_6; read all into ro_6;
I=I(7);
I1={0 1 0 0 0 0 0, 0 0 1 0 0 0 0, 0 0 0 1 0 0
0, 0 0 0 0 1 0 0, 0 0 0 0 0 1 0, 0 0 0 0 0 0
1, 0 0 0 0 0 0 0};
I2={0 0 1 0 0 0 0, 0 0 0 1 0 0 0, 0 0 0 0 1 0
0, 0 0 0 0 0 1 0, 0 0 0 0 0 0 1, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I3={0 0 0 1 0 0 0, 0 0 0 0 1 0 0, 0 0 0 0 0 1
0, 0 0 0 0 0 0 1, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I4={0 0 0 0 1 0 0, 0 0 0 0 0 1 0, 0 0 0 0 0 0
1, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I5={0 0 0 0 0 1 0, 0 0 0 0 0 0 1, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I6={0 0 0 0 0 0 1, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
Gama = I/ro_1 + dataro*I1/ro_1 +
t(dataro*I1/ro_1) + ro_2*I2/ro_1 +
t(ro_2*I2/ro_1) + ro_3*I3/ro_1 +
t(ro_3*I3/ro_1) + ro_4*I4/ro_1 +
t(ro_4*I4/ro_1) + ro_5*I5/ro_1 +
t(ro_5*I5/ro_1) + ro_6*I6/ro_1 +
t(ro_6*I6/ro_1);
create Gama from Gama;
append from Gama;
quit;

/*** Matriz G1 ***/

Proc iml;
use varv; read all into varv;
use Binv; read all into Binv;
G1=varv*Binv;
create G1 from G1; append from G1; quit;

/*** Matriz G2 ***/

Proc iml;
use varu; read all into varu;
use Gama; read all into Gama;
Ident=I(28);
G2=varu*(Ident@Gama);
create G2 from G2; append from G2; quit;

/*** Matriz G ***/

Proc iml;
use G1; read all into G1;
use G2; read all into G2;
Mat_GG=block (G1 , G2);
create Mat_GG from Mat_GG; append from Mat_GG;
quit;

Proc iml;
row=t(do(1,224,1));
create row from row; append from row; quit;

Data row;
set row;
rename col1=row;
run;

Data Mat_GG;
merge row Mat_GG;
run;

Proc sort data=s1;
by i t;
run;
```

```
Data s2;
set s1;
keep var;
run;

Data par;
set s2;
rename var=est;
run;

/*********************************************
*********************************************/
/** ESTIMAÇÃO DO MODELO DE LUIS PEREIRA *****/
/*********************************************
*********************************************/

Proc sort data=iabhipth1 out= cron;
by i t; run;

/* O Proc mixed só permite incluir uma matriz
G, independentemente no número de RANDOM
definidos. A matriz G definida tem que ser a
matriz completa para todo o modelo */

Proc mixed data=cron MMEqSol noprofile
noclprint;
        class i t;
        model y=x / solution;
        random i;
random i*t/type=un Gdata=Mat_GG G V S;
        repeated /group=i*t;
        parms /parmsdata=par noiter;
make 'solutionf' out=beta;
make 'solutionr' out=u;
make 'Fitstatistics' out=aic;
make 'G' out=Mat_Gaux;
make 'V' out=Mat_V;
run; quit;

/* Verificação se a matriz V produzida pelo
Proc Mixed está correcta, ou seja, se está de
acordo com as matrizes G1, G2 e R definidas no
proc mixed */
/*
Proc iml;
use G1; read all into G1;
use G2; read all into G2;
I=I(28);
um=J(7,1,1);
GG_lixo=(I@um)*G1*t(I@um) + G2;
create GG_lixo from GG_lixo; append from
GG_lixo; quit;

Proc iml;
use s2; read all into s2;
RR_lixo=diag(s2);
create RR_lixo from RR_lixo; append from
RR_lixo; quit;

Proc iml;
use GG_lixo; read all into GG_lixo;
use RR_lixo; read all into RR_lixo;
VV_lixo=GG_lixo + RR_lixo;
create VV_lixo from VV_lixo; append from
VV_lixo; quit;

Data Mat_Vlixo;
set Mat_V;
drop index row;
run;

Proc iml;
use VV_lixo; read all into VV_lixo;
use Mat_Vlixo; read all into Mat_Vlixo;
Compara=VV_lixo - Mat_Vlixo;
create Compara from Compara; append from
Compara; quit; */

/* Fim da verificação */

/*********************************************/
/* CÁLCULO DAS ESTIMATIVAS DO PREÇO MÉDIO DE
TRANSACÇÃO - NUT III */
/*********************************************/

Data id;
do id=1 to 2;
output; end;
run;

Data beta;
merge id beta;
run;

Data beta0;
set beta;
if id=1;
rename estimate=b0;
keep estimate;
run;

Data beta1;
set beta;
if id=2;
rename estimate=b1;
keep estimate;
run;

Data betan;
merge beta0 beta1;
_type_=0;
run;

/****** Efeitos Aleatórios: v ******/

Data u;
set u;
if estimate='.' then estimate=0;
run;

Data v;
set u;
if Effect='i';
keep i estimate;
run;

Proc sort data=v;
by i;
run;

Proc iml;
use v; read all into v;
one=J(7,1,1);
v1=v@one;
create v1 from v1; append from v1; quit;

Data v1;
set v1;
_type_=0;
rename col2=v; run;

/****** Efeitos Aleatórios: u ******/

Data u1;
set u;
if Effect='i*t';
_type_=0;
keep i t estimate _type_;
rename estimate=u;
run;

Proc sort data=u1;
by i t;
run;

Data beta_vu;
merge betan v1 u1;
by _type_;
run;

/*****/

Data Ef_Aleat_V;
set v;
drop i;
run;

Data Ef_Aleat_u;
set u1;
drop i t _type_;
run;

Proc iml;
use beta0; read all into beta0;
use beta1; read all into beta1;
use Ef_Aleat_V; read all into Ef_Aleat_V;
```

```
use Ef_Aleat_u; read all into Ef_Aleat_u;
csi=beta0//beta1//Ef_Aleat_V//Ef_Aleat_u;
create csi from csi; append from csi; quit;


/** Estimação do preço médio de transacção em
cada NUTIII **/

Proc sort data=cron;
by i t;
run;

Data cron;
merge cron beta_vu;
by i t;
run;

Data crono_resf;
set cron;
e = b0 + x*b1 + v + u;
keep i y t _type_ x b0 b1 u v e;
run;

/*******************************************/
/* Cálculo das estimativas do EQM do EBLUP **/
/*******************************************/

/* O PROC MIXED calcula a matriz de
variâncias-covariâncias, Mat_V=Cov(Y), a
matriz Gu=Cov(u) e a matriz Gv=Cov(Zv). Estas
duas matrizes podem ser obtidas a partir
de uma matriz denominada Mat_Gaux. Depois
calcula-se G=Gu+Gv. Neste caso a matriz Z
do modelo linear geral misto teórico é a
identidade, Z=I, porque se admitiu que no
modelo de Rao-Yu, u*=Zv+u */

/*** MATRIZ X ***/

Proc sort data=cron out=cronologico;
by i t; run;

Data XX1;
set cronologico;
keep x; run;

Proc iml;
J=J(196,1,1);
create J from J;
append from J;
quit;

Proc iml;
use J;
read all into J;
use XX1;
read all into XX1;
X=insert(J,XX1,0,2);
create X from X;
append from X;
quit;

/*** Colunas da Matriz GAMA ***/
/* Nota: a matriz Gama já foi construída acima
porque foi necessária para constrir G */

Proc iml;
use Gama; read all into Gama;
Gama1=Gama[,1];
create Gama1 from Gama1;
append from Gama1;
quit;
Proc iml;
use Gama; read all into Gama;
Gama2=Gama[,2];
create Gama2 from Gama2;
append from Gama2;
quit;
Proc iml;
use Gama; read all into Gama;
Gama3=Gama[,3];
create Gama3 from Gama3;
append from Gama3;
quit;
Proc iml;
use Gama; read all into Gama;
Gama4=Gama[,4];
create Gama4 from Gama4;
append from Gama4;
quit;
Proc iml;
use Gama; read all into Gama;
Gama5=Gama[,5];
create Gama5 from Gama5;
append from Gama5;
quit;
Proc iml;
use Gama; read all into Gama;
Gama6=Gama[,6];
create Gama6 from Gama6;
append from Gama6;
quit;
Proc iml;
use Gama; read all into Gama;
Gama7=Gama[,7];
create Gama7 from Gama7;
append from Gama7;
quit;

/*** MATRIZ V ***/
/* Nota: No modelo espaciotemporal a matriz V
não é diagonal por blocos */

Data MatV;
set Mat_v;
drop Index row;
run;

Proc iml;
use MatV; read all into MatV;
MatVinv=inv(MatV);
create MatVinv from MatVinv;
append from MatVinv; quit;

/***************************/
/****** Cálculo do g1 ******/
/***************************/

Data Mat_gg;
set Mat_gg;
drop row;
run;

Proc iml;
use MatVinv; read all into MatVinv;
use Mat_gg; read all into Mat_gg;
use Hb.Z; read all into Z;
G1_aux=Z*(Mat_gg -
Mat_gg*t(Z)*MatVinv*Z*Mat_gg)*t(Z);
create G1_aux from G1_aux;
append from G1_aux;
quit;

Proc iml;
use G1_aux;
read all into G1_aux;
G1=vecdiag(G1_aux);
create G1 from G1;
append from G1;
quit;

Data g1;
set g1;
rename col1=g1; run;

/***************************/
/****** Cálculo do g2 ******/
/***************************/

Proc iml;
use X; read all into X;
use MatVinv; read all into MatVinv;
use Mat_gg; read all into Mat_gg;
use Hb.Z; read all into Z;
A=I(196);
G2_aux=t(t(X) -
t(X)*MatVinv*Z*Mat_gg*t(Z))*inv(t(X)*MatVinv*X
)*(t(X) - t(X)*MatVinv*Z*Mat_gg*t(Z));
create G2_aux from G2_aux;
append from G2_aux;
quit;

Proc iml;
use G2_aux;
read all into G2_aux;
G2=vecdiag(G2_aux);
```

```
create G2 from G2;
append from G2;
quit;

Data g2;
set g2;
rename col1=g2;
run;

/***************************/
/****** Cálculo do g3 ******/
/***************************/

/* O Cálculo do g3 tem que ser feito
individualmente para cada trimestre EM CADA
DOMÍNIO. Note-se que o vector coluna "variante
de sigma_i" varia em função do DOMINIO e que o
vector "zeta_it" varia em função do DOMINIO e
do TEMPO. Basta calcular a matriz A para cada
trimestre de um determinado domínio, e depois
replicar para os restantes domínios, que é
sempre igual */

/* É necessário definir as 28 colunas da
matriz inv(B) */

Proc iml;
use Binv; read all into Binv;
Binv1=Binv[,1];
create Binv1 from Binv1;
append from Binv1; quit;
Proc iml;
use Binv; read all into Binv;
Binv2=Binv[,2];
create Binv2 from Binv2;
append from Binv2; quit;
Proc iml;
use Binv; read all into Binv;
Binv3=Binv[,3];
create Binv3 from Binv3;
append from Binv3; quit;
Proc iml;
use Binv; read all into Binv;
Binv4=Binv[,4];
create Binv4 from Binv4;
append from Binv4; quit;
Proc iml;
use Binv; read all into Binv;
Binv5=Binv[,5];
create Binv5 from Binv5;
append from Binv5; quit;
Proc iml;
use Binv; read all into Binv;
Binv6=Binv[,6];
create Binv6 from Binv6;
append from Binv6; quit;
Proc iml;
use Binv; read all into Binv;
Binv7=Binv[,7];
create Binv7 from Binv7;
append from Binv7; quit;
Proc iml;
use Binv; read all into Binv;
Binv8=Binv[,8];
create Binv8 from Binv8;
append from Binv8; quit;
Proc iml;
use Binv; read all into Binv;
Binv9=Binv[,9];
create Binv9 from Binv9;
append from Binv9; quit;
Proc iml;
use Binv; read all into Binv;
Binv10=Binv[,10];
create Binv10 from Binv10;
append from Binv10; quit;
Proc iml;
use Binv; read all into Binv;
Binv11=Binv[,11];
create Binv11 from Binv11;
append from Binv11; quit;
Proc iml;
use Binv; read all into Binv;
Binv12=Binv[,12];
create Binv12 from Binv12;
append from Binv12; quit;
Proc iml;
use Binv; read all into Binv;
Binv13=Binv[,13];
create Binv13 from Binv13;
append from Binv13; quit;
Proc iml;
use Binv; read all into Binv;
Binv14=Binv[,14];
create Binv14 from Binv14;
append from Binv14; quit;
Proc iml;
use Binv; read all into Binv;
Binv15=Binv[,15];
create Binv15 from Binv15;
append from Binv15; quit;
Proc iml;
use Binv; read all into Binv;
Binv16=Binv[,16];
create Binv16 from Binv16;
append from Binv16; quit;
Proc iml;
use Binv; read all into Binv;
Binv17=Binv[,17];
create Binv17 from Binv17;
append from Binv17; quit;
Proc iml;
use Binv; read all into Binv;
Binv18=Binv[,18];
create Binv18 from Binv18;
append from Binv18; quit;
Proc iml;
use Binv; read all into Binv;
Binv19=Binv[,19];
create Binv19 from Binv19;
append from Binv19; quit;
Proc iml;
use Binv; read all into Binv;
Binv20=Binv[,20];
create Binv20 from Binv20;
append from Binv20; quit;
Proc iml;
use Binv; read all into Binv;
Binv21=Binv[,21];
create Binv21 from Binv21;
append from Binv21; quit;
Proc iml;
use Binv; read all into Binv;
Binv22=Binv[,22];
create Binv22 from Binv22;
append from Binv22; quit;
Proc iml;
use Binv; read all into Binv;
Binv23=Binv[,23];
create Binv23 from Binv23;
append from Binv23; quit;
Proc iml;
use Binv; read all into Binv;
Binv24=Binv[,24];
create Binv24 from Binv24;
append from Binv24; quit;
Proc iml;
use Binv; read all into Binv;
Binv25=Binv[,25];
create Binv25 from Binv25;
append from Binv25; quit;
Proc iml;
use Binv; read all into Binv;
Binv26=Binv[,26];
create Binv26 from Binv26;
append from Binv26; quit;
Proc iml;
use Binv; read all into Binv;
Binv27=Binv[,27];
create Binv27 from Binv27;
append from Binv27; quit;
Proc iml;
use Binv; read all into Binv;
Binv28=Binv[,28];
create Binv28 from Binv28;
append from Binv28; quit;

/* Trabalho sobre a matriz A - cálculo de 7
matrizes: A1, A2, A3, A4, A5, A6 e A7 */

/* DOMÍNIO 1 */

Proc iml;
I1={1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
create I1 from I1; append from I1; quit;
```

```
/* 1º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama1; read all into Gama1;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama1) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama1) ) )
*MatVinv* ( (I1@Gama1) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama1)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama1; read all into Gama1;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama1) ) )
*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama1) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama1; read all into Gama1;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama1) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama1) ) )*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama1) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A11_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A11_ from A11_; append from A11_; quit;

/* 2º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama2; read all into Gama2;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama2) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama2) ) )
*MatVinv* ( (I1@Gama2) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama2)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama2; read all into Gama2;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama2) ) )
*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama2) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama2; read all into Gama2;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama2) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama2) ) )*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama2) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A12_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A12_ from A12_; append from A12_; quit;

/* 3º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama3; read all into Gama3;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama3) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama3) ) )
*MatVinv* ( (I1@Gama3) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama3)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama3; read all into Gama3;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama3) ) )
*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama3) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama3; read all into Gama3;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama3) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama3) ) )*MatVinv*
```

```
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama3) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A13_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A13_ from A13_; append from A13_; quit;

/* 4º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama4; read all into Gama4;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama4) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama4) ) )
*MatVinv* ( (I1@Gama4) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama4)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama4; read all into Gama4;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama4) ) )
*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama4) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama4; read all into Gama4;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama4) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama4) ) )*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama4) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A14_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A14_ from A14_; append from A14_; quit;

/* 5º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama5; read all into Gama5;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama5) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama5) ) )
*MatVinv* ( (I1@Gama5) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama5)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama5; read all into Gama5;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama5) ) )
*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama5) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama5; read all into Gama5;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama5) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama5) ) )*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama5) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A15_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A15_ from A15_; append from A15_; quit;

/* 6º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama6; read all into Gama6;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama6) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama6) ) )
*MatVinv* ( (I1@Gama6) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama6)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama6; read all into Gama6;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama6) ) )
*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama6) ) );
create a22 from a22; append from a22; quit;
```

```
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama6; read all into Gama6;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama6) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama6) ) )*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama6) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A16_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A16_ from A16_; append from A16_; quit;

/* 7º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama7; read all into Gama7;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama7) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama7) ) )
*MatVinv* ( (I1@Gama7) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama7)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama7; read all into Gama7;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama7) ) )
*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama7) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama7; read all into Gama7;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama7) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama7) ) )*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama7) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A17_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A17_ from A17_; append from A17_; quit;


/* DOMÍNIO 2 */ (…)
/* DOMÍNIO 3 */ (…)
/* DOMÍNIO 4 */ (…)
/* DOMÍNIO 5 */ (…)
(…)
/* DOMÍNIO 25 */ (…)
/* DOMÍNIO 26 */ (…)
/* DOMÍNIO 27 */ (…)
/* DOMÍNIO 28 */ (…)

/* *** Fim *** */

/**********************************/
/* Cálculo da matriz SIGMA* (2*2) */
/**********************************/

/* OBS.: Antes de calcular as covariâncias da
matriz SIGMA*, é necessário calcular as
matrizes C, C*, C1 e C2. Note-se que a matriz
V (que depois funciona como a matriz OMEGA já
está calculada, pois é a matriz Cov(Y) e note-
se também que não é necessário calcular a */

/* MatC */

Proc iml;
use D; read all into D;
use P; read all into P;
Ident=I(7);
IDP=(Ident - D)*P; /* Matriz (I - D)*P */
create IDP from IDP; append from IDP; quit;

Proc iml;
use IDP; read all into IDP;
blockIDP1=BLOCK(IDP,IDP,IDP,IDP,IDP,IDP,IDP,ID
P,IDP,IDP,IDP,IDP,IDP,IDP);
create blockIDP1 from blockIDP1; append from
blockIDP1; quit;
Proc iml;
use blockIDP1; read all into blockIDP1;
MatC=BLOCK(blockIDP1,blockIDP1);
create MatC from MatC; append from MatC; quit;

/* MatC* */

Proc iml;
use f; read all into f;
use P; read all into P;
use c; read all into c;
cfP1=(1/sqrt(c))*t(f)*P; /* Matriz c(-
1/2)*fT*P */
create cfP1 from cfP1; append from cfP1; quit;

Proc iml;
use cfP1; read all into cfP1;
blockcfP1=BLOCK(cfP1,cfP1,cfP1,cfP1,cfP1,cfP1,
cfP1,cfP1,cfP1,cfP1,cfP1,cfP1,cfP1,cfP1);
create blockcfP1 from blockcfP1; append from
blockcfP1; quit;
Proc iml;
use blockcfP1; read all into blockcfP1;
MatC_ast=BLOCK(blockcfP1,blockcfP1);
create MatC_ast from MatC_ast; append from
MatC_ast; quit;

/* MatC1 */

Proc iml;
use MatC; read all into MatC;
use X; read all into X;
Ident=I(196);
MatC1=t(MatC)* (Ident - MatC*X*
ginv(t(X)*t(MatC)*MatC*X) *t(X)*t(MatC))
*MatC;
create MatC1 from MatC1; append from MatC1;
quit;

/* MatC2 */

Proc iml;
use MatC_ast; read all into MatC_ast;
use X; read all into X;
Ident=I(28);
```

```
MatC2=t(MatC_ast)* (Ident - MatC_ast*X*
ginv(t(X)*t(MatC_ast)*MatC_ast*X)
*t(X)*t(MatC_ast)) *MatC_ast;
create MatC2 from MatC2; append from MatC2;
quit;

/* varu = k1* a'C1a + k2 */

Proc iml;
use caracH1; read all into caracH1;
k1= 1/(28*6 - caracH1);
create k1 from k1; append from k1; quit;

Proc iml;
use blockPP; read all into blockPP;
use blockID; read all into blockID;
use Hs1; read all into Hs1;
use k1; read all into k1;
k2= -trace((blockID - Hs1)*blockPP)*k1;
create k2 from k2; append from k2; quit;

/* varv = k3* a'C2a + k4* a'C1a + k5 */

Proc iml;
use PH2; read all into PH2;
use Binv; read all into Binv;
use c; read all into c;
k3= 1/(c*trace(PH2*Binv));
create k3 from k3; append from k3; quit;

Proc iml;
use k1; read all into k1;
use k3; read all into k3;
use c; read all into c;
use caracH2; read all into caracH2;
k4= -k1*(28-caracH2)*k3;
create k4 from k4; append from k4; quit;

Proc iml;
use PH2; read all into PH2;
use cfp; read all into cfp;
use c; read all into c;
use k2; read all into k2;
use k3; read all into k3;
use caracH2; read all into caracH2;
k5= (-k2*(28-caracH2) - trace(PH2*cfp))*k3;
create k5 from k5; append from k5; quit;

/* SIGMA11 = Var(varu) = 2* k1^2 * tr(C1 * V *
C1 * V) */
/* SIGMA11 */

Proc iml;
use MatC1; read all into MatC1;
use MatV; read all into MatV;
use k1; read all into k1;
Sigma11_=2*k1*k1* trace(MatC1 * MatV * MatC1 *
MatV);
create Sigma11_ from Sigma11_; append from
Sigma11_; quit;

/* SIGMA22 = Var(varv)= 2*k3^2 * tr(C2*V*C2*V)
+ 4* k3*k4* tr(C1*V*C2*V) + 2* k4^2 *
tr(C1*V*C1*V)*/
/* SIGMA22 */

Proc iml;
use MatC1; read all into MatC1;
use MatC2; read all into MatC2;
use MatV; read all into MatV;
use k3; read all into k3;
use k4; read all into k4;
Sigma22_=2*k3*k3*trace(MatC2 * MatV * MatC2 *
MatV) + 4*k3*k4*trace(MatC1 * MatV * MatC2 *
MatV) +
2*k4*k4*trace(MatC1 * MatV * MatC1 * MatV);
create Sigma22_ from Sigma22_; append from
Sigma22_; quit;

/* SIGMA21 = Cov(varu,varv)= 2*k1*k3 *
tr(C1*V*C2*V) + 2* k1*k4* tr(C1*V*C1*V) */
/* SIGMA21 */

Proc iml;
use MatC1; read all into MatC1;
use MatC2; read all into MatC2;
use MatV; read all into MatV;
use k1; read all into k1;
use k3; read all into k3;
use k4; read all into k4;
Sigma21_=2*k1*k3* trace(MatC1 * MatV * MatC2 *
MatV) + 2*k1*k4* trace(MatC1 * MatV * MatC1 *
MatV);
create Sigma21_ from Sigma21_; append from
Sigma21_; quit;

Proc iml;
use Sigma11_; read all into Sigma11_;
use Sigma22_; read all into Sigma22_;
use Sigma21_; read all into Sigma21_;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
SIGMA=Sigma11_*I1 + Sigma21_*I2 + Sigma21_*I3
+ Sigma22_*I4;
create SIGMA from SIGMA;
append from SIGMA; quit;

/* *** */

/* Cálculo do trace(Ait * SIGMA) */

Proc iml;
use SIGMA; read all into SIGMA;
use A11_; read all into A11_;
use A12_; read all into A12_;
use A13_; read all into A13_;
use A14_; read all into A14_;
use A15_; read all into A15_;
use A16_; read all into A16_;
use A17_; read all into A17_;
use A21_; read all into A21_;
use A22_; read all into A22_;
use A23_; read all into A23_;
use A24_; read all into A24_;
use A25_; read all into A25_;
use A26_; read all into A26_;
use A27_; read all into A27_;
use A31_; read all into A31_;
use A32_; read all into A32_;
use A33_; read all into A33_;
use A34_; read all into A34_;
use A35_; read all into A35_;
use A36_; read all into A36_;
use A37_; read all into A37_;
use A41_; read all into A41_;
use A42_; read all into A42_;
use A43_; read all into A43_;
use A44_; read all into A44_;
use A45_; read all into A45_;
use A46_; read all into A46_;
use A47_; read all into A47_;
use A51_; read all into A51_;
use A52_; read all into A52_;
use A53_; read all into A53_;
use A54_; read all into A54_;
use A55_; read all into A55_;
use A56_; read all into A56_;
use A57_; read all into A57_;
use A61_; read all into A61_;
use A62_; read all into A62_;
use A63_; read all into A63_;
use A64_; read all into A64_;
use A65_; read all into A65_;
use A66_; read all into A66_;
use A67_; read all into A67_;
use A71_; read all into A71_;
use A72_; read all into A72_;
use A73_; read all into A73_;
use A74_; read all into A74_;
use A75_; read all into A75_;
use A76_; read all into A76_;
use A77_; read all into A77_;
use A81_; read all into A81_;
use A82_; read all into A82_;
use A83_; read all into A83_;
use A84_; read all into A84_;
use A85_; read all into A85_;
use A86_; read all into A86_;
use A87_; read all into A87_;
use A91_; read all into A91_;
use A92_; read all into A92_;
use A93_; read all into A93_;
use A94_; read all into A94_;
use A95_; read all into A95_;
```

```
use A96_; read all into A96_;
use A97_; read all into A97_;
use A101_; read all into A101_;
use A102_; read all into A102_;
use A103_; read all into A103_;
use A104_; read all into A104_;
use A105_; read all into A105_;
use A106_; read all into A106_;
use A107_; read all into A107_;
G3_11=trace(A11_ * SIGMA);
G3_12=trace(A12_ * SIGMA);
G3_13=trace(A13_ * SIGMA);
G3_14=trace(A14_ * SIGMA);
G3_15=trace(A15_ * SIGMA);
G3_16=trace(A16_ * SIGMA);
G3_17=trace(A17_ * SIGMA);
G3_21=trace(A21_ * SIGMA);
G3_22=trace(A22_ * SIGMA);
G3_23=trace(A23_ * SIGMA);
G3_24=trace(A24_ * SIGMA);
G3_25=trace(A25_ * SIGMA);
G3_26=trace(A26_ * SIGMA);
G3_27=trace(A27_ * SIGMA);
G3_31=trace(A31_ * SIGMA);
G3_32=trace(A32_ * SIGMA);
G3_33=trace(A33_ * SIGMA);
G3_34=trace(A34_ * SIGMA);
G3_35=trace(A35_ * SIGMA);
G3_36=trace(A36_ * SIGMA);
G3_37=trace(A37_ * SIGMA);
G3_41=trace(A41_ * SIGMA);
G3_42=trace(A42_ * SIGMA);
G3_43=trace(A43_ * SIGMA);
G3_44=trace(A44_ * SIGMA);
G3_45=trace(A45_ * SIGMA);
G3_46=trace(A46_ * SIGMA);
G3_47=trace(A47_ * SIGMA);
G3_51=trace(A51_ * SIGMA);
G3_52=trace(A52_ * SIGMA);
G3_53=trace(A53_ * SIGMA);
G3_54=trace(A54_ * SIGMA);
G3_55=trace(A55_ * SIGMA);
G3_56=trace(A56_ * SIGMA);
G3_57=trace(A57_ * SIGMA);
G3_61=trace(A61_ * SIGMA);
G3_62=trace(A62_ * SIGMA);
G3_63=trace(A63_ * SIGMA);
G3_64=trace(A64_ * SIGMA);
G3_65=trace(A65_ * SIGMA);
G3_66=trace(A66_ * SIGMA);
G3_67=trace(A67_ * SIGMA);
G3_71=trace(A71_ * SIGMA);
G3_72=trace(A72_ * SIGMA);
G3_73=trace(A73_ * SIGMA);
G3_74=trace(A74_ * SIGMA);
G3_75=trace(A75_ * SIGMA);
G3_76=trace(A76_ * SIGMA);
G3_77=trace(A77_ * SIGMA);
G3_81=trace(A81_ * SIGMA);
G3_82=trace(A82_ * SIGMA);
G3_83=trace(A83_ * SIGMA);
G3_84=trace(A84_ * SIGMA);
G3_85=trace(A85_ * SIGMA);
G3_86=trace(A86_ * SIGMA);
G3_87=trace(A87_ * SIGMA);
G3_91=trace(A91_ * SIGMA);
G3_92=trace(A92_ * SIGMA);
G3_93=trace(A93_ * SIGMA);
G3_94=trace(A94_ * SIGMA);
G3_95=trace(A95_ * SIGMA);
G3_96=trace(A96_ * SIGMA);
G3_97=trace(A97_ * SIGMA);
G3_101=trace(A101_ * SIGMA);
G3_102=trace(A102_ * SIGMA);
G3_103=trace(A103_ * SIGMA);
G3_104=trace(A104_ * SIGMA);
G3_105=trace(A105_ * SIGMA);
G3_106=trace(A106_ * SIGMA);
G3_107=trace(A107_ * SIGMA);
G3_aux1=G3_11//G3_12//G3_13//G3_14//G3_15//G3_
16//G3_17//G3_21//G3_22//G3_23//G3_24//G3_25//
G3_26//G3_27//G3_31//G3_32//G3_33//G3_34//G3_3
5//G3_36//G3_37//G3_41//G3_42//G3_43//G3_44//G
3_45//G3_46//G3_47//G3_51//G3_52//G3_53//G3_54
//G3_55//G3_56//G3_57//G3_61//G3_62//G3_63//G3
_64//G3_65//G3_66//G3_67//G3_71//G3_72//G3_73/
/G3_74//G3_75//G3_76//G3_77//G3_81//G3_82//G3_
83//G3_84//G3_85//G3_86//G3_87//G3_91//G3_92//
G3_93//G3_94//G3_95//G3_96//G3_97//G3_101//G3_
102//G3_103//G3_104//G3_105//G3_106//G3_107;
create G3_aux1 from G3_aux1; append from
G3_aux1; quit;

Proc iml;
use SIGMA; read all into SIGMA;
use A111_; read all into A111_;
use A112_; read all into A112_;
use A113_; read all into A113_;
use A114_; read all into A114_;
use A115_; read all into A115_;
use A116_; read all into A116_;
use A117_; read all into A117_;
use A121_; read all into A121_;
use A122_; read all into A122_;
use A123_; read all into A123_;
use A124_; read all into A124_;
use A125_; read all into A125_;
use A126_; read all into A126_;
use A127_; read all into A127_;
use A131_; read all into A131_;
use A132_; read all into A132_;
use A133_; read all into A133_;
use A134_; read all into A134_;
use A135_; read all into A135_;
use A136_; read all into A136_;
use A137_; read all into A137_;
use A141_; read all into A141_;
use A142_; read all into A142_;
use A143_; read all into A143_;
use A144_; read all into A144_;
use A145_; read all into A145_;
use A146_; read all into A146_;
use A147_; read all into A147_;
use A151_; read all into A151_;
use A152_; read all into A152_;
use A153_; read all into A153_;
use A154_; read all into A154_;
use A155_; read all into A155_;
use A156_; read all into A156_;
use A157_; read all into A157_;
use A161_; read all into A161_;
use A162_; read all into A162_;
use A163_; read all into A163_;
use A164_; read all into A164_;
use A165_; read all into A165_;
use A166_; read all into A166_;
use A167_; read all into A167_;
use A171_; read all into A171_;
use A172_; read all into A172_;
use A173_; read all into A173_;
use A174_; read all into A174_;
use A175_; read all into A175_;
use A176_; read all into A176_;
use A177_; read all into A177_;
use A181_; read all into A181_;
use A182_; read all into A182_;
use A183_; read all into A183_;
use A184_; read all into A184_;
use A185_; read all into A185_;
use A186_; read all into A186_;
use A187_; read all into A187_;
use A191_; read all into A191_;
use A192_; read all into A192_;
use A193_; read all into A193_;
use A194_; read all into A194_;
use A195_; read all into A195_;
use A196_; read all into A196_;
use A197_; read all into A197_;
use A201_; read all into A201_;
use A202_; read all into A202_;
use A203_; read all into A203_;
use A204_; read all into A204_;
use A205_; read all into A205_;
use A206_; read all into A206_;
use A207_; read all into A207_;
G3_111=trace(A111_ * SIGMA);
G3_112=trace(A112_ * SIGMA);
G3_113=trace(A113_ * SIGMA);
G3_114=trace(A114_ * SIGMA);
G3_115=trace(A115_ * SIGMA);
G3_116=trace(A116_ * SIGMA);
G3_117=trace(A117_ * SIGMA);
G3_121=trace(A121_ * SIGMA);
G3_122=trace(A122_ * SIGMA);
G3_123=trace(A123_ * SIGMA);
G3_124=trace(A124_ * SIGMA);
```

```
G3_125=trace(A125_ * SIGMA);
G3_126=trace(A126_ * SIGMA);
G3_127=trace(A127_ * SIGMA);
G3_131=trace(A131_ * SIGMA);
G3_132=trace(A132_ * SIGMA);
G3_133=trace(A133_ * SIGMA);
G3_134=trace(A134_ * SIGMA);
G3_135=trace(A135_ * SIGMA);
G3_136=trace(A136_ * SIGMA);
G3_137=trace(A137_ * SIGMA);
G3_141=trace(A141_ * SIGMA);
G3_142=trace(A142_ * SIGMA);
G3_143=trace(A143_ * SIGMA);
G3_144=trace(A144_ * SIGMA);
G3_145=trace(A145_ * SIGMA);
G3_146=trace(A146_ * SIGMA);
G3_147=trace(A147_ * SIGMA);
G3_151=trace(A151_ * SIGMA);
G3_152=trace(A152_ * SIGMA);
G3_153=trace(A153_ * SIGMA);
G3_154=trace(A154_ * SIGMA);
G3_155=trace(A155_ * SIGMA);
G3_156=trace(A156_ * SIGMA);
G3_157=trace(A157_ * SIGMA);
G3_161=trace(A161_ * SIGMA);
G3_162=trace(A162_ * SIGMA);
G3_163=trace(A163_ * SIGMA);
G3_164=trace(A164_ * SIGMA);
G3_165=trace(A165_ * SIGMA);
G3_166=trace(A166_ * SIGMA);
G3_167=trace(A167_ * SIGMA);
G3_171=trace(A171_ * SIGMA);
G3_172=trace(A172_ * SIGMA);
G3_173=trace(A173_ * SIGMA);
G3_174=trace(A174_ * SIGMA);
G3_175=trace(A175_ * SIGMA);
G3_176=trace(A176_ * SIGMA);
G3_177=trace(A177_ * SIGMA);
G3_181=trace(A181_ * SIGMA);
G3_182=trace(A182_ * SIGMA);
G3_183=trace(A183_ * SIGMA);
G3_184=trace(A184_ * SIGMA);
G3_185=trace(A185_ * SIGMA);
G3_186=trace(A186_ * SIGMA);
G3_187=trace(A187_ * SIGMA);
G3_191=trace(A191_ * SIGMA);
G3_192=trace(A192_ * SIGMA);
G3_193=trace(A193_ * SIGMA);
G3_194=trace(A194_ * SIGMA);
G3_195=trace(A195_ * SIGMA);
G3_196=trace(A196_ * SIGMA);
G3_197=trace(A197_ * SIGMA);
G3_201=trace(A201_ * SIGMA);
G3_202=trace(A202_ * SIGMA);
G3_203=trace(A203_ * SIGMA);
G3_204=trace(A204_ * SIGMA);
G3_205=trace(A205_ * SIGMA);
G3_206=trace(A206_ * SIGMA);
G3_207=trace(A207_ * SIGMA);
G3_aux2=G3_111//G3_112//G3_113//G3_114//G3_115
//G3_116//G3_117//G3_121//G3_122//G3_123//G3_1
24//G3_125//G3_126//G3_127//G3_131//G3_132//G3
_133//G3_134//G3_135//G3_136//G3_137//G3_141//
G3_142//G3_143//G3_144//G3_145//G3_146//G3_147
//G3_151//G3_152//G3_153//G3_154//G3_155//G3_1
56//G3_157//G3_161//G3_162//G3_163//G3_164//G3
_165//G3_166//G3_167//G3_171//G3_172//G3_173//
G3_174//G3_175//G3_176//G3_177//G3_181//G3_182
//G3_183//G3_184//G3_185//G3_186//G3_187//G3_1
91//G3_192//G3_193//G3_194//G3_195//G3_196//G3
_197//G3_201//G3_202//G3_203//G3_204//G3_205//
G3_206//G3_207;
create G3_aux2 from G3_aux2; append from
G3_aux2; quit;

Proc iml;
use SIGMA; read all into SIGMA;
use A211_; read all into A211_;
use A212_; read all into A212_;
use A213_; read all into A213_;
use A214_; read all into A214_;
use A215_; read all into A215_;
use A216_; read all into A216_;
use A217_; read all into A217_;
use A221_; read all into A221_;
use A222_; read all into A222_;
use A223_; read all into A223_;
use A224_; read all into A224_;
use A225_; read all into A225_;
use A226_; read all into A226_;
use A227_; read all into A227_;
use A231_; read all into A231_;
use A232_; read all into A232_;
use A233_; read all into A233_;
use A234_; read all into A234_;
use A235_; read all into A235_;
use A236_; read all into A236_;
use A237_; read all into A237_;
use A241_; read all into A241_;
use A242_; read all into A242_;
use A243_; read all into A243_;
use A244_; read all into A244_;
use A245_; read all into A245_;
use A246_; read all into A246_;
use A247_; read all into A247_;
use A251_; read all into A251_;
use A252_; read all into A252_;
use A253_; read all into A253_;
use A254_; read all into A254_;
use A255_; read all into A255_;
use A256_; read all into A256_;
use A257_; read all into A257_;
use A261_; read all into A261_;
use A262_; read all into A262_;
use A263_; read all into A263_;
use A264_; read all into A264_;
use A265_; read all into A265_;
use A266_; read all into A266_;
use A267_; read all into A267_;
use A271_; read all into A271_;
use A272_; read all into A272_;
use A273_; read all into A273_;
use A274_; read all into A274_;
use A275_; read all into A275_;
use A276_; read all into A276_;
use A277_; read all into A277_;
use A281_; read all into A281_;
use A282_; read all into A282_;
use A283_; read all into A283_;
use A284_; read all into A284_;
use A285_; read all into A285_;
use A286_; read all into A286_;
use A287_; read all into A287_;
G3_211=trace(A211_ * SIGMA);
G3_212=trace(A212_ * SIGMA);
G3_213=trace(A213_ * SIGMA);
G3_214=trace(A214_ * SIGMA);
G3_215=trace(A215_ * SIGMA);
G3_216=trace(A216_ * SIGMA);
G3_217=trace(A217_ * SIGMA);
G3_221=trace(A221_ * SIGMA);
G3_222=trace(A222_ * SIGMA);
G3_223=trace(A223_ * SIGMA);
G3_224=trace(A224_ * SIGMA);
G3_225=trace(A225_ * SIGMA);
G3_226=trace(A226_ * SIGMA);
G3_227=trace(A227_ * SIGMA);
G3_231=trace(A231_ * SIGMA);
G3_232=trace(A232_ * SIGMA);
G3_233=trace(A233_ * SIGMA);
G3_234=trace(A234_ * SIGMA);
G3_235=trace(A235_ * SIGMA);
G3_236=trace(A236_ * SIGMA);
G3_237=trace(A237_ * SIGMA);
G3_241=trace(A241_ * SIGMA);
G3_242=trace(A242_ * SIGMA);
G3_243=trace(A243_ * SIGMA);
G3_244=trace(A244_ * SIGMA);
G3_245=trace(A245_ * SIGMA);
G3_246=trace(A246_ * SIGMA);
G3_247=trace(A247_ * SIGMA);
G3_251=trace(A251_ * SIGMA);
G3_252=trace(A252_ * SIGMA);
G3_253=trace(A253_ * SIGMA);
G3_254=trace(A254_ * SIGMA);
G3_255=trace(A255_ * SIGMA);
G3_256=trace(A256_ * SIGMA);
G3_257=trace(A257_ * SIGMA);
G3_261=trace(A261_ * SIGMA);
G3_262=trace(A262_ * SIGMA);
G3_263=trace(A263_ * SIGMA);
G3_264=trace(A264_ * SIGMA);
G3_265=trace(A265_ * SIGMA);
G3_266=trace(A266_ * SIGMA);
G3_267=trace(A267_ * SIGMA);
G3_271=trace(A271_ * SIGMA);
```

```
G3_272=trace(A272_ * SIGMA);
G3_273=trace(A273_ * SIGMA);
G3_274=trace(A274_ * SIGMA);
G3_275=trace(A275_ * SIGMA);
G3_276=trace(A276_ * SIGMA);
G3_277=trace(A277_ * SIGMA);
G3_281=trace(A281_ * SIGMA);
G3_282=trace(A282_ * SIGMA);
G3_283=trace(A283_ * SIGMA);
G3_284=trace(A284_ * SIGMA);
G3_285=trace(A285_ * SIGMA);
G3_286=trace(A286_ * SIGMA);
G3_287=trace(A287_ * SIGMA);
G3_aux3=G3_211//G3_212//G3_213//G3_214//G3_215
//G3_216//G3_217//G3_221//G3_222//G3_223//G3_2
24//G3_225//G3_226//G3_227//G3_231//G3_232//G3
_233//G3_234//G3_235//G3_236//G3_237//G3_241//
G3_242//G3_243//G3_244//G3_245//G3_246//G3_247
//G3_251//G3_252//G3_253//G3_254//G3_255//G3_2
56//G3_257//G3_261//G3_262//G3_263//G3_264//G3
_265//G3_266//G3_267//G3_271//G3_272//G3_273//
G3_274//G3_275//G3_276//G3_277//G3_281//G3_282
//G3_283//G3_284//G3_285//G3_286//G3_287;
create G3_aux3 from G3_aux3; append from
G3_aux3; quit;

Data G3_aux;
set G3_aux1 G3_aux2 G3_aux3;
run;

Data g3;
set g3_aux;
rename col1=g3;
run;

/* *** Fim *** */

/* **** Estimativa do EQMP do EBLUP *****/

Data Estimat_PC;
merge crono_resf g1 g2 g3;
EQM_EBLUP= G1 + G2 + 2*G3;
CV=sqrt(EQM_EBLUP)/e;
simul=&i;
run;

Proc sort data=Estimat_PC out=Estimat_PC;
by t i;
run;

Proc append base=HB.Est_LP_simulacao
data=Estimat_PC;
run;

/*********************************************
*********************************************/
/******* Estimação com restrições -
benchmarking ao nível de NUTS II ************/
/*********************************************
*********************************************/

Data q_;
set Estimat_directasnutsii_pseudo;
if simul=&i;
keep mean;
run;

Proc iml;
use rest.X0; read all into X0;
use Matv; read all into Matv;
C= ginv( t(X0) * inv(Matv) * X0 );
create C from C; append from C; quit;

/***/

Proc iml;
use C; read all into C;
use rest.Q; read all into Q;
A=C* t(Q) * inv( Q * C * t(Q) );
create A from A; append from A; quit;

/* Estimativas com restrições`*/

Proc iml;
use rest.X0; read all into X0;
use csi; read all into csi;
use A; read all into A;
use q_; read all into q_;
use rest.Q; read all into Q;
EstRest=X0 * CSI + X0 * A *(q_ - Q*CSI );
create EstRest from EstRest; append from
EstRest; quit;

Data EstRest;
set EstRest;
rename col1=e_restII;
run;

Data Estimat_PC_rest_nutII;
merge crono_resf EstRest;
simul=&i;
keep t i simul e_restII;
run;

Proc append base=HB.Est_LP_simulacao_restII
data=Estimat_PC_rest_nutII;
run;

/******* Estimação com restrições -
benchmarking ao nível de NUTS I ********/

Data q_;
set Estimat_directasnutsi_pseudo;
if simul=&i;
keep mean;
run;

Proc iml;
use rest1.X0; read all into X0;
use Matv; read all into Matv;
C= ginv( t(X0) * inv(Matv) * X0 );
create C from C; append from C; quit;

/***/

Proc iml;
use C; read all into C;
use rest1.Q; read all into Q;
A=C* t(Q) * inv( Q * C * t(Q) );
create A from A; append from A; quit;

/* Estimativas com restrições`*/

Proc iml;
use rest1.X0; read all into X0;
use csi; read all into csi;
use A; read all into A;
use q_; read all into q_;
use rest1.Q; read all into Q;
EstRest=X0 * CSI + X0 * A *(q_ - Q*CSI );
create EstRest from EstRest; append from
EstRest; quit;

Data EstRest;
set EstRest;
rename col1=e_restI;
run;

Data Estimat_PC_rest_nutI;
merge crono_resf EstRest;
simul=&i;
keep t i simul e_restI;
run;

Proc append base=HB.Est_LP_simulacao_restI
data=Estimat_PC_rest_nutI;
run;

DM 'CLEAR OUTPUT';
DM 'CLEAR LOG';

%mend;
%inc 'C:\Documents and Settings\Luis Pereira
\My Documents\EstudoEmpirico\chamamacro.txt';

/********************************************/
/*FIM DA SIMULAÇÃO COM ESTIMADOR EBLUP DE LP*/
/********************************************/
```

Apêndice 19 – Cálculo das medidas de avaliação da qualidade dos estimadores no estudo empírico por simulação *design-based*

CÓDIGO SAS

```
Data parametro;
set Hb.verdadeiro_parametro;
keep i t parametro; run;

Proc iml;
use parametro; read all into parametro;
J1000=J(1000,1,1);
J196=J(196,1,1);
v1000=(1:1000);
aux1=J1000@parametro;
aux2=t(v1000)@J196;
vector_parametro=aux2||aux1;
create vector_parametro from vector_parametro;
append from vector_parametro; quit;

Data vector_parametro;
set vector_parametro;
rename col1=simul col2=t col3=i
col4=parametro; run;

Proc sort data=vector_parametro;
by simul t i;
Proc sort data=hb.Estimat_directas_pseudo1
out=Estimat_directas_pseudo1;
by simul t i;
Proc sort data=hb.Est_sintquoc_simulacao
out=Est_sintquoc_simulacao;
by simul t i;
Proc sort data=hb.Est_sintreg_simulacao
out=Est_sintreg_simulacao;
by simul t i;
Proc sort data=hb.Est_comb_simulacao
out=Est_comb_simulacao;
by simul t i;
Proc sort data=hb.Est_fh_simulacao
out=Est_fh_simulacao;
by simul t i;
Proc sort data=hb.Est_ns_simulacao
out=Est_ns_simulacao;
by simul t i;
Proc sort data=hb.Est_ry_simulacao
out=Est_ry_simulacao;
by simul t i;
Proc sort data=hb.Est_pc_simulacao
out=Est_pc_simulacao;
by simul t i;
Proc sort data=hb.Est_pc_simulacao_resti
out=Est_pc_simulacao_resti;
by simul t i;
Proc sort data=hb.Est_pc_simulacao_restii
out=Est_pc_simulacao_restii;
by simul t i; run;

Data Est_fh_simulacao;
set Est_fh_simulacao;
rename e=EstFH EQM_EBLUP=EQMP_EstFH;
Data Est_ns_simulacao;
set Est_ns_simulacao;
rename e=EstNS EQM_EBLUP=EQMP_EstNS;
Data Est_ry_simulacao;
set Est_ry_simulacao;
rename e=EstRY EQM_EBLUP=EQMP_EstRY;
Data Est_pc_simulacao;
set Est_pc_simulacao;
rename e=EstPC EQM_EBLUP=EQMP_EstPC;
Data Est_pc_simulacao_resti;
set Est_pc_simulacao_resti;
rename e_restI=EstPCRi;
Data Est_pc_simulacao_restii;
set Est_pc_simulacao_restii;
rename e_restII=EstPCRii; run;

Data Est_simul_designbased;
merge vector_parametro
Estimat_directas_pseudo1
Est_sintquoc_simulacao Est_sintreg_simulacao
Est_comb_simulacao
Est_fh_simulacao Est_ns_simulacao
Est_ry_simulacao Est_pc_simulacao
Est_pc_simulacao_resti
Est_pc_simulacao_restii;
by simul t i;
keep simul t i parametro y estQ estR estC
estFH estNS estRY estPC estPCRi estPCRii
EQMP_EstFH EQMP_EstNS EQMP_EstRY EQMP_EstPC;
run;

Proc sort data=Estimat_directas_pseudo1;
by t i;
Proc sort data=Est_sintquoc_simulacao;
by t i;
Proc sort data=Est_sintreg_simulacao;
by t i;
Proc sort data=Est_comb_simulacao;
by t i;
Proc sort data=Est_fh_simulacao;
by t i;
Proc sort data=Est_ns_simulacao;
by t i;
Proc sort data=Est_ry_simulacao;
by t i;
Proc sort data=Est_pc_simulacao;
by t i;
Proc sort data=Est_pc_simulacao_resti;
by t i;
Proc sort data=Est_pc_simulacao_restii;
by t i; run;

proc means data=Estimat_directas_pseudo1;
by t i;
var y;
output out=Estm_directas_pseudo1 mean=ym; /*
Valor médio das estimativas directas */
proc means data=Est_sintquoc_simulacao;
by t i;
var estQ;
output out=Estm_sintquoc_simulacao mean=estQm;
proc means data=Est_sintreg_simulacao;
by t i;
var estR;
output out=Estm_sintreg_simulacao mean=estRm;
proc means data=Est_comb_simulacao;
by t i;
var estC;
output out=Estm_comb_simulacao mean=estCm;
proc means data=Est_fh_simulacao;
by t i;
var estFH;
output out=Estm_fh_simulacao mean=estFHm;
proc means data=Est_ns_simulacao;
by t i;
var estNS;
output out=Estm_ns_simulacao mean=estNSm;
proc means data=Est_ry_simulacao;
by t i;
var estRY;
output out=Estm_ry_simulacao mean=estRYm;
proc means data=Est_pc_simulacao;
by t i;
var estPC;
output out=Estm_pc_simulacao mean=estPCm;
proc means data=Est_pc_simulacao_resti;
by t i;
var EstPCRi;
output out=Estm_pc_simulacao_resti
mean=EstPCRim;
proc means data=Est_pc_simulacao_restii;
by t i;
var EstPCRii;
output out=Estm_pc_simulacao_restii
mean=EstPCRiim; run;
```

```
Data Estm_directas_pseudo1;
set Estm_directas_pseudo1;
drop _type_ _freq_;
Data Estm_sintquoc_simulacao;
set Estm_sintquoc_simulacao;
drop _type_ _freq_;
Data Estm_sintreg_simulacao;
set Estm_sintreg_simulacao;
drop _type_ _freq_;
Data Estm_comb_simulacao;
set Estm_comb_simulacao;
drop _type_ _freq_;
Data Estm_fh_simulacao;
set Estm_fh_simulacao;
drop _type_ _freq_;
Data Estm_ns_simulacao;
set Estm_ns_simulacao;
drop _type_ _freq_;
Data Estm_ry_simulacao;
set Estm_ry_simulacao;
drop _type_ _freq_;
Data Estm_pc_simulacao;
set Estm_pc_simulacao;
drop _type_ _freq_;
Data Estm_pc_simulacao_resti;
set Estm_pc_simulacao_resti;
drop _type_ _freq_;
Data Estm_pc_simulacao_restii;
set Estm_pc_simulacao_restii;
drop _type_ _freq_; run;

Proc iml;
use Estm_directas_pseudo1; read all into
Estm_directas_pseudo1;
J1000=J(1000,1,1);
J196=J(196,1,1);
v1000=(1:1000);
aux1=J1000@Estm_directas_pseudo1;
aux2=t(v1000)@J196;
Estm_directas=aux2||aux1;
create Estm_directas from Estm_directas;
append from Estm_directas; quit;
Data Estm_directas;
set Estm_directas;
rename col1=simul col2=t col3=i col4=ym;
run;

Proc iml;
use Estm_sintquoc_simulacao; read all into
Estm_sintquoc_simulacao;
J1000=J(1000,1,1);
J196=J(196,1,1);
v1000=(1:1000);
aux1=J1000@Estm_sintquoc_simulacao;
aux2=t(v1000)@J196;
Estm_sintquoc=aux2||aux1;
create Estm_sintquoc from Estm_sintquoc;
append from Estm_sintquoc; quit;
Data Estm_sintquoc;
set Estm_sintquoc;
rename col1=simul col2=t col3=i col4=estQm;
run;

Proc iml;
use Estm_sintreg_simulacao; read all into
Estm_sintreg_simulacao;
J1000=J(1000,1,1);
J196=J(196,1,1);
v1000=(1:1000);
aux1=J1000@Estm_sintreg_simulacao;
aux2=t(v1000)@J196;
Estm_sintreg=aux2||aux1;
create Estm_sintreg from Estm_sintreg; append
from Estm_sintreg; quit;
Data Estm_sintreg;
set Estm_sintreg;
rename col1=simul col2=t col3=i col4=estRm;
run;

Proc iml;
use Estm_comb_simulacao; read all into
Estm_comb_simulacao;
J1000=J(1000,1,1);
J196=J(196,1,1);
v1000=(1:1000);
aux1=J1000@Estm_comb_simulacao;
aux2=t(v1000)@J196;
Estm_comb=aux2||aux1;
create Estm_comb from Estm_comb; append from
Estm_comb; quit;
Data Estm_comb;
set Estm_comb;
rename col1=simul col2=t col3=i col4=estCm;
run;

Proc iml;
use Estm_fh_simulacao; read all into
Estm_fh_simulacao;
J1000=J(1000,1,1);
J196=J(196,1,1);
v1000=(1:1000);
aux1=J1000@Estm_fh_simulacao;
aux2=t(v1000)@J196;
Estm_fh=aux2||aux1;
create Estm_fh from Estm_fh; append from
Estm_fh; quit;
Data Estm_fh;
set Estm_fh;
rename col1=simul col2=t col3=i col4=estFHm;
run;

Proc iml;
use Estm_ns_simulacao; read all into
Estm_ns_simulacao;
J1000=J(1000,1,1);
J196=J(196,1,1);
v1000=(1:1000);
aux1=J1000@Estm_ns_simulacao;
aux2=t(v1000)@J196;
Estm_ns=aux2||aux1;
create Estm_ns from Estm_ns; append from
Estm_ns; quit;
Data Estm_ns;
set Estm_ns;
rename col1=simul col2=t col3=i col4=estNSm;
run;

Proc iml;
use Estm_ry_simulacao; read all into
Estm_ry_simulacao;
J1000=J(1000,1,1);
J196=J(196,1,1);
v1000=(1:1000);
aux1=J1000@Estm_ry_simulacao;
aux2=t(v1000)@J196;
Estm_ry=aux2||aux1;
create Estm_ry from Estm_ry; append from
Estm_ry; quit;
Data Estm_ry;
set Estm_ry;
rename col1=simul col2=t col3=i col4=estRYm;
run;

Proc iml;
use Estm_pc_simulacao; read all into
Estm_pc_simulacao;
J1000=J(1000,1,1);
J196=J(196,1,1);
v1000=(1:1000);
aux1=J1000@Estm_pc_simulacao;
aux2=t(v1000)@J196;
Estm_pc=aux2||aux1;
create Estm_pc from Estm_pc; append from
Estm_pc; quit;
Data Estm_pc;
set Estm_pc;
rename col1=simul col2=t col3=i col4=estPCm;
run;

Proc iml;
use Estm_pc_simulacao_resti; read all into
Estm_pc_simulacao_resti;
J1000=J(1000,1,1);
J196=J(196,1,1);
v1000=(1:1000);
aux1=J1000@Estm_pc_simulacao_resti;
aux2=t(v1000)@J196;
Estm_pc_resti=aux2||aux1;
create Estm_pc_resti from Estm_pc_resti;
append from Estm_pc_resti; quit;
Data Estm_pc_resti;
set Estm_pc_resti;
rename col1=simul col2=t col3=i col4=estPCRIm;
run;

Proc iml;
```

```
use Estm_pc_simulacao_restii; read all into
Estm_pc_simulacao_restii;
J1000=J(1000,1,1);
J196=J(196,1,1);
v1000=(1:1000);
aux1=J1000@Estm_pc_simulacao_restii;
aux2=t(v1000)@J196;
Estm_pc_restii=aux2||aux1;
create Estm_pc_restii from Estm_pc_restii;
append from Estm_pc_restii; quit;
Data Estm_pc_restii;
set Estm_pc_restii;
rename col1=simul col2=t col3=i
col4=estPCRIIm; run;

Data hb.Est_simul1000_designbased;
merge Est_simul_designbased Estm_directas
Estm_sintquoc Estm_sintreg Estm_comb
Estm_fh Estm_ns Estm_ry Estm_pc Estm_pc_resti
Estm_pc_restii;
by simul t i; run;

/*********************************************/
/* FIM DA PREPARAÇÃO DO FICHEIRO COM 1000
RÉPLICAS - AGORA É SÓ FAZER ANÁLISE DE DADOS*/
/*********************************************/

Data auxiliar;
set hb.Est_simul1000_designbased;

/* Enviesamento (B) */
B_y= y - parametro;
B_estQ= estQ - parametro;
B_estR= estR - parametro;
B_estC= estC - parametro;
B_estFH= estFH - parametro;
B_estNS= estNS - parametro;
B_estRY= estRY - parametro;
B_estPC= estPC - parametro;
B_estPCRI= estPCRI - parametro;
B_estPCRII= estPCRII - parametro;

/* Enviesamento relativo (BR) */
BR_y= (y - parametro)/parametro;
BR_estQ= (estQ - parametro)/parametro;
BR_estR= (estR - parametro)/parametro;
BR_estC= (estC - parametro)/parametro;
BR_estFH= (estFH - parametro)/parametro;
BR_estNS= (estNS - parametro)/parametro;
BR_estRY= (estRY - parametro)/parametro;
BR_estPC= (estPC - parametro)/parametro;
BR_estPCRI= (estPCRI - parametro)/parametro;
BR_estPCRII= (estPCRII - parametro)/parametro;

/* Erro absoluto (EA) */
EA_y= abs(y - parametro);
EA_estQ= abs(estQ - parametro);
EA_estR= abs(estR - parametro);
EA_estC= abs(estC - parametro);
EA_estFH= abs(estFH - parametro);
EA_estNS= abs(estNS - parametro);
EA_estRY= abs(estRY - parametro);
EA_estPC= abs(estPC - parametro);
EA_estPCRI= abs(estPCRI - parametro);
EA_estPCRII= abs(estPCRII - parametro);

/* Erro relativo absoluto (ERA) */
ERA_y= abs((y - parametro)/parametro);
ERA_estQ= abs((estQ - parametro)/parametro);
ERA_estR= abs((estR - parametro)/parametro);
ERA_estC= abs((estC - parametro)/parametro);
ERA_estFH= abs((estFH - parametro)/parametro);
ERA_estNS= abs((estNS - parametro)/parametro);
ERA_estRY= abs((estRY - parametro)/parametro);
ERA_estPC= abs((estPC - parametro)/parametro);
ERA_estPCRI= abs((estPCRI -
parametro)/parametro);
ERA_estPCRII= abs((estPCRII -
parametro)/parametro);

/* Erro Quadrático Médio (EQM) */
EQM_y= (y - parametro)*(y - parametro);
EQM_estQ= (estQ - parametro)*(estQ -
parametro);
EQM_estR= (estR - parametro)*(estR -
parametro);
EQM_estC= (estC - parametro)*(estC -
parametro);
EQM_estFH= (estFH - parametro)*(estFH -
parametro);
EQM_estNS= (estNS - parametro)*(estNS -
parametro);
EQM_estRY= (estRY - parametro)*(estRY -
parametro);
EQM_estPC= (estPC - parametro)*(estPC -
parametro);
EQM_estPCRI= (estPCRI - parametro)*(estPCRI -
parametro);
EQM_estPCRII= (estPCRII - parametro)*(estPCRII
- parametro);

/* Variância (V) */
V_y= (y - ym)*(y - ym);
V_estQ= (estQ - estQm)*(estQ - estQm);
V_estR= (estR - estRm)*(estR - estRm);
V_estC= (estC - estCm)*(estC - estCm);
V_estFH= (estFH - estFHm)*(estFH - estFHm);
V_estNS= (estNS - estNSm)*(estNS - estNSm);
V_estRY= (estRY - estRYm)*(estRY - estRYm);
V_estPC= (estPC - estPCm)*(estPC - estPCm);
V_estPCRI= (estPCRI - estPCRIm)*(estPCRI -
estPCRIm);
V_estPCRII= (estPCRII - estPCRIIm)*(estPCRII -
estPCRIIm); run;

/* Para o cálculo dos IC design-based e IC
model-based*/
Proc sort data=auxiliar out=auxiliar1;
by t i; run;
Proc means data=auxiliar1;
by t i;
var V_y V_estQ V_estR V_estC V_estFH V_estNS
V_estRY V_estPC V_estPCRI V_estPCRII;
output out=auxiliar2 mean=; run;

Proc iml;
use auxiliar2; read all into auxiliar2;
J1000=J(1000,1,1);
J196=J(196,1,1);
v1000=(1:1000);
aux1=J1000@auxiliar2;
aux2=t(v1000)@J196;
auxiliar3=aux2||aux1;
create auxiliar3 from auxiliar3; append from
auxiliar3; quit;
Data auxiliar3;
set auxiliar3;
rename col1=simul col2=t col3=i col6=Vm_y
col7=Vm_estQ col8=Vm_estR col9=Vm_estC
col10=Vm_estFH col11=Vm_estNS col12=Vm_estRY
col13=Vm_estPC col14=Vm_estPCRI
col15=Vm_estPCRII;
drop col4 col5; run;
Data auxiliar4;
merge auxiliar auxiliar3;
by simul t i;

/*Intervalo de confiança model-based (ICDB) */
if y - 1.96*sqrt(Vm_y) <= parametro <= y +
1.96*sqrt(Vm_y) then ICDB_y=1; else ICDB_y=0;
if estQ - 1.96*sqrt(Vm_estQ) <= parametro <=
estQ + 1.96*sqrt(Vm_estQ) then ICDB_estQ=1;
else ICDB_estQ=0;
if estR - 1.96*sqrt(Vm_estR) <= parametro <=
estR + 1.96*sqrt(Vm_estR) then ICDB_estR=1;
else ICDB_estR=0;
if estC - 1.96*sqrt(Vm_estC) <= parametro <=
estC + 1.96*sqrt(Vm_estC) then ICDB_estC=1;
else ICDB_estC=0;
if estFH - 1.96*sqrt(Vm_estFH) <= parametro <=
estFH + 1.96*sqrt(Vm_estFH) then ICDB_estFH=1;
else ICDB_estFH=0;
if estNS - 1.96*sqrt(Vm_estNS) <= parametro <=
estNS + 1.96*sqrt(Vm_estNS) then ICDB_estNS=1;
else ICDB_estNS=0;
if estRY - 1.96*sqrt(Vm_estRY) <= parametro <=
estRY + 1.96*sqrt(Vm_estRY) then ICDB_estRY=1;
else ICDB_estRY=0;
if estPC - 1.96*sqrt(Vm_estPC) <= parametro <=
estPC + 1.96*sqrt(Vm_estPC) then ICDB_estPC=1;
else ICDB_estPC=0;
if estPCRI - 1.96*sqrt(Vm_estPCRI) <=
parametro <= estPCRI + 1.96*sqrt(Vm_estPCRI)
then ICDB_estPCRI=1; else ICDB_estPCRI=0;
```

```
if estPCRII - 1.96*sqrt(Vm_estPCRII) <=
parametro <= estPCRII + 1.96*sqrt(Vm_estPCRII)
then ICDB_estPCRII=1; else ICDB_estPCRII=0;

/*Intervalo de confiança model-based (ICMB) */
if estFH - 1.96*sqrt(EQMP_estFH) <= parametro
<= estFH + 1.96*sqrt(EQMP_estFH) then
ICMB_estFH=1; else ICMB_estFH=0;
if estNS - 1.96*sqrt(EQMP_estNS) <= parametro
<= estNS + 1.96*sqrt(EQMP_estNS) then
ICMB_estNS=1; else ICMB_estNS=0;
if estRY - 1.96*sqrt(EQMP_estRY) <= parametro
<= estRY + 1.96*sqrt(EQMP_estRY) then
ICMB_estRY=1; else ICMB_estRY=0;
if estPC - 1.96*sqrt(EQMP_estPC) <= parametro
<= estPC + 1.96*sqrt(EQMP_estPC) then
ICMB_estPC=1; else ICMB_estPC=0; run;

/*********************************************/
/**** RESULTADOS POR TRIMESTRE - DOMÍNIO ****/
/*********************************************/
Proc sort data=auxiliar4 out=auxiliar5;
by t i; run;
Proc means data=auxiliar5;
by t i;
var parametro
/* Enviesamento (B) */
B_y B_estQ B_estR B_estC B_estFH B_estNS
B_estRY B_estPC B_estPCRI B_estPCRII
/* Enviesamento relativo (BR) */
BR_y BR_estQ BR_estR BR_estC BR_estFH BR_estNS
BR_estRY BR_estPC BR_estPCRI BR_estPCRII
/* Erro absoluto (EA) */
EA_y EA_estQ EA_estR EA_estC EA_estFH EA_estNS
EA_estRY EA_estPC EA_estPCRI EA_estPCRII
/* Erro relativo absoluto (ERA) */
ERA_y ERA_estQ ERA_estR ERA_estC ERA_estFH
ERA_estNS ERA_estRY ERA_estPC ERA_estPCRI
ERA_estPCRII
/* Erro Quadrático Médio (EQM) */
EQM_y EQM_estQ EQM_estR EQM_estC EQM_estFH
EQM_estNS EQM_estRY EQM_estPC EQM_estPCRI
EQM_estPCRII
/* Variância (V) */
V_y V_estQ V_estR V_estC V_estFH V_estNS
V_estRY V_estPC V_estPCRI V_estPCRII;
output out=Result_simul1000_ti mean=; run;

Proc means data=auxiliar5;
by t i;
var
/* IC design-based (ICDB) */
ICDB_y ICDB_estQ ICDB_estR ICDB_estC
ICDB_estFH ICDB_estNS ICDB_estRY ICDB_estPC
ICDB_estPCRI ICDB_estPCRII
/* IC design-based (ICMB) */
ICMB_estFH ICMB_estNS ICMB_estRY ICMB_estPC;
output out=auxiliar6 sum=;
run;
Data Result_simul1000_ti;
merge Result_simul1000_ti auxiliar6;
by t i; run;
Data HB.Result_simul1000_ti;
set Result_simul1000_ti;

/* Enviesamento relativo absoluto (BRA) */
BRA_y=abs(BR_y);
BRA_estQ=abs(BR_estQ);
BRA_estR=abs(BR_estR);
BRA_estC=abs(BR_estC);
BRA_estFH=abs(BR_estFH);
BRA_estNS=abs(BR_estNS);
BRA_estRY=abs(BR_estRY);
BRA_estPC=abs(BR_estPC);
BRA_estPCRI=abs(BR_estPCRI);
BRA_estPCRII=abs(BR_estPCRII);

/* Erro padrão relativo (EPR) */
EPR_y=sqrt(EQM_y)/parametro;
EPR_estQ=sqrt(EQM_estQ)/parametro;
EPR_estR=sqrt(EQM_estR)/parametro;
EPR_estC=sqrt(EQM_estC)/parametro;
EPR_estFH=sqrt(EQM_estFH)/parametro;
EPR_estNS=sqrt(EQM_estNS)/parametro;
EPR_estRY=sqrt(EQM_estRY)/parametro;
EPR_estPC=sqrt(EQM_estPC)/parametro;
EPR_estPCRI=sqrt(EQM_estPCRI)/parametro;
EPR_estPCRII=sqrt(EQM_estPCRII)/parametro;

/* Rácio de enviesamento (RB) */
RBA_y=abs(B_y/sqrt(V_y));
RBA_estQ=abs(B_estQ/sqrt(V_estQ));
RBA_estR=abs(B_estR/sqrt(V_estR));
RBA_estC=abs(B_estC/sqrt(V_estC));
RBA_estFH=abs(B_estFH/sqrt(V_estFH));
RBA_estNS=abs(B_estNS/sqrt(V_estNS));
RBA_estRY=abs(B_estRY/sqrt(V_estRY));
RBA_estPC=abs(B_estPC/sqrt(V_estPC));
RBA_estPCRI=abs(B_estPCRI/sqrt(V_estPCRI));
RBA_estPCRII=abs(B_estPCRII/sqrt(V_estPCRII));

/* Coeficiente de Variação (CV) */
CV_y=sqrt(V_y)/parametro;
CV_estQ=sqrt(V_estQ)/parametro;
CV_estR=sqrt(V_estR)/parametro;
CV_estC=sqrt(V_estC)/parametro;
CV_estFH=sqrt(V_estFH)/parametro;
CV_estNS=sqrt(V_estNS)/parametro;
CV_estRY=sqrt(V_estRY)/parametro;
CV_estPC=sqrt(V_estPC)/parametro;
CV_estPCRI=sqrt(V_estPCRI)/parametro;
CV_estPCRII=sqrt(V_estPCRII)/parametro;

/* Eficiência relativa (EFR) */
EFR_y=sqrt(EQM_y/EQM_y);
EFR_estQ=sqrt(EQM_y/EQM_estQ);
EFR_estR=sqrt(EQM_y/EQM_estR);
EFR_estC=sqrt(EQM_y/EQM_estC);
EFR_estFH=sqrt(EQM_y/EQM_estFH);
EFR_estNS=sqrt(EQM_y/EQM_estNS);
EFR_estRY=sqrt(EQM_y/EQM_estRY);
EFR_estPC=sqrt(EQM_y/EQM_estPC);
EFR_estPCRI=sqrt(EQM_y/EQM_estPCRI);
EFR_estPCRII=sqrt(EQM_y/EQM_estPCRII);

/*Taxa de cobertura do IC design-based
(TCDB)*/
TCDB_y=ICDB_y/1000;
TCDB_estQ=ICDB_estQ/1000;
TCDB_estR=ICDB_estR/1000;
TCDB_estC=ICDB_estC/1000;
TCDB_estFH=ICDB_estFH/1000;
TCDB_estNS=ICDB_estNS/1000;
TCDB_estRY=ICDB_estRY/1000;
TCDB_estPC=ICDB_estPC/1000;
TCDB_estPCRI=ICDB_estPCRI/1000;
TCDB_estPCRII=ICDB_estPCRII/1000;

/*Taxa de cobertura do IC model-based (TCMB)*/
TCMB_estFH=ICMB_estFH/1000;
TCMB_estNS=ICMB_estNS/1000;
TCMB_estRY=ICMB_estRY/1000;
TCMB_estPC=ICMB_estPC/1000; run;

/*********************************************/
/********* RESULTADOS POR DOMÍNIO ***********/
/*********************************************/
Proc sort data=HB.Result_simul1000_ti
out=Result_simul1000_ti;
by i; run;
Proc means data=Result_simul1000_ti;
by i; var ;
output out=Hb.Result_simul1000_i mean=; run;

/*********************************************/
/*** RESULTADOS POR GRUPO DE DOMÍNIOS *******/
/*********************************************/
Data Result_simul1000_i;
merge Hb.Result_simul1000_i Hb.grupo;
by i;
drop cod_nut_iii_trans; run;
Proc sort data=Result_simul1000_i;
by grupo; run;
Proc means data=Result_simul1000_i;
by grupo;
var ;
output out=Hb.Result_simul1000_grupo mean=;
run;

/*********************************************/
/************* RESULTADOS GLOBAIS ***********/
/*********************************************/
Proc means data=HB.Result_simul1000_ti;
var ;
output out=Hb.Result_simul1000_global mean=;
run;
```

Apêndice 20 – Valores dos parâmetros na pseudo-população

A20.1 Verdadeiros valores dos parâmetros média, variância e CV (%) na pseudo-população, em cada domínio de cada trimestre

t	*i*	N_{ti}	Média	Variância	CV	*t*	*i*	N_{ti}	Média	Variância	CV
1	1	102	751	66.678	34,4%	3	1	99	853	97.409	36,6%
1	2	586	708	33.211	25,7%	3	2	290	672	34.873	27,8%
1	3	445	732	16.913	17,8%	3	3	472	732	14.479	16,4%
1	4	5.517	1.040	90.705	28,9%	3	4	4.284	986	86.988	29,9%
1	5	245	582	6.187	13,5%	3	5	321	644	9.753	15,3%
1	6	279	769	33.339	23,8%	3	6	220	731	6.077	10,7%
1	7	135	711	36.733	27,0%	3	7	259	626	35.333	30,0%
1	8	67	697	10.334	14,6%	3	8	69	766	8.723	12,2%
1	9	1.019	855	68.641	30,6%	3	9	656	825	56.550	28,8%
1	10	987	979	117.697	35,0%	3	10	584	877	112.912	38,3%
1	11	929	737	35.572	25,6%	3	11	542	706	24.721	22,3%
1	12	170	708	25.953	22,8%	3	12	170	708	22.473	21,2%
1	13	264	769	81.505	37,1%	3	13	209	644	31.374	27,5%
1	14	2	377	0	0,0%	3	14	13	699	20.436	20,5%
1	15	4	500	0	0,0%	3	15	4	500	0	0,0%
1	16	12	464	21.005	31,3%	3	16	12	708	6.485	11,4%
1	17	42	832	31.358	21,3%	3	17	30	741	60.568	33,2%
1	18	50	559	34.043	33,0%	3	18	74	555	8.702	16,8%
1	19	797	887	54.210	26,3%	3	19	802	929	93.196	32,9%
1	20	8.357	1.264	177.106	33,3%	3	20	6.669	1.282	179.952	33,1%
1	21	3.970	1.024	71.338	26,1%	3	21	2.973	1.002	89.577	29,9%
1	22	258	605	21.572	24,3%	3	22	342	694	69.999	38,1%
1	23	421	789	29.252	21,7%	3	23	598	705	48.546	31,3%
1	24	108	1.062	144.454	35,8%	3	24	333	1.622	261.417	31,5%
1	25	79	802	44.071	26,2%	3	25	55	690	39.336	28,8%
1	26	146	1.067	39.650	18,7%	3	26	216	945	79.790	29,9%
1	27	70	873	18.125	15,4%	3	27	130	912	33.633	20,1%
1	28	1.255	959	146.072	39,8%	3	28	1.551	1.166	296.212	46,7%
2	1	93	847	112.753	39,7%	4	1	91	810	96.420	38,3%
2	2	425	820	23.742	18,8%	4	2	155	679	17.772	19,6%
2	3	278	711	17.420	18,6%	4	3	149	689	16.938	18,9%
2	4	5.819	968	60.063	25,3%	4	4	3.502	1.020	109.799	32,5%
2	5	280	665	26.669	24,5%	4	5	209	630	23.043	24,1%
2	6	199	732	25.015	21,6%	4	6	300	673	18.439	20,2%
2	7	141	681	15.340	18,2%	4	7	133	696	3.714	8,8%
2	8	73	799	40.043	25,1%	4	8	86	689	12.241	16,1%
2	9	706	889	96.051	34,9%	4	9	552	866	82.899	33,2%
2	10	798	928	95.229	33,2%	4	10	538	960	79.939	29,5%
2	11	672	711	19.642	19,7%	4	11	431	685	25.943	23,5%
2	12	86	631	24.691	24,9%	4	12	122	722	27.450	22,9%
2	13	174	707	33.870	26,0%	4	13	207	722	53.557	32,0%
2	14	2	377	0	0,0%	4	14	2	377	0	0,0%
2	15	4	500	0	0,0%	4	15	4	500	0	0,0%
2	16	18	522	30.394	33,4%	4	16	12	661	26.291	24,5%
2	17	15	688	41.481	29,6%	4	17	18	751	48.240	29,2%
2	18	20	488	32.586	37,0%	4	18	32	540	24.543	29,0%
2	19	627	906	66.804	28,5%	4	19	776	932	86.008	31,5%
2	20	7.466	1.319	191.292	33,2%	4	20	7.342	1.306	203.238	34,5%
2	21	3.454	1.084	116.660	31,5%	4	21	2.617	1.031	111.956	32,4%
2	22	266	685	34.267	27,0%	4	22	219	731	45.124	29,1%
2	23	353	789	35.618	23,9%	4	23	216	758	41.853	27,0%
2	24	76	1.121	131.780	32,4%	4	24	382	1.647	315.306	34,1%
2	25	31	795	51.870	28,7%	4	25	37	735	111.617	45,4%
2	26	183	1.016	96.295	30,5%	4	26	105	1.053	59.917	23,2%
2	27	80	791	48.890	27,9%	4	27	63	856	47.315	25,4%
2	28	1.515	1.044	213.995	44,3%	4	28	1.420	1.028	137.300	36,0%

t	*i*	N_{ti}	Média	Variância	CV
5	1	80	752	50.090	29,7%
5	2	262	810	82.181	35,4%
5	3	153	708	15.964	17,8%
5	4	3.259	1.030	105.096	31,5%
5	5	136	644	9.136	14,8%
5	6	202	694	22.294	21,5%
5	7	133	652	8.110	13,8%
5	8	73	662	5.843	11,5%
5	9	599	783	49.684	28,5%
5	10	479	891	96.700	34,9%
5	11	332	758	40.314	26,5%
5	12	77	630	42.937	32,9%
5	13	312	734	29.786	23,5%
5	14	10	408	270	4,0%
5	15	4	500	0	0,0%
5	16	9	710	5.786	10,7%
5	17	15	796	83.749	36,3%
5	18	29	465	9.859	21,3%
5	19	833	946	77.694	29,5%
5	20	6.616	1.322	172.031	31,4%
5	21	2.168	1.036	103.109	31,0%
5	22	137	752	42.074	27,3%
5	23	186	735	31.127	24,0%
5	24	257	1.426	168.712	28,8%
5	25	43	797	51.963	28,6%
5	26	99	789	52.520	29,0%
5	27	48	919	31.535	19,3%
5	28	1.135	1.088	193.169	40,4%
6	1	87	812	71.392	32,9%
6	2	330	801	47.900	27,3%
6	3	144	740	35.771	25,6%
6	4	3.785	1.013	78.464	27,7%
6	5	143	649	17.398	20,3%
6	6	186	712	24.192	21,8%
6	7	63	608	2.537	8,3%
6	8	55	781	5.972	9,9%
6	9	596	829	61.122	29,8%
6	10	463	994	111.709	33,6%
6	11	493	739	58.488	32,7%
6	12	71	572	24.716	27,5%
6	13	192	744	92.327	40,8%
6	14	2	377	0	0,0%
6	15	4	500	0	0,0%
6	16	12	749	4.154	8,6%
6	17	18	793	65.213	32,2%
6	18	20	452	34.948	41,4%
6	19	488	1.020	47.660	21,4%
6	20	6.979	1.383	181.049	30,8%
6	21	2.662	1.063	110.075	31,2%
6	22	186	704	21.264	20,7%
6	23	289	786	23.776	19,6%
6	24	122	1.031	104.716	31,4%
6	25	36	758	59.045	32,0%
6	26	164	932	53.403	24,8%
6	27	56	956	57.286	25,0%
6	28	1.367	1.212	308.986	45,8%
7	1	100	770	92.851	39,6%
7	2	318	653	35.325	28,8%
7	3	258	736	17.815	18,1%
7	4	4.527	1.058	145.359	36,0%
7	5	209	625	13.587	18,7%
7	6	194	687	22.804	22,0%
7	7	133	626	2.779	8,4%
7	8	123	744	37.046	25,9%
7	9	570	858	83.000	33,6%
7	10	562	944	79.370	29,9%
7	11	588	729	32.185	24,6%
7	12	109	650	19.010	21,2%
7	13	156	852	209.205	53,7%
7	14	2	377	0	0,0%
7	15	4	500	0	0,0%
7	16	18	726	46.949	29,8%
7	17	36	782	36.754	24,5%
7	18	48	653	27.332	25,3%
7	19	665	949	63.087	26,5%
7	20	6.813	1.388	181.566	30,7%
7	21	2.375	1.089	105.667	29,9%
7	22	215	825	29.778	20,9%
7	23	308	814	17.624	16,3%
7	24	340	1.358	92.395	22,4%
7	25	92	794	59.963	30,9%
7	26	126	967	64.394	26,3%
7	27	51	1.075	34.730	17,3%
7	28	1.341	1.188	220.812	39,6%

Apêndice 21 – Medidas da qualidade dos estimadores num estudo por simulação *design-based* com uma amostra de 458 empresas (*L*=1.000)

A21.1 Enviesamento e EQM

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$
	Enviesamento				**EQM**			
	***t*=1**							
111	-8,9	115,3	-54,5	-63,1	4759	14725	3086	6622
112	-2,7	86,2	-80,9	-28,2	1259	8629	6719	2822
113	0,5	107,1	-61,7	-7,1	195	12794	3946	304
114	-9,1	74,1	-104,9	-28,9	1335	7837	11224	2991
115	2,4	248,9	80,4	23,2	1319	63262	6611	2609
116	1,6	119,1	-51,4	-39,6	1751	15686	2759	3184
117	-14,0	71,9	-94,8	-55,0	5492	6327	9174	5488
118	-14,6	55,5	-110,1	-41,7	3370	4152	12343	5376
161	-6,4	-13,0	-76,4	-35,7	1616	914	5949	3345
162	4,4	52,7	6,1	6,7	4635	3899	342	2237
163	-3,7	95,2	30,8	19,2	1972	9782	1058	2271
164	1,1	-1,1	-76,5	-30,6	1903	525	6016	2429
165	-3,6	-15,5	-86,8	-58,0	6094	836	7658	10293
166	0,0	360,5	287,8	1,7	0	130554	82986	3
167	0,0	220,9	146,8	3,4	0	49353	21699	12
168	0,9	155,5	72,3	15,2	818	24576	5526	1305
169	-19,3	-79,3	-150,7	-59,7	4447	6890	22839	10532
16A	-28,6	187,9	116,1	45,2	10211	35906	13607	18922
16B	-9,8	115,9	68,7	33,4	2940	14447	4974	5222
171	3,4	2,3	-21,1	-25,2	2717	1626	1448	2628
172	-1,8	71,3	32,5	22,9	1116	6293	1501	1748
16C	1,4	216,2	152,6	10,2	785	47430	23408	1083
185	-1,7	133,1	78,6	38,2	3274	18587	6338	6044
181	-19,4	43,2	-129,9	-115,8	20202	5268	17097	22221
182	-16,5	136,2	-31,5	-51,2	7142	21010	1102	5822
183	-4,0	-27,3	-198,3	-41,5	6475	3759	39477	6744
184	-10,3	-34,4	-199,0	-25,7	1812	3142	39719	2753
150	5,2	5,2	73,9	49,3	2463	2463	5850	5576

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$
	Enviesamento				**EQM**			
	***t*=2**							
111	-1,7	32,5	-117,2	-81,2	6016	1372	13855	11612
112	-3,3	-34,9	-188,6	-12,1	831	1473	35764	1302
113	0,2	105,8	-46,5	-18,6	593	11474	2329	1125
114	1,0	123,5	-17,2	-10,4	481	15746	566	597
115	2,6	127,8	-25,4	-23,3	2116	16601	847	2545
116	-3,3	118,7	-32,2	-28,0	1136	14376	1176	1685
117	-0,7	104,7	-48,9	-16,5	340	11210	2606	554
118	-9,5	-48,8	-203,9	-28,4	2460	2613	41858	4148
161	-12,1	-64,5	-99,0	-52,9	4141	4892	9920	7459
162	0,9	75,4	73,9	40,3	1579	6763	5833	4254
163	-0,3	125,9	93,8	6,2	316	16613	8914	395
164	-3,2	77,4	21,4	9,5	1033	6523	643	1036
165	-8,4	44,5	-3,5	9,3	3508	2584	149	1882
166	0,0	283,3	218,5	2,3	0	80741	47998	5
167	0,0	205,2	148,6	3,4	0	42627	22283	12
168	-0,3	89,1	15,3	-11,1	11229	8336	620	9744
169	3,0	88,7	45,4	71,6	4764	8509	2177	7261
16A	-12,6	241,5	189,4	24,8	4507	58902	36038	6146
16B	-3,4	108,2	19,7	18,8	2045	13791	617	1573
171	3,7	33,6	-22,3	-35,6	3983	4814	1963	3310
172	4,2	70,0	-5,0	-13,7	3756	7584	602	1646
16C	2,4	180,1	77,1	22,4	817	33943	6065	1938
185	-5,1	178,3	85,3	43,6	4411	33690	7436	8124
181	-10,2	23,5	-127,3	-119,2	18024	4188	16556	25753
182	21,7	110,9	-47,3	-18,8	12321	14576	2356	9649
183	-1,5	-15,4	-170,7	-56,4	4419	3021	29266	11076
184	-12,1	79,1	-80,2	-42,3	2633	8364	6566	4841
150	4,6	4,6	18,4	12,7	2843	2843	865	1899
NUTSIII	***t*=3**							
111	-5,6	24,4	-95,7	-90,5	8710	1028	9324	12786
112	1,6	122,7	0,9	-7,4	1841	15419	208	1088
113	-0,9	82,6	-38,9	-17,3	754	7191	1695	1177
114	0,3	87,0	-29,0	-12,4	373	8223	1353	653
115	18,1	118,4	-4,1	11,5	2495	14355	274	2333
116	-2,1	107,0	-13,9	-8,7	310	11852	361	438
117	-0,5	139,0	16,5	7,1	153	19645	525	280
118	-1,2	-18,8	-141,7	-7,6	335	669	20369	621
161	-3,4	-51,2	-33,2	-18,6	2747	3038	1273	2306
162	3,3	63,9	115,5	52,5	2413	4701	13982	7407
163	-1,1	56,5	72,1	21,8	797	3601	5362	1916
164	-5,4	-43,6	-47,9	-20,8	1233	2209	2516	1689
165	2,0	52,8	55,2	24,4	916	3134	3223	1822
166	-102,4	-88,1	-103,1	-113,8	35716	8019	10989	35367
167	0,0	138,8	129,5	3,9	0	19564	17038	16
168	-1,5	-53,0	-59,1	-6,3	19	3109	3729	103
169	-32,0	-21,7	-14,9	-30,8	12351	835	385	7124
16A	-37,9	160,0	166,0	-29,0	7753	25956	27737	6655
16B	-5,8	39,7	7,6	6,5	3626	2454	504	2083
171	2,8	-16,3	-15,7	-14,4	2059	1764	2606	3461
172	1,1	95,0	77,1	35,4	1460	10155	6999	3592
16C	-3,1	128,3	80,1	20,2	1135	17086	6584	1983
185	14,2	201,8	162,9	54,0	4929	41485	26809	10426
181	-149,5	-247,3	-654,7	-213,8	137481	120443	429244	156505
182	-6,0	466,9	102,2	56,0	4206	259970	10609	10067
183	3,1	260,9	-113,5	-52,6	3326	113709	13089	6510
184	-0,4	159,4	-188,6	-23,9	2201	61407	35748	3800
150	4,0	4,0	-56,3	-66,1	9007	9007	4399	8412

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	Enviesamento				**EQM**			
	***t*=4**							
111	0,8	104,8	-52,0	-77,6	7458	11695	2846	8928
112	1,4	153,6	-12,0	-9,9	1076	24196	258	743
113	-0,4	163,5	0,0	6,2	1065	27355	112	604
114	-2,1	90,1	-45,7	-20,3	589	9170	2668	1113
115	0,2	166,7	-2,8	-16,2	1658	28313	137	1364
116	5,5	205,9	45,2	24,5	1360	43044	2161	1855
117	0,0	102,5	-66,8	-7,7	0	11048	4591	62
118	-0,1	68,5	-105,3	-5,8	347	5176	11248	459
161	-7,7	-60,0	-87,7	-50,9	3307	4222	7853	5577
162	1,6	40,4	67,3	35,0	1629	2580	5293	3790
163	-2,8	114,2	84,5	29,3	1524	13657	7286	2956
164	-4,9	-24,1	-82,2	-17,1	629	1042	6876	1510
165	-3,8	12,1	-35,8	-57,2	4651	658	1397	4646
166	0,0	307,0	244,7	2,0	0	94678	60017	4
167	0,0	146,4	73,6	7,0	0	21822	5595	52
168	-2,0	10,7	-55,0	-42,6	4205	542	3172	7154
169	-1,6	8,5	-32,4	-18,5	5594	620	1166	4672
16A	-28,4	210,2	166,6	0,3	7093	44732	27881	7073
16B	-6,2	106,4	31,3	27,8	3563	12609	1518	3175
171	2,2	29,3	5,7	9,3	3231	3006	2309	3566
172	5,3	114,4	57,8	34,7	2352	14655	4356	4429
16C	-0,6	130,4	24,6	16,8	2590	17895	747	2439
185	-1,5	202,0	113,3	50,5	3827	41927	13138	6892
181	-194,9	-220,0	-695,6	-232,1	189320	143913	484327	182981
182	15,5	548,0	94,6	97,7	11736	377588	9188	18424
183	-8,1	255,2	-202,0	-34,9	2697	145413	41052	6017
184	-2,1	350,8	-90,7	-37,1	1592	191319	8378	5038
150	5,6	5,6	95,4	42,4	2278	2278	10283	5443
NUTSIII	***t*=5**							
111	-2,3	177,0	-7,6	-2,6	2101	32006	180	1084
112	-5,1	29,6	-167,9	-62,4	6632	1420	28272	9860
113	5,7	148,1	-47,0	-24,6	1039	22505	2301	1988
114	-2,4	109,9	-44,7	-20,8	648	13087	2689	1163
115	-3,3	174,1	-26,5	-19,9	481	30835	808	709
116	-1,2	203,0	13,6	16,9	759	41818	283	760
117	0,0	147,9	-55,2	-9,5	0	22358	3176	95
118	-0,9	114,9	-91,5	-4,6	212	13666	8522	262
161	-2,5	5,6	26,3	27,6	1680	507	897	1817
162	-0,8	57,3	146,5	29,4	1835	3966	22374	4219
163	-3,6	12,5	25,3	11,7	1650	609	807	1501
164	-13,3	41,6	12,0	6,6	4405	2077	239	2694
165	1,6	-11,3	-19,1	-26,6	3384	527	464	2627
166	-8,7	222,2	174,9	-7,9	308	49678	30727	263
167	0,0	167,8	136,6	3,7	0	28489	18766	14
168	11,0	-62,5	-102,4	-10,0	3111	4230	10600	3781
169	4,3	-63,5	-66,8	-56,9	9788	4438	4571	11948
16A	-10,0	238,6	222,9	2,5	1833	57285	49753	1397
16B	-3,1	122,2	3,5	7,9	3712	16026	575	2066
171	0,0	-22,2	-97,9	-38,7	1848	2095	11543	4555
172	2,8	154,9	58,9	41,4	4115	25330	4656	6542
16C	-6,8	193,9	52,3	29,3	2704	38429	2934	4443
185	-7,9	218,6	78,6	36,9	2527	48645	6384	4223
181	-142,0	-134,7	-440,7	-171,3	109572	76425	194880	96922
182	-0,5	237,0	-71,3	-17,4	523	93499	5184	753
183	-0,3	333,7	26,3	9,6	4444	155472	907	3289
184	-2,0	121,8	-186,4	-30,5	2323	52722	34858	4823
150	3,1	3,1	55,1	41,1	3277	3277	4486	5534

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$
	Enviesamento				**EQM**			
	***t*=6**							
111	0,3	145,7	-32,5	-46,2	2449	21740	1142	3197
112	3,7	35,2	-144,6	-25,8	2406	1618	21028	2938
113	2,2	120,2	-59,2	-33,5	1449	14861	3606	3361
114	-0,2	141,5	-34,1	-16,4	452	20756	1491	788
115	-1,3	155,7	-24,5	-12,2	329	24604	738	489
116	-4,4	197,1	18,3	12,1	641	39314	418	793
117	0,3	192,8	12,6	5,2	18	37533	302	166
118	16,6	-4,7	-185,3	6,8	2457	351	34505	1854
161	-5,1	-30,1	-54,4	-38,5	1871	1451	3047	3250
162	3,3	-11,7	2,2	-1,1	2996	960	372	1616
163	-0,7	137,6	129,4	24,5	1643	19598	16903	3405
164	-6,1	122,9	76,9	24,8	1479	15508	6025	2737
165	-10,8	19,0	-12,8	-25,3	7571	858	246	4149
166	0,0	324,9	280,2	1,8	0	105950	78638	3
167	0,0	237,0	199,7	2,5	0	56614	39976	6
168	-3,7	-98,1	-153,2	-19,3	985	9976	23646	2154
169	-4,2	-70,1	-110,3	-79,6	8234	5365	12262	15907
16A	-5,2	276,2	237,1	41,9	4662	76716	56305	11373
16B	-0,1	19,9	-88,2	-33,8	1651	1530	8023	3776
171	1,5	54,0	-22,5	-32,7	2406	5084	2256	3231
172	2,3	132,9	37,1	29,6	2677	19150	2019	3314
16C	6,1	251,7	136,9	17,7	790	64294	18863	1722
185	0,8	163,1	47,8	27,6	1642	27547	2401	2608
181	-4,1	162,5	30,6	0,9	14504	29807	1467	8587
182	5,2	109,6	-38,3	-51,6	3077	13819	1551	4289
183	2,4	17,4	-126,5	-40,8	3141	2452	16114	4607
184	8,1	-26,6	-171,5	-15,0	2904	2773	29518	2260
150	0,8	0,8	-109,7	-86,9	8493	8493	12687	13250
NUTSIII	***t*=7**							
111	-1,7	196,1	23,0	6,8	4535	41469	634	2923
112	2,8	230,9	58,1	34,5	1627	55821	3452	3442
113	-1,3	121,7	-51,0	-23,8	923	17181	2681	1997
114	1,3	89,9	-84,0	-42,4	4220	12305	7384	6545
115	2,6	212,8	40,1	24,2	1537	47523	1696	2002
116	-5,5	214,2	41,3	20,3	901	48475	1785	1785
117	0,0	164,3	-8,2	-12,2	0	28986	172	376
118	-0,3	21,6	-150,7	-1,3	6	2350	22845	7
161	-3,6	-7,7	-22,0	-19,8	3059	542	623	1970
162	1,6	25,2	32,4	22,1	1406	1261	1380	1836
163	-0,4	92,0	72,3	27,1	980	8913	5343	2392
164	-5,8	68,2	29,8	17,9	912	4998	969	1259
165	-7,3	-91,6	-122,3	-31,1	1546	8769	15039	4426
166	0,0	363,5	329,1	1,5	0	132524	108407	2
167	0,0	152,8	102,5	4,9	0	23628	10613	25
168	3,9	-39,1	-83,2	-1,8	379	1842	7011	402
169	14,9	-55,4	-92,2	-62,8	5209	3418	8586	10868
16A	-4,9	118,2	89,4	37,3	1483	14360	8074	3228
16B	-3,6	97,3	48,9	42,2	2468	10128	2761	4491
171	0,2	27,0	21,1	14,6	1492	1917	2190	2561
172	0,2	80,5	46,3	34,5	2470	7291	2843	3932
16C	-2,0	140,8	83,2	34,4	1621	20394	7139	4137
185	-6,6	135,0	75,4	18,3	1220	18773	5883	1257
181	-71,6	-34,7	-325,2	-119,4	34042	26166	106206	51171
182	-3,9	231,9	-31,2	-15,9	6526	68773	1066	4287
183	1,3	235,4	-44,0	-36,8	3345	75974	2171	4249
184	3,9	-60,0	-322,2	-20,9	6484	18279	103930	7543
150	2,9	2,9	-19,8	-27,6	4044	4044	1192	3115

A21.2 Variância e Erro absoluto

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$
	Variância				**Erro absoluto**			
	***t*=1**							
111	4681	1422	120	2646	53,5	115,3	54,5	69,8
112	1252	1195	171	2026	28,4	86,2	80,9	42,6
113	195	1334	135	254	10,8	107,1	61,7	13,1
114	1252	2352	227	2158	28,5	77,0	104,9	41,2
115	1313	1307	141	2071	29,5	248,9	80,4	42,5
116	1748	1493	113	1613	32,7	119,1	51,4	49,3
117	5296	1160	182	2459	57,9	72,1	94,8	65,0
118	3158	1074	217	3640	44,2	56,7	110,1	59,6
161	1575	744	108	2071	30,4	23,7	76,4	45,2
162	4615	1118	304	2192	54,9	54,7	15,0	37,5
163	1959	726	107	1904	35,6	95,2	30,9	40,6
164	1901	523	167	1496	31,0	18,3	76,5	37,6
165	6080	595	128	6931	63,9	22,8	86,8	90,0
166	0	570	139	0	0,0	360,5	287,8	1,7
167	0	545	153	0	0,0	220,9	146,8	3,4
168	817	402	293	1073	22,2	155,5	72,3	31,6
169	4074	594	129	6966	48,1	79,3	150,7	76,8
16A	9390	585	132	16874	73,6	187,9	116,1	130,1
16B	2843	1016	255	4109	42,9	115,9	68,7	62,2
171	2705	1621	1004	1991	41,7	32,3	30,7	41,4
172	1112	1212	447	1226	26,6	71,8	33,6	34,4
16C	783	682	106	978	22,4	216,2	152,6	25,5
185	3271	859	152	4585	46,8	133,1	78,6	67,2
181	19826	3404	222	8817	114,1	59,2	129,9	134,3
182	6871	2455	107	3200	69,5	136,2	31,6	68,1
183	6459	3013	153	5022	67,4	49,2	198,3	71,5
184	1707	1961	133	2093	30,8	45,2	199,0	40,0
150	2435	2435	395	3150	39,5	39,5	73,9	62,1

NUTSIII	$\widehat{\mu}_i^{dir}$	$\widehat{\mu}_i^{sinQ}$	$\widehat{\mu}_i^{sinR}$	$\widehat{\mu}_i^{com}$	$\widehat{\mu}_i^{dir}$	$\widehat{\mu}_i^{sinQ}$	$\widehat{\mu}_i^{sinR}$	$\widehat{\mu}_i^{com}$
	Variância				**Erro absoluto**			
	***t*=2**							
111	6013	316	122	5019	61,2	33,0	117,2	90,2
112	820	252	211	1155	21,1	35,0	188,6	25,5
113	593	273	171	780	19,7	105,8	46,5	27,7
114	480	487	269	490	17,4	123,5	19,5	19,5
115	2110	257	200	2003	36,6	127,8	25,9	43,3
116	1125	296	139	902	26,8	118,7	32,2	34,6
117	340	252	211	281	14,2	104,7	48,9	18,0
118	2370	230	270	3343	36,6	48,8	203,9	46,7
161	3994	730	114	4660	48,3	64,6	99,0	70,5
162	1578	1082	372	2629	31,1	75,6	73,9	53,1
163	316	753	118	357	13,6	125,9	93,8	15,2
164	1023	539	185	946	24,8	77,4	22,1	26,2
165	3438	607	136	1796	44,7	45,1	9,7	31,7
166	0	468	269	0	0,0	283,3	218,5	2,3
167	0	534	189	0	0,0	205,2	148,6	3,4
168	11229	402	387	9622	81,3	89,1	20,1	75,1
169	4756	648	120	2137	54,0	88,7	45,4	74,8
16A	4347	571	158	5529	51,7	241,5	189,4	68,3
16B	2033	2074	228	1218	36,4	108,3	21,2	32,5
171	3970	3686	1468	2046	50,6	54,9	35,5	47,6
172	3738	2686	577	1459	49,3	72,8	19,6	32,2
16C	812	1508	113	1435	21,1	180,1	77,1	31,8
185	4385	1886	163	6226	53,6	178,3	85,3	77,9
181	17920	3637	353	11544	109,9	51,2	127,3	143,5
182	11852	2279	116	9294	78,9	111,0	47,3	75,3
183	4416	2783	139	7900	52,1	44,2	170,7	82,5
184	2486	2104	132	3055	33,8	80,2	80,2	50,5
150	2822	2822	526	1739	41,9	41,9	24,1	34,8
NUTSIII	***t*=3**							
111	8679	433	159	4594	74,4	26,9	95,7	99,4
112	1838	355	207	1033	33,1	122,7	11,9	26,4
113	754	373	184	876	18,7	82,6	38,9	25,1
114	373	647	513	498	15,2	87,0	31,5	20,4
115	2169	327	257	2201	35,6	118,4	13,1	37,4
116	305	395	167	363	13,7	107,0	15,3	16,5
117	153	329	253	230	9,0	139,0	19,8	11,7
118	334	314	287	564	12,0	21,3	141,7	15,2
161	2735	422	174	1961	42,3	51,2	33,2	40,6
162	2403	622	645	4655	39,1	64,0	115,5	68,7
163	796	409	165	1442	23,0	56,5	72,1	35,4
164	1205	310	224	1258	28,7	43,6	47,9	35,6
165	912	342	179	1227	22,4	52,9	55,2	32,6
166	25221	262	352	22427	117,9	88,1	103,1	125,9
167	0	287	276	0	0,0	138,8	129,5	3,9
168	17	302	241	63	3,0	53,0	59,1	6,5
169	11327	364	163	6174	78,8	24,2	16,0	60,5
16A	6319	360	165	5811	49,5	160,0	166,0	47,5
16B	3592	876	446	2040	47,5	42,2	17,8	37,2
171	2051	1498	2359	3255	36,3	33,8	41,7	47,4
172	1459	1125	1055	2341	30,4	95,0	77,1	49,8
16C	1125	632	164	1574	25,2	128,3	80,1	34,7
185	4728	768	274	7508	52,3	201,8	162,9	77,7
181	115131	59267	549	110804	276,5	278,9	654,7	298,5
182	4170	41958	174	6930	52,0	466,9	102,2	86,8
183	3316	45614	214	3740	43,4	271,6	113,5	62,9
184	2201	35991	164	3229	32,9	211,3	188,6	40,4
150	8991	8991	1228	4044	77,2	77,2	58,2	78,4

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$
	Variância				**Erro absoluto**			
	***t*=4**							
111	7457	711	143	2909	67,8	104,8	52,0	83,2
112	1074	589	114	644	26,5	153,6	13,5	21,9
113	1065	617	112	565	26,3	163,5	8,5	20,1
114	584	1046	576	701	18,9	90,4	46,2	26,3
115	1658	540	129	1103	33,3	166,7	9,4	31,8
116	1330	656	119	1254	29,1	205,9	45,2	36,1
117	0	542	128	2	0,0	102,5	66,8	7,7
118	347	487	167	425	15,3	68,6	105,3	18,4
161	3248	617	162	2986	45,4	60,2	87,7	63,1
162	1626	949	759	2567	31,7	43,2	67,4	48,9
163	1516	606	152	2100	31,3	114,2	84,5	45,2
164	605	463	122	1218	15,8	26,7	82,2	23,7
165	4636	512	112	1379	54,7	20,4	35,8	60,8
166	0	444	133	0	0,0	307,0	244,7	2,0
167	0	397	178	2	0,0	146,4	73,6	7,0
168	4201	428	145	5344	50,3	18,5	55,0	82,8
169	5591	548	119	4328	58,1	19,8	32,4	59,5
16A	6287	534	115	7073	56,6	210,2	166,6	69,0
16B	3525	1297	542	2404	46,7	106,4	32,9	47,1
171	3226	2145	2277	3481	45,7	44,9	38,5	47,3
172	2323	1578	1014	3222	39,4	114,4	58,3	54,6
16C	2589	892	141	2157	41,9	130,4	24,8	41,4
185	3824	1108	305	4338	49,5	202,0	113,3	69,1
181	151320	95506	506	129122	306,0	285,8	695,6	311,4
182	11495	77259	230	8871	85,4	548,0	94,6	118,5
183	2632	80265	266	4801	40,1	335,9	202,0	53,9
184	1588	68277	149	3661	29,8	360,3	90,7	51,6
150	2247	2247	1174	3643	37,5	37,5	95,4	58,4
NUTSIII	***t*=5**							
111	2096	668	122	1077	36,4	177,0	10,9	26,5
112	6607	546	95	5965	66,1	32,4	167,9	81,3
113	1006	567	90	1381	24,9	148,1	47,0	37,5
114	643	1005	692	729	20,0	109,9	45,4	27,2
115	470	517	108	311	17,7	174,1	26,5	22,9
116	758	622	97	473	21,7	203,0	14,4	22,5
117	0	495	124	5	0,0	147,9	55,2	9,5
118	211	467	152	240	11,8	114,9	91,5	13,6
161	1674	475	205	1054	31,9	18,0	26,6	35,3
162	1834	687	912	3356	34,3	57,6	146,5	49,5
163	1637	453	166	1364	33,2	19,9	25,5	32,4
164	4228	344	95	2651	49,8	41,8	13,1	37,9
165	3381	399	101	1918	42,9	18,3	19,2	38,8
166	232	303	138	201	10,8	222,2	174,9	10,1
167	0	341	97	0	0,0	167,8	136,6	3,7
168	2989	320	115	3682	40,8	62,5	102,4	55,2
169	9769	410	110	8706	77,4	63,5	66,8	97,9
16A	1733	378	91	1391	31,1	238,6	222,9	28,3
16B	3702	1083	563	2004	47,1	122,2	19,6	36,2
171	1848	1603	1962	3054	34,5	36,4	98,1	52,1
172	4107	1346	1188	4826	51,6	154,9	59,8	68,7
16C	2658	849	197	3582	41,4	193,9	52,3	57,8
185	2464	864	213	2860	39,4	218,6	78,6	56,0
181	89406	58280	693	67565	241,4	214,3	440,7	242,5
182	523	37330	107	450	18,7	250,4	71,3	22,4
183	4443	44092	217	3197	54,4	334,0	26,6	47,7
184	2319	37891	112	3893	37,4	210,4	186,4	48,4
150	3267	3267	1452	3844	44,9	44,9	57,2	61,1

NUTSIII	$\hat{\boldsymbol{\mu}}_i^{dir}$	$\hat{\boldsymbol{\mu}}_i^{sinQ}$	$\hat{\boldsymbol{\mu}}_i^{sinR}$	$\hat{\boldsymbol{\mu}}_i^{com}$	$\hat{\boldsymbol{\mu}}_i^{dir}$	$\hat{\boldsymbol{\mu}}_i^{sinQ}$	$\hat{\boldsymbol{\mu}}_i^{sinR}$	$\hat{\boldsymbol{\mu}}_i^{com}$
	Variância				**Erro absoluto**			
	***t*=6**							
111	2449	499	89	1065	38,6	145,7	32,5	48,3
112	2392	381	112	2275	38,0	35,9	144,6	44,2
113	1444	403	97	2237	29,7	120,2	59,2	47,3
114	452	726	329	518	17,0	141,5	34,5	23,5
115	328	353	139	340	12,4	155,7	24,7	16,4
116	621	451	83	648	19,9	197,1	18,4	23,2
117	18	350	143	139	3,3	192,8	14,3	8,0
118	2180	329	171	1808	30,6	14,9	185,3	26,4
161	1844	544	87	1766	32,9	32,3	54,4	47,6
162	2985	823	367	1614	43,4	24,6	15,7	32,6
163	1642	655	153	2804	32,7	137,6	129,4	46,3
164	1441	412	118	2124	28,0	122,9	76,9	39,9
165	7455	496	82	3508	69,7	24,0	13,4	49,5
166	0	420	112	0	0,0	324,9	280,2	1,8
167	0	463	89	0	0,0	237,0	199,7	2,5
168	971	361	172	1782	23,7	98,1	153,2	35,7
169	8216	446	96	9576	70,5	70,1	110,3	111,8
16A	4634	452	93	9619	54,1	276,2	237,1	85,7
16B	1651	1133	242	2632	32,4	31,6	88,2	48,9
171	2403	2164	1749	2163	39,7	59,7	37,6	45,8
172	2672	1498	642	2437	41,6	132,9	38,5	47,6
16C	753	958	127	1408	20,8	251,7	136,9	28,9
185	1641	944	120	1849	32,2	163,1	47,8	43,0
181	14487	3398	529	8586	97,9	162,5	32,4	75,1
182	3050	1798	84	1630	44,0	109,6	38,3	55,6
183	3135	2150	101	2942	43,7	39,1	126,5	56,4
184	2838	2063	91	2036	41,5	43,8	171,5	37,4
150	8493	8493	652	5697	73,9	73,9	109,7	101,5
NUTSIII	***t*=7**							
111	4532	3000	106	2877	52,8	196,1	23,0	43,1
112	1620	2507	80	2254	30,5	230,9	58,1	48,9
113	921	2361	81	1432	24,7	121,7	51,0	37,4
114	4219	4229	323	4743	53,3	91,3	84,0	70,6
115	1530	2252	85	1416	31,0	212,8	40,1	37,8
116	871	2606	82	1372	23,4	214,2	41,3	36,2
117	0	2006	105	228	0,0	164,3	10,8	17,1
118	5	1883	122	5	1,6	39,6	150,7	1,5
161	3045	483	138	1579	44,4	18,7	22,3	36,4
162	1403	626	327	1348	29,1	29,3	32,7	35,7
163	979	450	111	1656	25,3	92,0	72,3	40,3
164	878	344	82	939	19,6	68,2	29,8	27,5
165	1493	386	82	3457	27,6	91,6	122,3	46,2
166	0	366	80	0	0,0	363,5	329,1	1,5
167	0	284	115	0	0,0	152,8	102,5	4,9
168	364	315	93	399	14,2	39,3	83,2	16,3
169	4988	352	81	6928	53,3	55,4	92,2	96,0
16A	1459	397	84	1833	27,9	118,2	89,4	41,8
16B	2456	651	368	2713	39,7	97,3	48,9	59,1
171	1492	1190	1746	2347	30,7	35,7	38,3	40,7
172	2470	813	702	2739	40,0	80,6	46,9	52,6
16C	1617	555	219	2957	32,1	140,8	83,2	54,5
185	1177	536	195	922	27,1	135,0	75,4	29,8
181	28921	24960	442	36925	135,8	131,4	325,2	168,5
182	6511	14979	91	4033	65,3	232,0	31,2	53,3
183	3343	20576	239	2895	47,0	236,7	44,0	55,0
184	6468	14675	87	7105	66,8	104,9	322,2	75,3
150	4035	4035	801	2355	50,9	50,9	28,0	45,9

Apêndice 22 – Medidas da qualidade dos estimadores EBLUP com restrições (L=1.000)

A22.1 Enviesamento

NUTSIII	$\hat{\mu}_{it}^{LPR1}$	$\hat{\mu}_{it}^{LPR2}$	$\hat{\mu}_{it}^{LPR1}$	$\hat{\mu}_{it}^{LPR2}$	$\hat{\mu}_{it}^{LPR1}$	$\hat{\mu}_{it}^{LPR2}$	$\hat{\mu}_{it}^{LPR1}$	$\hat{\mu}_{it}^{LPR2}$
	t=1		t=2		t=3		t=4	
111	-90,4	-23,4	-146,3	-93,0	-144,8	-89,9	-86,4	-29,5
112	-223,8	0,5	-296,2	-80,2	-176,6	39,7	-171,6	44,0
113	-288,2	-5,8	-276,4	2,5	-286,0	-5,9	-237,4	41,1
114	-375,6	13,1	-314,9	26,1	-331,3	8,2	-339,1	3,1
115	-335,8	49,3	-402,0	-18,9	-390,2	-6,9	-379,4	4,0
116	-323,6	-38,0	-301,6	-23,5	-301,6	-23,5	-245,8	33,0
117	-323,8	-90,6	-268,1	-37,7	-238,2	-7,9	-268,0	-37,7
118	-297,5	-36,4	-354,8	-95,3	-305,6	-46,6	-264,1	-5,0
161	-267,8	-32,2	-289,7	-69,9	-242,7	-32,0	-277,9	-45,4
162	-76,2	46,6	-31,9	23,7	-0,6	43,7	-47,2	19,0
163	221,8	61,0	265,2	77,5	245,6	54,8	275,5	92,5
164	-5,8	-75,8	82,4	2,1	33,0	-48,6	36,7	-41,6
165	-382,2	-84,4	-305,4	-27,8	-274,2	-1,4	-329,0	-39,0
166	274,5	96,8	249,0	71,3	-13,5	-191,2	268,5	90,8
167	-30,8	13,7	-27,4	17,1	-44,7	-0,3	-53,6	-9,2
168	-59,6	26,1	-81,6	5,2	-125,9	-40,8	-109,0	-22,5
169	-65,6	-157,3	107,2	12,0	33,6	-63,9	45,2	-44,8
16A	49,6	2,7	119,7	62,6	70,0	10,9	95,5	41,3
16B	248,3	74,3	208,5	40,4	193,8	17,7	222,0	49,9
171	212,0	-7,5	200,9	17,5	193,9	-23,6	217,1	2,6
172	316,4	92,3	287,1	76,9	337,0	110,5	324,1	101,0
16C	352,9	102,7	304,3	55,1	301,7	50,4	273,1	22,6
185	490,6	142,8	489,4	145,3	553,4	202,9	522,0	175,2
181	108,2	30,7	104,0	-6,1	-389,4	28,1	-403,4	88,7
182	248,7	18,2	239,8	-0,2	359,6	193,4	377,2	326,9
183	196,9	-9,2	228,3	-9,2	273,4	225,3	204,3	76,1
184	51,0	-44,2	148,9	57,1	49,8	47,5	144,3	110,0
150	140,5	5,2	82,7	4,6	12,0	4,0	133,3	5,6

NUTSIII	$\widehat{\mu}_{it}^{LPR1}$	$\widehat{\mu}_{it}^{LPR2}$	$\widehat{\mu}_{it}^{LPR1}$	$\widehat{\mu}_{it}^{LPR2}$	$\widehat{\mu}_{it}^{LPR1}$	$\widehat{\mu}_{it}^{LPR2}$
	***t*=5**		***t*=6**		***t*=7**	
111	-27,4	26,7	-56,8	-7,3	-3,5	50,4
112	-282,0	-60,1	-272,1	-58,7	-107,3	108,6
113	-253,8	25,4	-269,3	7,1	-259,9	18,6
114	-333,1	13,3	-314,8	18,6	-371,3	-0,2
115	-385,2	-2,1	-389,1	-6,5	-331,8	51,4
116	-258,9	20,2	-265,2	12,3	-240,3	37,8
117	-248,7	-18,2	-232,3	-2,0	-231,6	-1,3
118	-246,8	12,4	-311,8	-52,8	-271,7	-12,7
161	-172,5	8,2	-234,2	-27,5	-205,9	-6,1
162	29,3	27,1	-64,9	-33,6	-27,3	5,6
163	234,1	26,7	316,0	120,2	279,8	82,5
164	95,9	-3,7	126,3	42,5	88,7	4,9
165	-295,1	-47,9	-288,1	-19,6	-363,5	-93,3
166	224,6	46,6	275,0	97,2	308,3	130,6
167	-17,6	26,7	7,3	51,6	-33,8	10,5
168	-123,0	-39,1	-171,3	-86,4	-120,8	-35,9
169	28,8	-88,4	-7,2	-107,5	13,8	-88,1
16A	157,1	92,3	179,8	119,1	59,4	-3,1
16B	218,7	54,2	155,0	-14,6	270,6	92,7
171	145,1	-30,6	210,6	11,3	220,9	1,3
172	345,0	141,8	336,6	118,9	334,1	95,6
16C	292,5	43,7	354,9	105,3	319,9	67,9
185	513,9	169,7	489,3	144,0	515,5	168,1
181	-151,2	0,3	309,3	61,6	-11,3	-26,4
182	274,6	50,0	291,5	21,5	295,1	139,7
183	417,8	274,6	286,9	8,4	358,5	189,4
184	81,8	24,8	92,7	-24,5	-35,9	-99,9
150	107,9	3,1	1,6	0,8	82,6	2,9

A22.2 EQM

NUTSIII	$\widehat{\mu}_{it}^{LPR1}$	$\widehat{\mu}_{it}^{LPR2}$	$\widehat{\mu}_{it}^{LPR1}$	$\widehat{\mu}_{it}^{LPR2}$	$\widehat{\mu}_{it}^{LPR1}$	$\widehat{\mu}_{it}^{LPR2}$	$\widehat{\mu}_{it}^{LPR1}$	$\widehat{\mu}_{it}^{LPR2}$
	***t*=1**		***t*=2**		***t*=3**		***t*=4**	
111	9.092	1.990	22.463	9.341	21.951	9.189	8.368	1.851
112	53.515	848	93.615	7.665	34.271	2.990	32.162	3.415
113	88.157	638	81.200	878	86.692	800	60.510	3.167
114	151.089	3.167	107.087	1.219	117.780	836	123.557	1.238
115	122.005	6.790	172.339	2.965	163.180	2.366	153.961	3.234
116	111.227	2.269	97.169	1.338	96.755	1.317	65.482	2.996
117	109.298	8.932	76.140	1.774	60.172	799	76.201	1.711
118	94.386	3.337	133.122	10.454	99.781	3.148	74.167	943
161	76.721	2.311	89.310	7.064	63.593	2.514	81.701	4.007
162	6.865	5.126	1.790	2.328	595	3.528	2.886	1.986
163	52.975	5.172	74.464	6.614	64.310	3.813	80.329	9.462
164	2.107	7.820	8.207	1.248	3.605	5.078	4.139	4.701
165	153.347	9.729	99.793	2.550	80.596	1.794	114.079	4.552
166	79.993	9.830	65.529	5.517	8.334	46.272	76.209	8.656
167	1.166	583	994	767	2.168	225	3.062	262
168	4.794	2.377	10.529	4.284	18.050	2.934	14.790	2.755
169	8.025	29.689	13.719	3.006	3.584	7.154	4.901	5.276
16A	4.905	2.563	16.034	5.301	6.532	1.260	10.883	3.086
16B	65.244	6.272	46.599	2.575	40.772	1.315	52.474	3.523
171	49.724	2.557	48.479	4.586	43.007	2.910	54.365	3.158
172	105.964	10.100	89.389	9.167	120.548	13.866	112.850	12.245
16C	131.523	11.618	98.070	4.637	96.401	4.425	80.034	3.111
185	255.107	23.081	254.859	24.204	322.549	43.828	287.681	33.134
181	15.980	22.358	15.391	13.438	160.137	203.658	173.523	276.282
182	68.230	3.902	63.801	4.453	136.374	51.751	148.653	174.331
183	44.567	3.657	58.330	4.997	80.875	98.860	48.169	40.397
184	4.717	3.997	23.963	6.078	4.693	19.117	22.493	25.492
150	21.089	2.463	8.104	2.843	2.125	9.007	19.367	2.278

NUTSIII	$\hat{\mu}_{it}^{LPR1}$	$\hat{\mu}_{it}^{LPR2}$	$\hat{\mu}_{it}^{LPR1}$	$\hat{\mu}_{it}^{LPR2}$	$\hat{\mu}_{it}^{LPR1}$	$\hat{\mu}_{it}^{LPR2}$
	***t*=5**		***t*=6**		***t*=7**	
111	1.286	1.111	3.862	635	1.059	3.721
112	82.678	4.679	77.305	4.144	14.739	14.966
113	68.725	1.961	77.077	982	71.967	1.637
114	119.174	1.369	106.313	1.161	147.003	5.673
115	157.748	2.541	160.473	2.109	119.201	6.396
116	71.741	2.065	75.278	1.928	62.490	3.890
117	65.520	852	57.257	1.154	56.834	976
118	64.806	1.487	102.986	3.573	78.184	625
161	32.986	1.763	58.586	1.998	46.151	1.700
162	1.700	1.831	5.074	2.594	1.414	965
163	58.902	1.807	104.882	15.571	83.119	7.696
164	10.665	1.687	17.278	2.331	8.988	691
165	93.310	4.071	89.645	2.544	141.200	10.658
166	53.866	2.630	80.947	10.117	102.151	18.413
167	748	1.517	1.067	4.283	1.382	543
168	18.364	3.630	32.989	9.780	16.376	2.071
169	3.731	13.175	2.901	15.674	2.070	10.736
16A	26.810	9.714	35.083	16.311	4.379	992
16B	50.909	3.993	27.037	2.208	76.438	9.517
171	26.215	3.111	50.315	2.650	53.420	1.559
172	128.110	23.725	120.303	16.160	118.171	10.608
16C	91.171	3.670	132.945	12.310	108.147	6.169
185	279.464	31.101	254.157	23.198	280.791	30.399
181	33.552	137.087	102.636	11.857	10.023	32.466
182	80.993	14.275	90.746	5.754	93.100	44.901
183	183.566	116.388	89.148	2.454	135.540	51.055
184	8.378	14.895	10.210	2.914	4.293	17.937
150	13.121	3.277	1.863	8.493	8.314	4.044

A22.3 Variância

NUTSIII	$\hat{\mu}_{it}^{LPR1}$	$\hat{\mu}_{it}^{LPR2}$	$\hat{\mu}_{it}^{LPR1}$	$\hat{\mu}_{it}^{LPR2}$	$\hat{\mu}_{it}^{LPR1}$	$\hat{\mu}_{it}^{LPR2}$	$\hat{\mu}_{it}^{LPR1}$	$\hat{\mu}_{it}^{LPR2}$
	***t*=1**		***t*=2**		***t*=3**		***t*=4**	
111	914	1.440	1.060	691	979	1.105	898	980
112	3.448	848	5.886	1.240	3.099	1.417	2.703	1.481
113	5.075	604	4.814	871	4.892	765	4.146	1.474
114	10.021	2.994	7.911	538	8.014	770	8.585	1.228
115	9.252	4.364	10.712	2.608	10.952	2.318	10.024	3.217
116	6.510	826	6.183	784	5.793	764	5.069	1.908
117	4.441	720	4.279	349	3.413	737	4.361	294
118	5.874	2.013	7.258	1.374	6.362	975	4.402	919
161	5.013	1.276	5.392	2.176	4.687	1.488	4.446	1.948
162	1.058	2.958	776	1.767	594	1.620	655	1.627
163	3.788	1.445	4.139	611	3.994	814	4.421	911
164	2.074	2.073	1.418	1.244	2.513	2.720	2.794	2.969
165	7.307	2.612	6.503	1.776	5.396	1.792	5.865	3.029
166	4.662	453	3.540	439	8.153	9.701	4.126	417
167	215	397	246	476	168	225	185	177
168	1.242	1.698	3.878	4.256	2.194	1.273	2.908	2.249
169	3.725	4.948	2.230	2.863	2.454	3.073	2.857	3.272
16A	2.443	2.556	1.702	1.380	1.633	1.141	1.770	1.383
16B	3.581	752	3.113	945	3.207	1.003	3.168	1.029
171	4.770	2.501	8.103	4.280	5.409	2.351	7.250	3.152
172	5.826	1.588	6.951	3.255	6.975	1.656	7.829	2.036
16C	6.980	1.072	5.455	1.602	5.356	1.885	5.460	2.598
185	14.458	2.691	15.332	3.079	16.303	2.656	15.207	2.444
181	4.263	21.418	4.585	13.400	8.506	202.867	10.801	268.422
182	6.364	3.573	6.317	4.453	7.038	14.355	6.348	67.451
183	5.792	3.572	6.198	4.913	6.153	48.080	6.412	34.602
184	2.118	2.045	1.795	2.812	2.209	16.858	1.668	13.397
150	1.345	2.435	1.272	2.822	1.981	8.991	1.592	2.247

NUTSIII	$\hat{\mu}_{it}^{LPR1}$	$\hat{\mu}_{it}^{LPR2}$	$\hat{\mu}_{it}^{LPR1}$	$\hat{\mu}_{it}^{LPR2}$	$\hat{\mu}_{it}^{LPR1}$	$\hat{\mu}_{it}^{LPR2}$
	***t*=5**		***t*=6**		***t*=7**	
111	535	398	632	582	1.047	1.178
112	3.142	1.067	3.285	695	3.224	3.168
113	4.288	1.313	4.536	932	4.428	1.292
114	8.240	1.193	7.241	814	9.125	5.673
115	9.332	2.537	9.085	2.067	9.103	3.758
116	4.723	1.655	4.949	1.775	4.765	2.463
117	3.678	520	3.305	1.150	3.200	974
118	3.888	1.333	5.771	786	4.385	464
161	3.231	1.696	3.739	1.242	3.742	1.663
162	844	1.098	859	1.467	672	933
163	4.121	1.094	5.057	1.124	4.848	888
164	1.473	1.673	1.322	521	1.125	667
165	6.241	1.775	6.657	2.159	9.044	1.957
166	3.427	456	5.315	664	7.096	1.367
167	440	806	1.014	1.618	237	433
168	3.226	2.104	3.644	2.310	1.780	780
169	2.901	5.352	2.850	4.124	1.880	2.970
16A	2.145	1.198	2.764	2.115	848	983
16B	3.085	1.057	3.017	1.994	3.221	926
171	5.155	2.173	5.971	2.522	4.636	1.558
172	9.084	3.620	7.015	2.027	6.574	1.473
16C	5.611	1.760	6.986	1.228	5.816	1.557
185	15.322	2.291	14.777	2.474	14.999	2.134
181	10.702	137.087	6.970	8.067	9.894	31.771
182	5.610	11.772	5.798	5.293	6.042	25.384
183	8.999	40.969	6.853	2.384	7.036	15.173
184	1.687	14.281	1.616	2.315	3.005	7.951
150	1.481	3.267	1.861	8.493	1.495	4.035

A22.4 Erro Absoluto

NUTSIII	$\hat{\mu}_{it}^{LPR1}$	$\hat{\mu}_{it}^{LPR2}$	$\hat{\mu}_{it}^{LPR1}$	$\hat{\mu}_{it}^{LPR2}$	$\hat{\mu}_{it}^{LPR1}$	$\hat{\mu}_{it}^{LPR2}$	$\hat{\mu}_{it}^{LPR1}$	$\hat{\mu}_{it}^{LPR2}$
	***t*=1**		***t*=2**		***t*=3**		***t*=4**	
111	90	34	146	93	145	90	86	34
112	224	23	296	80	177	47	172	51
113	288	20	276	24	286	22	237	49
114	376	48	315	29	331	23	339	28
115	336	71	402	42	390	39	379	47
116	324	41	302	28	302	28	246	47
117	324	91	268	38	238	22	268	38
118	298	43	355	96	306	47	264	25
161	268	39	290	72	243	39	278	52
162	77	59	36	38	19	48	48	34
163	222	62	265	77	246	55	276	92
164	36	77	84	28	49	58	55	55
165	382	88	305	40	274	34	329	55
166	274	97	249	71	74	195	268	91
167	31	19	28	22	45	12	54	13
168	61	38	88	49	126	41	110	41
169	79	160	107	42	45	75	52	64
16A	65	39	121	68	76	30	99	51
16B	248	74	209	42	194	28	222	51
171	212	40	201	54	194	43	217	45
172	316	92	287	80	337	111	324	101
16C	353	103	304	56	302	53	273	41
185	491	143	489	146	553	203	522	175
181	111	115	107	89	391	412	404	507
182	249	51	240	51	360	193	377	329
183	197	49	228	54	273	229	205	147
184	56	54	149	64	57	96	144	114
150	141	40	83	42	35	77	133	38

NUTSIII	$\widehat{\mu}_{it}^{LPR1}$	$\widehat{\mu}_{it}^{LPR2}$	$\widehat{\mu}_{it}^{LPR1}$	$\widehat{\mu}_{it}^{LPR2}$	$\widehat{\mu}_{it}^{LPR1}$	$\widehat{\mu}_{it}^{LPR2}$
	***t*=5**		***t*=6**		***t*=7**	
111	30	29	57	19	23	56
112	282	61	272	59	109	112
113	254	38	269	26	260	34
114	333	30	315	28	371	62
115	385	41	389	36	332	69
116	259	39	265	37	240	53
117	249	23	232	28	232	26
118	247	32	312	56	272	19
161	173	34	234	35	206	32
162	34	35	66	42	31	24
163	234	34	316	120	280	83
164	96	31	126	44	89	21
165	295	52	288	39	364	94
166	225	47	275	97	308	131
167	23	31	28	53	34	19
168	124	54	171	88	121	38
169	42	101	40	115	34	92
16A	157	93	180	122	61	24
16B	219	55	155	40	271	93
171	147	45	211	41	221	32
172	345	142	337	119	334	96
16C	293	47	355	105	320	68
185	514	170	489	144	516	168
181	167	333	309	85	77	149
182	275	92	291	59	295	175
183	418	276	287	39	358	192
184	82	91	93	43	53	112
150	108	45	34	74	83	51

Apêndice 23 – Estimação do EQMP do EBLUP temporal no âmbito do estudo empírico por simulação *model-based*

CÓDIGO SAS

```
/* Considera-se o modelo de Rao-Yu com p=2
(constante + variável explicativa):
 Yit = B1 + B2Xit + Vi + Uit + eit
 Uit = ro*Ui,t-1 + Eit

Verdadeiros coeficientes do modelo:
        B1=1 , B2=2
        -------  V(Vi) = sigma2v = 0.5
        MODELO 1 V(Eit) = sigma2 = 0.25
        -------  ro=0.2

Considera-se:    m=28 pequenos domínios
                 T=7 periodos de tempo

Foram geradas:
K=1000 amostras independentes de Yit
Xit foram gerados a partir de uma distribuição
uniforme no intervalo (0, 1)
Vi foram gerados a partir de uma distribuição
normal: N(0, sigma2v)
eit foram gerados a partir de uma distribuição
normal: N(0, 1)
Eit foram gerados a partir de uma distribuição
normal: N(0, sigma2u) */

Data null;
file 'C:\Documents and Settings\Politriudo\Os
meus documentos\Luís\SAS code\chamamacro.txt';
   do i= 1 to 1000; k=i;
var='%ext_am('||i||','||k||');'; put var; end;
run;

%macro ext_am(i,k);

Data a1;
do i=1 to 28;
do t=1 to 7;
        c=1;
        x=ranuni(&k);
        e=rannor(&k+1);
        Eu=sqrt(0.25)*rannor(&k+2);
        output;
end;
end; run;

Data a2;
do i=1 to 28;
        v=sqrt(0.5)*rannor(&k+3);
        output;
end; run;

Proc iml;
use a2; read all into a2;
        um=J(7,1);
        a3=a2@um;
        create a3 from a3;
        append from a3;
quit;

Data a3;
set a3;
rename col1=i col2=v;
run;

Data a4;
set a1;
keep Eu;
run;

Proc iml;
use a4; read all into a4;
Ident7=I(7);
Ident28=I(28);
I1={0 0 0 0 0 0 0,
    1 0 0 0 0 0 0,
    0 1 0 0 0 0 0,
    0 0 1 0 0 0 0,
    0 0 0 1 0 0 0,
    0 0 0 0 1 0 0,
    0 0 0 0 0 1 0};
I2={0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    1 0 0 0 0 0 0,
    0 1 0 0 0 0 0,
    0 0 1 0 0 0 0,
    0 0 0 1 0 0 0,
    0 0 0 0 1 0 0};
I3={0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    1 0 0 0 0 0 0,
    0 1 0 0 0 0 0,
    0 0 1 0 0 0 0,
    0 0 0 1 0 0 0};
I4={0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    1 0 0 0 0 0 0,
    0 1 0 0 0 0 0,
    0 0 1 0 0 0 0};
I5={0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    1 0 0 0 0 0 0,
    0 1 0 0 0 0 0};
I6={0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    1 0 0 0 0 0 0};
ro=0.2;
Q=ro*I1 + ro*ro*I2 + ro*ro*ro*I3 +
ro*ro*ro*ro*I4 + ro*ro*ro*ro*ro*I5 +
ro*ro*ro*ro*ro*ro*I6 + Ident7;
a5=(Ident28@Q)*a4;
create a5 from a5; append from a5;
quit;

Data a5;
set a5;
rename col1=u;
run;

Data a6;
merge a1 a3 a5;
y=1*c + 2*x + v + u + e;
run;

Data a7;
set a6;
simul=&i;
run;

Proc append base=RaoYu.gerac_RYU_modelo1
data=a7;
run;

/* Fim da geração dos dados */

Data dataro;
ro=0.2;
run;
Data dataro2;
ro=0.2;
ro2= sqrt(1-ro*ro);
keep ro2;
run;
```

```
Proc iml;
use dataro;
read all into dataro;
use dataro2;
read all into dataro2;
I1={1 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I2={0 0 0 0 0 0 0, 0 1 0 0 0 0 0, 0 0 1 0 0 0
0, 0 0 0 1 0 0 0, 0 0 0 0 1 0 0, 0 0 0 0 0 1
0, 0 0 0 0 0 0 1};
I4={0 0 0 0 0 0 0, 1 0 0 0 0 0 0, 0 1 0 0 0 0
0, 0 0 1 0 0 0 0, 0 0 0 1 0 0 0, 0 0 0 0 1 0
0, 0 0 0 0 0 1 0};
P = dataro2*I1 + I2 - dataro*I4;
create P from P;
append from P;
quit;

Proc iml;
use dataro;
read all into dataro;
use dataro2;
read all into dataro2;
C1={1, 0, 0, 0, 0, 0, 0};
C2={0, 1, 1, 1, 1, 1, 1};
f = dataro2*C1 + C2 - dataro*C2;
create f from f;
append from f;
quit;

Proc iml;
use f;
read all into f;
c=f`*f;
create c from c;
append from c;
quit;

Proc iml;
use f;
read all into f;
use c;
read all into c;
D=(1/c)*f*f`;
create D from D;
append from D;
quit;

/* *** */

Data iabh;
set a6;
keep i t c x;
run;

Data ipth;
set a6;
keep i t y;
run;

Data iabhipth1;
set a6;
keep i t y x;
run;

/** Preparação do ficheiro das observações da
variável de interesse **/

Proc sort data=ipth;
by i t; run;

Data tetad1; set ipth; where i= 1; keep y;
Data tetad2; set ipth; where i= 2; keep y;
Data tetad3; set ipth; where i= 3; keep y;
Data tetad4; set ipth; where i= 4; keep y;
Data tetad5; set ipth; where i= 5; keep y;
Data tetad6; set ipth; where i= 6; keep y;
Data tetad7; set ipth; where i= 7; keep y;
Data tetad8; set ipth; where i= 8; keep y;
Data tetad9; set ipth; where i= 9; keep y;
Data tetad10; set ipth; where i= 10; keep y;
Data tetad11; set ipth; where i= 11; keep y;
Data tetad12; set ipth; where i= 12; keep y;
Data tetad13; set ipth; where i= 13; keep y;
Data tetad14; set ipth; where i= 14; keep y;
Data tetad15; set ipth; where i= 15; keep y;
Data tetad16; set ipth; where i= 16; keep y;
Data tetad17; set ipth; where i= 17; keep y;
Data tetad18; set ipth; where i= 18; keep y;
Data tetad19; set ipth; where i= 19; keep y;
Data tetad20; set ipth; where i= 20; keep y;
Data tetad21; set ipth; where i= 21; keep y;
Data tetad22; set ipth; where i= 22; keep y;
Data tetad23; set ipth; where i= 23; keep y;
Data tetad24; set ipth; where i= 24; keep y;
Data tetad25; set ipth; where i= 25; keep y;
Data tetad26; set ipth; where i= 26; keep y;
Data tetad27; set ipth; where i= 27; keep y;
Data tetad28; set ipth; where i= 28; keep y;
run;

/** Preparação do ficheiro das observações da
variável auxiliar **/

Proc sort data=iabh;
by i t; run;

Proc iml;
j=J(196,1);
create j from j; append from j; quit;

Data iabh0;
merge j iabh; run;

Data xd1; set iabh0; where i= 1; keep c x;
Data xd2; set iabh0; where i= 2; keep c x;
Data xd3; set iabh0; where i= 3; keep c x;
Data xd4; set iabh0; where i= 4; keep c x;
Data xd5; set iabh0; where i= 5; keep c x;
Data xd6; set iabh0; where i= 6; keep c x;
Data xd7; set iabh0; where i= 7; keep c x;
Data xd8; set iabh0; where i= 8; keep c x;
Data xd9; set iabh0; where i= 9; keep c x;
Data xd10; set iabh0; where i= 10; keep c x;
Data xd11; set iabh0; where i= 11; keep c x;
Data xd12; set iabh0; where i= 12; keep c x;
Data xd13; set iabh0; where i= 13; keep c x;
Data xd14; set iabh0; where i= 14; keep c x;
Data xd15; set iabh0; where i= 15; keep c x;
Data xd16; set iabh0; where i= 16; keep c x;
Data xd17; set iabh0; where i= 17; keep c x;
Data xd18; set iabh0; where i= 18; keep c x;
Data xd19; set iabh0; where i= 19; keep c x;
Data xd20; set iabh0; where i= 20; keep c x;
Data xd21; set iabh0; where i= 21; keep c x;
Data xd22; set iabh0; where i= 22; keep c x;
Data xd23; set iabh0; where i= 23; keep c x;
Data xd24; set iabh0; where i= 24; keep c x;
Data xd25; set iabh0; where i= 25; keep c x;
Data xd26; set iabh0; where i= 26; keep c x;
Data xd27; set iabh0; where i= 27; keep c x;
Data xd28; set iabh0; where i= 28; keep c x;
run;

/** Cálculo da matriz Z(1) **/

Proc iml;
use tetad1; read all into tetad1;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd1=(Ident - D)*P*tetad1;
create zd1 from zd1; append from zd1; quit;
Proc iml;
use tetad2; read all into tetad2;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd2=(Ident - D)*P*tetad2;
create zd2 from zd2; append from zd2; quit;
Proc iml;
use tetad3; read all into tetad3;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd3=(Ident - D)*P*tetad3;
create zd3 from zd3; append from zd3; quit;
Proc iml;
use tetad4; read all into tetad4;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd4=(Ident - D)*P*tetad4;
create zd4 from zd4; append from zd4; quit;
```

```
Proc iml;
use tetad5; read all into tetad5;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd5=(Ident - D)*P*tetad5;
create zd5 from zd5; append from zd5; quit;
Proc iml;
use tetad6; read all into tetad6;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd6=(Ident - D)*P*tetad6;
create zd6 from zd6; append from zd6; quit;
Proc iml;
use tetad7; read all into tetad7;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd7=(Ident - D)*P*tetad7;
create zd7 from zd7; append from zd7; quit;
Proc iml;
use tetad8; read all into tetad8;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd8=(Ident - D)*P*tetad8;
create zd8 from zd8; append from zd8; quit;
Proc iml;
use tetad9; read all into tetad9;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd9=(Ident - D)*P*tetad9;
create zd9 from zd9; append from zd9; quit;
Proc iml;
use tetad10; read all into tetad10;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd10=(Ident - D)*P*tetad10;
create zd10 from zd10; append from zd10; quit;
Proc iml;
use tetad11; read all into tetad11;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd11=(Ident - D)*P*tetad11;
create zd11 from zd11; append from zd11; quit;
Proc iml;
use tetad12; read all into tetad12;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd12=(Ident - D)*P*tetad12;
create zd12 from zd12; append from zd12; quit;
Proc iml;
use tetad13; read all into tetad13;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd13=(Ident - D)*P*tetad13;
create zd13 from zd13; append from zd13; quit;
Proc iml;
use tetad14; read all into tetad14;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd14=(Ident - D)*P*tetad14;
create zd14 from zd14; append from zd14; quit;
Proc iml;
use tetad15; read all into tetad15;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd15=(Ident - D)*P*tetad15;
create zd15 from zd15; append from zd15; quit;
Proc iml;
use tetad16; read all into tetad16;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd16=(Ident - D)*P*tetad16;
create zd16 from zd16; append from zd16; quit;
Proc iml;
use tetad17; read all into tetad17;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd17=(Ident - D)*P*tetad17;
create zd17 from zd17; append from zd17; quit;
Proc iml;
use tetad18; read all into tetad18;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd18=(Ident - D)*P*tetad18;
create zd18 from zd18; append from zd18; quit;
Proc iml;
use tetad19; read all into tetad19;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd19=(Ident - D)*P*tetad19;
create zd19 from zd19; append from zd19; quit;
Proc iml;
use tetad20; read all into tetad20;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd20=(Ident - D)*P*tetad20;
create zd20 from zd20; append from zd20; quit;
Proc iml;
use tetad21; read all into tetad21;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd21=(Ident - D)*P*tetad21;
create zd21 from zd21; append from zd21; quit;
Proc iml;
use tetad22; read all into tetad22;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd22=(Ident - D)*P*tetad22;
create zd22 from zd22; append from zd22; quit;
Proc iml;
use tetad23; read all into tetad23;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd23=(Ident - D)*P*tetad23;
create zd23 from zd23; append from zd23; quit;
Proc iml;
use tetad24; read all into tetad24;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd24=(Ident - D)*P*tetad24;
create zd24 from zd24; append from zd24; quit;
Proc iml;
use tetad25; read all into tetad25;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd25=(Ident - D)*P*tetad25;
create zd25 from zd25; append from zd25; quit;
Proc iml;
use tetad26; read all into tetad26;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd26=(Ident - D)*P*tetad26;
create zd26 from zd26; append from zd26; quit;
Proc iml;
use tetad27; read all into tetad27;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd27=(Ident - D)*P*tetad27;
create zd27 from zd27; append from zd27; quit;
Proc iml;
use tetad28; read all into tetad28;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd28=(Ident - D)*P*tetad28;
create zd28 from zd28; append from zd28; quit;

Data Z1;
set zd1 zd2 zd3 zd4 zd5 zd6 zd7 zd8 zd9 zd10
zd11 zd12 zd13 zd14 zd15
zd16 zd17 zd18 zd19 zd20 zd21 zd22 zd23 zd24
zd25 zd26 zd27 zd28;
run;
```

```
/** Cálculo da matriz H(1) **/

Proc iml;
use xd1; read all into xd1;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd1=(Ident - D)*P*xd1;
create hd1 from hd1; append from hd1; quit;
Proc iml;
use xd2; read all into xd2;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd2=(Ident - D)*P*xd2;
create hd2 from hd2; append from hd2; quit;
Proc iml;
use xd3; read all into xd3;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd3=(Ident - D)*P*xd3;
create hd3 from hd3; append from hd3; quit;
Proc iml;
use xd4; read all into xd4;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd4=(Ident - D)*P*xd4;
create hd4 from hd4; append from hd4; quit;
Proc iml;
use xd5; read all into xd5;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd5=(Ident - D)*P*xd5;
create hd5 from hd5; append from hd5; quit;
Proc iml;
use xd6; read all into xd6;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd6=(Ident - D)*P*xd6;
create hd6 from hd6; append from hd6; quit;
Proc iml;
use xd7; read all into xd7;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd7=(Ident - D)*P*xd7;
create hd7 from hd7; append from hd7; quit;
Proc iml;
use xd8; read all into xd8;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd8=(Ident - D)*P*xd8;
create hd8 from hd8; append from hd8; quit;
Proc iml;
use xd9; read all into xd9;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd9=(Ident - D)*P*xd9;
create hd9 from hd9; append from hd9; quit;
Proc iml;
use xd10; read all into xd10;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd10=(Ident - D)*P*xd10;
create hd10 from hd10; append from hd10; quit;
Proc iml;
use xd11; read all into xd11;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd11=(Ident - D)*P*xd11;
create hd11 from hd11; append from hd11; quit;
Proc iml;
use xd12; read all into xd12;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd12=(Ident - D)*P*xd12;
create hd12 from hd12; append from hd12; quit;
Proc iml;
use xd13; read all into xd13;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd13=(Ident - D)*P*xd13;
create hd13 from hd13; append from hd13; quit;
Proc iml;
use xd14; read all into xd14;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd14=(Ident - D)*P*xd14;
create hd14 from hd14; append from hd14; quit;
Proc iml;
use xd15; read all into xd15;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd15=(Ident - D)*P*xd15;
create hd15 from hd15; append from hd15; quit;
Proc iml;
use xd16; read all into xd16;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd16=(Ident - D)*P*xd16;
create hd16 from hd16; append from hd16; quit;
Proc iml;
use xd17; read all into xd17;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd17=(Ident - D)*P*xd17;
create hd17 from hd17; append from hd17; quit;
Proc iml;
use xd18; read all into xd18;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd18=(Ident - D)*P*xd18;
create hd18 from hd18; append from hd18; quit;
Proc iml;
use xd19; read all into xd19;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd19=(Ident - D)*P*xd19;
create hd19 from hd19; append from hd19; quit;
Proc iml;
use xd20; read all into xd20;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd20=(Ident - D)*P*xd20;
create hd20 from hd20; append from hd20; quit;
Proc iml;
use xd21; read all into xd21;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd21=(Ident - D)*P*xd21;
create hd21 from hd21; append from hd21; quit;
Proc iml;
use xd22; read all into xd22;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd22=(Ident - D)*P*xd22;
create hd22 from hd22; append from hd22; quit;
Proc iml;
use xd23; read all into xd23;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd23=(Ident - D)*P*xd23;
create hd23 from hd23; append from hd23; quit;
Proc iml;
use xd24; read all into xd24;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd24=(Ident - D)*P*xd24;
create hd24 from hd24; append from hd24; quit;
Proc iml;
use xd25; read all into xd25;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd25=(Ident - D)*P*xd25;
```

```
create hd25 from hd25; append from hd25; quit;
Proc iml;
use xd26; read all into xd26;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd26=(Ident - D)*P*xd26;
create hd26 from hd26; append from hd26; quit;
Proc iml;
use xd27; read all into xd27;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd27=(Ident - D)*P*xd27;
create hd27 from hd27; append from hd27; quit;
Proc iml;
use xd28; read all into xd28;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd28=(Ident - D)*P*xd28;
create hd28 from hd28; append from hd28; quit;

Data H1;
set hd1 hd2 hd3 hd4 hd5 hd6 hd7 hd8 hd9 hd10
hd11 hd12 hd13 hd14 hd15 hd16 hd17 hd18 hd19
hd20 hd21 hd22 hd23 hd24 hd25 hd26 hd27 hd28;
run;

/** Regressão de Z(1) sobre H(1) **/

Data Z1;
set Z1;
rename col1=col0;
run;

Data reg;
merge Z1 H1;
run;

proc reg data=reg noprint;
        model col0=col1 col2 / noint;
        output out=rg1  residual=r1;
run; quit;

Data rg1;
set rg1;
qr1=r1*r1;
Keep qr1;
run;

Proc means data=rg1 sum noprint;
var qr1;
output out=rg1_ sum=sqr1;
run;

Data rg1_;
set rg1_;
keep sqr1;
run;

/** Cálculo da variância sigma2_u **/

Proc iml;
use H1; read all into H1;
Hs1=H1*ginv(t(H1)*H1)*t(H1); /* ginv(A) is the
Moore-Penrose generalized inverse matrix */
create Hs1 from Hs1; append from Hs1; quit; /*
Hs1 é a matriz H*(1) */

Proc iml;
use D; read all into D;
Ident=I(7);
ID=Ident - D; /* Matriz I - D */
create ID from ID; append from ID; quit;

Proc iml;
use ID; read all into ID;
blockID1=BLOCK(ID,ID,ID,ID,ID,ID,ID,ID,ID,ID,I
D,ID,ID,ID);
create blockID1 from blockID1; append from
blockID1; quit;
Proc iml;
use blockID1; read all into blockID1;
blockID=BLOCK(blockID1,blockID1);
create blockID from blockID; append from
blockID; quit;
Proc iml;
use P; read all into P;
P1=P*t(P); /* Admite-se que SIGMA_d = I */
create P1 from P1; append from P1; quit;

Proc iml;
use P1; read all into P1;
blockPP1=BLOCK(P1,P1,P1,P1,P1,P1,P1,P1,P1,P1,P
1,P1,P1,P1);
create blockPP1 from blockPP1; append from
blockPP1; quit;
Proc iml;
use blockPP1; read all into blockPP1;
blockPP=BLOCK(blockPP1,blockPP1);
create blockPP from blockPP; append from
blockPP; quit;

/* Obs: The function 'rank' in SAS IML does
not compute the rank of a matrix */

Proc iml;
use H1; read all into H1;
HH=H1*t(H1);
eval=eigval(HH);
create eval from eval; append from eval; quit;

Data id;
do id=1 to 196;
output; end;
run;

Data eval;
merge id eval;
run;

Data new;
set eval;
by id;
retain count 0;
if first.id then count=0;
if col1 > 0.000001 then count = count + 1;
run;

Proc means data=new sum noprint;
var count;
output out=caracH1 sum=caracH1;  /* Conta o nº
de valores próprios não nulos */
run;

Data caracH1;
set caracH1;
keep caracH1;
run;

Proc iml;
use blockPP; read all into blockPP;
use blockID; read all into blockID;
use Hs1; read all into Hs1;
use rg1_; read all into rg1_;
use caracH1; read all into caracH1;
varu= (rg1_ -  trace((blockID -
Hs1)*blockPP))*(1/(28*6 - caracH1));
create varu from varu; append from varu; quit;

/* Termina aqui o cálculo da variância
sigma2_u */

/** Cálculo da matriz: c^(-0.5)*f'*P*Teta -
var. dependente do modelo transformado **/

Proc iml;
use tetad1; read all into tetad1;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y1=(1/sqrt(c))*t(f)*P*tetad1;
create y1 from y1; append from y1; quit;
Proc iml;
use tetad2; read all into tetad2;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y2=(1/sqrt(c))*t(f)*P*tetad2;
create y2 from y2; append from y2; quit;
Proc iml;
use tetad3; read all into tetad3;
use P; read all into P;
use f; read all into f;
```

```
use c; read all into c;
y3=(1/sqrt(c))*t(f)*P*tetad3;
create y3 from y3; append from y3; quit;
Proc iml;
use tetad4; read all into tetad4;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y4=(1/sqrt(c))*t(f)*P*tetad4;
create y4 from y4; append from y4; quit;
Proc iml;
use tetad5; read all into tetad5;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y5=(1/sqrt(c))*t(f)*P*tetad5;
create y5 from y5; append from y5; quit;
Proc iml;
use tetad6; read all into tetad6;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y6=(1/sqrt(c))*t(f)*P*tetad6;
create y6 from y6; append from y6; quit;
Proc iml;
use tetad7; read all into tetad7;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y7=(1/sqrt(c))*t(f)*P*tetad7;
create y7 from y7; append from y7; quit;
Proc iml;
use tetad8; read all into tetad8;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y8=(1/sqrt(c))*t(f)*P*tetad8;
create y8 from y8; append from y8; quit;
Proc iml;
use tetad9; read all into tetad9;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y9=(1/sqrt(c))*t(f)*P*tetad9;
create y9 from y9; append from y9; quit;
Proc iml;
use tetad10; read all into tetad10;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y10=(1/sqrt(c))*t(f)*P*tetad10;
create y10 from y10; append from y10; quit;
Proc iml;
use tetad11; read all into tetad11;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y11=(1/sqrt(c))*t(f)*P*tetad11;
create y11 from y11; append from y11; quit;
Proc iml;
use tetad12; read all into tetad12;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y12=(1/sqrt(c))*t(f)*P*tetad12;
create y12 from y12; append from y12; quit;
Proc iml;
use tetad13; read all into tetad13;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y13=(1/sqrt(c))*t(f)*P*tetad13;
create y13 from y13; append from y13; quit;
Proc iml;
use tetad14; read all into tetad14;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y14=(1/sqrt(c))*t(f)*P*tetad14;
create y14 from y14; append from y14; quit;
Proc iml;
use tetad15; read all into tetad15;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y15=(1/sqrt(c))*t(f)*P*tetad15;
create y15 from y15; append from y15; quit;
Proc iml;
use tetad16; read all into tetad16;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y16=(1/sqrt(c))*t(f)*P*tetad16;
create y16 from y16; append from y16; quit;
Proc iml;
use tetad17; read all into tetad17;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y17=(1/sqrt(c))*t(f)*P*tetad17;
create y17 from y17; append from y17; quit;
Proc iml;
use tetad18; read all into tetad18;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y18=(1/sqrt(c))*t(f)*P*tetad18;
create y18 from y18; append from y18; quit;
Proc iml;
use tetad19; read all into tetad19;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y19=(1/sqrt(c))*t(f)*P*tetad19;
create y19 from y19; append from y19; quit;
Proc iml;
use tetad20; read all into tetad20;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y20=(1/sqrt(c))*t(f)*P*tetad20;
create y20 from y20; append from y20; quit;
Proc iml;
use tetad21; read all into tetad21;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y21=(1/sqrt(c))*t(f)*P*tetad21;
create y21 from y21; append from y21; quit;
Proc iml;
use tetad22; read all into tetad22;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y22=(1/sqrt(c))*t(f)*P*tetad22;
create y22 from y22; append from y22; quit;
Proc iml;
use tetad23; read all into tetad23;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y23=(1/sqrt(c))*t(f)*P*tetad23;
create y23 from y23; append from y23; quit;
Proc iml;
use tetad24; read all into tetad24;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y24=(1/sqrt(c))*t(f)*P*tetad24;
create y24 from y24; append from y24; quit;
Proc iml;
use tetad25; read all into tetad25;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y25=(1/sqrt(c))*t(f)*P*tetad25;
create y25 from y25; append from y25; quit;
Proc iml;
use tetad26; read all into tetad26;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y26=(1/sqrt(c))*t(f)*P*tetad26;
create y26 from y26; append from y26; quit;
Proc iml;
use tetad27; read all into tetad27;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y27=(1/sqrt(c))*t(f)*P*tetad27;
create y27 from y27; append from y27; quit;
Proc iml;
use tetad28; read all into tetad28;
use P; read all into P;
use f; read all into f;
use c; read all into c;
```

```
y28=(1/sqrt(c))*t(f)*P*tetad28;
create y28 from y28; append from y28; quit;
run;

Data Y;
set y1 y2 y3 y4 y5 y6 y7 y8 y9 y10 y11 y12 y13
y14 y15
y16 y17 y18 y19 y20 y21 y22 y23 y24 y25 y26
y27 y28;
run;

/** Cálculo da matriz: c^(-0.5)*f'*P*X - var.
independentes do modelo transformado **/

Proc iml;
use xd1; read all into xd1;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x1=(1/sqrt(c))*t(f)*P*xd1;
create x1 from x1; append from x1; quit;
Proc iml;
use xd2; read all into xd2;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x2=(1/sqrt(c))*t(f)*P*xd2;
create x2 from x2; append from x2; quit;
Proc iml;
use xd3; read all into xd3;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x3=(1/sqrt(c))*t(f)*P*xd3;
create x3 from x3; append from x3; quit;
Proc iml;
use xd4; read all into xd4;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x4=(1/sqrt(c))*t(f)*P*xd4;
create x4 from x4; append from x4; quit;
Proc iml;
use xd5; read all into xd5;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x5=(1/sqrt(c))*t(f)*P*xd5;
create x5 from x5; append from x5; quit;
Proc iml;
use xd6; read all into xd6;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x6=(1/sqrt(c))*t(f)*P*xd6;
create x6 from x6; append from x6; quit;
Proc iml;
use xd7; read all into xd7;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x7=(1/sqrt(c))*t(f)*P*xd7;
create x7 from x7; append from x7; quit;
Proc iml;
use xd8; read all into xd8;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x8=(1/sqrt(c))*t(f)*P*xd8;
create x8 from x8; append from x8; quit;
Proc iml;
use xd9; read all into xd9;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x9=(1/sqrt(c))*t(f)*P*xd9;
create x9 from x9; append from x9; quit;
Proc iml;
use xd10; read all into xd10;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x10=(1/sqrt(c))*t(f)*P*xd10;
create x10 from x10; append from x10; quit;
Proc iml;
use xd11; read all into xd11;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x11=(1/sqrt(c))*t(f)*P*xd11;
create x11 from x11; append from x11; quit;
Proc iml;
use xd12; read all into xd12;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x12=(1/sqrt(c))*t(f)*P*xd12;
create x12 from x12; append from x12; quit;
Proc iml;
use xd13; read all into xd13;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x13=(1/sqrt(c))*t(f)*P*xd13;
create x13 from x13; append from x13; quit;
Proc iml;
use xd14; read all into xd14;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x14=(1/sqrt(c))*t(f)*P*xd14;
create x14 from x14; append from x14; quit;
Proc iml;
use xd15; read all into xd15;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x15=(1/sqrt(c))*t(f)*P*xd15;
create x15 from x15; append from x15; quit;
Proc iml;
use xd16; read all into xd16;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x16=(1/sqrt(c))*t(f)*P*xd16;
create x16 from x16; append from x16; quit;
Proc iml;
use xd17; read all into xd17;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x17=(1/sqrt(c))*t(f)*P*xd17;
create x17 from x17; append from x17; quit;
Proc iml;
use xd18; read all into xd18;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x18=(1/sqrt(c))*t(f)*P*xd18;
create x18 from x18; append from x18; quit;
Proc iml;
use xd19; read all into xd19;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x19=(1/sqrt(c))*t(f)*P*xd19;
create x19 from x19; append from x19; quit;
Proc iml;
use xd20; read all into xd20;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x20=(1/sqrt(c))*t(f)*P*xd20;
create x20 from x20; append from x20; quit;
Proc iml;
use xd21; read all into xd21;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x21=(1/sqrt(c))*t(f)*P*xd21;
create x21 from x21; append from x21; quit;
Proc iml;
use xd22; read all into xd22;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x22=(1/sqrt(c))*t(f)*P*xd22;
create x22 from x22; append from x22; quit;
Proc iml;
use xd23; read all into xd23;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x23=(1/sqrt(c))*t(f)*P*xd23;
create x23 from x23; append from x23; quit;
Proc iml;
```

```
use xd24; read all into xd24;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x24=(1/sqrt(c))*t(f)*P*xd24;
create x24 from x24; append from x24; quit;
Proc iml;
use xd25; read all into xd25;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x25=(1/sqrt(c))*t(f)*P*xd25;
create x25 from x25; append from x25; quit;
Proc iml;
use xd26; read all into xd26;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x26=(1/sqrt(c))*t(f)*P*xd26;
create x26 from x26; append from x26; quit;
Proc iml;
use xd27; read all into xd27;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x27=(1/sqrt(c))*t(f)*P*xd27;
create x27 from x27; append from x27; quit;
Proc iml;
use xd28; read all into xd28;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x28=(1/sqrt(c))*t(f)*P*xd28;
create x28 from x28; append from x28; quit;
run;

Data X;
set x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13
x14 x15
x16 x17 x18 x19 x20 x21 x22 x23 x24 x25 x26
x27 x28;
run;

/** Regressão de Y sobre X **/

Data Y;
set Y;
rename col1=col0;
run;

Data reg2;
merge Y X;
run;

proc reg data=reg2 noprint;
        model col0=col1 col2 / noint;
        output out=rg2  residual=r2;
run; quit;

Data rg2;
set rg2;
qr2=r2*r2;
Keep qr2;
run;

Proc means data=rg2 sum noprint;
var qr2;
output out=rg2_ sum=sqr2;
run;

Data rg2_;
set rg2_;
keep sqr2;
run;

/** Cálculo da variância sigma2_v **/

Proc iml;
use xd1; read all into xd1;
use P; read all into P;
use f; read all into f;
xpf1=t(xd1)*t(P)*f;
create xpf1 from xpf1; append from xpf1; quit;
Proc iml;
use xd2; read all into xd2;
use P; read all into P;
use f; read all into f;
xpf2=t(xd2)*t(P)*f;
create xpf2 from xpf2; append from xpf2; quit;
Proc iml;
use xd3; read all into xd3;
use P; read all into P;
use f; read all into f;
xpf3=t(xd3)*t(P)*f;
create xpf3 from xpf3; append from xpf3; quit;
Proc iml;
use xd4; read all into xd4;
use P; read all into P;
use f; read all into f;
xpf4=t(xd4)*t(P)*f;
create xpf4 from xpf4; append from xpf4; quit;
Proc iml;
use xd5; read all into xd5;
use P; read all into P;
use f; read all into f;
xpf5=t(xd5)*t(P)*f;
create xpf5 from xpf5; append from xpf5; quit;
Proc iml;
use xd6; read all into xd6;
use P; read all into P;
use f; read all into f;
xpf6=t(xd6)*t(P)*f;
create xpf6 from xpf6; append from xpf6; quit;
Proc iml;
use xd7; read all into xd7;
use P; read all into P;
use f; read all into f;
xpf7=t(xd7)*t(P)*f;
create xpf7 from xpf7; append from xpf7; quit;
Proc iml;
use xd8; read all into xd8;
use P; read all into P;
use f; read all into f;
xpf8=t(xd8)*t(P)*f;
create xpf8 from xpf8; append from xpf8; quit;
Proc iml;
use xd9; read all into xd9;
use P; read all into P;
use f; read all into f;
xpf9=t(xd9)*t(P)*f;
create xpf9 from xpf9; append from xpf9; quit;
Proc iml;
use xd10; read all into xd10;
use P; read all into P;
use f; read all into f;
xpf10=t(xd10)*t(P)*f;
create xpf10 from xpf10; append from xpf10;
quit;
Proc iml;
use xd11; read all into xd11;
use P; read all into P;
use f; read all into f;
xpf11=t(xd11)*t(P)*f;
create xpf11 from xpf11; append from xpf11;
quit;
Proc iml;
use xd12; read all into xd12;
use P; read all into P;
use f; read all into f;
xpf12=t(xd12)*t(P)*f;
create xpf12 from xpf12; append from xpf12;
quit;
Proc iml;
use xd13; read all into xd13;
use P; read all into P;
use f; read all into f;
xpf13=t(xd13)*t(P)*f;
create xpf13 from xpf13; append from xpf13;
quit;
Proc iml;
use xd14; read all into xd14;
use P; read all into P;
use f; read all into f;
xpf14=t(xd14)*t(P)*f;
create xpf14 from xpf14; append from xpf14;
quit;
Proc iml;
use xd15; read all into xd15;
use P; read all into P;
use f; read all into f;
xpf15=t(xd15)*t(P)*f;
create xpf15 from xpf15; append from xpf15;
quit;
Proc iml;
use xd16; read all into xd16;
use P; read all into P;
```

```
use f; read all into f;
xpf16=t(xd16)*t(P)*f;
create xpf16 from xpf16; append from xpf16;
quit;
Proc iml;
use xd17; read all into xd17;
use P; read all into P;
use f; read all into f;
xpf17=t(xd17)*t(P)*f;
create xpf17 from xpf17; append from xpf17;
quit;
Proc iml;
use xd18; read all into xd18;
use P; read all into P;
use f; read all into f;
xpf18=t(xd18)*t(P)*f;
create xpf18 from xpf18; append from xpf18;
quit;
Proc iml;
use xd19; read all into xd19;
use P; read all into P;
use f; read all into f;
xpf19=t(xd19)*t(P)*f;
create xpf19 from xpf19; append from xpf19;
quit;
Proc iml;
use xd20; read all into xd20;
use P; read all into P;
use f; read all into f;
xpf20=t(xd20)*t(P)*f;
create xpf20 from xpf20; append from xpf20;
quit;
Proc iml;
use xd21; read all into xd21;
use P; read all into P;
use f; read all into f;
xpf21=t(xd21)*t(P)*f;
create xpf21 from xpf21; append from xpf21;
quit;
Proc iml;
use xd22; read all into xd22;
use P; read all into P;
use f; read all into f;
xpf22=t(xd22)*t(P)*f;
create xpf22 from xpf22; append from xpf22;
quit;
Proc iml;
use xd23; read all into xd23;
use P; read all into P;
use f; read all into f;
xpf23=t(xd23)*t(P)*f;
create xpf23 from xpf23; append from xpf23;
quit;
Proc iml;
use xd24; read all into xd24;
use P; read all into P;
use f; read all into f;
xpf24=t(xd24)*t(P)*f;
create xpf24 from xpf24; append from xpf24;
quit;
Proc iml;
use xd25; read all into xd25;
use P; read all into P;
use f; read all into f;
xpf25=t(xd25)*t(P)*f;
create xpf25 from xpf25; append from xpf25;
quit;
Proc iml;
use xd26; read all into xd26;
use P; read all into P;
use f; read all into f;
xpf26=t(xd26)*t(P)*f;
create xpf26 from xpf26; append from xpf26;
quit;
Proc iml;
use xd27; read all into xd27;
use P; read all into P;
use f; read all into f;
xpf27=t(xd27)*t(P)*f;
create xpf27 from xpf27; append from xpf27;
quit;
Proc iml;
use xd28; read all into xd28;
use P; read all into P;
use f; read all into f;
xpf28=t(xd28)*t(P)*f;
create xpf28 from xpf28; append from xpf28;
quit;
run;

Proc iml;
use xpf1; read all into xpf1;
use xpf2; read all into xpf2;
use xpf3; read all into xpf3;
use xpf4; read all into xpf4;
use xpf5; read all into xpf5;
use xpf6; read all into xpf6;
use xpf7; read all into xpf7;
use xpf8; read all into xpf8;
use xpf9; read all into xpf9;
use xpf10; read all into xpf10;
use xpf11; read all into xpf11;
use xpf12; read all into xpf12;
use xpf13; read all into xpf13;
use xpf14; read all into xpf14;
use xpf15; read all into xpf15;
use xpf16; read all into xpf16;
use xpf17; read all into xpf17;
use xpf18; read all into xpf18;
use xpf19; read all into xpf19;
use xpf20; read all into xpf20;
use xpf21; read all into xpf21;
use xpf22; read all into xpf22;
use xpf23; read all into xpf23;
use xpf24; read all into xpf24;
use xpf25; read all into xpf25;
use xpf26; read all into xpf26;
use xpf27; read all into xpf27;
use xpf28; read all into xpf28;
Ft=xpf1||xpf2||xpf3||xpf4||xpf5||xpf6||xpf7||x
pf8||xpf9||xpf10||xpf11||xpf12||xpf13||xpf14
||xpf15||xpf16||xpf17||xpf18||xpf19||xpf20||xp
f21||xpf22||xpf23||xpf24||xpf25||xpf26||xpf27||
|xpf28;
create Ft from Ft; append from Ft; quit;

Proc iml;
use Ft; read all into Ft;
F1=t(Ft);
create F1 from F1; append from F1; quit;

Proc iml;
use F1; read all into F1;
Fast=F1*ginv(t(F1)*F1)*t(F1);
create Fast from Fast; append from Fast; quit;
/* Fast é a matriz F* */

Proc iml;
use Fast; read all into Fast;
Ident=I(28);
IF=Ident - Fast; /* Matriz I - F* */
create IF from IF; append from IF; quit;

Proc iml;
use f; read all into f;
use P; read all into P;
use c; read all into c;
d1=t(f)*P*t(P)*f*(1 / c); /* Admite-se que
SIGMA_d = I */
Ident=I(28);
cfp=d1*Ident;
create cfp from cfp; append from cfp; quit;

Proc iml;
use F1; read all into F1;
FF=F1*t(F1);
eval1=eigval(FF);
create eval1 from eval1; append from eval1;
quit;

Data id1;
do id=1 to 28;
output; end;
run;

Data eval1;
merge id1 eval1;
run;

Data new1;
set eval1;
by id;
retain count 0;
if first.id then count=0;
if col1 > 0.000001 then count = count + 1;
run;
```

```
Proc means data=new1 sum noprint;
var count;
output out=caracF sum=caracF;
run;

Data caracF;
set caracF;
keep caracF;  /* Caracteristica de F */
run;

Proc iml;
use IF; read all into IF;
use cfp; read all into cfp;
use c; read all into c;
use rg2_; read all into rg2_;
use caracF; read all into caracF;
use varu; read all into varu;
varv= (rg2_ -  trace(IF*cfp))*(1/(c*(28 -
caracF))) - varu/c;
create varv from varv; append from varv; quit;

/* Termina aqui o cálculo da variância
sigma2_v */

/* Truncagem a zero das variâncias sigma2_u e
sigma2_v */

Data varu;
set varu;
sigma2u= max( col1 , 0);
keep sigma2u;
run;
Data varv;
set varv;
sigma2v= max( col1 , 0);
keep sigma2v;
run;

/* Construção do ficheiro com os parâmetros de
variância e de correlação */

Proc iml;
use dataro; read all into dataro;
use varu; read all into varu;
use varv; read all into varv;
C1={1, 0, 0, 0};
C2={0, 1, 0, 0};
C3={0, 0, 1, 0};
C4={0, 0, 0, 1};
par1 = varv*C1 + (varu/(1 - dataro*dataro))*C2
+ dataro*C3 + C4;
create par1 from par1; append from par1; quit;

Data par;
set par1;
rename col1=est;
run;

/*********************************************
*********************************************/
/***** ESTIMAÇÃO DO MODELO DE RAO-YU ********/
/*********************************************
*********************************************/

Proc sort data=iabhipth1 out= cron;
by i t;
run;

Proc mixed data=cron MMEqSol noinfo noprofile
noclprint;
        class i t;
        model y=x / solution;
        random i / solution G V;
        random t / subject=i type=ar(1);
        parms /parmsdata=par noiter;
make 'solutionf' out=beta;
make 'solutionr' out=u;
make 'Fitstatistics' out=aic;
make 'G' out=Mat_Gaux;
make 'V' out=Mat_V;
run; quit;

Data aic;
set aic;
if descr = 'AIC (smaller is better)';
run;
```

```
DM 'CLEAR OUTPUT';

Data aic;
merge aic;
simul=&i;
run;

proc append base=RaoYu.aic_RYU_modelo1
data=aic;
run;

/*******************************************/
/* CÁLCULO DAS ESTIMATIVAS DO PARÂMETRO DE
INTERESSE *********************************/
/*******************************************/

Data id;
do id=1 to 2;
output; end;
run;

Data beta;
merge id beta;
run;

Data beta0;
set beta;
if id=1;
rename estimate=b0;
keep estimate;
run;

Data beta1;
set beta;
if id=2;
rename estimate=b1;
keep estimate;
run;

Data betan;
merge beta0 beta1;
_type_=0;
run;

/****** Efeitos Aleatórios: v ******/

Data u;
set u;
if estimate='.' then estimate=0;
run;

Data v;
set u;
if Effect='i';
keep i estimate;
run;

Proc sort data=v;
by i;
run;

Proc iml;
use v; read all into v;
one=J(7,1,1);
v1=v@one;
create v1 from v1; append from v1; quit;

Data v1;
set v1;
_type_=0;
rename col2=v;
run;

/****** Efeitos Aleatórios: u ******/

Data u1;
set u;
if Effect='t';
_type_=0;
keep i t estimate _type_;
rename estimate=u;
run;

Proc sort data=u1;
by i t;
run;

Data beta_vu;
```

```
merge betan v1 u1;
by _type_;
run;

/** Estimação do parâmetro de interesse em
cada domínio **/

Proc sort data=cron;
by i t;
run;

Data cron;
merge cron beta_vu;
by i t;
run;

Data crono_resf;
set cron;
eb=b0+x*b1;
e = eb + v + u;
keep i t _type_ x b0 b1 eb u v e;
run;

Data est_RYU;
set crono_resf;
simul=&i;
drop _type_;
run;

DM 'CLEAR OUTPUT';
DM 'CLEAR LOG';

/*********************************************
*********************************************/
/* Estimação do EQMP do EBLUP utilizando a
aproximação analítica ***********************/
/*********************************************
*********************************************/

/*** MATRIZ X ***/

Proc sort data=cron out=cronologico;
by i t;
run;

Data XX1;
set cronologico;
keep x;
run;

Proc iml;
J=J(196,1,1);
create J from J;
append from J;
quit;

Proc iml;
use J;
read all into J;
use XX1;
read all into XX1;
X=insert(J,XX1,0,2);
create X from X;
append from X;
quit;

/*** MATRIZ G ***/

Proc iml;
J7=J(7,1,1);
create J7 from J7;
append from J7;
quit;

Proc iml;
use J7;
read all into J7;
use varv;
read all into varv;
J77=J7*J7`;
BlockMat_Gv=varv*J77;
create BlockMat_Gv from BlockMat_Gv;
append from BlockMat_Gv;
quit;

Proc iml;
use BlockMat_Gv;
read all into BlockMat_Gv;
Mat_Gv1=block (BlockMat_Gv, BlockMat_Gv,
BlockMat_Gv, BlockMat_Gv, BlockMat_Gv,
BlockMat_Gv, BlockMat_Gv,
BlockMat_Gv, BlockMat_Gv, BlockMat_Gv,
BlockMat_Gv, BlockMat_Gv, BlockMat_Gv,
BlockMat_Gv);
create Mat_Gv1 from Mat_Gv1;
append from Mat_Gv1; quit;

Proc iml;
use Mat_Gv1;
read all into Mat_Gv1;
Mat_Gv=block (Mat_Gv1, Mat_Gv1);
create Mat_Gv from Mat_Gv;
append from Mat_Gv; quit;

/***/

Data Mat_Gu;
set Mat_Gaux;
where row>28;
drop row effect i t col1-col28;
run;

/***/

Proc iml;
use Mat_Gv;
read all into Mat_Gv;
use Mat_Gu;
read all into Mat_Gu;
MatG=Mat_Gv+Mat_Gu;
create MatG from MatG;
append from MatG;
quit;

/*** Matriz GAMA e suas colunas ***/

Data ro_1;
set dataro;
ro_1= 1-ro*ro;
keep ro_1;
run;
Data ro_2;
set dataro;
ro_2= ro*ro;
keep ro_2;
run;
Data ro_3;
set dataro;
ro_3= ro*ro*ro;
keep ro_3;
run;
Data ro_4;
set dataro;
ro_4= ro*ro*ro*ro;
keep ro_4;
run;
Data ro_5;
set dataro;
ro_5= ro*ro*ro*ro*ro;
keep ro_5;
run;
Data ro_6;
set dataro;
ro_6= ro*ro*ro*ro*ro*ro;
keep ro_6;
run;

Proc iml;
use ro_1; read all into ro_1;
use dataro; read all into dataro;
use ro_2; read all into ro_2;
use ro_3; read all into ro_3;
use ro_4; read all into ro_4;
use ro_5; read all into ro_5;
use ro_6; read all into ro_6;
I=I(7);
I1={0 1 0 0 0 0 0, 0 0 1 0 0 0 0, 0 0 0 1 0 0
0, 0 0 0 0 1 0 0, 0 0 0 0 0 1 0, 0 0 0 0 0 0
1, 0 0 0 0 0 0 0};
I2={0 0 1 0 0 0 0, 0 0 0 1 0 0 0, 0 0 0 0 1 0
0, 0 0 0 0 0 1 0, 0 0 0 0 0 0 1, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I3={0 0 0 1 0 0 0, 0 0 0 0 1 0 0, 0 0 0 0 0 1
0, 0 0 0 0 0 0 1, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
```

```
I4={0 0 0 0 1 0 0, 0 0 0 0 0 1 0, 0 0 0 0 0 0
1, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I5={0 0 0 0 0 1 0, 0 0 0 0 0 0 1, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I6={0 0 0 0 0 0 1, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
Gama = I/ro_1 + dataro*I1/ro_1 +
t(dataro*I1/ro_1) + ro_2*I2/ro_1 +
t(ro_2*I2/ro_1) + ro_3*I3/ro_1 +
t(ro_3*I3/ro_1) + ro_4*I4/ro_1 +
t(ro_4*I4/ro_1) + ro_5*I5/ro_1 +
t(ro_5*I5/ro_1) + ro_6*I6/ro_1 +
t(ro_6*I6/ro_1);
create Gama from Gama;
append from Gama;
quit;

Proc iml;
use Gama; read all into Gama;
Gama1=Gama[,1];
create Gama1 from Gama1;
append from Gama1;
quit;
Proc iml;
use Gama; read all into Gama;
Gama2=Gama[,2];
create Gama2 from Gama2;
append from Gama2;
quit;
Proc iml;
use Gama; read all into Gama;
Gama3=Gama[,3];
create Gama3 from Gama3;
append from Gama3;
quit;
Proc iml;
use Gama; read all into Gama;
Gama4=Gama[,4];
create Gama4 from Gama4;
append from Gama4;
quit;
Proc iml;
use Gama; read all into Gama;
Gama5=Gama[,5];
create Gama5 from Gama5;
append from Gama5;
quit;
Proc iml;
use Gama; read all into Gama;
Gama6=Gama[,6];
create Gama6 from Gama6;
append from Gama6;
quit;
Proc iml;
use Gama; read all into Gama;
Gama7=Gama[,7];
create Gama7 from Gama7;
append from Gama7;
quit;

/*** MATRIZ V e seus Blocos ***/

Data MatV;
set Mat_v;
drop Index row;
run;

Proc iml;
use MatV; read all into MatV;
MatV1=MatV[1:7,1:7];
create MatV1 from MatV1; append from MatV1;
quit;
Proc iml;
use MatV; read all into MatV;
MatV2=MatV[8:14,8:14];
create MatV2 from MatV2; append from MatV2;
quit;
Proc iml;
use MatV; read all into MatV;
MatV3=MatV[15:21,15:21];
create MatV3 from MatV3; append from MatV3;
quit;
Proc iml;
use MatV; read all into MatV;
MatV4=MatV[22:28,22:28];
create MatV4 from MatV4; append from MatV4;
quit;
Proc iml;
use MatV; read all into MatV;
MatV5=MatV[29:35,29:35];
create MatV5 from MatV5; append from MatV5;
quit;
Proc iml;
use MatV; read all into MatV;
MatV6=MatV[36:42,36:42];
create MatV6 from MatV6; append from MatV6;
quit;
Proc iml;
use MatV; read all into MatV;
MatV7=MatV[43:49,43:49];
create MatV7 from MatV7; append from MatV7;
quit;
Proc iml;
use MatV; read all into MatV;
MatV8=MatV[50:56,50:56];
create MatV8 from MatV8; append from MatV8;
quit;
Proc iml;
use MatV; read all into MatV;
MatV9=MatV[57:63,57:63];
create MatV9 from MatV9; append from MatV9;
quit;
Proc iml;
use MatV; read all into MatV;
MatV10=MatV[64:70,64:70];
create MatV10 from MatV10; append from MatV10;
quit;
Proc iml;
use MatV; read all into MatV;
MatV11=MatV[71:77,71:77];
create MatV11 from MatV11; append from MatV11;
quit;
Proc iml;
use MatV; read all into MatV;
MatV12=MatV[78:84,78:84];
create MatV12 from MatV12; append from MatV12;
quit;
Proc iml;
use MatV; read all into MatV;
MatV13=MatV[85:91,85:91];
create MatV13 from MatV13; append from MatV13;
quit;
Proc iml;
use MatV; read all into MatV;
MatV14=MatV[92:98,92:98];
create MatV14 from MatV14; append from MatV14;
quit;
Proc iml;
use MatV; read all into MatV;
MatV15=MatV[99:105,99:105];
create MatV15 from MatV15; append from MatV15;
quit;
Proc iml;
use MatV; read all into MatV;
MatV16=MatV[106:112,106:112];
create MatV16 from MatV16; append from MatV16;
quit;
Proc iml;
use MatV; read all into MatV;
MatV17=MatV[113:119,113:119];
create MatV17 from MatV17; append from MatV17;
quit;
Proc iml;
use MatV; read all into MatV;
MatV18=MatV[120:126,120:126];
create MatV18 from MatV18; append from MatV18;
quit;
Proc iml;
use MatV; read all into MatV;
MatV19=MatV[127:133,127:133];
create MatV19 from MatV19; append from MatV19;
quit;
Proc iml;
use MatV; read all into MatV;
MatV20=MatV[134:140,134:140];
create MatV20 from MatV20; append from MatV20;
quit;
Proc iml;
use MatV; read all into MatV;
MatV21=MatV[141:147,141:147];
create MatV21 from MatV21; append from MatV21;
quit;
Proc iml;
```

```
use MatV; read all into MatV;
MatV22=MatV[148:154,148:154];
create MatV22 from MatV22; append from MatV22;
quit;
Proc iml;
use MatV; read all into MatV;
MatV23=MatV[155:161,155:161];
create MatV23 from MatV23; append from MatV23;
quit;
Proc iml;
use MatV; read all into MatV;
MatV24=MatV[162:168,162:168];
create MatV24 from MatV24; append from MatV24;
quit;
Proc iml;
use MatV; read all into MatV;
MatV25=MatV[169:175,169:175];
create MatV25 from MatV25; append from MatV25;
quit;
Proc iml;
use MatV; read all into MatV;
MatV26=MatV[176:182,176:182];
create MatV26 from MatV26; append from MatV26;
quit;
Proc iml;
use MatV; read all into MatV;
MatV27=MatV[183:189,183:189];
create MatV27 from MatV27; append from MatV27;
quit;
Proc iml;
use MatV; read all into MatV;
MatV28=MatV[190:196,190:196];
create MatV28 from MatV28; append from MatV28;
quit;

/* *** Fim *** */

/***************************/
/****** Cálculo do g1 ******/
/***************************/

Proc iml;
use MatV; read all into MatV;
use MatG; read all into MatG;
G1_aux=MatG - MatG*inv(MatV)*MatG;
create G1_aux from G1_aux;
append from G1_aux;
quit;

Proc iml;
use G1_aux;
read all into G1_aux;
G1=vecdiag(G1_aux);
create G1 from G1;
append from G1;
quit;

Data g1;
set g1;
rename col1=g1;
run;

/* Verificação de G1 - cálculo de apenas um
valor: g1,13*/
/*
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use dataro; read all into dataro;
use Gama3; read all into Gama3;
use MatV1; read all into MatV1;
J7=J(7,1,1);
G1_13=varv + varu/(1-dataro*dataro) -
t(varv*J7 + varu*Gama3)*inv(MatV1)*(varv*J7 +
varu*Gama3);
create G1_13 from G1_13;
append from G1_13;
quit;
*/
/* Fim da verificação */

/***************************/
/****** Cálculo do g2 ******/
/***************************/

Proc iml;
use X; read all into X;
use MatV; read all into MatV;
use MatG; read all into MatG;
A=I(196);
G2_aux=t(t(X) -
t(X)*inv(MatV)*MatG*A)*inv(t(X)*inv(MatV)*X)*(
t(X) - t(X)*inv(MatV)*MatG*A);
create G2_aux from G2_aux;
append from G2_aux;
quit;

Proc iml;
use G2_aux;
read all into G2_aux;
G2=vecdiag(G2_aux);
create G2 from G2;
append from G2;
quit;

Data g2;
set g2;
rename col1=g2;
run;

/* Verificação de G2 - cálculo de apenas um
valor: g2,13*/
/*
Data lixo;
merge new X;
run;

Data X_13;
set lixo;
if id=3;
drop id count;
run;

Data X_1;
set lixo;
if id<8;
drop id count;
run;

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use dataro; read all into dataro;
use Gama3; read all into Gama3;
use MatV1; read all into MatV1;
use Matv; read all into Matv;
use X; read all into X;
use X_13; read all into X_13;
use X_1; read all into X_1;
J7=J(7,1,1);
G2_13=t(t(X_13)-
t(X_1)*inv(MatV1)*(varv*J7+varu*Gama3))*inv(t(
X)*inv(MatV)*X)*(t(X_13)-
t(X_1)*inv(MatV1)*(varv*J7+varu*Gama3));
create G2_13 from G2_13;
append from G2_13;
quit;*/
/* Fim da verificação */

/***************************/
/****** Cálculo do g3 ******/
/***************************/

/* O Cálculo do g3 tem que ser feito
individualmente para cada trimestre EM CADA
DOMÍNIO. Note-se que o vector coluna "gama"
varia em função do trimeste e que a matriz
"Vi" É DIFERENTE para diferentes domínios.
Basta calcular a matriz A para cada trimestre
de um determinado domínio, e depois replicar
para os restantes domínios, que é sempre igual
*/

/*****************************/
/* Cálculo da matriz A (2*2) */
/*****************************/

/* DOMÍNIO 1 */
/* 1º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama1; read all into Gama1;
use MatV1; read all into MatV1;
```

```
J7=J(7,1,1);
J77=J(7,7,1);
a11=t(Gama1 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama1))*inv(MatV1)*(Gama1 -
Gama*inv(MatV1)*(varv*J7 + varu*Gama1));
create a11 from a11; append from a11; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama1; read all into Gama1;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a22=t(J7 - J77*inv(MatV1)*(varv*J7 +
varu*Gama1))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama1));
create a22 from a22; append from a22; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama1; read all into Gama1;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a21=t(Gama1 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama1))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama1));
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A11_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A11_ from A11_; append from A11_; quit;

/* 2º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama2; read all into Gama2;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a11=t(Gama2 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama2))*inv(MatV1)*(Gama2 -
Gama*inv(MatV1)*(varv*J7 + varu*Gama2));
create a11 from a11; append from a11; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama2; read all into Gama2;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a22=t(J7 - J77*inv(MatV1)*(varv*J7 +
varu*Gama2))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama2));
create a22 from a22; append from a22; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama2; read all into Gama2;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a21=t(Gama2 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama2))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama2));
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A12_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A12_ from A12_; append from A12_; quit;

/* 3º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama3; read all into Gama3;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a11=t(Gama3 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama3))*inv(MatV1)*(Gama3 -
Gama*inv(MatV1)*(varv*J7 + varu*Gama3));
create a11 from a11; append from a11; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama3; read all into Gama3;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a22=t(J7 - J77*inv(MatV1)*(varv*J7 +
varu*Gama3))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama3));
create a22 from a22; append from a22; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama3; read all into Gama3;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a21=t(Gama3 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama3))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama3));
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A13_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A13_ from A13_; append from A13_; quit;

/* 4º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama4; read all into Gama4;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a11=t(Gama4 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama4))*inv(MatV1)*(Gama4 -
Gama*inv(MatV1)*(varv*J7 + varu*Gama4));
create a11 from a11; append from a11; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama4; read all into Gama4;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a22=t(J7 - J77*inv(MatV1)*(varv*J7 +
varu*Gama4))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama4));
create a22 from a22; append from a22; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama4; read all into Gama4;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a21=t(Gama4 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama4))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama4));
```

```
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A14_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A14_ from A14_; append from A14_; quit;

/* 5º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama5; read all into Gama5;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a11=t(Gama5 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama5))*inv(MatV1)*(Gama5 -
Gama*inv(MatV1)*(varv*J7 + varu*Gama5));
create a11 from a11; append from a11; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama5; read all into Gama5;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a22=t(J7 - J77*inv(MatV1)*(varv*J7 +
varu*Gama5))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama5));
create a22 from a22; append from a22; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama5; read all into Gama5;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a21=t(Gama5 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama5))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama5));
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A15_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A15_ from A15_; append from A15_; quit;

/* 6º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama6; read all into Gama6;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a11=t(Gama6 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama6))*inv(MatV1)*(Gama6 -
Gama*inv(MatV1)*(varv*J7 + varu*Gama6));
create a11 from a11; append from a11; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama6; read all into Gama6;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a22=t(J7 - J77*inv(MatV1)*(varv*J7 +
varu*Gama6))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama6));
create a22 from a22; append from a22; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama6; read all into Gama6;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a21=t(Gama6 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama6))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama6));
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A16_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A16_ from A16_; append from A16_; quit;

/* 7º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama7; read all into Gama7;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a11=t(Gama7 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama7))*inv(MatV1)*(Gama7 -
Gama*inv(MatV1)*(varv*J7 + varu*Gama7));
create a11 from a11; append from a11; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama7; read all into Gama7;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a22=t(J7 - J77*inv(MatV1)*(varv*J7 +
varu*Gama7))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama7));
create a22 from a22; append from a22; quit;
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama7; read all into Gama7;
use MatV1; read all into MatV1;
J7=J(7,1,1);
J77=J(7,7,1);
a21=t(Gama7 - Gama*inv(MatV1)*(varv*J7 +
varu*Gama7))*inv(MatV1)*(J7 -
J77*inv(MatV1)*(varv*J7 + varu*Gama7));
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A17_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A17_ from A17_; append from A17_; quit;

/* DOMÍNIO 3 */  (…)
/* DOMÍNIO 4 */  (…)
/* DOMÍNIO 5 */  (…)
/* DOMÍNIO 6 */  (…)
(…)
/* DOMÍNIO 25 */  (…)
/* DOMÍNIO 26 */  (…)
/* DOMÍNIO 27 */  (…)
/* DOMÍNIO 28 */  (…)

/* *** Fim *** */

/**********************************/
/* Cálculo da matriz SIGMA* (2*2) */
/**********************************/
```

```
/* OBS.: Antes de calcular as covariâncias da
matriz SIGMA*, é necessário calcular as
matrizes C, C*, C1 e C2. Note-se que a matriz
V (que depois funciona como a matriz OMEGA já
está calculada, pois é a matriz Cov(Y) e note-
se também que não é necessário calcular a */

/* MatC */

Proc iml;
use D; read all into D;
use P; read all into P;
Ident=I(7);
IDP=(Ident - D)*P; /* Matriz (I - D)*P */
create IDP from IDP; append from IDP; quit;

Proc iml;
use IDP; read all into IDP;
blockIDP1=BLOCK(IDP,IDP,IDP,IDP,IDP,IDP,IDP,ID
P,IDP,IDP,IDP,IDP,IDP,IDP);
create blockIDP1 from blockIDP1; append from
blockIDP1; quit;
Proc iml;
use blockIDP1; read all into blockIDP1;
MatC=BLOCK(blockIDP1,blockIDP1);
create MatC from MatC; append from MatC; quit;

/* MatC* */

Proc iml;
use f; read all into f;
use P; read all into P;
use c; read all into c;
cfP1=(1/sqrt(c))*t(f)*P; /* Matriz c(-
1/2)*fT*P */
create cfP1 from cfP1; append from cfP1; quit;

Proc iml;
use cfP1; read all into cfP1;
blockcfP1=BLOCK(cfP1,cfP1,cfP1,cfP1,cfP1,cfP1,
cfP1,cfP1,cfP1,cfP1,cfP1,cfP1,cfP1,cfP1);
create blockcfP1 from blockcfP1; append from
blockcfP1; quit;
Proc iml;
use blockcfP1; read all into blockcfP1;
MatC_ast=BLOCK(blockcfP1,blockcfP1);
create MatC_ast from MatC_ast; append from
MatC_ast; quit;

/* MatC1 */

Proc iml;
use MatC; read all into MatC;
use X; read all into X;
Ident=I(196);
MatC1=t(MatC)* (Ident - MatC*X*
ginv(t(X)*t(MatC)*MatC*X) *t(X)*t(MatC))
*MatC;
create MatC1 from MatC1; append from MatC1;
quit;

/* MatC2 */

Proc iml;
use MatC_ast; read all into MatC_ast;
use X; read all into X;
Ident=I(28);
MatC2=t(MatC_ast)* (Ident - MatC_ast*X*
ginv(t(X)*t(MatC_ast)*MatC_ast*X)
*t(X)*t(MatC_ast)) *MatC_ast;
create MatC2 from MatC2; append from MatC2;
quit;

/* varu = k1* a'C1a + k2 */

Proc iml;
use caracH1; read all into caracH1;
k1= 1/(28*6 - caracH1);
create k1 from k1; append from k1; quit;

Proc iml;
use blockPP; read all into blockPP;
use blockID; read all into blockID;
use Hs1; read all into Hs1;
use k1; read all into k1;
k2= -trace((blockID - Hs1)*blockPP)*k1;
create k2 from k2; append from k2; quit;
```

```
/* varv = k3* a'C2a + k4* a'C1a + k5 */

Proc iml;
use caracH2; read all into caracH2;
use c; read all into c;
k3= 1/(c*(28 - caracH2));
create k3 from k3; append from k3; quit;

Proc iml;
use k1; read all into k1;
use c; read all into c;
k4= -(1/c)*k1;
create k4 from k4; append from k4; quit;

Proc iml;
use IF; read all into IF;
use cfp; read all into cfp;
use c; read all into c;
use k2; read all into k2;
use k3; read all into k3;
k5= -(1/c)*k2 - trace(IF*cfp)*k3;
create k5 from k5; append from k5; quit;

/* SIGMA11 = Var(varu) = 2* k1^2 * tr(C1 * V *
C1 * V) */
/* SIGMA11 */

Proc iml;
use MatC1; read all into MatC1;
use MatV; read all into MatV;
use k1; read all into k1;
Sigma11_=2*k1*k1* trace(MatC1 * MatV * MatC1 *
MatV);
create Sigma11_ from Sigma11_; append from
Sigma11_; quit;

/* SIGMA22 = Var(varv)= 2*k3^2 * tr(C2*V*C2*V)
+ 4* k3*k4* tr(C1*V*C2*V) + 2* k4^2 *
tr(C1*V*C1*V)*/
/* SIGMA22 */

Proc iml;
use MatC1; read all into MatC1;
use MatC2; read all into MatC2;
use MatV; read all into MatV;
use k3; read all into k3;
use k4; read all into k4;
Sigma22_=2*k3*k3*trace(MatC2 * MatV * MatC2 *
MatV) + 4*k3*k4*trace(MatC1 * MatV * MatC2 *
MatV) +
2*k4*k4*trace(MatC1 * MatV * MatC1 * MatV);
create Sigma22_ from Sigma22_; append from
Sigma22_; quit;

/* SIGMA21 = Cov(varu,varv)= 2*k1*k3 *
tr(C1*V*C2*V) + 2* k1*k4* tr(C1*V*C1*V) */
/* SIGMA21 */

Proc iml;
use MatC1; read all into MatC1;
use MatC2; read all into MatC2;
use MatV; read all into MatV;
use k1; read all into k1;
use k3; read all into k3;
use k4; read all into k4;
Sigma21_=2*k1*k3* trace(MatC1 * MatV * MatC2 *
MatV) + 2*k1*k4* trace(MatC1 * MatV * MatC1 *
MatV);
create Sigma21_ from Sigma21_; append from
Sigma21_; quit;

Proc iml;
use Sigma11_; read all into Sigma11_;
use Sigma22_; read all into Sigma22_;
use Sigma21_; read all into Sigma21_;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
SIGMA=Sigma11_*I1 + Sigma21_*I2 + Sigma21_*I3
+ Sigma22_*I4;
create SIGMA from SIGMA;
append from SIGMA; quit;

/* *** */

/* Cálculo do trace(Ait * SIGMA) */
```

```
Proc iml;
use SIGMA; read all into SIGMA;
use A11_; read all into A11_;
use A12_; read all into A12_;
use A13_; read all into A13_;
use A14_; read all into A14_;
use A15_; read all into A15_;
use A16_; read all into A16_;
use A17_; read all into A17_;
use A21_; read all into A21_;
use A22_; read all into A22_;
use A23_; read all into A23_;
use A24_; read all into A24_;
use A25_; read all into A25_;
use A26_; read all into A26_;
use A27_; read all into A27_;
use A31_; read all into A31_;
use A32_; read all into A32_;
use A33_; read all into A33_;
use A34_; read all into A34_;
use A35_; read all into A35_;
use A36_; read all into A36_;
use A37_; read all into A37_;
use A41_; read all into A41_;
use A42_; read all into A42_;
use A43_; read all into A43_;
use A44_; read all into A44_;
use A45_; read all into A45_;
use A46_; read all into A46_;
use A47_; read all into A47_;
use A51_; read all into A51_;
use A52_; read all into A52_;
use A53_; read all into A53_;
use A54_; read all into A54_;
use A55_; read all into A55_;
use A56_; read all into A56_;
use A57_; read all into A57_;
use A61_; read all into A61_;
use A62_; read all into A62_;
use A63_; read all into A63_;
use A64_; read all into A64_;
use A65_; read all into A65_;
use A66_; read all into A66_;
use A67_; read all into A67_;
use A71_; read all into A71_;
use A72_; read all into A72_;
use A73_; read all into A73_;
use A74_; read all into A74_;
use A75_; read all into A75_;
use A76_; read all into A76_;
use A77_; read all into A77_;
use A81_; read all into A81_;
use A82_; read all into A82_;
use A83_; read all into A83_;
use A84_; read all into A84_;
use A85_; read all into A85_;
use A86_; read all into A86_;
use A87_; read all into A87_;
use A91_; read all into A91_;
use A92_; read all into A92_;
use A93_; read all into A93_;
use A94_; read all into A94_;
use A95_; read all into A95_;
use A96_; read all into A96_;
use A97_; read all into A97_;
use A101_; read all into A101_;
use A102_; read all into A102_;
use A103_; read all into A103_;
use A104_; read all into A104_;
use A105_; read all into A105_;
use A106_; read all into A106_;
use A107_; read all into A107_;
G3_11=trace(A11_ * SIGMA);
G3_12=trace(A12_ * SIGMA);
G3_13=trace(A13_ * SIGMA);
G3_14=trace(A14_ * SIGMA);
G3_15=trace(A15_ * SIGMA);
G3_16=trace(A16_ * SIGMA);
G3_17=trace(A17_ * SIGMA);
G3_21=trace(A21_ * SIGMA);
G3_22=trace(A22_ * SIGMA);
G3_23=trace(A23_ * SIGMA);
G3_24=trace(A24_ * SIGMA);
G3_25=trace(A25_ * SIGMA);
G3_26=trace(A26_ * SIGMA);
G3_27=trace(A27_ * SIGMA);
G3_31=trace(A31_ * SIGMA);
G3_32=trace(A32_ * SIGMA);
G3_33=trace(A33_ * SIGMA);
G3_34=trace(A34_ * SIGMA);
G3_35=trace(A35_ * SIGMA);
G3_36=trace(A36_ * SIGMA);
G3_37=trace(A37_ * SIGMA);
G3_41=trace(A41_ * SIGMA);
G3_42=trace(A42_ * SIGMA);
G3_43=trace(A43_ * SIGMA);
G3_44=trace(A44_ * SIGMA);
G3_45=trace(A45_ * SIGMA);
G3_46=trace(A46_ * SIGMA);
G3_47=trace(A47_ * SIGMA);
G3_51=trace(A51_ * SIGMA);
G3_52=trace(A52_ * SIGMA);
G3_53=trace(A53_ * SIGMA);
G3_54=trace(A54_ * SIGMA);
G3_55=trace(A55_ * SIGMA);
G3_56=trace(A56_ * SIGMA);
G3_57=trace(A57_ * SIGMA);
G3_61=trace(A61_ * SIGMA);
G3_62=trace(A62_ * SIGMA);
G3_63=trace(A63_ * SIGMA);
G3_64=trace(A64_ * SIGMA);
G3_65=trace(A65_ * SIGMA);
G3_66=trace(A66_ * SIGMA);
G3_67=trace(A67_ * SIGMA);
G3_71=trace(A71_ * SIGMA);
G3_72=trace(A72_ * SIGMA);
G3_73=trace(A73_ * SIGMA);
G3_74=trace(A74_ * SIGMA);
G3_75=trace(A75_ * SIGMA);
G3_76=trace(A76_ * SIGMA);
G3_77=trace(A77_ * SIGMA);
G3_81=trace(A81_ * SIGMA);
G3_82=trace(A82_ * SIGMA);
G3_83=trace(A83_ * SIGMA);
G3_84=trace(A84_ * SIGMA);
G3_85=trace(A85_ * SIGMA);
G3_86=trace(A86_ * SIGMA);
G3_87=trace(A87_ * SIGMA);
G3_91=trace(A91_ * SIGMA);
G3_92=trace(A92_ * SIGMA);
G3_93=trace(A93_ * SIGMA);
G3_94=trace(A94_ * SIGMA);
G3_95=trace(A95_ * SIGMA);
G3_96=trace(A96_ * SIGMA);
G3_97=trace(A97_ * SIGMA);
G3_101=trace(A101_ * SIGMA);
G3_102=trace(A102_ * SIGMA);
G3_103=trace(A103_ * SIGMA);
G3_104=trace(A104_ * SIGMA);
G3_105=trace(A105_ * SIGMA);
G3_106=trace(A106_ * SIGMA);
G3_107=trace(A107_ * SIGMA);
G3_aux1=G3_11//G3_12//G3_13//G3_14//G3_15//G3_
16//G3_17//G3_21//G3_22//G3_23//G3_24//G3_25//
G3_26//G3_27//G3_31//G3_32//G3_33//G3_34//G3_3
5//G3_36//G3_37//G3_41//G3_42//G3_43//G3_44//G
3_45//G3_46//G3_47//G3_51//G3_52//G3_53//G3_54
//G3_55//G3_56//G3_57//G3_61//G3_62//G3_63//G3
_64//G3_65//G3_66//G3_67//G3_71//G3_72//G3_73/
/G3_74//G3_75//G3_76//G3_77//G3_81//G3_82//G3_
83//G3_84//G3_85//G3_86//G3_87//G3_91//G3_92//
G3_93//G3_94//G3_95//G3_96//G3_97//G3_101//G3_
102//G3_103//G3_104//G3_105//G3_106//G3_107;
create G3_aux1 from G3_aux1; append from
G3_aux1; quit;

Proc iml;
use SIGMA; read all into SIGMA;
use A111_; read all into A111_;
use A112_; read all into A112_;
use A113_; read all into A113_;
use A114_; read all into A114_;
use A115_; read all into A115_;
use A116_; read all into A116_;
use A117_; read all into A117_;
use A121_; read all into A121_;
use A122_; read all into A122_;
use A123_; read all into A123_;
use A124_; read all into A124_;
use A125_; read all into A125_;
use A126_; read all into A126_;
use A127_; read all into A127_;
use A131_; read all into A131_;
use A132_; read all into A132_;
use A133_; read all into A133_;
use A134_; read all into A134_;
```

```
use A135_; read all into A135_;
use A136_; read all into A136_;
use A137_; read all into A137_;
use A141_; read all into A141_;
use A142_; read all into A142_;
use A143_; read all into A143_;
use A144_; read all into A144_;
use A145_; read all into A145_;
use A146_; read all into A146_;
use A147_; read all into A147_;
use A151_; read all into A151_;
use A152_; read all into A152_;
use A153_; read all into A153_;
use A154_; read all into A154_;
use A155_; read all into A155_;
use A156_; read all into A156_;
use A157_; read all into A157_;
use A161_; read all into A161_;
use A162_; read all into A162_;
use A163_; read all into A163_;
use A164_; read all into A164_;
use A165_; read all into A165_;
use A166_; read all into A166_;
use A167_; read all into A167_;
use A171_; read all into A171_;
use A172_; read all into A172_;
use A173_; read all into A173_;
use A174_; read all into A174_;
use A175_; read all into A175_;
use A176_; read all into A176_;
use A177_; read all into A177_;
use A181_; read all into A181_;
use A182_; read all into A182_;
use A183_; read all into A183_;
use A184_; read all into A184_;
use A185_; read all into A185_;
use A186_; read all into A186_;
use A187_; read all into A187_;
use A191_; read all into A191_;
use A192_; read all into A192_;
use A193_; read all into A193_;
use A194_; read all into A194_;
use A195_; read all into A195_;
use A196_; read all into A196_;
use A197_; read all into A197_;
use A201_; read all into A201_;
use A202_; read all into A202_;
use A203_; read all into A203_;
use A204_; read all into A204_;
use A205_; read all into A205_;
use A206_; read all into A206_;
use A207_; read all into A207_;
G3_111=trace(A111_ * SIGMA);
G3_112=trace(A112_ * SIGMA);
G3_113=trace(A113_ * SIGMA);
G3_114=trace(A114_ * SIGMA);
G3_115=trace(A115_ * SIGMA);
G3_116=trace(A116_ * SIGMA);
G3_117=trace(A117_ * SIGMA);
G3_121=trace(A121_ * SIGMA);
G3_122=trace(A122_ * SIGMA);
G3_123=trace(A123_ * SIGMA);
G3_124=trace(A124_ * SIGMA);
G3_125=trace(A125_ * SIGMA);
G3_126=trace(A126_ * SIGMA);
G3_127=trace(A127_ * SIGMA);
G3_131=trace(A131_ * SIGMA);
G3_132=trace(A132_ * SIGMA);
G3_133=trace(A133_ * SIGMA);
G3_134=trace(A134_ * SIGMA);
G3_135=trace(A135_ * SIGMA);
G3_136=trace(A136_ * SIGMA);
G3_137=trace(A137_ * SIGMA);
G3_141=trace(A141_ * SIGMA);
G3_142=trace(A142_ * SIGMA);
G3_143=trace(A143_ * SIGMA);
G3_144=trace(A144_ * SIGMA);
G3_145=trace(A145_ * SIGMA);
G3_146=trace(A146_ * SIGMA);
G3_147=trace(A147_ * SIGMA);
G3_151=trace(A151_ * SIGMA);
G3_152=trace(A152_ * SIGMA);
G3_153=trace(A153_ * SIGMA);
G3_154=trace(A154_ * SIGMA);
G3_155=trace(A155_ * SIGMA);
G3_156=trace(A156_ * SIGMA);
G3_157=trace(A157_ * SIGMA);
G3_161=trace(A161_ * SIGMA);
G3_162=trace(A162_ * SIGMA);
G3_163=trace(A163_ * SIGMA);
G3_164=trace(A164_ * SIGMA);
G3_165=trace(A165_ * SIGMA);
G3_166=trace(A166_ * SIGMA);
G3_167=trace(A167_ * SIGMA);
G3_171=trace(A171_ * SIGMA);
G3_172=trace(A172_ * SIGMA);
G3_173=trace(A173_ * SIGMA);
G3_174=trace(A174_ * SIGMA);
G3_175=trace(A175_ * SIGMA);
G3_176=trace(A176_ * SIGMA);
G3_177=trace(A177_ * SIGMA);
G3_181=trace(A181_ * SIGMA);
G3_182=trace(A182_ * SIGMA);
G3_183=trace(A183_ * SIGMA);
G3_184=trace(A184_ * SIGMA);
G3_185=trace(A185_ * SIGMA);
G3_186=trace(A186_ * SIGMA);
G3_187=trace(A187_ * SIGMA);
G3_191=trace(A191_ * SIGMA);
G3_192=trace(A192_ * SIGMA);
G3_193=trace(A193_ * SIGMA);
G3_194=trace(A194_ * SIGMA);
G3_195=trace(A195_ * SIGMA);
G3_196=trace(A196_ * SIGMA);
G3_197=trace(A197_ * SIGMA);
G3_201=trace(A201_ * SIGMA);
G3_202=trace(A202_ * SIGMA);
G3_203=trace(A203_ * SIGMA);
G3_204=trace(A204_ * SIGMA);
G3_205=trace(A205_ * SIGMA);
G3_206=trace(A206_ * SIGMA);
G3_207=trace(A207_ * SIGMA);
G3_aux2=G3_111//G3_112//G3_113//G3_114//G3_115
//G3_116//G3_117//G3_121//G3_122//G3_123//G3_1
24//G3_125//G3_126//G3_127//G3_131//G3_132//G3
_133//G3_134//G3_135//G3_136//G3_137//G3_141//
G3_142//G3_143//G3_144//G3_145//G3_146//G3_147
//G3_151//G3_152//G3_153//G3_154//G3_155//G3_1
56//G3_157//G3_161//G3_162//G3_163//G3_164//G3
_165//G3_166//G3_167//G3_171//G3_172//G3_173//
G3_174//G3_175//G3_176//G3_177//G3_181//G3_182
//G3_183//G3_184//G3_185//G3_186//G3_187//G3_1
91//G3_192//G3_193//G3_194//G3_195//G3_196//G3
_197//G3_201//G3_202//G3_203//G3_204//G3_205//
G3_206//G3_207;
create G3_aux2 from G3_aux2; append from
G3_aux2; quit;

Proc iml;
use SIGMA; read all into SIGMA;
use A211_; read all into A211_;
use A212_; read all into A212_;
use A213_; read all into A213_;
use A214_; read all into A214_;
use A215_; read all into A215_;
use A216_; read all into A216_;
use A217_; read all into A217_;
use A221_; read all into A221_;
use A222_; read all into A222_;
use A223_; read all into A223_;
use A224_; read all into A224_;
use A225_; read all into A225_;
use A226_; read all into A226_;
use A227_; read all into A227_;
use A231_; read all into A231_;
use A232_; read all into A232_;
use A233_; read all into A233_;
use A234_; read all into A234_;
use A235_; read all into A235_;
use A236_; read all into A236_;
use A237_; read all into A237_;
use A241_; read all into A241_;
use A242_; read all into A242_;
use A243_; read all into A243_;
use A244_; read all into A244_;
use A245_; read all into A245_;
use A246_; read all into A246_;
use A247_; read all into A247_;
use A251_; read all into A251_;
use A252_; read all into A252_;
use A253_; read all into A253_;
use A254_; read all into A254_;
use A255_; read all into A255_;
use A256_; read all into A256_;
use A257_; read all into A257_;
use A261_; read all into A261_;
```

```
use A262_; read all into A262_;
use A263_; read all into A263_;
use A264_; read all into A264_;
use A265_; read all into A265_;
use A266_; read all into A266_;
use A267_; read all into A267_;
use A271_; read all into A271_;
use A272_; read all into A272_;
use A273_; read all into A273_;
use A274_; read all into A274_;
use A275_; read all into A275_;
use A276_; read all into A276_;
use A277_; read all into A277_;
use A281_; read all into A281_;
use A282_; read all into A282_;
use A283_; read all into A283_;
use A284_; read all into A284_;
use A285_; read all into A285_;
use A286_; read all into A286_;
use A287_; read all into A287_;
G3_211=trace(A211_ * SIGMA);
G3_212=trace(A212_ * SIGMA);
G3_213=trace(A213_ * SIGMA);
G3_214=trace(A214_ * SIGMA);
G3_215=trace(A215_ * SIGMA);
G3_216=trace(A216_ * SIGMA);
G3_217=trace(A217_ * SIGMA);
G3_221=trace(A221_ * SIGMA);
G3_222=trace(A222_ * SIGMA);
G3_223=trace(A223_ * SIGMA);
G3_224=trace(A224_ * SIGMA);
G3_225=trace(A225_ * SIGMA);
G3_226=trace(A226_ * SIGMA);
G3_227=trace(A227_ * SIGMA);
G3_231=trace(A231_ * SIGMA);
G3_232=trace(A232_ * SIGMA);
G3_233=trace(A233_ * SIGMA);
G3_234=trace(A234_ * SIGMA);
G3_235=trace(A235_ * SIGMA);
G3_236=trace(A236_ * SIGMA);
G3_237=trace(A237_ * SIGMA);
G3_241=trace(A241_ * SIGMA);
G3_242=trace(A242_ * SIGMA);
G3_243=trace(A243_ * SIGMA);
G3_244=trace(A244_ * SIGMA);
G3_245=trace(A245_ * SIGMA);
G3_246=trace(A246_ * SIGMA);
G3_247=trace(A247_ * SIGMA);
G3_251=trace(A251_ * SIGMA);
G3_252=trace(A252_ * SIGMA);
G3_253=trace(A253_ * SIGMA);
G3_254=trace(A254_ * SIGMA);
G3_255=trace(A255_ * SIGMA);
G3_256=trace(A256_ * SIGMA);
G3_257=trace(A257_ * SIGMA);
G3_261=trace(A261_ * SIGMA);
G3_262=trace(A262_ * SIGMA);
G3_263=trace(A263_ * SIGMA);
G3_264=trace(A264_ * SIGMA);
G3_265=trace(A265_ * SIGMA);
G3_266=trace(A266_ * SIGMA);
G3_267=trace(A267_ * SIGMA);
G3_271=trace(A271_ * SIGMA);
G3_272=trace(A272_ * SIGMA);
G3_273=trace(A273_ * SIGMA);
G3_274=trace(A274_ * SIGMA);
G3_275=trace(A275_ * SIGMA);
G3_276=trace(A276_ * SIGMA);
G3_277=trace(A277_ * SIGMA);
G3_281=trace(A281_ * SIGMA);
G3_282=trace(A282_ * SIGMA);
G3_283=trace(A283_ * SIGMA);
G3_284=trace(A284_ * SIGMA);
G3_285=trace(A285_ * SIGMA);
G3_286=trace(A286_ * SIGMA);
G3_287=trace(A287_ * SIGMA);
G3_aux3=G3_211//G3_212//G3_213//G3_214//G3_215
//G3_216//G3_217//G3_221//G3_222//G3_223//G3_2
24//G3_225//G3_226//G3_227//G3_231//G3_232//G3
_233//G3_234//G3_235//G3_236//G3_237//G3_241//
G3_242//G3_243//G3_244//G3_245//G3_246//G3_247
//G3_251//G3_252//G3_253//G3_254//G3_255//G3_2
56//G3_257//G3_261//G3_262//G3_263//G3_264//G3
_265//G3_266//G3_267//G3_271//G3_272//G3_273//
G3_274//G3_275//G3_276//G3_277//G3_281//G3_282
//G3_283//G3_284//G3_285//G3_286//G3_287;
create G3_aux3 from G3_aux3; append from
G3_aux3; quit;
```

```
Data G3_aux;
set G3_aux1 G3_aux2 G3_aux3;
run;

Data g3;
set g3_aux;
rename col1=g3; run;

/* *** Fim *** */

/** Estimativas analíticas do EQM do EBLUP **/

Proc iml;
use G1; read all into G1;
use G2; read all into G2;
use G3; read all into G3;
EQM_EBLUP= G1 + G2 + 2*G3;
create EQM_EBLUP from EQM_EBLUP; append from
EQM_EBLUP; quit;

Data EQM_EBLUP;
set EQM_EBLUP;
rename col1=EQM_EBLUP;
run;

Data Est_RYU1;
merge Est_RYU G1 G2 G3 EQM_EBLUP;
EQM_Naive=G1 + G2;
run;

Proc append base=RaoYu.est_RYU_modelo1
data=est_RYU1;
run;

%mend;
%inc 'C:\Documents and Settings\Politriudo\Os
meus documentos\Luís\SAS code\chamamacro.txt';

/*********************************************
**********************************************/
/* Estimação do EQMP do EBLUP utilizando a
aproximação Bootstrap */
/* Código SAS utilizado depois de estimado o
modelo de Rao-Yu para cada conjunto de dados*/
/*********************************************
**********************************************/

/*********************************************/
/* CÁLCULO DOS EFEITOS FIXOS - QUE SE ADMITE
SEREM OS VERDADEIROS */
/*********************************************/

Data id;
do id=1 to 2;
output; end;
run;

Data beta;
merge id beta;
run;

Data beta0;
set beta;
if id=1;
rename estimate=b0;
keep estimate;
run;

Data beta1;
set beta;
if id=2;
rename estimate=b1;
keep estimate;
run;

Data betan_base;
merge beta0 beta1;
run;

/*********************************************
**********************************************/
/* GERAÇÃO DE DADOS BOOTSTRAP COM BASE NOS
PARÂMETROS ESTIMADOS */
/*********************************************
**********************************************/

Data null;
```

```
file 'C:\Documents and Settings\Politriudo\Os
meus documentos\Luís\SAS code\chamamacro.txt';
   do i= 1 to 250; k=i;
var='%ext_am('||i||','||k||');'; put var; end;
run;

%macro ext_am(i,k);

Data a1;
set varu;
do i=1 to 28;
        do t=1 to 7;
        e=rannor(&k+10000);
        Eu=sqrt(sigma2u)*rannor(&k+20000);
        output;
        end;
end; drop sigma2u; run;

Data a2;
set varv;
do i=1 to 28;
        v=sqrt(sigma2v)*rannor(&k+30000);
        output;
end; drop sigma2v; run;

Proc iml;
use a2; read all into a2;
        um=J(7,1);
        a3=a2@um;
        create a3 from a3;
        append from a3;
quit;

Data a3;
set a3;
rename col1=i col2=v;
run;

Data a4;
set a1;
keep Eu;
run;

Proc iml;
use a4; read all into a4;
Ident7=I(7);
Ident28=I(28);
I1={0 0 0 0 0 0 0,
    1 0 0 0 0 0 0,
    0 1 0 0 0 0 0,
    0 0 1 0 0 0 0,
    0 0 0 1 0 0 0,
    0 0 0 0 1 0 0,
    0 0 0 0 0 1 0};
I2={0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    1 0 0 0 0 0 0,
    0 1 0 0 0 0 0,
    0 0 1 0 0 0 0,
    0 0 0 1 0 0 0,
    0 0 0 0 1 0 0};
I3={0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    1 0 0 0 0 0 0,
    0 1 0 0 0 0 0,
    0 0 1 0 0 0 0,
    0 0 0 1 0 0 0};
I4={0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    1 0 0 0 0 0 0,
    0 1 0 0 0 0 0,
    0 0 1 0 0 0 0};
I5={0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    1 0 0 0 0 0 0,
    0 1 0 0 0 0 0};
I6={0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
```

```
    1 0 0 0 0 0 0};
ro=0.2;
Q=ro*I1 + ro*ro*I2 + ro*ro*ro*I3 +
ro*ro*ro*ro*I4 + ro*ro*ro*ro*ro*I5 +
ro*ro*ro*ro*ro*ro*I6 + Ident7;
a5=(Ident28@Q)*a4;
create a5 from a5; append from a5;
quit;

Data a5;
set a5;
rename col1=u;
run;

Proc iml;
use MatX; read all into MatX;
use Betan_base; read all into Betan_base;
a7=MatX*t(Betan_base);
create a7 from a7; append from a7;
quit;

Data a7;
set a7;
rename col1=XB;
run;

Data a6;
merge a1 a3 a5 a7 MatX;
y=XB + v + u + e;
run;

Data a8;
set a6;
boot=&i;
run;

/********************************************/
/** Fim da geração dos dados no bootstrap ***/
/********************************************/

Data dataro;
ro=0.2;
run;
Data dataro2;
ro=0.2;
ro2= sqrt(1-ro*ro);
keep ro2;
run;

Proc iml;
use dataro;
read all into dataro;
use dataro2;
read all into dataro2;
I1={1 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I2={0 0 0 0 0 0 0, 0 1 0 0 0 0 0, 0 0 1 0 0 0
0, 0 0 0 1 0 0 0, 0 0 0 0 1 0 0, 0 0 0 0 0 1
0, 0 0 0 0 0 0 1};
I4={0 0 0 0 0 0 0, 1 0 0 0 0 0 0, 0 1 0 0 0 0
0, 0 0 1 0 0 0 0, 0 0 0 1 0 0 0, 0 0 0 0 1 0
0, 0 0 0 0 0 1 0};
P = dataro2*I1 + I2 - dataro*I4;
create P from P;
append from P;
quit;

Proc iml;
use dataro;
read all into dataro;
use dataro2;
read all into dataro2;
C1={1, 0, 0, 0, 0, 0, 0};
C2={0, 1, 1, 1, 1, 1, 1};
f = dataro2*C1 + C2 - dataro*C2;
create f from f;
append from f;
quit;

Proc iml;
use f;
read all into f;
c=f`*f;
create c from c;
append from c;
quit;
```

```
Proc iml;
use f;
read all into f;
use c;
read all into c;
D=(1/c)*f*f`;
create D from D;
append from D;
quit;

/* *** */

Data iabh;
set a6;
keep i t c x;
run;

Data ipth;
set a6;
keep i t y;
run;

Data iabhipth1;
set a6;
keep i t y x;
run;

/** Preparação do ficheiro das observações da
variável de interesse **/

Proc sort data=ipth;
by i t; run;

Data tetad1; set ipth; where i= 1; keep y;
Data tetad2; set ipth; where i= 2; keep y;
Data tetad3; set ipth; where i= 3; keep y;
Data tetad4; set ipth; where i= 4; keep y;
Data tetad5; set ipth; where i= 5; keep y;
Data tetad6; set ipth; where i= 6; keep y;
Data tetad7; set ipth; where i= 7; keep y;
Data tetad8; set ipth; where i= 8; keep y;
Data tetad9; set ipth; where i= 9; keep y;
Data tetad10; set ipth; where i= 10; keep y;
Data tetad11; set ipth; where i= 11; keep y;
Data tetad12; set ipth; where i= 12; keep y;
Data tetad13; set ipth; where i= 13; keep y;
Data tetad14; set ipth; where i= 14; keep y;
Data tetad15; set ipth; where i= 15; keep y;
Data tetad16; set ipth; where i= 16; keep y;
Data tetad17; set ipth; where i= 17; keep y;
Data tetad18; set ipth; where i= 18; keep y;
Data tetad19; set ipth; where i= 19; keep y;
Data tetad20; set ipth; where i= 20; keep y;
Data tetad21; set ipth; where i= 21; keep y;
Data tetad22; set ipth; where i= 22; keep y;
Data tetad23; set ipth; where i= 23; keep y;
Data tetad24; set ipth; where i= 24; keep y;
Data tetad25; set ipth; where i= 25; keep y;
Data tetad26; set ipth; where i= 26; keep y;
Data tetad27; set ipth; where i= 27; keep y;
Data tetad28; set ipth; where i= 28; keep y;
run;
/** Preparação do ficheiro das observações da
variável auxiliar **/

Proc sort data=iabh;
by i t; run;

Proc iml;
j=J(196,1);
create j from j; append from j; quit;

Data iabh0;
merge j iabh;
run;

Data xd1; set iabh0; where i= 1; keep c x;
Data xd2; set iabh0; where i= 2; keep c x;
Data xd3; set iabh0; where i= 3; keep c x;
Data xd4; set iabh0; where i= 4; keep c x;
Data xd5; set iabh0; where i= 5; keep c x;
Data xd6; set iabh0; where i= 6; keep c x;
Data xd7; set iabh0; where i= 7; keep c x;
Data xd8; set iabh0; where i= 8; keep c x;
Data xd9; set iabh0; where i= 9; keep c x;
Data xd10; set iabh0; where i= 10; keep c x;
Data xd11; set iabh0; where i= 11; keep c x;
Data xd12; set iabh0; where i= 12; keep c x;
Data xd13; set iabh0; where i= 13; keep c x;
Data xd14; set iabh0; where i= 14; keep c x;
Data xd15; set iabh0; where i= 15; keep c x;
Data xd16; set iabh0; where i= 16; keep c x;
Data xd17; set iabh0; where i= 17; keep c x;
Data xd18; set iabh0; where i= 18; keep c x;
Data xd19; set iabh0; where i= 19; keep c x;
Data xd20; set iabh0; where i= 20; keep c x;
Data xd21; set iabh0; where i= 21; keep c x;
Data xd22; set iabh0; where i= 22; keep c x;
Data xd23; set iabh0; where i= 23; keep c x;
Data xd24; set iabh0; where i= 24; keep c x;
Data xd25; set iabh0; where i= 25; keep c x;
Data xd26; set iabh0; where i= 26; keep c x;
Data xd27; set iabh0; where i= 27; keep c x;
Data xd28; set iabh0; where i= 28; keep c x;
run;

/** Cálculo da matriz Z(1) **/

Proc iml;
use tetad1; read all into tetad1;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd1=(Ident - D)*P*tetad1;
create zd1 from zd1; append from zd1; quit;
Proc iml;
use tetad2; read all into tetad2;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd2=(Ident - D)*P*tetad2;
create zd2 from zd2; append from zd2; quit;
Proc iml;
use tetad3; read all into tetad3;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd3=(Ident - D)*P*tetad3;
create zd3 from zd3; append from zd3; quit;
Proc iml;
use tetad4; read all into tetad4;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd4=(Ident - D)*P*tetad4;
create zd4 from zd4; append from zd4; quit;
Proc iml;
use tetad5; read all into tetad5;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd5=(Ident - D)*P*tetad5;
create zd5 from zd5; append from zd5; quit;
Proc iml;
use tetad6; read all into tetad6;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd6=(Ident - D)*P*tetad6;
create zd6 from zd6; append from zd6; quit;
Proc iml;
use tetad7; read all into tetad7;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd7=(Ident - D)*P*tetad7;
create zd7 from zd7; append from zd7; quit;
Proc iml;
use tetad8; read all into tetad8;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd8=(Ident - D)*P*tetad8;
create zd8 from zd8; append from zd8; quit;
Proc iml;
use tetad9; read all into tetad9;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd9=(Ident - D)*P*tetad9;
create zd9 from zd9; append from zd9; quit;
Proc iml;
use tetad10; read all into tetad10;
use P; read all into P;
use D; read all into D;
Ident=I(7);
```

```
zd10=(Ident - D)*P*tetad10;
create zd10 from zd10; append from zd10; quit;
Proc iml;
use tetad11; read all into tetad11;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd11=(Ident - D)*P*tetad11;
create zd11 from zd11; append from zd11; quit;
Proc iml;
use tetad12; read all into tetad12;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd12=(Ident - D)*P*tetad12;
create zd12 from zd12; append from zd12; quit;
Proc iml;
use tetad13; read all into tetad13;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd13=(Ident - D)*P*tetad13;
create zd13 from zd13; append from zd13; quit;
Proc iml;
use tetad14; read all into tetad14;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd14=(Ident - D)*P*tetad14;
create zd14 from zd14; append from zd14; quit;
Proc iml;
use tetad15; read all into tetad15;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd15=(Ident - D)*P*tetad15;
create zd15 from zd15; append from zd15; quit;
Proc iml;
use tetad16; read all into tetad16;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd16=(Ident - D)*P*tetad16;
create zd16 from zd16; append from zd16; quit;
Proc iml;
use tetad17; read all into tetad17;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd17=(Ident - D)*P*tetad17;
create zd17 from zd17; append from zd17; quit;
Proc iml;
use tetad18; read all into tetad18;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd18=(Ident - D)*P*tetad18;
create zd18 from zd18; append from zd18; quit;
Proc iml;
use tetad19; read all into tetad19;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd19=(Ident - D)*P*tetad19;
create zd19 from zd19; append from zd19; quit;
Proc iml;
use tetad20; read all into tetad20;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd20=(Ident - D)*P*tetad20;
create zd20 from zd20; append from zd20; quit;
Proc iml;
use tetad21; read all into tetad21;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd21=(Ident - D)*P*tetad21;
create zd21 from zd21; append from zd21; quit;
Proc iml;
use tetad22; read all into tetad22;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd22=(Ident - D)*P*tetad22;
create zd22 from zd22; append from zd22; quit;
Proc iml;
use tetad23; read all into tetad23;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd23=(Ident - D)*P*tetad23;
create zd23 from zd23; append from zd23; quit;
Proc iml;
use tetad24; read all into tetad24;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd24=(Ident - D)*P*tetad24;
create zd24 from zd24; append from zd24; quit;
Proc iml;
use tetad25; read all into tetad25;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd25=(Ident - D)*P*tetad25;
create zd25 from zd25; append from zd25; quit;
Proc iml;
use tetad26; read all into tetad26;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd26=(Ident - D)*P*tetad26;
create zd26 from zd26; append from zd26; quit;
Proc iml;
use tetad27; read all into tetad27;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd27=(Ident - D)*P*tetad27;
create zd27 from zd27; append from zd27; quit;
Proc iml;
use tetad28; read all into tetad28;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd28=(Ident - D)*P*tetad28;
create zd28 from zd28; append from zd28; quit;

Data Z1;
set zd1 zd2 zd3 zd4 zd5 zd6 zd7 zd8 zd9 zd10
zd11 zd12 zd13 zd14 zd15
zd16 zd17 zd18 zd19 zd20 zd21 zd22 zd23 zd24
zd25 zd26 zd27 zd28;
run;

/** Cálculo da matriz H(1) **/

Proc iml;
use xd1; read all into xd1;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd1=(Ident - D)*P*xd1;
create hd1 from hd1; append from hd1; quit;
Proc iml;
use xd2; read all into xd2;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd2=(Ident - D)*P*xd2;
create hd2 from hd2; append from hd2; quit;
Proc iml;
use xd3; read all into xd3;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd3=(Ident - D)*P*xd3;
create hd3 from hd3; append from hd3; quit;
Proc iml;
use xd4; read all into xd4;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd4=(Ident - D)*P*xd4;
create hd4 from hd4; append from hd4; quit;
Proc iml;
use xd5; read all into xd5;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd5=(Ident - D)*P*xd5;
create hd5 from hd5; append from hd5; quit;
Proc iml;
use xd6; read all into xd6;
use P; read all into P;
```

```
use D; read all into D;
Ident=I(7);
hd6=(Ident - D)*P*xd6;
create hd6 from hd6; append from hd6; quit;
Proc iml;
use xd7; read all into xd7;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd7=(Ident - D)*P*xd7;
create hd7 from hd7; append from hd7; quit;
Proc iml;
use xd8; read all into xd8;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd8=(Ident - D)*P*xd8;
create hd8 from hd8; append from hd8; quit;
Proc iml;
use xd9; read all into xd9;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd9=(Ident - D)*P*xd9;
create hd9 from hd9; append from hd9; quit;
Proc iml;
use xd10; read all into xd10;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd10=(Ident - D)*P*xd10;
create hd10 from hd10; append from hd10; quit;
Proc iml;
use xd11; read all into xd11;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd11=(Ident - D)*P*xd11;
create hd11 from hd11; append from hd11; quit;
Proc iml;
use xd12; read all into xd12;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd12=(Ident - D)*P*xd12;
create hd12 from hd12; append from hd12; quit;
Proc iml;
use xd13; read all into xd13;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd13=(Ident - D)*P*xd13;
create hd13 from hd13; append from hd13; quit;
Proc iml;
use xd14; read all into xd14;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd14=(Ident - D)*P*xd14;
create hd14 from hd14; append from hd14; quit;
Proc iml;
use xd15; read all into xd15;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd15=(Ident - D)*P*xd15;
create hd15 from hd15; append from hd15; quit;
Proc iml;
use xd16; read all into xd16;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd16=(Ident - D)*P*xd16;
create hd16 from hd16; append from hd16; quit;
Proc iml;
use xd17; read all into xd17;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd17=(Ident - D)*P*xd17;
create hd17 from hd17; append from hd17; quit;
Proc iml;
use xd18; read all into xd18;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd18=(Ident - D)*P*xd18;
create hd18 from hd18; append from hd18; quit;
Proc iml;
use xd19; read all into xd19;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd19=(Ident - D)*P*xd19;
create hd19 from hd19; append from hd19; quit;
Proc iml;
use xd20; read all into xd20;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd20=(Ident - D)*P*xd20;
create hd20 from hd20; append from hd20; quit;
Proc iml;
use xd21; read all into xd21;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd21=(Ident - D)*P*xd21;
create hd21 from hd21; append from hd21; quit;
Proc iml;
use xd22; read all into xd22;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd22=(Ident - D)*P*xd22;
create hd22 from hd22; append from hd22; quit;
Proc iml;
use xd23; read all into xd23;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd23=(Ident - D)*P*xd23;
create hd23 from hd23; append from hd23; quit;
Proc iml;
use xd24; read all into xd24;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd24=(Ident - D)*P*xd24;
create hd24 from hd24; append from hd24; quit;
Proc iml;
use xd25; read all into xd25;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd25=(Ident - D)*P*xd25;
create hd25 from hd25; append from hd25; quit;
Proc iml;
use xd26; read all into xd26;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd26=(Ident - D)*P*xd26;
create hd26 from hd26; append from hd26; quit;
Proc iml;
use xd27; read all into xd27;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd27=(Ident - D)*P*xd27;
create hd27 from hd27; append from hd27; quit;
Proc iml;
use xd28; read all into xd28;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd28=(Ident - D)*P*xd28;
create hd28 from hd28; append from hd28; quit;

Data H1;
set hd1 hd2 hd3 hd4 hd5 hd6 hd7 hd8 hd9 hd10
hd11 hd12 hd13 hd14 hd15
hd16 hd17 hd18 hd19 hd20 hd21 hd22 hd23 hd24
hd25 hd26 hd27 hd28;
run;

/** Regressão de Z(1) sobre H(1) **/

Data Z1;
set Z1;
rename col1=col0;
run;

Data reg;
merge Z1 H1;
run;
```

```
proc reg data=reg noprint;
        model col0=col1 col2 / noint;
        output out=rg1  residual=r1;
run; quit;

Data rg1;
set rg1;
qr1=r1*r1;
Keep qr1;
run;

Proc means data=rg1 sum noprint;
var qr1;
output out=rg1_ sum=sqr1;
run;

Data rg1_;
set rg1_;
keep sqr1;
run;

/** Cálculo da variância sigma2_u **/

Proc iml;
use H1; read all into H1;
Hs1=H1*ginv(t(H1)*H1)*t(H1); /* ginv(A) is the
Moore-Penrose generalized inverse matrix */
create Hs1 from Hs1; append from Hs1; quit; /*
Hs1 é a matriz H*(1) */

Proc iml;
use D; read all into D;
Ident=I(7);
ID=Ident - D; /* Matriz I - D */
create ID from ID; append from ID; quit;

Proc iml;
use ID; read all into ID;
blockID1=BLOCK(ID,ID,ID,ID,ID,ID,ID,ID,ID,ID,I
D,ID,ID,ID);
create blockID1 from blockID1; append from
blockID1; quit;
Proc iml;
use blockID1; read all into blockID1;
blockID=BLOCK(blockID1,blockID1);
create blockID from blockID; append from
blockID; quit;

Proc iml;
use P; read all into P;
P1=P*t(P); /* Admite-se que SIGMA_d = I */
create P1 from P1; append from P1; quit;

Proc iml;
use P1; read all into P1;
blockPP1=BLOCK(P1,P1,P1,P1,P1,P1,P1,P1,P1,P1,P
1,P1,P1,P1);
create blockPP1 from blockPP1; append from
blockPP1; quit;
Proc iml;
use blockPP1; read all into blockPP1;
blockPP=BLOCK(blockPP1,blockPP1);
create blockPP from blockPP; append from
blockPP; quit;

Proc iml;
use H1; read all into H1;
HH=H1*t(H1);
eval=eigval(HH);
create eval from eval; append from eval; quit;

Data id;
do id=1 to 196;
output; end;
run;

Data eval;
merge id eval;
run;

Data new;
set eval;
by id;
retain count 0;
if first.id then count=0;
if col1 > 0.000001 then count = count + 1;
run;
```

```
Proc means data=new sum noprint;
var count;
output out=caracH1 sum=caracH1;
run;

Data caracH1;
set caracH1;
keep caracH1;
run;

Proc iml;
use blockPP; read all into blockPP;
use blockID; read all into blockID;
use Hs1; read all into Hs1;
use rg1_; read all into rg1_;
use caracH1; read all into caracH1;
varu_Boot= (rg1_ -  trace((blockID -
Hs1)*blockPP))*(1/(28*6 - caracH1));
create varu_Boot from varu_Boot; append from
varu_Boot; quit;

/* Termina aqui o cálculo da variância
sigma2_u */

/** Cálculo da matriz: c^(-0.5)*f'*P*Teta -
var. dependente do modelo transformado **/

Proc iml;
use tetad1; read all into tetad1;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y1=(1/sqrt(c))*t(f)*P*tetad1;
create y1 from y1; append from y1; quit;
Proc iml;
use tetad2; read all into tetad2;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y2=(1/sqrt(c))*t(f)*P*tetad2;
create y2 from y2; append from y2; quit;
Proc iml;
use tetad3; read all into tetad3;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y3=(1/sqrt(c))*t(f)*P*tetad3;
create y3 from y3; append from y3; quit;
Proc iml;
use tetad4; read all into tetad4;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y4=(1/sqrt(c))*t(f)*P*tetad4;
create y4 from y4; append from y4; quit;
Proc iml;
use tetad5; read all into tetad5;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y5=(1/sqrt(c))*t(f)*P*tetad5;
create y5 from y5; append from y5; quit;
Proc iml;
use tetad6; read all into tetad6;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y6=(1/sqrt(c))*t(f)*P*tetad6;
create y6 from y6; append from y6; quit;
Proc iml;
use tetad7; read all into tetad7;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y7=(1/sqrt(c))*t(f)*P*tetad7;
create y7 from y7; append from y7; quit;
Proc iml;
use tetad8; read all into tetad8;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y8=(1/sqrt(c))*t(f)*P*tetad8;
create y8 from y8; append from y8; quit;
Proc iml;
use tetad9; read all into tetad9;
use P; read all into P;
use f; read all into f;
```

```
use c; read all into c;
y9=(1/sqrt(c))*t(f)*P*tetad9;
create y9 from y9; append from y9; quit;
Proc iml;
use tetad10; read all into tetad10;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y10=(1/sqrt(c))*t(f)*P*tetad10;
create y10 from y10; append from y10; quit;
Proc iml;
use tetad11; read all into tetad11;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y11=(1/sqrt(c))*t(f)*P*tetad11;
create y11 from y11; append from y11; quit;
Proc iml;
use tetad12; read all into tetad12;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y12=(1/sqrt(c))*t(f)*P*tetad12;
create y12 from y12; append from y12; quit;
Proc iml;
use tetad13; read all into tetad13;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y13=(1/sqrt(c))*t(f)*P*tetad13;
create y13 from y13; append from y13; quit;
Proc iml;
use tetad14; read all into tetad14;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y14=(1/sqrt(c))*t(f)*P*tetad14;
create y14 from y14; append from y14; quit;
Proc iml;
use tetad15; read all into tetad15;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y15=(1/sqrt(c))*t(f)*P*tetad15;
create y15 from y15; append from y15; quit;
Proc iml;
use tetad16; read all into tetad16;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y16=(1/sqrt(c))*t(f)*P*tetad16;
create y16 from y16; append from y16; quit;
Proc iml;
use tetad17; read all into tetad17;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y17=(1/sqrt(c))*t(f)*P*tetad17;
create y17 from y17; append from y17; quit;
Proc iml;
use tetad18; read all into tetad18;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y18=(1/sqrt(c))*t(f)*P*tetad18;
create y18 from y18; append from y18; quit;
Proc iml;
use tetad19; read all into tetad19;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y19=(1/sqrt(c))*t(f)*P*tetad19;
create y19 from y19; append from y19; quit;
Proc iml;
use tetad20; read all into tetad20;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y20=(1/sqrt(c))*t(f)*P*tetad20;
create y20 from y20; append from y20; quit;
Proc iml;
use tetad21; read all into tetad21;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y21=(1/sqrt(c))*t(f)*P*tetad21;
create y21 from y21; append from y21; quit;
Proc iml;
use tetad22; read all into tetad22;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y22=(1/sqrt(c))*t(f)*P*tetad22;
create y22 from y22; append from y22; quit;
Proc iml;
use tetad23; read all into tetad23;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y23=(1/sqrt(c))*t(f)*P*tetad23;
create y23 from y23; append from y23; quit;
Proc iml;
use tetad24; read all into tetad24;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y24=(1/sqrt(c))*t(f)*P*tetad24;
create y24 from y24; append from y24; quit;
Proc iml;
use tetad25; read all into tetad25;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y25=(1/sqrt(c))*t(f)*P*tetad25;
create y25 from y25; append from y25; quit;
Proc iml;
use tetad26; read all into tetad26;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y26=(1/sqrt(c))*t(f)*P*tetad26;
create y26 from y26; append from y26; quit;
Proc iml;
use tetad27; read all into tetad27;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y27=(1/sqrt(c))*t(f)*P*tetad27;
create y27 from y27; append from y27; quit;
Proc iml;
use tetad28; read all into tetad28;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y28=(1/sqrt(c))*t(f)*P*tetad28;
create y28 from y28; append from y28; quit;
run;

Data Y;
set y1 y2 y3 y4 y5 y6 y7 y8 y9 y10 y11 y12 y13
y14 y15 y16 y17 y18 y19 y20 y21 y22 y23 y24
y25 y26 y27 y28;
run;

/** Cálculo da matriz: c^(-0.5)*f'*P*X - var.
independentes do modelo transformado **/

Proc iml;
use xd1; read all into xd1;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x1=(1/sqrt(c))*t(f)*P*xd1;
create x1 from x1; append from x1; quit;
Proc iml;
use xd2; read all into xd2;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x2=(1/sqrt(c))*t(f)*P*xd2;
create x2 from x2; append from x2; quit;
Proc iml;
use xd3; read all into xd3;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x3=(1/sqrt(c))*t(f)*P*xd3;
create x3 from x3; append from x3; quit;
Proc iml;
use xd4; read all into xd4;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x4=(1/sqrt(c))*t(f)*P*xd4;
create x4 from x4; append from x4; quit;
Proc iml;
```

```
use xd5; read all into xd5;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x5=(1/sqrt(c))*t(f)*P*xd5;
create x5 from x5; append from x5; quit;
Proc iml;
use xd6; read all into xd6;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x6=(1/sqrt(c))*t(f)*P*xd6;
create x6 from x6; append from x6; quit;
Proc iml;
use xd7; read all into xd7;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x7=(1/sqrt(c))*t(f)*P*xd7;
create x7 from x7; append from x7; quit;
Proc iml;
use xd8; read all into xd8;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x8=(1/sqrt(c))*t(f)*P*xd8;
create x8 from x8; append from x8; quit;
Proc iml;
use xd9; read all into xd9;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x9=(1/sqrt(c))*t(f)*P*xd9;
create x9 from x9; append from x9; quit;
Proc iml;
use xd10; read all into xd10;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x10=(1/sqrt(c))*t(f)*P*xd10;
create x10 from x10; append from x10; quit;
Proc iml;
use xd11; read all into xd11;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x11=(1/sqrt(c))*t(f)*P*xd11;
create x11 from x11; append from x11; quit;
Proc iml;
use xd12; read all into xd12;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x12=(1/sqrt(c))*t(f)*P*xd12;
create x12 from x12; append from x12; quit;
Proc iml;
use xd13; read all into xd13;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x13=(1/sqrt(c))*t(f)*P*xd13;
create x13 from x13; append from x13; quit;
Proc iml;
use xd14; read all into xd14;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x14=(1/sqrt(c))*t(f)*P*xd14;
create x14 from x14; append from x14; quit;
Proc iml;
use xd15; read all into xd15;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x15=(1/sqrt(c))*t(f)*P*xd15;
create x15 from x15; append from x15; quit;
Proc iml;
use xd16; read all into xd16;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x16=(1/sqrt(c))*t(f)*P*xd16;
create x16 from x16; append from x16; quit;
Proc iml;
use xd17; read all into xd17;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x17=(1/sqrt(c))*t(f)*P*xd17;
create x17 from x17; append from x17; quit;
Proc iml;
use xd18; read all into xd18;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x18=(1/sqrt(c))*t(f)*P*xd18;
create x18 from x18; append from x18; quit;
Proc iml;
use xd19; read all into xd19;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x19=(1/sqrt(c))*t(f)*P*xd19;
create x19 from x19; append from x19; quit;
Proc iml;
use xd20; read all into xd20;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x20=(1/sqrt(c))*t(f)*P*xd20;
create x20 from x20; append from x20; quit;
Proc iml;
use xd21; read all into xd21;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x21=(1/sqrt(c))*t(f)*P*xd21;
create x21 from x21; append from x21; quit;
Proc iml;
use xd22; read all into xd22;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x22=(1/sqrt(c))*t(f)*P*xd22;
create x22 from x22; append from x22; quit;
Proc iml;
use xd23; read all into xd23;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x23=(1/sqrt(c))*t(f)*P*xd23;
create x23 from x23; append from x23; quit;
Proc iml;
use xd24; read all into xd24;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x24=(1/sqrt(c))*t(f)*P*xd24;
create x24 from x24; append from x24; quit;
Proc iml;
use xd25; read all into xd25;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x25=(1/sqrt(c))*t(f)*P*xd25;
create x25 from x25; append from x25; quit;
Proc iml;
use xd26; read all into xd26;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x26=(1/sqrt(c))*t(f)*P*xd26;
create x26 from x26; append from x26; quit;
Proc iml;
use xd27; read all into xd27;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x27=(1/sqrt(c))*t(f)*P*xd27;
create x27 from x27; append from x27; quit;
Proc iml;
use xd28; read all into xd28;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x28=(1/sqrt(c))*t(f)*P*xd28;
create x28 from x28; append from x28; quit;
run;

Data X;
set x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13
x14 x15
x16 x17 x18 x19 x20 x21 x22 x23 x24 x25 x26
x27 x28;
run;
```

```
/** Regressão de Y sobre X **/

Data Y;
set Y;
rename col1=col0;
run;

Data reg2;
merge Y X; run;

proc reg data=reg2 noprint;
        model col0=col1 col2 / noint;
        output out=rg2  residual=r2;
run; quit;

Data rg2;
set rg2;
qr2=r2*r2;
Keep qr2;
run;

Proc means data=rg2 sum noprint;
var qr2;
output out=rg2_ sum=sqr2;
run;

Data rg2_;
set rg2_;
keep sqr2;
run;

/** Cálculo da variância sigma2_v **/

Proc iml;
use xd1; read all into xd1;
use P; read all into P;
use f; read all into f;
xpf1=t(xd1)*t(P)*f;
create xpf1 from xpf1; append from xpf1; quit;
Proc iml;
use xd2; read all into xd2;
use P; read all into P;
use f; read all into f;
xpf2=t(xd2)*t(P)*f;
create xpf2 from xpf2; append from xpf2; quit;
Proc iml;
use xd3; read all into xd3;
use P; read all into P;
use f; read all into f;
xpf3=t(xd3)*t(P)*f;
create xpf3 from xpf3; append from xpf3; quit;
Proc iml;
use xd4; read all into xd4;
use P; read all into P;
use f; read all into f;
xpf4=t(xd4)*t(P)*f;
create xpf4 from xpf4; append from xpf4; quit;
Proc iml;
use xd5; read all into xd5;
use P; read all into P;
use f; read all into f;
xpf5=t(xd5)*t(P)*f;
create xpf5 from xpf5; append from xpf5; quit;
Proc iml;
use xd6; read all into xd6;
use P; read all into P;
use f; read all into f;
xpf6=t(xd6)*t(P)*f;
create xpf6 from xpf6; append from xpf6; quit;
Proc iml;
use xd7; read all into xd7;
use P; read all into P;
use f; read all into f;
xpf7=t(xd7)*t(P)*f;
create xpf7 from xpf7; append from xpf7; quit;
Proc iml;
use xd8; read all into xd8;
use P; read all into P;
use f; read all into f;
xpf8=t(xd8)*t(P)*f;
create xpf8 from xpf8; append from xpf8; quit;
Proc iml;
use xd9; read all into xd9;
use P; read all into P;
use f; read all into f;
xpf9=t(xd9)*t(P)*f;
create xpf9 from xpf9; append from xpf9; quit;
Proc iml;
use xd10; read all into xd10;
use P; read all into P;
use f; read all into f;
xpf10=t(xd10)*t(P)*f;
create xpf10 from xpf10; append from xpf10;
quit;
Proc iml;
use xd11; read all into xd11;
use P; read all into P;
use f; read all into f;
xpf11=t(xd11)*t(P)*f;
create xpf11 from xpf11; append from xpf11;
quit;
Proc iml;
use xd12; read all into xd12;
use P; read all into P;
use f; read all into f;
xpf12=t(xd12)*t(P)*f;
create xpf12 from xpf12; append from xpf12;
quit;
Proc iml;
use xd13; read all into xd13;
use P; read all into P;
use f; read all into f;
xpf13=t(xd13)*t(P)*f;
create xpf13 from xpf13; append from xpf13;
quit;
Proc iml;
use xd14; read all into xd14;
use P; read all into P;
use f; read all into f;
xpf14=t(xd14)*t(P)*f;
create xpf14 from xpf14; append from xpf14;
quit;
Proc iml;
use xd15; read all into xd15;
use P; read all into P;
use f; read all into f;
xpf15=t(xd15)*t(P)*f;
create xpf15 from xpf15; append from xpf15;
quit;
Proc iml;
use xd16; read all into xd16;
use P; read all into P;
use f; read all into f;
xpf16=t(xd16)*t(P)*f;
create xpf16 from xpf16; append from xpf16;
quit;
Proc iml;
use xd17; read all into xd17;
use P; read all into P;
use f; read all into f;
xpf17=t(xd17)*t(P)*f;
create xpf17 from xpf17; append from xpf17;
quit;
Proc iml;
use xd18; read all into xd18;
use P; read all into P;
use f; read all into f;
xpf18=t(xd18)*t(P)*f;
create xpf18 from xpf18; append from xpf18;
quit;
Proc iml;
use xd19; read all into xd19;
use P; read all into P;
use f; read all into f;
xpf19=t(xd19)*t(P)*f;
create xpf19 from xpf19; append from xpf19;
quit;
Proc iml;
use xd20; read all into xd20;
use P; read all into P;
use f; read all into f;
xpf20=t(xd20)*t(P)*f;
create xpf20 from xpf20; append from xpf20;
quit;
Proc iml;
use xd21; read all into xd21;
use P; read all into P;
use f; read all into f;
xpf21=t(xd21)*t(P)*f;
create xpf21 from xpf21; append from xpf21;
quit;
Proc iml;
use xd22; read all into xd22;
use P; read all into P;
use f; read all into f;
xpf22=t(xd22)*t(P)*f;
```

```
create xpf22 from xpf22; append from xpf22;
quit;
Proc iml;
use xd23; read all into xd23;
use P; read all into P;
use f; read all into f;
xpf23=t(xd23)*t(P)*f;
create xpf23 from xpf23; append from xpf23;
quit;
Proc iml;
use xd24; read all into xd24;
use P; read all into P;
use f; read all into f;
xpf24=t(xd24)*t(P)*f;
create xpf24 from xpf24; append from xpf24;
quit;
Proc iml;
use xd25; read all into xd25;
use P; read all into P;
use f; read all into f;
xpf25=t(xd25)*t(P)*f;
create xpf25 from xpf25; append from xpf25;
quit;
Proc iml;
use xd26; read all into xd26;
use P; read all into P;
use f; read all into f;
xpf26=t(xd26)*t(P)*f;
create xpf26 from xpf26; append from xpf26;
quit;
Proc iml;
use xd27; read all into xd27;
use P; read all into P;
use f; read all into f;
xpf27=t(xd27)*t(P)*f;
create xpf27 from xpf27; append from xpf27;
quit;
Proc iml;
use xd28; read all into xd28;
use P; read all into P;
use f; read all into f;
xpf28=t(xd28)*t(P)*f;
create xpf28 from xpf28; append from xpf28;
quit; run;

Proc iml;
use xpf1; read all into xpf1;
use xpf2; read all into xpf2;
use xpf3; read all into xpf3;
use xpf4; read all into xpf4;
use xpf5; read all into xpf5;
use xpf6; read all into xpf6;
use xpf7; read all into xpf7;
use xpf8; read all into xpf8;
use xpf9; read all into xpf9;
use xpf10; read all into xpf10;
use xpf11; read all into xpf11;
use xpf12; read all into xpf12;
use xpf13; read all into xpf13;
use xpf14; read all into xpf14;
use xpf15; read all into xpf15;
use xpf16; read all into xpf16;
use xpf17; read all into xpf17;
use xpf18; read all into xpf18;
use xpf19; read all into xpf19;
use xpf20; read all into xpf20;
use xpf21; read all into xpf21;
use xpf22; read all into xpf22;
use xpf23; read all into xpf23;
use xpf24; read all into xpf24;
use xpf25; read all into xpf25;
use xpf26; read all into xpf26;
use xpf27; read all into xpf27;
use xpf28; read all into xpf28;
Ft=xpf1||xpf2||xpf3||xpf4||xpf5||xpf6||xpf7||x
pf8||xpf9||xpf10||xpf11||xpf12||xpf13||xpf14||
xpf15||xpf16||xpf17||xpf18||xpf19||xpf20||xpf2
1||xpf22||xpf23||xpf24||xpf25||xpf26||xpf27||x
pf28;
create Ft from Ft; append from Ft; quit;

Proc iml;
use Ft; read all into Ft;
F1=t(Ft);
create F1 from F1; append from F1; quit;

Proc iml;
use F1; read all into F1;
Fast=F1*ginv(t(F1)*F1)*t(F1);
create Fast from Fast; append from Fast; quit;

Proc iml;
use Fast; read all into Fast;
Ident=I(28);
IF=Ident - Fast; /* Matriz I - F* */
create IF from IF; append from IF; quit;

Proc iml;
use f; read all into f;
use P; read all into P;
use c; read all into c;
d1=t(f)*P*t(P)*f*(1 / c); /* Admite-se que
SIGMA_d = I */
Ident=I(28);
cfp=d1*Ident;
create cfp from cfp; append from cfp; quit;

Proc iml;
use F1; read all into F1;
FF=F1*t(F1);
eval1=eigval(FF);
create eval1 from eval1; append from eval1;
quit;

Data id1;
do id=1 to 28;
output; end;
run;

Data eval1;
merge id1 eval1;
run;

Data new1;
set eval1;
by id;
retain count 0;
if first.id then count=0;
if col1 > 0.000001 then count = count + 1;
run;

Proc means data=new1 sum noprint;
var count;
output out=caracF sum=caracF;
run;

Data caracF;
set caracF;
keep caracF;
run;

Proc iml;
use IF; read all into IF;
use cfp; read all into cfp;
use c; read all into c;
use rg2_; read all into rg2_;
use caracF; read all into caracF;
use varu_Boot; read all into varu_Boot;
varv_Boot= (rg2_ -  trace(IF*cfp))*(1/(c*(28 -
caracF))) - varu_Boot/c;
create varv_Boot from varv_Boot; append from
varv_Boot; quit;

/* Termina aqui o cálculo da variância
sigma2_v */
/* Truncagem a zero das variâncias sigma2_u e
sigma2_v */

Data varu_Boot;
set varu_Boot;
sigma2u= max( col1 , 0);
keep sigma2u;
run;
Data varv_Boot;
set varv_Boot;
sigma2v= max( col1 , 0);
keep sigma2v;
run;

/* Construção do ficheiro com os parâmetros de
variância e de correlação */

Proc iml;
use dataro; read all into dataro;
use varu_Boot; read all into varu_Boot;
use varv_Boot; read all into varv_Boot;
```

```
C1={1, 0, 0, 0};
C2={0, 1, 0, 0};
C3={0, 0, 1, 0};
C4={0, 0, 0, 1};
par1 = varv_Boot*C1 + (varu_Boot/(1 -
dataro*dataro))*C2 + dataro*C3 + C4;
create par1 from par1; append from par1; quit;

Data par_Boot;
set par1;
rename col1=est;
run;

Data CompVar_Boot;
set par_Boot;
boot=&i;
run;

/*******************************************/
/* Fim da estimação das comp. variância com um
dos conjuntos de dados Bootstrap */
/*******************************************/

/********************************************
********************************************/
/****** ESTIMAÇÃO DO MODELO DE RAO-YU *******/
/** utilizando as estimativas das componentes
de variância Base - par_base - *************/
/********************************************
********************************************/

Proc sort data=iabhipth1 out= cron;
by i t;
run;

Proc mixed data=cron MMEqSol noinfo noprofile
noclprint;
        class i t;
        model y=x / solution;
        random i / solution G V;
        random t / subject=i type=ar(1);
        parms /parmsdata=par_base noiter;
make 'solutionf' out=beta;
make 'solutionr' out=u;
run; quit;

/********************************************
********************************************/
/******* CÁLCULO DOS EFEITOS FIXOS  ********/
/** utilizando as estimativas das componentes
de variância Base - par_base - **************/
/********************************************
********************************************/

Data id;
do id=1 to 2;
output; end;
run;

Data beta;
merge id beta;
run;

Data beta0;
set beta;
if id=1;
rename estimate=b0;
keep estimate;
run;

Data beta1;
set beta;
if id=2;
rename estimate=b1;
keep estimate;
run;

Data betan_boot_base;
merge beta0 beta1;
_type_=0;
run;

/****** Efeitos Aleatórios: v ******/

Data u;
set u;
if estimate='.' then estimate=0;
run;

Data v;
set u;
if Effect='i';
keep i estimate;
run;

Proc sort data=v;
by i;
run;

Proc iml;
use v; read all into v;
one=J(7,1,1);
v1=v@one;
create v1 from v1; append from v1; quit;

Data v1;
set v1;
_type_=0;
rename col2=v;
run;

/****** Efeitos Aleatórios: u ******/

Data u1;
set u;
if Effect='t';
_type_=0;
keep i t estimate _type_;
rename estimate=u;
run;

Proc sort data=u1;
by i t;
run;

Data beta_vu;
merge betan_boot_base v1 u1;
by _type_;
run;

/** Estimação do preço médio de transacção em
cada NUTIII **/

Proc sort data=cron;
by i t;
run;

Data cron1;
merge cron beta_vu;
by i t;
run;

Data est_RYU_boot_base;
set cron1;
eb=b0+x*b1;
e_base = eb + v + u;
boot=&i;
keep i t x b0 b1 eb u v e_base boot;
run;

/********************************************
********************************************/
/***** ESTIMAÇÃO DO MODELO DE RAO-YU ********/
/** utilizando as estimativas das componentes
de variância Boot - par_boot - **************/
/********************************************
********************************************/

Proc sort data=iabhipth1 out= cron;
by i t;
run;

Proc mixed data=cron MMEqSol noinfo noprofile
noclprint;
        class i t;
        model y=x / solution;
        random i / solution G V;
        random t / subject=i type=ar(1);
        parms /parmsdata=par_boot noiter;
make 'solutionf' out=beta;
make 'solutionr' out=u;
make 'G' out=Mat_Gaux;
make 'V' out=Mat_V;
run; quit;
```

```
/*********************************************
*********************************************/
/******** CÁLCULO DOS EFEITOS FIXOS  ********/
/** utilizando as estimativas das componentes
de variância Boot - par_boot - ************/
/*********************************************
*********************************************/

Data id;
do id=1 to 2;
output; end;
run;

Data beta;
merge id beta;
run;

Data beta0;
set beta;
if id=1;
rename estimate=b0;
keep estimate;
run;

Data beta1;
set beta;
if id=2;
rename estimate=b1;
keep estimate;
run;

Data betan_boot_boot;
merge beta0 beta1;
_type_=0;
run;

/****** Efeitos Aleatórios: v ******/

Data u;
set u;
if estimate='.' then estimate=0;
run;

Data v;
set u;
if Effect='i';
keep i estimate;
run;

Proc sort data=v;
by i;
run;

Proc iml;
use v; read all into v;
one=J(7,1,1);
v1=v@one;
create v1 from v1; append from v1; quit;

Data v1;
set v1;
_type_=0;
rename col2=v;
run;

/****** Efeitos Aleatórios: u ******/

Data u1;
set u;
if Effect='t';
_type_=0;
keep i t estimate _type_;
rename estimate=u;
run;

Proc sort data=u1;
by i t;
run;

Data beta_vu;
merge betan_boot_boot v1 u1;
by _type_;
run;

Proc sort data=cron;
by i t;
run;

Data cron2;
merge cron beta_vu;
by i t;
run;

Data est_RYU_boot_boot;
set cron2;
eb=b0+x*b1;
e_boot = eb + v + u;
boot=&i;
keep i t x b0 b1 eb u v e_boot boot;
run;

/*********************************************
*********************************************/
/** Cálculo das estimativas do G3 do EBLUP **/
/*** utilizando as estimativas das componentes
de variância Boot - par_boot - *************/
/*********************************************
*********************************************/

Data est_RYU_mod_g3;
merge est_RYU_boot_boot est_RYU_boot_base;
by i t;
DIF2=(e_boot - e_base)*(e_boot - e_base);
keep i t e_base e_boot boot dif2;
run;

/*********************************************
*********************************************/
/*Cálculo das estimativas do G1, G2 do EBLUP*/
/*** utilizando as estimativas das componentes
de variância Boot - par_boot - *************/
/*********************************************
*********************************************/

/*** MATRIZ X ***/

Proc sort data=cron out=cronologico;
by i t;
run;

Data X1;
set cronologico;
keep x;
run;

Proc iml;
J=J(196,1,1);
create J from J;
append from J;
quit;

Proc iml;
use J;
read all into J;
use X1;
read all into X1;
X=insert(J,X1,0,2);
create X from X;
append from X;
quit;

/*** MATRIZ G ***/

Proc iml;
J7=J(7,1,1);
create J7 from J7;
append from J7;
quit;

Proc iml;
use J7;
read all into J7;
use varv_boot;
read all into varv_boot;
J77=J7*t(J7);
BlockMat_Gv=varv_boot*J77;
create BlockMat_Gv from BlockMat_Gv;
append from BlockMat_Gv;
quit;

Proc iml;
use BlockMat_Gv;
read all into BlockMat_Gv;
Mat_Gv1=block (BlockMat_Gv, BlockMat_Gv,
BlockMat_Gv, BlockMat_Gv, BlockMat_Gv,
BlockMat_Gv, BlockMat_Gv,
```

```
BlockMat_Gv, BlockMat_Gv, BlockMat_Gv,
BlockMat_Gv, BlockMat_Gv, BlockMat_Gv,
BlockMat_Gv);
create Mat_Gv1 from Mat_Gv1;
append from Mat_Gv1;
quit;

Proc iml;
use Mat_Gv1;
read all into Mat_Gv1;
Mat_Gv=block (Mat_Gv1, Mat_Gv1);
create Mat_Gv from Mat_Gv;
append from Mat_Gv;
quit;

/***/

Data Mat_Gu;
set Mat_Gaux;
where row>28;
drop row effect i t col1-col28;
run;

/***/

Proc iml;
use Mat_Gv;
read all into Mat_Gv;
use Mat_Gu;
read all into Mat_Gu;
MatG=Mat_Gv+Mat_Gu;
create MatG from MatG;
append from MatG;
quit;

/*** Matriz GAMA e suas colunas ***/

Data ro_1;
set dataro;
ro_1= 1-ro*ro;
keep ro_1;
run;
Data ro_2;
set dataro;
ro_2= ro*ro;
keep ro_2;
run;
Data ro_3;
set dataro;
ro_3= ro*ro*ro;
keep ro_3;
run;
Data ro_4;
set dataro;
ro_4= ro*ro*ro*ro;
keep ro_4;
run;
Data ro_5;
set dataro;
ro_5= ro*ro*ro*ro*ro;
keep ro_5;
run;
Data ro_6;
set dataro;
ro_6= ro*ro*ro*ro*ro*ro;
keep ro_6;
run;

Proc iml;
use ro_1; read all into ro_1;
use dataro; read all into dataro;
use ro_2; read all into ro_2;
use ro_3; read all into ro_3;
use ro_4; read all into ro_4;
use ro_5; read all into ro_5;
use ro_6; read all into ro_6;
I=I(7);
I1={0 1 0 0 0 0 0, 0 0 1 0 0 0 0, 0 0 0 1 0 0
0, 0 0 0 0 1 0 0, 0 0 0 0 0 1 0, 0 0 0 0 0 0
1, 0 0 0 0 0 0 0};
I2={0 0 1 0 0 0 0, 0 0 0 1 0 0 0, 0 0 0 0 1 0
0, 0 0 0 0 0 1 0, 0 0 0 0 0 0 1, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I3={0 0 0 1 0 0 0, 0 0 0 0 1 0 0, 0 0 0 0 0 1
0, 0 0 0 0 0 0 1, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I4={0 0 0 0 1 0 0, 0 0 0 0 0 1 0, 0 0 0 0 0 0
1, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I5={0 0 0 0 0 1 0, 0 0 0 0 0 0 1, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I6={0 0 0 0 0 0 1, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
Gama = I/ro_1 + dataro*I1/ro_1 +
t(dataro*I1/ro_1) + ro_2*I2/ro_1 +
t(ro_2*I2/ro_1) + ro_3*I3/ro_1 +
t(ro_3*I3/ro_1) + ro_4*I4/ro_1 +
t(ro_4*I4/ro_1) + ro_5*I5/ro_1 +
t(ro_5*I5/ro_1) + ro_6*I6/ro_1 +
t(ro_6*I6/ro_1);
create Gama from Gama;
append from Gama;
quit;

Proc iml;
use Gama; read all into Gama;
Gama1=Gama[,1];
create Gama1 from Gama1;
append from Gama1;
quit;
Proc iml;
use Gama; read all into Gama;
Gama2=Gama[,2];
create Gama2 from Gama2;
append from Gama2;
quit;
Proc iml;
use Gama; read all into Gama;
Gama3=Gama[,3];
create Gama3 from Gama3;
append from Gama3;
quit;
Proc iml;
use Gama; read all into Gama;
Gama4=Gama[,4];
create Gama4 from Gama4;
append from Gama4;
quit;
Proc iml;
use Gama; read all into Gama;
Gama5=Gama[,5];
create Gama5 from Gama5;
append from Gama5;
quit;
Proc iml;
use Gama; read all into Gama;
Gama6=Gama[,6];
create Gama6 from Gama6;
append from Gama6;
quit;
Proc iml;
use Gama; read all into Gama;
Gama7=Gama[,7];
create Gama7 from Gama7;
append from Gama7;
quit;

/*** MATRIZ V e seus Blocos ***/

Data MatV;
set Mat_v;
drop Index row;
run;

Proc iml;
use BlockMat_Gv; read all into BlockMat_Gv;
use Gama; read all into Gama;
use varu_boot; read all into varu_boot;
I=I(7);
MatVi=I + varu_boot*Gama + BlockMat_Gv;
create MatVi from MatVi;
append from MatVi;
quit;

/**************************/
/****** Cálculo do g1 ******/
/**************************/

Proc iml;
use MatV; read all into MatV;
use MatG; read all into MatG;
G1_aux=MatG - MatG*inv(MatV)*MatG;
```

```
create G1_aux from G1_aux;
append from G1_aux;
quit;

Proc iml;
use G1_aux;
read all into G1_aux;
G1=vecdiag(G1_aux);
create G1 from G1;
append from G1;
quit;

Data g1_boot;
set g1;
rename col1=g1_boot;
run;

/**************************/
/****** Cálculo do g2 ******/
/**************************/

Proc iml;
use X; read all into X;
use MatV; read all into MatV;
use MatG; read all into MatG;
A=I(196);
G2_aux=t(t(X) -
t(X)*inv(MatV)*MatG*A)*inv(t(X)*inv(MatV)*X)*(
t(X) - t(X)*inv(MatV)*MatG*A);
create G2_aux from G2_aux;
append from G2_aux;
quit;

Proc iml;
use G2_aux;
read all into G2_aux;
G2=vecdiag(G2_aux);
create G2 from G2;
append from G2;
quit;

Data g2_boot;
set g2;
rename col1=g2_boot;
run;

Data Est_RYU_EQM_mod;
merge est_ryu_mod_g3 g1_boot g2_boot;
run;

Proc append base=RaoYu.Gerac_RYU_mod1_Boot_S50
data=a8;   /*SSS*/
run;
Proc append
base=RaoYu.CompVar_RYU_mod1_Boot_S50
data=CompVar_Boot;   /*SSS*/
run;
Proc append
base=RaoYu.Est_RYU_mod1_boot_base_S50
data=est_RYU_boot_base;   /*SSS*/
run;
Proc append
base=RaoYu.Est_RYU_mod1_boot_boot_S50
data=est_RYU_boot_boot;   /*SSS*/
run;
Proc append
base=RaoYu.Est_RYU_EQM_mod1_Boot_S50
data=Est_RYU_EQM_mod;   /*SSS*/
run;

DM 'CLEAR OUTPUT';
DM 'CLEAR LOG';

%mend;
%inc 'C:\Documents and Settings\Politriudo\Os
meus documentos\Luís\SAS code\chamamacro.txt';


/******************************************
******************************************/
/***** Estimação do EQMP do EBLUP utilizando a
aproximação Jackknife *********************/
/** Código SAS utilizado depois de estimado o
modelo de Rao-Yu utilizado para cada conjunto
de dados ***********************************/
/******************************************
******************************************/
```

```
/* O PROC MIXED calcula a matriz de
variâncias-covariâncias, Mat_V=Cov(Y), a
matriz Gu=Cov(u) e a matriz Gv=Cov(Zv). Estas
duas matrizes podem ser obtidas a partir de
uma matriz denominada Mat_Gaux. Depois
calcula-se G=Gu+Gv. Neste caso a matriz Z do
modelo linear geral misto teórico é a
identidade, Z=I, porque se admitiu que no
modelo de Rao-Yu, u*=Zv+u */

/*** MATRIZ X ***/

Proc sort data=cron out=cronologico;
by i t;
run;

Data X1;
set cronologico;
keep x;
run;

Proc iml;
J=J(196,1,1);
create J from J;
append from J;
quit;

Proc iml;
use J;
read all into J;
use X1;
read all into X1;
X=insert(J,X1,0,2);
create X from X;
append from X;
quit;

/*** MATRIZ G ***/

Proc iml;
J7=J(7,1,1);
create J7 from J7;
append from J7;
quit;

Proc iml;
use J7;
read all into J7;
use varv;
read all into varv;
J77=J7*J7`;
BlockMat_Gv=varv*J77;
create BlockMat_Gv from BlockMat_Gv;
append from BlockMat_Gv;
quit;

Proc iml;
use BlockMat_Gv;
read all into BlockMat_Gv;
Mat_Gv1=block (BlockMat_Gv, BlockMat_Gv,
BlockMat_Gv, BlockMat_Gv, BlockMat_Gv,
BlockMat_Gv, BlockMat_Gv,
BlockMat_Gv, BlockMat_Gv, BlockMat_Gv,
BlockMat_Gv, BlockMat_Gv, BlockMat_Gv,
BlockMat_Gv);
create Mat_Gv1 from Mat_Gv1;
append from Mat_Gv1;
quit;

Proc iml;
use Mat_Gv1;
read all into Mat_Gv1;
Mat_Gv=block (Mat_Gv1, Mat_Gv1);
create Mat_Gv from Mat_Gv;
append from Mat_Gv;
quit;

/***/

Data Mat_Gu;
set Mat_Gaux;
where row>28;
drop row effect i t col1-col28;
run;

/***/

Proc iml;
```

```
use Mat_Gv;
read all into Mat_Gv;
use Mat_Gu;
read all into Mat_Gu;
MatG=Mat_Gv+Mat_Gu;
create MatG from MatG;
append from MatG;
quit;

/*** Matriz GAMA e suas colunas ***/

Data ro_1;
set dataro;
ro_1= 1-ro*ro;
keep ro_1;
run;
Data ro_2;
set dataro;
ro_2= ro*ro;
keep ro_2;
run;
Data ro_3;
set dataro;
ro_3= ro*ro*ro;
keep ro_3;
run;
Data ro_4;
set dataro;
ro_4= ro*ro*ro*ro;
keep ro_4;
run;
Data ro_5;
set dataro;
ro_5= ro*ro*ro*ro*ro;
keep ro_5;
run;
Data ro_6;
set dataro;
ro_6= ro*ro*ro*ro*ro*ro;
keep ro_6;
run;

Proc iml;
use ro_1; read all into ro_1;
use dataro; read all into dataro;
use ro_2; read all into ro_2;
use ro_3; read all into ro_3;
use ro_4; read all into ro_4;
use ro_5; read all into ro_5;
use ro_6; read all into ro_6;
I=I(7);
I1={0 1 0 0 0 0 0, 0 0 1 0 0 0 0, 0 0 0 1 0 0
0, 0 0 0 0 1 0 0, 0 0 0 0 0 1 0, 0 0 0 0 0 0
1, 0 0 0 0 0 0 0};
I2={0 0 1 0 0 0 0, 0 0 0 1 0 0 0, 0 0 0 0 1 0
0, 0 0 0 0 0 1 0, 0 0 0 0 0 0 1, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I3={0 0 0 1 0 0 0, 0 0 0 0 1 0 0, 0 0 0 0 0 1
0, 0 0 0 0 0 0 1, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I4={0 0 0 0 1 0 0, 0 0 0 0 0 1 0, 0 0 0 0 0 0
1, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I5={0 0 0 0 0 1 0, 0 0 0 0 0 0 1, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I6={0 0 0 0 0 0 1, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
Gama = I/ro_1 + dataro*I1/ro_1 +
t(dataro*I1/ro_1) + ro_2*I2/ro_1 +
t(ro_2*I2/ro_1) + ro_3*I3/ro_1 +
t(ro_3*I3/ro_1) +
ro_4*I4/ro_1 + t(ro_4*I4/ro_1) + ro_5*I5/ro_1
+ t(ro_5*I5/ro_1) + ro_6*I6/ro_1 +
t(ro_6*I6/ro_1);
create Gama from Gama;
append from Gama;
quit;

Proc iml;
use Gama; read all into Gama;
Gama1=Gama[,1];
create Gama1 from Gama1;
append from Gama1;
quit;
Proc iml;
use Gama; read all into Gama;
Gama2=Gama[,2];
create Gama2 from Gama2;
append from Gama2;
quit;
Proc iml;
use Gama; read all into Gama;
Gama3=Gama[,3];
create Gama3 from Gama3;
append from Gama3;
quit;
Proc iml;
use Gama; read all into Gama;
Gama4=Gama[,4];
create Gama4 from Gama4;
append from Gama4;
quit;
Proc iml;
use Gama; read all into Gama;
Gama5=Gama[,5];
create Gama5 from Gama5;
append from Gama5;
quit;
Proc iml;
use Gama; read all into Gama;
Gama6=Gama[,6];
create Gama6 from Gama6;
append from Gama6;
quit;
Proc iml;
use Gama; read all into Gama;
Gama7=Gama[,7];
create Gama7 from Gama7;
append from Gama7;
quit;

/*** MATRIZ V e seus Blocos ***/

Data MatV;
set Mat_v;
drop Index row;
run;

Proc iml;
use BlockMat_Gv; read all into BlockMat_Gv;
use Gama; read all into Gama;
use varu; read all into varu;
I=I(7);
MatVi=I + varu*Gama + BlockMat_Gv;
create MatVi from MatVi;
append from MatVi;
quit;

/***************************************/
/****** Cálculo da derivada de G1 ******/
/***************************************/

/* O Cálculo da derivada de G1 tem que ser
feito individualmente para cada trimestre.
Note-se que o vector coluna "gama" varia em
função do trimeste e que a matriz "Vi" é
constante para diferentes domínios. */

/* Trabalho sobre a matriz da derivada de G1-
cálculo de 7 matrizes: dg1_1, dg1_2, dg1_3,
dg1_4, dg1_5, dg1_6 e dg1_7 */
/* 1º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama1; read all into Gama1;
use MatVi; read all into MatVi;
use ro_1; read all into ro_1;
J7=J(7,1,1);
J77=J(7,7,1);
dg1_varu= 1/ro_1 + ( t(varv*J7 +
varu*Gama1)*inv(MatVi)*Gama - 2*t(Gama1)
)*inv(MatVi)*(varv*J7 + varu*Gama1);
dg1_varv= 1 + ( t(varv*J7 +
varu*Gama1)*inv(MatVi)*J77 - 2*t(J7)
)*inv(MatVi)*(varv*J7 + varu*Gama1);
dg1_1=dg1_varu//dg1_varv;
create dg1_1 from dg1_1; append from dg1_1;
quit;

/* 2º trimestre */
```

```
Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama2; read all into Gama2;
use MatVi; read all into MatVi;
use ro_1; read all into ro_1;
J7=J(7,1,1);
J77=J(7,7,1);
dg1_varu= 1/ro_1 + ( t(varv*J7 +
varu*Gama2)*inv(MatVi)*Gama - 2*t(Gama2)
)*inv(MatVi)*(varv*J7 + varu*Gama2);
dg1_varv= 1 + ( t(varv*J7 +
varu*Gama2)*inv(MatVi)*J77 - 2*t(J7)
)*inv(MatVi)*(varv*J7 + varu*Gama2);
dg1_2=dg1_varu//dg1_varv;
create dg1_2 from dg1_2; append from dg1_2;
quit;

/* 3º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama3; read all into Gama3;
use MatVi; read all into MatVi;
use ro_1; read all into ro_1;
J7=J(7,1,1);
J77=J(7,7,1);
dg1_varu= 1/ro_1 + ( t(varv*J7 +
varu*Gama3)*inv(MatVi)*Gama - 2*t(Gama3)
)*inv(MatVi)*(varv*J7 + varu*Gama3);
dg1_varv= 1 + ( t(varv*J7 +
varu*Gama3)*inv(MatVi)*J77 - 2*t(J7)
)*inv(MatVi)*(varv*J7 + varu*Gama3);
dg1_3=dg1_varu//dg1_varv;
create dg1_3 from dg1_3; append from dg1_3;
quit;

/* 4º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama4; read all into Gama4;
use MatVi; read all into MatVi;
use ro_1; read all into ro_1;
J7=J(7,1,1);
J77=J(7,7,1);
dg1_varu= 1/ro_1 + ( t(varv*J7 +
varu*Gama4)*inv(MatVi)*Gama - 2*t(Gama4)
)*inv(MatVi)*(varv*J7 + varu*Gama4);
dg1_varv= 1 + ( t(varv*J7 +
varu*Gama4)*inv(MatVi)*J77 - 2*t(J7)
)*inv(MatVi)*(varv*J7 + varu*Gama4);
dg1_4=dg1_varu//dg1_varv;
create dg1_4 from dg1_4; append from dg1_4;
quit;

/* 5º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama5; read all into Gama5;
use MatVi; read all into MatVi;
use ro_1; read all into ro_1;
J7=J(7,1,1);
J77=J(7,7,1);
dg1_varu= 1/ro_1 + ( t(varv*J7 +
varu*Gama5)*inv(MatVi)*Gama - 2*t(Gama5)
)*inv(MatVi)*(varv*J7 + varu*Gama5);
dg1_varv= 1 + ( t(varv*J7 +
varu*Gama5)*inv(MatVi)*J77 - 2*t(J7)
)*inv(MatVi)*(varv*J7 + varu*Gama5);
dg1_5=dg1_varu//dg1_varv;
create dg1_5 from dg1_5; append from dg1_5;
quit;

/* 6º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
```

```
use Gama6; read all into Gama6;
use MatVi; read all into MatVi;
use ro_1; read all into ro_1;
J7=J(7,1,1);
J77=J(7,7,1);
dg1_varu= 1/ro_1 + ( t(varv*J7 +
varu*Gama6)*inv(MatVi)*Gama - 2*t(Gama6)
)*inv(MatVi)*(varv*J7 + varu*Gama6);
dg1_varv= 1 + ( t(varv*J7 +
varu*Gama6)*inv(MatVi)*J77 - 2*t(J7)
)*inv(MatVi)*(varv*J7 + varu*Gama6);
dg1_6=dg1_varu//dg1_varv;
create dg1_6 from dg1_6; append from dg1_6;
quit;

/* 7º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama7; read all into Gama7;
use MatVi; read all into MatVi;
use ro_1; read all into ro_1;
J7=J(7,1,1);
J77=J(7,7,1);
dg1_varu= 1/ro_1 + ( t(varv*J7 +
varu*Gama7)*inv(MatVi)*Gama - 2*t(Gama7)
)*inv(MatVi)*(varv*J7 + varu*Gama7);
dg1_varv= 1 + ( t(varv*J7 +
varu*Gama7)*inv(MatVi)*J77 - 2*t(J7)
)*inv(MatVi)*(varv*J7 + varu*Gama7);
dg1_7=dg1_varu//dg1_varv;
create dg1_7 from dg1_7; append from dg1_7;
quit;

/***************************************/
/****** Cálculo do L(delta) ******/
/***************************************/

/* O Cálculo do L(delta) tem que ser feito
individualmente para cada trimestre. Note-se
que o vector
coluna "gama" varia em função do trimeste e
que a matriz "Vi" é constante para diferentes
domínios. */

/* Trabalho sobre a matriz L(delta)  - cálculo
de 7 matrizes: L1, L2, L3, L4, L5, L6 e L7 */
/* 1º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama1; read all into Gama1;
use MatVi; read all into MatVi;
J7=J(7,1,1);
J77=J(7,7,1);
db1_varu= inv(MatVi)*( Gama1 -
Gama*inv(MatVi)*(varv*J7 + varu*Gama1) );
db1_varv= inv(MatVi)*( J7 -
J77*inv(MatVi)*(varv*J7 + varu*Gama1) );
L1=t(db1_varu)//t(db1_varv);
create L1 from L1; append from L1; quit;

/* 2º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama2; read all into Gama2;
use MatVi; read all into MatVi;
J7=J(7,1,1);
J77=J(7,7,1);
db1_varu= inv(MatVi)*( Gama2 -
Gama*inv(MatVi)*(varv*J7 + varu*Gama2) );
db1_varv= inv(MatVi)*( J7 -
J77*inv(MatVi)*(varv*J7 + varu*Gama2) );
L2=t(db1_varu)//t(db1_varv);
create L2 from L2; append from L2; quit;

/* 3º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
```

```
use Gama; read all into Gama;
use Gama3; read all into Gama3;
use MatVi; read all into MatVi;
J7=J(7,1,1);
J77=J(7,7,1);
db1_varu= inv(MatVi)*( Gama3 -
Gama*inv(MatVi)*(varv*J7 + varu*Gama3) );
db1_varv= inv(MatVi)*( J7 -
J77*inv(MatVi)*(varv*J7 + varu*Gama3) );
L3=t(db1_varu)//t(db1_varv);
create L3 from L3; append from L3; quit;

/* 4º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama4; read all into Gama4;
use MatVi; read all into MatVi;
J7=J(7,1,1);
J77=J(7,7,1);
db1_varu= inv(MatVi)*( Gama4 -
Gama*inv(MatVi)*(varv*J7 + varu*Gama4) );
db1_varv= inv(MatVi)*( J7 -
J77*inv(MatVi)*(varv*J7 + varu*Gama4) );
L4=t(db1_varu)//t(db1_varv);
create L4 from L4; append from L4; quit;

/* 5º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama5; read all into Gama5;
use MatVi; read all into MatVi;
J7=J(7,1,1);
J77=J(7,7,1);
db1_varu= inv(MatVi)*( Gama5 -
Gama*inv(MatVi)*(varv*J7 + varu*Gama5) );
db1_varv= inv(MatVi)*( J7 -
J77*inv(MatVi)*(varv*J7 + varu*Gama5) );
L5=t(db1_varu)//t(db1_varv);
create L5 from L5; append from L5; quit;

/* 6º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama6; read all into Gama6;
use MatVi; read all into MatVi;
J7=J(7,1,1);
J77=J(7,7,1);
db1_varu= inv(MatVi)*( Gama6 -
Gama*inv(MatVi)*(varv*J7 + varu*Gama6) );
db1_varv= inv(MatVi)*( J7 -
J77*inv(MatVi)*(varv*J7 + varu*Gama6) );
L6=t(db1_varu)//t(db1_varv);
create L6 from L6; append from L6; quit;

/* 7º trimestre */

Proc iml;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Gama7; read all into Gama7;
use MatVi; read all into MatVi;
J7=J(7,1,1);
J77=J(7,7,1);
db1_varu= inv(MatVi)*( Gama7 -
Gama*inv(MatVi)*(varv*J7 + varu*Gama7) );
db1_varv= inv(MatVi)*( J7 -
J77*inv(MatVi)*(varv*J7 + varu*Gama7) );
L7=t(db1_varu)//t(db1_varv);
create L7 from L7; append from L7; quit;

/*** Confirmação ***/

/*Proc iml;
use L1; read all into L1;
use MatVi; read all into MatVi;
A11111 = L1*MatVi*t(L1);
create A11111 from A11111; append from A11111;
quit;*/

/**** Este código serve para verificar se a
matriz A1 é igual à calculada através dos
elementos a11 , a22 e a12 dados nas pag. 519
de Rao-Yu.
OBSERVAÇÃO: Confirmei e está correcto ****/

/*******************************************/
/***** Cálculo dos ponderadores Wu *********/
/*******************************************

/*** Wu = 1 - X'u (X'X)-1 Xu ***/
/*** O vector Wu tem dimensão 7*1 porque ao
eliminar um domínio
estou a eliminar 7 observações ***/

Proc iml;
use X; read all into X;
M=inv(t(X)*X);
create M from M; append from M; quit;

Proc iml;
use M; read all into M;
use Xd1; read all into Xd1;
J7=J(7,1,1);
w1=J7 - vecdiag(Xd1*M*t(Xd1));
create w1 from w1; append from w1; quit;
Proc iml;
use M; read all into M;
use Xd2; read all into Xd2;
J7=J(7,1,1);
w2=J7 - vecdiag(Xd2*M*t(Xd2));
create w2 from w2; append from w2; quit;
Proc iml;
use M; read all into M;
use Xd3; read all into Xd3;
J7=J(7,1,1);
w3=J7 - vecdiag(Xd3*M*t(Xd3));
create w3 from w3; append from w3; quit;
Proc iml;
use M; read all into M;
use Xd4; read all into Xd4;
J7=J(7,1,1);
w4=J7 - vecdiag(Xd4*M*t(Xd4));
create w4 from w4; append from w4; quit;
Proc iml;
use M; read all into M;
use Xd5; read all into Xd5;
J7=J(7,1,1);
w5=J7 - vecdiag(Xd5*M*t(Xd5));
create w5 from w5; append from w5; quit;
Proc iml;
use M; read all into M;
use Xd6; read all into Xd6;
J7=J(7,1,1);
w6=J7 - vecdiag(Xd6*M*t(Xd6));
create w6 from w6; append from w6; quit;
Proc iml;
use M; read all into M;
use Xd7; read all into Xd7;
J7=J(7,1,1);
w7=J7 - vecdiag(Xd7*M*t(Xd7));
create w7 from w7; append from w7; quit;
Proc iml;
use M; read all into M;
use Xd8; read all into Xd8;
J7=J(7,1,1);
w8=J7 - vecdiag(Xd8*M*t(Xd8));
create w8 from w8; append from w8; quit;
Proc iml;
use M; read all into M;
use Xd9; read all into Xd9;
J7=J(7,1,1);
w9=J7 - vecdiag(Xd9*M*t(Xd9));
create w9 from w9; append from w9; quit;
Proc iml;
use M; read all into M;
use Xd10; read all into Xd10;
J7=J(7,1,1);
w10=J7 - vecdiag(Xd10*M*t(Xd10));
create w10 from w10; append from w10; quit;
Proc iml;
use M; read all into M;
use Xd11; read all into Xd11;
J7=J(7,1,1);
w11=J7 - vecdiag(Xd11*M*t(Xd11));
create w11 from w11; append from w11; quit;
Proc iml;
```

```
use M; read all into M;
use Xd12; read all into Xd12;
J7=J(7,1,1);
w12=J7 - vecdiag(Xd12*M*t(Xd12));
create w12 from w12; append from w12; quit;
Proc iml;
use M; read all into M;
use Xd13; read all into Xd13;
J7=J(7,1,1);
w13=J7 - vecdiag(Xd13*M*t(Xd13));
create w13 from w13; append from w13; quit;
Proc iml;
use M; read all into M;
use Xd14; read all into Xd14;
J7=J(7,1,1);
w14=J7 - vecdiag(Xd14*M*t(Xd14));
create w14 from w14; append from w14; quit;
Proc iml;
use M; read all into M;
use Xd15; read all into Xd15;
J7=J(7,1,1);
w15=J7 - vecdiag(Xd15*M*t(Xd15));
create w15 from w15; append from w15; quit;
Proc iml;
use M; read all into M;
use Xd16; read all into Xd16;
J7=J(7,1,1);
w16=J7 - vecdiag(Xd16*M*t(Xd16));
create w16 from w16; append from w16; quit;
Proc iml;
use M; read all into M;
use Xd17; read all into Xd17;
J7=J(7,1,1);
w17=J7 - vecdiag(Xd17*M*t(Xd17));
create w17 from w17; append from w17; quit;
Proc iml;
use M; read all into M;
use Xd18; read all into Xd18;
J7=J(7,1,1);
w18=J7 - vecdiag(Xd18*M*t(Xd18));
create w18 from w18; append from w18; quit;
Proc iml;
use M; read all into M;
use Xd19; read all into Xd19;
J7=J(7,1,1);
w19=J7 - vecdiag(Xd19*M*t(Xd19));
create w19 from w19; append from w19; quit;
Proc iml;
use M; read all into M;
use Xd20; read all into Xd20;
J7=J(7,1,1);
w20=J7 - vecdiag(Xd20*M*t(Xd20));
create w20 from w20; append from w20; quit;
Proc iml;
use M; read all into M;
use Xd21; read all into Xd21;
J7=J(7,1,1);
w21=J7 - vecdiag(Xd21*M*t(Xd21));
create w21 from w21; append from w21; quit;
Proc iml;
use M; read all into M;
use Xd22; read all into Xd22;
J7=J(7,1,1);
w22=J7 - vecdiag(Xd22*M*t(Xd22));
create w22 from w22; append from w22; quit;
Proc iml;
use M; read all into M;
use Xd23; read all into Xd23;
J7=J(7,1,1);
w23=J7 - vecdiag(Xd23*M*t(Xd23));
create w23 from w23; append from w23; quit;
Proc iml;
use M; read all into M;
use Xd24; read all into Xd24;
J7=J(7,1,1);
w24=J7 - vecdiag(Xd24*M*t(Xd24));
create w24 from w24; append from w24; quit;
Proc iml;
use M; read all into M;
use Xd25; read all into Xd25;
J7=J(7,1,1);
w25=J7 - vecdiag(Xd25*M*t(Xd25));
create w25 from w25; append from w25; quit;
Proc iml;
use M; read all into M;
use Xd26; read all into Xd26;
J7=J(7,1,1);
w26=J7 - vecdiag(Xd26*M*t(Xd26));
create w26 from w26; append from w26; quit;
Proc iml;
use M; read all into M;
use Xd27; read all into Xd27;
J7=J(7,1,1);
w27=J7 - vecdiag(Xd27*M*t(Xd27));
create w27 from w27; append from w27; quit;
Proc iml;
use M; read all into M;
use Xd28; read all into Xd28;
J7=J(7,1,1);
w28=J7 - vecdiag(Xd28*M*t(Xd28));
create w28 from w28; append from w28; quit;

/** Até aqui calculei as seguintes matrizes
necessárias para o jackknife: Dg1_1, Dg1_2,
Dg1_3, Dg1_4, Dg1_5, Dg1_6, Dg1_7; L1, L2, L3,
L4, L5, L6 e L7; MatVi; Crono_resf (que contém
os resid=Y-X*Beta); w1, w2, ..., w28  **/

/*** Cálculo das estimativas das componentes
de variância excluindo observações -  são
excluídas de cada vez todas as 7 observações
de um dado domínio ***/

Data null;
file 'C:\Documents and Settings\Politriudo\Os
meus documentos\Luís\SAS code\chamamacro.txt';
   do l= 1 to 28; k=l;
var='%ext_am('||l||','||k||');'; put var; end;
run;

%macro ext_am(l,k);

Data a61;
set a6;
rename i=j;
run;

Data k2;
do i=1 to 27;
        do v=1 to 7;
        p=1;
        output;
end;
end; keep i; run;

Proc sql;
delete from a61 where j=&l;   /* Eliminação
dos dados de um domínio j */

Data a62;
merge k2 a61;
run;

Data iabh;
set a62;
keep i t c x;
run;

Data ipth;
set a62;
keep i t y;
run;

Data iabhipth1;
set a62;
keep i t y x;
run;

/** Preparação do ficheiro das observações da
variável de interesse **/

Proc sort data=ipth;
by i t;
run;

Data tetad1; set ipth; where i= 1; keep y;
Data tetad2; set ipth; where i= 2; keep y;
Data tetad3; set ipth; where i= 3; keep y;
Data tetad4; set ipth; where i= 4; keep y;
Data tetad5; set ipth; where i= 5; keep y;
Data tetad6; set ipth; where i= 6; keep y;
Data tetad7; set ipth; where i= 7; keep y;
Data tetad8; set ipth; where i= 8; keep y;
Data tetad9; set ipth; where i= 9; keep y;
Data tetad10; set ipth; where i= 10; keep y;
Data tetad11; set ipth; where i= 11; keep y;
```

```
Data tetad12; set ipth; where i= 12; keep y;
Data tetad13; set ipth; where i= 13; keep y;
Data tetad14; set ipth; where i= 14; keep y;
Data tetad15; set ipth; where i= 15; keep y;
Data tetad16; set ipth; where i= 16; keep y;
Data tetad17; set ipth; where i= 17; keep y;
Data tetad18; set ipth; where i= 18; keep y;
Data tetad19; set ipth; where i= 19; keep y;
Data tetad20; set ipth; where i= 20; keep y;
Data tetad21; set ipth; where i= 21; keep y;
Data tetad22; set ipth; where i= 22; keep y;
Data tetad23; set ipth; where i= 23; keep y;
Data tetad24; set ipth; where i= 24; keep y;
Data tetad25; set ipth; where i= 25; keep y;
Data tetad26; set ipth; where i= 26; keep y;
Data tetad27; set ipth; where i= 27; keep y;
run;
/** Preparação do ficheiro das observações da
variável auxiliar **/

Proc sort data=iabh;
by i t;
run;

Proc iml;
j=J(189,1);
create j from j; append from j; quit;

Data iabh0;
merge j iabh;
run;

Data xd1; set iabh0; where i= 1; keep c x;
Data xd2; set iabh0; where i= 2; keep c x;
Data xd3; set iabh0; where i= 3; keep c x;
Data xd4; set iabh0; where i= 4; keep c x;
Data xd5; set iabh0; where i= 5; keep c x;
Data xd6; set iabh0; where i= 6; keep c x;
Data xd7; set iabh0; where i= 7; keep c x;
Data xd8; set iabh0; where i= 8; keep c x;
Data xd9; set iabh0; where i= 9; keep c x;
Data xd10; set iabh0; where i= 10; keep c x;
Data xd11; set iabh0; where i= 11; keep c x;
Data xd12; set iabh0; where i= 12; keep c x;
Data xd13; set iabh0; where i= 13; keep c x;
Data xd14; set iabh0; where i= 14; keep c x;
Data xd15; set iabh0; where i= 15; keep c x;
Data xd16; set iabh0; where i= 16; keep c x;
Data xd17; set iabh0; where i= 17; keep c x;
Data xd18; set iabh0; where i= 18; keep c x;
Data xd19; set iabh0; where i= 19; keep c x;
Data xd20; set iabh0; where i= 20; keep c x;
Data xd21; set iabh0; where i= 21; keep c x;
Data xd22; set iabh0; where i= 22; keep c x;
Data xd23; set iabh0; where i= 23; keep c x;
Data xd24; set iabh0; where i= 24; keep c x;
Data xd25; set iabh0; where i= 25; keep c x;
Data xd26; set iabh0; where i= 26; keep c x;
Data xd27; set iabh0; where i= 27; keep c x;
run;

/** Cálculo da matriz Z(1) **/

Proc iml;
use tetad1; read all into tetad1;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd1=(Ident - D)*P*tetad1;
create zd1 from zd1; append from zd1; quit;
Proc iml;
use tetad2; read all into tetad2;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd2=(Ident - D)*P*tetad2;
create zd2 from zd2; append from zd2; quit;
Proc iml;
use tetad3; read all into tetad3;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd3=(Ident - D)*P*tetad3;
create zd3 from zd3; append from zd3; quit;
Proc iml;
use tetad4; read all into tetad4;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd4=(Ident - D)*P*tetad4;
create zd4 from zd4; append from zd4; quit;
Proc iml;
use tetad5; read all into tetad5;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd5=(Ident - D)*P*tetad5;
create zd5 from zd5; append from zd5; quit;
Proc iml;
use tetad6; read all into tetad6;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd6=(Ident - D)*P*tetad6;
create zd6 from zd6; append from zd6; quit;
Proc iml;
use tetad7; read all into tetad7;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd7=(Ident - D)*P*tetad7;
create zd7 from zd7; append from zd7; quit;
Proc iml;
use tetad8; read all into tetad8;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd8=(Ident - D)*P*tetad8;
create zd8 from zd8; append from zd8; quit;
Proc iml;
use tetad9; read all into tetad9;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd9=(Ident - D)*P*tetad9;
create zd9 from zd9; append from zd9; quit;
Proc iml;
use tetad10; read all into tetad10;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd10=(Ident - D)*P*tetad10;
create zd10 from zd10; append from zd10; quit;
Proc iml;
use tetad11; read all into tetad11;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd11=(Ident - D)*P*tetad11;
create zd11 from zd11; append from zd11; quit;
Proc iml;
use tetad12; read all into tetad12;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd12=(Ident - D)*P*tetad12;
create zd12 from zd12; append from zd12; quit;
Proc iml;
use tetad13; read all into tetad13;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd13=(Ident - D)*P*tetad13;
create zd13 from zd13; append from zd13; quit;
Proc iml;
use tetad14; read all into tetad14;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd14=(Ident - D)*P*tetad14;
create zd14 from zd14; append from zd14; quit;
Proc iml;
use tetad15; read all into tetad15;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd15=(Ident - D)*P*tetad15;
create zd15 from zd15; append from zd15; quit;
Proc iml;
use tetad16; read all into tetad16;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd16=(Ident - D)*P*tetad16;
create zd16 from zd16; append from zd16; quit;
Proc iml;
```

```
use tetad17; read all into tetad17;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd17=(Ident - D)*P*tetad17;
create zd17 from zd17; append from zd17; quit;
Proc iml;
use tetad18; read all into tetad18;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd18=(Ident - D)*P*tetad18;
create zd18 from zd18; append from zd18; quit;
Proc iml;
use tetad19; read all into tetad19;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd19=(Ident - D)*P*tetad19;
create zd19 from zd19; append from zd19; quit;
Proc iml;
use tetad20; read all into tetad20;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd20=(Ident - D)*P*tetad20;
create zd20 from zd20; append from zd20; quit;
Proc iml;
use tetad21; read all into tetad21;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd21=(Ident - D)*P*tetad21;
create zd21 from zd21; append from zd21; quit;
Proc iml;
use tetad22; read all into tetad22;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd22=(Ident - D)*P*tetad22;
create zd22 from zd22; append from zd22; quit;
Proc iml;
use tetad23; read all into tetad23;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd23=(Ident - D)*P*tetad23;
create zd23 from zd23; append from zd23; quit;
Proc iml;
use tetad24; read all into tetad24;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd24=(Ident - D)*P*tetad24;
create zd24 from zd24; append from zd24; quit;
Proc iml;
use tetad25; read all into tetad25;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd25=(Ident - D)*P*tetad25;
create zd25 from zd25; append from zd25; quit;
Proc iml;
use tetad26; read all into tetad26;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd26=(Ident - D)*P*tetad26;
create zd26 from zd26; append from zd26; quit;
Proc iml;
use tetad27; read all into tetad27;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd27=(Ident - D)*P*tetad27;
create zd27 from zd27; append from zd27; quit;

Data Z1;
set zd1 zd2 zd3 zd4 zd5 zd6 zd7 zd8 zd9 zd10
zd11 zd12 zd13 zd14 zd15
zd16 zd17 zd18 zd19 zd20 zd21 zd22 zd23 zd24
zd25 zd26 zd27;
run;

/** Cálculo da matriz H(1) **/

Proc iml;
use xd1; read all into xd1;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd1=(Ident - D)*P*xd1;
create hd1 from hd1; append from hd1; quit;
Proc iml;
use xd2; read all into xd2;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd2=(Ident - D)*P*xd2;
create hd2 from hd2; append from hd2; quit;
Proc iml;
use xd3; read all into xd3;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd3=(Ident - D)*P*xd3;
create hd3 from hd3; append from hd3; quit;
Proc iml;
use xd4; read all into xd4;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd4=(Ident - D)*P*xd4;
create hd4 from hd4; append from hd4; quit;
Proc iml;
use xd5; read all into xd5;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd5=(Ident - D)*P*xd5;
create hd5 from hd5; append from hd5; quit;
Proc iml;
use xd6; read all into xd6;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd6=(Ident - D)*P*xd6;
create hd6 from hd6; append from hd6; quit;
Proc iml;
use xd7; read all into xd7;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd7=(Ident - D)*P*xd7;
create hd7 from hd7; append from hd7; quit;
Proc iml;
use xd8; read all into xd8;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd8=(Ident - D)*P*xd8;
create hd8 from hd8; append from hd8; quit;
Proc iml;
use xd9; read all into xd9;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd9=(Ident - D)*P*xd9;
create hd9 from hd9; append from hd9; quit;
Proc iml;
use xd10; read all into xd10;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd10=(Ident - D)*P*xd10;
create hd10 from hd10; append from hd10; quit;
Proc iml;
use xd11; read all into xd11;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd11=(Ident - D)*P*xd11;
create hd11 from hd11; append from hd11; quit;
Proc iml;
use xd12; read all into xd12;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd12=(Ident - D)*P*xd12;
create hd12 from hd12; append from hd12; quit;
Proc iml;
use xd13; read all into xd13;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd13=(Ident - D)*P*xd13;
```

```
create hd13 from hd13; append from hd13; quit;
Proc iml;
use xd14; read all into xd14;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd14=(Ident - D)*P*xd14;
create hd14 from hd14; append from hd14; quit;
Proc iml;
use xd15; read all into xd15;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd15=(Ident - D)*P*xd15;
create hd15 from hd15; append from hd15; quit;
Proc iml;
use xd16; read all into xd16;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd16=(Ident - D)*P*xd16;
create hd16 from hd16; append from hd16; quit;
Proc iml;
use xd17; read all into xd17;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd17=(Ident - D)*P*xd17;
create hd17 from hd17; append from hd17; quit;
Proc iml;
use xd18; read all into xd18;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd18=(Ident - D)*P*xd18;
create hd18 from hd18; append from hd18; quit;
Proc iml;
use xd19; read all into xd19;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd19=(Ident - D)*P*xd19;
create hd19 from hd19; append from hd19; quit;
Proc iml;
use xd20; read all into xd20;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd20=(Ident - D)*P*xd20;
create hd20 from hd20; append from hd20; quit;
Proc iml;
use xd21; read all into xd21;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd21=(Ident - D)*P*xd21;
create hd21 from hd21; append from hd21; quit;
Proc iml;
use xd22; read all into xd22;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd22=(Ident - D)*P*xd22;
create hd22 from hd22; append from hd22; quit;
Proc iml;
use xd23; read all into xd23;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd23=(Ident - D)*P*xd23;
create hd23 from hd23; append from hd23; quit;
Proc iml;
use xd24; read all into xd24;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd24=(Ident - D)*P*xd24;
create hd24 from hd24; append from hd24; quit;
Proc iml;
use xd25; read all into xd25;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd25=(Ident - D)*P*xd25;
create hd25 from hd25; append from hd25; quit;
Proc iml;
use xd26; read all into xd26;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd26=(Ident - D)*P*xd26;
create hd26 from hd26; append from hd26; quit;
Proc iml;
use xd27; read all into xd27;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd27=(Ident - D)*P*xd27;
create hd27 from hd27; append from hd27; quit;

Data H1;
set hd1 hd2 hd3 hd4 hd5 hd6 hd7 hd8 hd9 hd10
hd11 hd12 hd13 hd14 hd15
hd16 hd17 hd18 hd19 hd20 hd21 hd22 hd23 hd24
hd25 hd26 hd27;
run;

/** Regressão de Z(1) sobre H(1) **/

Data Z1;
set Z1;
rename col1=col0;
run;

Data reg;
merge Z1 H1;
run;

proc reg data=reg noprint;
        model col0=col1 col2 / noint;
        output out=rg1  residual=r1;
run; quit;

Data rg1;
set rg1;
qr1=r1*r1;
Keep qr1;
run;

Proc means data=rg1 sum noprint;
var qr1;
output out=rg1_ sum=sqr1;
run;

Data rg1_;
set rg1_;
keep sqr1;
run;

/** Cálculo da variância sigma2_u **/

Proc iml;
use H1; read all into H1;
Hs1=H1*ginv(t(H1)*H1)*t(H1);
create Hs1 from Hs1; append from Hs1; quit;

Proc iml;
use D; read all into D;
Ident=I(7);
ID=Ident - D; /* Matriz I - D */
create ID from ID; append from ID; quit;

Proc iml;
use ID; read all into ID;
blockID1=BLOCK(ID,ID,ID,ID,ID,ID,ID,ID,ID,ID,I
D,ID,ID,ID);
create blockID1 from blockID1; append from
blockID1; quit;
Proc iml;
use ID; read all into ID;
blockID2=BLOCK(ID,ID,ID,ID,ID,ID,ID,ID,ID,ID,I
D,ID,ID);
create blockID2 from blockID2; append from
blockID2; quit;
Proc iml;
use blockID1; read all into blockID1;
use blockID2; read all into blockID2;
blockID=BLOCK(blockID1,blockID2);
create blockID from blockID; append from
blockID; quit;

Proc iml;
use P; read all into P;
P1=P*t(P); /* Admite-se que SIGMA_d = I */
create P1 from P1; append from P1; quit;
```

```
Proc iml;
use P1; read all into P1;
blockPP1=BLOCK(P1,P1,P1,P1,P1,P1,P1,P1,P1,P1,P
1,P1,P1,P1);
create blockPP1 from blockPP1; append from
blockPP1; quit;
Proc iml;
use P1; read all into P1;
blockPP2=BLOCK(P1,P1,P1,P1,P1,P1,P1,P1,P1,P1,P
1,P1,P1);
create blockPP2 from blockPP2; append from
blockPP2; quit;
Proc iml;
use blockPP1; read all into blockPP1;
use blockPP2; read all into blockPP2;
blockPP=BLOCK(blockPP1,blockPP2);
create blockPP from blockPP; append from
blockPP; quit;

Proc iml;
use H1; read all into H1;
HH=H1*t(H1);
eval=eigval(HH);
create eval from eval; append from eval; quit;

Data id;
do id=1 to 189;
output; end;
run;

Data eval;
merge id eval;
run;

Data new;
set eval;
by id;
retain count 0;
if first.id then count=0;
if col1 > 0.000001 then count = count + 1;
run;

Proc means data=new sum noprint;
var count;
output out=caracH1 sum=caracH1;
run;

Data caracH1;
set caracH1;
keep caracH1;
run;

Proc iml;
use blockPP; read all into blockPP;
use blockID; read all into blockID;
use Hs1; read all into Hs1;
use rg1_; read all into rg1_;
use caracH1; read all into caracH1;
varu= (rg1_ -  trace((blockID -
Hs1)*blockPP))*(1/(27*6 - caracH1));
create varu from varu; append from varu; quit;

/* Termina aqui o cálculo da variância
sigma2_u */

/** Cálculo da matriz: c^(-0.5)*f'*P*Teta -
var. dependente do modelo transformado **/

Proc iml;
use tetad1; read all into tetad1;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y1=(1/sqrt(c))*t(f)*P*tetad1;
create y1 from y1; append from y1; quit;
Proc iml;
use tetad2; read all into tetad2;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y2=(1/sqrt(c))*t(f)*P*tetad2;
create y2 from y2; append from y2; quit;
Proc iml;
use tetad3; read all into tetad3;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y3=(1/sqrt(c))*t(f)*P*tetad3;
create y3 from y3; append from y3; quit;
Proc iml;
use tetad4; read all into tetad4;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y4=(1/sqrt(c))*t(f)*P*tetad4;
create y4 from y4; append from y4; quit;
Proc iml;
use tetad5; read all into tetad5;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y5=(1/sqrt(c))*t(f)*P*tetad5;
create y5 from y5; append from y5; quit;
Proc iml;
use tetad6; read all into tetad6;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y6=(1/sqrt(c))*t(f)*P*tetad6;
create y6 from y6; append from y6; quit;
Proc iml;
use tetad7; read all into tetad7;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y7=(1/sqrt(c))*t(f)*P*tetad7;
create y7 from y7; append from y7; quit;
Proc iml;
use tetad8; read all into tetad8;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y8=(1/sqrt(c))*t(f)*P*tetad8;
create y8 from y8; append from y8; quit;
Proc iml;
use tetad9; read all into tetad9;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y9=(1/sqrt(c))*t(f)*P*tetad9;
create y9 from y9; append from y9; quit;
Proc iml;
use tetad10; read all into tetad10;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y10=(1/sqrt(c))*t(f)*P*tetad10;
create y10 from y10; append from y10; quit;
Proc iml;
use tetad11; read all into tetad11;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y11=(1/sqrt(c))*t(f)*P*tetad11;
create y11 from y11; append from y11; quit;
Proc iml;
use tetad12; read all into tetad12;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y12=(1/sqrt(c))*t(f)*P*tetad12;
create y12 from y12; append from y12; quit;
Proc iml;
use tetad13; read all into tetad13;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y13=(1/sqrt(c))*t(f)*P*tetad13;
create y13 from y13; append from y13; quit;
Proc iml;
use tetad14; read all into tetad14;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y14=(1/sqrt(c))*t(f)*P*tetad14;
create y14 from y14; append from y14; quit;
Proc iml;
use tetad15; read all into tetad15;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y15=(1/sqrt(c))*t(f)*P*tetad15;
create y15 from y15; append from y15; quit;
Proc iml;
use tetad16; read all into tetad16;
use P; read all into P;
```

```
use f; read all into f;
use c; read all into c;
y16=(1/sqrt(c))*t(f)*P*tetad16;
create y16 from y16; append from y16; quit;
Proc iml;
use tetad17; read all into tetad17;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y17=(1/sqrt(c))*t(f)*P*tetad17;
create y17 from y17; append from y17; quit;
Proc iml;
use tetad18; read all into tetad18;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y18=(1/sqrt(c))*t(f)*P*tetad18;
create y18 from y18; append from y18; quit;
Proc iml;
use tetad19; read all into tetad19;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y19=(1/sqrt(c))*t(f)*P*tetad19;
create y19 from y19; append from y19; quit;
Proc iml;
use tetad20; read all into tetad20;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y20=(1/sqrt(c))*t(f)*P*tetad20;
create y20 from y20; append from y20; quit;
Proc iml;
use tetad21; read all into tetad21;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y21=(1/sqrt(c))*t(f)*P*tetad21;
create y21 from y21; append from y21; quit;
Proc iml;
use tetad22; read all into tetad22;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y22=(1/sqrt(c))*t(f)*P*tetad22;
create y22 from y22; append from y22; quit;
Proc iml;
use tetad23; read all into tetad23;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y23=(1/sqrt(c))*t(f)*P*tetad23;
create y23 from y23; append from y23; quit;
Proc iml;
use tetad24; read all into tetad24;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y24=(1/sqrt(c))*t(f)*P*tetad24;
create y24 from y24; append from y24; quit;
Proc iml;
use tetad25; read all into tetad25;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y25=(1/sqrt(c))*t(f)*P*tetad25;
create y25 from y25; append from y25; quit;
Proc iml;
use tetad26; read all into tetad26;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y26=(1/sqrt(c))*t(f)*P*tetad26;
create y26 from y26; append from y26; quit;
Proc iml;
use tetad27; read all into tetad27;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y27=(1/sqrt(c))*t(f)*P*tetad27;
create y27 from y27; append from y27; quit;

Data Y;
set y1 y2 y3 y4 y5 y6 y7 y8 y9 y10 y11 y12 y13
y14 y15 y16 y17 y18 y19 y20 y21 y22 y23 y24
y25 y26 y27;
run;

/** Cálculo da matriz: c^(-0.5)*f'*P*X - var.
independentes do modelo transformado **/

Proc iml;
use xd1; read all into xd1;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x1=(1/sqrt(c))*t(f)*P*xd1;
create x1 from x1; append from x1; quit;
Proc iml;
use xd2; read all into xd2;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x2=(1/sqrt(c))*t(f)*P*xd2;
create x2 from x2; append from x2; quit;
Proc iml;
use xd3; read all into xd3;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x3=(1/sqrt(c))*t(f)*P*xd3;
create x3 from x3; append from x3; quit;
Proc iml;
use xd4; read all into xd4;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x4=(1/sqrt(c))*t(f)*P*xd4;
create x4 from x4; append from x4; quit;
Proc iml;
use xd5; read all into xd5;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x5=(1/sqrt(c))*t(f)*P*xd5;
create x5 from x5; append from x5; quit;
Proc iml;
use xd6; read all into xd6;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x6=(1/sqrt(c))*t(f)*P*xd6;
create x6 from x6; append from x6; quit;
Proc iml;
use xd7; read all into xd7;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x7=(1/sqrt(c))*t(f)*P*xd7;
create x7 from x7; append from x7; quit;
Proc iml;
use xd8; read all into xd8;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x8=(1/sqrt(c))*t(f)*P*xd8;
create x8 from x8; append from x8; quit;
Proc iml;
use xd9; read all into xd9;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x9=(1/sqrt(c))*t(f)*P*xd9;
create x9 from x9; append from x9; quit;
Proc iml;
use xd10; read all into xd10;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x10=(1/sqrt(c))*t(f)*P*xd10;
create x10 from x10; append from x10; quit;
Proc iml;
use xd11; read all into xd11;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x11=(1/sqrt(c))*t(f)*P*xd11;
create x11 from x11; append from x11; quit;
Proc iml;
use xd12; read all into xd12;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x12=(1/sqrt(c))*t(f)*P*xd12;
create x12 from x12; append from x12; quit;
Proc iml;
```

```
use xd13; read all into xd13;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x13=(1/sqrt(c))*t(f)*P*xd13;
create x13 from x13; append from x13; quit;
Proc iml;
use xd14; read all into xd14;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x14=(1/sqrt(c))*t(f)*P*xd14;
create x14 from x14; append from x14; quit;
Proc iml;
use xd15; read all into xd15;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x15=(1/sqrt(c))*t(f)*P*xd15;
create x15 from x15; append from x15; quit;
Proc iml;
use xd16; read all into xd16;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x16=(1/sqrt(c))*t(f)*P*xd16;
create x16 from x16; append from x16; quit;
Proc iml;
use xd17; read all into xd17;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x17=(1/sqrt(c))*t(f)*P*xd17;
create x17 from x17; append from x17; quit;
Proc iml;
use xd18; read all into xd18;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x18=(1/sqrt(c))*t(f)*P*xd18;
create x18 from x18; append from x18; quit;
Proc iml;
use xd19; read all into xd19;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x19=(1/sqrt(c))*t(f)*P*xd19;
create x19 from x19; append from x19; quit;
Proc iml;
use xd20; read all into xd20;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x20=(1/sqrt(c))*t(f)*P*xd20;
create x20 from x20; append from x20; quit;
Proc iml;
use xd21; read all into xd21;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x21=(1/sqrt(c))*t(f)*P*xd21;
create x21 from x21; append from x21; quit;
Proc iml;
use xd22; read all into xd22;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x22=(1/sqrt(c))*t(f)*P*xd22;
create x22 from x22; append from x22; quit;
Proc iml;
use xd23; read all into xd23;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x23=(1/sqrt(c))*t(f)*P*xd23;
create x23 from x23; append from x23; quit;
Proc iml;
use xd24; read all into xd24;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x24=(1/sqrt(c))*t(f)*P*xd24;
create x24 from x24; append from x24; quit;
Proc iml;
use xd25; read all into xd25;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x25=(1/sqrt(c))*t(f)*P*xd25;
create x25 from x25; append from x25; quit;
Proc iml;
use xd26; read all into xd26;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x26=(1/sqrt(c))*t(f)*P*xd26;
create x26 from x26; append from x26; quit;
Proc iml;
use xd27; read all into xd27;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x27=(1/sqrt(c))*t(f)*P*xd27;
create x27 from x27; append from x27; quit;

Data X;
set x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13
x14 x15 x16 x17 x18 x19 x20 x21 x22 x23 x24
x25 x26 x27;
run;

/** Regressão de Y sobre X **/

Data Y;
set Y;
rename col1=col0;
run;

Data reg2;
merge Y X;
run;

proc reg data=reg2 noprint;
        model col0=col1 col2 / noint;
        output out=rg2  residual=r2;
run; quit;

Data rg2;
set rg2;
qr2=r2*r2;
Keep qr2;
run;

Proc means data=rg2 sum noprint;
var qr2;
output out=rg2_ sum=sqr2;
run;

Data rg2_;
set rg2_;
keep sqr2;
run;

/** Cálculo da variância sigma2_v **/

Proc iml;
use xd1; read all into xd1;
use P; read all into P;
use f; read all into f;
xpf1=t(xd1)*t(P)*f;
create xpf1 from xpf1; append from xpf1; quit;
Proc iml;
use xd2; read all into xd2;
use P; read all into P;
use f; read all into f;
xpf2=t(xd2)*t(P)*f;
create xpf2 from xpf2; append from xpf2; quit;
Proc iml;
use xd3; read all into xd3;
use P; read all into P;
use f; read all into f;
xpf3=t(xd3)*t(P)*f;
create xpf3 from xpf3; append from xpf3; quit;
Proc iml;
use xd4; read all into xd4;
use P; read all into P;
use f; read all into f;
xpf4=t(xd4)*t(P)*f;
create xpf4 from xpf4; append from xpf4; quit;
Proc iml;
use xd5; read all into xd5;
use P; read all into P;
use f; read all into f;
xpf5=t(xd5)*t(P)*f;
create xpf5 from xpf5; append from xpf5; quit;
Proc iml;
```

```
use xd6; read all into xd6;
use P; read all into P;
use f; read all into f;
xpf6=t(xd6)*t(P)*f;
create xpf6 from xpf6; append from xpf6; quit;
Proc iml;
use xd7; read all into xd7;
use P; read all into P;
use f; read all into f;
xpf7=t(xd7)*t(P)*f;
create xpf7 from xpf7; append from xpf7; quit;
Proc iml;
use xd8; read all into xd8;
use P; read all into P;
use f; read all into f;
xpf8=t(xd8)*t(P)*f;
create xpf8 from xpf8; append from xpf8; quit;
Proc iml;
use xd9; read all into xd9;
use P; read all into P;
use f; read all into f;
xpf9=t(xd9)*t(P)*f;
create xpf9 from xpf9; append from xpf9; quit;
Proc iml;
use xd10; read all into xd10;
use P; read all into P;
use f; read all into f;
xpf10=t(xd10)*t(P)*f;
create xpf10 from xpf10; append from xpf10;
quit;
Proc iml;
use xd11; read all into xd11;
use P; read all into P;
use f; read all into f;
xpf11=t(xd11)*t(P)*f;
create xpf11 from xpf11; append from xpf11;
quit;
Proc iml;
use xd12; read all into xd12;
use P; read all into P;
use f; read all into f;
xpf12=t(xd12)*t(P)*f;
create xpf12 from xpf12; append from xpf12;
quit;
Proc iml;
use xd13; read all into xd13;
use P; read all into P;
use f; read all into f;
xpf13=t(xd13)*t(P)*f;
create xpf13 from xpf13; append from xpf13;
quit;
Proc iml;
use xd14; read all into xd14;
use P; read all into P;
use f; read all into f;
xpf14=t(xd14)*t(P)*f;
create xpf14 from xpf14; append from xpf14;
quit;
Proc iml;
use xd15; read all into xd15;
use P; read all into P;
use f; read all into f;
xpf15=t(xd15)*t(P)*f;
create xpf15 from xpf15; append from xpf15;
quit;
Proc iml;
use xd16; read all into xd16;
use P; read all into P;
use f; read all into f;
xpf16=t(xd16)*t(P)*f;
create xpf16 from xpf16; append from xpf16;
quit;
Proc iml;
use xd17; read all into xd17;
use P; read all into P;
use f; read all into f;
xpf17=t(xd17)*t(P)*f;
create xpf17 from xpf17; append from xpf17;
quit;
Proc iml;
use xd18; read all into xd18;
use P; read all into P;
use f; read all into f;
xpf18=t(xd18)*t(P)*f;
create xpf18 from xpf18; append from xpf18;
quit;
Proc iml;
use xd19; read all into xd19;
use P; read all into P;
use f; read all into f;
xpf19=t(xd19)*t(P)*f;
create xpf19 from xpf19; append from xpf19;
quit;
Proc iml;
use xd20; read all into xd20;
use P; read all into P;
use f; read all into f;
xpf20=t(xd20)*t(P)*f;
create xpf20 from xpf20; append from xpf20;
quit;
Proc iml;
use xd21; read all into xd21;
use P; read all into P;
use f; read all into f;
xpf21=t(xd21)*t(P)*f;
create xpf21 from xpf21; append from xpf21;
quit;
Proc iml;
use xd22; read all into xd22;
use P; read all into P;
use f; read all into f;
xpf22=t(xd22)*t(P)*f;
create xpf22 from xpf22; append from xpf22;
quit;
Proc iml;
use xd23; read all into xd23;
use P; read all into P;
use f; read all into f;
xpf23=t(xd23)*t(P)*f;
create xpf23 from xpf23; append from xpf23;
quit;
Proc iml;
use xd24; read all into xd24;
use P; read all into P;
use f; read all into f;
xpf24=t(xd24)*t(P)*f;
create xpf24 from xpf24; append from xpf24;
quit;
Proc iml;
use xd25; read all into xd25;
use P; read all into P;
use f; read all into f;
xpf25=t(xd25)*t(P)*f;
create xpf25 from xpf25; append from xpf25;
quit;
Proc iml;
use xd26; read all into xd26;
use P; read all into P;
use f; read all into f;
xpf26=t(xd26)*t(P)*f;
create xpf26 from xpf26; append from xpf26;
quit;
Proc iml;
use xd27; read all into xd27;
use P; read all into P;
use f; read all into f;
xpf27=t(xd27)*t(P)*f;
create xpf27 from xpf27; append from xpf27;
quit;

Proc iml;
use xpf1; read all into xpf1;
use xpf2; read all into xpf2;
use xpf3; read all into xpf3;
use xpf4; read all into xpf4;
use xpf5; read all into xpf5;
use xpf6; read all into xpf6;
use xpf7; read all into xpf7;
use xpf8; read all into xpf8;
use xpf9; read all into xpf9;
use xpf10; read all into xpf10;
use xpf11; read all into xpf11;
use xpf12; read all into xpf12;
use xpf13; read all into xpf13;
use xpf14; read all into xpf14;
use xpf15; read all into xpf15;
use xpf16; read all into xpf16;
use xpf17; read all into xpf17;
use xpf18; read all into xpf18;
use xpf19; read all into xpf19;
use xpf20; read all into xpf20;
use xpf21; read all into xpf21;
use xpf22; read all into xpf22;
use xpf23; read all into xpf23;
use xpf24; read all into xpf24;
use xpf25; read all into xpf25;
```

```
use xpf26; read all into xpf26;
use xpf27; read all into xpf27;
Ft=xpf1||xpf2||xpf3||xpf4||xpf5||xpf6||xpf7||x
pf8||xpf9||xpf10||xpf11||xpf12||xpf13||xpf14
||xpf15||xpf16||xpf17||xpf18||xpf19||xpf20||xp
f21||xpf22||xpf23||xpf24||xpf25||xpf26||xpf27;
create Ft from Ft; append from Ft; quit;

Proc iml;
use Ft; read all into Ft;
F1=t(Ft);
create F1 from F1; append from F1; quit;

Proc iml;
use F1; read all into F1;
Fast=F1*ginv(t(F1)*F1)*t(F1); /* ginv(A) is
the Moore-Penrose generalized inverse matrix
*/
create Fast from Fast; append from Fast; quit;
/* Fast é a matriz F* */

Proc iml;
use Fast; read all into Fast;
Ident=I(27);
IF=Ident - Fast; /* Matriz I - F* */
create IF from IF; append from IF; quit;

Proc iml;
use f; read all into f;
use P; read all into P;
use c; read all into c;
d1=t(f)*P*t(P)*f*(1 / c); /* Admite-se que
SIGMA_d = I */
Ident=I(27);
cfp=d1*Ident;
create cfp from cfp; append from cfp; quit;

Proc iml;
use F1; read all into F1;
FF=F1*t(F1);
eval1=eigval(FF);
create eval1 from eval1; append from eval1;
quit;

Data id1;
do id=1 to 27;
output; end;
run;

Data eval1;
merge id1 eval1;
run;

Data new1;
set eval1;
by id;
retain count 0;
if first.id then count=0;
if col1 > 0.000001 then count = count + 1;
run;

Proc means data=new1 sum noprint;
var count;
output out=caracF sum=caracF;
run;

Data caracF;
set caracF;
keep caracF;

Proc iml;
use IF; read all into IF;
use cfp; read all into cfp;
use c; read all into c;
use rg2_; read all into rg2_;
use caracF; read all into caracF;
use varu; read all into varu;
varv= (rg2_ -  trace(IF*cfp))*(1/(c*(27 -
caracF))) - varu/c;
create varv from varv; append from varv; quit;

/* Termina aqui o cálculo da variância
sigma2_v */

/* Truncagem a zero das variâncias sigma2_u e
sigma2_v */

Data varu;
set varu;
sigma2u= max( col1 , 0);
keep sigma2u;
Data varv;
set varv;
sigma2v= max( col1 , 0);
keep sigma2v;
run;

Proc iml;
use varu; read all into varu;
use varv; read all into varv;
paruv_=varu//varv;
create paruv_ from paruv_; append from paruv_;
quit;

Proc iml;
use Paruv; read all into Paruv;
use Paruv_; read all into Paruv_;
use W&l; read all into W&l;
tBwj_&l=W&l @ t(paruv_ - paruv);   /* Aqui
obtem-se a transposta de Bwj - enviesamento */
create tBwj_&l from tBwj_&l; append from
tBwj_&l; quit;

/* tBwj_&l tem dimensão 7*2 */
/* Cada linha refere-se a um trimestre */

Proc iml;
use Paruv; read all into Paruv;
use Paruv_; read all into Paruv_;
use W&l; read all into W&l;
Uwj_&l=W&l @ ((paruv_ - paruv)*t(paruv_ -
paruv));   /* Aqui obtem-se o Uwj -
covariância */
create Uwj_&l from Uwj_&l; append from Uwj_&l;
quit;

/* Uwj_&l tem dimensão 14*2 */
/* Cada submatriz 2*2 refere-se a um trimestre
*/

%mend;
%inc 'C:\Documents and Settings\Politriudo\Os
meus documentos\Luís\SAS code\chamamacro.txt';

/* Cálculo da parcela tBwj*Dg1 */

Proc iml;
use tBwj_1; read all into tBwj_1;
use tBwj_2; read all into tBwj_2;
use tBwj_3; read all into tBwj_3;
use tBwj_4; read all into tBwj_4;
use tBwj_5; read all into tBwj_5;
use tBwj_6; read all into tBwj_6;
use tBwj_7; read all into tBwj_7;
use tBwj_8; read all into tBwj_8;
use tBwj_9; read all into tBwj_9;
use tBwj_10; read all into tBwj_10;
use tBwj_11; read all into tBwj_11;
use tBwj_12; read all into tBwj_12;
use tBwj_13; read all into tBwj_13;
use tBwj_14; read all into tBwj_14;
use tBwj_15; read all into tBwj_15;
use tBwj_16; read all into tBwj_16;
use tBwj_17; read all into tBwj_17;
use tBwj_18; read all into tBwj_18;
use tBwj_19; read all into tBwj_19;
use tBwj_20; read all into tBwj_20;
use tBwj_21; read all into tBwj_21;
use tBwj_22; read all into tBwj_22;
use tBwj_23; read all into tBwj_23;
use tBwj_24; read all into tBwj_24;
use tBwj_25; read all into tBwj_25;
use tBwj_26; read all into tBwj_26;
use tBwj_27; read all into tBwj_27;
use tBwj_28; read all into tBwj_28;
tBwj= tBwj_1 + tBwj_2 + tBwj_3 + tBwj_4 +
tBwj_5 + tBwj_6 + tBwj_7 + tBwj_8 + tBwj_9 +
tBwj_10 +
tBwj_11 + tBwj_12 + tBwj_13 + tBwj_14 +
tBwj_15 + tBwj_16 + tBwj_17 + tBwj_18 +
tBwj_19 + tBwj_20 +
tBwj_21 + tBwj_22 + tBwj_23 + tBwj_24 +
tBwj_25 + tBwj_26 + tBwj_27 + tBwj_28;
create tBwj from tBwj; append from tBwj; quit;
/* Matriz 7*2 */
```

```
Proc iml;
use Dg1_1; read all into Dg1_1;
use tBwj; read all into tBwj;
tBwj_=tBwj[1,];
tBwjDg1_1=tBwj_*Dg1_1;
create tBwjDg1_1 from tBwjDg1_1;
append from tBwjDg1_1; quit;
Proc iml;
use Dg1_2; read all into Dg1_2;
use tBwj; read all into tBwj;
tBwj_=tBwj[2,];
tBwjDg1_2=tBwj_*Dg1_2;
create tBwjDg1_2 from tBwjDg1_2;
append from tBwjDg1_2; quit;
Proc iml;
use Dg1_3; read all into Dg1_3;
use tBwj; read all into tBwj;
tBwj_=tBwj[3,];
tBwjDg1_3=tBwj_*Dg1_3;
create tBwjDg1_3 from tBwjDg1_3;
append from tBwjDg1_3; quit;
Proc iml;
use Dg1_4; read all into Dg1_4;
use tBwj; read all into tBwj;
tBwj_=tBwj[4,];
tBwjDg1_4=tBwj_*Dg1_4;
create tBwjDg1_4 from tBwjDg1_4;
append from tBwjDg1_4; quit;
Proc iml;
use Dg1_5; read all into Dg1_5;
use tBwj; read all into tBwj;
tBwj_=tBwj[5,];
tBwjDg1_5=tBwj_*Dg1_5;
create tBwjDg1_5 from tBwjDg1_5;
append from tBwjDg1_5; quit;
Proc iml;
use Dg1_6; read all into Dg1_6;
use tBwj; read all into tBwj;
tBwj_=tBwj[6,];
tBwjDg1_6=tBwj_*Dg1_6;
create tBwjDg1_6 from tBwjDg1_6;
append from tBwjDg1_6; quit;
Proc iml;
use Dg1_7; read all into Dg1_7;
use tBwj; read all into tBwj;
tBwj_=tBwj[7,];
tBwjDg1_7=tBwj_*Dg1_7;
create tBwjDg1_7 from tBwjDg1_7;
append from tBwjDg1_7; quit;

Proc iml;
use tBwjDg1_1; read all into tBwjDg1_1;
use tBwjDg1_2; read all into tBwjDg1_2;
use tBwjDg1_3; read all into tBwjDg1_3;
use tBwjDg1_4; read all into tBwjDg1_4;
use tBwjDg1_5; read all into tBwjDg1_5;
use tBwjDg1_6; read all into tBwjDg1_6;
use tBwjDg1_7; read all into tBwjDg1_7;
J28=J(28,1,1);
tBwjDg_=tBwjDg1_1//tBwjDg1_2//tBwjDg1_3//tBwjD
g1_4//tBwjDg1_5//tBwjDg1_6//tBwjDg1_7;
tBwjDg1=J28 @ tBwjDg_;
create tBwjDg1 from tBwjDg1;
append from tBwjDg1; quit;

Data tBwjDg1;
set tBwjDg1;
rename col1=tBwjDg1;
run;

/* Cálculo da parcela tr[L * V * L' * Uwj] */

Proc iml;
use Uwj_1; read all into Uwj_1;
use Uwj_2; read all into Uwj_2;
use Uwj_3; read all into Uwj_3;
use Uwj_4; read all into Uwj_4;
use Uwj_5; read all into Uwj_5;
use Uwj_6; read all into Uwj_6;
use Uwj_7; read all into Uwj_7;
use Uwj_8; read all into Uwj_8;
use Uwj_9; read all into Uwj_9;
use Uwj_10; read all into Uwj_10;
use Uwj_11; read all into Uwj_11;
use Uwj_12; read all into Uwj_12;
use Uwj_13; read all into Uwj_13;
use Uwj_14; read all into Uwj_14;
use Uwj_15; read all into Uwj_15;
```

```
use Uwj_16; read all into Uwj_16;
use Uwj_17; read all into Uwj_17;
use Uwj_18; read all into Uwj_18;
use Uwj_19; read all into Uwj_19;
use Uwj_20; read all into Uwj_20;
use Uwj_21; read all into Uwj_21;
use Uwj_22; read all into Uwj_22;
use Uwj_23; read all into Uwj_23;
use Uwj_24; read all into Uwj_24;
use Uwj_25; read all into Uwj_25;
use Uwj_26; read all into Uwj_26;
use Uwj_27; read all into Uwj_27;
use Uwj_28; read all into Uwj_28;
Uwj= Uwj_1 + Uwj_2 + Uwj_3 + Uwj_4 + Uwj_5 +
Uwj_6 + Uwj_7 + Uwj_8 + Uwj_9 + Uwj_10 +
Uwj_11 + Uwj_12 + Uwj_13 + Uwj_14 + Uwj_15 +
Uwj_16 + Uwj_17 + Uwj_18 + Uwj_19 + Uwj_20 +
Uwj_21 + Uwj_22 + Uwj_23 + Uwj_24 + Uwj_25 +
Uwj_26 + Uwj_27 + Uwj_28;
create Uwj from Uwj; append from Uwj; quit;
/* Matriz 14*2 */

Proc iml;
use MatVi; read all into MatVi;
use Uwj; read all into Uwj;
use L1; read all into L1;
Uwj_=Uwj[1:2,];   /* Cria uma matriz com as
duas primeiras linhas de Uwj */
LVLUwj_1=trace( L1*MatVi*t(L1)*Uwj_ );
create LVLUwj_1 from LVLUwj_1;
append from LVLUwj_1; quit;
Proc iml;
use MatVi; read all into MatVi;
use Uwj; read all into Uwj;
use L2; read all into L2;
Uwj_=Uwj[3:4,];
LVLUwj_2=trace( L2*MatVi*t(L2)*Uwj_ );
create LVLUwj_2 from LVLUwj_2;
append from LVLUwj_2; quit;
Proc iml;
use MatVi; read all into MatVi;
use Uwj; read all into Uwj;
use L3; read all into L3;
Uwj_=Uwj[5:6,];
LVLUwj_3=trace( L3*MatVi*t(L3)*Uwj_ );
create LVLUwj_3 from LVLUwj_3;
append from LVLUwj_3; quit;
Proc iml;
use MatVi; read all into MatVi;
use Uwj; read all into Uwj;
use L4; read all into L4;
Uwj_=Uwj[7:8,];
LVLUwj_4=trace( L4*MatVi*t(L4)*Uwj_ );
create LVLUwj_4 from LVLUwj_4;
append from LVLUwj_4; quit;
Proc iml;
use MatVi; read all into MatVi;
use Uwj; read all into Uwj;
use L5; read all into L5;
Uwj_=Uwj[9:10,];
LVLUwj_5=trace( L5*MatVi*t(L5)*Uwj_ );
create LVLUwj_5 from LVLUwj_5;
append from LVLUwj_5; quit;
Proc iml;
use MatVi; read all into MatVi;
use Uwj; read all into Uwj;
use L6; read all into L6;
Uwj_=Uwj[11:12,];
LVLUwj_6=trace( L6*MatVi*t(L6)*Uwj_ );
create LVLUwj_6 from LVLUwj_6;
append from LVLUwj_6; quit;
Proc iml;
use MatVi; read all into MatVi;
use Uwj; read all into Uwj;
use L7; read all into L7;
Uwj_=Uwj[13:14,];
LVLUwj_7=trace( L7*MatVi*t(L7)*Uwj_ );
create LVLUwj_7 from LVLUwj_7;
append from LVLUwj_7; quit;

Proc iml;
use LVLUwj_1; read all into LVLUwj_1;
use LVLUwj_2; read all into LVLUwj_2;
use LVLUwj_3; read all into LVLUwj_3;
use LVLUwj_4; read all into LVLUwj_4;
use LVLUwj_5; read all into LVLUwj_5;
use LVLUwj_6; read all into LVLUwj_6;
use LVLUwj_7; read all into LVLUwj_7;
```

```
J28=J(28,1,1);
LVLUwj_=LVLUwj_1//LVLUwj_2//LVLUwj_3//LVLUwj_4
//LVLUwj_5//LVLUwj_6//LVLUwj_7;
LVLUwj=J28 @ LVLUwj_;
create LVLUwj from LVLUwj;
append from LVLUwj; quit;

Data LVLUwj;
set LVLUwj;
rename col1=LVLUwj;
run;

/* Cálculo da parcela tr[ L * (Y-X*Beta) * (Y-
X*Beta)' * L' * Uwj ] */

Data null;
file 'C:\Documents and Settings\Politriudo\Os
meus documentos\Luís\SAS code\chamamacro.txt';
   do l= 1 to 28; k=l;
var='%ext_am('||l||','||k||');'; put var; end;
run;

%macro ext_am(l,k);

Data res;
set crono_resf;
where i=&l;
keep resid;
run;

Proc iml;
use res; read all into res;
use L1; read all into L1;
use Uwj; read all into Uwj;
Uwj_=Uwj[1:2,];
elem1= trace( L1*res*t(res)*t(L1)*Uwj_ );
create elem1 from elem1; append from elem1;
quit;
Proc iml;
use res; read all into res;
use L2; read all into L2;
use Uwj; read all into Uwj;
Uwj_=Uwj[3:4,];
elem2= trace( L2*res*t(res)*t(L2)*Uwj_ );
create elem2 from elem2; append from elem2;
quit;
Proc iml;
use res; read all into res;
use L3; read all into L3;
use Uwj; read all into Uwj;
Uwj_=Uwj[5:6,];
elem3= trace( L3*res*t(res)*t(L3)*Uwj_ );
create elem3 from elem3; append from elem3;
quit;
Proc iml;
use res; read all into res;
use L4; read all into L4;
use Uwj; read all into Uwj;
Uwj_=Uwj[7:8,];
elem4= trace( L4*res*t(res)*t(L4)*Uwj_ );
create elem4 from elem4; append from elem4;
quit;
Proc iml;
use res; read all into res;
use L5; read all into L5;
use Uwj; read all into Uwj;
Uwj_=Uwj[9:10,];
elem5= trace( L5*res*t(res)*t(L5)*Uwj_ );
create elem5 from elem5; append from elem5;
quit;
Proc iml;
use res; read all into res;
use L6; read all into L6;
use Uwj; read all into Uwj;
Uwj_=Uwj[11:12,];
elem6= trace( L6*res*t(res)*t(L6)*Uwj_ );
create elem6 from elem6; append from elem6;
quit;
Proc iml;
use res; read all into res;
use L7; read all into L7;
use Uwj; read all into Uwj;
Uwj_=Uwj[13:14,];
elem7= trace( L7*res*t(res)*t(L7)*Uwj_ );
create elem7 from elem7; append from elem7;
quit;

Proc iml;
use elem1; read all into elem1;
use elem2; read all into elem2;
use elem3; read all into elem3;
use elem4; read all into elem4;
use elem5; read all into elem5;
use elem6; read all into elem6;
use elem7; read all into elem7;
elem=elem1//elem2//elem3//elem4//elem5//elem6/
/elem7;
create elem from elem;
append from elem; quit;

Proc append base=Lres2LUwj data=elem;
run;
%mend;
%inc 'C:\Documents and Settings\Politriudo\Os
meus documentos\Luís\Doutoramento\SAS
code\chamamacro.txt';

Data Lres2LUwj1;
set Lres2LUwj;
rename col1=Lres2LUwj;
run;

Proc datasets;
Delete Lres2LUwj;
run;


/* Aqui termina o cálculo das seguintes
parcelas do EQM:
tBwj*Dg1 - está no ficheiro "tBwjDg1"
tr[ L * V * L' * Uwj ] - está no ficheiro
"LVLUwj"
tr[ L * (Y-X*Beta) * (Y-X*Beta)' * L' * Uwj ]
- está no ficheiro "Lres2LUwj" */

Data indice;
        do i=1 to 28;
                do t=1 to 7;
                output;
                end;
        end;
run;

Data Est_ryu_eqm_mod_jack_s;
merge indice tBwjDg1 LVLUwj Lres2LUwj1;
simul=1;   /*SSS*/
run;

Proc append base=RaoYu.Est_ryu_eqm_mod1_jack
data=Est_ryu_eqm_mod_jack_s;
run;

DM 'CLEAR OUTPUT';
DM 'CLEAR LOG';

/********************************************
********************************************/
/********************************************
********************************************/
/** Cálculo das medidas de avaliação dos
estimadores do EQMP no estudo por simulação
model-based *******************************/
/* Código SAS utilizado para calcular essas
medidas para cada combinação de parâmetros **/
/********************************************
********************************************/
/********************************************
********************************************/

/* 1 - Cálculo das estimativas bootstrap
(componentes g1_boot, g2_boot e g3_boot */

Data null;
file 'C:\Documents and Settings\Luis
Pereira\My Documents\SAS code\chamamacro.txt';
   do i= 1 to 250; k=i;
var='%ext_am('||i||','||k||');'; put var; end;
run;

%macro ext_am(i,k);

Proc sort data=
RaoYu.Est_Ryu_EQM_mod1_Boot_S&i out=EQM_aux;
/*SSS*/
by i t;
run;
```

```
Proc means data=EQM_aux mean noprint;
var dif2 g1_boot g2_boot;
by i t;
output out=EQM_out mean=g3_boot g1_boot
g2_boot;
run;

Data EQM_out;
set EQM_out;
simul=&i;
run;

Proc append base= RaoYu.Est_Ryu_EQMBoot_mod1
data=EQM_out;   /*SSS*/
run;

%mend;
%inc 'C:\Documents and Settings\Luis
Pereira\My Documents\SAS code\chamamacro.txt';

/* 2 - Cálculo das aproximações aos
verdadeiros valores do EQMP */

Data Gerac_Ryu_modelo;
set RaoYu.Gerac_Ryu_modelo1;  /*SSS*/
keep i t simul y;
run;

Data EQM_empiricototal;
Merge Raoyu.Est_Ryu_modelo1 Gerac_Ryu_modelo;
/*SSS*/
by simul i t;
EQM_EMPIRICO= (e-y1)*(e-y1);
keep i t simul y1 e EQM_EMPIRICO;
run;

Proc sort data=EQM_empiricototal;
by i t;
run;

Proc means data=EQM_empiricototal mean;
var EQM_EMPIRICO;
by i t;
output out=EQM_empirico mean=;
run;

Proc iml;
use EQM_empirico; read all into EQM_empirico;
J1000=J(1000,1,1);
J196=J(196,1,1);
V1000=(1:1000);
aux1=J1000@EQM_empirico;
aux2=t(v1000)@J196;
vector_eqm=aux2||aux1;
create vector_eqm from vector_eqm; append from
vector_eqm; quit;

Data vector_eqm;
set vector_eqm;
rename col1=simul col2=i col3=t
col6=EQM_EMPIRICO;
drop col4 col5;
run;

Data Est_Ryu_modelo;
set RaoYu.Est_Ryu_modelo1;
keep i t simul g1 g2 EQM_EBLUP;
run;

Proc sort data=RaoYu.Est_RYU_EQM_mod1_jack;
/*SSS*/
by simul i t;
Proc sort data=RaoYu.Est_RYU_EQMBoot_mod1;
/*SSS*/
by simul i t;
run;

Data RaoYu.Est_RYU_EQM_mod1_jack_wm;  /*SSS*/
merge Est_Ryu_modelo
RaoYu.Est_RYU_EQM_mod1_jack_wm;  /*SSS*/
by simul i t;
EQM_JACK_1= g1 + g2 - tBwjDg1 + LVLUwj +
Lres2LUwj;
keep i t simul tBwjDg1 LVLUwj Lres2LUwj
EQM_JACK_1; run;

Data RaoYu.Est_RYU_EQM_mod1_jack;   /*SSS*/
merge Est_Ryu_modelo
RaoYu.Est_RYU_EQM_mod1_jack;   /*SSS*/
by simul i t;
EQM_JACK_2= g1 + g2 - tBwjDg1 + LVLUwj +
Lres2LUwj;
keep i t simul tBwjDg1 LVLUwj Lres2LUwj
EQM_JACK_2; run;

/* 3 - Cálculo das estimativas do EQMP_BOOT e
respectiva análise de dados */

Data RaoYu.Est_Ryu_modelo1_completo_novo;
/*SSS*/
merge vector_eqm Est_Ryu_modelo
RaoYu.Est_RYU_EQMBoot_mod1_novo
RaoYu.Est_RYU_EQM_mod1_jack
RaoYu.Est_RYU_EQM_mod1_jack_wm;
by simul i t;

EQM_BOOT_1= 2*(g1 + g2) - g1_boot - g2_boot +
g3_boot;

RB_EQM_EBLUP=100*(EQM_EBLUP -
EQM_EMPIRICO)/EQM_EMPIRICO;
RB_EQM_BOOT_1=100*(EQM_BOOT_1 -
EQM_EMPIRICO)/EQM_EMPIRICO;
RB_EQM_JACK_1=100*(EQM_JACK_1 -
EQM_EMPIRICO)/EQM_EMPIRICO;
RB_EQM_JACK_2=100*(EQM_JACK_2 -
EQM_EMPIRICO)/EQM_EMPIRICO;

B_EQM_EBLUP=(EQM_EBLUP - EQM_EMPIRICO);
B_EQM_BOOT_1=(EQM_BOOT_1 - EQM_EMPIRICO);
B_EQM_JACK_1=(EQM_JACK_1 - EQM_EMPIRICO);
B_EQM_JACK_2=(EQM_JACK_2 - EQM_EMPIRICO);

REQM_EQM_EBLUP=100*(EQM_EBLUP -
EQM_EMPIRICO)*(EQM_EBLUP -
EQM_EMPIRICO)/EQM_EMPIRICO;
REQM_EQM_BOOT_1=100*(EQM_BOOT_1 -
EQM_EMPIRICO)*(EQM_BOOT_1 -
EQM_EMPIRICO)/EQM_EMPIRICO;
REQM_EQM_JACK_1=100*(EQM_JACK_1 -
EQM_EMPIRICO)*(EQM_JACK_1 -
EQM_EMPIRICO)/EQM_EMPIRICO;
REQM_EQM_JACK_2=100*(EQM_JACK_2 -
EQM_EMPIRICO)*(EQM_JACK_2 -
EQM_EMPIRICO)/EQM_EMPIRICO;

AB_EQM_EBLUP=ABS(EQM_EBLUP - EQM_EMPIRICO);
AB_EQM_BOOT_1=ABS(EQM_BOOT_1 - EQM_EMPIRICO);
AB_EQM_JACK_1=ABS(EQM_JACK_1 - EQM_EMPIRICO);
AB_EQM_JACK_2=ABS(EQM_JACK_2 - EQM_EMPIRICO);

EQM_EQM_EBLUP=(EQM_EBLUP -
EQM_EMPIRICO)*(EQM_EBLUP - EQM_EMPIRICO);
EQM_EQM_BOOT_1=(EQM_BOOT_1 -
EQM_EMPIRICO)*(EQM_BOOT_1 - EQM_EMPIRICO);
EQM_EQM_JACK_1=(EQM_JACK_1 -
EQM_EMPIRICO)*(EQM_JACK_1 - EQM_EMPIRICO);
EQM_EQM_JACK_2=(EQM_JACK_2 -
EQM_EMPIRICO)*(EQM_JACK_2 - EQM_EMPIRICO);

keep i t simul EQM_EMPIRICO EQM_EBLUP
EQM_BOOT_1 EQM_JACK_1 EQM_JACK_2 RB_EQM_EBLUP
RB_EQM_BOOT_1 RB_EQM_JACK_1 RB_EQM_JACK_2
B_EQM_EBLUP B_EQM_BOOT_1 B_EQM_JACK_1
B_EQM_JACK_2 REQM_EQM_EBLUP REQM_EQM_BOOT_1
REQM_EQM_JACK_1 REQM_EQM_JACK_2 AB_EQM_EBLUP
AB_EQM_BOOT_1 AB_EQM_JACK_1 AB_EQM_JACK_2
EQM_EQM_EBLUP EQM_EQM_BOOT_1 EQM_EQM_JACK_1
EQM_EQM_JACK_2;
run;

Proc sort
data=RaoYu.Est_Ryu_modelo1_completo_novo
out=Est_Ryu_modelo_completo;   /*SSS*/
by i t; run;

Proc means data=Est_Ryu_modelo_completo mean
std median;  /*SSS*/
var EQM_EMPIRICO EQM_EBLUP EQM_BOOT_1
EQM_JACK_1 EQM_JACK_2
RB_EQM_EBLUP RB_EQM_BOOT_1 RB_EQM_JACK_1
RB_EQM_JACK_2
B_EQM_EBLUP B_EQM_BOOT_1 B_EQM_JACK_1
B_EQM_JACK_2
```

```
REQM_EQM_EBLUP REQM_EQM_BOOT_1 REQM_EQM_JACK_1
REQM_EQM_JACK_2
AB_EQM_EBLUP AB_EQM_BOOT_1 AB_EQM_JACK_1
AB_EQM_JACK_2
EQM_EQM_EBLUP EQM_EQM_BOOT_1 EQM_EQM_JACK_1
EQM_EQM_JACK_2;
*by i t;
output
out=RaoYu.Res_media_mod1_it_bootjackk_novo
mean=;  /*SSS*/
run;

Data Est_res;
set RaoYu.Res_media_mod1_it_bootjackk_novo;
BR_EBLUP=AB_EQM_EBLUP/sqrt(EQM_EQM_EBLUP);
BR_BOOT1=AB_EQM_BOOT_1/sqrt(EQM_EQM_BOOT_1);
BR_JACK1=AB_EQM_JACK_1/sqrt(EQM_EQM_JACK_1);
BR_JACK2=AB_EQM_JACK_2/sqrt(EQM_EQM_JACK_2);
run;

Proc means data=Est_res mean;  /*SSS*/
var BR_EBLUP BR_BOOT1 BR_JACK1 BR_JACK2;
output out=Out1 mean=; run;

Data Est_Ryu;
set RaoYu.Est_Ryu_modelo1_completo_novo ;
if RB_EQM_EBLUP <0 then EBLUP =1;
if RB_EQM_BOOT_1 <0 then Boot_1 =1;
if RB_EQM_JACK_1 <0 then Jack_1 =1;
if RB_EQM_JACK_2 <0 then Jack_2 =1;
ARB_EQM_EBLUP =abs(RB_EQM_EBLUP);
ARB_EQM_BOOT_1 =abs(RB_EQM_BOOT_1);
ARB_EQM_JACK_1 =abs(RB_EQM_JACK_1);
ARB_EQM_JACK_2 =abs(RB_EQM_JACK_2);
run;
Proc means data=Est_Ryu sum;
var EBLUP Boot_1 Jack_1 Jack_2 ;
output out= Est_res11 sum=;
run;
Proc means data=Est_Ryu mean;
var ARB_EQM_EBLUP ARB_EQM_BOOT_1
ARB_EQM_JACK_1 ARB_EQM_JACK_2;
output out= Est_res12 mean=;
run;

Data RaoYu.Res_media_mod1_it_bootjackk_novo;
set RaoYu.Res_media_mod1_it_bootjackk_novo
Est_res11 Est_res12;
mod=1;
run;
```

Apêndice 24 – Estimação do EQMP do EBLUP espaciotemporal no âmbito do estudo empírico por simulação *model-based*

CÓDIGO SAS

```
/* Considera-se o modelo espaciotemporal com
p=2 (constante + variável explicativa):
 Yit = B1 + B2Xit + Vi + U2it + eit
 U2it = ro*U2i,t-1 + Eit
 V = (I - phi*W)-1 * U1

Verdadeiros coeficientes do modelo: B1=1 ,
B2=2
        ----------        V(U1i) = sigma2v =
0.5
        MODELO 111        V(Eit) = sigma2 =
0.25
        ----------        ro=0.2
                          phi=0.25

Considera-se: m=28 small areas
           T=7 periods of time

Foram geradas:
K=1000 amostras independentes de Yit
Xit foram gerados a partir de uma distribuição
uniforme no intervalo (0, 1)
U1i foram gerados a partir de uma distribuição
normal: N(0, sigma2v)
eit foram gerados a partir de uma distribuição
normal: N(0, 1)
Eit foram gerados a partir de uma distribuição
normal: N(0, sigma2)

Data null;
file 'C:\Documents and Settings\Luis
Pereira\My
Documents\Doutoramento\chamamacro.txt';
   do i= 1 to 1000; k=i;
var='%ext_am('||i||','||k||');'; put var; end;
run;

%macro ext_am(i,k);

Data a1;
do i=1 to 28;
        do t=1 to 7;
                c=1;
                x=ranuni(&k);
                e=rannor(&k+1);
                Eu=sqrt(0.25)*rannor(&k+2);
        output;
end; end; run;

Data a2;
do i=1 to 28;
        v=sqrt(0.5)*rannor(&k+3);
        output;
end; run;

/* Determinação dos efeitos aleatórios de
domínio-tempo : u */
/* Autocorrelação temporal */

Data dataro;
ro=0.2;
run;

Data a4;
set a1;
keep Eu;
run;

Proc iml;
use a4; read all into a4;
use dataro; read all into dataro;
Ident7=I(7);
Ident28=I(28);
I1={0 0 0 0 0 0 0,
    1 0 0 0 0 0 0,
    0 1 0 0 0 0 0,
    0 0 1 0 0 0 0,
    0 0 0 1 0 0 0,
    0 0 0 0 1 0 0,
    0 0 0 0 0 1 0};
I2={0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    1 0 0 0 0 0 0,
    0 1 0 0 0 0 0,
    0 0 1 0 0 0 0,
    0 0 0 1 0 0 0,
    0 0 0 0 1 0 0};
I3={0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    1 0 0 0 0 0 0,
    0 1 0 0 0 0 0,
    0 0 1 0 0 0 0,
    0 0 0 1 0 0 0};
I4={0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    1 0 0 0 0 0 0,
    0 1 0 0 0 0 0,
    0 0 1 0 0 0 0};
I5={0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    1 0 0 0 0 0 0,
    0 1 0 0 0 0 0};
I6={0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    1 0 0 0 0 0 0};
Q=dataro*I1 + dataro*dataro*I2 +
dataro*dataro*dataro*I3 +
dataro*dataro*dataro*dataro*I4 +
dataro*dataro*dataro*dataro*dataro*I5 +
dataro*dataro*dataro*dataro*dataro*dataro*I6 +
Ident7;
a5=(Ident28@Q)*a4;
create a5 from a5; append from a5;
quit;

Data a5;
set a5;
rename col1=u; run;

/* Determinação dos efeitos aleatórios de
domínio : v */
/* Dependência espacial */

Data dataphi;
phi=0.25;
run;

Data a2;
set a2;
drop i; run;

Proc iml;
use a2; read all into a2;
use dataphi; read all into dataphi;
use Pc.W2; read all into W2;
Ident=I(28);
v=inv(Ident - dataphi*W2)*a2;
um=J(7,1);
a3=v@um;
create a3 from a3; append from a3;
quit;

Data a3;
set a3;
rename col1=v; run;
```

```
/****/

Data a6;
merge a1 a3 a5;
y=1*c + 2*x + v + u + e;
drop eu;
run;

Data a7;
set a6;
simul=&i;
run;

/*******************************************/
/******** Fim da geração dos dados *********/
/*******************************************/

/* Matriz B=(I-phi*W)'*(I-phi*W) */

Data W;
set PC.W2;   /* Matriz de distâncias */
run;

Proc iml;
use W; read all into W;
use dataphi; read all into dataphi;
Ident=I(28);
B = t(Ident - dataphi*W)*(Ident - dataphi*W);
create B from B; append from B; quit;

Proc iml;
use B; read all into B;
Binv=inv(B);   /* Inversa de B */
create Binv from Binv; append from Binv; quit;

/* Matriz P */

Data dataro2;
set dataro;
ro2= sqrt(1-ro*ro);
keep ro2;
run;

Proc iml;
use dataro;
read all into dataro;
use dataro2;
read all into dataro2;
I1={1 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I2={0 0 0 0 0 0 0, 0 1 0 0 0 0 0, 0 0 1 0 0 0
0, 0 0 0 1 0 0 0, 0 0 0 0 1 0 0, 0 0 0 0 0 1
0, 0 0 0 0 0 0 1};
I4={0 0 0 0 0 0 0, 1 0 0 0 0 0 0, 0 1 0 0 0 0
0, 0 0 1 0 0 0 0, 0 0 0 1 0 0 0, 0 0 0 0 1 0
0, 0 0 0 0 0 1 0};
P = dataro2*I1 + I2 - dataro*I4;
create P from P;
append from P;
quit;

Proc iml;
use dataro;
read all into dataro;
use dataro2;
read all into dataro2;
C1={1, 0, 0, 0, 0, 0, 0};
C2={0, 1, 1, 1, 1, 1, 1};
f = dataro2*C1 + C2 - dataro*C2;
create f from f;
append from f; quit;

Proc iml;
use f;
read all into f;
c=f`*f;
create c from c;
append from c; quit;

Proc iml;
use f;
read all into f;
use c;
read all into c;
D=(1/c)*f*f`;
create D from D;
append from D; quit;

/* *** */

Data iabh;
set a6;
keep i t c x; run;

Data ipth;
set a6;
keep i t y; run;

Data iabhipth1;
set a6;
keep i t y x; run;

/** Preparação do ficheiro das observações da
variável de interesse **/

Proc sort data=ipth;
by i t; run;

Data tetad1; set ipth; where i= 1; keep y;
Data tetad2; set ipth; where i= 2; keep y;
Data tetad3; set ipth; where i= 3; keep y;
Data tetad4; set ipth; where i= 4; keep y;
Data tetad5; set ipth; where i= 5; keep y;
Data tetad6; set ipth; where i= 6; keep y;
Data tetad7; set ipth; where i= 7; keep y;
Data tetad8; set ipth; where i= 8; keep y;
Data tetad9; set ipth; where i= 9; keep y;
Data tetad10; set ipth; where i= 10; keep y;
Data tetad11; set ipth; where i= 11; keep y;
Data tetad12; set ipth; where i= 12; keep y;
Data tetad13; set ipth; where i= 13; keep y;
Data tetad14; set ipth; where i= 14; keep y;
Data tetad15; set ipth; where i= 15; keep y;
Data tetad16; set ipth; where i= 16; keep y;
Data tetad17; set ipth; where i= 17; keep y;
Data tetad18; set ipth; where i= 18; keep y;
Data tetad19; set ipth; where i= 19; keep y;
Data tetad20; set ipth; where i= 20; keep y;
Data tetad21; set ipth; where i= 21; keep y;
Data tetad22; set ipth; where i= 22; keep y;
Data tetad23; set ipth; where i= 23; keep y;
Data tetad24; set ipth; where i= 24; keep y;
Data tetad25; set ipth; where i= 25; keep y;
Data tetad26; set ipth; where i= 26; keep y;
Data tetad27; set ipth; where i= 27; keep y;
Data tetad28; set ipth; where i= 28; keep y;
run;
/** Preparação do ficheiro das observações da
variável auxiliar **/

Proc sort data=iabh;
by i t; run;

Proc sql; create table iabh0 as select c, x,
i, t from iabh;

Data xd1; set iabh0; where i= 1; keep c x;
Data xd2; set iabh0; where i= 2; keep c x;
Data xd3; set iabh0; where i= 3; keep c x;
Data xd4; set iabh0; where i= 4; keep c x;
Data xd5; set iabh0; where i= 5; keep c x;
Data xd6; set iabh0; where i= 6; keep c x;
Data xd7; set iabh0; where i= 7; keep c x;
Data xd8; set iabh0; where i= 8; keep c x;
Data xd9; set iabh0; where i= 9; keep c x;
Data xd10; set iabh0; where i= 10; keep c x;
Data xd11; set iabh0; where i= 11; keep c x;
Data xd12; set iabh0; where i= 12; keep c x;
Data xd13; set iabh0; where i= 13; keep c x;
Data xd14; set iabh0; where i= 14; keep c x;
Data xd15; set iabh0; where i= 15; keep c x;
Data xd16; set iabh0; where i= 16; keep c x;
Data xd17; set iabh0; where i= 17; keep c x;
Data xd18; set iabh0; where i= 18; keep c x;
Data xd19; set iabh0; where i= 19; keep c x;
Data xd20; set iabh0; where i= 20; keep c x;
Data xd21; set iabh0; where i= 21; keep c x;
Data xd22; set iabh0; where i= 22; keep c x;
Data xd23; set iabh0; where i= 23; keep c x;
Data xd24; set iabh0; where i= 24; keep c x;
Data xd25; set iabh0; where i= 25; keep c x;
Data xd26; set iabh0; where i= 26; keep c x;
Data xd27; set iabh0; where i= 27; keep c x;
Data xd28; set iabh0; where i= 28; keep c x;
```

```
run;

/** Cálculo da matriz Z(1) **/

Proc iml;
use tetad1; read all into tetad1;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd1=(Ident - D)*P*tetad1;
create zd1 from zd1; append from zd1; quit;
Proc iml;
use tetad2; read all into tetad2;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd2=(Ident - D)*P*tetad2;
create zd2 from zd2; append from zd2; quit;
Proc iml;
use tetad3; read all into tetad3;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd3=(Ident - D)*P*tetad3;
create zd3 from zd3; append from zd3; quit;
Proc iml;
use tetad4; read all into tetad4;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd4=(Ident - D)*P*tetad4;
create zd4 from zd4; append from zd4; quit;
Proc iml;
use tetad5; read all into tetad5;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd5=(Ident - D)*P*tetad5;
create zd5 from zd5; append from zd5; quit;
Proc iml;
use tetad6; read all into tetad6;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd6=(Ident - D)*P*tetad6;
create zd6 from zd6; append from zd6; quit;
Proc iml;
use tetad7; read all into tetad7;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd7=(Ident - D)*P*tetad7;
create zd7 from zd7; append from zd7; quit;
Proc iml;
use tetad8; read all into tetad8;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd8=(Ident - D)*P*tetad8;
create zd8 from zd8; append from zd8; quit;
Proc iml;
use tetad9; read all into tetad9;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd9=(Ident - D)*P*tetad9;
create zd9 from zd9; append from zd9; quit;
Proc iml;
use tetad10; read all into tetad10;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd10=(Ident - D)*P*tetad10;
create zd10 from zd10; append from zd10; quit;
Proc iml;
use tetad11; read all into tetad11;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd11=(Ident - D)*P*tetad11;
create zd11 from zd11; append from zd11; quit;
Proc iml;
use tetad12; read all into tetad12;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd12=(Ident - D)*P*tetad12;
create zd12 from zd12; append from zd12; quit;
Proc iml;
use tetad13; read all into tetad13;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd13=(Ident - D)*P*tetad13;
create zd13 from zd13; append from zd13; quit;
Proc iml;
use tetad14; read all into tetad14;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd14=(Ident - D)*P*tetad14;
create zd14 from zd14; append from zd14; quit;
Proc iml;
use tetad15; read all into tetad15;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd15=(Ident - D)*P*tetad15;
create zd15 from zd15; append from zd15; quit;
Proc iml;
use tetad16; read all into tetad16;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd16=(Ident - D)*P*tetad16;
create zd16 from zd16; append from zd16; quit;
Proc iml;
use tetad17; read all into tetad17;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd17=(Ident - D)*P*tetad17;
create zd17 from zd17; append from zd17; quit;
Proc iml;
use tetad18; read all into tetad18;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd18=(Ident - D)*P*tetad18;
create zd18 from zd18; append from zd18; quit;
Proc iml;
use tetad19; read all into tetad19;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd19=(Ident - D)*P*tetad19;
create zd19 from zd19; append from zd19; quit;
Proc iml;
use tetad20; read all into tetad20;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd20=(Ident - D)*P*tetad20;
create zd20 from zd20; append from zd20; quit;
Proc iml;
use tetad21; read all into tetad21;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd21=(Ident - D)*P*tetad21;
create zd21 from zd21; append from zd21; quit;
Proc iml;
use tetad22; read all into tetad22;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd22=(Ident - D)*P*tetad22;
create zd22 from zd22; append from zd22; quit;
Proc iml;
use tetad23; read all into tetad23;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd23=(Ident - D)*P*tetad23;
create zd23 from zd23; append from zd23; quit;
Proc iml;
use tetad24; read all into tetad24;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd24=(Ident - D)*P*tetad24;
create zd24 from zd24; append from zd24; quit;
Proc iml;
use tetad25; read all into tetad25;
use P; read all into P;
use D; read all into D;
```

```
Ident=I(7);
zd25=(Ident - D)*P*tetad25;
create zd25 from zd25; append from zd25; quit;
Proc iml;
use tetad26; read all into tetad26;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd26=(Ident - D)*P*tetad26;
create zd26 from zd26; append from zd26; quit;
Proc iml;
use tetad27; read all into tetad27;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd27=(Ident - D)*P*tetad27;
create zd27 from zd27; append from zd27; quit;
Proc iml;
use tetad28; read all into tetad28;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd28=(Ident - D)*P*tetad28;
create zd28 from zd28; append from zd28; quit;

Data Z1;
set zd1 zd2 zd3 zd4 zd5 zd6 zd7 zd8 zd9 zd10
zd11 zd12 zd13 zd14 zd15
zd16 zd17 zd18 zd19 zd20 zd21 zd22 zd23 zd24
zd25 zd26 zd27 zd28;
run;

/** Cálculo da matriz H(1) **/

Proc iml;
use xd1; read all into xd1;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd1=(Ident - D)*P*xd1;
create hd1 from hd1; append from hd1; quit;
Proc iml;
use xd2; read all into xd2;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd2=(Ident - D)*P*xd2;
create hd2 from hd2; append from hd2; quit;
Proc iml;
use xd3; read all into xd3;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd3=(Ident - D)*P*xd3;
create hd3 from hd3; append from hd3; quit;
Proc iml;
use xd4; read all into xd4;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd4=(Ident - D)*P*xd4;
create hd4 from hd4; append from hd4; quit;
Proc iml;
use xd5; read all into xd5;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd5=(Ident - D)*P*xd5;
create hd5 from hd5; append from hd5; quit;
Proc iml;
use xd6; read all into xd6;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd6=(Ident - D)*P*xd6;
create hd6 from hd6; append from hd6; quit;
Proc iml;
use xd7; read all into xd7;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd7=(Ident - D)*P*xd7;
create hd7 from hd7; append from hd7; quit;
Proc iml;
use xd8; read all into xd8;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd8=(Ident - D)*P*xd8;
create hd8 from hd8; append from hd8; quit;
Proc iml;
use xd9; read all into xd9;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd9=(Ident - D)*P*xd9;
create hd9 from hd9; append from hd9; quit;
Proc iml;
use xd10; read all into xd10;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd10=(Ident - D)*P*xd10;
create hd10 from hd10; append from hd10; quit;
Proc iml;
use xd11; read all into xd11;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd11=(Ident - D)*P*xd11;
create hd11 from hd11; append from hd11; quit;
Proc iml;
use xd12; read all into xd12;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd12=(Ident - D)*P*xd12;
create hd12 from hd12; append from hd12; quit;
Proc iml;
use xd13; read all into xd13;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd13=(Ident - D)*P*xd13;
create hd13 from hd13; append from hd13; quit;
Proc iml;
use xd14; read all into xd14;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd14=(Ident - D)*P*xd14;
create hd14 from hd14; append from hd14; quit;
Proc iml;
use xd15; read all into xd15;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd15=(Ident - D)*P*xd15;
create hd15 from hd15; append from hd15; quit;
Proc iml;
use xd16; read all into xd16;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd16=(Ident - D)*P*xd16;
create hd16 from hd16; append from hd16; quit;
Proc iml;
use xd17; read all into xd17;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd17=(Ident - D)*P*xd17;
create hd17 from hd17; append from hd17; quit;
Proc iml;
use xd18; read all into xd18;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd18=(Ident - D)*P*xd18;
create hd18 from hd18; append from hd18; quit;
Proc iml;
use xd19; read all into xd19;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd19=(Ident - D)*P*xd19;
create hd19 from hd19; append from hd19; quit;
Proc iml;
use xd20; read all into xd20;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd20=(Ident - D)*P*xd20;
create hd20 from hd20; append from hd20; quit;
Proc iml;
use xd21; read all into xd21;
```

```
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd21=(Ident - D)*P*xd21;
create hd21 from hd21; append from hd21; quit;
Proc iml;
use xd22; read all into xd22;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd22=(Ident - D)*P*xd22;
create hd22 from hd22; append from hd22; quit;
Proc iml;
use xd23; read all into xd23;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd23=(Ident - D)*P*xd23;
create hd23 from hd23; append from hd23; quit;
Proc iml;
use xd24; read all into xd24;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd24=(Ident - D)*P*xd24;
create hd24 from hd24; append from hd24; quit;
Proc iml;
use xd25; read all into xd25;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd25=(Ident - D)*P*xd25;
create hd25 from hd25; append from hd25; quit;
Proc iml;
use xd26; read all into xd26;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd26=(Ident - D)*P*xd26;
create hd26 from hd26; append from hd26; quit;
Proc iml;
use xd27; read all into xd27;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd27=(Ident - D)*P*xd27;
create hd27 from hd27; append from hd27; quit;
Proc iml;
use xd28; read all into xd28;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd28=(Ident - D)*P*xd28;
create hd28 from hd28; append from hd28; quit;

Data H1;
set hd1 hd2 hd3 hd4 hd5 hd6 hd7 hd8 hd9 hd10
hd11 hd12 hd13 hd14 hd15
hd16 hd17 hd18 hd19 hd20 hd21 hd22 hd23 hd24
hd25 hd26 hd27 hd28;
run;

/** Regressão de Z(1) sobre H(1) **/

Data Z1;
set Z1;
rename col1=col0;
run;

Data reg;
merge Z1 H1;
run;

proc reg data=reg noprint;
        model col0=col1 col2 / noint;
        output out=rg1  residual=r1;
run; quit;

Data rg1;
set rg1;
qr1=r1*r1;
Keep qr1;
run;

Proc means data=rg1 sum noprint;
var qr1;
output out=rg1_ sum=sqr1;
run;

Data rg1_;
set rg1_;
keep sqr1;
run;

/** Cálculo da variância sigma2_u **/

Proc iml;
use H1; read all into H1;
Hs1=H1*ginv(t(H1)*H1)*t(H1); /* ginv(A) is the
Moore-Penrose generalized inverse matrix of A
*/
create Hs1 from Hs1; append from Hs1; quit; /*
Hs1 é a matriz H*(1) */

Proc iml;
use D; read all into D;
Ident=I(7);
ID=Ident - D; /* Matriz I - D */
create ID from ID; append from ID; quit;

Proc iml;
use ID; read all into ID;
blockID1=BLOCK(ID,ID,ID,ID,ID,ID,ID,ID,ID,ID,I
D,ID,ID,ID);
create blockID1 from blockID1; append from
blockID1; quit;
Proc iml;
use blockID1; read all into blockID1;
blockID=BLOCK(blockID1,blockID1);
create blockID from blockID; append from
blockID; quit;

Data s1;
set a6;
keep c t i;  /*Admite-se que a variância dos
erros da sondagem é igual a 1, logo é igual a
c */
run;    /* Neste caso, as matrizes Pi são
todas iguais */

Data s1;
set s1;
rename c=var;
run;

Data Sigma_1;
set s1;
if i=1;
keep var;
run;
Proc iml;
use Sigma_1; read all into Sigma_1;
Sigma1=diag(Sigma_1);
create Sigma1 from Sigma1; append from Sigma1;
quit;
Proc iml;
use P; read all into P;
use Sigma1; read all into Sigma1;
P1=P*Sigma1*t(P);
create P1 from P1; append from P1; quit;

Data Sigma_2;
set s1;
if i=2;
keep var;
run;
Proc iml;
use Sigma_2; read all into Sigma_2;
Sigma2=diag(Sigma_2);
create Sigma2 from Sigma2; append from Sigma2;
quit;
Proc iml;
use P; read all into P;
use Sigma2; read all into Sigma2;
P2=P*Sigma2*t(P);
create P2 from P2; append from P2; quit;

Data Sigma_3;
set s1;
if i=3;
keep var;
run;
Proc iml;
use Sigma_3; read all into Sigma_3;
Sigma3=diag(Sigma_3);
```

```
create Sigma3 from Sigma3; append from Sigma3;
quit;
Proc iml;
use P; read all into P;
use Sigma3; read all into Sigma3;
P3=P*Sigma3*t(P);
create P3 from P3; append from P3; quit;

Data Sigma_4;
set s1;
if i=4;
keep var;
run;
Proc iml;
use Sigma_4; read all into Sigma_4;
Sigma4=diag(Sigma_4);
create Sigma4 from Sigma4; append from Sigma4;
quit;
Proc iml;
use P; read all into P;
use Sigma4; read all into Sigma4;
P4=P*Sigma4*t(P);
create P4 from P4; append from P4; quit;

Data Sigma_5;
set s1;
if i=5;
keep var;
run;
Proc iml;
use Sigma_5; read all into Sigma_5;
Sigma5=diag(Sigma_5);
create Sigma5 from Sigma5; append from Sigma5;
quit;
Proc iml;
use P; read all into P;
use Sigma5; read all into Sigma5;
P5=P*Sigma5*t(P);
create P5 from P5; append from P5; quit;

Data Sigma_6;
set s1;
if i=6;
keep var;
run;
Proc iml;
use Sigma_6; read all into Sigma_6;
Sigma6=diag(Sigma_6);
create Sigma6 from Sigma6; append from Sigma6;
quit;
Proc iml;
use P; read all into P;
use Sigma6; read all into Sigma6;
P6=P*Sigma6*t(P);
create P6 from P6; append from P6; quit;

Data Sigma_7;
set s1;
if i=7;
keep var;
run;
Proc iml;
use Sigma_7; read all into Sigma_7;
Sigma7=diag(Sigma_7);
create Sigma7 from Sigma7; append from Sigma7;
quit;
Proc iml;
use P; read all into P;
use Sigma7; read all into Sigma7;
P7=P*Sigma7*t(P);
create P7 from P7; append from P7; quit;

Data Sigma_8;
set s1;
if i=8;
keep var;
run;
Proc iml;
use Sigma_8; read all into Sigma_8;
Sigma8=diag(Sigma_8);
create Sigma8 from Sigma8; append from Sigma8;
quit;
Proc iml;
use P; read all into P;
use Sigma8; read all into Sigma8;
P8=P*Sigma8*t(P);
create P8 from P8; append from P8; quit;

Data Sigma_9;
set s1;
if i=9;
keep var;
run;
Proc iml;
use Sigma_9; read all into Sigma_9;
Sigma9=diag(Sigma_9);
create Sigma9 from Sigma9; append from Sigma9;
quit;
Proc iml;
use P; read all into P;
use Sigma9; read all into Sigma9;
P9=P*Sigma9*t(P);
create P9 from P9; append from P9; quit;

Data Sigma_10;
set s1;
if i=10;
keep var;
run;
Proc iml;
use Sigma_10; read all into Sigma_10;
Sigma10=diag(Sigma_10);
create Sigma10 from Sigma10; append from
Sigma10; quit;
Proc iml;
use P; read all into P;
use Sigma10; read all into Sigma10;
P10=P*Sigma10*t(P);
create P10 from P10; append from P10; quit;

Data Sigma_11;
set s1;
if i=11;
keep var;
run;
Proc iml;
use Sigma_11; read all into Sigma_11;
Sigma11=diag(Sigma_11);
create Sigma11 from Sigma11; append from
Sigma11; quit;
Proc iml;
use P; read all into P;
use Sigma11; read all into Sigma11;
P11=P*Sigma11*t(P);
create P11 from P11; append from P11; quit;

Data Sigma_12;
set s1;
if i=12;
keep var;
run;
Proc iml;
use Sigma_12; read all into Sigma_12;
Sigma12=diag(Sigma_12);
create Sigma12 from Sigma12; append from
Sigma12; quit;
Proc iml;
use P; read all into P;
use Sigma12; read all into Sigma12;
P12=P*Sigma12*t(P);
create P12 from P12; append from P12; quit;

Data Sigma_13;
set s1;
if i=13;
keep var;
run;
Proc iml;
use Sigma_13; read all into Sigma_13;
Sigma13=diag(Sigma_13);
create Sigma13 from Sigma13; append from
Sigma13; quit;
Proc iml;
use P; read all into P;
use Sigma13; read all into Sigma13;
P13=P*Sigma13*t(P);
create P13 from P13; append from P13; quit;

Data Sigma_14;
set s1;
if i=14;
keep var;
run;
Proc iml;
use Sigma_14; read all into Sigma_14;
Sigma14=diag(Sigma_14);
```

```
create Sigma14 from Sigma14; append from
Sigma14; quit;
Proc iml;
use P; read all into P;
use Sigma14; read all into Sigma14;
P14=P*Sigma14*t(P);
create P14 from P14; append from P14; quit;

Data Sigma_15;
set s1;
if i=15;
keep var;
run;
Proc iml;
use Sigma_15; read all into Sigma_15;
Sigma15=diag(Sigma_15);
create Sigma15 from Sigma15; append from
Sigma15; quit;
Proc iml;
use P; read all into P;
use Sigma15; read all into Sigma15;
P15=P*Sigma15*t(P);
create P15 from P15; append from P15; quit;

Data Sigma_16;
set s1;
if i=16;
keep var;
run;
Proc iml;
use Sigma_16; read all into Sigma_16;
Sigma16=diag(Sigma_16);
create Sigma16 from Sigma16; append from
Sigma16; quit;
Proc iml;
use P; read all into P;
use Sigma16; read all into Sigma16;
P16=P*Sigma16*t(P);
create P16 from P16; append from P16; quit;

Data Sigma_17;
set s1;
if i=17;
keep var;
run;
Proc iml;
use Sigma_17; read all into Sigma_17;
Sigma17=diag(Sigma_17);
create Sigma17 from Sigma17; append from
Sigma17; quit;
Proc iml;
use P; read all into P;
use Sigma17; read all into Sigma17;
P17=P*Sigma17*t(P);
create P17 from P17; append from P17; quit;

Data Sigma_18;
set s1;
if i=18;
keep var;
run;
Proc iml;
use Sigma_18; read all into Sigma_18;
Sigma18=diag(Sigma_18);
create Sigma18 from Sigma18; append from
Sigma18; quit;
Proc iml;
use P; read all into P;
use Sigma18; read all into Sigma18;
P18=P*Sigma18*t(P);
create P18 from P18; append from P18; quit;

Data Sigma_19;
set s1;
if i=19;
keep var;
run;
Proc iml;
use Sigma_19; read all into Sigma_19;
Sigma19=diag(Sigma_19);
create Sigma19 from Sigma19; append from
Sigma19; quit;
Proc iml;
use P; read all into P;
use Sigma19; read all into Sigma19;
P19=P*Sigma19*t(P);
create P19 from P19; append from P19; quit;

Data Sigma_20;
set s1;
if i=20;
keep var;
run;
Proc iml;
use Sigma_20; read all into Sigma_20;
Sigma20=diag(Sigma_20);
create Sigma20 from Sigma20; append from
Sigma20; quit;
Proc iml;
use P; read all into P;
use Sigma20; read all into Sigma20;
P20=P*Sigma20*t(P);
create P20 from P20; append from P20; quit;

Data Sigma_21;
set s1;
if i=21;
keep var;
run;
Proc iml;
use Sigma_21; read all into Sigma_21;
Sigma21=diag(Sigma_21);
create Sigma21 from Sigma21; append from
Sigma21; quit;
Proc iml;
use P; read all into P;
use Sigma21; read all into Sigma21;
P21=P*Sigma21*t(P);
create P21 from P21; append from P21; quit;

Data Sigma_22;
set s1;
if i=22;
keep var;
run;
Proc iml;
use Sigma_22; read all into Sigma_22;
Sigma22=diag(Sigma_22);
create Sigma22 from Sigma22; append from
Sigma22; quit;
Proc iml;
use P; read all into P;
use Sigma22; read all into Sigma22;
P22=P*Sigma22*t(P);
create P22 from P22; append from P22; quit;

Data Sigma_23;
set s1;
if i=23;
keep var;
run;
Proc iml;
use Sigma_23; read all into Sigma_23;
Sigma23=diag(Sigma_23);
create Sigma23 from Sigma23; append from
Sigma23; quit;
Proc iml;
use P; read all into P;
use Sigma23; read all into Sigma23;
P23=P*Sigma23*t(P);
create P23 from P23; append from P23; quit;

Data Sigma_24;
set s1;
if i=24;
keep var;
run;
Proc iml;
use Sigma_24; read all into Sigma_24;
Sigma24=diag(Sigma_24);
create Sigma24 from Sigma24; append from
Sigma24; quit;
Proc iml;
use P; read all into P;
use Sigma24; read all into Sigma24;
P24=P*Sigma24*t(P);
create P24 from P24; append from P24; quit;

Data Sigma_25;
set s1;
if i=25;
keep var;
run;
Proc iml;
use Sigma_25; read all into Sigma_25;
Sigma25=diag(Sigma_25);
```

```
create Sigma25 from Sigma25; append from
Sigma25; quit;
Proc iml;
use P; read all into P;
use Sigma25; read all into Sigma25;
P25=P*Sigma25*t(P);
create P25 from P25; append from P25; quit;

Data Sigma_26;
set s1;
if i=26;
keep var;
run;
Proc iml;
use Sigma_26; read all into Sigma_26;
Sigma26=diag(Sigma_26);
create Sigma26 from Sigma26; append from
Sigma26; quit;
Proc iml;
use P; read all into P;
use Sigma26; read all into Sigma26;
P26=P*Sigma26*t(P);
create P26 from P26; append from P26; quit;

Data Sigma_27;
set s1;
if i=27;
keep var;
run;
Proc iml;
use Sigma_27; read all into Sigma_27;
Sigma27=diag(Sigma_27);
create Sigma27 from Sigma27; append from
Sigma27; quit;
Proc iml;
use P; read all into P;
use Sigma27; read all into Sigma27;
P27=P*Sigma27*t(P);
create P27 from P27; append from P27; quit;

Data Sigma_28;
set s1;
if i=28;
keep var;
run;
Proc iml;
use Sigma_28; read all into Sigma_28;
Sigma28=diag(Sigma_28);
create Sigma28 from Sigma28; append from
Sigma28; quit;
Proc iml;
use P; read all into P;
use Sigma28; read all into Sigma28;
P28=P*Sigma28*t(P);
create P28 from P28; append from P28; quit;

Proc iml;
use P1; read all into P1;
use P2; read all into P2;
use P3; read all into P3;
use P4; read all into P4;
use P5; read all into P5;
use P6; read all into P6;
use P7; read all into P7;
use P8; read all into P8;
use P9; read all into P9;
use P10; read all into P10;
use P11; read all into P11;
use P12; read all into P12;
use P13; read all into P13;
use P14; read all into P14;
blockPP1=BLOCK(P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,
P11,P12,P13,P14);
create blockPP1 from blockPP1; append from
blockPP1; quit;
Proc iml;
use P15; read all into P15;
use P16; read all into P16;
use P17; read all into P17;
use P18; read all into P18;
use P19; read all into P19;
use P20; read all into P20;
use P21; read all into P21;
use P22; read all into P22;
use P23; read all into P23;
use P24; read all into P24;
use P25; read all into P25;
use P26; read all into P26;
use P27; read all into P27;
use P28; read all into P28;
blockPP2=BLOCK(P15,P16,P17,P18,P19,P20,P21,P22
,P23,P24,P25,P26,P27,P28);
create blockPP2 from blockPP2; append from
blockPP2; quit;

Proc iml;
use blockPP1; read all into blockPP1;
use blockPP2; read all into blockPP2;
blockPP=BLOCK(blockPP1,blockPP2);
create blockPP from blockPP; append from
blockPP; quit;

/* *** Fim *** */

Proc iml;
use H1; read all into H1;
HH=H1*t(H1);
eval=eigval(HH);
create eval from eval; append from eval; quit;

Data id;
do id=1 to 196;
output; end; run;

Data eval;
merge id eval;
run;

Data new;
set eval;
by id;
retain count 0;
if first.id then count=0;
if col1 > 0.000001 then count = count + 1;
run;

Proc means data=new sum noprint;
var count;
output out=caracH1 sum=caracH1;
run;

Data caracH1;
set caracH1;
keep caracH1;
run;

Proc iml;
use blockPP; read all into blockPP;
use blockID; read all into blockID;
use Hs1; read all into Hs1;
use rg1_; read all into rg1_;
use caracH1; read all into caracH1;
varu= (rg1_ -  trace((blockID -
Hs1)*blockPP))*(1/(28*6 - caracH1));
create varu from varu; append from varu; quit;

/*Termina aqui o cálculo da variância
sigma2_u*/

/** Cálculo da matriz: z2= c^(-0.5)*f'*P*Teta
- var. dependente do modelo transformado **/

Proc iml;
use tetad1; read all into tetad1;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y1=(1/sqrt(c))*t(f)*P*tetad1;
create y1 from y1; append from y1; quit;
Proc iml;
use tetad2; read all into tetad2;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y2=(1/sqrt(c))*t(f)*P*tetad2;
create y2 from y2; append from y2; quit;
Proc iml;
use tetad3; read all into tetad3;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y3=(1/sqrt(c))*t(f)*P*tetad3;
create y3 from y3; append from y3; quit;
Proc iml;
use tetad4; read all into tetad4;
use P; read all into P;
```

```
use f; read all into f;
use c; read all into c;
y4=(1/sqrt(c))*t(f)*P*tetad4;
create y4 from y4; append from y4; quit;
Proc iml;
use tetad5; read all into tetad5;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y5=(1/sqrt(c))*t(f)*P*tetad5;
create y5 from y5; append from y5; quit;
Proc iml;
use tetad6; read all into tetad6;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y6=(1/sqrt(c))*t(f)*P*tetad6;
create y6 from y6; append from y6; quit;
Proc iml;
use tetad7; read all into tetad7;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y7=(1/sqrt(c))*t(f)*P*tetad7;
create y7 from y7; append from y7; quit;
Proc iml;
use tetad8; read all into tetad8;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y8=(1/sqrt(c))*t(f)*P*tetad8;
create y8 from y8; append from y8; quit;
Proc iml;
use tetad9; read all into tetad9;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y9=(1/sqrt(c))*t(f)*P*tetad9;
create y9 from y9; append from y9; quit;
Proc iml;
use tetad10; read all into tetad10;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y10=(1/sqrt(c))*t(f)*P*tetad10;
create y10 from y10; append from y10; quit;
Proc iml;
use tetad11; read all into tetad11;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y11=(1/sqrt(c))*t(f)*P*tetad11;
create y11 from y11; append from y11; quit;
Proc iml;
use tetad12; read all into tetad12;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y12=(1/sqrt(c))*t(f)*P*tetad12;
create y12 from y12; append from y12; quit;
Proc iml;
use tetad13; read all into tetad13;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y13=(1/sqrt(c))*t(f)*P*tetad13;
create y13 from y13; append from y13; quit;
Proc iml;
use tetad14; read all into tetad14;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y14=(1/sqrt(c))*t(f)*P*tetad14;
create y14 from y14; append from y14; quit;
Proc iml;
use tetad15; read all into tetad15;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y15=(1/sqrt(c))*t(f)*P*tetad15;
create y15 from y15; append from y15; quit;
Proc iml;
use tetad16; read all into tetad16;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y16=(1/sqrt(c))*t(f)*P*tetad16;
create y16 from y16; append from y16; quit;
Proc iml;
use tetad17; read all into tetad17;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y17=(1/sqrt(c))*t(f)*P*tetad17;
create y17 from y17; append from y17; quit;
Proc iml;
use tetad18; read all into tetad18;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y18=(1/sqrt(c))*t(f)*P*tetad18;
create y18 from y18; append from y18; quit;
Proc iml;
use tetad19; read all into tetad19;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y19=(1/sqrt(c))*t(f)*P*tetad19;
create y19 from y19; append from y19; quit;
Proc iml;
use tetad20; read all into tetad20;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y20=(1/sqrt(c))*t(f)*P*tetad20;
create y20 from y20; append from y20; quit;
Proc iml;
use tetad21; read all into tetad21;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y21=(1/sqrt(c))*t(f)*P*tetad21;
create y21 from y21; append from y21; quit;
Proc iml;
use tetad22; read all into tetad22;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y22=(1/sqrt(c))*t(f)*P*tetad22;
create y22 from y22; append from y22; quit;
Proc iml;
use tetad23; read all into tetad23;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y23=(1/sqrt(c))*t(f)*P*tetad23;
create y23 from y23; append from y23; quit;
Proc iml;
use tetad24; read all into tetad24;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y24=(1/sqrt(c))*t(f)*P*tetad24;
create y24 from y24; append from y24; quit;
Proc iml;
use tetad25; read all into tetad25;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y25=(1/sqrt(c))*t(f)*P*tetad25;
create y25 from y25; append from y25; quit;
Proc iml;
use tetad26; read all into tetad26;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y26=(1/sqrt(c))*t(f)*P*tetad26;
create y26 from y26; append from y26; quit;
Proc iml;
use tetad27; read all into tetad27;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y27=(1/sqrt(c))*t(f)*P*tetad27;
create y27 from y27; append from y27; quit;
Proc iml;
use tetad28; read all into tetad28;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y28=(1/sqrt(c))*t(f)*P*tetad28;
create y28 from y28; append from y28; quit;
run;

Data z2;
```

```
set y1 y2 y3 y4 y5 y6 y7 y8 y9 y10 y11 y12 y13
y14 y15 y16 y17 y18 y19 y20 y21 y22 y23 y24
y25 y26 y27 y28;
run;

/** Cálculo da matriz: H2= c^(-0.5)*f'*P*X -
var. independentes do modelo transformado **/

Proc iml;
use xd1; read all into xd1;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x1=(1/sqrt(c))*t(f)*P*xd1;
create x1 from x1; append from x1; quit;
Proc iml;
use xd2; read all into xd2;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x2=(1/sqrt(c))*t(f)*P*xd2;
create x2 from x2; append from x2; quit;
Proc iml;
use xd3; read all into xd3;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x3=(1/sqrt(c))*t(f)*P*xd3;
create x3 from x3; append from x3; quit;
Proc iml;
use xd4; read all into xd4;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x4=(1/sqrt(c))*t(f)*P*xd4;
create x4 from x4; append from x4; quit;
Proc iml;
use xd5; read all into xd5;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x5=(1/sqrt(c))*t(f)*P*xd5;
create x5 from x5; append from x5; quit;
Proc iml;
use xd6; read all into xd6;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x6=(1/sqrt(c))*t(f)*P*xd6;
create x6 from x6; append from x6; quit;
Proc iml;
use xd7; read all into xd7;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x7=(1/sqrt(c))*t(f)*P*xd7;
create x7 from x7; append from x7; quit;
Proc iml;
use xd8; read all into xd8;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x8=(1/sqrt(c))*t(f)*P*xd8;
create x8 from x8; append from x8; quit;
Proc iml;
use xd9; read all into xd9;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x9=(1/sqrt(c))*t(f)*P*xd9;
create x9 from x9; append from x9; quit;
Proc iml;
use xd10; read all into xd10;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x10=(1/sqrt(c))*t(f)*P*xd10;
create x10 from x10; append from x10; quit;
Proc iml;
use xd11; read all into xd11;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x11=(1/sqrt(c))*t(f)*P*xd11;
create x11 from x11; append from x11; quit;
Proc iml;
use xd12; read all into xd12;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x12=(1/sqrt(c))*t(f)*P*xd12;
create x12 from x12; append from x12; quit;
Proc iml;
use xd13; read all into xd13;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x13=(1/sqrt(c))*t(f)*P*xd13;
create x13 from x13; append from x13; quit;
Proc iml;
use xd14; read all into xd14;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x14=(1/sqrt(c))*t(f)*P*xd14;
create x14 from x14; append from x14; quit;
Proc iml;
use xd15; read all into xd15;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x15=(1/sqrt(c))*t(f)*P*xd15;
create x15 from x15; append from x15; quit;
Proc iml;
use xd16; read all into xd16;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x16=(1/sqrt(c))*t(f)*P*xd16;
create x16 from x16; append from x16; quit;
Proc iml;
use xd17; read all into xd17;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x17=(1/sqrt(c))*t(f)*P*xd17;
create x17 from x17; append from x17; quit;
Proc iml;
use xd18; read all into xd18;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x18=(1/sqrt(c))*t(f)*P*xd18;
create x18 from x18; append from x18; quit;
Proc iml;
use xd19; read all into xd19;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x19=(1/sqrt(c))*t(f)*P*xd19;
create x19 from x19; append from x19; quit;
Proc iml;
use xd20; read all into xd20;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x20=(1/sqrt(c))*t(f)*P*xd20;
create x20 from x20; append from x20; quit;
Proc iml;
use xd21; read all into xd21;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x21=(1/sqrt(c))*t(f)*P*xd21;
create x21 from x21; append from x21; quit;
Proc iml;
use xd22; read all into xd22;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x22=(1/sqrt(c))*t(f)*P*xd22;
create x22 from x22; append from x22; quit;
Proc iml;
use xd23; read all into xd23;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x23=(1/sqrt(c))*t(f)*P*xd23;
create x23 from x23; append from x23; quit;
Proc iml;
use xd24; read all into xd24;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x24=(1/sqrt(c))*t(f)*P*xd24;
create x24 from x24; append from x24; quit;
```

```
Proc iml;
use xd25; read all into xd25;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x25=(1/sqrt(c))*t(f)*P*xd25;
create x25 from x25; append from x25; quit;
Proc iml;
use xd26; read all into xd26;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x26=(1/sqrt(c))*t(f)*P*xd26;
create x26 from x26; append from x26; quit;
Proc iml;
use xd27; read all into xd27;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x27=(1/sqrt(c))*t(f)*P*xd27;
create x27 from x27; append from x27; quit;
Proc iml;
use xd28; read all into xd28;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x28=(1/sqrt(c))*t(f)*P*xd28;
create x28 from x28; append from x28; quit;
run;

Data H2;
set x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13
x14 x15
x16 x17 x18 x19 x20 x21 x22 x23 x24 x25 x26
x27 x28;
run;

/** Regressão de z2(Y) sobre H2(X) **/

Data z2;
set z2;
rename col1=col0;
run;

Data reg2;
merge z2 H2;
run;

proc reg data=reg2 noprint;
        model col0=col1 col2 / noint;
        output out=rg2  residual=r2;
run; quit;

Data rg2;
set rg2;
qr2=r2*r2;
Keep qr2; run;

Proc means data=rg2 sum noprint;
var qr2;
output out=rg2_ sum=sqr2;
run;

Data rg2_;
set rg2_;
keep sqr2; run;

/** Cálculo da variância sigma2_v **/

Proc iml;
use H2; read all into H2;
Ident=I(28);
PH2=Ident - H2*ginv(t(H2)*H2)*t(H2);
create PH2 from PH2; append from PH2; quit;

Proc iml;
use f; read all into f;
use P1; read all into P1;
use P2; read all into P2;
use P3; read all into P3;
use P4; read all into P4;
use P5; read all into P5;
use P6; read all into P6;
use P7; read all into P7;
use P8; read all into P8;
use P9; read all into P9;
use P10; read all into P10;
use P11; read all into P11;
use P12; read all into P12;
use P13; read all into P13;
use P14; read all into P14;
use P15; read all into P15;
use P16; read all into P16;
use P17; read all into P17;
use P18; read all into P18;
use P19; read all into P19;
use P20; read all into P20;
use P21; read all into P21;
use P22; read all into P22;
use P23; read all into P23;
use P24; read all into P24;
use P25; read all into P25;
use P26; read all into P26;
use P27; read all into P27;
use P28; read all into P28;
use c; read all into c;
cfp1=t(f)*P1*f*(1 / c);
cfp2=t(f)*P2*f*(1 / c);
cfp3=t(f)*P3*f*(1 / c);
cfp4=t(f)*P4*f*(1 / c);
cfp5=t(f)*P5*f*(1 / c);
cfp6=t(f)*P6*f*(1 / c);
cfp7=t(f)*P7*f*(1 / c);
cfp8=t(f)*P8*f*(1 / c);
cfp9=t(f)*P9*f*(1 / c);
cfp10=t(f)*P10*f*(1 / c);
cfp11=t(f)*P11*f*(1 / c);
cfp12=t(f)*P12*f*(1 / c);
cfp13=t(f)*P13*f*(1 / c);
cfp14=t(f)*P14*f*(1 / c);
cfp15=t(f)*P15*f*(1 / c);
cfp16=t(f)*P16*f*(1 / c);
cfp17=t(f)*P17*f*(1 / c);
cfp18=t(f)*P18*f*(1 / c);
cfp19=t(f)*P19*f*(1 / c);
cfp20=t(f)*P20*f*(1 / c);
cfp21=t(f)*P21*f*(1 / c);
cfp22=t(f)*P22*f*(1 / c);
cfp23=t(f)*P23*f*(1 / c);
cfp24=t(f)*P24*f*(1 / c);
cfp25=t(f)*P25*f*(1 / c);
cfp26=t(f)*P26*f*(1 / c);
cfp27=t(f)*P27*f*(1 / c);
cfp28=t(f)*P28*f*(1 / c);
cfp_=cfp1//cfp2//cfp3//cfp4//cfp5//cfp6//cfp7/
/cfp8//cfp9//cfp10//cfp11//cfp12//cfp13//cfp14
//cfp15//cfp16//cfp17//cfp18//cfp19//cfp20//cf
p21//cfp22//cfp23//cfp24//cfp25//cfp26//cfp27/
/cfp28;
cfp=diag(cfp_);
create cfp from cfp; append from cfp; quit;

/* *** Fim *** */

Proc iml;
use H2; read all into H2;
FF=H2*t(H2);
eval1=eigval(FF);
create eval1 from eval1; append from eval1;
quit;

Data id1;
do id=1 to 28;
output; end;
run;

Data eval1;
merge id1 eval1;
run;

Data new1;
set eval1;
by id;
retain count 0;
if first.id then count=0;
if col1 > 0.000001 then count = count + 1;
run;

Proc means data=new1 sum noprint;
var count;
output out=caracH2 sum=caracH2;
run;

Data caracH2;
set caracH2;
keep caracH2;  /* Caracteristica de H2 */
```

```
run;

Proc iml;
use PH2; read all into PH2;
use cfp; read all into cfp;
use c; read all into c;
use rg2_; read all into rg2_;
use caracH2; read all into caracH2;
use varu; read all into varu;
use Binv; read all into Binv;
varv= (rg2_ -  trace(PH2*cfp) - varu*(28 -
caracH2))/(c*trace(PH2*Binv));
create varv from varv; append from varv; quit;

/*Termina aqui o cálculo da variância
sigma2_v*/
/* Truncagem a zero das variâncias sigma2_u e
sigma2_v */

Data varu;
set varu;
sigma2u= max( col1 , 0);
keep sigma2u;
run;
Data varv;
set varv;
sigma2v= max( col1 , 0);
keep sigma2v;
run;

/* Construção da matriz G, sendo para tal
necessária a matriz Gama */

/*** Matriz GAMA  ***/

Data ro_1;
set dataro;
ro_1= 1-ro*ro;
keep ro_1;
Data ro_2;
set dataro;
ro_2= ro*ro;
keep ro_2;
Data ro_3;
set dataro;
ro_3= ro*ro*ro;
keep ro_3;
Data ro_4;
set dataro;
ro_4= ro*ro*ro*ro;
keep ro_4;
Data ro_5;
set dataro;
ro_5= ro*ro*ro*ro*ro;
keep ro_5;
Data ro_6;
set dataro;
ro_6= ro*ro*ro*ro*ro*ro;
keep ro_6;
run;

Proc iml;
use ro_1; read all into ro_1;
use dataro; read all into dataro;
use ro_2; read all into ro_2;
use ro_3; read all into ro_3;
use ro_4; read all into ro_4;
use ro_5; read all into ro_5;
use ro_6; read all into ro_6;
I=I(7);
I1={0 1 0 0 0 0 0, 0 0 1 0 0 0 0, 0 0 0 1 0 0
0, 0 0 0 0 1 0 0, 0 0 0 0 0 1 0, 0 0 0 0 0 0
1, 0 0 0 0 0 0 0};
I2={0 0 1 0 0 0 0, 0 0 0 1 0 0 0, 0 0 0 0 1 0
0, 0 0 0 0 0 1 0, 0 0 0 0 0 0 1, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I3={0 0 0 1 0 0 0, 0 0 0 0 1 0 0, 0 0 0 0 0 1
0, 0 0 0 0 0 0 1, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I4={0 0 0 0 1 0 0, 0 0 0 0 0 1 0, 0 0 0 0 0 0
1, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I5={0 0 0 0 0 1 0, 0 0 0 0 0 0 1, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I6={0 0 0 0 0 0 1, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
```

```
Gama = I/ro_1 + dataro*I1/ro_1 +
t(dataro*I1/ro_1) + ro_2*I2/ro_1 +
t(ro_2*I2/ro_1) + ro_3*I3/ro_1 +
t(ro_3*I3/ro_1) + ro_4*I4/ro_1 +
t(ro_4*I4/ro_1) + ro_5*I5/ro_1 +
t(ro_5*I5/ro_1) + ro_6*I6/ro_1 +
t(ro_6*I6/ro_1);
create Gama from Gama;
append from Gama;
quit;

/*** Matriz G1 ***/

Proc iml;
use varv; read all into varv;
use Binv; read all into Binv;
G1=varv*Binv;
create G1 from G1; append from G1; quit;

/*** Matriz G2 ***/

Proc iml;
use varu; read all into varu;
use Gama; read all into Gama;
Ident=I(28);
G2=varu*(Ident@Gama);
create G2 from G2; append from G2; quit;

/*** Matriz G ***/

Proc iml;
use G1; read all into G1;
use G2; read all into G2;
Mat_GG=block (G1 , G2);
create Mat_GG from Mat_GG; append from Mat_GG;
quit;

Proc iml;
row=t(do(1,224,1));
create row from row; append from row; quit;

Data row;
set row;
rename col1=row;
run;

Data Mat_GG;
merge row Mat_GG;
run;

/* Construção do ficheiro com os parâmetros de
variância dos erros da sondagem */

Proc sort data=s1;
by i t;
run;

Data s2;
set s1;
keep var;
run;

Data par;
set s2;
rename var=est;
run;

/*********************************************
*********************************************/
/** ESTIMAÇÃO DO MODELO DE LUÍS PEREIRA *****/
/*********************************************
*********************************************/

Proc sort data=iabhipth1 out= cron;
by i t; run;

/* O Proc mixed só permite incluir uma matriz
G, independentemente no número de RANDOM
definidos.
A matriz G definida tem que ser a matriz
completa para todo o modelo */

Proc mixed data=cron MMEqSol noprofile
noclprint;
        class i t;
        model y=x / solution ddfm=residual;
        random i;
```

```
        random i*t / type=un Gdata=Mat_GG G V
S;
        repeated /group=i*t;
        parms /parmsdata=par noiter;
make 'solutionf' out=beta;
make 'solutionr' out=u;
make 'Fitstatistics' out=aic;
make 'G' out=Mat_Gaux;
make 'V' out=Mat_V;
run; quit;

/********************************************/
/********* CÁLCULO DAS ESTIMATIVAS DO
PARÂMETRO DE INTERESSE *************/
/********************************************/

Data id;
do id=1 to 2;
output; end; run;

Data beta;
merge id beta; run;

Data beta0;
set beta;
if id=1;
rename estimate=b0;
keep estimate; run;

Data beta1;
set beta;
if id=2;
rename estimate=b1;
keep estimate; run;

Data betan;
merge beta0 beta1;
_type_=0; run;

/****** Efeitos Aleatórios: v ******/

Data u;
set u;
if estimate='.' then estimate=0;
run;

Data v;
set u;
if Effect='i';
keep i estimate;
run;

Proc sort data=v;
by i;
run;

Proc iml;
use v; read all into v;
one=J(7,1,1);
v1=v@one;
create v1 from v1; append from v1; quit;

Data v1;
set v1;
_type_=0;
rename col2=v;
run;

/****** Efeitos Aleatórios: u ******/

Data u1;
set u;
if Effect='i*t';
_type_=0;
keep i t estimate _type_;
rename estimate=u;
run;

Proc sort data=u1;
by i t;
run;

Data beta_vu;
merge betan v1 u1;
by _type_;
run;

/** Estimação do parâmetro de interesse em
cada domínio **/

Proc sort data=cron;
by i t;
run;

Data cron;
merge cron beta_vu;
by i t;
run;

Data crono_resf;
set cron;
e = b0 + x*b1 + v + u;
keep i y t _type_ x b0 b1 u v e;
run;

Data est_PC;
set crono_resf;
simul=&i;
drop _type_;
run;

DM 'CLEAR OUTPUT';
DM 'CLEAR LOG';

/*********************************************
*********************************************/
/*********************************************
*********************************************/
/* Estimação do EQMP do EBLUP utilizando a
aproximação analítica */
/*********************************************
*********************************************/
/*********************************************
*********************************************/

/*** MATRIZ X ***/

Proc sort data=cron out=cronologico;
by i t;
run;

Data XX1;
set cronologico;
keep x;
run;

Proc iml;
J=J(196,1,1);
create J from J;
append from J;
quit;

Proc iml;
use J;
read all into J;
use XX1;
read all into XX1;
X=insert(J,XX1,0,2);
create X from X;
append from X;
quit;

/*** Colunas da Matriz GAMA ***/

Proc iml;
use Gama; read all into Gama;
Gama1=Gama[,1];
create Gama1 from Gama1;
append from Gama1;
quit;
Proc iml;
use Gama; read all into Gama;
Gama2=Gama[,2];
create Gama2 from Gama2;
append from Gama2;
quit;
Proc iml;
use Gama; read all into Gama;
Gama3=Gama[,3];
create Gama3 from Gama3;
append from Gama3;
quit;
Proc iml;
use Gama; read all into Gama;
Gama4=Gama[,4];
```

```
create Gama4 from Gama4;
append from Gama4;
quit;
Proc iml;
use Gama; read all into Gama;
Gama5=Gama[,5];
create Gama5 from Gama5;
append from Gama5;
quit;
Proc iml;
use Gama; read all into Gama;
Gama6=Gama[,6];
create Gama6 from Gama6;
append from Gama6;
quit;
Proc iml;
use Gama; read all into Gama;
Gama7=Gama[,7];
create Gama7 from Gama7;
append from Gama7;
quit;

/*** MATRIZ V ***/

Data MatV;
set Mat_v;
drop Index row;
run;

Proc iml;
use MatV; read all into MatV;
MatVinv=inv(MatV);
create MatVinv from MatVinv;
append from MatVinv; quit;

/***************************/
/****** Cálculo do g1 ******/
/***************************/

Data Mat_gg;
set Mat_gg;
drop row;
run;

Proc iml;
use MatVinv; read all into MatVinv;
use Mat_gg; read all into Mat_gg;
use PC.Z; read all into Z;
G1_aux=Z*(Mat_gg -
Mat_gg*t(Z)*MatVinv*Z*Mat_gg)*t(Z);
create G1_aux from G1_aux;
append from G1_aux;
quit;

Proc iml;
use G1_aux;
read all into G1_aux;
G1=vecdiag(G1_aux);
create G1 from G1;
append from G1;
quit;

Data g1;
set g1;
rename col1=g1;
run;

/***************************/
/****** Cálculo do g2 ******/
/***************************/

Proc iml;
use X; read all into X;
use MatVinv; read all into MatVinv;
use Mat_gg; read all into Mat_gg;
use PC.Z; read all into Z;
A=I(196);
G2_aux=t(t(X) -
t(X)*MatVinv*Z*Mat_gg*t(Z))*inv(t(X)*MatVinv*X
)*(t(X) - t(X)*MatVinv*Z*Mat_gg*t(Z));
create G2_aux from G2_aux;
append from G2_aux;
quit;

Proc iml;
use G2_aux;
read all into G2_aux;
G2=vecdiag(G2_aux);
create G2 from G2;
append from G2;
quit;

Data g2;
set g2;
rename col1=g2;
run;

/***************************/
/****** Cálculo do g3 ******/
/***************************/

/* O Cálculo do g3 tem que ser feito
individualmente para cada trimestre EM CADA
DOMÍNIO. Note-se que o vector coluna "variante
de sigma_i" varia em função do DOMINIO e que o
vector "zeta_it" varia em função do DOMINIO e
do TEMPO.
Basta calcular a matriz A para cada trimestre
de um determinado domínio, e depois replicar
para os restantes domínios, que é sempre igual
*/

/* É necessário definir as 28 colunas da
matriz inv(B) */

Proc iml;
use Binv; read all into Binv;
Binv1=Binv[,1];
create Binv1 from Binv1;
append from Binv1; quit;
Proc iml;
use Binv; read all into Binv;
Binv2=Binv[,2];
create Binv2 from Binv2;
append from Binv2; quit;
Proc iml;
use Binv; read all into Binv;
Binv3=Binv[,3];
create Binv3 from Binv3;
append from Binv3; quit;
Proc iml;
use Binv; read all into Binv;
Binv4=Binv[,4];
create Binv4 from Binv4;
append from Binv4; quit;
Proc iml;
use Binv; read all into Binv;
Binv5=Binv[,5];
create Binv5 from Binv5;
append from Binv5; quit;
Proc iml;
use Binv; read all into Binv;
Binv6=Binv[,6];
create Binv6 from Binv6;
append from Binv6; quit;
Proc iml;
use Binv; read all into Binv;
Binv7=Binv[,7];
create Binv7 from Binv7;
append from Binv7; quit;
Proc iml;
use Binv; read all into Binv;
Binv8=Binv[,8];
create Binv8 from Binv8;
append from Binv8; quit;
Proc iml;
use Binv; read all into Binv;
Binv9=Binv[,9];
create Binv9 from Binv9;
append from Binv9; quit;
Proc iml;
use Binv; read all into Binv;
Binv10=Binv[,10];
create Binv10 from Binv10;
append from Binv10; quit;
Proc iml;
use Binv; read all into Binv;
Binv11=Binv[,11];
create Binv11 from Binv11;
append from Binv11; quit;
Proc iml;
use Binv; read all into Binv;
Binv12=Binv[,12];
create Binv12 from Binv12;
append from Binv12; quit;
Proc iml;
```

```
use Binv; read all into Binv;
Binv13=Binv[,13];
create Binv13 from Binv13;
append from Binv13; quit;
Proc iml;
use Binv; read all into Binv;
Binv14=Binv[,14];
create Binv14 from Binv14;
append from Binv14; quit;
Proc iml;
use Binv; read all into Binv;
Binv15=Binv[,15];
create Binv15 from Binv15;
append from Binv15; quit;
Proc iml;
use Binv; read all into Binv;
Binv16=Binv[,16];
create Binv16 from Binv16;
append from Binv16; quit;
Proc iml;
use Binv; read all into Binv;
Binv17=Binv[,17];
create Binv17 from Binv17;
append from Binv17; quit;
Proc iml;
use Binv; read all into Binv;
Binv18=Binv[,18];
create Binv18 from Binv18;
append from Binv18; quit;
Proc iml;
use Binv; read all into Binv;
Binv19=Binv[,19];
create Binv19 from Binv19;
append from Binv19; quit;
Proc iml;
use Binv; read all into Binv;
Binv20=Binv[,20];
create Binv20 from Binv20;
append from Binv20; quit;
Proc iml;
use Binv; read all into Binv;
Binv21=Binv[,21];
create Binv21 from Binv21;
append from Binv21; quit;
Proc iml;
use Binv; read all into Binv;
Binv22=Binv[,22];
create Binv22 from Binv22;
append from Binv22; quit;
Proc iml;
use Binv; read all into Binv;
Binv23=Binv[,23];
create Binv23 from Binv23;
append from Binv23; quit;
Proc iml;
use Binv; read all into Binv;
Binv24=Binv[,24];
create Binv24 from Binv24;
append from Binv24; quit;
Proc iml;
use Binv; read all into Binv;
Binv25=Binv[,25];
create Binv25 from Binv25;
append from Binv25; quit;
Proc iml;
use Binv; read all into Binv;
Binv26=Binv[,26];
create Binv26 from Binv26;
append from Binv26; quit;
Proc iml;
use Binv; read all into Binv;
Binv27=Binv[,27];
create Binv27 from Binv27;
append from Binv27; quit;
Proc iml;
use Binv; read all into Binv;
Binv28=Binv[,28];
create Binv28 from Binv28;
append from Binv28; quit;

/*******************************************/
/* Trabalho sobre a matriz A_it - cálculo de
196 matrizes: uma para cada trimestre*dominio
*/
/*******************************************/

/* DOMÍNIO 1 */

Proc iml;
I1={1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
create I1 from I1; append from I1; quit;

/* 1º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama1; read all into Gama1;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama1) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama1) ) )
*MatVinv* ( (I1@Gama1) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama1)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama1; read all into Gama1;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama1) ) )
*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama1) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama1; read all into Gama1;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama1) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama1) ) )*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama1) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A11_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A11_ from A11_; append from A11_; quit;

/* 2º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama2; read all into Gama2;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama2) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama2) ) )
*MatVinv* ( (I1@Gama2) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama2)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
```

```
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama2; read all into Gama2;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama2) ) )
*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama2) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama2; read all into Gama2;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama2) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama2) ) )*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama2) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A12_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A12_ from A12_; append from A12_; quit;

/* 3º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama3; read all into Gama3;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama3) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama3) ) )
*MatVinv* ( (I1@Gama3) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama3)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama3; read all into Gama3;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama3) ) )
*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama3) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama3; read all into Gama3;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama3) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama3) ) )*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama3) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A13_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A13_ from A13_; append from A13_; quit;

/* 4º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama4; read all into Gama4;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama4) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama4) ) )
*MatVinv* ( (I1@Gama4) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama4)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama4; read all into Gama4;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama4) ) )
*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama4) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama4; read all into Gama4;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama4) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama4) ) )*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama4) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A14_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A14_ from A14_; append from A14_; quit;

/* 5º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
```

```
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama5; read all into Gama5;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama5) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama5) ) )
*MatVinv* ( (I1@Gama5) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama5)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama5; read all into Gama5;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama5) ) )
*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama5) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama5; read all into Gama5;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama5) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama5) ) )*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama5) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A15_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A15_ from A15_; append from A15_; quit;

/* 6º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama6; read all into Gama6;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama6) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama6) ) )
*MatVinv* ( (I1@Gama6) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama6)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama6; read all into Gama6;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama6) ) )
*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama6) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama6; read all into Gama6;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama6) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama6) ) )*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama6) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A16_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A16_ from A16_; append from A16_; quit;

/* 7º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama7; read all into Gama7;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama7) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama7) ) )
*MatVinv* ( (I1@Gama7) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama7)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama7; read all into Gama7;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama7) ) )
*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama7) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama7; read all into Gama7;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama7) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama7) ) )*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama7) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
```

```
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A17_=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A17_ from A17_; append from A17_; quit;

/* DOMÍNIO 2 */  (…)
/* DOMÍNIO 3 */  (…)
/* DOMÍNIO 4 */  (…)
 (…)
/* DOMÍNIO 26 */  (…)
/* DOMÍNIO 27 */  (…)
/* DOMÍNIO 28 */  (…)

/* *** Fim *** */

/**********************************/
/* Cálculo da matriz SIGMA* (2*2) */
/**********************************/

/* OBS.: Antes de calcular as covariâncias da
matriz SIGMA*, é necessário calcular as
matrizes C, C*, C1 e C2. Note-se que a matriz
V (que depois funciona como a matriz OMEGA já
está calculada, pois é a matriz Cov(Y) e note-
se também que não é necessário calcular a */

/* MatC */

Proc iml;
use D; read all into D;
use P; read all into P;
Ident=I(7);
IDP=(Ident - D)*P; /* Matriz (I - D)*P */
create IDP from IDP; append from IDP; quit;

Proc iml;
use IDP; read all into IDP;
blockIDP1=BLOCK(IDP,IDP,IDP,IDP,IDP,IDP,IDP,ID
P,IDP,IDP,IDP,IDP,IDP,IDP);
create blockIDP1 from blockIDP1; append from
blockIDP1; quit;
Proc iml;
use blockIDP1; read all into blockIDP1;
MatC=BLOCK(blockIDP1,blockIDP1);
create MatC from MatC; append from MatC; quit;

/* MatC* */

Proc iml;
use f; read all into f;
use P; read all into P;
use c; read all into c;
cfP1=(1/sqrt(c))*t(f)*P; /* Matriz c(-
1/2)*fT*P */
create cfP1 from cfP1; append from cfP1; quit;

Proc iml;
use cfP1; read all into cfP1;
blockcfP1=BLOCK(cfP1,cfP1,cfP1,cfP1,cfP1,cfP1,
cfP1,cfP1,cfP1,cfP1,cfP1,cfP1,cfP1,cfP1);
create blockcfP1 from blockcfP1; append from
blockcfP1; quit;
Proc iml;
use blockcfP1; read all into blockcfP1;
MatC_ast=BLOCK(blockcfP1,blockcfP1);
create MatC_ast from MatC_ast; append from
MatC_ast; quit;

/* MatC1 */

Proc iml;
use MatC; read all into MatC;
use X; read all into X;
Ident=I(196);
MatC1=t(MatC)* (Ident - MatC*X*
ginv(t(X)*t(MatC)*MatC*X) *t(X)*t(MatC))
*MatC;
create MatC1 from MatC1; append from MatC1;
quit;

/* MatC2 */

Proc iml;
use MatC_ast; read all into MatC_ast;
use X; read all into X;
Ident=I(28);
MatC2=t(MatC_ast)* (Ident - MatC_ast*X*
ginv(t(X)*t(MatC_ast)*MatC_ast*X)
*t(X)*t(MatC_ast)) *MatC_ast;
create MatC2 from MatC2; append from MatC2;
quit;

/* varu = k1* a'C1a + k2 */

Proc iml;
use caracH1; read all into caracH1;
k1= 1/(28*6 - caracH1);
create k1 from k1; append from k1; quit;

Proc iml;
use blockPP; read all into blockPP;
use blockID; read all into blockID;
use Hs1; read all into Hs1;
use k1; read all into k1;
k2= -trace((blockID - Hs1)*blockPP)*k1;
create k2 from k2; append from k2; quit;

/* varv = k3* a'C2a + k4* a'C1a + k5 */

Proc iml;
use PH2; read all into PH2;
use Binv; read all into Binv;
use c; read all into c;
k3= 1/(c*trace(PH2*Binv));
create k3 from k3; append from k3; quit;

Proc iml;
use k1; read all into k1;
use k3; read all into k3;
use c; read all into c;
use caracH2; read all into caracH2;
k4= -k1*(28-caracH2)*k3;
create k4 from k4; append from k4; quit;

Proc iml;
use PH2; read all into PH2;
use cfp; read all into cfp;
use c; read all into c;
use k2; read all into k2;
use k3; read all into k3;
use caracH2; read all into caracH2;
k5= (-k2*(28-caracH2) - trace(PH2*cfp))*k3;
create k5 from k5; append from k5; quit;

/* SIGMA11 = Var(varu) = 2* k1^2 * tr(C1 * V *
C1 * V) */
/* SIGMA11 */

Proc iml;
use MatC1; read all into MatC1;
use MatV; read all into MatV;
use k1; read all into k1;
Sigma11_=2*k1*k1* trace(MatC1 * MatV * MatC1 *
MatV);
create Sigma11_ from Sigma11_; append from
Sigma11_; quit;

/* SIGMA22 = Var(varv)= 2*k3^2 * tr(C2*V*C2*V)
+ 4* k3*k4* tr(C1*V*C2*V) + 2* k4^2 *
tr(C1*V*C1*V)*/
/* SIGMA22 */

Proc iml;
use MatC1; read all into MatC1;
use MatC2; read all into MatC2;
use MatV; read all into MatV;
use k3; read all into k3;
use k4; read all into k4;
Sigma22_=2*k3*k3*trace(MatC2 * MatV * MatC2 *
MatV) + 4*k3*k4*trace(MatC1 * MatV * MatC2 *
MatV) +
2*k4*k4*trace(MatC1 * MatV * MatC1 * MatV);
create Sigma22_ from Sigma22_; append from
Sigma22_; quit;

/* SIGMA21 = Cov(varu,varv)= 2*k1*k3 *
tr(C1*V*C2*V) + 2* k1*k4* tr(C1*V*C1*V) */
/* SIGMA21 */

Proc iml;
use MatC1; read all into MatC1;
use MatC2; read all into MatC2;
use MatV; read all into MatV;
use k1; read all into k1;
```

```
use k3; read all into k3;
use k4; read all into k4;
Sigma21_=2*k1*k3* trace(MatC1 * MatV * MatC2 *
MatV) + 2*k1*k4* trace(MatC1 * MatV * MatC1 *
MatV);
create Sigma21_ from Sigma21_; append from
Sigma21_; quit;

Proc iml;
use Sigma11_; read all into Sigma11_;
use Sigma22_; read all into Sigma22_;
use Sigma21_; read all into Sigma21_;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
SIGMA=Sigma11_*I1 + Sigma21_*I2 + Sigma21_*I3
+ Sigma22_*I4;
create SIGMA from SIGMA;
append from SIGMA; quit;

/* Cálculo do traço(Ait * SIGMA) */

Proc iml;
use SIGMA; read all into SIGMA;
use A11_; read all into A11_;
use A12_; read all into A12_;
use A13_; read all into A13_;
use A14_; read all into A14_;
use A15_; read all into A15_;
use A16_; read all into A16_;
use A17_; read all into A17_;
use A21_; read all into A21_;
use A22_; read all into A22_;
use A23_; read all into A23_;
use A24_; read all into A24_;
use A25_; read all into A25_;
use A26_; read all into A26_;
use A27_; read all into A27_;
use A31_; read all into A31_;
use A32_; read all into A32_;
use A33_; read all into A33_;
use A34_; read all into A34_;
use A35_; read all into A35_;
use A36_; read all into A36_;
use A37_; read all into A37_;
use A41_; read all into A41_;
use A42_; read all into A42_;
use A43_; read all into A43_;
use A44_; read all into A44_;
use A45_; read all into A45_;
use A46_; read all into A46_;
use A47_; read all into A47_;
use A51_; read all into A51_;
use A52_; read all into A52_;
use A53_; read all into A53_;
use A54_; read all into A54_;
use A55_; read all into A55_;
use A56_; read all into A56_;
use A57_; read all into A57_;
use A61_; read all into A61_;
use A62_; read all into A62_;
use A63_; read all into A63_;
use A64_; read all into A64_;
use A65_; read all into A65_;
use A66_; read all into A66_;
use A67_; read all into A67_;
use A71_; read all into A71_;
use A72_; read all into A72_;
use A73_; read all into A73_;
use A74_; read all into A74_;
use A75_; read all into A75_;
use A76_; read all into A76_;
use A77_; read all into A77_;
use A81_; read all into A81_;
use A82_; read all into A82_;
use A83_; read all into A83_;
use A84_; read all into A84_;
use A85_; read all into A85_;
use A86_; read all into A86_;
use A87_; read all into A87_;
use A91_; read all into A91_;
use A92_; read all into A92_;
use A93_; read all into A93_;
use A94_; read all into A94_;
use A95_; read all into A95_;
use A96_; read all into A96_;
use A97_; read all into A97_;
use A101_; read all into A101_;
use A102_; read all into A102_;
use A103_; read all into A103_;
use A104_; read all into A104_;
use A105_; read all into A105_;
use A106_; read all into A106_;
use A107_; read all into A107_;
G3_11=trace(A11_ * SIGMA);
G3_12=trace(A12_ * SIGMA);
G3_13=trace(A13_ * SIGMA);
G3_14=trace(A14_ * SIGMA);
G3_15=trace(A15_ * SIGMA);
G3_16=trace(A16_ * SIGMA);
G3_17=trace(A17_ * SIGMA);
G3_21=trace(A21_ * SIGMA);
G3_22=trace(A22_ * SIGMA);
G3_23=trace(A23_ * SIGMA);
G3_24=trace(A24_ * SIGMA);
G3_25=trace(A25_ * SIGMA);
G3_26=trace(A26_ * SIGMA);
G3_27=trace(A27_ * SIGMA);
G3_31=trace(A31_ * SIGMA);
G3_32=trace(A32_ * SIGMA);
G3_33=trace(A33_ * SIGMA);
G3_34=trace(A34_ * SIGMA);
G3_35=trace(A35_ * SIGMA);
G3_36=trace(A36_ * SIGMA);
G3_37=trace(A37_ * SIGMA);
G3_41=trace(A41_ * SIGMA);
G3_42=trace(A42_ * SIGMA);
G3_43=trace(A43_ * SIGMA);
G3_44=trace(A44_ * SIGMA);
G3_45=trace(A45_ * SIGMA);
G3_46=trace(A46_ * SIGMA);
G3_47=trace(A47_ * SIGMA);
G3_51=trace(A51_ * SIGMA);
G3_52=trace(A52_ * SIGMA);
G3_53=trace(A53_ * SIGMA);
G3_54=trace(A54_ * SIGMA);
G3_55=trace(A55_ * SIGMA);
G3_56=trace(A56_ * SIGMA);
G3_57=trace(A57_ * SIGMA);
G3_61=trace(A61_ * SIGMA);
G3_62=trace(A62_ * SIGMA);
G3_63=trace(A63_ * SIGMA);
G3_64=trace(A64_ * SIGMA);
G3_65=trace(A65_ * SIGMA);
G3_66=trace(A66_ * SIGMA);
G3_67=trace(A67_ * SIGMA);
G3_71=trace(A71_ * SIGMA);
G3_72=trace(A72_ * SIGMA);
G3_73=trace(A73_ * SIGMA);
G3_74=trace(A74_ * SIGMA);
G3_75=trace(A75_ * SIGMA);
G3_76=trace(A76_ * SIGMA);
G3_77=trace(A77_ * SIGMA);
G3_81=trace(A81_ * SIGMA);
G3_82=trace(A82_ * SIGMA);
G3_83=trace(A83_ * SIGMA);
G3_84=trace(A84_ * SIGMA);
G3_85=trace(A85_ * SIGMA);
G3_86=trace(A86_ * SIGMA);
G3_87=trace(A87_ * SIGMA);
G3_91=trace(A91_ * SIGMA);
G3_92=trace(A92_ * SIGMA);
G3_93=trace(A93_ * SIGMA);
G3_94=trace(A94_ * SIGMA);
G3_95=trace(A95_ * SIGMA);
G3_96=trace(A96_ * SIGMA);
G3_97=trace(A97_ * SIGMA);
G3_101=trace(A101_ * SIGMA);
G3_102=trace(A102_ * SIGMA);
G3_103=trace(A103_ * SIGMA);
G3_104=trace(A104_ * SIGMA);
G3_105=trace(A105_ * SIGMA);
G3_106=trace(A106_ * SIGMA);
G3_107=trace(A107_ * SIGMA);
G3_aux1=G3_11//G3_12//G3_13//G3_14//G3_15//G3_
16//G3_17//G3_21//G3_22//G3_23//G3_24//G3_25//
G3_26//G3_27//G3_31//G3_32//G3_33//G3_34//G3_3
5//G3_36//G3_37//G3_41//G3_42//G3_43//G3_44//G
3_45//G3_46//G3_47//G3_51//G3_52//G3_53//G3_54
//G3_55//G3_56//G3_57//G3_61//G3_62//G3_63//G3
_64//G3_65//G3_66//G3_67//G3_71//G3_72//G3_73/
/G3_74//G3_75//G3_76//G3_77//G3_81//G3_82//G3_
83//G3_84//G3_85//G3_86//G3_87//G3_91//G3_92//
G3_93//G3_94//G3_95//G3_96//G3_97//G3_101//G3_
102//G3_103//G3_104//G3_105//G3_106//G3_107;
```

```
create G3_aux1 from G3_aux1; append from
G3_aux1; quit;

Proc iml;
use SIGMA; read all into SIGMA;
use A111_; read all into A111_;
use A112_; read all into A112_;
use A113_; read all into A113_;
use A114_; read all into A114_;
use A115_; read all into A115_;
use A116_; read all into A116_;
use A117_; read all into A117_;
use A121_; read all into A121_;
use A122_; read all into A122_;
use A123_; read all into A123_;
use A124_; read all into A124_;
use A125_; read all into A125_;
use A126_; read all into A126_;
use A127_; read all into A127_;
use A131_; read all into A131_;
use A132_; read all into A132_;
use A133_; read all into A133_;
use A134_; read all into A134_;
use A135_; read all into A135_;
use A136_; read all into A136_;
use A137_; read all into A137_;
use A141_; read all into A141_;
use A142_; read all into A142_;
use A143_; read all into A143_;
use A144_; read all into A144_;
use A145_; read all into A145_;
use A146_; read all into A146_;
use A147_; read all into A147_;
use A151_; read all into A151_;
use A152_; read all into A152_;
use A153_; read all into A153_;
use A154_; read all into A154_;
use A155_; read all into A155_;
use A156_; read all into A156_;
use A157_; read all into A157_;
use A161_; read all into A161_;
use A162_; read all into A162_;
use A163_; read all into A163_;
use A164_; read all into A164_;
use A165_; read all into A165_;
use A166_; read all into A166_;
use A167_; read all into A167_;
use A171_; read all into A171_;
use A172_; read all into A172_;
use A173_; read all into A173_;
use A174_; read all into A174_;
use A175_; read all into A175_;
use A176_; read all into A176_;
use A177_; read all into A177_;
use A181_; read all into A181_;
use A182_; read all into A182_;
use A183_; read all into A183_;
use A184_; read all into A184_;
use A185_; read all into A185_;
use A186_; read all into A186_;
use A187_; read all into A187_;
use A191_; read all into A191_;
use A192_; read all into A192_;
use A193_; read all into A193_;
use A194_; read all into A194_;
use A195_; read all into A195_;
use A196_; read all into A196_;
use A197_; read all into A197_;
use A201_; read all into A201_;
use A202_; read all into A202_;
use A203_; read all into A203_;
use A204_; read all into A204_;
use A205_; read all into A205_;
use A206_; read all into A206_;
use A207_; read all into A207_;
G3_111=trace(A111_ * SIGMA);
G3_112=trace(A112_ * SIGMA);
G3_113=trace(A113_ * SIGMA);
G3_114=trace(A114_ * SIGMA);
G3_115=trace(A115_ * SIGMA);
G3_116=trace(A116_ * SIGMA);
G3_117=trace(A117_ * SIGMA);
G3_121=trace(A121_ * SIGMA);
G3_122=trace(A122_ * SIGMA);
G3_123=trace(A123_ * SIGMA);
G3_124=trace(A124_ * SIGMA);
G3_125=trace(A125_ * SIGMA);
G3_126=trace(A126_ * SIGMA);
G3_127=trace(A127_ * SIGMA);
G3_131=trace(A131_ * SIGMA);
G3_132=trace(A132_ * SIGMA);
G3_133=trace(A133_ * SIGMA);
G3_134=trace(A134_ * SIGMA);
G3_135=trace(A135_ * SIGMA);
G3_136=trace(A136_ * SIGMA);
G3_137=trace(A137_ * SIGMA);
G3_141=trace(A141_ * SIGMA);
G3_142=trace(A142_ * SIGMA);
G3_143=trace(A143_ * SIGMA);
G3_144=trace(A144_ * SIGMA);
G3_145=trace(A145_ * SIGMA);
G3_146=trace(A146_ * SIGMA);
G3_147=trace(A147_ * SIGMA);
G3_151=trace(A151_ * SIGMA);
G3_152=trace(A152_ * SIGMA);
G3_153=trace(A153_ * SIGMA);
G3_154=trace(A154_ * SIGMA);
G3_155=trace(A155_ * SIGMA);
G3_156=trace(A156_ * SIGMA);
G3_157=trace(A157_ * SIGMA);
G3_161=trace(A161_ * SIGMA);
G3_162=trace(A162_ * SIGMA);
G3_163=trace(A163_ * SIGMA);
G3_164=trace(A164_ * SIGMA);
G3_165=trace(A165_ * SIGMA);
G3_166=trace(A166_ * SIGMA);
G3_167=trace(A167_ * SIGMA);
G3_171=trace(A171_ * SIGMA);
G3_172=trace(A172_ * SIGMA);
G3_173=trace(A173_ * SIGMA);
G3_174=trace(A174_ * SIGMA);
G3_175=trace(A175_ * SIGMA);
G3_176=trace(A176_ * SIGMA);
G3_177=trace(A177_ * SIGMA);
G3_181=trace(A181_ * SIGMA);
G3_182=trace(A182_ * SIGMA);
G3_183=trace(A183_ * SIGMA);
G3_184=trace(A184_ * SIGMA);
G3_185=trace(A185_ * SIGMA);
G3_186=trace(A186_ * SIGMA);
G3_187=trace(A187_ * SIGMA);
G3_191=trace(A191_ * SIGMA);
G3_192=trace(A192_ * SIGMA);
G3_193=trace(A193_ * SIGMA);
G3_194=trace(A194_ * SIGMA);
G3_195=trace(A195_ * SIGMA);
G3_196=trace(A196_ * SIGMA);
G3_197=trace(A197_ * SIGMA);
G3_201=trace(A201_ * SIGMA);
G3_202=trace(A202_ * SIGMA);
G3_203=trace(A203_ * SIGMA);
G3_204=trace(A204_ * SIGMA);
G3_205=trace(A205_ * SIGMA);
G3_206=trace(A206_ * SIGMA);
G3_207=trace(A207_ * SIGMA);
G3_aux2=G3_111//G3_112//G3_113//G3_114//G3_115
//G3_116//G3_117//G3_121//G3_122//G3_123//G3_1
24//G3_125//G3_126//G3_127//G3_131//G3_132//G3
_133//G3_134//G3_135//G3_136//G3_137//G3_141//
G3_142//G3_143//G3_144//G3_145//G3_146//G3_147
//G3_151//G3_152//G3_153//G3_154//G3_155//G3_1
56//G3_157//G3_161//G3_162//G3_163//G3_164//G3
_165//G3_166//G3_167//G3_171//G3_172//G3_173//
G3_174//G3_175//G3_176//G3_177//G3_181//G3_182
//G3_183//G3_184//G3_185//G3_186//G3_187//G3_1
91//G3_192//G3_193//G3_194//G3_195//G3_196//G3
_197//G3_201//G3_202//G3_203//G3_204//G3_205//
G3_206//G3_207;
create G3_aux2 from G3_aux2; append from
G3_aux2; quit;

Proc iml;
use SIGMA; read all into SIGMA;
use A211_; read all into A211_;
use A212_; read all into A212_;
use A213_; read all into A213_;
use A214_; read all into A214_;
use A215_; read all into A215_;
use A216_; read all into A216_;
use A217_; read all into A217_;
use A221_; read all into A221_;
use A222_; read all into A222_;
use A223_; read all into A223_;
use A224_; read all into A224_;
use A225_; read all into A225_;
use A226_; read all into A226_;
```

```
use A227_; read all into A227_;
use A231_; read all into A231_;
use A232_; read all into A232_;
use A233_; read all into A233_;
use A234_; read all into A234_;
use A235_; read all into A235_;
use A236_; read all into A236_;
use A237_; read all into A237_;
use A241_; read all into A241_;
use A242_; read all into A242_;
use A243_; read all into A243_;
use A244_; read all into A244_;
use A245_; read all into A245_;
use A246_; read all into A246_;
use A247_; read all into A247_;
use A251_; read all into A251_;
use A252_; read all into A252_;
use A253_; read all into A253_;
use A254_; read all into A254_;
use A255_; read all into A255_;
use A256_; read all into A256_;
use A257_; read all into A257_;
use A261_; read all into A261_;
use A262_; read all into A262_;
use A263_; read all into A263_;
use A264_; read all into A264_;
use A265_; read all into A265_;
use A266_; read all into A266_;
use A267_; read all into A267_;
use A271_; read all into A271_;
use A272_; read all into A272_;
use A273_; read all into A273_;
use A274_; read all into A274_;
use A275_; read all into A275_;
use A276_; read all into A276_;
use A277_; read all into A277_;
use A281_; read all into A281_;
use A282_; read all into A282_;
use A283_; read all into A283_;
use A284_; read all into A284_;
use A285_; read all into A285_;
use A286_; read all into A286_;
use A287_; read all into A287_;
G3_211=trace(A211_ * SIGMA);
G3_212=trace(A212_ * SIGMA);
G3_213=trace(A213_ * SIGMA);
G3_214=trace(A214_ * SIGMA);
G3_215=trace(A215_ * SIGMA);
G3_216=trace(A216_ * SIGMA);
G3_217=trace(A217_ * SIGMA);
G3_221=trace(A221_ * SIGMA);
G3_222=trace(A222_ * SIGMA);
G3_223=trace(A223_ * SIGMA);
G3_224=trace(A224_ * SIGMA);
G3_225=trace(A225_ * SIGMA);
G3_226=trace(A226_ * SIGMA);
G3_227=trace(A227_ * SIGMA);
G3_231=trace(A231_ * SIGMA);
G3_232=trace(A232_ * SIGMA);
G3_233=trace(A233_ * SIGMA);
G3_234=trace(A234_ * SIGMA);
G3_235=trace(A235_ * SIGMA);
G3_236=trace(A236_ * SIGMA);
G3_237=trace(A237_ * SIGMA);
G3_241=trace(A241_ * SIGMA);
G3_242=trace(A242_ * SIGMA);
G3_243=trace(A243_ * SIGMA);
G3_244=trace(A244_ * SIGMA);
G3_245=trace(A245_ * SIGMA);
G3_246=trace(A246_ * SIGMA);
G3_247=trace(A247_ * SIGMA);
G3_251=trace(A251_ * SIGMA);
G3_252=trace(A252_ * SIGMA);
G3_253=trace(A253_ * SIGMA);
G3_254=trace(A254_ * SIGMA);
G3_255=trace(A255_ * SIGMA);
G3_256=trace(A256_ * SIGMA);
G3_257=trace(A257_ * SIGMA);
G3_261=trace(A261_ * SIGMA);
G3_262=trace(A262_ * SIGMA);
G3_263=trace(A263_ * SIGMA);
G3_264=trace(A264_ * SIGMA);
G3_265=trace(A265_ * SIGMA);
G3_266=trace(A266_ * SIGMA);
G3_267=trace(A267_ * SIGMA);
G3_271=trace(A271_ * SIGMA);
G3_272=trace(A272_ * SIGMA);
G3_273=trace(A273_ * SIGMA);
```

```
G3_274=trace(A274_ * SIGMA);
G3_275=trace(A275_ * SIGMA);
G3_276=trace(A276_ * SIGMA);
G3_277=trace(A277_ * SIGMA);
G3_281=trace(A281_ * SIGMA);
G3_282=trace(A282_ * SIGMA);
G3_283=trace(A283_ * SIGMA);
G3_284=trace(A284_ * SIGMA);
G3_285=trace(A285_ * SIGMA);
G3_286=trace(A286_ * SIGMA);
G3_287=trace(A287_ * SIGMA);
G3_aux3=G3_211//G3_212//G3_213//G3_214//G3_215
//G3_216//G3_217//G3_221//G3_222//G3_223//G3_2
24//G3_225//G3_226//G3_227//G3_231//G3_232//G3
_233//G3_234//G3_235//G3_236//G3_237//G3_241//
G3_242//G3_243//G3_244//G3_245//G3_246//G3_247
//G3_251//G3_252//G3_253//G3_254//G3_255//G3_2
56//G3_257//G3_261//G3_262//G3_263//G3_264//G3
_265//G3_266//G3_267//G3_271//G3_272//G3_273//
G3_274//G3_275//G3_276//G3_277//G3_281//G3_282
//G3_283//G3_284//G3_285//G3_286//G3_287;creat
e G3_aux3 from G3_aux3; append from G3_aux3;
quit;

Data G3_aux;
set G3_aux1 G3_aux2 G3_aux3;
run;

Data g3;
set g3_aux;
rename col1=g3;
run;

/* *** Fim *** */

/* **** Estimativa do EQMP do EBLUP *****/

Data Est_PC;
merge crono_resf g1 g2 g3;
EQM_EBLUP= G1 + G2 + 2*G3;
simul=&i;
run;

Proc append base=PC.Est_PC_EQM_mod111_Anal
data=Est_PC;
run;

DM 'CLEAR OUTPUT';
DM 'CLEAR LOG';

%mend;
%inc 'C:\Documents and Settings\Luis
Pereira\My
Documents\Doutoramento\chamamacro.txt';

/*********************************************
*********************************************/
/*********************************************
*********************************************/
/*** Estimação do EQMP do EBLUP utilizando a
aproximação Bootstrap *****/
/** Código SAS utilizado depois de estimado o
modelo espaciotemporal para cada conjunto de
dados *******************/
/*********************************************
*********************************************/
/*********************************************
*********************************************/

/********************************************/
/* CÁLCULO DOS EFEITOS FIXOS - QUE SE ADMITE
SEREM OS VERDADEIROS */
/********************************************/

Data id;
do id=1 to 2;
output; end; run;

Data beta;
merge id beta;
run;

Data beta0;
set beta;
if id=1;
rename estimate=b0;
keep estimate; run;
```

```
Data beta1;
set beta;
if id=2;
rename estimate=b1;
keep estimate; run;

Data betan_base;
merge beta0 beta1;
_type_=0; run;

/****** Efeitos Aleatórios: v ******/

Data u;
set u;
if estimate='.' then estimate=0;
run;

Data v;
set u;
if Effect='i';
keep i estimate;
run;

Proc sort data=v;
by i; run;

Proc iml;
use v; read all into v;
one=J(7,1,1);
v1=v@one;
create v1 from v1; append from v1; quit;

Data v1;
set v1;
_type_=0;
rename col2=v;
run;

/****** Efeitos Aleatórios: u ******/

Data u1;
set u;
if Effect='i*t';
_type_=0;
keep i t estimate _type_;
rename estimate=u; run;

Proc sort data=u1;
by i t; run;

Data beta_vu;
merge betan_base v1 u1;
by _type_; run;

Data betan_base;
set betan_base;
drop _type_;
run;

/** Estimação do parâmetro de interesse em
cada NUTSIII **/

Proc sort data=cron;
by i t; run;

Data cron1;
merge cron beta_vu;
by i t; run;

Data crono_resf;
set cron1;
e = b0 + x*b1 + v + u;
keep i y t _type_ x b0 b1 u v e;
run;

Data est_PC;
set crono_resf;
simul=1;    /*SSS*/
drop _type_; run;

/*********************************************
*********************************************/
/* GERAÇÃO DE DADOS BOOTSTRAP COM BASE NOS
PARÂMETROS ESTIMADOS   */
/*********************************************
*********************************************/

Data null;
file 'C:\Documents and Settings\Luis
Pereira\My
Documents\Doutoramento\chamamacro.txt';
   do i= 1 to 250; k=i;
var='%ext_am('||i||','||k||');'; put var; end;
run;

%macro ext_am(i,k);

Data a1;
set varu;
do i=1 to 28;
        do t=1 to 7;
        e=rannor(&k+101);
        Eu=sqrt(sigma2u)*rannor(&k+102);
        output;
end;
end; drop sigma2u; run;

Data a2;
set varv;
        do i=1 to 28;

        v=sqrt(sigma2v)*rannor(&k+103);
                output;
end; drop sigma2v; run;

Data a4;
set a1;
keep Eu;
run;

Proc iml;
use a4; read all into a4;
use dataro; read all into dataro;
Ident7=I(7);
Ident28=I(28);
I1={0 0 0 0 0 0 0,
    1 0 0 0 0 0 0,
    0 1 0 0 0 0 0,
    0 0 1 0 0 0 0,
    0 0 0 1 0 0 0,
    0 0 0 0 1 0 0,
    0 0 0 0 0 1 0};
I2={0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    1 0 0 0 0 0 0,
    0 1 0 0 0 0 0,
    0 0 1 0 0 0 0,
    0 0 0 1 0 0 0,
    0 0 0 0 1 0 0};
I3={0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    1 0 0 0 0 0 0,
    0 1 0 0 0 0 0,
    0 0 1 0 0 0 0,
    0 0 0 1 0 0 0};
I4={0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    1 0 0 0 0 0 0,
    0 1 0 0 0 0 0,
    0 0 1 0 0 0 0};
I5={0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    1 0 0 0 0 0 0,
    0 1 0 0 0 0 0};
I6={0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    0 0 0 0 0 0 0,
    1 0 0 0 0 0 0};
Q=dataro*I1 + dataro*dataro*I2 +
dataro*dataro*dataro*I3 +
dataro*dataro*dataro*dataro*I4 +
dataro*dataro*dataro*dataro*dataro*I5 +
dataro*dataro*dataro*dataro*dataro*dataro*I6 +
Ident7;
a5=(Ident28@Q)*a4;
create a5 from a5; append from a5;
quit;
```

```
Data a5;
set a5;
rename col1=u; run;

/* Determinação dos efeitos aleatórios de
domínio : v */

Data a2;
set a2;
drop i;
run;

Proc iml;
use a2; read all into a2;
use dataphi; read all into dataphi;
use Pc.W2; read all into W2;
Ident=I(28);
v=inv(Ident - dataphi*W2)*a2;
um=J(7,1);
a3=v@um;
create a3 from a3; append from a3;
quit;

Data a3;
set a3;
rename col1=v;
run;

/****/

Proc iml;
use MatX; read all into MatX;
use Betan_base; read all into Betan_base;
a7=MatX*t(Betan_base);   /* Efeitos fixos do
modelo */
create a7 from a7; append from a7;
quit;

Data a7;
set a7;
rename col1=XB;
run;

Data a6;
merge a1 a3 a5 a7 MatX;
y=XB + v + u + e;
drop eu; run;

Data a8;
set a6;
boot=&i;
run;

/********************************************/
/*** Fim da geração dos dados no bootstrap **/
/********************************************/

Data iabh;
set a6;
keep i t c x;
run;

Data ipth;
set a6;
keep i t y;
run;

Data iabhipth1;
set a6;
keep i t y x;
run;

/** Preparação do ficheiro das observações da
variável de interesse **/

Proc sort data=ipth;
by i t; run;

Data tetad1; set ipth; where i= 1; keep y;
Data tetad2; set ipth; where i= 2; keep y;
Data tetad3; set ipth; where i= 3; keep y;
Data tetad4; set ipth; where i= 4; keep y;
Data tetad5; set ipth; where i= 5; keep y;
Data tetad6; set ipth; where i= 6; keep y;
Data tetad7; set ipth; where i= 7; keep y;
Data tetad8; set ipth; where i= 8; keep y;
Data tetad9; set ipth; where i= 9; keep y;
Data tetad10; set ipth; where i= 10; keep y;
Data tetad11; set ipth; where i= 11; keep y;
Data tetad12; set ipth; where i= 12; keep y;
Data tetad13; set ipth; where i= 13; keep y;
Data tetad14; set ipth; where i= 14; keep y;
Data tetad15; set ipth; where i= 15; keep y;
Data tetad16; set ipth; where i= 16; keep y;
Data tetad17; set ipth; where i= 17; keep y;
Data tetad18; set ipth; where i= 18; keep y;
Data tetad19; set ipth; where i= 19; keep y;
Data tetad20; set ipth; where i= 20; keep y;
Data tetad21; set ipth; where i= 21; keep y;
Data tetad22; set ipth; where i= 22; keep y;
Data tetad23; set ipth; where i= 23; keep y;
Data tetad24; set ipth; where i= 24; keep y;
Data tetad25; set ipth; where i= 25; keep y;
Data tetad26; set ipth; where i= 26; keep y;
Data tetad27; set ipth; where i= 27; keep y;
Data tetad28; set ipth; where i= 28; keep y;
run;
/** Preparação do ficheiro das observações da
variável auxiliar **/

Proc sort data=iabh;
by i t; run;

Proc sql; create table iabh0 as select c, x,
i, t from iabh;

Data xd1; set iabh0; where i= 1; keep c x;
Data xd2; set iabh0; where i= 2; keep c x;
Data xd3; set iabh0; where i= 3; keep c x;
Data xd4; set iabh0; where i= 4; keep c x;
Data xd5; set iabh0; where i= 5; keep c x;
Data xd6; set iabh0; where i= 6; keep c x;
Data xd7; set iabh0; where i= 7; keep c x;
Data xd8; set iabh0; where i= 8; keep c x;
Data xd9; set iabh0; where i= 9; keep c x;
Data xd10; set iabh0; where i= 10; keep c x;
Data xd11; set iabh0; where i= 11; keep c x;
Data xd12; set iabh0; where i= 12; keep c x;
Data xd13; set iabh0; where i= 13; keep c x;
Data xd14; set iabh0; where i= 14; keep c x;
Data xd15; set iabh0; where i= 15; keep c x;
Data xd16; set iabh0; where i= 16; keep c x;
Data xd17; set iabh0; where i= 17; keep c x;
Data xd18; set iabh0; where i= 18; keep c x;
Data xd19; set iabh0; where i= 19; keep c x;
Data xd20; set iabh0; where i= 20; keep c x;
Data xd21; set iabh0; where i= 21; keep c x;
Data xd22; set iabh0; where i= 22; keep c x;
Data xd23; set iabh0; where i= 23; keep c x;
Data xd24; set iabh0; where i= 24; keep c x;
Data xd25; set iabh0; where i= 25; keep c x;
Data xd26; set iabh0; where i= 26; keep c x;
Data xd27; set iabh0; where i= 27; keep c x;
Data xd28; set iabh0; where i= 28; keep c x;
run;

/** Cálculo da matriz Z(1) **/

Proc iml;
use tetad1; read all into tetad1;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd1=(Ident - D)*P*tetad1;
create zd1 from zd1; append from zd1; quit;
Proc iml;
use tetad2; read all into tetad2;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd2=(Ident - D)*P*tetad2;
create zd2 from zd2; append from zd2; quit;
Proc iml;
use tetad3; read all into tetad3;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd3=(Ident - D)*P*tetad3;
create zd3 from zd3; append from zd3; quit;
Proc iml;
use tetad4; read all into tetad4;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd4=(Ident - D)*P*tetad4;
```

```
create zd4 from zd4; append from zd4; quit;
Proc iml;
use tetad5; read all into tetad5;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd5=(Ident - D)*P*tetad5;
create zd5 from zd5; append from zd5; quit;
Proc iml;
use tetad6; read all into tetad6;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd6=(Ident - D)*P*tetad6;
create zd6 from zd6; append from zd6; quit;
Proc iml;
use tetad7; read all into tetad7;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd7=(Ident - D)*P*tetad7;
create zd7 from zd7; append from zd7; quit;
Proc iml;
use tetad8; read all into tetad8;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd8=(Ident - D)*P*tetad8;
create zd8 from zd8; append from zd8; quit;
Proc iml;
use tetad9; read all into tetad9;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd9=(Ident - D)*P*tetad9;
create zd9 from zd9; append from zd9; quit;
Proc iml;
use tetad10; read all into tetad10;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd10=(Ident - D)*P*tetad10;
create zd10 from zd10; append from zd10; quit;
Proc iml;
use tetad11; read all into tetad11;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd11=(Ident - D)*P*tetad11;
create zd11 from zd11; append from zd11; quit;
Proc iml;
use tetad12; read all into tetad12;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd12=(Ident - D)*P*tetad12;
create zd12 from zd12; append from zd12; quit;
Proc iml;
use tetad13; read all into tetad13;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd13=(Ident - D)*P*tetad13;
create zd13 from zd13; append from zd13; quit;
Proc iml;
use tetad14; read all into tetad14;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd14=(Ident - D)*P*tetad14;
create zd14 from zd14; append from zd14; quit;
Proc iml;
use tetad15; read all into tetad15;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd15=(Ident - D)*P*tetad15;
create zd15 from zd15; append from zd15; quit;
Proc iml;
use tetad16; read all into tetad16;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd16=(Ident - D)*P*tetad16;
create zd16 from zd16; append from zd16; quit;
Proc iml;
use tetad17; read all into tetad17;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd17=(Ident - D)*P*tetad17;
create zd17 from zd17; append from zd17; quit;
Proc iml;
use tetad18; read all into tetad18;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd18=(Ident - D)*P*tetad18;
create zd18 from zd18; append from zd18; quit;
Proc iml;
use tetad19; read all into tetad19;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd19=(Ident - D)*P*tetad19;
create zd19 from zd19; append from zd19; quit;
Proc iml;
use tetad20; read all into tetad20;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd20=(Ident - D)*P*tetad20;
create zd20 from zd20; append from zd20; quit;
Proc iml;
use tetad21; read all into tetad21;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd21=(Ident - D)*P*tetad21;
create zd21 from zd21; append from zd21; quit;
Proc iml;
use tetad22; read all into tetad22;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd22=(Ident - D)*P*tetad22;
create zd22 from zd22; append from zd22; quit;
Proc iml;
use tetad23; read all into tetad23;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd23=(Ident - D)*P*tetad23;
create zd23 from zd23; append from zd23; quit;
Proc iml;
use tetad24; read all into tetad24;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd24=(Ident - D)*P*tetad24;
create zd24 from zd24; append from zd24; quit;
Proc iml;
use tetad25; read all into tetad25;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd25=(Ident - D)*P*tetad25;
create zd25 from zd25; append from zd25; quit;
Proc iml;
use tetad26; read all into tetad26;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd26=(Ident - D)*P*tetad26;
create zd26 from zd26; append from zd26; quit;
Proc iml;
use tetad27; read all into tetad27;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd27=(Ident - D)*P*tetad27;
create zd27 from zd27; append from zd27; quit;
Proc iml;
use tetad28; read all into tetad28;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd28=(Ident - D)*P*tetad28;
create zd28 from zd28; append from zd28; quit;

Data Z1;
set zd1 zd2 zd3 zd4 zd5 zd6 zd7 zd8 zd9 zd10
zd11 zd12 zd13 zd14 zd15
zd16 zd17 zd18 zd19 zd20 zd21 zd22 zd23 zd24
zd25 zd26 zd27 zd28;
run;
```

```
/** Cálculo da matriz H(1) **/

Proc iml;
use xd1; read all into xd1;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd1=(Ident - D)*P*xd1;
create hd1 from hd1; append from hd1; quit;
Proc iml;
use xd2; read all into xd2;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd2=(Ident - D)*P*xd2;
create hd2 from hd2; append from hd2; quit;
Proc iml;
use xd3; read all into xd3;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd3=(Ident - D)*P*xd3;
create hd3 from hd3; append from hd3; quit;
Proc iml;
use xd4; read all into xd4;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd4=(Ident - D)*P*xd4;
create hd4 from hd4; append from hd4; quit;
Proc iml;
use xd5; read all into xd5;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd5=(Ident - D)*P*xd5;
create hd5 from hd5; append from hd5; quit;
Proc iml;
use xd6; read all into xd6;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd6=(Ident - D)*P*xd6;
create hd6 from hd6; append from hd6; quit;
Proc iml;
use xd7; read all into xd7;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd7=(Ident - D)*P*xd7;
create hd7 from hd7; append from hd7; quit;
Proc iml;
use xd8; read all into xd8;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd8=(Ident - D)*P*xd8;
create hd8 from hd8; append from hd8; quit;
Proc iml;
use xd9; read all into xd9;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd9=(Ident - D)*P*xd9;
create hd9 from hd9; append from hd9; quit;
Proc iml;
use xd10; read all into xd10;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd10=(Ident - D)*P*xd10;
create hd10 from hd10; append from hd10; quit;
Proc iml;
use xd11; read all into xd11;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd11=(Ident - D)*P*xd11;
create hd11 from hd11; append from hd11; quit;
Proc iml;
use xd12; read all into xd12;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd12=(Ident - D)*P*xd12;
create hd12 from hd12; append from hd12; quit;
Proc iml;
use xd13; read all into xd13;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd13=(Ident - D)*P*xd13;
create hd13 from hd13; append from hd13; quit;
Proc iml;
use xd14; read all into xd14;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd14=(Ident - D)*P*xd14;
create hd14 from hd14; append from hd14; quit;
Proc iml;
use xd15; read all into xd15;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd15=(Ident - D)*P*xd15;
create hd15 from hd15; append from hd15; quit;
Proc iml;
use xd16; read all into xd16;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd16=(Ident - D)*P*xd16;
create hd16 from hd16; append from hd16; quit;
Proc iml;
use xd17; read all into xd17;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd17=(Ident - D)*P*xd17;
create hd17 from hd17; append from hd17; quit;
Proc iml;
use xd18; read all into xd18;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd18=(Ident - D)*P*xd18;
create hd18 from hd18; append from hd18; quit;
Proc iml;
use xd19; read all into xd19;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd19=(Ident - D)*P*xd19;
create hd19 from hd19; append from hd19; quit;
Proc iml;
use xd20; read all into xd20;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd20=(Ident - D)*P*xd20;
create hd20 from hd20; append from hd20; quit;
Proc iml;
use xd21; read all into xd21;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd21=(Ident - D)*P*xd21;
create hd21 from hd21; append from hd21; quit;
Proc iml;
use xd22; read all into xd22;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd22=(Ident - D)*P*xd22;
create hd22 from hd22; append from hd22; quit;
Proc iml;
use xd23; read all into xd23;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd23=(Ident - D)*P*xd23;
create hd23 from hd23; append from hd23; quit;
Proc iml;
use xd24; read all into xd24;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd24=(Ident - D)*P*xd24;
create hd24 from hd24; append from hd24; quit;
Proc iml;
use xd25; read all into xd25;
use P; read all into P;
use D; read all into D;
Ident=I(7);
```

```
hd25=(Ident - D)*P*xd25;
create hd25 from hd25; append from hd25; quit;
Proc iml;
use xd26; read all into xd26;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd26=(Ident - D)*P*xd26;
create hd26 from hd26; append from hd26; quit;
Proc iml;
use xd27; read all into xd27;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd27=(Ident - D)*P*xd27;
create hd27 from hd27; append from hd27; quit;
Proc iml;
use xd28; read all into xd28;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd28=(Ident - D)*P*xd28;
create hd28 from hd28; append from hd28; quit;

Data H1;
set hd1 hd2 hd3 hd4 hd5 hd6 hd7 hd8 hd9 hd10
hd11 hd12 hd13 hd14 hd15
hd16 hd17 hd18 hd19 hd20 hd21 hd22 hd23 hd24
hd25 hd26 hd27 hd28;
run;

/** Regressão de Z(1) sobre H(1) **/

Data Z1;
set Z1;
rename col1=col0;
run;

Data reg;
merge Z1 H1;
run;

proc reg data=reg noprint;
        model col0=col1 col2 / noint;
        output out=rg1  residual=r1;
run; quit;

Data rg1;
set rg1;
qr1=r1*r1;
Keep qr1; run;

Proc means data=rg1 sum noprint;
var qr1;
output out=rg1_ sum=sqr1;
run;

Data rg1_;
set rg1_;
keep sqr1; run;

/** Cálculo da variância sigma2_u **/

Proc iml;
use H1; read all into H1;
Hs1=H1*ginv(t(H1)*H1)*t(H1); /* ginv(A) is the
Moore-Penrose generalized inverse matrix of A
*/
create Hs1 from Hs1; append from Hs1; quit; /*
Hs1 é a matriz H*(1) */

Proc iml;
use D; read all into D;
Ident=I(7);
ID=Ident - D; /* Matriz I - D */
create ID from ID; append from ID; quit;

Proc iml;
use ID; read all into ID;
blockID1=BLOCK(ID,ID,ID,ID,ID,ID,ID,ID,ID,I
D,ID,ID,ID);
create blockID1 from blockID1; append from
blockID1; quit;
Proc iml;
use blockID1; read all into blockID1;
blockID=BLOCK(blockID1,blockID1);
create blockID from blockID; append from
blockID; quit;
```

```
Data s1;
set a6;
keep c t i;
run;

Data s1;
set s1;
rename c=var; run;

Data Sigma_1;
set s1;
if i=1;
keep var;
run;
Proc iml;
use Sigma_1; read all into Sigma_1;
Sigma1=diag(Sigma_1);
create Sigma1 from Sigma1; append from Sigma1;
quit;
Proc iml;
use P; read all into P;
use Sigma1; read all into Sigma1;
P1=P*Sigma1*t(P);
create P1 from P1; append from P1; quit;

Data Sigma_2;
set s1;
if i=2;
keep var;
run;
Proc iml;
use Sigma_2; read all into Sigma_2;
Sigma2=diag(Sigma_2);
create Sigma2 from Sigma2; append from Sigma2;
quit;
Proc iml;
use P; read all into P;
use Sigma2; read all into Sigma2;
P2=P*Sigma2*t(P);
create P2 from P2; append from P2; quit;

Data Sigma_3;
set s1;
if i=3;
keep var;
run;
Proc iml;
use Sigma_3; read all into Sigma_3;
Sigma3=diag(Sigma_3);
create Sigma3 from Sigma3; append from Sigma3;
quit;
Proc iml;
use P; read all into P;
use Sigma3; read all into Sigma3;
P3=P*Sigma3*t(P);
create P3 from P3; append from P3; quit;

Data Sigma_4;
set s1;
if i=4;
keep var;
run;
Proc iml;
use Sigma_4; read all into Sigma_4;
Sigma4=diag(Sigma_4);
create Sigma4 from Sigma4; append from Sigma4;
quit;
Proc iml;
use P; read all into P;
use Sigma4; read all into Sigma4;
P4=P*Sigma4*t(P);
create P4 from P4; append from P4; quit;

Data Sigma_5;
set s1;
if i=5;
keep var;
run;
Proc iml;
use Sigma_5; read all into Sigma_5;
Sigma5=diag(Sigma_5);
create Sigma5 from Sigma5; append from Sigma5;
quit;
Proc iml;
use P; read all into P;
use Sigma5; read all into Sigma5;
P5=P*Sigma5*t(P);
```

```
create P5 from P5; append from P5; quit;

Data Sigma_6;
set s1;
if i=6;
keep var;
run;
Proc iml;
use Sigma_6; read all into Sigma_6;
Sigma6=diag(Sigma_6);
create Sigma6 from Sigma6; append from Sigma6;
quit;
Proc iml;
use P; read all into P;
use Sigma6; read all into Sigma6;
P6=P*Sigma6*t(P);
create P6 from P6; append from P6; quit;

Data Sigma_7;
set s1;
if i=7;
keep var;
run;
Proc iml;
use Sigma_7; read all into Sigma_7;
Sigma7=diag(Sigma_7);
create Sigma7 from Sigma7; append from Sigma7;
quit;
Proc iml;
use P; read all into P;
use Sigma7; read all into Sigma7;
P7=P*Sigma7*t(P);
create P7 from P7; append from P7; quit;

Data Sigma_8;
set s1;
if i=8;
keep var;
run;
Proc iml;
use Sigma_8; read all into Sigma_8;
Sigma8=diag(Sigma_8);
create Sigma8 from Sigma8; append from Sigma8;
quit;
Proc iml;
use P; read all into P;
use Sigma8; read all into Sigma8;
P8=P*Sigma8*t(P);
create P8 from P8; append from P8; quit;

Data Sigma_9;
set s1;
if i=9;
keep var;
run;
Proc iml;
use Sigma_9; read all into Sigma_9;
Sigma9=diag(Sigma_9);
create Sigma9 from Sigma9; append from Sigma9;
quit;
Proc iml;
use P; read all into P;
use Sigma9; read all into Sigma9;
P9=P*Sigma9*t(P);
create P9 from P9; append from P9; quit;

Data Sigma_10;
set s1;
if i=10;
keep var;
run;
Proc iml;
use Sigma_10; read all into Sigma_10;
Sigma10=diag(Sigma_10);
create Sigma10 from Sigma10; append from
Sigma10; quit;
Proc iml;
use P; read all into P;
use Sigma10; read all into Sigma10;
P10=P*Sigma10*t(P);
create P10 from P10; append from P10; quit;

Data Sigma_11;
set s1;
if i=11;
keep var;
run;
Proc iml;
use Sigma_11; read all into Sigma_11;
Sigma11=diag(Sigma_11);
create Sigma11 from Sigma11; append from
Sigma11; quit;
Proc iml;
use P; read all into P;
use Sigma11; read all into Sigma11;
P11=P*Sigma11*t(P);
create P11 from P11; append from P11; quit;

Data Sigma_12;
set s1;
if i=12;
keep var;
run;
Proc iml;
use Sigma_12; read all into Sigma_12;
Sigma12=diag(Sigma_12);
create Sigma12 from Sigma12; append from
Sigma12; quit;
Proc iml;
use P; read all into P;
use Sigma12; read all into Sigma12;
P12=P*Sigma12*t(P);
create P12 from P12; append from P12; quit;

Data Sigma_13;
set s1;
if i=13;
keep var;
run;
Proc iml;
use Sigma_13; read all into Sigma_13;
Sigma13=diag(Sigma_13);
create Sigma13 from Sigma13; append from
Sigma13; quit;
Proc iml;
use P; read all into P;
use Sigma13; read all into Sigma13;
P13=P*Sigma13*t(P);
create P13 from P13; append from P13; quit;

Data Sigma_14;
set s1;
if i=14;
keep var;
run;
Proc iml;
use Sigma_14; read all into Sigma_14;
Sigma14=diag(Sigma_14);
create Sigma14 from Sigma14; append from
Sigma14; quit;
Proc iml;
use P; read all into P;
use Sigma14; read all into Sigma14;
P14=P*Sigma14*t(P);
create P14 from P14; append from P14; quit;

Data Sigma_15;
set s1;
if i=15;
keep var;
run;
Proc iml;
use Sigma_15; read all into Sigma_15;
Sigma15=diag(Sigma_15);
create Sigma15 from Sigma15; append from
Sigma15; quit;
Proc iml;
use P; read all into P;
use Sigma15; read all into Sigma15;
P15=P*Sigma15*t(P);
create P15 from P15; append from P15; quit;

Data Sigma_16;
set s1;
if i=16;
keep var;
run;
Proc iml;
use Sigma_16; read all into Sigma_16;
Sigma16=diag(Sigma_16);
create Sigma16 from Sigma16; append from
Sigma16; quit;
Proc iml;
use P; read all into P;
use Sigma16; read all into Sigma16;
P16=P*Sigma16*t(P);
```

```
create P16 from P16; append from P16; quit;

Data Sigma_17;
set s1;
if i=17;
keep var;
run;
Proc iml;
use Sigma_17; read all into Sigma_17;
Sigma17=diag(Sigma_17);
create Sigma17 from Sigma17; append from
Sigma17; quit;
Proc iml;
use P; read all into P;
use Sigma17; read all into Sigma17;
P17=P*Sigma17*t(P);
create P17 from P17; append from P17; quit;

Data Sigma_18;
set s1;
if i=18;
keep var;
run;
Proc iml;
use Sigma_18; read all into Sigma_18;
Sigma18=diag(Sigma_18);
create Sigma18 from Sigma18; append from
Sigma18; quit;
Proc iml;
use P; read all into P;
use Sigma18; read all into Sigma18;
P18=P*Sigma18*t(P);
create P18 from P18; append from P18; quit;

Data Sigma_19;
set s1;
if i=19;
keep var;
run;
Proc iml;
use Sigma_19; read all into Sigma_19;
Sigma19=diag(Sigma_19);
create Sigma19 from Sigma19; append from
Sigma19; quit;
Proc iml;
use P; read all into P;
use Sigma19; read all into Sigma19;
P19=P*Sigma19*t(P);
create P19 from P19; append from P19; quit;

Data Sigma_20;
set s1;
if i=20;
keep var;
run;
Proc iml;
use Sigma_20; read all into Sigma_20;
Sigma20=diag(Sigma_20);
create Sigma20 from Sigma20; append from
Sigma20; quit;
Proc iml;
use P; read all into P;
use Sigma20; read all into Sigma20;
P20=P*Sigma20*t(P);
create P20 from P20; append from P20; quit;

Data Sigma_21;
set s1;
if i=21;
keep var;
run;
Proc iml;
use Sigma_21; read all into Sigma_21;
Sigma21=diag(Sigma_21);
create Sigma21 from Sigma21; append from
Sigma21; quit;
Proc iml;
use P; read all into P;
use Sigma21; read all into Sigma21;
P21=P*Sigma21*t(P);
create P21 from P21; append from P21; quit;

Data Sigma_22;
set s1;
if i=22;
keep var;
run;
Proc iml;
use Sigma_22; read all into Sigma_22;
Sigma22=diag(Sigma_22);
create Sigma22 from Sigma22; append from
Sigma22; quit;
Proc iml;
use P; read all into P;
use Sigma22; read all into Sigma22;
P22=P*Sigma22*t(P);
create P22 from P22; append from P22; quit;

Data Sigma_23;
set s1;
if i=23;
keep var;
run;
Proc iml;
use Sigma_23; read all into Sigma_23;
Sigma23=diag(Sigma_23);
create Sigma23 from Sigma23; append from
Sigma23; quit;
Proc iml;
use P; read all into P;
use Sigma23; read all into Sigma23;
P23=P*Sigma23*t(P);
create P23 from P23; append from P23; quit;

Data Sigma_24;
set s1;
if i=24;
keep var;
run;
Proc iml;
use Sigma_24; read all into Sigma_24;
Sigma24=diag(Sigma_24);
create Sigma24 from Sigma24; append from
Sigma24; quit;
Proc iml;
use P; read all into P;
use Sigma24; read all into Sigma24;
P24=P*Sigma24*t(P);
create P24 from P24; append from P24; quit;

Data Sigma_25;
set s1;
if i=25;
keep var;
run;
Proc iml;
use Sigma_25; read all into Sigma_25;
Sigma25=diag(Sigma_25);
create Sigma25 from Sigma25; append from
Sigma25; quit;
Proc iml;
use P; read all into P;
use Sigma25; read all into Sigma25;
P25=P*Sigma25*t(P);
create P25 from P25; append from P25; quit;

Data Sigma_26;
set s1;
if i=26;
keep var;
run;
Proc iml;
use Sigma_26; read all into Sigma_26;
Sigma26=diag(Sigma_26);
create Sigma26 from Sigma26; append from
Sigma26; quit;
Proc iml;
use P; read all into P;
use Sigma26; read all into Sigma26;
P26=P*Sigma26*t(P);
create P26 from P26; append from P26; quit;

Data Sigma_27;
set s1;
if i=27;
keep var;
run;
Proc iml;
use Sigma_27; read all into Sigma_27;
Sigma27=diag(Sigma_27);
create Sigma27 from Sigma27; append from
Sigma27; quit;
Proc iml;
use P; read all into P;
use Sigma27; read all into Sigma27;
P27=P*Sigma27*t(P);
```

```
create P27 from P27; append from P27; quit;

Data Sigma_28;
set s1;
if i=28;
keep var;
run;
Proc iml;
use Sigma_28; read all into Sigma_28;
Sigma28=diag(Sigma_28);
create Sigma28 from Sigma28; append from
Sigma28; quit;
Proc iml;
use P; read all into P;
use Sigma28; read all into Sigma28;
P28=P*Sigma28*t(P);
create P28 from P28; append from P28; quit;

Proc iml;
use P1; read all into P1;
use P2; read all into P2;
use P3; read all into P3;
use P4; read all into P4;
use P5; read all into P5;
use P6; read all into P6;
use P7; read all into P7;
use P8; read all into P8;
use P9; read all into P9;
use P10; read all into P10;
use P11; read all into P11;
use P12; read all into P12;
use P13; read all into P13;
use P14; read all into P14;
blockPP1=BLOCK(P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,
P11,P12,P13,P14);
create blockPP1 from blockPP1; append from
blockPP1; quit;

Proc iml;
use P15; read all into P15;
use P16; read all into P16;
use P17; read all into P17;
use P18; read all into P18;
use P19; read all into P19;
use P20; read all into P20;
use P21; read all into P21;
use P22; read all into P22;
use P23; read all into P23;
use P24; read all into P24;
use P25; read all into P25;
use P26; read all into P26;
use P27; read all into P27;
use P28; read all into P28;
blockPP2=BLOCK(P15,P16,P17,P18,P19,P20,P21,P22
,P23,P24,P25,P26,P27,P28);
create blockPP2 from blockPP2; append from
blockPP2; quit;

Proc iml;
use blockPP1; read all into blockPP1;
use blockPP2; read all into blockPP2;
blockPP=BLOCK(blockPP1,blockPP2);
create blockPP from blockPP; append from
blockPP; quit;

/* *** Fim *** */

Proc iml;
use H1; read all into H1;
HH=H1*t(H1);
eval=eigval(HH);
create eval from eval; append from eval; quit;

Data id;
do id=1 to 196;
output; end; run;

Data eval;
merge id eval; run;

Data new;
set eval;
by id;
retain count 0;
if first.id then count=0;
if col1 > 0.000001 then count = count + 1;
run;

Proc means data=new sum noprint;
var count;
output out=caracH1 sum=caracH1;
run;

Data caracH1;
set caracH1;
keep caracH1;
run;

Proc iml;
use blockPP; read all into blockPP;
use blockID; read all into blockID;
use Hs1; read all into Hs1;
use rg1_; read all into rg1_;
use caracH1; read all into caracH1;
varu_boot= (rg1_ -  trace((blockID -
Hs1)*blockPP))*(1/(28*6 - caracH1));
create varu_boot from varu_boot; append from
varu_boot; quit;

/* Termina aqui o cálculo da variância
sigma2_u */

/** Cálculo da matriz: z2= c^(-0.5)*f'*P*Teta
- var. dependente do modelo transformado **/

Proc iml;
use tetad1; read all into tetad1;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y1=(1/sqrt(c))*t(f)*P*tetad1;
create y1 from y1; append from y1; quit;
Proc iml;
use tetad2; read all into tetad2;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y2=(1/sqrt(c))*t(f)*P*tetad2;
create y2 from y2; append from y2; quit;
Proc iml;
use tetad3; read all into tetad3;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y3=(1/sqrt(c))*t(f)*P*tetad3;
create y3 from y3; append from y3; quit;
Proc iml;
use tetad4; read all into tetad4;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y4=(1/sqrt(c))*t(f)*P*tetad4;
create y4 from y4; append from y4; quit;
Proc iml;
use tetad5; read all into tetad5;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y5=(1/sqrt(c))*t(f)*P*tetad5;
create y5 from y5; append from y5; quit;
Proc iml;
use tetad6; read all into tetad6;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y6=(1/sqrt(c))*t(f)*P*tetad6;
create y6 from y6; append from y6; quit;
Proc iml;
use tetad7; read all into tetad7;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y7=(1/sqrt(c))*t(f)*P*tetad7;
create y7 from y7; append from y7; quit;
Proc iml;
use tetad8; read all into tetad8;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y8=(1/sqrt(c))*t(f)*P*tetad8;
create y8 from y8; append from y8; quit;
Proc iml;
use tetad9; read all into tetad9;
use P; read all into P;
use f; read all into f;
use c; read all into c;
```

```
y9=(1/sqrt(c))*t(f)*P*tetad9;
create y9 from y9; append from y9; quit;
Proc iml;
use tetad10; read all into tetad10;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y10=(1/sqrt(c))*t(f)*P*tetad10;
create y10 from y10; append from y10; quit;
Proc iml;
use tetad11; read all into tetad11;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y11=(1/sqrt(c))*t(f)*P*tetad11;
create y11 from y11; append from y11; quit;
Proc iml;
use tetad12; read all into tetad12;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y12=(1/sqrt(c))*t(f)*P*tetad12;
create y12 from y12; append from y12; quit;
Proc iml;
use tetad13; read all into tetad13;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y13=(1/sqrt(c))*t(f)*P*tetad13;
create y13 from y13; append from y13; quit;
Proc iml;
use tetad14; read all into tetad14;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y14=(1/sqrt(c))*t(f)*P*tetad14;
create y14 from y14; append from y14; quit;
Proc iml;
use tetad15; read all into tetad15;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y15=(1/sqrt(c))*t(f)*P*tetad15;
create y15 from y15; append from y15; quit;
Proc iml;
use tetad16; read all into tetad16;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y16=(1/sqrt(c))*t(f)*P*tetad16;
create y16 from y16; append from y16; quit;
Proc iml;
use tetad17; read all into tetad17;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y17=(1/sqrt(c))*t(f)*P*tetad17;
create y17 from y17; append from y17; quit;
Proc iml;
use tetad18; read all into tetad18;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y18=(1/sqrt(c))*t(f)*P*tetad18;
create y18 from y18; append from y18; quit;
Proc iml;
use tetad19; read all into tetad19;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y19=(1/sqrt(c))*t(f)*P*tetad19;
create y19 from y19; append from y19; quit;
Proc iml;
use tetad20; read all into tetad20;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y20=(1/sqrt(c))*t(f)*P*tetad20;
create y20 from y20; append from y20; quit;
Proc iml;
use tetad21; read all into tetad21;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y21=(1/sqrt(c))*t(f)*P*tetad21;
create y21 from y21; append from y21; quit;
Proc iml;
use tetad22; read all into tetad22;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y22=(1/sqrt(c))*t(f)*P*tetad22;
create y22 from y22; append from y22; quit;
Proc iml;
use tetad23; read all into tetad23;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y23=(1/sqrt(c))*t(f)*P*tetad23;
create y23 from y23; append from y23; quit;
Proc iml;
use tetad24; read all into tetad24;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y24=(1/sqrt(c))*t(f)*P*tetad24;
create y24 from y24; append from y24; quit;
Proc iml;
use tetad25; read all into tetad25;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y25=(1/sqrt(c))*t(f)*P*tetad25;
create y25 from y25; append from y25; quit;
Proc iml;
use tetad26; read all into tetad26;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y26=(1/sqrt(c))*t(f)*P*tetad26;
create y26 from y26; append from y26; quit;
Proc iml;
use tetad27; read all into tetad27;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y27=(1/sqrt(c))*t(f)*P*tetad27;
create y27 from y27; append from y27; quit;
Proc iml;
use tetad28; read all into tetad28;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y28=(1/sqrt(c))*t(f)*P*tetad28;
create y28 from y28; append from y28; quit;
run;

Data z2;
set y1 y2 y3 y4 y5 y6 y7 y8 y9 y10 y11 y12 y13
y14 y15 y16 y17 y18 y19 y20 y21 y22 y23 y24
y25 y26 y27 y28;
run;

/** Cálculo da matriz: H2= c^(-0.5)*f'*P*X -
var. independentes do modelo transformado **/

Proc iml;
use xd1; read all into xd1;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x1=(1/sqrt(c))*t(f)*P*xd1;
create x1 from x1; append from x1; quit;
Proc iml;
use xd2; read all into xd2;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x2=(1/sqrt(c))*t(f)*P*xd2;
create x2 from x2; append from x2; quit;
Proc iml;
use xd3; read all into xd3;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x3=(1/sqrt(c))*t(f)*P*xd3;
create x3 from x3; append from x3; quit;
Proc iml;
use xd4; read all into xd4;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x4=(1/sqrt(c))*t(f)*P*xd4;
create x4 from x4; append from x4; quit;
Proc iml;
use xd5; read all into xd5;
```

```
use P; read all into P;
use f; read all into f;
use c; read all into c;
x5=(1/sqrt(c))*t(f)*P*xd5;
create x5 from x5; append from x5; quit;
Proc iml;
use xd6; read all into xd6;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x6=(1/sqrt(c))*t(f)*P*xd6;
create x6 from x6; append from x6; quit;
Proc iml;
use xd7; read all into xd7;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x7=(1/sqrt(c))*t(f)*P*xd7;
create x7 from x7; append from x7; quit;
Proc iml;
use xd8; read all into xd8;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x8=(1/sqrt(c))*t(f)*P*xd8;
create x8 from x8; append from x8; quit;
Proc iml;
use xd9; read all into xd9;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x9=(1/sqrt(c))*t(f)*P*xd9;
create x9 from x9; append from x9; quit;
Proc iml;
use xd10; read all into xd10;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x10=(1/sqrt(c))*t(f)*P*xd10;
create x10 from x10; append from x10; quit;
Proc iml;
use xd11; read all into xd11;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x11=(1/sqrt(c))*t(f)*P*xd11;
create x11 from x11; append from x11; quit;
Proc iml;
use xd12; read all into xd12;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x12=(1/sqrt(c))*t(f)*P*xd12;
create x12 from x12; append from x12; quit;
Proc iml;
use xd13; read all into xd13;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x13=(1/sqrt(c))*t(f)*P*xd13;
create x13 from x13; append from x13; quit;
Proc iml;
use xd14; read all into xd14;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x14=(1/sqrt(c))*t(f)*P*xd14;
create x14 from x14; append from x14; quit;
Proc iml;
use xd15; read all into xd15;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x15=(1/sqrt(c))*t(f)*P*xd15;
create x15 from x15; append from x15; quit;
Proc iml;
use xd16; read all into xd16;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x16=(1/sqrt(c))*t(f)*P*xd16;
create x16 from x16; append from x16; quit;
Proc iml;
use xd17; read all into xd17;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x17=(1/sqrt(c))*t(f)*P*xd17;
create x17 from x17; append from x17; quit;
Proc iml;
use xd18; read all into xd18;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x18=(1/sqrt(c))*t(f)*P*xd18;
create x18 from x18; append from x18; quit;
Proc iml;
use xd19; read all into xd19;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x19=(1/sqrt(c))*t(f)*P*xd19;
create x19 from x19; append from x19; quit;
Proc iml;
use xd20; read all into xd20;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x20=(1/sqrt(c))*t(f)*P*xd20;
create x20 from x20; append from x20; quit;
Proc iml;
use xd21; read all into xd21;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x21=(1/sqrt(c))*t(f)*P*xd21;
create x21 from x21; append from x21; quit;
Proc iml;
use xd22; read all into xd22;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x22=(1/sqrt(c))*t(f)*P*xd22;
create x22 from x22; append from x22; quit;
Proc iml;
use xd23; read all into xd23;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x23=(1/sqrt(c))*t(f)*P*xd23;
create x23 from x23; append from x23; quit;
Proc iml;
use xd24; read all into xd24;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x24=(1/sqrt(c))*t(f)*P*xd24;
create x24 from x24; append from x24; quit;
Proc iml;
use xd25; read all into xd25;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x25=(1/sqrt(c))*t(f)*P*xd25;
create x25 from x25; append from x25; quit;
Proc iml;
use xd26; read all into xd26;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x26=(1/sqrt(c))*t(f)*P*xd26;
create x26 from x26; append from x26; quit;
Proc iml;
use xd27; read all into xd27;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x27=(1/sqrt(c))*t(f)*P*xd27;
create x27 from x27; append from x27; quit;
Proc iml;
use xd28; read all into xd28;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x28=(1/sqrt(c))*t(f)*P*xd28;
create x28 from x28; append from x28; quit;
run;

Data H2;
set x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13
x14 x15 x16 x17 x18 x19 x20 x21 x22 x23 x24
x25 x26 x27 x28;
run;

/** Regressão de z2(Y) sobre H2(X) **/
```

```
Data z2;
set z2;
rename col1=col0;
run;

Data reg2;
merge z2 H2;
run;

proc reg data=reg2 noprint;
        model col0=col1 col2 / noint;
        output out=rg2  residual=r2;
run; quit;

Data rg2;
set rg2;
qr2=r2*r2;
Keep qr2;
run;

Proc means data=rg2 sum noprint;
var qr2;
output out=rg2_ sum=sqr2;
run;

Data rg2_;
set rg2_;
keep sqr2; run;

/** Cálculo da variância sigma2_v **/

Proc iml;
use H2; read all into H2;
Ident=I(28);
PH2=Ident - H2*ginv(t(H2)*H2)*t(H2);    /*
Matriz P_H2 */
create PH2 from PH2; append from PH2; quit;

Proc iml;
use f; read all into f;
use P1; read all into P1;
use P2; read all into P2;
use P3; read all into P3;
use P4; read all into P4;
use P5; read all into P5;
use P6; read all into P6;
use P7; read all into P7;
use P8; read all into P8;
use P9; read all into P9;
use P10; read all into P10;
use P11; read all into P11;
use P12; read all into P12;
use P13; read all into P13;
use P14; read all into P14;
use P15; read all into P15;
use P16; read all into P16;
use P17; read all into P17;
use P18; read all into P18;
use P19; read all into P19;
use P20; read all into P20;
use P21; read all into P21;
use P22; read all into P22;
use P23; read all into P23;
use P24; read all into P24;
use P25; read all into P25;
use P26; read all into P26;
use P27; read all into P27;
use P28; read all into P28;
use c; read all into c;
cfp1=t(f)*P1*f*(1 / c);
cfp2=t(f)*P2*f*(1 / c);
cfp3=t(f)*P3*f*(1 / c);
cfp4=t(f)*P4*f*(1 / c);
cfp5=t(f)*P5*f*(1 / c);
cfp6=t(f)*P6*f*(1 / c);
cfp7=t(f)*P7*f*(1 / c);
cfp8=t(f)*P8*f*(1 / c);
cfp9=t(f)*P9*f*(1 / c);
cfp10=t(f)*P10*f*(1 / c);
cfp11=t(f)*P11*f*(1 / c);
cfp12=t(f)*P12*f*(1 / c);
cfp13=t(f)*P13*f*(1 / c);
cfp14=t(f)*P14*f*(1 / c);
cfp15=t(f)*P15*f*(1 / c);
cfp16=t(f)*P16*f*(1 / c);
cfp17=t(f)*P17*f*(1 / c);
cfp18=t(f)*P18*f*(1 / c);
cfp19=t(f)*P19*f*(1 / c);
cfp20=t(f)*P20*f*(1 / c);
cfp21=t(f)*P21*f*(1 / c);
cfp22=t(f)*P22*f*(1 / c);
cfp23=t(f)*P23*f*(1 / c);
cfp24=t(f)*P24*f*(1 / c);
cfp25=t(f)*P25*f*(1 / c);
cfp26=t(f)*P26*f*(1 / c);
cfp27=t(f)*P27*f*(1 / c);
cfp28=t(f)*P28*f*(1 / c);
cfp_=cfp1//cfp2//cfp3//cfp4//cfp5//cfp6//cfp7/
/cfp8//cfp9//cfp10//cfp11//cfp12//cfp13//cfp14
//cfp15//cfp16//cfp17//cfp18//cfp19//cfp20//cf
p21//cfp22//cfp23//cfp24//cfp25//cfp26//cfp27/
/cfp28;
cfp=diag(cfp_);
create cfp from cfp; append from cfp; quit;

/* *** Fim *** */

Proc iml;
use H2; read all into H2;
FF=H2*t(H2);
eval1=eigval(FF);
create eval1 from eval1; append from eval1;
quit;

Data id1;
do id=1 to 28;
output; end; run;

Data eval1;
merge id1 eval1;
run;

Data new1;
set eval1;
by id;
retain count 0;
if first.id then count=0;
if col1 > 0.000001 then count = count + 1;
run;

Proc means data=new1 sum noprint;
var count;
output out=caracH2 sum=caracH2;
run;

Data caracH2;
set caracH2;
keep caracH2;  /* Caracteristica de H2 */
run;

Proc iml;
use PH2; read all into PH2;
use cfp; read all into cfp;
use c; read all into c;
use rg2_; read all into rg2_;
use caracH2; read all into caracH2;
use varu_boot; read all into varu_boot;
use Binv; read all into Binv;
varv_boot= (rg2_ -  trace(PH2*cfp) -
varu_boot*(28 - caracH2))/(c*trace(PH2*Binv));
create varv_boot from varv_boot; append from
varv_boot; quit;

/*Termina aqui o cálculo da variância
sigma2_v*/
/* Truncagem a zero das variâncias sigma2_u e
sigma2_v */

Data varu_boot;
set varu_boot;
sigma2u= max( col1 , 0);
keep sigma2u;
run;
Data varv_boot;
set varv_boot;
sigma2v= max( col1 , 0);
keep sigma2v;
run;

/*** Matriz G1_boot ***/

Proc iml;
use varv_boot; read all into varv_boot;
use Binv; read all into Binv;
G1_boot=varv_boot*Binv;
```

```
create G1_boot from G1_boot; append from
G1_boot; quit;

/*** Matriz G2_boot ***/

Proc iml;
use varu_boot; read all into varu_boot;
use Gama; read all into Gama;
Ident=I(28);
G2_boot=varu_boot*(Ident@Gama);
create G2_boot from G2_boot; append from
G2_boot; quit;

/*** Matriz G_boot ***/

Proc iml;
use G1_boot; read all into G1_boot;
use G2_boot; read all into G2_boot;
Mat_GG_boot=block (G1_boot , G2_boot);
create Mat_GG_boot from Mat_GG_boot; append
from Mat_GG_boot; quit;

Proc iml;
row=t(do(1,224,1));
create row from row; append from row; quit;

Data row;
set row;
rename col1=row; run;

Data Mat_GG_boot;
merge row Mat_GG_boot; run;

/* Construção do ficheiro com os parâmetros de
variância dos erros da sondagem */

Proc sort data=s1;
by i t; run;

Data s2;
set s1;
keep var; run;

Data par;
set s2;
rename var=est;
run;

/********************************************/
/*** Fim da estimação das componentes da
variância com um dos conjuntos de dados
Bootstrap ***/
/********************************************/

/*********************************************
********************************************/
/*** ESTIMAÇÃO DO MODELO DE LUÍS PEREIRA ****/
/** utilizando as estimativas das componentes
de variância Boot - Mat_GG_boot - ***********/
/*********************************************
********************************************/

/* NOTA: Tem que ser usado sempre o conjunto
de dados inicial de Y e X, guardado no
ficheiro "cron" */

Proc mixed data=cron MMEqSol noprofile
noclprint;
        class i t;
        model y=x / solution ddfm=residual;
        random i;
   random i*t / type=un Gdata=Mat_GG_boot G V
S;
        repeated /group=i*t;
        parms /parmsdata=par noiter;
make 'solutionf' out=beta;
make 'solutionr' out=u;
make 'G' out=Mat_Gaux;
make 'V' out=Mat_V;
run; quit;

/********************************************/
/******** CÁLCULO DOS EFEITOS FIXOS  ********/
/** utilizando as estimativas das componentes
de variância Boot - Mat_GG_boot -
***************/
/********************************************/

Data id;
do id=1 to 2;
output; end; run;

Data beta;
merge id beta; run;

Data beta0;
set beta;
if id=1;
rename estimate=b0;
keep estimate; run;

Data beta1;
set beta;
if id=2;
rename estimate=b1;
keep estimate; run;

Data betan_boot;
merge beta0 beta1;
_type_=0; run;

/****** Efeitos Aleatórios: v ******/

Data u;
set u;
if estimate='.' then estimate=0;
run;

Data v;
set u;
if Effect='i';
keep i estimate;
run;

Proc sort data=v;
by i; run;

Proc iml;
use v; read all into v;
one=J(7,1,1);
v1=v@one;
create v1 from v1; append from v1; quit;

Data v1;
set v1;
_type_=0;
rename col2=v;
run;

/****** Efeitos Aleatórios: u ******/

Data u1;
set u;
if Effect='i*t';
_type_=0;
keep i t estimate _type_;
rename estimate=u;
run;

Proc sort data=u1;
by i t; run;

Data beta_vu;
merge betan_boot v1 u1;
by _type_;
run;

/** Estimação do preço médio de transacção em
cada NUTIII **/

Proc sort data=cron;
by i t; run;

Data cron2;
merge cron beta_vu;
by i t;
run;

Data est_pc_boot;
set cron2;
e_boot = b0+x*b1 + v + u;
boot=&i;
keep i t x b0 b1 u v e_boot boot;
run;
```

```
/********************************************
********************************************/
/****** Cálculo das estimativas do G3 do EBLUP
utilizando as estimativas das componentes de
variância Boot - Mat_GG_boot - *************/
/********************************************
********************************************/

Data est_pc_mod_g3;
merge est_pc est_pc_boot;
by i t;
DIF2=(e_boot - e)*(e_boot - e);
keep i t e e_boot boot dif2;
run;

/********************************************
********************************************/
/****** Cálculo das estimativas do G1 e G2 do
EBLUP utilizando as estimativas das
componentes de variância Boot  - Mat_GG_boot -
***********/
/********************************************
********************************************/

/*** MATRIZ X ***/

Proc sort data=cron out=cronologico;
by i t; run;

Data XX1;
set cronologico;
keep x;
run;

Proc iml;
J=J(196,1,1);
create J from J;
append from J;
quit;

Proc iml;
use J;
read all into J;
use XX1;
read all into XX1;
X=insert(J,XX1,0,2);
create X from X;
append from X;
quit;

/*** Colunas da Matriz GAMA ***/

Proc iml;
use Gama; read all into Gama;
Gama1=Gama[,1];
create Gama1 from Gama1;
append from Gama1;
quit;
Proc iml;
use Gama; read all into Gama;
Gama2=Gama[,2];
create Gama2 from Gama2;
append from Gama2;
quit;
Proc iml;
use Gama; read all into Gama;
Gama3=Gama[,3];
create Gama3 from Gama3;
append from Gama3;
quit;
Proc iml;
use Gama; read all into Gama;
Gama4=Gama[,4];
create Gama4 from Gama4;
append from Gama4;
quit;
Proc iml;
use Gama; read all into Gama;
Gama5=Gama[,5];
create Gama5 from Gama5;
append from Gama5;
quit;
Proc iml;
use Gama; read all into Gama;
Gama6=Gama[,6];
create Gama6 from Gama6;
append from Gama6;
quit;
Proc iml;
use Gama; read all into Gama;
Gama7=Gama[,7];
create Gama7 from Gama7;
append from Gama7;
quit;

/*** MATRIZ V ***/

Data MatV;
set Mat_v;
drop Index row;
run;

Proc iml;
use MatV; read all into MatV;
MatVinv=inv(MatV);
create MatVinv from MatVinv;
append from MatVinv; quit;

/**************************/
/****** Cálculo do g1 ******/
/**************************/

Data Mat_gg_boot;
set Mat_gg_boot;
drop row;
run;

Proc iml;
use MatVinv; read all into MatVinv;
use Mat_gg_boot; read all into Mat_gg_boot;
use PC.Z; read all into Z;
G1_aux=Z*(Mat_gg_boot -
Mat_gg_boot*t(Z)*MatVinv*Z*Mat_gg_boot)*t(Z);
create G1_aux from G1_aux;
append from G1_aux;
quit;

Proc iml;
use G1_aux;
read all into G1_aux;
G1=vecdiag(G1_aux);
create G1 from G1;
append from G1;
quit;

Data g1_boot;
set g1;
rename col1=g1_boot;
run;

/**************************/
/****** Cálculo do g2 ******/
/**************************/

Proc iml;
use X; read all into X;
use MatVinv; read all into MatVinv;
use Mat_gg_boot; read all into Mat_gg_boot;
use PC.Z; read all into Z;
A=I(196);
G2_aux=t(t(X) -
t(X)*MatVinv*Z*Mat_gg_boot*t(Z))*inv(t(X)*MatV
inv*X)*(t(X) -
t(X)*MatVinv*Z*Mat_gg_boot*t(Z));
create G2_aux from G2_aux;
append from G2_aux;
quit;

Proc iml;
use G2_aux;
read all into G2_aux;
G2=vecdiag(G2_aux);
create G2 from G2;
append from G2;
quit;

Data g2_boot;
set g2;
rename col1=g2_boot;
run;

Data Est_PC_EQM_mod;
merge est_PC_mod_g3 g1_boot g2_boot;
run;
```

```
Proc append base=PC.Gerac_PC_mod111_Boot_S1
data=a8;   /*SSS*/
Proc append base=PC.Est_PC_mod111_boot_S1
data=est_pc_boot;   /*SSS*/
Proc append base=PC.Est_PC_EQM_mod111_Boot_S1
data=Est_pc_EQM_mod;   /*SSS*/
run;

DM 'CLEAR OUTPUT';
DM 'CLEAR LOG';

%mend;
%inc 'C:\Documents and Settings\Luis
Pereira\My
Documents\Doutoramento\chamamacro.txt';

/*********************************************
*********************************************/
/*********************************************
*********************************************/
/*** Estimação do EQMP do EBLUP utilizando a
aproximação Jackknife
************************/
/** Código SAS utilizado depois de estimado o
modelo espaciotemporal para cada conjunto de
dados ***************************************/
/*********************************************
*********************************************/
/*********************************************
*********************************************/

/*** MATRIZ X ***/

Proc sort data=cron out=cronologico;
by i t; run;

Data XX1;
set cronologico;
keep x; run;

Proc iml;
J=J(196,1,1);
create J from J;
append from J; quit;

Proc iml;
use J;
read all into J;
use XX1;
read all into XX1;
X=insert(J,XX1,0,2);
create X from X;
append from X; quit;

/*** Colunas da Matriz GAMA ***/

Proc iml;
use Gama; read all into Gama;
Gama1=Gama[,1];
create Gama1 from Gama1;
append from Gama1;
quit;
Proc iml;
use Gama; read all into Gama;
Gama2=Gama[,2];
create Gama2 from Gama2;
append from Gama2;
quit;
Proc iml;
use Gama; read all into Gama;
Gama3=Gama[,3];
create Gama3 from Gama3;
append from Gama3;
quit;
Proc iml;
use Gama; read all into Gama;
Gama4=Gama[,4];
create Gama4 from Gama4;
append from Gama4;
quit;
Proc iml;
use Gama; read all into Gama;
Gama5=Gama[,5];
create Gama5 from Gama5;
append from Gama5;
quit;
Proc iml;
use Gama; read all into Gama;
```

```
Gama6=Gama[,6];
create Gama6 from Gama6;
append from Gama6;
quit;
Proc iml;
use Gama; read all into Gama;
Gama7=Gama[,7];
create Gama7 from Gama7;
append from Gama7;
quit;

/*** MATRIZ V ***/

Data MatV;
set Mat_v;
drop Index row;
run;

Proc iml;
use MatV; read all into MatV;
MatVinv=inv(MatV);
create MatVinv from MatVinv;
append from MatVinv; quit;

/***************************************/
/****** Cálculo da matriz A_it *********/
/***************************************/

/* É necessário definir as 28 colunas da
matriz inv(B) */

Proc iml;
use Binv; read all into Binv;
Binv1=Binv[,1];
create Binv1 from Binv1;
append from Binv1; quit;
Proc iml;
use Binv; read all into Binv;
Binv2=Binv[,2];
create Binv2 from Binv2;
append from Binv2; quit;
Proc iml;
use Binv; read all into Binv;
Binv3=Binv[,3];
create Binv3 from Binv3;
append from Binv3; quit;
Proc iml;
use Binv; read all into Binv;
Binv4=Binv[,4];
create Binv4 from Binv4;
append from Binv4; quit;
Proc iml;
use Binv; read all into Binv;
Binv5=Binv[,5];
create Binv5 from Binv5;
append from Binv5; quit;
Proc iml;
use Binv; read all into Binv;
Binv6=Binv[,6];
create Binv6 from Binv6;
append from Binv6; quit;
Proc iml;
use Binv; read all into Binv;
Binv7=Binv[,7];
create Binv7 from Binv7;
append from Binv7; quit;
Proc iml;
use Binv; read all into Binv;
Binv8=Binv[,8];
create Binv8 from Binv8;
append from Binv8; quit;
Proc iml;
use Binv; read all into Binv;
Binv9=Binv[,9];
create Binv9 from Binv9;
append from Binv9; quit;
Proc iml;
use Binv; read all into Binv;
Binv10=Binv[,10];
create Binv10 from Binv10;
append from Binv10; quit;
Proc iml;
use Binv; read all into Binv;
Binv11=Binv[,11];
create Binv11 from Binv11;
append from Binv11; quit;
Proc iml;
use Binv; read all into Binv;
```

```
Binv12=Binv[,12];
create Binv12 from Binv12;
append from Binv12; quit;
Proc iml;
use Binv; read all into Binv;
Binv13=Binv[,13];
create Binv13 from Binv13;
append from Binv13; quit;
Proc iml;
use Binv; read all into Binv;
Binv14=Binv[,14];
create Binv14 from Binv14;
append from Binv14; quit;
Proc iml;
use Binv; read all into Binv;
Binv15=Binv[,15];
create Binv15 from Binv15;
append from Binv15; quit;
Proc iml;
use Binv; read all into Binv;
Binv16=Binv[,16];
create Binv16 from Binv16;
append from Binv16; quit;
Proc iml;
use Binv; read all into Binv;
Binv17=Binv[,17];
create Binv17 from Binv17;
append from Binv17; quit;
Proc iml;
use Binv; read all into Binv;
Binv18=Binv[,18];
create Binv18 from Binv18;
append from Binv18; quit;
Proc iml;
use Binv; read all into Binv;
Binv19=Binv[,19];
create Binv19 from Binv19;
append from Binv19; quit;
Proc iml;
use Binv; read all into Binv;
Binv20=Binv[,20];
create Binv20 from Binv20;
append from Binv20; quit;
Proc iml;
use Binv; read all into Binv;
Binv21=Binv[,21];
create Binv21 from Binv21;
append from Binv21; quit;
Proc iml;
use Binv; read all into Binv;
Binv22=Binv[,22];
create Binv22 from Binv22;
append from Binv22; quit;
Proc iml;
use Binv; read all into Binv;
Binv23=Binv[,23];
create Binv23 from Binv23;
append from Binv23; quit;
Proc iml;
use Binv; read all into Binv;
Binv24=Binv[,24];
create Binv24 from Binv24;
append from Binv24; quit;
Proc iml;
use Binv; read all into Binv;
Binv25=Binv[,25];
create Binv25 from Binv25;
append from Binv25; quit;
Proc iml;
use Binv; read all into Binv;
Binv26=Binv[,26];
create Binv26 from Binv26;
append from Binv26; quit;
Proc iml;
use Binv; read all into Binv;
Binv27=Binv[,27];
create Binv27 from Binv27;
append from Binv27; quit;
Proc iml;
use Binv; read all into Binv;
Binv28=Binv[,28];
create Binv28 from Binv28;
append from Binv28; quit;

/*******************************************/
/* Trabalho sobre a matriz A_it - cálculo de
196 matrizes: uma para cada trimestre*dominio
*/

/*******************************************/

/* DOMÍNIO 1 */

Proc iml;
I1={1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
create I1 from I1; append from I1; quit;

/* 1º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama1; read all into Gama1;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama1) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama1) ) )
*MatVinv* ( (I1@Gama1) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama1)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama1; read all into Gama1;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama1) ) )
*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama1) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama1; read all into Gama1;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama1) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama1) ) )*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama1) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A_1=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A_1 from A_1; append from A_1; quit;

/* 2º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama2; read all into Gama2;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama2) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama2) ) )
```

```
*MatVinv* ( (I1@Gama2) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama2)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama2; read all into Gama2;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama2) ) )
*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama2) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama2; read all into Gama2;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama2) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama2) ) )*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama2) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A_2=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A_2 from A_2; append from A_2; quit;

/* 3º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama3; read all into Gama3;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama3) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama3) ) )
*MatVinv* ( (I1@Gama3) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama3)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama3; read all into Gama3;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama3) ) )
*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama3) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama3; read all into Gama3;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama3) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama3) ) )*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama3) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A_3=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A_3 from A_3; append from A_3; quit;

/* 4º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama4; read all into Gama4;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama4) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama4) ) )
*MatVinv* ( (I1@Gama4) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama4)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama4; read all into Gama4;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama4) ) )
*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama4) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama4; read all into Gama4;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama4) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama4) ) )*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama4) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A_4=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A_4 from A_4; append from A_4; quit;

/* 5º trimestre */
```

```
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama5; read all into Gama5;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama5) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama5) ) )
*MatVinv* ( (I1@Gama5) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama5)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama5; read all into Gama5;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama5) ) )
*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama5) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama5; read all into Gama5;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama5) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama5) ) )*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama5) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A_5=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A_5 from A_5; append from A_5; quit;

/* 6º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama6; read all into Gama6;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama6) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama6) ) )
*MatVinv* ( (I1@Gama6) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama6)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama6; read all into Gama6;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama6) ) )
*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama6) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama6; read all into Gama6;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama6) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama6) ) )*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama6) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A_6=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A_6 from A_6; append from A_6; quit;

/* 7º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama7; read all into Gama7;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama7) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama7) ) )
*MatVinv* ( (I1@Gama7) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama7)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama7; read all into Gama7;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama7) ) )
*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama7) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama7; read all into Gama7;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama7) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama7) ) )*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama7) ) );
```

```
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
A_7=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create A_7 from A_7; append from A_7; quit;

/* DOMÍNIO 2 */  (…)
/* DOMÍNIO 3 */  (…)
/* DOMÍNIO 4 */  (…)
 (…)
/* DOMÍNIO 26 */  (…)
/* DOMÍNIO 27 */  (…)
/* DOMÍNIO 28 */  (…)

/* *** Fim *** */

/*********************************************/
/* Cálculo da matriz L_it(y-XB)(Y-
XB)'L'_it=K_it */
/*********************************************/

/*********************************************/
/* Trabalho sobre a matriz L(y-XB)(Y-
XB)'L'=K_it  - calculo de 196 matrizes */
/*********************************************/

Data Res;
set Crono_resf;
keep resid;
run;

/* DOMÍNIO 1 */

Proc iml;
I1={1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
create I1 from I1; append from I1; quit;

/* 1º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama1; read all into Gama1;
use res; read all into res;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama1) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama1) ) )
*MatVinv*res*t(res)*MatVinv* ( (I1@Gama1) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama1)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama1; read all into Gama1;
use res; read all into res;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama1) ) )
*MatVinv*res*t(res)*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama1) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama1; read all into Gama1;
use res; read all into res;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama1) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama1) )
)*MatVinv*res*t(res)*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama1) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
K1=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create K1 from K1; append from K1; quit;

/* 2º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama2; read all into Gama2;
use res; read all into res;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama2) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama2) ) )
*MatVinv*res*t(res)*MatVinv* ( (I1@Gama2) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama2)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama2; read all into Gama2;
use res; read all into res;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama2) ) )
*MatVinv*res*t(res)*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama2) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama2; read all into Gama2;
use res; read all into res;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama2) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama2) )
)*MatVinv*res*t(res)*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama2) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
K2=a11*I1 + a21*I2 + a21*I3 + a22*I4;
```

```
create K2 from K2; append from K2; quit;

/* 3º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama3; read all into Gama3;
use res; read all into res;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama3) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama3) ) )
*MatVinv*res*t(res)*MatVinv* ( (I1@Gama3) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama3)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama3; read all into Gama3;
use res; read all into res;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama3) ) )
*MatVinv*res*t(res)*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama3) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama3; read all into Gama3;
use res; read all into res;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama3) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama3) )
)*MatVinv*res*t(res)*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama3) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
K3=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create K3 from K3; append from K3; quit;

/* 4º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama4; read all into Gama4;
use res; read all into res;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama4) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama4) ) )
*MatVinv*res*t(res)*MatVinv* ( (I1@Gama4) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama4)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama4; read all into Gama4;
use res; read all into res;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama4) ) )
*MatVinv*res*t(res)*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama4) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama4; read all into Gama4;
use res; read all into res;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama4) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama4) )
)*MatVinv*res*t(res)*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama4) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
K4=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create K4 from K4; append from K4; quit;

/* 5º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama5; read all into Gama5;
use res; read all into res;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama5) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama5) ) )
*MatVinv*res*t(res)*MatVinv* ( (I1@Gama5) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama5)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama5; read all into Gama5;
use res; read all into res;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama5) ) )
*MatVinv*res*t(res)*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama5) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
```

```
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama5; read all into Gama5;
use res; read all into res;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama5) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama5) )
)*MatVinv*res*t(res)*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama5) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
K5=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create K5 from K5; append from K5; quit;

/* 6º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama6; read all into Gama6;
use res; read all into res;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama6) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama6) ) )
*MatVinv*res*t(res)*MatVinv* ( (I1@Gama6) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama6)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama6; read all into Gama6;
use res; read all into res;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama6) ) )
*MatVinv*res*t(res)*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama6) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama6; read all into Gama6;
use res; read all into res;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama6) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama6) )
)*MatVinv*res*t(res)*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama6) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
K6=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create K6 from K6; append from K6; quit;

/* 7º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama7; read all into Gama7;
use res; read all into res;
J7=J(7,1,1);
Ident=I(28);
a11=t( (I1@Gama7) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama7) ) )
*MatVinv*res*t(res)*MatVinv* ( (I1@Gama7) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama7)) );
create a11 from a11; append from a11; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama7; read all into Gama7;
use res; read all into res;
J7=J(7,1,1);
Ident=I(28);
a22=t( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama7) ) )
*MatVinv*res*t(res)*MatVinv* ( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama7) ) );
create a22 from a22; append from a22; quit;
Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use Binv; read all into Binv;
use MatVinv; read all into MatVinv;
use Gama7; read all into Gama7;
use res; read all into res;
J7=J(7,1,1);
Ident=I(28);
a21=t( (I1@Gama7) - (Ident@Gama)*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama7) )
)*MatVinv*res*t(res)*MatVinv*
( (Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama7) ) );
create a21 from a21; append from a21; quit;
Proc iml;
use a11; read all into a11;
use a22; read all into a22;
use a21; read all into a21;
I1={1 0,0 0};
I2={0 1,0 0};
I3={0 0,1 0};
I4={0 0,0 1};
K7=a11*I1 + a21*I2 + a21*I3 + a22*I4;
create K7 from K7; append from K7; quit;

/* DOMÍNIO 2 */  (…)
/* DOMÍNIO 3 */  (…)
/* DOMÍNIO 4 */  (…)
 (…)
/* DOMÍNIO 26 */  (…)
/* DOMÍNIO 27 */  (…)
/* DOMÍNIO 28 */  (…)

/* *** Fim *** */

/*****************************************/
/****** Cálculo da derivada de G1_it ******/
/*****************************************/

/*******************************************/
```

```
/* Cálculo da derivada de g1_it  - calculo de
196 matrizes */
/*******************************************/

/* DOMÍNIO 1 */

Proc iml;
I1={1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
create I1 from I1; append from I1; quit;

/* 1º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama1; read all into Gama1;
use ro_1; read all into ro_1;
use Binv; read all into Binv;
J7=J(7,1,1);
Ident=I(28);
dg1_varu= 1/ro_1 - t( 2*(I1@Gama1) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama1) ) )
*MatVinv*( varv*(Binv1@J7) + varu*(I1@Gama1));
dg1_varv= t(Binv1)*I1 - t( 2*(Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama1) ) )
*MatVinv*( varv*(Binv1@J7) + varu*(I1@Gama1));
dg1=dg1_varu//dg1_varv;
create dg1 from dg1; append from dg1; quit;


/* 2º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama2; read all into Gama2;
use ro_1; read all into ro_1;
use Binv; read all into Binv;
J7=J(7,1,1);
Ident=I(28);
dg1_varu= 1/ro_1 - t( 2*(I1@Gama2) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama2) ) )
*MatVinv*( varv*(Binv1@J7) + varu*(I1@Gama2));
dg1_varv= t(Binv1)*I1 - t( 2*(Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama2) ) )
*MatVinv*( varv*(Binv1@J7) + varu*(I1@Gama2));
dg2=dg1_varu//dg1_varv;
create dg2 from dg2; append from dg2; quit;

/* 3º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama3; read all into Gama3;
use ro_1; read all into ro_1;
use Binv; read all into Binv;
J7=J(7,1,1);
Ident=I(28);
dg1_varu= 1/ro_1 - t( 2*(I1@Gama3) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama3) ) )
*MatVinv*( varv*(Binv1@J7) + varu*(I1@Gama3));
dg1_varv= t(Binv1)*I1 - t( 2*(Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama3) ) )
*MatVinv*( varv*(Binv1@J7) + varu*(I1@Gama3));
dg3=dg1_varu//dg1_varv;
create dg3 from dg3; append from dg3; quit;

/* 4º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama4; read all into Gama4;
use ro_1; read all into ro_1;
use Binv; read all into Binv;
J7=J(7,1,1);
Ident=I(28);
dg1_varu= 1/ro_1 - t( 2*(I1@Gama4) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama4) ) )
*MatVinv*( varv*(Binv1@J7) + varu*(I1@Gama4));
dg1_varv= t(Binv1)*I1 - t( 2*(Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama4) ) )
*MatVinv*( varv*(Binv1@J7) + varu*(I1@Gama4));
dg4=dg1_varu//dg1_varv;
create dg4 from dg4; append from dg4; quit;

/* 5º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama5; read all into Gama5;
use ro_1; read all into ro_1;
use Binv; read all into Binv;
J7=J(7,1,1);
Ident=I(28);
dg1_varu= 1/ro_1 - t( 2*(I1@Gama5) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama5) ) )
*MatVinv*( varv*(Binv1@J7) + varu*(I1@Gama5));
dg1_varv= t(Binv1)*I1 - t( 2*(Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama5) ) )
*MatVinv*( varv*(Binv1@J7) + varu*(I1@Gama5));
dg5=dg1_varu//dg1_varv;
create dg5 from dg5; append from dg5; quit;

/* 6º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama6; read all into Gama6;
use ro_1; read all into ro_1;
use Binv; read all into Binv;
J7=J(7,1,1);
Ident=I(28);
dg1_varu= 1/ro_1 - t( 2*(I1@Gama6) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama6) ) )
*MatVinv*( varv*(Binv1@J7) + varu*(I1@Gama6));
dg1_varv= t(Binv1)*I1 - t( 2*(Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama6) ) )
*MatVinv*( varv*(Binv1@J7) + varu*(I1@Gama6));
dg6=dg1_varu//dg1_varv;
create dg6 from dg6; append from dg6; quit;

/* 7º trimestre */

Proc iml;
use I1; read all into I1;
use Binv1; read all into Binv1;
use varv; read all into varv;
use varu; read all into varu;
use Gama; read all into Gama;
use MatVinv; read all into MatVinv;
use Gama7; read all into Gama7;
use ro_1; read all into ro_1;
use Binv; read all into Binv;
J7=J(7,1,1);
Ident=I(28);
```

```
dg1_varu= 1/ro_1 - t( 2*(I1@Gama7) -
(Ident@Gama)*MatVinv*( varv*(Binv1@J7) +
varu*(I1@Gama7) ) )
*MatVinv*( varv*(Binv1@J7) + varu*(I1@Gama7));
dg1_varv= t(Binv1)*I1 - t( 2*(Binv1@J7) -
((Ident@J7)*Binv*t(Ident@J7))*MatVinv*(
varv*(Binv1@J7) + varu*(I1@Gama7) ) )
*MatVinv*( varv*(Binv1@J7) + varu*(I1@Gama7));
dg7=dg1_varu//dg1_varv;
create dg7 from dg7; append from dg7; quit;

/* DOMÍNIO 2 */  (…)
/* DOMÍNIO 3 */  (…)
/* DOMÍNIO 4 */  (…)
 (…)
/* DOMÍNIO 26 */  (…)
/* DOMÍNIO 27 */  (…)
/* DOMÍNIO 28 */  (…)

/* *** Fim *** */

/*******************************************/
/*******************************************/
/****** Cálculo dos ponderadores Wu ********/
/*******************************************/
/*******************************************/

/*** Wu = (m-1)/m = 27/28 ***/
/*** O vector Wu tem dimensão 7*1 porque ao
eliminar um domínio
estou a eliminar 7 observações ***/

Proc iml;
Wu=(27/28);
create Wu from Wu; append from Wu; quit; /* É
igual para todos os domínios

/*** Cálculo das estimativas das componentes
de variância excluindo observações - são
excluídas de cada vez todas as 7 observações
de um dado domínio ***/

Data null;
file 'C:\Documents and Settings\Luis
Pereira\My
Documents\Doutoramento\chamamacro.txt';
   do l= 1 to 28; k=l;
var='%ext_am('||l||','||k||');'; put var; end;
run;

%macro ext_am(l,k);

DM 'CLEAR OUTPUT';
DM 'CLEAR LOG';

/*******************************************/
/**** Eliminação de uma linha/coluna de B ***/
/*******************************************/

/* Matriz B=(I-phi*W)'*(I-phi*W) */

Data W;
set PC.W2;   /* Matriz de distâncias */
run;

Proc iml;
use W; read all into W;
Row=(1:28);
W_=W||t(Row);
create W_ from W_; append from W_; quit;

Data W_&l;
set W_;
drop col&l;
if col29 ne &l;
run;

Data W_&l;
set W_&l;
drop col29;
run;

Proc iml;
use W_&l; read all into W_&l;
use dataphi; read all into dataphi;
Ident=I(27);
Binv_&l = inv( t(Ident - dataphi*W_&l)*(Ident
- dataphi*W_&l) );
create Binv_&l from Binv_&l; append from
Binv_&l; quit;

/*******************************************/
/**** Eliminação de um conjunto de dados ****/
/*******************************************/

Data a61;
set a6;
rename i=j;
run;

Data k;
        do i=1 to 27;
                do v=1 to 7;
                p=1;
                output;
                end;
        end;
keep i; run;

Proc sql;
delete from a61 where j=&l;   /* Eliminação
dos dados de um domínio j */

Data a62;
merge k a61;
run;

Data iabh;
set a62;
keep i t c x;
run;

Data ipth;
set a62;
keep i t y;
run;

Data iabhipth1;
set a62;
keep i t y x;
run;

/** Preparação do ficheiro das observações da
variável de interesse **/

Proc sort data=ipth;
by i t;
run;

Data tetad1; set ipth; where i= 1; keep y;
Data tetad2; set ipth; where i= 2; keep y;
Data tetad3; set ipth; where i= 3; keep y;
Data tetad4; set ipth; where i= 4; keep y;
Data tetad5; set ipth; where i= 5; keep y;
Data tetad6; set ipth; where i= 6; keep y;
Data tetad7; set ipth; where i= 7; keep y;
Data tetad8; set ipth; where i= 8; keep y;
Data tetad9; set ipth; where i= 9; keep y;
Data tetad10; set ipth; where i= 10; keep y;
Data tetad11; set ipth; where i= 11; keep y;
Data tetad12; set ipth; where i= 12; keep y;
Data tetad13; set ipth; where i= 13; keep y;
Data tetad14; set ipth; where i= 14; keep y;
Data tetad15; set ipth; where i= 15; keep y;
Data tetad16; set ipth; where i= 16; keep y;
Data tetad17; set ipth; where i= 17; keep y;
Data tetad18; set ipth; where i= 18; keep y;
Data tetad19; set ipth; where i= 19; keep y;
Data tetad20; set ipth; where i= 20; keep y;
Data tetad21; set ipth; where i= 21; keep y;
Data tetad22; set ipth; where i= 22; keep y;
Data tetad23; set ipth; where i= 23; keep y;
Data tetad24; set ipth; where i= 24; keep y;
Data tetad25; set ipth; where i= 25; keep y;
Data tetad26; set ipth; where i= 26; keep y;
Data tetad27; set ipth; where i= 27; keep y;
run;

/** Preparação do ficheiro das observações da
variável auxiliar **/

Proc sort data=iabh;
by i t;
run;
```

```
Proc sql; create table iabh0 as select c, x,
i, t from iabh;

Data xd1; set iabh0; where i= 1; keep c x;
Data xd2; set iabh0; where i= 2; keep c x;
Data xd3; set iabh0; where i= 3; keep c x;
Data xd4; set iabh0; where i= 4; keep c x;
Data xd5; set iabh0; where i= 5; keep c x;
Data xd6; set iabh0; where i= 6; keep c x;
Data xd7; set iabh0; where i= 7; keep c x;
Data xd8; set iabh0; where i= 8; keep c x;
Data xd9; set iabh0; where i= 9; keep c x;
Data xd10; set iabh0; where i= 10; keep c x;
Data xd11; set iabh0; where i= 11; keep c x;
Data xd12; set iabh0; where i= 12; keep c x;
Data xd13; set iabh0; where i= 13; keep c x;
Data xd14; set iabh0; where i= 14; keep c x;
Data xd15; set iabh0; where i= 15; keep c x;
Data xd16; set iabh0; where i= 16; keep c x;
Data xd17; set iabh0; where i= 17; keep c x;
Data xd18; set iabh0; where i= 18; keep c x;
Data xd19; set iabh0; where i= 19; keep c x;
Data xd20; set iabh0; where i= 20; keep c x;
Data xd21; set iabh0; where i= 21; keep c x;
Data xd22; set iabh0; where i= 22; keep c x;
Data xd23; set iabh0; where i= 23; keep c x;
Data xd24; set iabh0; where i= 24; keep c x;
Data xd25; set iabh0; where i= 25; keep c x;
Data xd26; set iabh0; where i= 26; keep c x;
Data xd27; set iabh0; where i= 27; keep c x;
run;

/** Cálculo da matriz Z(1) **/

Proc iml;
use tetad1; read all into tetad1;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd1=(Ident - D)*P*tetad1;
create zd1 from zd1; append from zd1; quit;
Proc iml;
use tetad2; read all into tetad2;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd2=(Ident - D)*P*tetad2;
create zd2 from zd2; append from zd2; quit;
Proc iml;
use tetad3; read all into tetad3;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd3=(Ident - D)*P*tetad3;
create zd3 from zd3; append from zd3; quit;
Proc iml;
use tetad4; read all into tetad4;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd4=(Ident - D)*P*tetad4;
create zd4 from zd4; append from zd4; quit;
Proc iml;
use tetad5; read all into tetad5;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd5=(Ident - D)*P*tetad5;
create zd5 from zd5; append from zd5; quit;
Proc iml;
use tetad6; read all into tetad6;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd6=(Ident - D)*P*tetad6;
create zd6 from zd6; append from zd6; quit;
Proc iml;
use tetad7; read all into tetad7;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd7=(Ident - D)*P*tetad7;
create zd7 from zd7; append from zd7; quit;
Proc iml;
use tetad8; read all into tetad8;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd8=(Ident - D)*P*tetad8;
create zd8 from zd8; append from zd8; quit;
Proc iml;
use tetad9; read all into tetad9;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd9=(Ident - D)*P*tetad9;
create zd9 from zd9; append from zd9; quit;
Proc iml;
use tetad10; read all into tetad10;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd10=(Ident - D)*P*tetad10;
create zd10 from zd10; append from zd10; quit;
Proc iml;
use tetad11; read all into tetad11;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd11=(Ident - D)*P*tetad11;
create zd11 from zd11; append from zd11; quit;
Proc iml;
use tetad12; read all into tetad12;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd12=(Ident - D)*P*tetad12;
create zd12 from zd12; append from zd12; quit;
Proc iml;
use tetad13; read all into tetad13;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd13=(Ident - D)*P*tetad13;
create zd13 from zd13; append from zd13; quit;
Proc iml;
use tetad14; read all into tetad14;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd14=(Ident - D)*P*tetad14;
create zd14 from zd14; append from zd14; quit;
Proc iml;
use tetad15; read all into tetad15;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd15=(Ident - D)*P*tetad15;
create zd15 from zd15; append from zd15; quit;
Proc iml;
use tetad16; read all into tetad16;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd16=(Ident - D)*P*tetad16;
create zd16 from zd16; append from zd16; quit;
Proc iml;
use tetad17; read all into tetad17;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd17=(Ident - D)*P*tetad17;
create zd17 from zd17; append from zd17; quit;
Proc iml;
use tetad18; read all into tetad18;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd18=(Ident - D)*P*tetad18;
create zd18 from zd18; append from zd18; quit;
Proc iml;
use tetad19; read all into tetad19;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd19=(Ident - D)*P*tetad19;
create zd19 from zd19; append from zd19; quit;
Proc iml;
use tetad20; read all into tetad20;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd20=(Ident - D)*P*tetad20;
create zd20 from zd20; append from zd20; quit;
Proc iml;
use tetad21; read all into tetad21;
```

```
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd21=(Ident - D)*P*tetad21;
create zd21 from zd21; append from zd21; quit;
Proc iml;
use tetad22; read all into tetad22;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd22=(Ident - D)*P*tetad22;
create zd22 from zd22; append from zd22; quit;
Proc iml;
use tetad23; read all into tetad23;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd23=(Ident - D)*P*tetad23;
create zd23 from zd23; append from zd23; quit;
Proc iml;
use tetad24; read all into tetad24;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd24=(Ident - D)*P*tetad24;
create zd24 from zd24; append from zd24; quit;
Proc iml;
use tetad25; read all into tetad25;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd25=(Ident - D)*P*tetad25;
create zd25 from zd25; append from zd25; quit;
Proc iml;
use tetad26; read all into tetad26;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd26=(Ident - D)*P*tetad26;
create zd26 from zd26; append from zd26; quit;
Proc iml;
use tetad27; read all into tetad27;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd27=(Ident - D)*P*tetad27;
create zd27 from zd27; append from zd27; quit;

Data Z1;
set zd1 zd2 zd3 zd4 zd5 zd6 zd7 zd8 zd9 zd10
zd11 zd12 zd13 zd14 zd15 zd16 zd17 zd18 zd19
zd20 zd21 zd22 zd23 zd24 zd25 zd26 zd27;
run;

/** Cálculo da matriz H(1) **/

Proc iml;
use xd1; read all into xd1;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd1=(Ident - D)*P*xd1;
create hd1 from hd1; append from hd1; quit;
Proc iml;
use xd2; read all into xd2;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd2=(Ident - D)*P*xd2;
create hd2 from hd2; append from hd2; quit;
Proc iml;
use xd3; read all into xd3;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd3=(Ident - D)*P*xd3;
create hd3 from hd3; append from hd3; quit;
Proc iml;
use xd4; read all into xd4;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd4=(Ident - D)*P*xd4;
create hd4 from hd4; append from hd4; quit;
Proc iml;
use xd5; read all into xd5;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd5=(Ident - D)*P*xd5;
create hd5 from hd5; append from hd5; quit;
Proc iml;
use xd6; read all into xd6;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd6=(Ident - D)*P*xd6;
create hd6 from hd6; append from hd6; quit;
Proc iml;
use xd7; read all into xd7;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd7=(Ident - D)*P*xd7;
create hd7 from hd7; append from hd7; quit;
Proc iml;
use xd8; read all into xd8;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd8=(Ident - D)*P*xd8;
create hd8 from hd8; append from hd8; quit;
Proc iml;
use xd9; read all into xd9;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd9=(Ident - D)*P*xd9;
create hd9 from hd9; append from hd9; quit;
Proc iml;
use xd10; read all into xd10;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd10=(Ident - D)*P*xd10;
create hd10 from hd10; append from hd10; quit;
Proc iml;
use xd11; read all into xd11;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd11=(Ident - D)*P*xd11;
create hd11 from hd11; append from hd11; quit;
Proc iml;
use xd12; read all into xd12;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd12=(Ident - D)*P*xd12;
create hd12 from hd12; append from hd12; quit;
Proc iml;
use xd13; read all into xd13;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd13=(Ident - D)*P*xd13;
create hd13 from hd13; append from hd13; quit;
Proc iml;
use xd14; read all into xd14;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd14=(Ident - D)*P*xd14;
create hd14 from hd14; append from hd14; quit;
Proc iml;
use xd15; read all into xd15;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd15=(Ident - D)*P*xd15;
create hd15 from hd15; append from hd15; quit;
Proc iml;
use xd16; read all into xd16;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd16=(Ident - D)*P*xd16;
create hd16 from hd16; append from hd16; quit;
Proc iml;
use xd17; read all into xd17;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd17=(Ident - D)*P*xd17;
create hd17 from hd17; append from hd17; quit;
Proc iml;
```

```
use xd18; read all into xd18;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd18=(Ident - D)*P*xd18;
create hd18 from hd18; append from hd18; quit;
Proc iml;
use xd19; read all into xd19;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd19=(Ident - D)*P*xd19;
create hd19 from hd19; append from hd19; quit;
Proc iml;
use xd20; read all into xd20;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd20=(Ident - D)*P*xd20;
create hd20 from hd20; append from hd20; quit;
Proc iml;
use xd21; read all into xd21;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd21=(Ident - D)*P*xd21;
create hd21 from hd21; append from hd21; quit;
Proc iml;
use xd22; read all into xd22;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd22=(Ident - D)*P*xd22;
create hd22 from hd22; append from hd22; quit;
Proc iml;
use xd23; read all into xd23;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd23=(Ident - D)*P*xd23;
create hd23 from hd23; append from hd23; quit;
Proc iml;
use xd24; read all into xd24;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd24=(Ident - D)*P*xd24;
create hd24 from hd24; append from hd24; quit;
Proc iml;
use xd25; read all into xd25;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd25=(Ident - D)*P*xd25;
create hd25 from hd25; append from hd25; quit;
Proc iml;
use xd26; read all into xd26;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd26=(Ident - D)*P*xd26;
create hd26 from hd26; append from hd26; quit;
Proc iml;
use xd27; read all into xd27;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd27=(Ident - D)*P*xd27;
create hd27 from hd27; append from hd27; quit;

Data H1;
set hd1 hd2 hd3 hd4 hd5 hd6 hd7 hd8 hd9 hd10
hd11 hd12 hd13 hd14 hd15 hd16 hd17 hd18 hd19
hd20 hd21 hd22 hd23 hd24 hd25 hd26 hd27;
run;

/** Regressão de Z(1) sobre H(1) **/

Data Z1;
set Z1;
rename col1=col0; run;

Data reg;
merge Z1 H1; run;

proc reg data=reg noprint;
        model col0=col1 col2 / noint;
        output out=rg1  residual=r1;
run; quit;

Data rg1;
set rg1;
qr1=r1*r1;
Keep qr1;
run;

Proc means data=rg1 sum noprint;
var qr1;
output out=rg1_ sum=sqr1;
run;

Data rg1_;
set rg1_;
keep sqr1; run;

/** Cálculo da variância sigma2_u **/

Proc iml;
use H1; read all into H1;
Hs1=H1*ginv(t(H1)*H1)*t(H1);
create Hs1 from Hs1; append from Hs1; quit;

Proc iml;
use D; read all into D;
Ident=I(7);
ID=Ident - D; /* Matriz I - D */
create ID from ID; append from ID; quit;

Proc iml;
use ID; read all into ID;
blockID1=BLOCK(ID,ID,ID,ID,ID,ID,ID,ID,ID,ID,I
D,ID,ID,ID);
create blockID1 from blockID1; append from
blockID1; quit;
Proc iml;
use ID; read all into ID;
blockID2=BLOCK(ID,ID,ID,ID,ID,ID,ID,ID,ID,ID,I
D,ID,ID);
create blockID2 from blockID2; append from
blockID2; quit;
Proc iml;
use blockID1; read all into blockID1;
use blockID2; read all into blockID2;
blockID=BLOCK(blockID1,blockID2);
create blockID from blockID; append from
blockID; quit;

Data s1;
set a6;
keep c t i;
run;

Data s1;
set s1;
rename c=var; run;

Data Sigma_1;
set s1;
if i=1;
keep var;
run;
Proc iml;
use Sigma_1; read all into Sigma_1;
Sigma1=diag(Sigma_1);
create Sigma1 from Sigma1; append from Sigma1;
quit;
Proc iml;
use P; read all into P;
use Sigma1; read all into Sigma1;
P1=P*Sigma1*t(P);
create P1 from P1; append from P1; quit;

Data Sigma_2;
set s1;
if i=2;
keep var;
run;
Proc iml;
use Sigma_2; read all into Sigma_2;
Sigma2=diag(Sigma_2);
create Sigma2 from Sigma2; append from Sigma2;
quit;
Proc iml;
use P; read all into P;
use Sigma2; read all into Sigma2;
P2=P*Sigma2*t(P);
```

```
create P2 from P2; append from P2; quit;

Data Sigma_3;
set s1;
if i=3;
keep var;
run;
Proc iml;
use Sigma_3; read all into Sigma_3;
Sigma3=diag(Sigma_3);
create Sigma3 from Sigma3; append from Sigma3;
quit;
Proc iml;
use P; read all into P;
use Sigma3; read all into Sigma3;
P3=P*Sigma3*t(P);
create P3 from P3; append from P3; quit;

Data Sigma_4;
set s1;
if i=4;
keep var;
run;
Proc iml;
use Sigma_4; read all into Sigma_4;
Sigma4=diag(Sigma_4);
create Sigma4 from Sigma4; append from Sigma4;
quit;
Proc iml;
use P; read all into P;
use Sigma4; read all into Sigma4;
P4=P*Sigma4*t(P);
create P4 from P4; append from P4; quit;

Data Sigma_5;
set s1;
if i=5;
keep var;
run;
Proc iml;
use Sigma_5; read all into Sigma_5;
Sigma5=diag(Sigma_5);
create Sigma5 from Sigma5; append from Sigma5;
quit;
Proc iml;
use P; read all into P;
use Sigma5; read all into Sigma5;
P5=P*Sigma5*t(P);
create P5 from P5; append from P5; quit;

Data Sigma_6;
set s1;
if i=6;
keep var;
run;
Proc iml;
use Sigma_6; read all into Sigma_6;
Sigma6=diag(Sigma_6);
create Sigma6 from Sigma6; append from Sigma6;
quit;
Proc iml;
use P; read all into P;
use Sigma6; read all into Sigma6;
P6=P*Sigma6*t(P);
create P6 from P6; append from P6; quit;

Data Sigma_7;
set s1;
if i=7;
keep var;
run;
Proc iml;
use Sigma_7; read all into Sigma_7;
Sigma7=diag(Sigma_7);
create Sigma7 from Sigma7; append from Sigma7;
quit;
Proc iml;
use P; read all into P;
use Sigma7; read all into Sigma7;
P7=P*Sigma7*t(P);
create P7 from P7; append from P7; quit;

Data Sigma_8;
set s1;
if i=8;
keep var;
run;
Proc iml;
use Sigma_8; read all into Sigma_8;
Sigma8=diag(Sigma_8);
create Sigma8 from Sigma8; append from Sigma8;
quit;
Proc iml;
use P; read all into P;
use Sigma8; read all into Sigma8;
P8=P*Sigma8*t(P);
create P8 from P8; append from P8; quit;

Data Sigma_9;
set s1;
if i=9;
keep var;
run;
Proc iml;
use Sigma_9; read all into Sigma_9;
Sigma9=diag(Sigma_9);
create Sigma9 from Sigma9; append from Sigma9;
quit;
Proc iml;
use P; read all into P;
use Sigma9; read all into Sigma9;
P9=P*Sigma9*t(P);
create P9 from P9; append from P9; quit;

Data Sigma_10;
set s1;
if i=10;
keep var;
run;
Proc iml;
use Sigma_10; read all into Sigma_10;
Sigma10=diag(Sigma_10);
create Sigma10 from Sigma10; append from
Sigma10; quit;
Proc iml;
use P; read all into P;
use Sigma10; read all into Sigma10;
P10=P*Sigma10*t(P);
create P10 from P10; append from P10; quit;

Data Sigma_11;
set s1;
if i=11;
keep var;
run;
Proc iml;
use Sigma_11; read all into Sigma_11;
Sigma11=diag(Sigma_11);
create Sigma11 from Sigma11; append from
Sigma11; quit;
Proc iml;
use P; read all into P;
use Sigma11; read all into Sigma11;
P11=P*Sigma11*t(P);
create P11 from P11; append from P11; quit;

Data Sigma_12;
set s1;
if i=12;
keep var;
run;
Proc iml;
use Sigma_12; read all into Sigma_12;
Sigma12=diag(Sigma_12);
create Sigma12 from Sigma12; append from
Sigma12; quit;
Proc iml;
use P; read all into P;
use Sigma12; read all into Sigma12;
P12=P*Sigma12*t(P);
create P12 from P12; append from P12; quit;

Data Sigma_13;
set s1;
if i=13;
keep var;
run;
Proc iml;
use Sigma_13; read all into Sigma_13;
Sigma13=diag(Sigma_13);
create Sigma13 from Sigma13; append from
Sigma13; quit;
Proc iml;
use P; read all into P;
use Sigma13; read all into Sigma13;
P13=P*Sigma13*t(P);
```

```
create P13 from P13; append from P13; quit;

Data Sigma_14;
set s1;
if i=14;
keep var;
run;
Proc iml;
use Sigma_14; read all into Sigma_14;
Sigma14=diag(Sigma_14);
create Sigma14 from Sigma14; append from
Sigma14; quit;
Proc iml;
use P; read all into P;
use Sigma14; read all into Sigma14;
P14=P*Sigma14*t(P);
create P14 from P14; append from P14; quit;

Data Sigma_15;
set s1;
if i=15;
keep var;
run;
Proc iml;
use Sigma_15; read all into Sigma_15;
Sigma15=diag(Sigma_15);
create Sigma15 from Sigma15; append from
Sigma15; quit;
Proc iml;
use P; read all into P;
use Sigma15; read all into Sigma15;
P15=P*Sigma15*t(P);
create P15 from P15; append from P15; quit;

Data Sigma_16;
set s1;
if i=16;
keep var;
run;
Proc iml;
use Sigma_16; read all into Sigma_16;
Sigma16=diag(Sigma_16);
create Sigma16 from Sigma16; append from
Sigma16; quit;
Proc iml;
use P; read all into P;
use Sigma16; read all into Sigma16;
P16=P*Sigma16*t(P);
create P16 from P16; append from P16; quit;

Data Sigma_17;
set s1;
if i=17;
keep var;
run;
Proc iml;
use Sigma_17; read all into Sigma_17;
Sigma17=diag(Sigma_17);
create Sigma17 from Sigma17; append from
Sigma17; quit;
Proc iml;
use P; read all into P;
use Sigma17; read all into Sigma17;
P17=P*Sigma17*t(P);
create P17 from P17; append from P17; quit;

Data Sigma_18;
set s1;
if i=18;
keep var;
run;
Proc iml;
use Sigma_18; read all into Sigma_18;
Sigma18=diag(Sigma_18);
create Sigma18 from Sigma18; append from
Sigma18; quit;
Proc iml;
use P; read all into P;
use Sigma18; read all into Sigma18;
P18=P*Sigma18*t(P);
create P18 from P18; append from P18; quit;

Data Sigma_19;
set s1;
if i=19;
keep var;
run;
Proc iml;
use Sigma_19; read all into Sigma_19;
Sigma19=diag(Sigma_19);
create Sigma19 from Sigma19; append from
Sigma19; quit;
Proc iml;
use P; read all into P;
use Sigma19; read all into Sigma19;
P19=P*Sigma19*t(P);
create P19 from P19; append from P19; quit;

Data Sigma_20;
set s1;
if i=20;
keep var;
run;
Proc iml;
use Sigma_20; read all into Sigma_20;
Sigma20=diag(Sigma_20);
create Sigma20 from Sigma20; append from
Sigma20; quit;
Proc iml;
use P; read all into P;
use Sigma20; read all into Sigma20;
P20=P*Sigma20*t(P);
create P20 from P20; append from P20; quit;

Data Sigma_21;
set s1;
if i=21;
keep var;
run;
Proc iml;
use Sigma_21; read all into Sigma_21;
Sigma21=diag(Sigma_21);
create Sigma21 from Sigma21; append from
Sigma21; quit;
Proc iml;
use P; read all into P;
use Sigma21; read all into Sigma21;
P21=P*Sigma21*t(P);
create P21 from P21; append from P21; quit;

Data Sigma_22;
set s1;
if i=22;
keep var;
run;
Proc iml;
use Sigma_22; read all into Sigma_22;
Sigma22=diag(Sigma_22);
create Sigma22 from Sigma22; append from
Sigma22; quit;
Proc iml;
use P; read all into P;
use Sigma22; read all into Sigma22;
P22=P*Sigma22*t(P);
create P22 from P22; append from P22; quit;

Data Sigma_23;
set s1;
if i=23;
keep var;
run;
Proc iml;
use Sigma_23; read all into Sigma_23;
Sigma23=diag(Sigma_23);
create Sigma23 from Sigma23; append from
Sigma23; quit;
Proc iml;
use P; read all into P;
use Sigma23; read all into Sigma23;
P23=P*Sigma23*t(P);
create P23 from P23; append from P23; quit;

Data Sigma_24;
set s1;
if i=24;
keep var;
run;
Proc iml;
use Sigma_24; read all into Sigma_24;
Sigma24=diag(Sigma_24);
create Sigma24 from Sigma24; append from
Sigma24; quit;
Proc iml;
use P; read all into P;
use Sigma24; read all into Sigma24;
P24=P*Sigma24*t(P);
```

```
create P24 from P24; append from P24; quit;

Data Sigma_25;
set s1;
if i=25;
keep var;
run;
Proc iml;
use Sigma_25; read all into Sigma_25;
Sigma25=diag(Sigma_25);
create Sigma25 from Sigma25; append from
Sigma25; quit;
Proc iml;
use P; read all into P;
use Sigma25; read all into Sigma25;
P25=P*Sigma25*t(P);
create P25 from P25; append from P25; quit;

Data Sigma_26;
set s1;
if i=26;
keep var;
run;
Proc iml;
use Sigma_26; read all into Sigma_26;
Sigma26=diag(Sigma_26);
create Sigma26 from Sigma26; append from
Sigma26; quit;
Proc iml;
use P; read all into P;
use Sigma26; read all into Sigma26;
P26=P*Sigma26*t(P);
create P26 from P26; append from P26; quit;

Data Sigma_27;
set s1;
if i=27;
keep var;
run;
Proc iml;
use Sigma_27; read all into Sigma_27;
Sigma27=diag(Sigma_27);
create Sigma27 from Sigma27; append from
Sigma27; quit;
Proc iml;
use P; read all into P;
use Sigma27; read all into Sigma27;
P27=P*Sigma27*t(P);
create P27 from P27; append from P27; quit;

Proc iml;
use P1; read all into P1;
use P2; read all into P2;
use P3; read all into P3;
use P4; read all into P4;
use P5; read all into P5;
use P6; read all into P6;
use P7; read all into P7;
use P8; read all into P8;
use P9; read all into P9;
use P10; read all into P10;
use P11; read all into P11;
use P12; read all into P12;
use P13; read all into P13;
use P14; read all into P14;
blockPP1=BLOCK(P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,
P11,P12,P13,P14);
create blockPP1 from blockPP1; append from
blockPP1; quit;

Proc iml;
use P15; read all into P15;
use P16; read all into P16;
use P17; read all into P17;
use P18; read all into P18;
use P19; read all into P19;
use P20; read all into P20;
use P21; read all into P21;
use P22; read all into P22;
use P23; read all into P23;
use P24; read all into P24;
use P25; read all into P25;
use P26; read all into P26;
use P27; read all into P27;
blockPP2=BLOCK(P15,P16,P17,P18,P19,P20,P21,P22
,P23,P24,P25,P26,P27);
create blockPP2 from blockPP2; append from
blockPP2; quit;

Proc iml;
use blockPP1; read all into blockPP1;
use blockPP2; read all into blockPP2;
blockPP=BLOCK(blockPP1,blockPP2);
create blockPP from blockPP; append from
blockPP; quit;

/* *** Fim *** */

Proc iml;
use H1; read all into H1;
HH=H1*t(H1);
eval=eigval(HH);
create eval from eval; append from eval; quit;

Data id;
do id=1 to 189;
output; end;
run;

Data eval;
merge id eval;
run;

Data new;
set eval;
by id;
retain count 0;
if first.id then count=0;
if col1 > 0.000001 then count = count + 1;
run;

Proc means data=new sum noprint;
var count;
output out=caracH1 sum=caracH1;
run;

Data caracH1;
set caracH1;
keep caracH1;
run;

Proc iml;
use blockPP; read all into blockPP;
use blockID; read all into blockID;
use Hs1; read all into Hs1;
use rg1_; read all into rg1_;
use caracH1; read all into caracH1;
varu= (rg1_ -  trace((blockID -
Hs1)*blockPP))*(1/(27*6 - caracH1));
create varu from varu; append from varu; quit;

/* Termina aqui o cálculo da variância
sigma2_u */

/** Cálculo da matriz: z2= c^(-0.5)*f'*P*Teta
- var. dependente do modelo transformado **/

Proc iml;
use tetad1; read all into tetad1;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y1=(1/sqrt(c))*t(f)*P*tetad1;
create y1 from y1; append from y1; quit;
Proc iml;
use tetad2; read all into tetad2;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y2=(1/sqrt(c))*t(f)*P*tetad2;
create y2 from y2; append from y2; quit;
Proc iml;
use tetad3; read all into tetad3;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y3=(1/sqrt(c))*t(f)*P*tetad3;
create y3 from y3; append from y3; quit;
Proc iml;
use tetad4; read all into tetad4;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y4=(1/sqrt(c))*t(f)*P*tetad4;
create y4 from y4; append from y4; quit;
Proc iml;
```

```
use tetad5; read all into tetad5;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y5=(1/sqrt(c))*t(f)*P*tetad5;
create y5 from y5; append from y5; quit;
Proc iml;
use tetad6; read all into tetad6;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y6=(1/sqrt(c))*t(f)*P*tetad6;
create y6 from y6; append from y6; quit;
Proc iml;
use tetad7; read all into tetad7;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y7=(1/sqrt(c))*t(f)*P*tetad7;
create y7 from y7; append from y7; quit;
Proc iml;
use tetad8; read all into tetad8;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y8=(1/sqrt(c))*t(f)*P*tetad8;
create y8 from y8; append from y8; quit;
Proc iml;
use tetad9; read all into tetad9;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y9=(1/sqrt(c))*t(f)*P*tetad9;
create y9 from y9; append from y9; quit;
Proc iml;
use tetad10; read all into tetad10;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y10=(1/sqrt(c))*t(f)*P*tetad10;
create y10 from y10; append from y10; quit;
Proc iml;
use tetad11; read all into tetad11;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y11=(1/sqrt(c))*t(f)*P*tetad11;
create y11 from y11; append from y11; quit;
Proc iml;
use tetad12; read all into tetad12;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y12=(1/sqrt(c))*t(f)*P*tetad12;
create y12 from y12; append from y12; quit;
Proc iml;
use tetad13; read all into tetad13;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y13=(1/sqrt(c))*t(f)*P*tetad13;
create y13 from y13; append from y13; quit;
Proc iml;
use tetad14; read all into tetad14;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y14=(1/sqrt(c))*t(f)*P*tetad14;
create y14 from y14; append from y14; quit;
Proc iml;
use tetad15; read all into tetad15;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y15=(1/sqrt(c))*t(f)*P*tetad15;
create y15 from y15; append from y15; quit;
Proc iml;
use tetad16; read all into tetad16;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y16=(1/sqrt(c))*t(f)*P*tetad16;
create y16 from y16; append from y16; quit;
Proc iml;
use tetad17; read all into tetad17;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y17=(1/sqrt(c))*t(f)*P*tetad17;
create y17 from y17; append from y17; quit;
Proc iml;
use tetad18; read all into tetad18;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y18=(1/sqrt(c))*t(f)*P*tetad18;
create y18 from y18; append from y18; quit;
Proc iml;
use tetad19; read all into tetad19;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y19=(1/sqrt(c))*t(f)*P*tetad19;
create y19 from y19; append from y19; quit;
Proc iml;
use tetad20; read all into tetad20;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y20=(1/sqrt(c))*t(f)*P*tetad20;
create y20 from y20; append from y20; quit;
Proc iml;
use tetad21; read all into tetad21;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y21=(1/sqrt(c))*t(f)*P*tetad21;
create y21 from y21; append from y21; quit;
Proc iml;
use tetad22; read all into tetad22;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y22=(1/sqrt(c))*t(f)*P*tetad22;
create y22 from y22; append from y22; quit;
Proc iml;
use tetad23; read all into tetad23;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y23=(1/sqrt(c))*t(f)*P*tetad23;
create y23 from y23; append from y23; quit;
Proc iml;
use tetad24; read all into tetad24;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y24=(1/sqrt(c))*t(f)*P*tetad24;
create y24 from y24; append from y24; quit;
Proc iml;
use tetad25; read all into tetad25;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y25=(1/sqrt(c))*t(f)*P*tetad25;
create y25 from y25; append from y25; quit;
Proc iml;
use tetad26; read all into tetad26;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y26=(1/sqrt(c))*t(f)*P*tetad26;
create y26 from y26; append from y26; quit;
Proc iml;
use tetad27; read all into tetad27;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y27=(1/sqrt(c))*t(f)*P*tetad27;
create y27 from y27; append from y27; quit;

Data z2;
set y1 y2 y3 y4 y5 y6 y7 y8 y9 y10 y11 y12 y13
y14 y15 y16 y17 y18 y19 y20 y21 y22 y23 y24
y25 y26 y27;
run;

/** Cálculo da matriz: H2= c^(-0.5)*f'*P*X -
var. independentes do modelo transformado **/

Proc iml;
use xd1; read all into xd1;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x1=(1/sqrt(c))*t(f)*P*xd1;
```

```
create x1 from x1; append from x1; quit;
Proc iml;
use xd2; read all into xd2;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x2=(1/sqrt(c))*t(f)*P*xd2;
create x2 from x2; append from x2; quit;
Proc iml;
use xd3; read all into xd3;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x3=(1/sqrt(c))*t(f)*P*xd3;
create x3 from x3; append from x3; quit;
Proc iml;
use xd4; read all into xd4;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x4=(1/sqrt(c))*t(f)*P*xd4;
create x4 from x4; append from x4; quit;
Proc iml;
use xd5; read all into xd5;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x5=(1/sqrt(c))*t(f)*P*xd5;
create x5 from x5; append from x5; quit;
Proc iml;
use xd6; read all into xd6;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x6=(1/sqrt(c))*t(f)*P*xd6;
create x6 from x6; append from x6; quit;
Proc iml;
use xd7; read all into xd7;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x7=(1/sqrt(c))*t(f)*P*xd7;
create x7 from x7; append from x7; quit;
Proc iml;
use xd8; read all into xd8;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x8=(1/sqrt(c))*t(f)*P*xd8;
create x8 from x8; append from x8; quit;
Proc iml;
use xd9; read all into xd9;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x9=(1/sqrt(c))*t(f)*P*xd9;
create x9 from x9; append from x9; quit;
Proc iml;
use xd10; read all into xd10;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x10=(1/sqrt(c))*t(f)*P*xd10;
create x10 from x10; append from x10; quit;
Proc iml;
use xd11; read all into xd11;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x11=(1/sqrt(c))*t(f)*P*xd11;
create x11 from x11; append from x11; quit;
Proc iml;
use xd12; read all into xd12;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x12=(1/sqrt(c))*t(f)*P*xd12;
create x12 from x12; append from x12; quit;
Proc iml;
use xd13; read all into xd13;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x13=(1/sqrt(c))*t(f)*P*xd13;
create x13 from x13; append from x13; quit;
Proc iml;
use xd14; read all into xd14;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x14=(1/sqrt(c))*t(f)*P*xd14;
create x14 from x14; append from x14; quit;
Proc iml;
use xd15; read all into xd15;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x15=(1/sqrt(c))*t(f)*P*xd15;
create x15 from x15; append from x15; quit;
Proc iml;
use xd16; read all into xd16;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x16=(1/sqrt(c))*t(f)*P*xd16;
create x16 from x16; append from x16; quit;
Proc iml;
use xd17; read all into xd17;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x17=(1/sqrt(c))*t(f)*P*xd17;
create x17 from x17; append from x17; quit;
Proc iml;
use xd18; read all into xd18;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x18=(1/sqrt(c))*t(f)*P*xd18;
create x18 from x18; append from x18; quit;
Proc iml;
use xd19; read all into xd19;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x19=(1/sqrt(c))*t(f)*P*xd19;
create x19 from x19; append from x19; quit;
Proc iml;
use xd20; read all into xd20;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x20=(1/sqrt(c))*t(f)*P*xd20;
create x20 from x20; append from x20; quit;
Proc iml;
use xd21; read all into xd21;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x21=(1/sqrt(c))*t(f)*P*xd21;
create x21 from x21; append from x21; quit;
Proc iml;
use xd22; read all into xd22;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x22=(1/sqrt(c))*t(f)*P*xd22;
create x22 from x22; append from x22; quit;
Proc iml;
use xd23; read all into xd23;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x23=(1/sqrt(c))*t(f)*P*xd23;
create x23 from x23; append from x23; quit;
Proc iml;
use xd24; read all into xd24;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x24=(1/sqrt(c))*t(f)*P*xd24;
create x24 from x24; append from x24; quit;
Proc iml;
use xd25; read all into xd25;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x25=(1/sqrt(c))*t(f)*P*xd25;
create x25 from x25; append from x25; quit;
Proc iml;
use xd26; read all into xd26;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x26=(1/sqrt(c))*t(f)*P*xd26;
create x26 from x26; append from x26; quit;
```

```
Proc iml;
use xd27; read all into xd27;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x27=(1/sqrt(c))*t(f)*P*xd27;
create x27 from x27; append from x27; quit;

Data H2;
set x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13
x14 x15 x16 x17 x18 x19 x20 x21 x22 x23 x24
x25 x26 x27;
run;

/** Regressão de z2(Y) sobre H2(X) **/

Data z2;
set z2;
rename col1=col0; run;

Data reg2;
merge z2 H2; run;

proc reg data=reg2 noprint;
        model col0=col1 col2 / noint;
        output out=rg2  residual=r2;
run; quit;

Data rg2;
set rg2;
qr2=r2*r2;
Keep qr2; run;

Proc means data=rg2 sum noprint;
var qr2;
output out=rg2_ sum=sqr2;
run;

Data rg2_;
set rg2_;
keep sqr2;
run;

/** Cálculo da variância sigma2_v **/

Proc iml;
use H2; read all into H2;
Ident=I(27);
PH2=Ident - H2*ginv(t(H2)*H2)*t(H2);    /*
Matriz P_H2 */
create PH2 from PH2; append from PH2; quit;

Proc iml;
use f; read all into f;
use P1; read all into P1;
use P2; read all into P2;
use P3; read all into P3;
use P4; read all into P4;
use P5; read all into P5;
use P6; read all into P6;
use P7; read all into P7;
use P8; read all into P8;
use P9; read all into P9;
use P10; read all into P10;
use P11; read all into P11;
use P12; read all into P12;
use P13; read all into P13;
use P14; read all into P14;
use P15; read all into P15;
use P16; read all into P16;
use P17; read all into P17;
use P18; read all into P18;
use P19; read all into P19;
use P20; read all into P20;
use P21; read all into P21;
use P22; read all into P22;
use P23; read all into P23;
use P24; read all into P24;
use P25; read all into P25;
use P26; read all into P26;
use P27; read all into P27;
use c; read all into c;
cfp1=t(f)*P1*f*(1 / c);
cfp2=t(f)*P2*f*(1 / c);
cfp3=t(f)*P3*f*(1 / c);
cfp4=t(f)*P4*f*(1 / c);
cfp5=t(f)*P5*f*(1 / c);
cfp6=t(f)*P6*f*(1 / c);
```

```
cfp7=t(f)*P7*f*(1 / c);
cfp8=t(f)*P8*f*(1 / c);
cfp9=t(f)*P9*f*(1 / c);
cfp10=t(f)*P10*f*(1 / c);
cfp11=t(f)*P11*f*(1 / c);
cfp12=t(f)*P12*f*(1 / c);
cfp13=t(f)*P13*f*(1 / c);
cfp14=t(f)*P14*f*(1 / c);
cfp15=t(f)*P15*f*(1 / c);
cfp16=t(f)*P16*f*(1 / c);
cfp17=t(f)*P17*f*(1 / c);
cfp18=t(f)*P18*f*(1 / c);
cfp19=t(f)*P19*f*(1 / c);
cfp20=t(f)*P20*f*(1 / c);
cfp21=t(f)*P21*f*(1 / c);
cfp22=t(f)*P22*f*(1 / c);
cfp23=t(f)*P23*f*(1 / c);
cfp24=t(f)*P24*f*(1 / c);
cfp25=t(f)*P25*f*(1 / c);
cfp26=t(f)*P26*f*(1 / c);
cfp27=t(f)*P27*f*(1 / c);
cfp_=cfp1//cfp2//cfp3//cfp4//cfp5//cfp6//cfp7/
/cfp8//cfp9//cfp10//cfp11//cfp12//cfp13//cfp14
//cfp15//cfp16//cfp17//cfp18//cfp19//cfp20//cf
p21//cfp22//cfp23//cfp24//cfp25//cfp26//cfp27;
cfp=diag(cfp_);
create cfp from cfp; append from cfp; quit;

/* *** Fim *** */

Proc iml;
use H2; read all into H2;
FF=H2*t(H2);
eval1=eigval(FF);
create eval1 from eval1; append from eval1;
quit;

Data id1;
do id=1 to 27;
output; end;
run;

Data eval1;
merge id1 eval1;
run;

Data new1;
set eval1;
by id;
retain count 0;
if first.id then count=0;
if col1 > 0.000001 then count = count + 1;
run;

Proc means data=new1 sum noprint;
var count;
output out=caracH2 sum=caracH2;
run;

Data caracH2;
set caracH2;
keep caracH2;  /* Caracteristica de H2 */
run;

Proc iml;
use PH2; read all into PH2;
use cfp; read all into cfp;
use c; read all into c;
use rg2_; read all into rg2_;
use caracH2; read all into caracH2;
use varu; read all into varu;
use Binv_&l; read all into Binv_&l;
varv= (rg2_ - trace(PH2*cfp) - varu*(27 -
caracH2))/(c*trace(PH2*Binv_&l));
create varv from varv; append from varv; quit;

/* Termina aqui o cálculo da variância
sigma2_v */
/* Truncagem a zero das variâncias sigma2_u e
sigma2_v */

Data varu;
set varu;
sigma2u= max( col1 , 0);
keep sigma2u;
run;
Data varv;
set varv;
```

```
sigma2v= max( col1 , 0);
keep sigma2v;
run;

Proc iml;
use varu; read all into varu;
use varv; read all into varv;
paruv_=varu//varv;
create paruv_ from paruv_; append from paruv_;
quit;

Proc iml;
use Paruv; read all into Paruv;
use Paruv_; read all into Paruv_;
use Wu; read all into Wu;
tBwj_&l=Wu * t(paruv_ - paruv);   /* Aqui
obtem-se a transposta de Cwj_t - enviesamento
*/
create tBwj_&l from tBwj_&l; append from
tBwj_&l; quit;

/* tBwj_&l tem dimensão 1*2 */

Proc iml;
use Paruv; read all into Paruv;
use Paruv_; read all into Paruv_;
use Wu; read all into Wu;
Uwj_&l=Wu * ((paruv_ - paruv)*t(paruv_ -
paruv));   /* Aqui obtem-se o Uwj_t -
covariância */
create Uwj_&l from Uwj_&l; append from Uwj_&l;
quit;

/* Uwj_&l tem dimensão 2*2 */

%mend;
%inc 'C:\Documents and Settings\Luis
Pereira\My
Documents\Doutoramento\chamamacro.txt';

/* *** Fim *** */

/* Cálculo do factor tCwj_t - neste caso é
constante porque os pesos são constantes */

Proc iml;
use tBwj_1; read all into tBwj_1;
use tBwj_2; read all into tBwj_2;
use tBwj_3; read all into tBwj_3;
use tBwj_4; read all into tBwj_4;
use tBwj_5; read all into tBwj_5;
use tBwj_6; read all into tBwj_6;
use tBwj_7; read all into tBwj_7;
use tBwj_8; read all into tBwj_8;
use tBwj_9; read all into tBwj_9;
use tBwj_10; read all into tBwj_10;
use tBwj_11; read all into tBwj_11;
use tBwj_12; read all into tBwj_12;
use tBwj_13; read all into tBwj_13;
use tBwj_14; read all into tBwj_14;
use tBwj_15; read all into tBwj_15;
use tBwj_16; read all into tBwj_16;
use tBwj_17; read all into tBwj_17;
use tBwj_18; read all into tBwj_18;
use tBwj_19; read all into tBwj_19;
use tBwj_20; read all into tBwj_20;
use tBwj_21; read all into tBwj_21;
use tBwj_22; read all into tBwj_22;
use tBwj_23; read all into tBwj_23;
use tBwj_24; read all into tBwj_24;
use tBwj_25; read all into tBwj_25;
use tBwj_26; read all into tBwj_26;
use tBwj_27; read all into tBwj_27;
use tBwj_28; read all into tBwj_28;
tBwj= tBwj_1 + tBwj_2 + tBwj_3 + tBwj_4 +
tBwj_5 + tBwj_6 + tBwj_7 + tBwj_8 + tBwj_9 +
tBwj_10 + tBwj_11 + tBwj_12 + tBwj_13 +
tBwj_14 + tBwj_15 + tBwj_16 + tBwj_17 +
tBwj_18 + tBwj_19 + tBwj_20 + tBwj_21 +
tBwj_22 + tBwj_23 + tBwj_24 + tBwj_25 +
tBwj_26 + tBwj_27 + tBwj_28;
create tBwj from tBwj; append from tBwj; quit;
/* Matriz 1*2 */

/* Cálculo do factor Uwj_t - neste caso é
constante porque os pesos são constantes */

Proc iml;
use Uwj_1; read all into Uwj_1;
use Uwj_2; read all into Uwj_2;
use Uwj_3; read all into Uwj_3;
use Uwj_4; read all into Uwj_4;
use Uwj_5; read all into Uwj_5;
use Uwj_6; read all into Uwj_6;
use Uwj_7; read all into Uwj_7;
use Uwj_8; read all into Uwj_8;
use Uwj_9; read all into Uwj_9;
use Uwj_10; read all into Uwj_10;
use Uwj_11; read all into Uwj_11;
use Uwj_12; read all into Uwj_12;
use Uwj_13; read all into Uwj_13;
use Uwj_14; read all into Uwj_14;
use Uwj_15; read all into Uwj_15;
use Uwj_16; read all into Uwj_16;
use Uwj_17; read all into Uwj_17;
use Uwj_18; read all into Uwj_18;
use Uwj_19; read all into Uwj_19;
use Uwj_20; read all into Uwj_20;
use Uwj_21; read all into Uwj_21;
use Uwj_22; read all into Uwj_22;
use Uwj_23; read all into Uwj_23;
use Uwj_24; read all into Uwj_24;
use Uwj_25; read all into Uwj_25;
use Uwj_26; read all into Uwj_26;
use Uwj_27; read all into Uwj_27;
use Uwj_28; read all into Uwj_28;
Uwj= Uwj_1 + Uwj_2 + Uwj_3 + Uwj_4 + Uwj_5 +
Uwj_6 + Uwj_7 + Uwj_8 + Uwj_9 + Uwj_10 +
Uwj_11 + Uwj_12 + Uwj_13 + Uwj_14 + Uwj_15 +
Uwj_16 + Uwj_17 + Uwj_18 + Uwj_19 + Uwj_20 +
Uwj_21 + Uwj_22 + Uwj_23 + Uwj_24 + Uwj_25 +
Uwj_26 + Uwj_27 + Uwj_28;
create Uwj from Uwj; append from Uwj; quit;
/* Matriz 2*2 */

/*Cálculo da parcela g3_aux1= tCwj_t * dg1_it
*/

Data null;
file 'C:\Documents and Settings\Luis
Pereira\My
Documents\Doutoramento\chamamacro.txt';
   do i= 1 to 196; k=l;
var='%ext_am('||i||','||k||');'; put var; end;
run;

%macro ext_am(i,k);

Proc iml;
use tBwj; read all into tBwj;
use Dg&i; read all into Dg&i;
CWJ&i=tBwj*Dg&i;
create CWJ&i from CWJ&i; append from CWJ&i;
quit;

Proc append base=g3_aux1 data=CWJ&i;
run;

%mend;
%inc 'C:\Documents and Settings\Luis
Pereira\My
Documents\Doutoramento\chamamacro.txt';

/* Cálculo da parcela g3_aux2 = tr (A_it *
Uwj_t) */

Data null;
file 'C:\Documents and Settings\Luis
Pereira\My
Documents\Doutoramento\chamamacro.txt';
   do i= 1 to 196; k=l;
var='%ext_am('||i||','||k||');'; put var; end;
run;

%macro ext_am(i,k);

Proc iml;
use Uwj; read all into Uwj;
use A_&i; read all into A_&i;
tr_A&i=trace(A_&i*Uwj);
create tr_A&i from tr_A&i; append from tr_A&i;
quit;

Proc append base=g3_aux2 data=tr_A&i;
run;
```

```
%mend;
%inc 'C:\Documents and Settings\Luis
Pereira\My
Documents\Doutoramento\chamamacro.txt';

/* Cálculo da parcela g3_aux3 = tr (K_it *
Uwj_t) */

Data null;
file 'C:\Documents and Settings\Luis
Pereira\My
Documents\Doutoramento\chamamacro.txt';
   do i= 1 to 196; k=l;
var='%ext_am('||i||','||k||');'; put var; end;
run;

%macro ext_am(i,k);

Proc iml;
use Uwj; read all into Uwj;
use K&i; read all into K&i;
tr_K&i=trace(K&i*Uwj);
create tr_K&i from tr_K&i; append from tr_K&i;
quit;

Proc append base=g3_aux3 data=tr_K&i;
run;

%mend;
%inc 'C:\Documents and Settings\Luis
Pereira\My
Documents\Doutoramento\chamamacro.txt';

Data indice;
do i=1 to 28;
        do t=1 to 7;
                output;
        end;
end; run;

Proc iml;
use g3_aux1; read all into g3_aux1;
use g3_aux2; read all into g3_aux2;
use g3_aux3; read all into g3_aux3;
use indice; read all into indice;
g3_aux=indice||g3_aux1||g3_aux2||g3_aux3;
create g3_aux from g3_aux; append from g3_aux;
quit;

Data g3_jack;
set g3_aux;
g3_jack=-col3 + col4 + col5;
rename col1=i col2=t;
simul=1;   /*SSS*/
run;

Proc append base=PC.Est_PC_eqm_mod111_jack
data=g3_jack;
run;

DM 'CLEAR OUTPUT';
DM 'CLEAR LOG';
```

Apêndice 25 – Divisão de Portugal em NUTS

A25.1 Divisão de Portugal em NUTSI, NUTSII e NUTSIII, de acordo com o Decreto-lei n.º 244/2002

NUTSI	NUTSII	NUTSIII
Portugal continental	Norte	Minho-Lima
		Cávado
		Ave
		Grande Porto
		Tâmega
		Entre Douro e Vouga
		Douro
		Alto Trás-os-Montes
	Centro	Baixo Vouga
		Baixo Mondego
		Pinhal Litoral
		Pinhal Interior Norte
		Dão-Lafões
		Pinhal Interior Sul
		Serra da Estrela
		Beira Interior Norte
		Beira Interior Sul
		Cova da Beira
		Oeste
		Médio Tejo
	Lisboa	Grande Lisboa
		Península de Setúbal
	Alentejo	Lezíria do Tejo
		Alentejo Litoral
		Alto Alentejo
		Alentejo Central
		Baixo Alentejo
	Algarve	Algarve
Região Autónoma dos Açores	Região Autónoma dos Açores	Região Autónoma dos Açores
Região Autónoma da Madeira	Região Autónoma da Madeira	Região Autónoma da Madeira

Apêndice 26 – Estimativas directas do preço médio de transacção da habitação ao nível de Portugal continental e de NUTSII

A26.1 Dimensões amostrais de empresas de mediação imobiliária (a_i) e de transacções (n_i), estimativas directas do preço médio de transacção da habitação ($\hat{\mu}_i^{dir}$) e respectivos EPR $\left[epr\left(\hat{\mu}_i^{dir}\right)\right]$, ao nível de Portugal continental e de NUTSII

Regiões	a_i	n_i	$\hat{\mu}_i^{dir}$	$epr\left(\hat{\mu}_i^{dir}\right)$	a_i	n_i	$\hat{\mu}_i^{dir}$	$epr\left(\hat{\mu}_i^{dir}\right)$
NUTSII		***t*=1**				***t*=2**		
Norte	127	706	959	4,4%	127	690	925	2,1%
Centro	110	561	837	3,1%	111	430	832	2,9%
Lisboa	184	1.031	1.187	3,6%	181	907	1.244	4,8%
Alentejo	32	136	881	5,4%	32	119	880	5,4%
Algarve	32	78	958	5,2%	35	110	1.043	5,1%
Portugal continental	**458**	**2.512**	**1.041**	**2,3%**	**458**	**2.256**	**1.062**	**3,2%**
NUTSII		***t*=3**				***t*=4**		
Norte	125	585	904	2,7%	126	453	940	3,0%
Centro	109	448	804	3,1%	108	360	844	3,1%
Lisboa	182	876	1.196	3,3%	182	785	1.234	3,9%
Alentejo	36	182	987	17,2%	33	107	1.219	20,3%
Algarve	35	111	1.165	8,5%	37	94	1.028	4,4%
Portugal continental	**458**	**2.202**	**1.039**	**2,6%**	**458**	**1.799**	**1.091**	**3,1%**
NUTSII		***t*=5**				***t*=6**		
Norte	126	420	955	2,9%	125	464	956	2,4%
Centro	109	330	832	3,5%	106	317	854	2,8%
Lisboa	175	746	1.252	3,7%	179	764	1.295	3,7%
Alentejo	33	92	1.038	16,9%	31	111	877	4,4%
Algarve	36	77	1.086	5,5%	37	88	1.215	7,7%
Portugal continental	**458**	**1.665**	**1.093**	**2,7%**	**458**	**1.744**	**1.129**	**2,7%**

Apêndice 27 – Estimativas directas do preço médio de transacção da habitação ao nível das NUTSIII do continente

A27.1 Dimensões amostrais de empresas de mediação imobiliária (a_i) e de transacções (n_i), estimativas directas do preço médio de transacção da habitação ($\hat{\mu}_i^{dir}$) e respectivos EPR $\left[epr\left(\hat{\mu}_i^{dir}\right)\right]$, ao nível das NUTSIII do continente

NUTSIII	a_i	n_i	$\hat{\mu}_i^{dir}$	$epr(\hat{\mu}_i^{dir})$	a_i	n_i	$\hat{\mu}_i^{dir}$	$epr(\hat{\mu}_i^{dir})$
	***t*=1**				***t*=2**			
111	10	24	761	9,5%	9	21	849	10,7%
112	12	54	705	5,2%	11	43	821	3,1%
113	13	67	733	1,9%	11	43	712	3,9%
114	78	488	1042	3,2%	78	516	970	2,3%
115	5	22	582	6,5%	5	25	667	7,3%
116	12	33	767	5,7%	11	24	731	5,3%
117	3	8	712	10,9%	2	8	680	3,0%
118	3	10	692	9,3%	4	10	796	6,9%
161	16	88	856	4,4%	16	59	890	6,5%
162	24	144	979	7,3%	25	123	936	4,7%
163	14	91	737	6,7%	14	66	711	2,5%
164	8	37	708	5,3%	7	22	633	5,2%
165	6	29	771	11,2%	6	19	706	8,2%
166	1	1	377	-	1	1	377	-
167	1	1	500	-	1	1	500	-
168	2	4	464	7,5%	3	4	515	19,2%
169	6	10	837	7,2%	5	5	688	14,3%
16A	5	17	558	18,3%	5	7	488	16,3%
16B	29	96	888	5,9%	29	83	907	5,3%
171	116	607	1264	4,4%	113	539	1318	4,9%
172	76	424	1023	3,7%	72	368	1084	5,8%
16C	7	43	612	5,1%	7	40	686	4,0%
185	10	51	790	8,4%	9	42	789	9,4%
181	5	17	1053	13,9%	5	12	1112	13,6%
182	5	25	808	12,2%	4	10	788	14,1%
183	7	27	1069	9,0%	9	36	1019	7,6%
184	5	16	876	4,3%	5	19	792	3,8%
150	32	78	958	5,2%	35	110	1043	5,1%

NUTSIII	a_i	n_i	$\hat{\mu}_i^{dir}$	$epr(\hat{\mu}_i^{dir})$	a_i	n_i	$\hat{\mu}_i^{dir}$	$epr(\hat{\mu}_i^{dir})$
	t=3				*t*=4			
111	9	23	864	12,1%	9	21	825	12,0%
112	11	32	674	7,3%	10	15	681	5,5%
113	12	57	733	2,3%	12	23	689	5,1%
114	80	394	987	2,0%	80	323	1021	2,4%
115	5	29	644	5,8%	4	19	630	7,6%
116	8	26	731	2,6%	11	32	673	5,6%
117	2	14	626	2,2%	1	7	696	-
118	3	10	765	2,0%	3	13	686	3,6%
161	14	56	827	6,9%	15	48	867	6,7%
162	26	90	881	6,5%	24	84	964	4,6%
163	13	50	706	4,4%	13	41	684	6,3%
164	6	44	710	4,8%	7	33	724	2,3%
165	6	23	642	4,1%	7	23	723	10,8%
166	2	2	698	7,1%	1	1	377	-
167	1	1	500	-	1	1	500	-
168	2	4	708	0,5%	2	4	661	12,0%
169	5	10	741	14,7%	5	6	751	13,6%
16A	5	25	554	6,3%	5	11	538	13,3%
16B	29	90	932	6,8%	26	77	934	6,3%
171	114	530	1282	3,8%	116	508	1305	4,5%
172	72	346	1003	4,1%	75	277	1032	5,1%
16C	7	53	687	4,8%	7	31	726	7,9%
185	12	66	707	9,5%	11	32	759	9,0%
181	6	27	1621	15,5%	7	29	1641	14,7%
182	6	16	706	10,8%	5	11	742	18,1%
183	8	42	945	6,1%	7	21	1056	5,4%
184	5	31	916	3,7%	5	14	858	4,7%
150	35	111	1165	8,5%	37	94	1028	4,4%
NUTSIII	*t*=5				*t*=6			
111	9	18	755	7,5%	10	18	815	7,1%
112	10	25	806	10,7%	11	32	803	6,5%
113	12	24	707	4,7%	12	23	740	5,5%
114	80	299	1031	2,5%	79	344	1012	2,1%
115	5	12	643	3,6%	4	13	649	2,2%
116	11	24	693	4,4%	9	22	712	3,8%
117	1	7	652	-	2	4	607	0,9%
118	3	11	661	2,9%	3	8	781	4,4%
161	14	52	784	5,5%	14	52	832	5,3%
162	25	74	900	5,5%	21	75	998	6,1%
163	14	31	758	6,1%	14	45	739	5,8%
164	8	17	637	8,7%	8	20	570	6,4%
165	8	35	734	6,6%	9	21	745	13,3%
166	2	2	408	1,5%	1	1	377	-
167	1	1	500	-	1	1	500	-
168	2	3	710	8,2%	3	4	749	5,6%
169	5	5	796	17,6%	5	6	793	15,5%
16A	5	10	466	9,7%	5	7	454	19,5%
16B	27	78	947	6,4%	23	56	1019	4,2%
171	110	514	1322	3,5%	111	479	1382	3,8%
172	72	232	1036	6,8%	73	285	1063	5,3%
16C	7	22	739	7,6%	7	29	705	3,9%
185	10	29	737	7,2%	10	37	787	5,5%
181	6	20	1422	14,3%	6	17	1026	12,0%
182	6	12	788	3,3%	5	11	750	8,8%
183	8	20	799	9,8%	7	33	933	7,3%
184	5	11	926	5,3%	5	13	958	5,8%
150	36	77	1086	5,5%	37	88	1215	7,7%

Apêndice 28 – Estimativas tradicionais do preço médio de transacção da habitação ao nível das NUTSIII do continente

A28.1 Estimativas do preço médio de transacção da habitação ao nível das NUTSIII do continente (valores em euros/m^2), produzidas através dos estimadores tradicionais

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$epr(\hat{\mu}_i^{dir})$	$epr(\hat{\mu}_i^{sinQ})$	$epr(\hat{\mu}_i^{sinR})$	$epr(\hat{\mu}_i^{com})$
					***t*=1**			
111	761	878	706	706	9,5%	4,4%	3,8%	3,8%
112	705	805	637	684	5,2%	4,4%	5,1%	4,0%
113	733	850	680	729	1,9%	4,4%	4,2%	1,8%
114	1042	1129	940	1031	3,2%	4,4%	3,7%	2,9%
115	582	842	672	599	6,5%	4,4%	4,4%	5,3%
116	767	900	726	726	5,7%	4,4%	3,5%	3,5%
117	712	793	626	642	10,9%	4,4%	5,3%	4,8%
118	692	763	598	648	9,3%	4,4%	6,1%	5,9%
161	856	808	787	835	4,4%	3,1%	3,1%	3,3%
162	979	990	990	990	7,3%	3,1%	4,0%	4,0%
163	737	798	776	776	6,7%	3,1%	3,2%	3,2%
164	708	678	641	686	5,3%	3,1%	5,0%	4,0%
165	771	723	691	691	11,2%	3,1%	4,0%	4,0%
166	377	707	674	377	-	3,1%	4,3%	-
167	500	692	657	500	-	3,1%	4,7%	-
168	464	594	548	478	7,5%	3,1%	7,7%	6,2%
169	837	722	690	813	7,2%	3,1%	4,0%	6,2%
16A	558	716	684	640	18,3%	3,1%	4,1%	6,3%
16B	888	700	960	926	5,9%	3,1%	3,8%	3,4%
171	1264	1254	1243	1243	4,4%	3,6%	5,9%	5,9%
172	1023	1085	1060	1060	3,7%	3,6%	4,6%	4,6%
16C	612	574	766	619	5,1%	3,1%	3,2%	4,9%
185	790	678	874	842	8,4%	5,4%	3,2%	3,6%
181	1053	979	937	937	13,9%	5,4%	3,6%	3,6%
182	808	831	778	778	12,2%	5,4%	3,2%	3,2%
183	1069	921	874	1021	9,0%	5,4%	3,2%	7,2%
184	876	743	683	868	4,3%	5,4%	4,2%	4,2%
150	958	958	1036	990	5,2%	5,2%	4,4%	3,5%

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$epr(\hat{\mu}_i^{dir})$	$epr(\hat{\mu}_i^{sinQ})$	$epr(\hat{\mu}_i^{sinR})$	$epr(\hat{\mu}_i^{com})$
				***t*=2**				
111	849	882	733	778	10,7%	2,1%	3,2%	4,9%
112	821	788	636	818	3,1%	2,1%	4,7%	3,1%
113	712	819	669	694	3,9%	2,1%	4,1%	2,8%
114	970	1095	954	954	2,3%	2,1%	3,4%	3,4%
115	667	796	644	644	7,3%	2,1%	4,5%	4,5%
116	731	854	704	704	5,3%	2,1%	3,5%	3,5%
117	680	788	636	671	3,0%	2,1%	4,7%	2,5%
118	796	752	599	780	6,9%	2,1%	5,5%	6,5%
161	890	813	793	855	6,5%	2,9%	2,9%	4,4%
162	936	989	1005	964	4,7%	2,9%	3,7%	3,1%
163	711	825	809	715	2,5%	2,9%	2,8%	2,4%
164	633	698	656	656	5,2%	2,9%	4,3%	4,3%
165	706	741	708	708	8,2%	2,9%	3,5%	3,5%
166	377	651	600	377	-	2,9%	5,5%	-
167	500	695	653	500	-	2,9%	4,3%	-
168	515	603	542	542	19,2%	2,9%	7,2%	7,2%
169	688	766	737	737	14,3%	2,9%	3,2%	3,2%
16A	488	719	681	521	16,3%	2,9%	3,9%	12,7%
16B	907	661	929	929	5,3%	2,9%	3,2%	3,2%
171	1318	1319	1297	1297	4,9%	4,8%	5,5%	5,5%
172	1084	1126	1081	1081	5,8%	4,8%	4,2%	4,2%
16C	686	564	766	695	4,0%	2,9%	3,0%	3,5%
185	789	667	877	852	9,4%	5,4%	3,0%	3,3%
181	1112	1015	996	996	13,6%	5,4%	3,6%	3,6%
182	788	804	751	751	14,1%	5,4%	3,1%	3,1%
183	1019	888	849	985	7,6%	5,4%	2,9%	6,2%
184	792	772	715	781	3,8%	5,4%	3,4%	3,3%
150	1043	1043	1064	1064	5,1%	5,1%	4,1%	4,1%
NUTSIII				***t*=3**				
111	864	880	765	765	12,1%	2,7%	4,1%	4,1%
112	674	797	684	684	7,3%	2,7%	5,6%	5,6%
113	733	817	704	722	2,3%	2,7%	5,1%	2,3%
114	987	1076	957	973	2,0%	2,7%	4,7%	2,4%
115	644	765	652	652	5,8%	2,7%	6,4%	6,4%
116	731	841	727	727	2,6%	2,7%	4,7%	4,7%
117	626	767	655	632	2,2%	2,7%	6,3%	2,2%
118	765	749	637	763	2,0%	2,7%	6,9%	1,9%
161	827	785	799	799	6,9%	3,1%	3,9%	3,9%
162	881	953	991	911	6,5%	3,1%	5,0%	4,8%
163	706	773	785	718	4,4%	3,1%	4,0%	3,7%
164	710	673	672	679	4,8%	3,1%	5,9%	4,8%
165	642	706	709	653	4,1%	3,1%	5,0%	3,5%
166	698	619	610	670	7,1%	3,1%	7,8%	5,6%
167	500	648	643	500	-	3,1%	6,7%	-
168	708	664	661	708	0,5%	3,1%	6,1%	0,5%
169	741	729	735	735	14,7%	3,1%	4,5%	4,5%
16A	554	725	731	561	6,3%	3,1%	4,6%	5,9%
16B	932	647	937	937	6,8%	3,1%	4,5%	4,5%
171	1282	1260	1254	1254	3,8%	3,3%	7,7%	7,7%
172	1003	1092	1075	1027	4,1%	3,3%	5,9%	3,4%
16C	687	550	782	698	4,8%	3,1%	4,0%	4,2%
185	707	745	871	734	9,5%	17,2%	4,0%	7,7%
181	1621	1121	967	1524	15,5%	17,2%	4,8%	14,0%
182	706	943	798	769	10,8%	17,2%	3,9%	4,2%
183	945	983	837	915	6,1%	17,2%	3,9%	4,6%
184	916	873	732	909	3,7%	17,2%	4,6%	3,6%
150	1165	1165	1104	1104	8,5%	8,5%	6,2%	6,2%

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$epr(\hat{\mu}_i^{dir})$	$epr(\hat{\mu}_i^{sinQ})$	$epr(\hat{\mu}_i^{sinR})$	$epr(\hat{\mu}_i^{com})$
				***t*=4**				
111	825	920	761	761	12,0%	3,0%	4,4%	4,4%
112	681	837	670	670	5,5%	3,0%	6,0%	6,0%
113	689	857	692	692	5,1%	3,0%	5,5%	5,5%
114	1021	1116	974	1009	2,4%	3,0%	4,8%	2,1%
115	630	801	632	632	7,6%	3,0%	7,0%	7,0%
116	673	884	721	702	5,6%	3,0%	4,9%	3,7%
117	696	803	633	696	-	3,0%	7,0%	-
118	686	761	588	680	3,6%	3,0%	8,5%	3,4%
161	867	808	781	828	6,7%	3,1%	4,3%	4,2%
162	964	1002	1026	996	4,6%	3,1%	5,3%	3,5%
163	684	801	771	706	6,3%	3,1%	4,3%	4,7%
164	724	700	644	721	2,3%	3,1%	6,7%	2,2%
165	723	736	690	690	10,8%	3,1%	5,5%	5,5%
166	377	685	626	377	-	3,1%	7,2%	-
167	500	648	578	500	-	3,1%	8,9%	-
168	661	673	610	610	12,0%	3,1%	7,7%	7,7%
169	751	761	721	721	13,6%	3,1%	4,9%	4,9%
16A	538	752	709	569	13,3%	3,1%	5,1%	10,5%
16B	934	695	963	963	6,3%	3,1%	4,7%	4,7%
171	1305	1307	1307	1307	4,5%	3,9%	7,6%	7,6%
172	1032	1121	1087	1083	5,1%	3,9%	5,8%	5,5%
16C	726	576	758	758	7,9%	3,1%	4,5%	4,5%
185	759	927	872	800	9,0%	20,3%	4,1%	5,7%
181	1641	1324	951	1557	14,7%	20,3%	4,6%	13,6%
182	742	1191	831	831	18,1%	20,3%	4,1%	4,1%
183	1056	1214	852	1040	5,4%	20,3%	4,1%	5,1%
184	858	1119	767	840	4,7%	20,3%	4,4%	3,9%
150	1028	1028	1121	1051	4,4%	4,4%	6,1%	3,7%
NUTSIII				***t*=5**				
111	755	933	746	746	7,5%	2,9%	3,6%	3,6%
112	806	843	644	760	10,7%	2,9%	5,2%	8,2%
113	707	859	663	683	4,7%	2,9%	4,8%	3,3%
114	1031	1144	986	1017	2,5%	2,9%	4,1%	2,1%
115	643	821	619	621	3,6%	2,9%	5,8%	5,4%
116	693	900	709	709	4,4%	2,9%	4,0%	4,0%
117	652	803	599	652	-	2,9%	6,3%	-
118	661	780	573	656	2,9%	2,9%	7,1%	2,8%
161	784	816	811	811	5,5%	3,5%	3,3%	3,3%
162	900	981	1038	917	5,5%	3,5%	4,5%	4,7%
163	758	797	785	785	6,1%	3,5%	3,4%	3,4%
164	637	695	644	644	8,7%	3,5%	5,2%	5,2%
165	734	747	716	716	6,6%	3,5%	3,9%	3,9%
166	408	652	585	408	1,5%	3,5%	6,7%	1,5%
167	500	691	639	500	-	3,5%	5,3%	-
168	710	670	610	676	8,2%	3,5%	6,0%	6,0%
169	796	758	731	731	17,6%	3,5%	3,7%	3,7%
16A	466	728	690	476	9,7%	3,5%	4,3%	9,1%
16B	947	703	950	950	6,4%	3,5%	3,9%	3,9%
171	1322	1287	1224	1300	3,5%	3,7%	5,8%	3,1%
172	1036	1179	1095	1095	6,8%	3,7%	4,9%	4,9%
16C	739	622	806	787	7,6%	3,5%	3,3%	3,2%
185	737	783	815	773	7,2%	16,9%	3,3%	4,0%
181	1422	1202	986	1327	14,3%	16,9%	4,1%	12,0%
182	788	962	727	776	3,3%	16,9%	3,8%	2,8%
183	799	1045	817	817	9,8%	16,9%	3,3%	3,3%
184	926	969	735	913	5,3%	16,9%	3,7%	5,1%
150	1086	1086	1143	1143	5,5%	5,5%	5,3%	5,3%

NUTSIII	$\hat{\mu}_i^{dir}$	$\hat{\mu}_i^{sinQ}$	$\hat{\mu}_i^{sinR}$	$\hat{\mu}_i^{com}$	$epr(\hat{\mu}_i^{dir})$	$epr(\hat{\mu}_i^{sinQ})$	$epr(\hat{\mu}_i^{sinR})$	$epr(\hat{\mu}_i^{com})$
				***t*=6**				
111	815	960	782	782	7,1%	2,4%	3,3%	3,3%
112	803	839	658	784	6,5%	2,4%	4,9%	5,8%
113	740	862	682	711	5,5%	2,4%	4,4%	3,6%
114	1012	1157	983	996	2,1%	2,4%	3,6%	2,2%
115	649	807	626	640	2,2%	2,4%	5,6%	2,5%
116	712	911	733	733	3,8%	2,4%	3,7%	3,7%
117	607	803	622	610	0,9%	2,4%	5,7%	1,2%
118	781	778	597	775	4,4%	2,4%	6,3%	4,3%
161	832	806	777	796	5,3%	2,8%	3,3%	2,8%
162	998	991	1001	1001	6,1%	2,8%	3,8%	3,8%
163	739	884	871	753	5,8%	2,8%	3,1%	5,1%
164	570	701	650	586	6,4%	2,8%	5,0%	5,1%
165	745	770	733	733	13,3%	2,8%	3,7%	3,7%
166	377	708	658	377	-	2,8%	4,9%	-
167	500	744	702	500	-	2,8%	4,1%	-
168	749	657	597	737	5,6%	2,8%	6,4%	5,3%
169	793	729	685	685	15,5%	2,8%	4,4%	4,4%
16A	454	735	691	487	19,5%	2,8%	4,3%	15,6%
16B	1019	654	935	997	4,2%	2,8%	3,4%	3,3%
171	1382	1370	1368	1368	3,8%	3,7%	6,1%	6,1%
172	1063	1140	1105	1105	5,3%	3,7%	4,5%	4,5%
16C	705	601	844	711	3,9%	2,8%	3,1%	3,7%
185	787	613	837	824	5,5%	4,4%	3,1%	2,7%
181	1026	1091	1066	1066	12,0%	4,4%	4,2%	4,2%
182	750	794	722	722	8,8%	4,4%	3,8%	3,8%
183	933	868	808	896	7,3%	4,4%	3,1%	5,4%
184	958	850	787	940	5,8%	4,4%	3,2%	5,3%
150	1215	1215	1108	1133	7,7%	7,7%	4,5%	3,9%

Apêndice 29 – Estimativas EBLUP do preço médio de transacção da habitação ao nível das NUTSIII do continente

A29.1 Estimativas do preço médio de transacção da habitação ao nível das NUTSIII do continente (valores em euros/m^2), produzidas através dos estimadores EBLUP

NUTSIII	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$	$epr(\hat{\mu}_i^{FH})$	$epr(\hat{\mu}_i^{NS})$	$epr(\hat{\mu}_{it}^{RY})$	$epr(\hat{\mu}_{it}^{LP})$
				***t*=1**				
111	746	747	753	721	8,5%	8,5%	5,8%	5,2%
112	698	709	717	706	5,1%	5,0%	4,7%	4,7%
113	732	732	728	729	1,9%	1,9%	2,2%	2,2%
114	1035	1031	1007	993	3,1%	3,1%	3,3%	3,3%
115	591	610	619	652	6,2%	5,9%	5,9%	5,4%
116	762	767	726	743	5,5%	5,3%	5,1%	4,8%
117	686	668	645	647	9,8%	9,5%	5,7%	5,4%
118	670	668	713	670	8,7%	8,3%	5,7%	5,6%
161	849	857	837	817	4,3%	4,2%	4,1%	4,2%
162	982	999	932	931	6,5%	6,2%	4,2%	3,9%
163	743	739	706	707	6,2%	6,1%	5,1%	4,9%
164	701	672	667	609	5,1%	5,3%	5,3%	5,6%
165	742	725	710	669	9,7%	9,2%	5,7%	5,5%
166	387	392	390	425	5,7%	5,6%	6,6%	6,0%
167	506	508	499	519	4,4%	4,3%	5,1%	5,0%
168	471	474	553	495	7,2%	7,1%	6,5%	7,0%
169	806	751	769	649	6,8%	7,1%	5,5%	5,8%
16A	613	563	527	553	13,1%	12,8%	7,8%	6,6%
16B	900	898	922	914	5,4%	5,3%	4,1%	4,0%
171	1260	1234	1274	1218	4,2%	4,2%	3,1%	3,1%
172	1026	1035	1021	1038	3,6%	3,5%	3,4%	3,3%
16C	623	617	645	640	4,9%	4,9%	4,9%	4,9%
185	810	815	773	816	7,3%	7,1%	5,4%	4,7%
181	982	1020	1096	972	9,8%	8,2%	5,5%	3,8%
182	796	752	774	712	9,8%	9,5%	5,9%	5,2%
183	990	1005	975	916	7,8%	7,2%	4,4%	4,1%
184	857	861	847	793	4,3%	4,3%	4,0%	4,3%
150	970	985	1006	1011	4,9%	4,8%	3,8%	3,6%

NUTSIII	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$	$epr(\hat{\mu}_i^{FH})$	$epr(\hat{\mu}_i^{NS})$	$epr(\hat{\mu}_{it}^{RY})$	$epr(\hat{\mu}_{it}^{LP})$
				***t*=2**				
111	800	794	779	757	9,0%	6,6%	5,5%	4,8%
112	811	791	768	762	3,1%	3,9%	3,4%	3,6%
113	709	735	716	717	3,8%	4,3%	4,0%	4,0%
114	969	981	981	975	2,2%	2,6%	2,5%	2,6%
115	663	704	635	665	6,9%	6,3%	5,7%	5,3%
116	728	737	718	727	5,1%	5,7%	4,6%	4,5%
117	678	678	667	670	2,9%	3,4%	3,5%	3,5%
118	754	729	742	714	6,7%	6,3%	4,9%	4,9%
161	868	829	845	826	6,0%	5,9%	4,3%	4,3%
162	946	948	931	932	4,4%	4,9%	3,8%	3,8%
163	714	722	715	719	2,4%	2,7%	2,7%	2,8%
164	635	567	661	622	5,0%	6,4%	4,8%	5,1%
165	706	688	711	686	7,4%	6,8%	5,2%	5,2%
166	386	407	383	405	5,7%	6,3%	6,6%	6,2%
167	506	522	503	521	4,4%	5,0%	4,9%	4,8%
168	528	466	607	545	14,6%	11,3%	5,6%	5,9%
169	711	602	784	674	10,6%	9,3%	5,5%	5,4%
16A	558	510	528	551	11,9%	10,1%	7,3%	6,5%
16B	911	900	910	894	4,9%	5,3%	3,9%	4,0%
171	1312	1264	1297	1252	4,6%	4,6%	2,9%	2,9%
172	1083	1100	1031	1052	5,3%	5,1%	3,4%	3,3%
16C	691	674	676	673	3,9%	4,7%	4,1%	4,3%
185	818	852	773	815	7,8%	6,3%	5,2%	4,6%
181	1034	1046	1129	1022	9,0%	5,1%	5,3%	3,6%
182	769	678	761	695	10,5%	7,8%	5,9%	5,2%
183	961	914	963	903	6,8%	6,2%	4,2%	4,1%
184	787	788	838	805	3,7%	4,5%	3,4%	3,7%
150	1047	1059	1031	1039	4,7%	5,0%	3,6%	3,5%
NUTSIII				***t*=3**				
111	833	833	785	767	10,7%	10,6%	5,3%	4,7%
112	675	690	722	717	7,0%	6,8%	4,7%	4,7%
113	732	733	727	729	2,3%	2,3%	2,7%	2,7%
114	986	985	988	982	2,0%	2,0%	2,3%	2,3%
115	644	647	629	650	5,7%	5,6%	5,5%	5,2%
116	731	732	720	725	2,6%	2,6%	2,9%	3,0%
117	626	627	632	632	2,1%	2,1%	2,5%	2,5%
118	763	763	754	747	2,0%	2,0%	2,3%	2,3%
161	824	820	829	806	6,6%	6,5%	4,4%	4,5%
162	894	898	915	911	6,1%	6,0%	3,9%	3,9%
163	709	707	699	699	4,3%	4,3%	4,3%	4,3%
164	708	695	685	647	4,7%	4,8%	4,6%	4,8%
165	644	643	674	655	4,0%	4,0%	3,9%	4,4%
166	690	679	437	468	7,0%	7,0%	7,8%	7,1%
167	503	505	495	506	4,4%	4,4%	5,0%	5,0%
168	708	707	703	695	1,5%	1,5%	1,7%	1,8%
169	739	735	771	650	12,4%	11,6%	5,8%	5,6%
16A	563	570	547	569	6,1%	5,9%	5,8%	5,9%
16B	933	926	918	901	6,4%	6,3%	4,0%	4,0%
171	1279	1265	1283	1237	3,7%	3,7%	2,8%	2,9%
172	1008	1016	1019	1036	4,0%	3,9%	3,3%	3,3%
16C	691	691	681	677	4,7%	4,7%	4,6%	4,7%
185	733	730	755	791	8,6%	8,4%	5,2%	4,7%
181	1148	1168	1119	1001	12,1%	9,8%	5,4%	3,6%
182	724	729	770	718	9,6%	9,2%	5,6%	5,1%
183	932	932	946	886	5,9%	5,7%	4,0%	4,2%
184	907	915	876	838	3,7%	3,7%	3,5%	3,8%
150	1148	1185	1061	1083	7,6%	7,3%	3,5%	3,3%

NUTSIII	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$	$epr(\hat{\mu}_i^{FH})$	$epr(\hat{\mu}_i^{NS})$	$epr(\hat{\mu}_{it}^{RY})$	$epr(\hat{\mu}_{it}^{LP})$
				***t*=4**				
111	808	796	793	783	10,8%	11,0%	5,1%	4,5%
112	680	685	719	720	5,4%	5,3%	4,7%	4,7%
113	689	692	717	725	5,0%	5,0%	4,4%	4,4%
114	1020	1019	1014	1009	2,4%	2,4%	2,6%	2,6%
115	631	637	635	661	7,4%	7,2%	5,4%	5,1%
116	675	679	702	715	5,5%	5,4%	4,6%	4,5%
117	695	696	676	680	3,2%	3,2%	3,7%	3,7%
118	684	684	702	690	3,6%	3,5%	3,8%	3,9%
161	857	853	831	811	6,5%	6,4%	4,3%	4,4%
162	968	972	943	948	4,5%	4,5%	3,6%	3,7%
163	690	691	702	706	6,1%	6,0%	4,9%	4,8%
164	723	721	709	692	2,3%	2,3%	2,6%	2,8%
165	717	729	711	685	10,1%	9,6%	5,0%	5,0%
166	381	382	398	423	5,8%	5,8%	6,5%	6,1%
167	501	503	490	498	4,4%	4,4%	5,1%	5,1%
168	651	640	684	643	11,2%	11,0%	5,1%	5,2%
169	743	689	784	668	12,0%	12,5%	5,8%	5,5%
16A	566	561	545	576	11,8%	11,7%	6,5%	6,1%
16B	937	927	938	930	6,0%	6,0%	3,9%	3,9%
171	1306	1290	1301	1260	4,4%	4,4%	2,8%	2,9%
172	1037	1052	1026	1046	4,9%	4,8%	3,5%	3,4%
16C	729	713	689	682	7,5%	7,6%	5,2%	5,1%
185	775	784	769	808	8,2%	8,0%	5,0%	4,5%
181	1170	1164	1117	997	12,1%	10,7%	5,4%	3,6%
182	778	754	807	779	13,8%	13,4%	4,7%	4,3%
183	1034	1035	973	923	5,3%	5,2%	3,8%	4,0%
184	853	860	893	863	4,6%	4,5%	3,7%	3,9%
150	1035	1042	1055	1073	4,3%	4,3%	3,3%	3,4%
NUTSIII				***t*=5**				
111	753	760	788	781	7,0%	7,0%	4,8%	4,7%
112	756	769	740	740	9,8%	9,1%	4,9%	4,7%
113	705	711	719	726	4,6%	4,5%	4,5%	4,4%
114	1029	1029	1024	1021	2,4%	2,4%	2,6%	2,7%
115	643	646	644	660	3,6%	3,6%	3,9%	4,0%
116	694	696	704	716	4,3%	4,3%	4,2%	4,2%
117	651	653	648	651	3,4%	3,4%	3,9%	3,9%
118	659	660	685	677	2,9%	2,9%	3,3%	3,3%
161	787	785	835	828	5,3%	5,3%	4,1%	4,2%
162	918	920	943	953	5,1%	5,1%	3,7%	3,7%
163	761	753	729	739	5,8%	5,8%	4,9%	4,7%
164	638	595	672	632	8,1%	8,4%	5,1%	5,3%
165	732	724	746	731	6,3%	6,2%	5,0%	5,0%
166	409	409	407	409	1,5%	1,5%	1,6%	1,6%
167	504	505	507	525	4,4%	4,4%	4,9%	4,8%
168	693	671	702	665	7,9%	7,9%	5,4%	5,4%
169	761	714	804	694	13,0%	12,4%	5,7%	5,2%
16A	491	488	531	561	8,8%	8,8%	6,7%	6,4%
16B	948	949	945	936	6,0%	5,8%	3,8%	3,8%
171	1311	1296	1272	1212	3,5%	3,5%	2,8%	2,9%
172	1049	1065	1034	1060	6,1%	5,8%	3,6%	3,4%
16C	750	741	716	723	7,0%	6,9%	5,0%	4,8%
185	749	764	752	778	6,6%	6,4%	4,8%	4,6%
181	1110	1104	1138	1032	10,4%	9,0%	5,3%	3,5%
182	785	779	778	749	3,3%	3,3%	3,1%	3,8%
183	804	842	937	879	8,6%	7,9%	4,1%	4,2%
184	901	902	905	864	5,2%	5,2%	4,0%	4,1%
150	1096	1116	1079	1106	5,1%	5,0%	3,5%	3,3%

NUTSIII	$\hat{\mu}_i^{FH}$	$\hat{\mu}_i^{NS}$	$\hat{\mu}_{it}^{RY}$	$\hat{\mu}_{it}^{LP}$	$epr(\hat{\mu}_i^{FH})$	$epr(\hat{\mu}_i^{NS})$	$epr(\hat{\mu}_{it}^{RY})$	$epr(\hat{\mu}_{it}^{LP})$
				t=6				
111	809	822	812	814	6,6%	6,4%	4,6%	4,5%
112	781	793	744	744	6,3%	5,9%	5,0%	4,8%
113	734	751	735	746	5,3%	5,1%	4,7%	4,5%
114	1011	1011	1021	1020	2,1%	2,1%	2,4%	2,4%
115	649	650	646	655	2,2%	2,2%	2,5%	2,6%
116	713	717	715	730	3,7%	3,6%	3,9%	3,9%
117	607	608	609	610	0,9%	0,9%	1,0%	1,0%
118	768	763	744	733	4,4%	4,3%	4,2%	4,3%
161	825	826	833	818	5,1%	4,9%	4,1%	4,3%
162	998	986	956	966	5,6%	5,4%	3,8%	3,7%
163	753	762	765	796	5,5%	5,2%	4,6%	4,3%
164	576	541	632	594	6,1%	6,4%	5,2%	5,5%
165	741	713	768	754	10,8%	9,5%	4,8%	4,6%
166	386	390	392	418	5,7%	5,6%	6,5%	6,1%
167	507	513	519	545	4,4%	4,3%	4,8%	4,6%
168	733	699	702	664	5,5%	5,6%	5,0%	5,1%
169	739	674	780	657	12,2%	11,1%	5,9%	5,5%
16A	536	518	546	578	13,9%	12,4%	6,8%	6,1%
16B	1010	998	964	958	4,1%	4,0%	3,6%	3,7%
171	1380	1368	1370	1356	3,6%	3,6%	2,6%	2,7%
172	1071	1084	1062	1099	4,9%	4,6%	3,6%	3,3%
16C	712	704	724	735	3,8%	3,8%	3,9%	4,0%
185	792	801	777	805	5,2%	5,0%	4,2%	4,3%
181	1046	1111	1188	1118	8,8%	6,9%	5,0%	3,3%
182	744	707	767	719	8,0%	7,8%	5,0%	4,9%
183	903	908	942	882	6,7%	6,2%	4,2%	4,2%
184	929	938	939	905	5,6%	5,4%	4,2%	4,1%
150	1175	1180	1096	1128	6,7%	6,1%	3,6%	3,2%

Apêndice 30 – Estimativas do EQMP dos estimadores EBLUP de Salvati e de Rao-Yu

A30.1 Estimativas analíticas do EQMP dos estimadores EBLUP de Salvati e de Rao-Yu do preço médio de transacção da habitação, referentes ao terceiro trimestre de 2003

NUTSIII	$eqmp(\hat{\mu}_i^{NS})$	$g_1(\hat{\psi})$	$g_2(\hat{\psi})$	$2g_3(\hat{\psi})$	$eqmp(\hat{\mu}_{it}^{RY})$	$g_1(\hat{\psi})$	$g_2(\hat{\psi})$	$2g_3(\hat{\psi})$
111	4.343	87,8%	6,6%	5,7%	1.709	77,4%	1,0%	21,6%
112	1.444	97,1%	0,1%	2,8%	1.337	53,4%	1,0%	45,6%
113	975	97,7%	0,1%	2,2%	1.120	47,9%	0,2%	52,0%
114	4.012	88,9%	5,6%	5,4%	1.359	62,2%	0,8%	37,0%
115	1.593	96,1%	0,3%	3,6%	1.264	50,7%	0,4%	48,9%
116	805	97,6%	0,7%	1,7%	929	50,9%	0,2%	49,0%
117	484	98,5%	0,2%	1,3%	696	46,4%	0,0%	53,6%
118	60	99,8%	0,0%	0,2%	68	83,2%	0,0%	16,8%
161	2.681	94,3%	1,0%	4,6%	1.491	62,4%	1,4%	36,3%
162	1.433	95,5%	1,5%	3,0%	1.235	60,5%	0,2%	39,4%
163	1.131	97,1%	0,4%	2,5%	1.183	51,0%	0,1%	48,8%
164	458	98,6%	0,2%	1,2%	604	55,4%	0,3%	44,4%
165	1.116	95,4%	1,9%	2,7%	1.169	57,3%	0,3%	42,4%
166	483	98,9%	0,1%	1,0%	675	49,1%	0,8%	50,1%
167	485	98,2%	0,6%	1,3%	655	50,8%	0,1%	49,1%
168	405	99,0%	0,0%	1,0%	540	57,4%	0,3%	42,3%
169	4.530	92,9%	0,9%	6,2%	2.152	81,1%	0,2%	18,7%
16A	1.518	96,1%	0,6%	3,4%	1.339	61,0%	0,6%	38,4%
16B	2.367	94,4%	1,2%	4,4%	1.446	63,3%	2,1%	34,6%
171	1.672	91,4%	5,6%	3,0%	1.341	58,2%	6,1%	35,7%
172	2.405	95,9%	0,2%	3,9%	1.565	59,9%	2,6%	37,5%
16C	1.860	94,6%	1,6%	3,7%	1.374	52,8%	3,7%	43,5%
185	1.057	95,9%	2,1%	2,0%	960	65,7%	0,8%	33,5%
181	6.980	94,6%	0,0%	5,4%	3.558	91,4%	2,6%	6,0%
182	5.376	93,7%	0,2%	6,1%	1.870	70,5%	0,5%	29,0%
183	3.838	95,3%	0,0%	4,6%	1.799	71,5%	3,2%	25,3%
184	6.339	87,3%	7,5%	5,2%	1.630	68,8%	0,8%	30,4%
150	3.654	90,1%	5,6%	4,3%	1.795	64,7%	4,3%	31,0%
Média	---	95,1%	1,6%	3,3%	---	61,6%	1,2%	37,2%

Apêndice 31 – Estimativas EBLUP do preço médio de transacção da habitação ao nível das NUTSIII do continente, que garantem a consistência interna ao nível de NUTSII

A31.1 Estimativas do preço médio de transacção da habitação ao nível das NUTSIII do continente (valores em euros/m^2), produzidas através do estimador EBLUP espaciotemporal com restrições ao nível de NUTSII

NUTSIII	$\hat{\mu}_{it}^{LPR2}$	$epr(\hat{\mu}_{it}^{LPR2})$	$\hat{\mu}_{it}^{LPR2}$	$epr(\hat{\mu}_{it}^{LPR2})$	$\hat{\mu}_{it}^{LPR2}$	$epr(\hat{\mu}_{it}^{LPR2})$
	***t*=1**		***t*=2**		***t*=3**	
111	733	9,7%	760	11,3%	768	8,9%
112	719	10,2%	769	11,6%	723	11,2%
113	739	9,5%	723	9,3%	734	9,9%
114	1.067	7,3%	993	7,6%	994	7,5%
115	653	8,8%	663	9,7%	648	9,4%
116	746	11,4%	725	10,7%	723	12,3%
117	642	13,1%	662	11,8%	624	12,9%
118	674	12,8%	715	13,6%	748	13,5%
161	827	9,3%	827	10,1%	807	9,9%
162	988	8,3%	943	9,2%	921	9,1%
163	765	10,6%	762	13,1%	741	11,3%
164	628	11,3%	637	12,6%	662	12,0%
165	673	12,4%	677	12,5%	645	14,1%
166	449	15,1%	429	16,2%	492	14,9%
167	510	14,6%	512	15,3%	498	13,6%
168	485	13,1%	535	12,5%	685	12,5%
169	664	12,9%	687	14,2%	663	13,0%
16A	565	11,7%	552	13,8%	569	12,5%
16B	974	9,3%	937	10,1%	945	12,1%
171	1.255	7,5%	1.325	6,9%	1.269	8,2%
172	1.092	8,3%	1.129	7,3%	1.087	8,0%
16C	691	10,2%	721	11,2%	725	11,7%
185	871	9,4%	851	10,4%	951	11,3%
181	937	12,1%	971	13,2%	1.291	13,1%
182	735	10,7%	720	12,0%	794	9,3%
183	964	11,9%	931	10,9%	995	13,7%
184	819	12,1%	832	13,2%	883	11,6%
150	958	9,7%	1.043	8,7%	1.165	8,0%

NUTSIII	$\hat{\mu}_{it}^{LPR2}$	$epr(\hat{\mu}_{it}^{LPR2})$	$\hat{\mu}_{it}^{LPR2}$	$epr(\hat{\mu}_{it}^{LPR2})$	$\hat{\mu}_{it}^{LPR2}$	$epr(\hat{\mu}_{it}^{LPR2})$
	***t*=4**		***t*=5**		***t*=6**	
111	789	8,7%	786	8,7%	817	10,4%
112	727	11,8%	753	9,8%	750	11,3%
113	731	10,3%	733	10,4%	751	10,5%
114	1.028	10,2%	1.047	7,5%	1.032	7,0%
115	660	8,9%	659	9,0%	653	8,3%
116	714	10,3%	716	12,7%	728	12,5%
117	672	13,2%	643	14,3%	601	14,2%
118	691	14,8%	678	13,7%	734	11,9%
161	821	10,3%	817	11,5%	816	9,6%
162	967	9,3%	943	8,4%	973	8,6%
163	752	12,4%	773	11,1%	836	10,2%
164	707	13,4%	642	11,9%	607	11,7%
165	684	11,9%	714	12,8%	748	14,3%
166	446	17,4%	433	14,7%	441	14,9%
167	489	19,2%	517	14,2%	537	15,6%
168	635	14,7%	654	13,8%	654	13,7%
169	684	14,0%	701	14,1%	671	13,9%
16A	581	12,3%	559	12,0%	579	10,5%
16B	990	8,7%	954	9,3%	996	10,1%
171	1.304	8,3%	1.294	7,8%	1.383	8,7%
172	1.104	7,9%	1.152	8,4%	1.146	9,0%
16C	737	11,4%	764	10,7%	781	10,9%
185	1.000	12,5%	921	9,9%	850	11,0%
181	1.899	10,6%	1.453	11,6%	1.006	14,6%
182	949	9,8%	813	9,8%	747	11,5%
183	1.048	14,6%	1.028	12,4%	872	12,6%
184	916	13,9%	916	11,9%	915	11,3%
150	1.028	8,5%	1.086	10,3%	1.215	9,6%

Apêndice 32 – Pesos da estimação com restrições

A32.1 Pesos utilizados na estimação com restrições

NUTSIII	Ψ_0	Ψ_1						
		***t*=1**	***t*=2**	***t*=3**	***t*=4**	***t*=5**	***t*=6**	***t*=7**
111	0,038	0,034	0,030	0,039	0,046	0,043	0,039	0,041
112	0,060	0,076	0,062	0,055	0,033	0,060	0,069	0,055
113	0,071	0,095	0,062	0,097	0,051	0,057	0,050	0,069
114	0,713	0,691	0,748	0,674	0,713	0,712	0,741	0,718
115	0,036	0,031	0,036	0,050	0,042	0,029	0,028	0,034
116	0,047	0,047	0,035	0,044	0,071	0,057	0,047	0,039
117	0,014	0,011	0,012	0,024	0,015	0,017	0,009	0,012
118	0,021	0,014	0,014	0,017	0,029	0,026	0,017	0,032
161	0,143	0,157	0,137	0,125	0,133	0,158	0,164	0,127
162	0,240	0,257	0,286	0,201	0,233	0,224	0,237	0,233
163	0,134	0,162	0,153	0,112	0,114	0,094	0,142	0,145
164	0,071	0,066	0,051	0,098	0,092	0,052	0,063	0,070
165	0,059	0,052	0,044	0,051	0,064	0,106	0,066	0,044
166	0,003	0,002	0,002	0,004	0,003	0,006	0,003	0,003
167	0,002	0,002	0,002	0,002	0,003	0,003	0,003	0,003
168	0,010	0,007	0,009	0,009	0,011	0,009	0,013	0,016
169	0,019	0,018	0,012	0,022	0,017	0,015	0,019	0,031
16A	0,033	0,030	0,016	0,056	0,031	0,030	0,022	0,044
16B	0,197	0,171	0,193	0,201	0,214	0,236	0,177	0,199
171	0,625	0,589	0,594	0,605	0,647	0,689	0,627	0,650
172	0,375	0,411	0,406	0,395	0,353	0,311	0,373	0,350
16C	0,089	0,077	0,093	0,118	0,086	0,067	0,091	0,088
185	0,342	0,375	0,353	0,363	0,299	0,315	0,333	0,331
181	0,171	0,125	0,101	0,148	0,271	0,217	0,153	0,215
182	0,120	0,184	0,084	0,088	0,103	0,130	0,099	0,157
183	0,234	0,199	0,303	0,231	0,196	0,217	0,297	0,198
184	0,134	0,118	0,160	0,170	0,131	0,120	0,117	0,099
150	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000

Apêndice 33 – Estimação dos parâmetros de interesse com base no modelo espaciotemporal com restrições ao nível de NUTSII

FICHEIROS DE DADOS UTILIZADOS:

Est_dir_nutsii.sas7bdat – ficheiro que contém as estimativas directas dos parâmetros de interesse ao nível de NUTSII.

Iabh.sas7bdat – ficheiro que contém informação sobre os preços médios de avaliação bancária da habitação ao nível de NUTSIII.

Matv.sas7bdat – ficheiro que contém a matriz V estimada no âmbito do modelo espaciotemporal.

Matg.sas7bdat – ficheiro que contém a matriz G estimada no âmbito do modelo espaciotemporal.

Z.sas7bdat ; X0.sas7bdat ; Xr.sas7bdat ; C.sas7bdat ; Q.sas7bdat ; Csi.sas7bdat ; Psi0.sas7bdat ; Psi1.sas7bdat – ficheiros de dados tal como definidos no subcapítulo 6.4.

Os resultados estão guardados nos seguintes ficheiros:
- Estimativas EBLUP com restrições: **Est_LP_rest_nutii.sas7bdat**
- Estimativas bootstrap do EQMP do EBLUP com restrições: **EQMP_EBLUP_BOOT_LP_REST.sas7bdat**

```
/*********************************************
*********************************************/
```

CÓDIGO SAS

```
Data rest.q_;
set rest.est_dir_nutsii;
keep mean;
run;

Proc iml;
use rest.X0; read all into X0;
use rest.Matv; read all into Matv;
C= ginv( t(X0) * inv(Matv) * X0 );
create C from C; append from C; quit;

Data rest.C;
set C;
run;

Proc iml;
use rest.C; read all into C;
use rest.Q; read all into Q;
A=C* t(Q) * inv( Q * C * t(Q) );
create A from A; append from A; quit;

Data rest.A;
set A;
run;

/* Estimativas EBLUP com restrições */

Proc iml;
use rest.X0; read all into X0;
use rest.csi; read all into csi;
use rest.A; read all into A;
use rest.q_; read all into q_;
use rest.Q; read all into Q;
EstRest=X0 * CSI + X0 * A *(q_ - Q*CSI );
create EstRest from EstRest; append from
EstRest; quit;

/* Nota: As estimativas produzidas estão
ordenadas da seguinte forma: domínio*tempo */

Data EstRest1;
merge rest.iabh EstRest;
run;

Data Rest.Est_LP_rest_nutII;
set EstRest1;
rename col1=EstRest;
drop c x;
run;

/* Cálculo do acréscimo de g1 com restrições
*/

Proc iml;
use rest.X0; read all into X0;
use rest.A; read all into A;
use rest.Qv; read all into Qv;
use rest.Matg; read all into Matg;
use rest.Z; read all into Z;
use rest.Matv; read all into Matv;
use rest.q_; read all into q_;
use rest.Xr; read all into Xr;
g1_a1= X0 * A * Qv * Matg * t(Z) * inv(Matv) *
Z * Matg * t(Qv) * t(A) * t(X0) ;
create g1_a1 from g1_a1; append from g1_a1;
quit;

Proc iml;
use g1_a1; read all into g1_a1;
g1_aa1=vecdiag(g1_a1);
create g1_aa1 from g1_aa1; append from g1_aa1;
quit;

Data g1_aa1;
set g1_aa1;
rename col1=g1_aa1;
run;

Proc iml;
use rest.X0; read all into X0;
use rest.A; read all into A;
use rest.q_; read all into q_;
use rest.Xr; read all into Xr;
use rest.Effixo; read all into Effixo;
g1_a2= X0 * A * (q_ - XR*Effixo) * t(q_ -
XR*Effixo) * t(A) * t(X0);
create g1_a2 from g1_a2; append from g1_a2;
quit;

Proc iml;
use g1_a2; read all into g1_a2;
g1_aa2=vecdiag(g1_a2);
create g1_aa2 from g1_aa2; append from g1_aa2;
quit;

Data g1_aa2;
set g1_aa2;
rename col1=g1_aa2;
run;

Data g1_a;
merge EstRest1 g1_aa1 g1_aa2;
run;

Data g1_a;
set g1_a;
rename col1=EstRest;
drop c x;
run;

Proc sort data=g1_a;
by t i;
run;

Data g1_a;
merge g1_a Hb.Est_pc;
by t i;
run;

Data g1_a;
set g1_a;
```

```
g1r=g1 + g1_aa1 + g1_aa2;
keep t i estrest g1 g1_aa1 g1_aa2 g1r;
run;

Proc sort data=g1_a;
by i t;
run;

Data Rest.Est_LP_rest_nutii;
set g1_a;
run;

/********************************************
********************************************/
/* Estimação do EQMP do EBLUP espaciotemporal
com restrições pelo método bootstrap ****/
/********************************************
********************************************/

Data null;
file 'C:\Documents and Settings\Luis Pereira
\My Documents\EstudoEmpirico\chamamacro.txt';
   do i= 1 to 250; k=i;
var='%ext_am('||i||','||k||');'; put var; end;
run;

%macro ext_am(i,k);

Data a1;
do i=1 to 28;
        do t=1 to 7;
                e1=rannor(&k+10000);
                col1=rannor(&k+20000);
                output;
        end;
end; run;

Data a2;
do i=1 to 28;
        col1=rannor(&k+30000);
        output;
end;
drop i; run;

/* Geração dos efeitos aleatórios */

Data a3;
set a1;
keep col1;
run;

Proc iml;
use a2; read all into a2;
use a3; read all into a3;
a4= a2//a3;
create a4 from a4; append from a4; quit;

Proc iml;
use Rest.Matg; read all into Matg;
use a4; read all into a4;
use Rest.Z; read all into Z;
Matg_cholesky=root(Matg); /* Decomposição de
cholesky */
v_ast=Z*Matg_cholesky*a4;
create v_ast from v_ast; append from v_ast;
quit;

Data v_ast;
set v_ast;
rename col1=v_ast;
run;

/* Geração dos erros da sondagem */

Proc sort data= Hb.Estimat_directas1
out=var_est_dir;
by i t;
run;

Data var_est_dir;
set var_est_dir;
keep var; run;

Data a5;
set a1;
keep e1;
run;

Proc iml;
use var_est_dir; read all into var_est_dir;
use a5; read all into a5;
R=diag(var_est_dir);
R_cholesky=root(R); /* Decomposição de
cholesky */
e_ast=R_cholesky*a5;
create e_ast from e_ast; append from e_ast;
quit;

Data e_ast;
set e_ast;
rename col1=e_ast;
run;

/* Geração dos dados bootstrap */

Proc iml;
use Rest.X; read all into X;
use Rest.Effixo; read all into Effixo;
a7=X*Effixo;   /* Efeitos fixos do modelo */
create a7 from a7; append from a7;
quit;

Data a7;
set a7;
rename col1=XB; run;

Data a6;
merge Rest.Iabh a7 v_ast e_ast var_est_dir;
y=XB + v_ast + e_ast;
run;

Data MatX;
set Rest.X; run;

/********************************************
********************************************/
/*** Fim da geração dos dados no bootstrap **/
/********************************************
********************************************/

/* Dependência espacial */

Data dataphi;
phi=0.29;
run;

/* Autocorrelação temporal */

Data dataro;
ro=0.37;
run;

/* Matriz B=(I-phi*W)'*(I-phi*W) */

Data W;
set PC.W2;   /* Matriz de distâncias */
run;

Proc iml;
use W; read all into W;
use dataphi; read all into dataphi;
Ident=I(28);
B = t(Ident - dataphi*W)*(Ident - dataphi*W);
create B from B; append from B; quit;

Proc iml;
use B; read all into B;
Binv=inv(B);   /* Inversa de B */
create Binv from Binv; append from Binv; quit;

/* Matriz P */

Data dataro2;
set dataro;
ro2= sqrt(1-ro*ro);
keep ro2;
run;

Proc iml;
use dataro;
read all into dataro;
use dataro2;
read all into dataro2;
I1={1 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
```

```
I2={0 0 0 0 0 0 0, 0 1 0 0 0 0 0, 0 0 1 0 0 0
0, 0 0 0 1 0 0 0, 0 0 0 0 1 0 0, 0 0 0 0 0 1
0, 0 0 0 0 0 0 1};
I4={0 0 0 0 0 0 0, 1 0 0 0 0 0 0, 0 1 0 0 0 0
0, 0 0 1 0 0 0 0, 0 0 0 1 0 0 0, 0 0 0 0 1 0
0, 0 0 0 0 0 1 0};
P = dataro2*I1 + I2 - dataro*I4;
create P from P;
append from P;
quit;

Proc iml;
use dataro;
read all into dataro;
use dataro2;
read all into dataro2;
C1={1, 0, 0, 0, 0, 0, 0};
C2={0, 1, 1, 1, 1, 1, 1};
f = dataro2*C1 + C2 - dataro*C2;
create f from f;
append from f;
quit;

Proc iml;
use f;
read all into f;
c=f`*f;
create c from c;
append from c;
quit;

Proc iml;
use f;
read all into f;
use c;
read all into c;
D=(1/c)*f*f`;
create D from D;
append from D;
quit;

/* *** */

Data iabh;
set a6;
keep i t c x; run;

Data ipth;
set a6;
keep i t y; run;

Data iabhipth1;
set a6;
keep i t y x; run;

/** Preparação do ficheiro das observações da
variável de interesse **/

Proc sort data=ipth;
by i t;
run;

Data tetad1; set ipth; where i= 1; keep y;
Data tetad2; set ipth; where i= 2; keep y;
Data tetad3; set ipth; where i= 3; keep y;
Data tetad4; set ipth; where i= 4; keep y;
Data tetad5; set ipth; where i= 5; keep y;
Data tetad6; set ipth; where i= 6; keep y;
Data tetad7; set ipth; where i= 7; keep y;
Data tetad8; set ipth; where i= 8; keep y;
Data tetad9; set ipth; where i= 9; keep y;
Data tetad10; set ipth; where i= 10; keep y;
Data tetad11; set ipth; where i= 11; keep y;
Data tetad12; set ipth; where i= 12; keep y;
Data tetad13; set ipth; where i= 13; keep y;
Data tetad14; set ipth; where i= 14; keep y;
Data tetad15; set ipth; where i= 15; keep y;
Data tetad16; set ipth; where i= 16; keep y;
Data tetad17; set ipth; where i= 17; keep y;
Data tetad18; set ipth; where i= 18; keep y;
Data tetad19; set ipth; where i= 19; keep y;
Data tetad20; set ipth; where i= 20; keep y;
Data tetad21; set ipth; where i= 21; keep y;
Data tetad22; set ipth; where i= 22; keep y;
Data tetad23; set ipth; where i= 23; keep y;
Data tetad24; set ipth; where i= 24; keep y;
Data tetad25; set ipth; where i= 25; keep y;
Data tetad26; set ipth; where i= 26; keep y;
Data tetad27; set ipth; where i= 27; keep y;
Data tetad28; set ipth; where i= 28; keep y;
run;
/** Preparação do ficheiro das observações da
variável auxiliar **/

Proc sort data=iabh;
by i t;
run;

Proc sql; create table iabh0 as select c, x,
i, t from iabh;

Data xd1; set iabh0; where i= 1; keep c x;
Data xd2; set iabh0; where i= 2; keep c x;
Data xd3; set iabh0; where i= 3; keep c x;
Data xd4; set iabh0; where i= 4; keep c x;
Data xd5; set iabh0; where i= 5; keep c x;
Data xd6; set iabh0; where i= 6; keep c x;
Data xd7; set iabh0; where i= 7; keep c x;
Data xd8; set iabh0; where i= 8; keep c x;
Data xd9; set iabh0; where i= 9; keep c x;
Data xd10; set iabh0; where i= 10; keep c x;
Data xd11; set iabh0; where i= 11; keep c x;
Data xd12; set iabh0; where i= 12; keep c x;
Data xd13; set iabh0; where i= 13; keep c x;
Data xd14; set iabh0; where i= 14; keep c x;
Data xd15; set iabh0; where i= 15; keep c x;
Data xd16; set iabh0; where i= 16; keep c x;
Data xd17; set iabh0; where i= 17; keep c x;
Data xd18; set iabh0; where i= 18; keep c x;
Data xd19; set iabh0; where i= 19; keep c x;
Data xd20; set iabh0; where i= 20; keep c x;
Data xd21; set iabh0; where i= 21; keep c x;
Data xd22; set iabh0; where i= 22; keep c x;
Data xd23; set iabh0; where i= 23; keep c x;
Data xd24; set iabh0; where i= 24; keep c x;
Data xd25; set iabh0; where i= 25; keep c x;
Data xd26; set iabh0; where i= 26; keep c x;
Data xd27; set iabh0; where i= 27; keep c x;
Data xd28; set iabh0; where i= 28; keep c x;
run;

/** Cálculo da matriz Z(1) **/

Proc iml;
use tetad1; read all into tetad1;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd1=(Ident - D)*P*tetad1;
create zd1 from zd1; append from zd1; quit;
Proc iml;
use tetad2; read all into tetad2;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd2=(Ident - D)*P*tetad2;
create zd2 from zd2; append from zd2; quit;
Proc iml;
use tetad3; read all into tetad3;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd3=(Ident - D)*P*tetad3;
create zd3 from zd3; append from zd3; quit;
Proc iml;
use tetad4; read all into tetad4;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd4=(Ident - D)*P*tetad4;
create zd4 from zd4; append from zd4; quit;
Proc iml;
use tetad5; read all into tetad5;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd5=(Ident - D)*P*tetad5;
create zd5 from zd5; append from zd5; quit;
Proc iml;
use tetad6; read all into tetad6;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd6=(Ident - D)*P*tetad6;
create zd6 from zd6; append from zd6; quit;
Proc iml;
```

```
use tetad7; read all into tetad7;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd7=(Ident - D)*P*tetad7;
create zd7 from zd7; append from zd7; quit;
Proc iml;
use tetad8; read all into tetad8;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd8=(Ident - D)*P*tetad8;
create zd8 from zd8; append from zd8; quit;
Proc iml;
use tetad9; read all into tetad9;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd9=(Ident - D)*P*tetad9;
create zd9 from zd9; append from zd9; quit;
Proc iml;
use tetad10; read all into tetad10;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd10=(Ident - D)*P*tetad10;
create zd10 from zd10; append from zd10; quit;
Proc iml;
use tetad11; read all into tetad11;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd11=(Ident - D)*P*tetad11;
create zd11 from zd11; append from zd11; quit;
Proc iml;
use tetad12; read all into tetad12;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd12=(Ident - D)*P*tetad12;
create zd12 from zd12; append from zd12; quit;
Proc iml;
use tetad13; read all into tetad13;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd13=(Ident - D)*P*tetad13;
create zd13 from zd13; append from zd13; quit;
Proc iml;
use tetad14; read all into tetad14;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd14=(Ident - D)*P*tetad14;
create zd14 from zd14; append from zd14; quit;
Proc iml;
use tetad15; read all into tetad15;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd15=(Ident - D)*P*tetad15;
create zd15 from zd15; append from zd15; quit;
Proc iml;
use tetad16; read all into tetad16;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd16=(Ident - D)*P*tetad16;
create zd16 from zd16; append from zd16; quit;
Proc iml;
use tetad17; read all into tetad17;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd17=(Ident - D)*P*tetad17;
create zd17 from zd17; append from zd17; quit;
Proc iml;
use tetad18; read all into tetad18;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd18=(Ident - D)*P*tetad18;
create zd18 from zd18; append from zd18; quit;
Proc iml;
use tetad19; read all into tetad19;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd19=(Ident - D)*P*tetad19;
create zd19 from zd19; append from zd19; quit;
Proc iml;
use tetad20; read all into tetad20;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd20=(Ident - D)*P*tetad20;
create zd20 from zd20; append from zd20; quit;
Proc iml;
use tetad21; read all into tetad21;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd21=(Ident - D)*P*tetad21;
create zd21 from zd21; append from zd21; quit;
Proc iml;
use tetad22; read all into tetad22;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd22=(Ident - D)*P*tetad22;
create zd22 from zd22; append from zd22; quit;
Proc iml;
use tetad23; read all into tetad23;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd23=(Ident - D)*P*tetad23;
create zd23 from zd23; append from zd23; quit;
Proc iml;
use tetad24; read all into tetad24;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd24=(Ident - D)*P*tetad24;
create zd24 from zd24; append from zd24; quit;
Proc iml;
use tetad25; read all into tetad25;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd25=(Ident - D)*P*tetad25;
create zd25 from zd25; append from zd25; quit;
Proc iml;
use tetad26; read all into tetad26;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd26=(Ident - D)*P*tetad26;
create zd26 from zd26; append from zd26; quit;
Proc iml;
use tetad27; read all into tetad27;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd27=(Ident - D)*P*tetad27;
create zd27 from zd27; append from zd27; quit;
Proc iml;
use tetad28; read all into tetad28;
use P; read all into P;
use D; read all into D;
Ident=I(7);
zd28=(Ident - D)*P*tetad28;
create zd28 from zd28; append from zd28; quit;

Data Z1;
set zd1 zd2 zd3 zd4 zd5 zd6 zd7 zd8 zd9 zd10
zd11 zd12 zd13 zd14 zd15
zd16 zd17 zd18 zd19 zd20 zd21 zd22 zd23 zd24
zd25 zd26 zd27 zd28;
run;

/** Cálculo da matriz H(1) **/

Proc iml;
use xd1; read all into xd1;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd1=(Ident - D)*P*xd1;
create hd1 from hd1; append from hd1; quit;
Proc iml;
use xd2; read all into xd2;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd2=(Ident - D)*P*xd2;
```

```
create hd2 from hd2; append from hd2; quit;
Proc iml;
use xd3; read all into xd3;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd3=(Ident - D)*P*xd3;
create hd3 from hd3; append from hd3; quit;
Proc iml;
use xd4; read all into xd4;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd4=(Ident - D)*P*xd4;
create hd4 from hd4; append from hd4; quit;
Proc iml;
use xd5; read all into xd5;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd5=(Ident - D)*P*xd5;
create hd5 from hd5; append from hd5; quit;
Proc iml;
use xd6; read all into xd6;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd6=(Ident - D)*P*xd6;
create hd6 from hd6; append from hd6; quit;
Proc iml;
use xd7; read all into xd7;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd7=(Ident - D)*P*xd7;
create hd7 from hd7; append from hd7; quit;
Proc iml;
use xd8; read all into xd8;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd8=(Ident - D)*P*xd8;
create hd8 from hd8; append from hd8; quit;
Proc iml;
use xd9; read all into xd9;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd9=(Ident - D)*P*xd9;
create hd9 from hd9; append from hd9; quit;
Proc iml;
use xd10; read all into xd10;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd10=(Ident - D)*P*xd10;
create hd10 from hd10; append from hd10; quit;
Proc iml;
use xd11; read all into xd11;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd11=(Ident - D)*P*xd11;
create hd11 from hd11; append from hd11; quit;
Proc iml;
use xd12; read all into xd12;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd12=(Ident - D)*P*xd12;
create hd12 from hd12; append from hd12; quit;
Proc iml;
use xd13; read all into xd13;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd13=(Ident - D)*P*xd13;
create hd13 from hd13; append from hd13; quit;
Proc iml;
use xd14; read all into xd14;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd14=(Ident - D)*P*xd14;
create hd14 from hd14; append from hd14; quit;
Proc iml;
use xd15; read all into xd15;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd15=(Ident - D)*P*xd15;
create hd15 from hd15; append from hd15; quit;
Proc iml;
use xd16; read all into xd16;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd16=(Ident - D)*P*xd16;
create hd16 from hd16; append from hd16; quit;
Proc iml;
use xd17; read all into xd17;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd17=(Ident - D)*P*xd17;
create hd17 from hd17; append from hd17; quit;
Proc iml;
use xd18; read all into xd18;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd18=(Ident - D)*P*xd18;
create hd18 from hd18; append from hd18; quit;
Proc iml;
use xd19; read all into xd19;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd19=(Ident - D)*P*xd19;
create hd19 from hd19; append from hd19; quit;
Proc iml;
use xd20; read all into xd20;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd20=(Ident - D)*P*xd20;
create hd20 from hd20; append from hd20; quit;
Proc iml;
use xd21; read all into xd21;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd21=(Ident - D)*P*xd21;
create hd21 from hd21; append from hd21; quit;
Proc iml;
use xd22; read all into xd22;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd22=(Ident - D)*P*xd22;
create hd22 from hd22; append from hd22; quit;
Proc iml;
use xd23; read all into xd23;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd23=(Ident - D)*P*xd23;
create hd23 from hd23; append from hd23; quit;
Proc iml;
use xd24; read all into xd24;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd24=(Ident - D)*P*xd24;
create hd24 from hd24; append from hd24; quit;
Proc iml;
use xd25; read all into xd25;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd25=(Ident - D)*P*xd25;
create hd25 from hd25; append from hd25; quit;
Proc iml;
use xd26; read all into xd26;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd26=(Ident - D)*P*xd26;
create hd26 from hd26; append from hd26; quit;
Proc iml;
use xd27; read all into xd27;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd27=(Ident - D)*P*xd27;
create hd27 from hd27; append from hd27; quit;
```

```
Proc iml;
use xd28; read all into xd28;
use P; read all into P;
use D; read all into D;
Ident=I(7);
hd28=(Ident - D)*P*xd28;
create hd28 from hd28; append from hd28; quit;

Data H1;
set hd1 hd2 hd3 hd4 hd5 hd6 hd7 hd8 hd9 hd10
hd11 hd12 hd13 hd14 hd15
hd16 hd17 hd18 hd19 hd20 hd21 hd22 hd23 hd24
hd25 hd26 hd27 hd28;
run;

/** Regressão de Z(1) sobre H(1) **/

Data Z1;
set Z1;
rename col1=col0; run;

Data reg;
merge Z1 H1; run;

proc reg data=reg noprint;
        model col0=col1 col2 / noint;
        output out=rg1  residual=r1;
run; quit;

Data rg1;
set rg1;
qr1=r1*r1;
Keep qr1;
run;

Proc means data=rg1 sum noprint;
var qr1;
output out=rg1_ sum=sqr1;
run;

Data rg1_;
set rg1_;
keep sqr1; run;

/** Cálculo da variância sigma2_u **/

Proc iml;
use H1; read all into H1;
Hs1=H1*ginv(t(H1)*H1)*t(H1); /* ginv(A) is the
Moore-Penrose generalized inverse matrix of A
*/
create Hs1 from Hs1; append from Hs1; quit; /*
Hs1 é a matriz H*(1) */

Proc iml;
use D; read all into D;
Ident=I(7);
ID=Ident - D; /* Matriz I - D */
create ID from ID; append from ID; quit;

Proc iml;
use ID; read all into ID;
blockID1=BLOCK(ID,ID,ID,ID,ID,ID,ID,ID,ID,I
D,ID,ID,ID);
create blockID1 from blockID1; append from
blockID1; quit;
Proc iml;
use blockID1; read all into blockID1;
blockID=BLOCK(blockID1,blockID1);
create blockID from blockID; append from
blockID; quit;

Data s1;
set a6;
keep var t i; run;

Proc sort data=s1;
by i t; run;

Data Sigma_1;
set s1;
if i=1;
keep var;
run;
Proc iml;
use Sigma_1; read all into Sigma_1;
Sigma1=diag(Sigma_1);
create Sigma1 from Sigma1; append from Sigma1;
quit;
Proc iml;
use P; read all into P;
use Sigma1; read all into Sigma1;
P1=P*Sigma1*t(P);
create P1 from P1; append from P1; quit;

Data Sigma_2;
set s1;
if i=2;
keep var;
run;
Proc iml;
use Sigma_2; read all into Sigma_2;
Sigma2=diag(Sigma_2);
create Sigma2 from Sigma2; append from Sigma2;
quit;
Proc iml;
use P; read all into P;
use Sigma2; read all into Sigma2;
P2=P*Sigma2*t(P);
create P2 from P2; append from P2; quit;

Data Sigma_3;
set s1;
if i=3;
keep var;
run;
Proc iml;
use Sigma_3; read all into Sigma_3;
Sigma3=diag(Sigma_3);
create Sigma3 from Sigma3; append from Sigma3;
quit;
Proc iml;
use P; read all into P;
use Sigma3; read all into Sigma3;
P3=P*Sigma3*t(P);
create P3 from P3; append from P3; quit;

Data Sigma_4;
set s1;
if i=4;
keep var;
run;
Proc iml;
use Sigma_4; read all into Sigma_4;
Sigma4=diag(Sigma_4);
create Sigma4 from Sigma4; append from Sigma4;
quit;
Proc iml;
use P; read all into P;
use Sigma4; read all into Sigma4;
P4=P*Sigma4*t(P);
create P4 from P4; append from P4; quit;

Data Sigma_5;
set s1;
if i=5;
keep var;
run;
Proc iml;
use Sigma_5; read all into Sigma_5;
Sigma5=diag(Sigma_5);
create Sigma5 from Sigma5; append from Sigma5;
quit;
Proc iml;
use P; read all into P;
use Sigma5; read all into Sigma5;
P5=P*Sigma5*t(P);
create P5 from P5; append from P5; quit;

Data Sigma_6;
set s1;
if i=6;
keep var;
run;
Proc iml;
use Sigma_6; read all into Sigma_6;
Sigma6=diag(Sigma_6);
create Sigma6 from Sigma6; append from Sigma6;
quit;
Proc iml;
use P; read all into P;
use Sigma6; read all into Sigma6;
P6=P*Sigma6*t(P);
create P6 from P6; append from P6; quit;
```

```
Data Sigma_7;
set s1;
if i=7;
keep var;
run;
Proc iml;
use Sigma_7; read all into Sigma_7;
Sigma7=diag(Sigma_7);
create Sigma7 from Sigma7; append from Sigma7;
quit;
Proc iml;
use P; read all into P;
use Sigma7; read all into Sigma7;
P7=P*Sigma7*t(P);
create P7 from P7; append from P7; quit;

Data Sigma_8;
set s1;
if i=8;
keep var;
run;
Proc iml;
use Sigma_8; read all into Sigma_8;
Sigma8=diag(Sigma_8);
create Sigma8 from Sigma8; append from Sigma8;
quit;
Proc iml;
use P; read all into P;
use Sigma8; read all into Sigma8;
P8=P*Sigma8*t(P);
create P8 from P8; append from P8; quit;

Data Sigma_9;
set s1;
if i=9;
keep var;
run;
Proc iml;
use Sigma_9; read all into Sigma_9;
Sigma9=diag(Sigma_9);
create Sigma9 from Sigma9; append from Sigma9;
quit;
Proc iml;
use P; read all into P;
use Sigma9; read all into Sigma9;
P9=P*Sigma9*t(P);
create P9 from P9; append from P9; quit;

Data Sigma_10;
set s1;
if i=10;
keep var;
run;
Proc iml;
use Sigma_10; read all into Sigma_10;
Sigma10=diag(Sigma_10);
create Sigma10 from Sigma10; append from
Sigma10; quit;
Proc iml;
use P; read all into P;
use Sigma10; read all into Sigma10;
P10=P*Sigma10*t(P);
create P10 from P10; append from P10; quit;

Data Sigma_11;
set s1;
if i=11;
keep var;
run;
Proc iml;
use Sigma_11; read all into Sigma_11;
Sigma11=diag(Sigma_11);
create Sigma11 from Sigma11; append from
Sigma11; quit;
Proc iml;
use P; read all into P;
use Sigma11; read all into Sigma11;
P11=P*Sigma11*t(P);
create P11 from P11; append from P11; quit;

Data Sigma_12;
set s1;
if i=12;
keep var;
run;
Proc iml;
use Sigma_12; read all into Sigma_12;
Sigma12=diag(Sigma_12);
create Sigma12 from Sigma12; append from
Sigma12; quit;
Proc iml;
use P; read all into P;
use Sigma12; read all into Sigma12;
P12=P*Sigma12*t(P);
create P12 from P12; append from P12; quit;

Data Sigma_13;
set s1;
if i=13;
keep var;
run;
Proc iml;
use Sigma_13; read all into Sigma_13;
Sigma13=diag(Sigma_13);
create Sigma13 from Sigma13; append from
Sigma13; quit;
Proc iml;
use P; read all into P;
use Sigma13; read all into Sigma13;
P13=P*Sigma13*t(P);
create P13 from P13; append from P13; quit;

Data Sigma_14;
set s1;
if i=14;
keep var;
run;
Proc iml;
use Sigma_14; read all into Sigma_14;
Sigma14=diag(Sigma_14);
create Sigma14 from Sigma14; append from
Sigma14; quit;
Proc iml;
use P; read all into P;
use Sigma14; read all into Sigma14;
P14=P*Sigma14*t(P);
create P14 from P14; append from P14; quit;

Data Sigma_15;
set s1;
if i=15;
keep var;
run;
Proc iml;
use Sigma_15; read all into Sigma_15;
Sigma15=diag(Sigma_15);
create Sigma15 from Sigma15; append from
Sigma15; quit;
Proc iml;
use P; read all into P;
use Sigma15; read all into Sigma15;
P15=P*Sigma15*t(P);
create P15 from P15; append from P15; quit;

Data Sigma_16;
set s1;
if i=16;
keep var;
run;
Proc iml;
use Sigma_16; read all into Sigma_16;
Sigma16=diag(Sigma_16);
create Sigma16 from Sigma16; append from
Sigma16; quit;
Proc iml;
use P; read all into P;
use Sigma16; read all into Sigma16;
P16=P*Sigma16*t(P);
create P16 from P16; append from P16; quit;

Data Sigma_17;
set s1;
if i=17;
keep var;
run;
Proc iml;
use Sigma_17; read all into Sigma_17;
Sigma17=diag(Sigma_17);
create Sigma17 from Sigma17; append from
Sigma17; quit;
Proc iml;
use P; read all into P;
use Sigma17; read all into Sigma17;
P17=P*Sigma17*t(P);
create P17 from P17; append from P17; quit;
```

```
Data Sigma_18;
set s1;
if i=18;
keep var;
run;
Proc iml;
use Sigma_18; read all into Sigma_18;
Sigma18=diag(Sigma_18);
create Sigma18 from Sigma18; append from
Sigma18; quit;
Proc iml;
use P; read all into P;
use Sigma18; read all into Sigma18;
P18=P*Sigma18*t(P);
create P18 from P18; append from P18; quit;

Data Sigma_19;
set s1;
if i=19;
keep var;
run;
Proc iml;
use Sigma_19; read all into Sigma_19;
Sigma19=diag(Sigma_19);
create Sigma19 from Sigma19; append from
Sigma19; quit;
Proc iml;
use P; read all into P;
use Sigma19; read all into Sigma19;
P19=P*Sigma19*t(P);
create P19 from P19; append from P19; quit;

Data Sigma_20;
set s1;
if i=20;
keep var;
run;
Proc iml;
use Sigma_20; read all into Sigma_20;
Sigma20=diag(Sigma_20);
create Sigma20 from Sigma20; append from
Sigma20; quit;
Proc iml;
use P; read all into P;
use Sigma20; read all into Sigma20;
P20=P*Sigma20*t(P);
create P20 from P20; append from P20; quit;

Data Sigma_21;
set s1;
if i=21;
keep var;
run;
Proc iml;
use Sigma_21; read all into Sigma_21;
Sigma21=diag(Sigma_21);
create Sigma21 from Sigma21; append from
Sigma21; quit;
Proc iml;
use P; read all into P;
use Sigma21; read all into Sigma21;
P21=P*Sigma21*t(P);
create P21 from P21; append from P21; quit;

Data Sigma_22;
set s1;
if i=22;
keep var;
run;
Proc iml;
use Sigma_22; read all into Sigma_22;
Sigma22=diag(Sigma_22);
create Sigma22 from Sigma22; append from
Sigma22; quit;
Proc iml;
use P; read all into P;
use Sigma22; read all into Sigma22;
P22=P*Sigma22*t(P);
create P22 from P22; append from P22; quit;

Data Sigma_23;
set s1;
if i=23;
keep var;
run;
Proc iml;
use Sigma_23; read all into Sigma_23;
Sigma23=diag(Sigma_23);
create Sigma23 from Sigma23; append from
Sigma23; quit;
Proc iml;
use P; read all into P;
use Sigma23; read all into Sigma23;
P23=P*Sigma23*t(P);
create P23 from P23; append from P23; quit;

Data Sigma_24;
set s1;
if i=24;
keep var;
run;
Proc iml;
use Sigma_24; read all into Sigma_24;
Sigma24=diag(Sigma_24);
create Sigma24 from Sigma24; append from
Sigma24; quit;
Proc iml;
use P; read all into P;
use Sigma24; read all into Sigma24;
P24=P*Sigma24*t(P);
create P24 from P24; append from P24; quit;

Data Sigma_25;
set s1;
if i=25;
keep var;
run;
Proc iml;
use Sigma_25; read all into Sigma_25;
Sigma25=diag(Sigma_25);
create Sigma25 from Sigma25; append from
Sigma25; quit;
Proc iml;
use P; read all into P;
use Sigma25; read all into Sigma25;
P25=P*Sigma25*t(P);
create P25 from P25; append from P25; quit;

Data Sigma_26;
set s1;
if i=26;
keep var;
run;
Proc iml;
use Sigma_26; read all into Sigma_26;
Sigma26=diag(Sigma_26);
create Sigma26 from Sigma26; append from
Sigma26; quit;
Proc iml;
use P; read all into P;
use Sigma26; read all into Sigma26;
P26=P*Sigma26*t(P);
create P26 from P26; append from P26; quit;

Data Sigma_27;
set s1;
if i=27;
keep var;
run;
Proc iml;
use Sigma_27; read all into Sigma_27;
Sigma27=diag(Sigma_27);
create Sigma27 from Sigma27; append from
Sigma27; quit;
Proc iml;
use P; read all into P;
use Sigma27; read all into Sigma27;
P27=P*Sigma27*t(P);
create P27 from P27; append from P27; quit;

Data Sigma_28;
set s1;
if i=28;
keep var;
run;
Proc iml;
use Sigma_28; read all into Sigma_28;
Sigma28=diag(Sigma_28);
create Sigma28 from Sigma28; append from
Sigma28; quit;
Proc iml;
use P; read all into P;
use Sigma28; read all into Sigma28;
P28=P*Sigma28*t(P);
create P28 from P28; append from P28; quit;
```

```
Proc iml;
use P1; read all into P1;
use P2; read all into P2;
use P3; read all into P3;
use P4; read all into P4;
use P5; read all into P5;
use P6; read all into P6;
use P7; read all into P7;
use P8; read all into P8;
use P9; read all into P9;
use P10; read all into P10;
use P11; read all into P11;
use P12; read all into P12;
use P13; read all into P13;
use P14; read all into P14;
blockPP1=BLOCK(P1,P2,P3,P4,P5,P6,P7,P8,P9,P10,
P11,P12,P13,P14);
create blockPP1 from blockPP1; append from
blockPP1; quit;

Proc iml;
use P15; read all into P15;
use P16; read all into P16;
use P17; read all into P17;
use P18; read all into P18;
use P19; read all into P19;
use P20; read all into P20;
use P21; read all into P21;
use P22; read all into P22;
use P23; read all into P23;
use P24; read all into P24;
use P25; read all into P25;
use P26; read all into P26;
use P27; read all into P27;
use P28; read all into P28;
blockPP2=BLOCK(P15,P16,P17,P18,P19,P20,P21,P22
,P23,P24,P25,P26,P27,P28);
create blockPP2 from blockPP2; append from
blockPP2; quit;

Proc iml;
use blockPP1; read all into blockPP1;
use blockPP2; read all into blockPP2;
blockPP=BLOCK(blockPP1,blockPP2);
create blockPP from blockPP; append from
blockPP; quit;

/* *** Fim *** */

/* Obs: The function 'rank' in SAS IML does
not compute the rank of a matrix */

Proc iml;
use H1; read all into H1;
HH=H1*t(H1);
eval=eigval(HH);
create eval from eval; append from eval; quit;

Data id;
do id=1 to 196;
output; end;
run;

Data eval;
merge id eval; run;

Data new;
set eval;
by id;
retain count 0;
if first.id then count=0;
if col1 > 0.000001 then count = count + 1;
run;

Proc means data=new sum noprint;
var count;
output out=caracH1 sum=caracH1;
run;

Data caracH1;
set caracH1;
keep caracH1; run;

Proc iml;
use blockPP; read all into blockPP;
use blockID; read all into blockID;
use Hs1; read all into Hs1;
use rg1_; read all into rg1_;
use caracH1; read all into caracH1;
varu= (rg1_ -  trace((blockID -
Hs1)*blockPP))*(1/(28*6 - caracH1));
create varu from varu; append from varu; quit;

/* Termina aqui o cálculo da variância
sigma2_u */

/** Cálculo da matriz: z2= c^(-0.5)*f'*P*Teta
- var. dependente do modelo transformado **/

Proc iml;
use tetad1; read all into tetad1;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y1=(1/sqrt(c))*t(f)*P*tetad1;
create y1 from y1; append from y1; quit;
Proc iml;
use tetad2; read all into tetad2;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y2=(1/sqrt(c))*t(f)*P*tetad2;
create y2 from y2; append from y2; quit;
Proc iml;
use tetad3; read all into tetad3;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y3=(1/sqrt(c))*t(f)*P*tetad3;
create y3 from y3; append from y3; quit;
Proc iml;
use tetad4; read all into tetad4;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y4=(1/sqrt(c))*t(f)*P*tetad4;
create y4 from y4; append from y4; quit;
Proc iml;
use tetad5; read all into tetad5;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y5=(1/sqrt(c))*t(f)*P*tetad5;
create y5 from y5; append from y5; quit;
Proc iml;
use tetad6; read all into tetad6;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y6=(1/sqrt(c))*t(f)*P*tetad6;
create y6 from y6; append from y6; quit;
Proc iml;
use tetad7; read all into tetad7;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y7=(1/sqrt(c))*t(f)*P*tetad7;
create y7 from y7; append from y7; quit;
Proc iml;
use tetad8; read all into tetad8;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y8=(1/sqrt(c))*t(f)*P*tetad8;
create y8 from y8; append from y8; quit;
Proc iml;
use tetad9; read all into tetad9;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y9=(1/sqrt(c))*t(f)*P*tetad9;
create y9 from y9; append from y9; quit;
Proc iml;
use tetad10; read all into tetad10;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y10=(1/sqrt(c))*t(f)*P*tetad10;
create y10 from y10; append from y10; quit;
Proc iml;
use tetad11; read all into tetad11;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y11=(1/sqrt(c))*t(f)*P*tetad11;
create y11 from y11; append from y11; quit;
```

```
Proc iml;
use tetad12; read all into tetad12;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y12=(1/sqrt(c))*t(f)*P*tetad12;
create y12 from y12; append from y12; quit;
Proc iml;
use tetad13; read all into tetad13;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y13=(1/sqrt(c))*t(f)*P*tetad13;
create y13 from y13; append from y13; quit;
Proc iml;
use tetad14; read all into tetad14;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y14=(1/sqrt(c))*t(f)*P*tetad14;
create y14 from y14; append from y14; quit;
Proc iml;
use tetad15; read all into tetad15;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y15=(1/sqrt(c))*t(f)*P*tetad15;
create y15 from y15; append from y15; quit;
Proc iml;
use tetad16; read all into tetad16;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y16=(1/sqrt(c))*t(f)*P*tetad16;
create y16 from y16; append from y16; quit;
Proc iml;
use tetad17; read all into tetad17;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y17=(1/sqrt(c))*t(f)*P*tetad17;
create y17 from y17; append from y17; quit;
Proc iml;
use tetad18; read all into tetad18;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y18=(1/sqrt(c))*t(f)*P*tetad18;
create y18 from y18; append from y18; quit;
Proc iml;
use tetad19; read all into tetad19;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y19=(1/sqrt(c))*t(f)*P*tetad19;
create y19 from y19; append from y19; quit;
Proc iml;
use tetad20; read all into tetad20;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y20=(1/sqrt(c))*t(f)*P*tetad20;
create y20 from y20; append from y20; quit;
Proc iml;
use tetad21; read all into tetad21;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y21=(1/sqrt(c))*t(f)*P*tetad21;
create y21 from y21; append from y21; quit;
Proc iml;
use tetad22; read all into tetad22;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y22=(1/sqrt(c))*t(f)*P*tetad22;
create y22 from y22; append from y22; quit;
Proc iml;
use tetad23; read all into tetad23;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y23=(1/sqrt(c))*t(f)*P*tetad23;
create y23 from y23; append from y23; quit;
Proc iml;
use tetad24; read all into tetad24;
use P; read all into P;
use f; read all into f;
```

```
use c; read all into c;
y24=(1/sqrt(c))*t(f)*P*tetad24;
create y24 from y24; append from y24; quit;
Proc iml;
use tetad25; read all into tetad25;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y25=(1/sqrt(c))*t(f)*P*tetad25;
create y25 from y25; append from y25; quit;
Proc iml;
use tetad26; read all into tetad26;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y26=(1/sqrt(c))*t(f)*P*tetad26;
create y26 from y26; append from y26; quit;
Proc iml;
use tetad27; read all into tetad27;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y27=(1/sqrt(c))*t(f)*P*tetad27;
create y27 from y27; append from y27; quit;
Proc iml;
use tetad28; read all into tetad28;
use P; read all into P;
use f; read all into f;
use c; read all into c;
y28=(1/sqrt(c))*t(f)*P*tetad28;
create y28 from y28; append from y28; quit;
run;

Data z2;
set y1 y2 y3 y4 y5 y6 y7 y8 y9 y10 y11 y12 y13
y14 y15 y16 y17 y18 y19 y20 y21 y22 y23 y24
y25 y26 y27 y28;
run;

/** Cálculo da matriz: H2= c^(-0.5)*f'*P*X -
var. independentes do modelo transformado **/

Proc iml;
use xd1; read all into xd1;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x1=(1/sqrt(c))*t(f)*P*xd1;
create x1 from x1; append from x1; quit;
Proc iml;
use xd2; read all into xd2;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x2=(1/sqrt(c))*t(f)*P*xd2;
create x2 from x2; append from x2; quit;
Proc iml;
use xd3; read all into xd3;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x3=(1/sqrt(c))*t(f)*P*xd3;
create x3 from x3; append from x3; quit;
Proc iml;
use xd4; read all into xd4;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x4=(1/sqrt(c))*t(f)*P*xd4;
create x4 from x4; append from x4; quit;
Proc iml;
use xd5; read all into xd5;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x5=(1/sqrt(c))*t(f)*P*xd5;
create x5 from x5; append from x5; quit;
Proc iml;
use xd6; read all into xd6;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x6=(1/sqrt(c))*t(f)*P*xd6;
create x6 from x6; append from x6; quit;
Proc iml;
use xd7; read all into xd7;
use P; read all into P;
use f; read all into f;
```

```
use c; read all into c;
x7=(1/sqrt(c))*t(f)*P*xd7;
create x7 from x7; append from x7; quit;
Proc iml;
use xd8; read all into xd8;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x8=(1/sqrt(c))*t(f)*P*xd8;
create x8 from x8; append from x8; quit;
Proc iml;
use xd9; read all into xd9;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x9=(1/sqrt(c))*t(f)*P*xd9;
create x9 from x9; append from x9; quit;
Proc iml;
use xd10; read all into xd10;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x10=(1/sqrt(c))*t(f)*P*xd10;
create x10 from x10; append from x10; quit;
Proc iml;
use xd11; read all into xd11;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x11=(1/sqrt(c))*t(f)*P*xd11;
create x11 from x11; append from x11; quit;
Proc iml;
use xd12; read all into xd12;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x12=(1/sqrt(c))*t(f)*P*xd12;
create x12 from x12; append from x12; quit;
Proc iml;
use xd13; read all into xd13;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x13=(1/sqrt(c))*t(f)*P*xd13;
create x13 from x13; append from x13; quit;
Proc iml;
use xd14; read all into xd14;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x14=(1/sqrt(c))*t(f)*P*xd14;
create x14 from x14; append from x14; quit;
Proc iml;
use xd15; read all into xd15;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x15=(1/sqrt(c))*t(f)*P*xd15;
create x15 from x15; append from x15; quit;
Proc iml;
use xd16; read all into xd16;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x16=(1/sqrt(c))*t(f)*P*xd16;
create x16 from x16; append from x16; quit;
Proc iml;
use xd17; read all into xd17;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x17=(1/sqrt(c))*t(f)*P*xd17;
create x17 from x17; append from x17; quit;
Proc iml;
use xd18; read all into xd18;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x18=(1/sqrt(c))*t(f)*P*xd18;
create x18 from x18; append from x18; quit;
Proc iml;
use xd19; read all into xd19;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x19=(1/sqrt(c))*t(f)*P*xd19;
create x19 from x19; append from x19; quit;
Proc iml;
use xd20; read all into xd20;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x20=(1/sqrt(c))*t(f)*P*xd20;
create x20 from x20; append from x20; quit;
Proc iml;
use xd21; read all into xd21;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x21=(1/sqrt(c))*t(f)*P*xd21;
create x21 from x21; append from x21; quit;
Proc iml;
use xd22; read all into xd22;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x22=(1/sqrt(c))*t(f)*P*xd22;
create x22 from x22; append from x22; quit;
Proc iml;
use xd23; read all into xd23;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x23=(1/sqrt(c))*t(f)*P*xd23;
create x23 from x23; append from x23; quit;
Proc iml;
use xd24; read all into xd24;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x24=(1/sqrt(c))*t(f)*P*xd24;
create x24 from x24; append from x24; quit;
Proc iml;
use xd25; read all into xd25;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x25=(1/sqrt(c))*t(f)*P*xd25;
create x25 from x25; append from x25; quit;
Proc iml;
use xd26; read all into xd26;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x26=(1/sqrt(c))*t(f)*P*xd26;
create x26 from x26; append from x26; quit;
Proc iml;
use xd27; read all into xd27;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x27=(1/sqrt(c))*t(f)*P*xd27;
create x27 from x27; append from x27; quit;
Proc iml;
use xd28; read all into xd28;
use P; read all into P;
use f; read all into f;
use c; read all into c;
x28=(1/sqrt(c))*t(f)*P*xd28;
create x28 from x28; append from x28; quit;
run;

Data H2;
set x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 x13
x14 x15
x16 x17 x18 x19 x20 x21 x22 x23 x24 x25 x26
x27 x28;
run;

/** Regressão de z2(Y) sobre H2(X) **/

Data z2;
set z2;
rename col1=col0; run;

Data reg2;
merge z2 H2; run;

proc reg data=reg2 noprint;
        model col0=col1 col2 / noint;
        output out=rg2  residual=r2;
run; quit;

Data rg2;
set rg2;
qr2=r2*r2;
```

```
Keep qr2;
run;

Proc means data=rg2 sum noprint;
var qr2;
output out=rg2_ sum=sqr2;
run;

Data rg2_;
set rg2_;
keep sqr2; run;

/** Cálculo da variância sigma2_v **/

Proc iml;
use H2; read all into H2;
Ident=I(28);
PH2=Ident - H2*ginv(t(H2)*H2)*t(H2);    /*
Matriz P_H2 */
create PH2 from PH2; append from PH2; quit;

Proc iml;
use f; read all into f;
use P1; read all into P1;
use P2; read all into P2;
use P3; read all into P3;
use P4; read all into P4;
use P5; read all into P5;
use P6; read all into P6;
use P7; read all into P7;
use P8; read all into P8;
use P9; read all into P9;
use P10; read all into P10;
use P11; read all into P11;
use P12; read all into P12;
use P13; read all into P13;
use P14; read all into P14;
use P15; read all into P15;
use P16; read all into P16;
use P17; read all into P17;
use P18; read all into P18;
use P19; read all into P19;
use P20; read all into P20;
use P21; read all into P21;
use P22; read all into P22;
use P23; read all into P23;
use P24; read all into P24;
use P25; read all into P25;
use P26; read all into P26;
use P27; read all into P27;
use P28; read all into P28;
use c; read all into c;
cfp1=t(f)*P1*f*(1 / c);
cfp2=t(f)*P2*f*(1 / c);
cfp3=t(f)*P3*f*(1 / c);
cfp4=t(f)*P4*f*(1 / c);
cfp5=t(f)*P5*f*(1 / c);
cfp6=t(f)*P6*f*(1 / c);
cfp7=t(f)*P7*f*(1 / c);
cfp8=t(f)*P8*f*(1 / c);
cfp9=t(f)*P9*f*(1 / c);
cfp10=t(f)*P10*f*(1 / c);
cfp11=t(f)*P11*f*(1 / c);
cfp12=t(f)*P12*f*(1 / c);
cfp13=t(f)*P13*f*(1 / c);
cfp14=t(f)*P14*f*(1 / c);
cfp15=t(f)*P15*f*(1 / c);
cfp16=t(f)*P16*f*(1 / c);
cfp17=t(f)*P17*f*(1 / c);
cfp18=t(f)*P18*f*(1 / c);
cfp19=t(f)*P19*f*(1 / c);
cfp20=t(f)*P20*f*(1 / c);
cfp21=t(f)*P21*f*(1 / c);
cfp22=t(f)*P22*f*(1 / c);
cfp23=t(f)*P23*f*(1 / c);
cfp24=t(f)*P24*f*(1 / c);
cfp25=t(f)*P25*f*(1 / c);
cfp26=t(f)*P26*f*(1 / c);
cfp27=t(f)*P27*f*(1 / c);
cfp28=t(f)*P28*f*(1 / c);
cfp_=cfp1//cfp2//cfp3//cfp4//cfp5//cfp6//cfp7/
/cfp8//cfp9//cfp10//cfp11//cfp12//cfp13//cfp14
//cfp15//cfp16//cfp17//cfp18//cfp19//cfp20//cf
p21//cfp22//cfp23//cfp24//cfp25//cfp26//cfp27/
/cfp28;
cfp=diag(cfp_);
create cfp from cfp; append from cfp; quit;

/* *** Fim *** */

/* Obs: The function 'rank' in SAS IML does
not compute the rank of a matrix */

Proc iml;
use H2; read all into H2;
FF=H2*t(H2);
eval1=eigval(FF);
create eval1 from eval1; append from eval1;
quit;

Data id1;
do id=1 to 28;
output; end; run;

Data eval1;
merge id1 eval1;
run;

Data new1;
set eval1;
by id;
retain count 0;
if first.id then count=0;
if col1 > 0.000001 then count = count + 1;
run;

Proc means data=new1 sum noprint;
var count;
output out=caracH2 sum=caracH2;
run;

Data caracH2;
set caracH2;
keep caracH2;  /* Caracteristica de H2 */
run;

Proc iml;
use PH2; read all into PH2;
use cfp; read all into cfp;
use c; read all into c;
use rg2_; read all into rg2_;
use caracH2; read all into caracH2;
use varu; read all into varu;
use Binv; read all into Binv;
varv= (rg2_ -  trace(PH2*cfp) - varu*(28 -
caracH2))/(c*trace(PH2*Binv));
create varv from varv; append from varv; quit;

/* Termina aqui o cálculo da variância
sigma2_v */

/* Truncagem a zero das variâncias sigma2_u e
sigma2_v */

Data varu;
set varu;
sigma2u= max( col1 , 0);
keep sigma2u;
run;
Data varv;
set varv;
sigma2v= max( col1 , 0);
keep sigma2v;
run;

/*** Matriz GAMA  ***/

Data ro_1;
set dataro;
ro_1= 1-ro*ro;
keep ro_1;
run;
Data ro_2;
set dataro;
ro_2= ro*ro;
keep ro_2;
run;
Data ro_3;
set dataro;
ro_3= ro*ro*ro;
keep ro_3;
run;
Data ro_4;
set dataro;
ro_4= ro*ro*ro*ro;
keep ro_4;
```

```
run;
Data ro_5;
set dataro;
ro_5= ro*ro*ro*ro*ro;
keep ro_5;
run;
Data ro_6;
set dataro;
ro_6= ro*ro*ro*ro*ro*ro;
keep ro_6;
run;

Proc iml;
use ro_1; read all into ro_1;
use dataro; read all into dataro;
use ro_2; read all into ro_2;
use ro_3; read all into ro_3;
use ro_4; read all into ro_4;
use ro_5; read all into ro_5;
use ro_6; read all into ro_6;
I=I(7);
I1={0 1 0 0 0 0 0, 0 0 1 0 0 0 0, 0 0 0 1 0 0
0, 0 0 0 0 1 0 0, 0 0 0 0 0 1 0, 0 0 0 0 0 0
1, 0 0 0 0 0 0 0};
I2={0 0 1 0 0 0 0, 0 0 0 1 0 0 0, 0 0 0 0 1 0
0, 0 0 0 0 0 1 0, 0 0 0 0 0 0 1, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I3={0 0 0 1 0 0 0, 0 0 0 0 1 0 0, 0 0 0 0 0 1
0, 0 0 0 0 0 0 1, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I4={0 0 0 0 1 0 0, 0 0 0 0 0 1 0, 0 0 0 0 0 0
1, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I5={0 0 0 0 0 1 0, 0 0 0 0 0 0 1, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
I6={0 0 0 0 0 0 1, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0, 0 0 0 0 0 0 0, 0 0 0 0 0 0
0, 0 0 0 0 0 0 0};
Gama = I/ro_1 + dataro*I1/ro_1 +
t(dataro*I1/ro_1) + ro_2*I2/ro_1 +
t(ro_2*I2/ro_1) + ro_3*I3/ro_1 +
t(ro_3*I3/ro_1) + ro_4*I4/ro_1 +
t(ro_4*I4/ro_1) + ro_5*I5/ro_1 +
t(ro_5*I5/ro_1) + ro_6*I6/ro_1 +
t(ro_6*I6/ro_1);
create Gama from Gama;
append from Gama;
quit;

/*** Matriz G1 ***/

Proc iml;
use varv; read all into varv;
use Binv; read all into Binv;
G1=varv*Binv;
create G1 from G1; append from G1; quit;

/*** Matriz G2 ***/

Proc iml;
use varu; read all into varu;
use Gama; read all into Gama;
Ident=I(28);
G2=varu*(Ident@Gama);
create G2 from G2; append from G2; quit;

/*** Matriz G ***/

Proc iml;
use G1; read all into G1;
use G2; read all into G2;
Mat_GG=block (G1 , G2);
create Mat_GG from Mat_GG; append from Mat_GG;
quit;

Proc iml;
row=t(do(1,224,1));
create row from row; append from row; quit;

Data row;
set row;
rename col1=row;
run;

Data Mat_GG;
merge row Mat_GG;
run;

/* Construção do ficheiro com os parâmetros de
variância dos erros da sondagem */

Proc sort data=s1;
by i t;
run;

Data s2;
set s1;
keep var;
run;

Data par;
set s2;
rename var=est;
run;

/*********************************************
*********************************************/
/*** Fim da estimação das componentes da
variância com uma das amostras Bootstrap ***/
/*********************************************
*********************************************/

/*********************************************
*********************************************/
/** ESTIMAÇÃO DO MODELO DE LUIS PEREIRA
utilizando as estimativas das componentes de
variância Bootstrap ***/
/*********************************************
*********************************************/

Proc sort data=iabhipth1 out= cron;
by i t;
run;

Proc mixed data=cron MMEqSol noprofile
noclprint;
        class i t;
        model y=x / solution ddfm=residual;
        random i;
        random i*t/type=un Gdata=Mat_GG G V S;
        repeated /group=i*t;
        parms /parmsdata=par noiter;
make 'solutionf' out=beta;
make 'solutionr' out=u;
make 'V' out=Mat_V;
run; quit;

/*********************************************
*********************************************/
/* CÁLCULO DOS EFEITOS FIXOS utilizando as
estimativas das componentes de variância
Bootstrap ***/
/*********************************************
*********************************************/

/****** Efeitos fixos ******/

Data Effixo;
set beta;
keep estimate;
run;

/****** Efeitos Aleatórios ******/

Data u;
set u;
if estimate='.' then estimate=0;
keep estimate;
run;

/****** Vector CSI ******/

Proc iml;
use u; read all into u;
use Effixo; read all into Effixo;
csi=Effixo//u;
create csi from csi; append from csi; quit;

/****** Matriz G ******/

Data Matg;
set Mat_gg;
drop row;
run;
```

```
/****** Matriz V ******/

Data Matv;
set Mat_v;
drop index row;
run;

/****** Matriz C ******/

Proc iml;
use rest.X0; read all into X0;
use Matv; read all into Matv;
C= ginv( t(X0) * inv(Matv) * X0 );
create C from C; append from C; quit;

/****** Matriz A ******/

Proc iml;
use C; read all into C;
use rest.Q; read all into Q;
A=C* t(Q) * inv( Q * C * t(Q) );
create A from A; append from A; quit;

/* Estimativas EBLUP com restrições para uma
dada amostra bootstrap */

Proc iml;
use rest.X0; read all into X0;
use csi; read all into csi;
use A; read all into A;
use rest.q_; read all into q_;
use rest.Q; read all into Q;
EstRest=X0 * CSI + X0 * A *(q_ - Q*CSI );
create EstRest from EstRest; append from
EstRest; quit;

/* Nota: As estimativas produzidas estão
ordenadas da seguinte forma: domínio*tempo */

Data EstRest1;
merge rest.iabh EstRest;
run;

Data est_LP_rest_boot;
set EstRest1;
boot=&i;
rename col1=EstRest_boot;
keep i t boot col1;
run;

/***************************/
/****** Cálculo do g1 ******/
/***************************/

Proc iml;
use Matv; read all into Matv;
use Matg; read all into Matg;
use Rest.Z; read all into Z;
G1_aux=Z*(Matg -
Matg*t(Z)*inv(MatV)*Z*Matg)*t(Z);
create G1_aux from G1_aux;
append from G1_aux;
quit;

Proc iml;
use G1_aux;
read all into G1_aux;
G1=vecdiag(G1_aux);
create G1 from G1;
append from G1;
quit;

Data g1;
set g1;
rename col1=g1_boot;
run;

/******************************************/
/****** Cálculo do acréscimo de g1_R ******/
/******************************************/

Proc iml;
use rest.X0; read all into X0;
use A; read all into A;
use rest.Qv; read all into Qv;
use Matg; read all into Matg;
use rest.Z; read all into Z;
use Matv; read all into Matv;
g1_a1= X0 * A * Qv * Matg * t(Z) * inv(Matv) *
Z * Matg * t(Qv) * t(A) * t(X0) ;
create g1_a1 from g1_a1; append from g1_a1;
quit;

Proc iml;
use g1_a1; read all into g1_a1;
g1_aa1=vecdiag(g1_a1);
create g1_aa1 from g1_aa1; append from g1_aa1;
quit;

Data g1_aa1;
set g1_aa1;
rename col1=g1_aa1_boot;
run;

Proc iml;
use rest.X0; read all into X0;
use A; read all into A;
use rest.q_; read all into q_;
use rest.Xr; read all into Xr;
use Effixo; read all into Effixo;
g1_a2= X0 * A * (q_ - XR*Effixo) * t(q_ -
XR*Effixo) * t(A) * t(X0);
create g1_a2 from g1_a2; append from g1_a2;
quit;

Proc iml;
use g1_a2; read all into g1_a2;
g1_aa2=vecdiag(g1_a2);
create g1_aa2 from g1_aa2; append from g1_aa2;
quit;

Data g1_aa2;
set g1_aa2;
rename col1=g1_aa2_boot;
run;

Data est_LP_rest_boot;
merge Rest.Est_LP_rest_nutii est_LP_rest_boot
g1 g1_aa1 g1_aa2;
drop g1_aa1 g1_aa2 g1;
run;

Data est_LP_rest_boot;
set est_LP_rest_boot;
g1r_boot=g1_boot + g1_aa1_boot + g1_aa2_boot;
run;

Proc append base=Rest.Est_LP_rest_boot
data=est_LP_rest_boot;
run;

DM 'CLEAR OUTPUT';
DM 'CLEAR LOG';

%mend;
%inc 'C:\Documents and Settings\Luis Pereira
\My Documents \EstudoEmpirico\chamamacro.txt';

/*********************************/
/*** FIM DAS RÉPLICAS BOOTSTRAP ***/
/*********************************/

Data Est_LP_rest_boot;
Merge Rest.Est_LP_rest_boot;
M2=(EstRest_boot - EstRest)*(EstRest_boot -
EstRest);
M1=(g1r_boot - g1r);
keep i t M1 M2;
run;

Proc sort data=Est_LP_rest_boot;
by i t; run;

Proc means data=Est_LP_rest_boot mean;
var M1 M2;
by i t;
output out=Res1 mean=;
run;

Data Rest.EQMP_EBLUP_BOOT_LP_REST;
merge Rest.Est_LP_rest_nutii Res1;
by i t;
eqmp=g1r-M1+M2;
cv=sqrt(eqmp)/EstRest;
run;
```