Analysis of Incomplete Survey Data – Multiple Imputation via Bayesian Bootstrap Predictive Mean Matching

Dissertation

zur Erlangung des akademischen Grades

eines Doktors der Sozial- und Wirtschaftswissenschaften
(Dr. rer. pol.)

an der Fakultät Sozial- und Wirtschaftswissenschaften
der Otto-Friedrich-Universität Bamberg

vorgelegt von
Dipl.-Sozw. Florian Koller-Meinfelder
aus Nürnberg

Bamberg, August 2009

Datum der Disputation: 1. Oktober 2009

Prüfungskommission:
Prof. Dr. Susanne Rässler (Erstgutachterin)
Prof. Trivellore E. Raghunathan, PhD (Zweitgutachter)
Prof. Dr. Hans-Peter Blossfeld

"I never predict anything, and I never will."
Paul 'Gazza' Gascoigne

Thesis Advisors: Prof. Dr. Susanne Rässler   Prof. Trivellore E. Raghunathan, PhD

Analysis of Incomplete Survey Data – Multiple Imputation via Bayesian Bootstrap Predictive Mean Matching

Abstract

Missing data in survey-based data sets can occur for various reasons: sometimes they are created by design, sometimes they exist due to nonresponse. Multiple Imputation (Rubin 1978, 1987*a*) is a generally accepted method to allow for analysis of these incomplete data sets. The task of Multiple Imputation (MI) for survey data can be hampered by the sometimes very large number of variables. Another challenge for the imputer is that survey data sets typically consist of mixed variable types. The unifying aspect of all survey variable types is that their measurement is of a discrete nature, and we introduce several imputation algorithms that focus on this property of survey variables. Distributional assumptions play also an important role in most multiple imputation algorithms, so we examine the effect of relaxing these assumptions on imputation results for simulated data sets. When used in combination with models for continuous variables, Predictive Mean Matching (Rubin 1986, Little 1988*a*) shows desirable properties both regarding the task of imputing discrete data as well as giving robustness towards model misspecification.

Keywords: Multiple Imputation, Predictive Mean Matching, Mass Imputation, Bayesian Bootstrap, Fully Conditional Specification

Contents

Title page i

Abstract iii

Table of Contents iv

List of Figures x

List of Tables xii

Acknowledgments xv

Preface xvi

1 An introduction to missing-data problems 1

1.1 Assumptions for missing-data analysis 3

1.1.1 Missing-data mechanisms 3

1.1.2 Distinctness and ignorability 4

1.1.3 Observed-data likelihood 5

1.1.4 Empirical relevance of the ignorability assumption 6

1.2 Missingness patterns 7

1.2.1 Item nonresponse 7

1.2.2 Unit nonresponse 12

1.3 Simple approaches for incomplete-data analysis 13

1.3.1 Complete-Case and Available-Case analysis 13

1.3.2 (Conditional) mean imputation 14

1.4 Advanced approaches for incomplete-data analysis 16

1.4.1 Direct Maximum-Likelihood 16

1.4.2 Expectation-Maximization 17

1.4.3 Doubly-Robust estimation 20

2 Multiple Imputation 21

2.1 The Bayesian Framework of MI 21

2.2 MI confidence intervals 22

2.3 Markov Chain Monte Carlo techniques for Multiple Imputation . . . 24

2.3.1 Data Augmentation 25

2.3.2 Sampling Importance Resampling 26

2.3.3 The Bayesian Bootstrap 27

2.4 Joint-Modeling and Fully-Conditional Specifications 29

2.4.1 Joint-Modeling 29

2.4.2 Fully-Conditional Specifications 29

3 Standard Predictive Mean Matching and extensions 31

3.1 Predictive Mean Matching 31

3.2 PMM for block-wise missing data 33

3.3 PMM extension for binary and categorical variables 34

3.4 A Goodness-of-Fit Measure for the matching quality of nearest neighbor techniques 36

3.5 Benefits and Drawbacks of PMM 37

4 Multiple Imputation for metric-scale discrete data 40

4.1 Parametric MI variants 40

4.1.1 Rounding to the Nearest Observed Value 42

4.1.2 Inverse Probability Rounding 42

4.1.3 PRIMA - Predictive Imputation Matching 42

4.1.4 Semi-PRIMA 43

4.2 Semi-parametric variants based on Predictive Mean Matching 43

4.2.1 M nearest neighbors 43

4.2.2 Rounded Predictive Mean Matching 44

4.2.3 Posterior Predictive Mean Matching 44

4.2.4 Bayesian Bootstrap Predictive Mean Matching 45

4.3 Performance assessment 46

5 Multiple Imputation for metric-scale discrete data: an experimental design 48

5.1 Description of the experimental design 48

5.2 MI variants used for the experimental design 49

5.3 Simulation study 50

5.3.1 Objective 50

5.3.2 Data generation 50

5.3.3 Missing-data mechanisms 52

5.3.4 Multiple Imputation analysis 52

5.3.5 Computational consequences of the factorial design 53

5.4 Analysis of the imputed data 53

5.4.1 Abbreviations 54

5.4.2 Averaged effects 54

5.4.3 Best and worst performing method 56

5.4.4 Detail analysis 62

5.4.5 Analysis overview 64

5.5 Summary of results 66

6 A comparison of MI algorithms using simulated multi-variable data sets and real survey data 69

6.1 Imputation algorithm 70

6.1.1 Chained Equations regression 70

6.1.2 Multiple imputations via Bayesian Bootstrap parameter draws 73

6.2 Description of the empirical data set 73

6.3 IVEware 75

6.4 Analysis based on the original data set 77

6.4.1 Data preparation 77

6.4.2 Methods and quantities of interest 78

6.4.3 Results 79

6.4.4 Summary 82

6.5 Analysis based on Monte Carlo simulations 83

6.5.1 Design of the simulation study 83

6.5.2 Results 84

6.5.3 Summary 88

6.6 Discussion 89

7 BBPMMimpute: An *R* package for handling missing data in large survey data sets 90

7.1 Introduction 90

7.2 BBPMM.col – 'column-wise' Multiple Imputation 91

7.2.1 Description 91

7.2.2 Arguments 92

7.2.3 Values 92

7.2.4 Troubleshooting 93

7.2.5 Logical inconsistencies 93

7.3 BBPMM.row – 'row-wise' Multiple Imputation 94

7.3.1 Description 94

7.3.2 Arguments 96

7.3.3 Values 97

7.3.4 Troubleshooting 97

8 Concluding remarks 98

Appendices 101

A Diagnostic plots for the alcoholism study 101

A.1 Transformed variables 101

A.1.1 Histograms 101

A.1.2 Q-Q plots 103

A.2 Untransformed variables 104

A.2.1 Histograms 104

A.2.2 Q-Q plots 105

B Tables 106

B.1 Basic tables from the discrete-data MCMC experimental design . . . 106

B.1.1 Bias tables: five methods and ten quantities of interest 106

B.1.2 Coverage tables: five methods and ten quantities of interest . 111

B.2 Tables from the analysis of the alcoholism study 116

B.2.1 Original data set: proportion and mean estimates 116

B.2.2 Original data set: variance estimates 117

B.2.3 Original data set: model parameter estimates 118

B.2.4 Original data set: correlation estimates 119

B.2.5 Jackknife simulations: proportion and mean estimates 123

B.2.6 Jackknife simulations: model parameter estimates 124

C R code 126

C.1 BBPMM.row – Bayesian Bootstrap Predictive Mean Matching for missing-by-design patterns 126

C.2 BBPMM.col – Bayesian Bootstrap Predictive Mean Matching based on Sequential Regression 132

C.3 Additional functions 137

C.3.1 Bayesian Bootstrap 137

C.3.2 Model stabilizer for bootstrapped data 137

References **139**

List of Figures

1.1 Monotone missing-data pattern 8

1.2 Non-monotone missing-data pattern 8

1.3 Missing-by-design I: Data fusion pattern 9

1.4 Missing-by-design Ib: Data fusion pattern with unit nonresponse in both studies 10

1.5 Missing-by-design II: Split questionnaire survey design pattern . . . 11

1.6 Schematic display of unit nonresponse 12

3.1 Identification and imputation of a nearest neighbor with PMM 32

3.2 Observed and PMM-imputed values (grey) for truncated data (and re-estimated regression line) 38

4.1 Observed and PRIMA-imputed MCAR values (grey) for two discrete variables (only y partially incomplete) 47

6.1 Starting solution for Y_1 based on completely observed variables . . . 70

6.2 Sequential regression: Imputation of Y_1 based on completely observed and partially imputed variables 72

6.3 The variable COMPETE before and after transformation 77

6.4 Bivariate correlations and confidence intervals for all eight variables 82

6.5 Coverages for all three methods over 19 analysis quantities 86

A.1 Histogram: ALC_QF before and after transformation 101

A.2 Histogram: AQF2 before and after transformation 102

A.3 Histogram: COMPETE before and after transformation 102

A.4 Q-Q plot: ALC_QF before and after transformation 103

A.5 Q-Q plot: AQF2 before and after transformation 103

A.6 Q-Q plot: COMPETE before and after transformation 103

A.7 Histogram: AUDIT 104

A.8 Histogram: AUDIT2 104

A.9 Histogram: CONSEQNC 104

A.10 Histogram: DRK_DRIV 104

A.11 Q-Q plot: AUDIT 105

A.12 Q-Q plot: AUDIT2 105

A.13 Q-Q plot: CONSEQNC 105

A.14 Q-Q plot: DRK_DRIV 105

List of Tables

5.1 True values for all three data sets and quantities of interest 52

5.2 Average relative bias (in %) by various factors 55

5.3 Average coverage (in %) by various factors 55

5.4 best/worst performing method for relative bias (in %) for all 12 conditions 57

5.5 best/worst performing method for coverage (in %) for all 12 conditions 58

5.6 Totals for best/worst occurrences for relative bias and coverage (in %) 58

5.7 *best* and *worst* method for the big data set (1), MCAR 60

5.8 *best* and *worst* method for the small data set (3), MAR 61

5.9 Relative bias: DS1 MAR small 62

5.10 Relative bias: DS3 MAR BIG 63

5.11 Coverage: DS3 MAR BIG 64

6.1 Proportion and mean estimates based on the different methods . . . 79

6.2 Parameter estimates based on the different methods 81

6.3 True values for all proportions/means and model parameters 84

6.4 Analysis figures for the Jackknife simulations 85

6.5 CC: Average parameter estimates from 500 jackknife samples 87

6.6 BBPMMO: Average parameter estimates from 500 jackknife samples 87

6.7 IVEware: Average parameter estimates from 500 jackknife samples . 88

B.1 Bias: DS1 MAR BIG 106
B.2 Bias: DS1 MAR small 107
B.3 Bias: DS1 MCAR BIG 107
B.4 Bias: DS1 MCAR small 107
B.5 Bias: DS2 MAR BIG 108
B.6 Bias: DS2 MAR small 108
B.7 Bias: DS2 MCAR BIG 108
B.8 Bias: DS2 MCAR small 109
B.9 Bias: DS3 MAR BIG 109
B.10 Bias: DS3 MAR small 109
B.11 Bias: DS3 MCAR BIG 110
B.12 Bias: DS3 MCAR small 110
B.13 Coverage: DS1 MAR BIG 111
B.14 Coverage: DS1 MAR small 111
B.15 Coverage: DS1 MCAR BIG 112
B.16 Coverage: DS1 MCAR small 112
B.17 Coverage: DS2 MAR BIG 112
B.18 Coverage: DS2 MAR small 113
B.19 Coverage: DS2 MCAR BIG 113
B.20 Coverage: DS2 MCAR small 113
B.21 Coverage: DS3 MAR BIG 114
B.22 Coverage: DS3 MAR small 114
B.23 Coverage: DS3 MCAR BIG 114
B.24 Coverage: DS3 MCAR small 115
B.25 Proportions and means based on the different imputation methods . 116
B.26 Variances based on the different imputation methods 117
B.27 Regression parameters based on the different imputation methods . 118
B.28 Correlation estimates based on the complete cases 119

B.29 Correlation estimates based on BBPMMO 120

B.30 Correlation estimates based on BBPMMT 121

B.31 Correlation estimates based on IVEware 122

B.32 CC: Average mean and proportion estimates from 500 jackknife samples 123

B.33 BBPMMO: Average mean and proportion estimates from 500 jackknife samples 123

B.34 BBPMMT: Average mean and proportion estimates from 500 jackknife samples 123

B.35 IVEware: Average mean and proportion estimates from 500 jackknife samples 124

B.36 CC: Average parameter estimates from 500 jackknife samples 124

B.37 BBPMMO: Average parameter estimates from 500 jackknife samples 124

B.38 BBPMMT: Average parameter estimates from 500 jackknife samples . 125

B.39 IVEware: Average parameter estimates from 500 jackknife samples . 125

Acknowledgments

I would like to thank my advisors Susanne Rässler and Trivellore E. Raghunathan for their support. In a slight abuse of the German term *Doktorvater* (doctoral father), Susanne Rässler and Trivellore Raghunathan were to me like perfect 'doctoral parents'. They pushed me when needed, but they were also very indulgent when my 'other professional life' at the market research company GfK got the upper-hand of me, and I did not make any progress for months. Susanne Rässler got me interested in missing-data problems, when she supervised my diploma thesis on split questionnaire survey designs – largely influenced by Trivellore Raghunathan's JASA article in 1995. She introduced me also to Donald B. Rubin who invited me to do research at the Harvard Statistics department in 2004, and who I would also like to thank at this point. This stay – in combination with several subsequent trips to Trivellore Raghunathan and the Institute for Social Research (ISR) at the University of Michigan, Ann Arbor – was the springboard for this thesis.

Although my work for GfK was a considerable factor in prolonging the submission of this thesis, it has also been a time full of interesting problems and stimulating discussions with colleagues and friends.

Obviously without my 'real parents' I would not have submitted this thesis either (I would not have been at all to begin with). But thanks to them I was nurtured in an environment that helped to develop scientific curiousness and an affection to solving puzzles.

Last but not least I would like to express my deepest love and gratitude to my wife Rica who had been caring and patient with me all the time.

Altdorf, August 2009 Florian Koller-Meinfelder

Preface

A small fraction of the statistical community has focussed their academic research on statistical analysis of incomplete data. It turned out that quite a few statistical problems can be regarded as missing-data problems. And thus, from a small niche of statistical research matters, missing-data analysis has evolved to a ‘bridge-builder’, contributing new insights to data fusion (file matching) or post stratification and weighting. Other fields in Statistics, where the connection is not as obvious as in the above cases, have been strongly influenced by perceiving the particular tasks as missing-data problems, too. Among the more prominent examples are ’confidentiality’ and ’causal inference’ – the latter one incorporates missingness through the concept of ’potential outcomes’ which bears close resemblance to Quantum Physics’ ’parallel universes’[1], or Philosophy’s ’possible worlds’.

We feel that distinguishing data in *observed* and *not observed* or *missing* (or somesometimes ’not observ*able*’) would help to prevent people from drawing hasty conclusions. One example: Many of you will have read at some point (maybe even in some zoological textbook) that the Blue Whale is the largest animal to have ever lived – just because no one has found the remains of a larger species, not even among the dinosaurs (at least that was the status 10 years ago). Meanwhile palaeontologists have found fossils of dinosaurs which very likely exceeded the Blue Whale in terms of length[2], albeit maybe not in terms of weight. It is careless to mistake ’present status of knowledge’ for ’certainty’. We probably only know of a very small fraction of all dinosaurs that have ever existed, but the sample of fossils already suggested that dinosaurs could have evolved species that were probably larger than any living species.

[1] ’Schröedinger’s cat’ is a thought experiment about ’potential outcomes’.

[2] Another missing-data problem: usually palaeontologists do not find complete skeletons, but only skeletal parts of a dinosaur. In one particular case, some petrified vertebrae, ribs, a shinbone and an incomplete femur of a dinosaur belonging to the Sauropod family was found in Patagonia, and by comparing the size of these bones with the corresponding ones of a fairly complete skeleton of a *Titanosaur*, another Sauropod, the overall size was extrapolated. The name of this dinosaur is *Argentinosaurus*, and its length is estimated to range from 30-35 meters. There are some other species, such as *Bruhathkayosaurus* or *Amphicoelias*, who were probably even larger, but their fossil record is even scarcer.

We could provide more illustrative examples from other fields, where missing data lead to erroneous conclusions, but the purpose of this example was merely to underline the importance of 1) *awareness*, if an analysis is based on partially unobserved data, 2) *requirement for assumptions*, one has to make in such a case, and 3) *remedies* for statistical analysis of incomplete data. Careful scientists implicitly or explicitly make these assumptions, and the above largest-animal-example is more symptomatic for the stage when research finds its way into ’popular science’, and from there into ’popular knowledge’.[3]

There are several solutions to handle missing values in data sets, ranging from partially or completely erasing incomplete cases to ’filling-in’ the gaps. The latter approach is known as *imputation*, and can be used for general-purpose analysis, if the imputation is carried out in a ’sensible’ way. We further distinguish between single and multiple imputation (MI) approaches which incorporate additional uncertainty created by the imputation themselves. The rough idea behind MI is that we do not want to treat an imputed value, as if it had been observed, therefore we make several ’guesses’ about it, and combine our beliefs. This principle has become a generally accepted way of sensibly dealing with missing data – in particular, if *imputer* and *data analyst* are not the same person).

Multiple Imputation in its original form relies on distributional assumptions incorporated in a Bayesian model framework. But ’sensible’ applicability to empirical data is occasionally in doubt, if the data do not resemble any theoretical statistical distributions. If the missing data are assumed to be *missing at random* – more about this to follow in section 1.1.1 – MI should yield unbiased estimates for any quantity of interest. But what if the imputation algorithm creates (additional) bias because it uses a misspecified imputation model? In such a case the cure might be worse than the disease. The imputation algorithms described in this thesis are evaluated by their robustness to imputation model misspecification. The tests in chapter 6 include missingness in empirical survey data sets, as these kinds of data are notorious for both: nonresponse and ’ugly’ data distributions.

[3] You can just imagine to pop up the question about the largest animal to have ever lived at *Who Wants to be a Millionaire?* – with the Blue Whale being the ’correct’ answer.

1

An introduction to missing-data problems

Multiple Imputation (Rubin 1978, 1987*a*) has become a generally accepted way to handle statistical analysis of incomplete data, and is the central theme of this thesis. A large part of the research on Multiple Imputation (MI) has focused on fully parametric variants with underlying distributional assumptions. However, surveys often yield mixed-scale data, and the variables do not resemble statistical distributions. Some multivariate approaches try to model continuous data within cell combinations of categorical variables. Schafer (1997) suggests to impose restrictions on those cell combinations via log-linear models, but usually the large number of variables and the potential number of combinations impede us from getting stable and computationally feasible solutions. Another approach is to impute variables sequentially, by conditioning the imputation of each variable on all other variables. These approaches are labeled as 'chained equation', 'sequential regression', or 'fully conditional' approaches, and can be applied to large data set, since only (univariate) multiple regressions instead of multivariate computations are needed. MI packages such as IVEware (Raghunathan et al. 2002) or MICE (van Buuren & Oudshoorn 1999) are using this algorithm[1], and MICE also features an optional semi-parametric approach which is called *Predictive Mean Matching* (PMM). In addition to MI this approach also plays a major role in the following

[1] 'MICE' is the abbreviation of 'Multiple Imputation by Chained Equations'.

chapters. PMM is known to have robustifying properties to model misspecification that are useful for imputing survey data. The big task is to integrate PMM into an MI algorithm – without biasing MI inference. In chapter 5 we investigate, if and under which conditions, various MI methods for discrete data yield unbiased MI estimates and variances. The 'winner' is then implemented into a sequential regression algorithm, and its performance is tested by comparing MI results with those from IVEware (see chapter 6). As already mentioned, other MI algorithms also feature *Predictive Mean Matching*– but only for metric-scale variables. One innovation of this thesis is the development of a PMM variant for (unordered) categorical variables, with a straightforward extension to MI via the Bayesian Bootstrap (Rubin 1981).

Before we approach the theoretical framework of Multiple Imputation in chapter 2, it makes sense to become acquainted with some underlying basic assumptions first, and to review other methods of incomplete-data analysis. By addressing theoretical implications and how they relate to various incomplete-data analysis techniques, we hope to generate an understanding for the benefits of 'advanced approaches' in general, and Multiple Imputation in particular.

We will start by introducing some fundamental theoretical assumptions that play also an important role for MI, followed by a short description of various missing-data patterns. These patterns, and its implications, will be re-visited at different points of this thesis. Since we feel it helps to appreciate the merits of Multiple Imputation if we re-cap methods to analyze incomplete data that have been around long before MI, but which are severely flawed[2], we will give a brief overview over these 'simple approaches'. Finally, the introductory chapter closes with a description of a group of incomplete-data analysis methods that can be described as 'advanced approaches'.

[2] Nevertheless, most of them are included in statistical software packages such as SAS, SPSS, or STATA.

1.1 Assumptions for missing-data analysis

This section introduces the – meanwhile standard – categorization of missing-data mechanisms that can be traced back to the terms ′missing at random′ (MAR) and ′observed at random′ in Rubin (1976). In the same article Rubin introduced a second necessary assumption: The parameters governing the missing-data mechanism and the parameters governing the analysis parameter have to be ′distinct′. If both, ′MAR′ and ′distinctness′ hold, the missing-data mechanism is said to be ′ignorable′ (see also Little & Rubin 2002).

1.1.1 Missing-data mechanisms

Let $\mathbf{Y} = [\mathbf{Y_{obs}}, \mathbf{Y_{mis}}]$ be an $n \times p$ data matrix consisting of an observed part $\mathbf{Y_{obs}}$ and an unobserved (missing) part $\mathbf{Y_{mis}}$. Furthermore, let $\mathbf{R} = [r_{ij}]$ ($i = 1, \ldots, n$ and $j = 1, \ldots, p$) define an indicator matrix, where $r_{ij} = 1$ if y_{ij} is missing, and $r_{ij} = 0$ if y_{ij} is observed. By treating r_{ij} as value of a random variable with an underlying distribution, we can formalize mechanisms that generate missing data. The categorization starts from the conditional distribution $f(\mathbf{R}|\mathbf{Y}, \psi)$, where ψ describes unknown parameters. Then, if missingness does not depend on the values of $\mathbf{Y}$ at all, the conditional distribution is reduced to

$$f(\mathbf{R}|\mathbf{Y}, \psi) = f(\mathbf{R}|\psi) \;\; \forall \;\; \mathbf{Y}, \psi. \tag{1.1}$$

We refer to this case as *Missing Completely at Random* or MCAR. To illustrate this with an example from Survey Methodology, suppose some variables from an interview were accidentally not transcribed during the coding process, i.e. some values of a ′row vector′ in the data set are missing. Analogously, several observations of a variable will be missing, if this happened with several interview codings. The mechanism that led to missing observations for this variable can be considered to be purely stochastic.

In another situation R depends on $\mathbf{Y}$, but only on the observed part Y_{obs}, so that

$$f(\mathbf{R}|\mathbf{Y},\psi) = f(\mathbf{R}|Y_{obs},\psi) \ \forall \ Y_{mis},\psi \tag{1.2}$$

leads to a less restrictive assumption regarding R. This mechanism is called *Missing at Random* (MAR) and plays an important role in all statistical analysis with missing data. It has also found its way into design-based concepts like post-stratification.

Let us consider another example from Survey Methodology to illustrate the difference between MCAR and MAR: Suppose we have missing values for the variable 'personal net income', and suppose further that 'age' and 'personal net income' are not independent. If, say, older people had a higher likelihood of refusing to answer questions about their income, the observed subsample and the (virtual) full sample would have different income distributions. MAR means that conditioning on 'age' would make the missing-data mechanism of 'net income' random. Note that MAR does *not* mean that $\mathbf{Y_{mis}}$ must be perfectly explained by $\mathbf{Y_{obs}}$. In the given example it is not required to perfectly explain 'personal net income' by other variables. Merely the conditional distribution of $\mathbf{R}$ needs to be independent of $\mathbf{Y_{mis}}$. If this is not the case, and the conditional distribution of $\mathbf{R}$ is not independent of $\mathbf{Y_{mis}}$, we describe the missingness mechanism as *Not Missing at Random* or NMAR. To stretch our survey example a bit more, let us assume that missing values depend now directly on the variable 'personal net income', for instance, if respondents with high earnings had a lower propensity to answer questions about their income.

1.1.2 Distinctness and ignorability

Another important property of the aforementioned mechanisms is the concept of *distinctness* between the unknown parameters θ that govern the distribution of $\mathbf{Y}$ and ψ. From the perspective of a Bayesian statistician this means that the joint prior distribution can be split into the product of the marginal prior distributions[3],

$$\pi(\theta, \psi) = \pi(\theta)\pi(\psi). \tag{1.3}$$

Distinctness therefore assumes that the parameters of the ′analyst′s scope′, and the parameters of the ′imputer′s scope′ are not related to each other. This implies that for estimating θ, we do not have to model the missing-data mechanism. The missing-data mechanism is said to be *ignorable*, if *MAR* and *distinctness* hold (see Little & Rubin 2002).

1.1.3 Observed-data likelihood

Applying the MAR and distinctness assumptions allows us to re-write the observed-data likelihood as

$$\begin{aligned} f(\mathbf{Y_{obs}}, \mathbf{R}|\theta, \psi) &= \int f(\mathbf{R}|\mathbf{Y}, \psi) f(\mathbf{Y}|\theta)\, d\mathbf{Y_{mis}} \\ &= f(\mathbf{R}|\mathbf{Y_{obs}}, \psi) \int f(\mathbf{Y}|\theta)\, d\mathbf{Y_{mis}} \\ &= f(\mathbf{R}|\mathbf{Y_{obs}}, \psi) f(\mathbf{Y_{obs}}|\theta) \end{aligned} \tag{1.4}$$

Schafer (1997) explicitly mentions that without the MAR assumption, we would not be able to factorize the terms this way, since θ originally pertained to the *complete-data* model parameters. The factorization illustrates, how the assumption of distinction allows us to proceed from here: since we are interested in inferences about θ, we can ′drop′ the first factor, and (1.4) becomes

$$f(\mathbf{Y_{obs}}|\theta) \equiv L(\theta; \mathbf{Y_{obs}}) \tag{1.5}$$

[3] *Distinctness* from the ′Frequentist′ perspective means that the joint parameter space of θ and ψ is the joint Cartesian cross-product of the single parameter spaces, as pointed out by Schafer (1997). The different definitions are necessary, because ′Frequentists′ – unlike ′Bayesians′ – do not treat parameters as random variables.

Note that without the ignorability assumption, we would not be allowed to simply us the right-hand-side of (1.5), as the observed-data likelihood consists of $f(\mathbf{Y_{obs}}$ *and* $\mathbf{R}$ (Schafer 1997).

Little & Rubin (2002) label the right-hand-side of (1.4) as 'full [observed-data] likelihood', and (1.5) as 'simpler [observed-data] likelihood that ignores the missing-data mechanism'. For reasons of simplification we will from now on refer to this term as 'observed-data likelihood'.[4]

1.1.4 Empirical relevance of the ignorability assumption

Distinctness is an assumption that is usually rather intuitive: Why should the model parameters carry any information about the parameters governing the missing-data mechanism?

But MCAR, MAR and NMAR are assumptions the imputer/analyst has to make about the underlying mechanism of missing data. MCAR can be tested (see Little 1988*b*, Chen & Little 1999), and MCAR and MAR can be compared to each other (Heitjan & Basu 1996). Unfortunately there is no way to test for NMAR. Some literature proposes solutions if the missing-data mechanism is not ignorable, but information is available about the mechanism (e.g. Herring et al. 2004), or by making additional assumptions regarding the data (Tang et al. 2003).

In a situation, where the 'imputer' has no substantial information about the missing-data mechanism (it might be NMAR), but the 'analyst' still wants to draw inference about θ, it is better to assume *ignorability*, and to carry out an incomplete-data analysis, than to *ignore* the fact that not assuming anything, and to simply analyze the observed cases, is by far the worst option. The rationale behind is that any bias created by a non-ignorable mechanism can at least be attenuated by making use of $\mathbf{Y_{obs}}$ as well as possible – unless each variable with missing values is completely independent of all other variables in the data.

[4] The more precise definition can be replaced, as we will not discuss non-ignorable missing-data mechanisms.

1.2 Missingness patterns

The most frequent case of a missing-data problem a statistician will encounter, is a data situation, where single values of an otherwise rectangular data set are missing. In Survey Methodology, this would mainly be due to item nonresponse, in Biology it could be due to contaminated samples, and in Meteorology it might be due to malfunctioning mercuries.

Alternatively, a whole observation could be missing from the sample – an event that is mainly a problem in survey data settings.

Since the research leading to this thesis was motivated by problems encountered in studies with a social scientific context, we will focus in the following on examples taken from the survey cosmos, and we will refer to the first case as *item nonresponse* and to the second case as *unit nonresponse*.[5]

1.2.1 Item nonresponse

Monotone and non-monotone patterns

Wherever the incomplete data set described above has come from, we can distinguish the missingness pattern between *monotone* and *non-monotone* (see Little & Rubin 2002). Monotone missingness patterns must fulfill the following condition: We re-arrange the order of every variable Y_j, with $j = 1, ..., p$, in a data set sorted by the percentage of missing data, starting with the variable that has the smallest percentage of missing values, such that $n_{mis}(Y_{(1)}) \leq n_{mis}(Y_{(2)}) \leq \ldots \leq n_{mis}(Y_{(p)})$. A monotone missingness pattern is given, if the observed cases of $Y_{(j)}$ are a subset of the observed cases of $Y_{(j-1)}$ (see fig. 1.1).

A monotone missingness pattern has some desirable properties, e.g. the imputation of variable $Y_{(j)}$ is fully conditional on all completely observed variables and

[5] Although, technically some examples for missing information are not really caused by nonresponse.

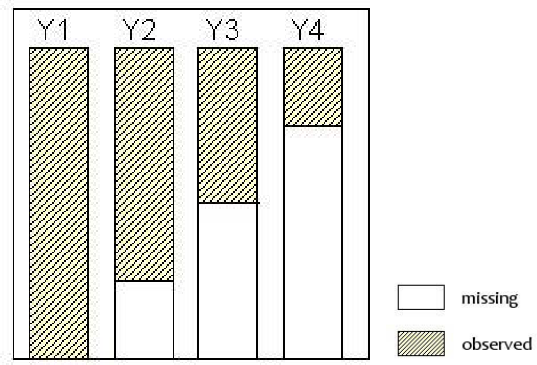

Figure 1.1: Monotone missing-data pattern

all variables $Y_{(1)}$ to $Y_{(j-1)}$. The most simple form of a monotone pattern is a data set, where only one variable is not completely observed.

Much more frequently, however, the missingness pattern will be non-monotone, i.e. we can not re-order the incomplete variables $Y_1, ..., Y_p$, such that the observed cases of Y_j are a subset of the observed cases of Y_{j-1} (see fig. 1.2).

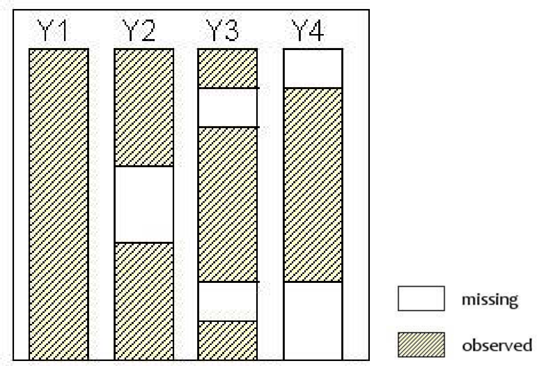

Figure 1.2: Non-monotone missing-data pattern

Basically, all missing-data patterns can be classified into either of these two meta-classes.

Data fusion design

A data fusion (Rässler 2002, Rodgers 1984) or statistical matching problem *is* a missing-data problem with a non-monotone missingness pattern, which we can clearly see when we stack the data files, such that there is only complete information on all joint variables (see fig. 1.3).

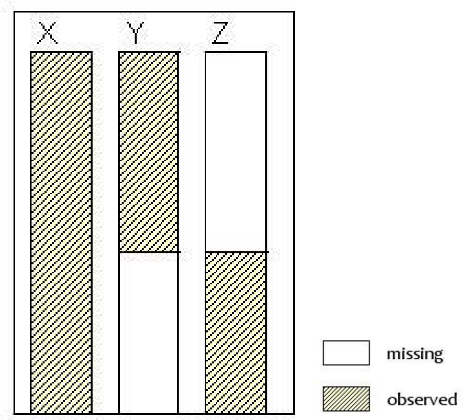

Figure 1.3: Missing-by-design I: Data fusion pattern

While various nearest neighbor matching techniques have become the predominant approach for statistical matching (see e.g. Rubin 1986), analysis is also possible via fully parametric imputation approaches (e.g. Kamakura & Wedel 1997). Because we actually created this missingness pattern, it is often described as a missing-by-design pattern. But a data fusion pattern is even more problematical than a ’standard’ non-monotone pattern, because we typically want to analyze variables which were never jointly observed. The inherent identification problem in statistical matching requires a conditional independence assumption between those variables that were not jointly observed given the joint variables (see Rässler 2002).

Data fusion faces another problem, if both studies suffer from unit nonresponse caused by different missing-data mechanisms. In this case, stacking the data leads to a missing-by-design pattern, but the missing-data mechanism is *not* MCAR, because there is another underlying missing-data problem. Figure 1.4 displays the different causes for missingness.

Some of the (more naïve) ’validation tests’ for data fusion compare (marginal) distributions of variables in the donor study with their counterparts from the fused

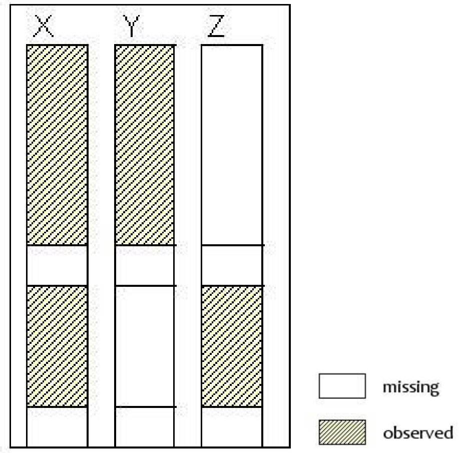

Figure 1.4: Missing-by-design Ib: Data fusion pattern with unit nonresponse in both studies

recipient study (Rässler 2002).[6] These tests are clearly rendered useless, if different unit nonresponse mechanisms occurred in the involved studies. If we treat the stacked studies as one incomplete data set (with *item* nonresponse), and apply the tests mentioned in section 1.1.4, we could probably reject that the missing-data mechanism is MCAR. But if we assume conditional independence we believe that it is still ignorable, as conditional independence should encase *ignorability*.

Split questionnaire design

Another missing-by-design pattern tries to avoid data fusion's identification problem: Split questionnaire survey designs or Multiple Matrix Sampling aim to reduce the response burden by grouping variables into components and administering only a selection of the total number of components[7] to every respondent (Raghunathan & Grizzle 1995, Gelman et al. 1998, Neal et al. 2006, Rassler et al. 2002). The key principle is to preserve a joint distribution of the observed-data for all variables that are to be analyzed jointly. In figure 1.5 we have assumed that later anal-

[6] Although the missing-data pattern suggests that both 'blocks' could be imputed, typically only one of the variable sets is imputed. Apart from the traditional distinction between 'donor' and 'recipient' study, this is also the case, if tailor-made software coerces the imputer to work with the predetermined case identifiers of the recipient study.

[7] Plus a core component that is always part of the questionnaire.

ysis includes bivariate associations, but no higher-order interactions. Therefore, for every bivariate combination, a reduced subsample remains observed, yielding $\binom{4}{2} = 6$ different missing-data patterns.

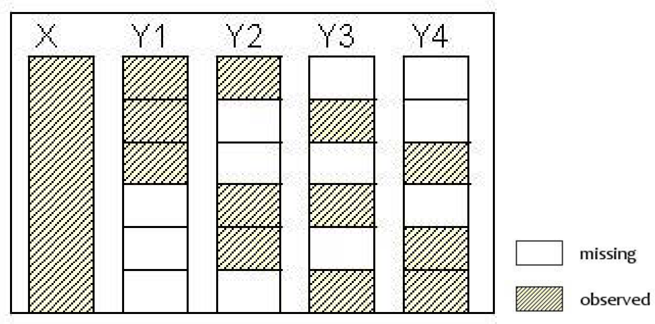

Figure 1.5: Missing-by-design II: Split questionnaire survey design pattern

The intentional creation of a specific missingness pattern also touches an issue that has become the focus of attention among imputers: The distinction between the 'analyst's model' and the 'imputer's model'. An interesting situation arises, if the analyst wants to run some regression model, and parts of $\mathbf{X}$ (the regressor variables) are missing. One might point out that Y (the analysis variable) should in these circumstances *not* be used for imputation, because the outcome variable would be used to impute the missing information in the model variables. But leaving out the dependent variable from the imputer's model would imply (conditional) independence, and lead to biased inferences (see Allison 2001). Note that at the imputation stage we are not interested in causality, but only in the preservation of (complete-data) multivariate associations.

Special cases of incomplete-data

A very special case of missing information are variables that are not as precisely observed as they could be. When interviewers require rather confidential information (like income) from the respondent, a bracket or class is offered as response option, and the precise income has to be imputed. Heitjan & Rubin (1990, 1991), Heeringa et al. (2002) describe this case as coarsened data. Little & Rubin (2002) also treat the outcome of multivariate methods, which imply an underlying latent variable, as a case of missing information.

1.2.2 Unit nonresponse

Unit nonresponse is a different problem, as it does not 'show'directly in the analyst's data set. In survey data we would have direct information about it only, if failed interview attempts were recorded at the data preparation step. If this information is not available, we could only compare sample structures with external information from a census study or another source that directly refers to the population the sample is drawn from. And then, there would still be uncertainty if differences are genuine sample bias or stochastic effects, whether nonresponse or the sampling scheme is responsible.

In some situations it is however possible, to gather some information – even in the case of nonresponse: In a study with personal contact between interviewer and respondent, unit nonresponse can be transformed into item nonresponse, if we take into account all the information[8] that can be gathered without compliance of the respondent.

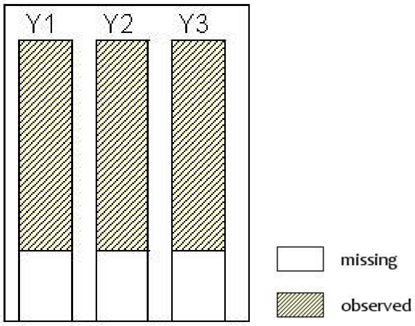

Figure 1.6: Schematic display of unit nonresponse

[8] E.g. living area, interviewer-estimated wealth of the household, etc.

1.3 Simple approaches for incomplete-data analysis

Note that, whatever we do to account for missing data, we apply some model based on assumptions. This is even true, if we decide to make no adjustment at all to account for missingness in the data.

In the following we will briefly discuss some methods that are ubiquitous, because their application is quite simple. Unfortunately, they either lead to estimators that are at best not *efficient*, and at worst lead to biased inferences.

There exists a variety of further methods that can also be classified as *simple methods* such as 'dummy-variable adjustment', 'cold deck' and (simple) 'hot deck', or 'raking' which we will not discuss here.

1.3.1 Complete-Case and Available-Case analysis

These two methods are also known as *listwise deletion* and *pairwise deletion*, and represent two variants of doing (almost) nothing about missing data. The terms also hint, what data analysts had been doing for a long time, when being confronted with incomplete data sets.

Complete-Case analysis

Complete-Case analysis (CC) is the simplest way of dealing with incomplete data: all missing cases of variables which are needed to estimate some quantity of interest, are deleted. The analysis is then carried out on the remaining cases. This 'strategy' of dealing with incomplete data has already been mentioned in section 1.1.4. Most users of listwise deletion are not aware that for (marginal) mean estimates they implicitly assume the missing-data mechanism to be MCAR. The more variables are involved in the analysis, the more likely it is that a substantial amount of observed data is not used for the analysis as well, since every unit with missing values for at least one variable is completely excluded, even if the remaining variables are observed. In summary, estimators based on listwise deletion often rely on the much stronger MCAR assumption while never being *efficient*. A notable exception, where listwise deletion under MAR does yield unbiased estimates are regression model estimates (see Little 1992). However, if the regression model is not correctly specified, and if variables or parameterizations are missing that are related to Y and the missing-data mechanism, the regression parameter estimates are not only biased, but would also be different to the estimates which we would have obtained from a completely observed data set.

Available-Case analysis

Available-Case analysis (AC) is a slightly more elaborate version of Complete-Case Analysis: Available information is used in a more efficient way, by computing summary statistics that can be used to estimate the actual quantity of interest. For instance, the main diagonal of an AC-estimated covariance matrix of p variables contains variance estimates based on the complete cases of *each* variable (in comparison: A CC-estimated covariance matrix would only use cases which are complete for all p variables).

According to Allison (2001) regression model estimates under MAR based on Available-Case analysis can be seriously biased (unlike model estimates based on Complete-Case analysis). Another downside of AC is that there is no general agreement on how to implement the method: a mean estimate $\hat{\mu}_x$ that is used to estimate the covariance $\sigma^2_{x,y}$ could be based on all complete cases for variable X or on the complete cases for X and Y.

1.3.2 (Conditional) mean imputation

Available-Case analysis tries to ‘wriggle’ its way around the missing parts of the data. An alternative is to 'fill-in' gaps in the data. Since we do not know what the values in the gaps really look like, we make *imputations* about them. This expression has become the predominant term for any technique that aims to produce a complete(d) rectangular data set.

Marginal mean imputation

Marginal mean imputation is the most basic form of imputation. Any missing value is simply replaced by the marginal mean estimate of the remaining cases for this variable. Two example shall briefly show the short-comings of this approach: Suppose we want to estimate the mean of a variable, where parts of the data are MCAR. Where listwise deletion would give us correct[9], albeit inefficient confidence intervals, mean imputation yields confidence intervals that are too narrow, due to an artificially inflated sample size. In a second example let X be a completely observed variable and Y a variable, where cases $1, \dots, n_{mis}$ are missing. Let us further assume that we want to estimate the correlation $\rho_{x,y}$ between the two variables. The marginal mean imputation step introduces a severe bias to $\hat{\rho}_{x,y}$: The variance estimates of Y is zero for the first n_{mis} cases. But this also means that the covariance between X and Y is zero for these case, leading to an estimate for $\rho_{x,y}$ that is also biased towards zero.

Conditional mean imputation

In contrast to marginal mean imputation, conditional mean or regression imputation is not 'careless' about bi- or multivariate associations, but 'over-caring'. Missing values are imputed via (generalized linear) regression models, but since all imputed values are on the regression line, the explained sum of squares is inflated. In the case of a linear regression imputation this means that the correlations between the imputed variable Y and the (completely observed) imputation model variables are biased away from zero.

Generalizations of regression imputation (iteratively) using weighted least squares yield better estimates, but the fundamental problem of underestimated standard errors remains.

[9] If we treat the complete cases as a 'new' sample.

1.4 Advanced approaches for incomplete-data analysis

In the following we will mainly discuss two methods that focus on the observed-data likelihoods. Direct ML methods relate to pairwise deletion, because they try to get MLE's without imputation from an incomplete data set. These methods are straightforward to implement, if the missing-data pattern is monotone, but computationally tricky otherwise.

Another method is the so-called (E)xpectation-(M)aximization algorithm (Dempster, Laird & Rubin 1977), which can be viewed as a two-step procedure, where one step consists of maximizing a set of parameters θ of the likelihood given the (completed) data, and the other step consists of replacing missing values by expectational values given the current estimates $\hat{\theta}$. The steps are iterated until some convergence criterion is fulfilled.

A rather new approach is the concept of 'doubly robust' (DR) estimators (introduced in the discussion from Scharfstein et al. 1994). Its authors claim that DR can have certain advantages over Multiple Imputation, if the data model is not correctly specified. We will critically discuss this concept, since MI under misspecified imputation models was also a primary research motivation of this thesis.

1.4.1 Direct Maximum-Likelihood

Let $\mathbf{Y} = [\mathbf{Y_{obs}}, \mathbf{Y_{obs}}]$ be an incomplete $n \times p$ data matrix, consisting of an observed part $\mathbf{Y_{obs}}$ and a missing part $\mathbf{Y_{mis}}$. Furthermore, let θ be the parameters belonging to the complete-data likelihood. The observed-data log-likelihood function can then be expressed as

$$\ell(\theta; \mathbf{Y_{obs}}) = \sum_{i=1}^{n} \ln f(\mathbf{Y_{obs,i}} | \theta).$$

Solving this function is feasible, but often computationally extremely expensive.

However, as Allison (2001) points out, there is specialized software available – mainly for Structural Equation Models – that can estimate these functions.

Monotone missing-data pattern are a special case, because here the observed-data likelihood can be factored into parts, such that the most-complete part is conditioned on the completely observed variables of $\mathbf{Y_{obs}}$, the second-most-complete part is conditioned on the completely observed variables and the most-complete part, and so on, until the least-complete part is conditioned on all other parts of $\mathbf{Y_{obs}}$. Little & Rubin (2002) write that Anderson (1957) was the first to factor likelihoods (for normally distributed data) that way. The advantage of expressing the likelihoods as conditional on the more-observed parts is that each part can be maximized separately which makes the derivation of the observed-data likelihood much more tractable.

In theory, Direct ML yields efficient and (asymptotically) unbiased estimates, but it requires a joint distribution for all variables with missing data (Allison 2001). This also means that the more variables with missing data and the more different missing-data patterns exist, the harder the computation will be.

1.4.2 Expectation-Maximization

The key idea behind the Expectation-Maximization (EM) algorithm is so intuitive that first applications can be traced back to 1926 (Little & Rubin 2002): Estimate a set of parameters, subsequently derive a statistical distribution out of it, re-estimate the set of parameters again, and so on. Hartley (1958) applied such an algorithm to incomplete categorical data, and Baum & Petrie (1966) implemented it into a Markov model. But whenever the EM algorithm is cited, the first reference is usually the landmark article by Dempster, Laird & Rubin (1977), who not only gave the algorithm its name, but also proved several key theorems, explored the full generality of EM, and gave a large variety of examples. For this reason, the 'DLR' paper deserves the credits it has been awarded.[10]

[10] Stephen Jay Gould, a former Harvard professor of Geology and Biology, who became well-known for his popular science books, wrote in an essay (Gould 1985) on the origins of the evolution

The EM algorithm can be used, if the maximum of a likelihood is hard or impossible to find analytically (for instance, it is often used for hidden Markov models), but it is predominantly applied to missing-data problems. As mentioned in the previous section, if the missing-data pattern is monotone or stems from a data fusion, the observed-data likelihood can be factorized.[11] But if the missing-data pattern is non-monotone the observed-data likelihood $L(\theta; \mathbf{Y_{obs}})$ typically consists of many different and complicated functions. In such a case the EM algorithm is defined by using an expectational value for the missing data and to subsequently maximize the likelihood for the parameters of interest θ, and to iterate the two steps, until convergence has been achieved.

Let us factor the complete-data distribution of $\mathbf{Y}$ into the observed-data likelihood and the *conditional predictive distribution* of $\mathbf{Y_{mis}}$,

$$f(\mathbf{Y}|\theta) = L(\theta; \mathbf{Y_{obs}}) f(\mathbf{Y_{mis}}|\mathbf{Y_{obs}}, \theta). \tag{1.6}$$

We transform (1.6) with the natural logarithm, and by switching from the Bayesian perspective to the 'classical' Frequentist perspective, each term can be expressed as a function of the model parameters,

$$\ell(\theta; \mathbf{Y}) = \ell(\theta; \mathbf{Y_{obs}}) + \ln f(\mathbf{Y_{mis}}|\mathbf{Y_{obs}}, \theta) + c, \tag{1.7}$$

where $\ell(\theta; \mathbf{Y})$ and $\ell(\theta; \mathbf{Y_{obs}})$ denote the complete- and the observed-data loglikelihoods, and c is an arbitrary constant. We often cannot calculate the *conditional predictive distribution* of the missing-data, so we use a *current* estimate $\theta = \theta^{(t)}$ to get

theory that the principle of natural selection had been published twice before Darwin. And Charles Darwin himself acknowledged this in a later edition of *On the Origin of Species*, but stated that he had not been aware of the existence of these publications (one predecessor of Darwin was Patrick Matthew, who added his views on natural selection in the appendix of his work *Naval Timber and Arboriculture*). Gould points out that there is a difference between the simple formulation of a thesis, and the full understanding of its importance and consequences.

[11] This is true for data fusion patterns, if the conditional independence assumption holds (Rässler 2002).

$$Q(\theta|\theta^{(t)}) = \ell(\theta; \mathbf{Y_{obs}}) + H(\theta|\theta^{(t)}) + c, \tag{1.8}$$

where

$$H(\theta|\theta^{(t)}) = \int \ln f(\mathbf{Y_{mis}}|\mathbf{Y_{obs}}, \theta) f(\mathbf{Y_{mis}}|\mathbf{Y_{obs}}, \theta^{(t)})\, d\mathbf{Y_{mis}},$$

and the E-step consists of identifying the *current* expected complete-data loglikelihood[12],

$$Q(\theta|\theta^{(t)}) = \int \ell(\theta; \mathbf{Y}) f(\mathbf{Y_{mis}}|\mathbf{Y_{obs}}, \theta^{(t)})\, d\mathbf{Y_{mis}}.$$

The M-step – as the name already suggests – maximizes the 'current' expected complete-data loglikelihood to get updated parameters $\theta^{(t+1)}$:

$$Q(\theta^{(t+1)}|\theta^{(t)}) \geq Q(\theta|\theta^{(t)}), \quad \forall\theta. \tag{1.9}$$

A proof that demonstrates why the observed-data likelihood of $\theta^{(t+1)}$ is at least as high as the one of $\theta^{(t)}$, is given in the DLR paper where the terms were re-written such that eventually Jensen's inequality could be applied to a remaining term.

Repeating (1.8) and (1.9) until convergence is achieved, should ideally yield a global maximum for θ, but sometimes the EM algorithm is caught in a local maximum instead. One way of reducing this risk is making parallel runs from different 'initial guesses' for $\theta^{(0)}$ (see e.g. Schafer 1997).

Due to the generality of EM, and due to different ways of computational implementation, there exists a large variety of similar algorithms which Little & Rubin (2002) label "EM-type" algorithms.

[12] In particular for exponential family distributions it is enough to calculate the expected complete-data sufficient statistics.

The benefit of EM over Direct ML is that it is computationally inexpensive, and that therefore it will return a solution, where the direct or raw ML approach does not work anymore. The major drawback of EM is that it 'pretends' to have a complete-data loglikelihood at its disposal, and as a consequence, standard errors are too low when EM is applied to missing-data problems.

1.4.3 Doubly-Robust estimation

Likelihood-based approaches prepare the ground for MI in the next chapter, but before we would like to introduce another method that has gained a considerable amount of attention during the last decade. Doubly-robust (DR) estimation is assuming a model (1) for the probability that a specific value is observed (also called *inverse-probability weighting*, where Robins et al. (1995), Robins & Rotnitzky (1995) contributed to its more recent improvement), and (2) for the joint distribution of the partially and fully observed data. The term 'doubly robust' or 'doubly-protected' stems from the fact that DR estimators are consistent, if *either* model is correctly specified. In theory, where MI might yield biased estimates (if the imputation model is misspecified), DR has a 'second chance', if model (1) is correctly specified. Carpenter et al. (2006) describe such a case in a simulation study.

We feel, however, that in practice the probability of this case is extremely small and not very intuitive. Besides, just like the two likelihood-based approaches, DR can be used to estimate just a limited number of quantities of interest. Therefore, either 'analyst' and 'imputer' are in close touch with each other, or they are one and the same person. Under these circumstances, it is unlikely that the imputation model is 'wrong', but the analysis model is not. As mentioned in the beginning of this section, we will re-visit the issue of MI under misspecified models for $\mathbf{Y_{mis}}|\mathbf{Y_{obs}}, \theta$ at a later stage.

The benefit of Multiple Imputation is that it provides a general purpose solution for incomplete-data analysis, rather than a solution for one specific analysis.

2

Multiple Imputation

2.1 The Bayesian Framework of MI

In Bayesian statistics inferences are typically drawn from the posterior distribution. Under *ignorability*, applying Bayes' theorem to the model parameters of an incomplete data set yields

$$\pi(\theta|\mathbf{Y_{obs}}) = \frac{\pi(\theta)f(\mathbf{Y_{obs}}|\theta)}{f(\mathbf{Y_{obs}})}, \tag{2.1}$$

where the left-hand-side is the observed-data *posterior* distribution, and $\pi(\theta)$ is the *prior* distribution of the model parameters. Since the marginal distribution of $\mathbf{Y_{obs}}$ can be regarded as a normalizing constant that does not influence the location and scale parameters of $\pi(\theta|\mathbf{Y_{obs}})$, this can be re-written as

$$\pi(\theta|\mathbf{Y_{obs}}) \propto \pi(\theta)f(\mathbf{Y_{obs}}|\theta). \tag{2.2}$$

Moreover, we can 'update' our knowledge of the data (the observed-data likelihood), because the *posterior predictive distribution* of the missing part of $\mathbf{Y}$ is given by

$$f(\mathbf{Y_{mis}}|\mathbf{Y_{obs}}) = \int (\mathbf{Y_{mis}}|\mathbf{Y_{obs}}, \theta)\pi(\theta|\mathbf{Y_{obs}})\, d\theta. \tag{2.3}$$

Suppose we have a data situation, where only one variable of an $n \times p$ matrix $\mathbf{Y}$ has missing values. Then 'Proper' MI as defined by Rubin (1987*a*) consists of making random draws from the second term of the right hand side of 2.3 – the *posterior distribution* of θ given the observed data – followed by random draws from the first term of the right hand side – the *conditonal predictive distribution* of $\mathbf{Y_{mis}}$ given the observed data and θ. This two-step procedure is necessary, since we cannot draw directly from the *posterior predictive distribution* of the missing data given the observed data. Carrying out these two steps $M > 1$ times yields M data sets that are identical for $\mathbf{Y_{obs}}$, and different for $\mathbf{Y_{mis}}$. Note that 'Proper' MI is only straightforward, if the missing-data pattern is monotone. If this is not the case, we have to apply strategies that will give us asymptotical draws from the correct distributions. Some of these strategies – which can be classified as Markov Chain Monte Carlo (MCMC) techniques – are introduced later in this chapter.

Note that random draws from $\mathbf{Y_{mis}}|\mathbf{Y_{obs}}, \theta$ only, would mean that we would assume that the model parameters stem from a complete-data likelihood. Earlier versions of statistical software SPSS featured such an 'improper' MI algorithm in its MVA module (see also Allison 2001).[1]

2.2 MI confidence intervals

Rubin (1987*a*) gave definitions for multiple imputation confidence intervals which are in general more conservative than the usual single data set confidence intervals. The theoretically bigger width is created by combining the *within* variance W for some quantity of interest θ with the *between* variance B of the M different estimates for θ. W is defined by averaging over the variances obtained from the M data sets:

[1] Later SPSS/PASW releases feature a 'proper MI' algorithm.

$$W = \frac{1}{M}\sum_{m=1}^{M}\widehat{var}(\widehat{\theta}_m). \tag{2.4}$$

The MI estimate itself for θ is given by

$$\widehat{\theta}_{MI} = \frac{1}{M}\sum_{m=1}^{M}\widehat{\theta}_m, \tag{2.5}$$

which allows us to calculate the *between* variance,

$$B = \frac{1}{M-1}\sum_{m=1}^{M}(\widehat{\theta}_m - \widehat{\theta}_{MI})^2. \tag{2.6}$$

The *total* variance T is the sum of the *within* and *between* variance – accounting for finite values of M – and is defined by

$$T = W + \frac{M+1}{M}B. \tag{2.7}$$

If the z transformation for any quantity of interest θ, $(\widehat{\theta}-\theta)/\sqrt{Var(\theta)}$, asymptotically follows a standard normal distribution, the MI confidence intervals can be established using a *t* distribution,

$$\frac{(\widehat{\theta}_{MI}-\theta)}{\sqrt{T}} \sim t_v, \tag{2.8}$$

with

$$v = (M-1)\widehat{\gamma}_M^{-2} \tag{2.9}$$

degrees of freedom (Rubin & Schenker 1986, Rubin 1987*a*), where

$$\widehat{\gamma}_M = (1+M^{-1})tr(B\,T^{-1})/K, \tag{2.10}$$

is the fraction of missing information about the K-dimensional θ.

Barnard & Rubin (1999) developed an improved expression, for example for small sample sizes,

$$\upsilon^{*} = (\upsilon^{-1} + \widehat{\upsilon}_{obs}^{-1})^{-1}, \tag{2.11}$$

where

$$\widehat{\upsilon}_{obs} = (1 - \widehat{\gamma}_{M}) \left(\frac{\upsilon_{com} + 1}{\upsilon_{com} + 3} \right) \upsilon_{com},$$

and υ_{com} represents the (hypothetical) complete-data *t* inferences.

The higher total variance accounts for additional uncertainty due to the imputed part $\mathbf{Y_{mis}}$, and yields information with respect to the shape of the empirical distribution of $\widehat{\theta}_{MI}$.This distinguishes MI from single imputation approaches which generally 'treat' imputed values as if they were observed.

2.3 Markov Chain Monte Carlo techniques for Multiple Imputation

As mentioned in the previous section, some workaround is required, if the missing-data pattern is non-monotone. The proposed strategies resemble the basic idea of the EM algorithm insofar, as the concept of 'filling-in' missing data, subsequently using these data to get a better estimate for the model parameters, and iterating the two steps, is also key to these approaches. The difference is that all of them contain at least one stochastic component.

Markov Chain Monte Carlo (MCMC) describes a collection of techniques that aim to generate (pseudo-)random draws from a stationary target distribution, where the Markov Chain is the mean to achieve stationarity. Markov chains are defined as stochastic processes, where future states only depend on the current state. The

Monte Carlo component means that the (pseudo-)random draws from the probability distributions behave like independent draws.

2.3.1 Data Augmentation

Data Augmentation (DA) (Tanner & Wong 1987, Li 1988) is closely related to a more general class of MCMC methods, called Gibbs Sampling (Geman & Geman 1984, Gelfand & Smith 1990). This technique provides a way to sample from a multivariate probability density, by employing only the densities of partitions $j = 1, \dots, r$ of some parameter $\theta = [\theta_1, \dots, \theta_r]^T$ conditional on all the other partitions. Let us assume that the conditional distributions

$$\theta_j | \theta_1, \dots, \theta_{j-1}, \theta_{j+1}, \dots, \theta_r \sim \pi(\theta_j | \theta_1, \dots, \theta_{j-1}, \theta_{j+1}, \dots, \theta_r)$$

are known, and the joint distribution $\pi(\theta)$ exists. Then we can start the Gibbs Sampler by choosing a point $\theta^{(0)}$ in the parameter space Θ to initiate the process:

$$\begin{aligned}
&\theta_j^{(1)} | \theta_1^{(1)}, \dots, \theta_{j-1}^{(1)}, \theta_{j+1}^{(0)}, \dots, \theta_r^{(0)} \sim \pi(\theta_j^{(1)} | \theta_1^{(1)}, \dots, \theta_{j-1}^{(1)}, \theta_{j+1}^{(0)}, \dots, \theta_r^{(0)}) \\
&\quad \vdots \\
&\theta_j^{(t)} | \theta_1^{(t)}, \dots, \theta_{j-1}^{(t)}, \theta_{j+1}^{(t-1)}, \dots, \theta_r^{(t-1)} \sim \pi(\theta_j^{(t)} | \theta_1^{(t)}, \dots, \theta_{j-1}^{(t)}, \theta_{j+1}^{(t-1)}, \dots, \theta_r^{(t-1)}), \\
&\quad \text{for } j = 1, \dots, r.
\end{aligned} \tag{2.12}$$

If t is chosen large enough, $\theta^{(t)}$ should converge in distribution to the stationary density $\pi(\theta)$ (see e.g. Geweke 1992).

Schafer (1997) gives a detailed explanation, how DA relates to Gibbs Sampling, but for our purposes it is enough to know that for missing-data problems, draws from the posterior predictive distribution of the missing data are incorporated into the sequence. If we distinguish between the draws from the posterior distribution for θ and the draws from the posterior predictive distribution for $\mathbf{Y_{mis}}$, these steps can be viewed as the stochastic counterparts to the corresponding steps in Expectation-Maximization.[2] Tanner & Wong (1987) therefore referred to drawing

$$\mathbf{Y}_{\mathbf{mis}}^{\mathbf{(t+1)}} \sim f(\mathbf{Y}_{\mathbf{mis}}|\mathbf{Y}_{\mathbf{obs}}, \theta^{(\mathbf{t})}), \tag{2.13}$$

as the *Imputation* (I-)step, and the *Maximization* step is replaced by the *Posterior* (P-)step, where we draw

$$\theta^{(t+1)} \sim \pi(\theta|\mathbf{Y}_{\mathbf{obs}}, \mathbf{Y}_{\mathbf{mis}}^{\mathbf{(t+1)}}). \tag{2.14}$$

In order to get independent random draws from a stationary distribution, typically a so-called 'burn-in period' is used before the first imputation draw, and between any two of the $M-1$ remaining imputation draws k iteration cycles are discarded. Alternatively, different starting values can be used to kick off M different cycles, where the first k iterations are likewise discarded. The latter approach makes the Gibbs Sampler less sensitive to the choice of the starting value.

2.3.2 Sampling Importance Resampling

The Sampling Importance Resampling (SIR) algorithm (Rubin 1987*b*) is an alternative to DA, and more related to the basic Metropolis-Hastings algorithm (Metropolis et al. 1953, Hastings 1970). Suppose we are again confronted with a non-monotone missing-data pattern that makes it extremely difficult to sample directly from the observed-data posterior or the posterior predictive distribution of $\mathbf{Y}_{\mathbf{mis}}$. Suppose further we are able to define a "good approximation" (Rubin 1987*a*) to the joint posterior distribution of $(\mathbf{Y}_{\mathbf{mis}}, \theta)$,

$$\tilde{g}(\mathbf{Y}_{\mathbf{mis}}, \theta|\mathbf{Y}_{\mathbf{obs}}) = \tilde{\pi}(\theta|\mathbf{Y}_{\mathbf{obs}})\tilde{f}(\mathbf{Y}_{\mathbf{mis}}|\mathbf{Y}_{\mathbf{obs}}, \theta). \tag{2.15}$$

[2] Note that Rubin (1994) explicitly states that MI did not evolve from EM or DA, "both logically and historically".

We can use $\tilde{g}$ as a proposal or importance sampling function. The algorithm can be divided into three steps:

1. Draw J values of $(\mathbf{Y_{mis}}, \theta)$ from the proposal density $\tilde{g}(\mathbf{Y_{mis}}, \theta|\mathbf{Y_{obs}})$
2. calculate the importance sampling weights
$$w_i \propto \frac{f(\mathbf{Y_{obs}}, \mathbf{Y^{(i)}_{mis}}|\theta^{(i)})\pi(\theta^{(i)})}{\tilde{g}(\mathbf{Y^{(i)}_{mis}}, \theta^{(i)}|\mathbf{Y_{obs}})} \quad \text{, for } i = 1, \ldots, J \tag{2.16}$$
3. Draw $M < J$ values of $\mathbf{Y_{mis}}$ from $(\mathbf{Y^{(1)}_{mis}}, \ldots, \mathbf{Y^{(J)}_{mis}})$ with replacement and weights proportional to $w_1, \ldots, w_J$

to generate multiple imputations of $\mathbf{Y_{mis}}$. Unlike DA the SIR algorithm is non-iterative. Therefore, it is technically not a Markov Chain procedure. The choice of the ratio J/M depends on the fraction of missing information given in (2.10). Gelfand & Smith (1990) modified the SIR algorithm by splitting step 3 into two conditional draws.

2.3.3 The Bayesian Bootstrap

The Bayesian Bootstrap (BB) (Rubin 1981) is a Bayesian equivalent to classical Bootstrapping (Efron 1979), although its original purpose was to approximate the posterior distribution of θ. Therefore, we consider the BB not as an alternative approach to DA, but rather to its parametric posterior step.

Let Y denote a partially observed variable, where cases $1, \ldots, n_{obs}$ are observed, and the remaining $n - n_{obs}$ cases are missing. Then the algorithm is implemented the following way:

1. Generate $n_{obs} - 1$ random draws $U = [u_1, \ldots, u_{n_{obs}-1}]$ from a [0,1] uniform distribution, and sort the values in ascending order to get $U^* = [u_{(1)}, \ldots, u_{(n_{obs}-1)}]$, where $u_{(1)} < u_{(2)}, \ldots, u_{(n_{obs}-2)} < u_{(n_{obs}-1)}$. Create two

$n_{obs} \times 1$ vectors $W = [U^*, 1]^T$ and $V = [0, U^*]^T$, and calculate the differences $D = [(d_1 = w_1 - v_1), \ldots, (d_{n_{obs}} = w_{n_{obs}} - v_{n_{obs}})]^T$, where $\sum_{i=1}^{n_{obs}} d_i = 1$.

2. Perform one random draw of size n_{obs} from a multinomial distribution with the $n_{obs} \times 1$ vector D as probability weights to obtain sample n^*_{obs}.

Based on this data set random draws for θ are performed. This procedure replaces random draws from a theoretical distribution for $\theta|\mathbf{Y}$. The BB steps given above vary slightly from the original notation given in Rubin (1981, 1987*a*), but are computationally equivalent. Note that Efron and Rubin have different views on whether (classical) Bootstrapping can be used as well for sampling from a posterior distribution in order to generate multiple imputations (see Efron 1994, Rubin 1994).

Where fully-parametric posterior distribution can lead to biased imputations, because the functionality of the model was misspecified, Bayesian Bootstrapping can at least alleviate the introduction of bias due to model misspecification due to its robust properties. We will revisit the BB in chapters 5 and 6, where it is implemented into a semi-parametric MI algorithm.

A variant of the BB, called the Approximate Bayesian Bootstrap (ABB), was proposed by Rubin & Schenker (1986) as an alternative MI technique. Suppose we have a partially observed variable Y, and other (completely observed) variables from the same data set can be used to stratify the sample into G cells, with $g = 1, \ldots, G$, where the elements in each cell are independent and identically distributed. $n_{obs,g}$ ($n_{mis,g}$) denotes the number of observed (missing) units of cell g. For each cell the following three steps are carried out:

1. Draw a random sample $n^*_{obs,g}$ of size $n_{obs,g}$ (with replacement) from $n_{obs,g}$.
2. Draw another random sample $n^*_{mis,g}$ of size $n_{mis,g}$ (with replacement) from $n^*_{obs,g}$.
3. Impute the missing values of Y in cell g with the values obtained from $n^*_{mis,g}$.

Multiple imputations are created by repeating these steps M times. Note that the BB can be extended to MI in a similar way (e.g. suggested by Cohen 1997). Kim (2002) introduced a modified ABB approach that reduces MI variance bias due to small cell sample sizes, but the problem remains that for application to large survey data, no combination of strata might be identifiable that fulfils the requirement of identical distributions within each of the G cells.

2.4 Joint-Modeling and Fully-Conditional Specifications

2.4.1 Joint-Modeling

The 'classical' literature on MI typically assumes a multivariate distribution $f(\mathbf{Y}|\theta)$, for the $n \times p$ dimensional data matrix $\mathbf{Y} = [\mathbf{Y_{obs}}, \mathbf{Y_{mis}}]$. The Bayesian framework of MI described in section 2.1 then allows us to draw from the posterior predictive distribution of $f(\mathbf{Y_{mis}}|\mathbf{Y_{obs}})$. Recently, this strategy has been labeled as 'Joint Modeling' (JM) approach (see e.g. van Buuren et al. 2006, Drechsler & Rässler 2008).

Schafer (1997) postulated such models for continuous, categorical, and mixed-scale data. Especially the latter two suffered from the 'curse of dimensionality' which could only be alleviated by imposing restrictions on the underlying loglinear models. In general, JM approaches require $n >> p$, or more precisely, n should be considerably larger than the combined number of dimensions of all subparameters of θ.

2.4.2 Fully-Conditional Specifications

Fully Conditional Specifications (FCS) describe a class of strategies that do not aim to estimate the parameters of the joint distribution $f(\mathbf{Y}|\theta)$. The objective is rather to model only a subset via conditional distributions. For instance, (unrestricted) Chained Equation Regression models are carried out variable by variable, such that the univariate conditional distributions are given by $f(\mathbf{Y_j}|\mathbf{Y_{-j}}, \theta_j)$, where $\mathbf{Y_{-j}} = [\mathbf{Y_1}, \dots, \mathbf{Y_{j-1}}, \mathbf{Y_{j+1}}, \dots, \mathbf{Y_p}]$. Instead of having to deal with one p-dimensional problem, FCS splits the task into p one-dimensional problems (van Buuren et al. 2006). Imputation of data sets with mixed-scale variables which often occur in Survey Methodology, becomes much more tractable, because models for different scale-levels can be flexibly applied. Moreover, logical inconsistencies or filter-caused missing values can be easily handled under FCS as well. Another advantage is the applicability to data sets with an extremely large number of variables or an p/n-ratio that is close to or sometimes even exceeding 1. Of course, in such cases restrictions have to be imposed on the imputation model to reduce the number of predictors. For instance IVEware (Raghunathan et al. 2001), a statistical MI software based on chained equations, uses stepwise regressions. Finally, attention should be paid to the issue of convergence: we may often not be able to prove that the joint distribution exists[3], since the specifications of the distributions are all conditional. FCS also requires a much larger number of parameters than JM which makes it difficult to monitor the rates of convergence, even if methods like the *worst linear function of the parameters* (see e.g. Schafer 1997) are applied. Results from Drechsler & Rässler (2008) who conducted a simulation study with incompatible Data Augmentation, suggest that a lack of convergence can lead to biased inferences, while others figured out only minor problems. The academic debate about best practices for incompatibility seems not to be settled yet.

Despite of the aforementioned caveat, we will focus in chapters 6 and 7 on applications with FCS. The decision in favor of FCS for our own developed routines corresponds with the popularity of chained-equation based MI software like IVEware or MICE (*Multivariate Imputation by Chained Equations*) (van Buuren & Oudshoorn 1999).

[3] Which is of course a pre-requisite for assuming that FCS draws converge to joint distributional draws.

3

Standard Predictive Mean Matching and extensions

3.1 Predictive Mean Matching

This section gives a short introduction to a nearest-neighbor matching technique that was first published by Rubin (1986) in the context of statistical file matching (data fusion), and emerged under the term *Predictive Mean Matching* (PMM) coined by Little (1988*a*). In the following we describe this technique in detail, because all semi-parametric MI algorithms introduced in the next chapter are based on it. PMM is widely regarded as a hot-deck imputation technique with a connotation to single imputation, although Rubin (1986), Little (1988*a*) already described MI extensions for PMM.

The basic concept of PMM is to impute a missing value by matching its predictive mean to a nearest neighbor among the predictive means of the observed values, and to adopt the actual observed value. Let y_i be a value of a variable Y, with $i = 1, \ldots, n$, where only units $1, \ldots, n_{obs}$ are observed. Moreover, let $\hat{y}_i$ be a corresponding predictor from a regression of Y on some explanatory variables. Then the distance metric between $\hat{y}_i$ and $\hat{y}_j$ is given by

$$D_{i,j} = |\hat{y}_i - \hat{y}_j|, \tag{3.1}$$

and we impute $y_{obs,j}$ for $y_{mis,i}$, if $D_{i,j} \leq D_{i,k} \; \forall \; k = 1, \ldots, n_{obs}$.

By imputing *observed* values, PMM fulfills the requirement of getting *plausible* values which is important, if our objective is to impute for discrete data with an underlying continuous distribution assumption. But PMM has another useful property: It gives more robust estimates in the presence of model misspecification, as figure 3.1 illustrates.

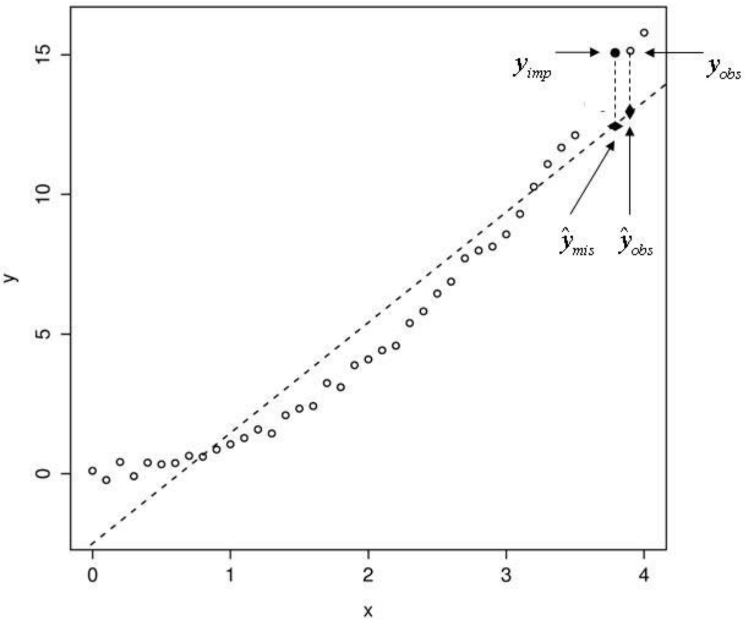

Figure 3.1: Identification and imputation of a nearest neighbor with PMM

The incomplete variable Y is a quadratic function of X (with some random noise added) but the imputation model uses a linear function for missing values in Y. As we can see, the imputed value y_{imp} still remains close to the real function, whereas a purely model-based imputation would have given (more) biased inferences. Beissel-Durrant & Skinner (2004) applied PMM to hourly pay distributions, where PMM could preserve the ”spiky behaviour” better than fully-parametric imputation methods.

The difficult part about Predictive Mean Matching is to utilize its robust properties within the Multiple Imputation framework in a way that Rubin’s combination rules still yield unbiased variance estimates.

3.2 PMM for block-wise missing data

The PMM distance metric for predictors of a single variable is straightforward. But if we need to define the term 'nearest' over more than one variable, it already matters, whether we are using squared or absolute distances, and whether all variables are treated identically or not.

Little (1988*a*) proposes to use the Mahalanobis distance metric to solve this problem. The underlying principle of this approach is to penalize distances between predictors of a variable more severely, if the goodness of fit for the imputation model is high.

Let $\mathbf{X}$ be some design matrix based on the completely observed part of the data set, and let $\mathbf{Y} = [Y_1, \dots, Y_p]$ be a matrix consisting of p variables with an identical missingness pattern, where the first n_{obs} cases are observed, and cases $n_{obs}+1, \dots, n$ are missing. Furthermore, let $\mathbf{y}_i$ denote a vector of length p for unit i, containing either p completely observed or completely missing values of $\mathbf{Y}$. $\hat{\mathbf{y}}_i$ is the corresponding vector of predictors based on $\mathbf{Y_{obs}} = g(\mathbf{X_{obs}})$, where $\mathbf{Y_{obs}}$ and $\mathbf{X_{obs}}$ are the first n_{obs} rows of $\mathbf{Y}$ and $\mathbf{X}$. In the current implementation the identity link is used for the link function $g(\cdot)$. The distance of the predictive means is given by

$$D^2_{j,k} = (\hat{\mathbf{y}}_j - \hat{\mathbf{y}}_k)^T \mathbf{S}_{\mathbf{Y_{obs}} \cdot \mathbf{X_{obs}}}{}^{-1} (\hat{\mathbf{y}}_j - \hat{\mathbf{y}}_k), \tag{3.2}$$

where $k \in 1, \dots, n_{obs}$, $j \in n_{obs}+1, \dots, n$, and $S_{\mathbf{Y_{obs}} \cdot \mathbf{X_{obs}}}$ is the $p \times p$ covariance matrix of the residuals from the regression of $\mathbf{Y_{obs}}$ on $\mathbf{X_{obs}}$. The resulting matrix $\mathbf{D^2}$ has dimension $(n - n_{obs}) \times n_{obs}$. The nearest neighbor for each $\mathbf{y}_j$ is given by the corresponding row minimum of $\mathbf{D^2}$.

In order to get a better overview of the influence of each matched predictor, we can transform $S_{\mathbf{Y_{obs}} \cdot \mathbf{X_{obs}}}$ into a diagonal matrix (i.e. no covariances among the residuals from the p regressions). Let $\hat{e}_{obs,1}$ be the estimated residuals from the regression of $Y_{obs,1}$ on $\mathbf{X_{obs}}$. Regressing $Y_{obs,2}$ on $\mathbf{X_{obs}}$ and $\hat{e}_{obs,1}$, then regressing $Y_{obs,3}$ on $\mathbf{X_{obs}}$ and $\hat{e}_{obs,1,2}$, until finally regressing $Y_{obs,p}$ on $\mathbf{X_{obs}}$ and $\hat{e}_{obs,1 \dots p-1}$, eventually yields the desired diagonal matrix for $S_{\mathbf{Y_{obs}}\cdot\mathbf{X_{obs}}}$, because the covariances between the residuals were partialled out. Proceeding this way eases the introduction of manual weights, if the imputer wants to modify the solution created by the distance metric, in case he feels that some $Y's$ are more important than others for the analysis (and thus should be matched with relatively higher precision). In order to generate multiple imputations, Little (1988*a*) suggested to draw $\tilde{\mathbf{y}}_j$ from the posterior predictive distribution instead of using $\hat{\mathbf{y}}_j$. This coincides with one of the algorithm we tested before finally deciding which methods should be used for the experimental design.

3.3 PMM extension for binary and categorical variables

Survey data sets often feature binary and unordered categorical variables. In order to make use of PMM's desirable properties, it is necessary to extend Predictive Mean Matching to these variable types as well (since 'classical' PMM was designed for metric scale variables only).

The PMM extension to the binary case is straightforward: Instead of matching on a linear predictor, we are using a binomial logit model[1] for dichotomous data. Hence $\hat{\pi}_i = (1+\exp\{-\mathbf{x}_i^T\hat{\beta}\})^{-1}$ for $i = 1, \ldots, n$, where $\mathbf{x_i}$ is the i^{th} row of some (completely observed) design matrix $\mathbf{X}$, and the distance is calculated analogously to (3.1) with $D_{i,j} = |\hat{\pi}_i - \hat{\pi}_j|$.

One innovation of this thesis is the introduction of a PMM extension for unordered categorical variables based on a multinomial logit model. The *aregImpute* function of the *Hmisc* library (Harrell 2006) in *R* contains a PMM variant for categorical data

[1] In the case of binary variables, PMM was suggested even before 1986 for a different objective: Rosenbaum & Rubin (1983) introduced the technique in for causal inference problems and called it *Propensity Score Matching*, but instead of imputing dichotomous variables, they used it to balance treatment and control groups (which, under closer inspection, *is* also an imputation of missing data). If we treat the group identifying variable like all other variables, Propensity Score Matching could be regarded as a special case of PMM, which is why we will use the term Predictive Mean Matching throughout this paper – for any variable type.

as well, but its procedure relies on splitting the categories into dummies.

The main tasks of developing a PMM variant for unordered categorical variables are similar to the ones discussed in the previous section: We have to match over several variables (categories), and we need a method that relates the distances for the matches over the categories to each other.

Let C denote the total number of categories for some categorical variable Y, where the probability of observation i of being in category c is given by

$$\hat{\pi}_{c,i} = \frac{\exp\{-\mathbf{x}_i^T \hat{\beta}_c\}}{1 + \sum_{s=1}^{C-1} \exp\{-\mathbf{x}_i^T \hat{\beta}_s\}}$$

for $c = 1, ..., C-1$, and $\hat{\pi}_{C,i} = 1 - \sum_{c=1}^{C-1} \hat{\pi}_{c,i}$ for category C.

For the multinomial logits we have to find a metric that identifies nearest neighbors for a vector, rather than a scalar as in the metric-scale and binary case – at least, if we try to find a nearest neighbor not only for one (the highest) predicted p-value, but rather over the complete range of all $\pi_1, \ldots, \pi_C$ of some categorical variable Y. We presume that matching over all C categories yields more reliable matches. But that also means that choosing different distance functions yields different results. One suggestion for such a distance metric between predicted values of missing and observed values of Y would be the sum of the squared differences over all C categories,

$$D_{i,j}^{cs} = \sum_{c=1}^{C} (\hat{\pi}_{c,i} - \hat{\pi}_{c,j})^2.$$

However, sometimes the probabilities for some categories are rather small, and the above suggestion does not take into account relative deviations. Therefore, we are using the logits rather than the probabilities themselves,

$$D_{i,j}^{cl} = \sum_{c=1}^{C} (z_{c,i} - z_{c,j})^2, \text{ with } z_c = \ln\left(\frac{\hat{\pi}_c}{1 - \hat{\pi}_c}\right). \tag{3.3}$$

Of course, we could have used absolute deviations instead of squared differences. One effect of choosing squared differences, however, is that 'outliers' (with respect to matched categories) are punished more severely, which is why we are minimizing the sum of squared deviations rather than the sum of absolute deviations.

Sometimes the fit of the multinomial logit model is (virtually) perfect, such that π_k for some category k of the 'factor' variable to be estimated equals or gets extremely close to 1. In this case the odds ratios $\lim\limits_{\pi_k \to 1} \pi_k/(1-\pi_k) = \infty$, and no nearest neighbor can be identified. For this reason, p values should be truncated within the bound $[0.001; 0.999]$. This still ensures that matching on the log-odds ratios 'favors' very small and very large p values, but it limits the log-odds ratios to the bound $[-6.90675; 6.90675]$.

3.4 A Goodness-of-Fit Measure for the matching quality of nearest neighbor techniques

Suppose the PMM imputation model is correctly specified, but the nearest neighbor is not 'near' at all. Large distances can considerably reduce the benefits of nearest-neighbor imputation techniques. These circumstances may arise for predictors of extreme values in the variable space, or if the donor pool is small in general. Therefore, we propose a GoF-measure that monitors the matching quality for metric-scale variables. The basic idea is to compare average distances of actually matched pairs with average distances from a randomized matching between recipients and donors. Since occasionally $n_{mis} > n_{obs}$, assignment of a donor is always assumed to be with replacement. Assuming further the predictors are normally distributed we get

$$\widehat{Y}_{obs} \sim N(\mu^*_{obs}, \sigma^{2*}_{obs}) \quad \text{and} \quad \widehat{Y}_{mis} \sim N(\mu^*_{mis}, \sigma^{2*}_{mis}.$$

Assuming independence between the predictors of observed and unobserved values, the difference of two normally distributed variables is also normally distributed with

$$\underbrace{(\widehat{Y}_{mis} - \widehat{Y}_{obs})}_{D} \sim N(\underbrace{(\mu^*_{mis} - \mu^*_{obs})}_{\mu_D}, \underbrace{(\sigma^{2*}_{mis} + \sigma^{2*}_{obs})}_{\sigma^2_D}).$$

The squared z-transformations of differences $D = [d_i]$, with $i = 1, \ldots, n_{mis}$, are χ^2-distributed,

$$\left(\frac{d_i - \mu_D}{\sigma_D}\right)^2 \sim \chi^2_1,$$

and we obtain

$$E(D^2) = \sigma^2_D + \mu^2_D.$$

Assuming that squared distances from the actual matching process should on average be at least as small as squared distances based on random assignment, our efficiency measure is given by

$$0 \leq G_{match} = \frac{\hat{\sigma}^2_D + \hat{\mu}^2_D - \overline{D^*}}{\hat{\sigma}^2_D + \hat{\mu}^2_D} \leq 1, \tag{3.4}$$

where $\overline{D^*} = 1/n \sum_{i=1}^{n_{mis}} d^*_i$, and $d^*_i = min(\widehat{y}_{mis,i} - \widehat{y}_{obs,j})^2, \forall j \in N_{obs}$.

Own applications have shown that G_{match} is often close to 1, and it might be sensible to modify it with an exponentiation term k to get a higher variance for this efficiency measure. However, this would make the interpretation of G_{match} more difficult.

3.5 Benefits and Drawbacks of PMM

The PMM variant for block-wise missing data is particularly suited for missing-by-design patterns, where a large number of variables is simultaneously observed or missing. Using this PMM variant automatically guarantees logical consistency among the imputed Y variables. Moreover, it can be applied to problems, where the number of variables is very large or stems from relational databases (where turning the data into a rectangular data sets induces a large number of variables as well). The original data can be used to derive some proxy variables – e.g. by using factor analysis – to reduce the number of variables for the imputation task. Thereafter the original variable set can be matched to a pairlist obtained from the imputation of the proxy variables.

A drawback of all nearest-neighbor approaches is that they can not capitalize on the complete domain of a variable, but only on the observed part of it. Missing parts of truncated data can not be sensibly imputed with nearest-neighbor approaches, even under a correctly specified imputation model (see fig. 3.2).

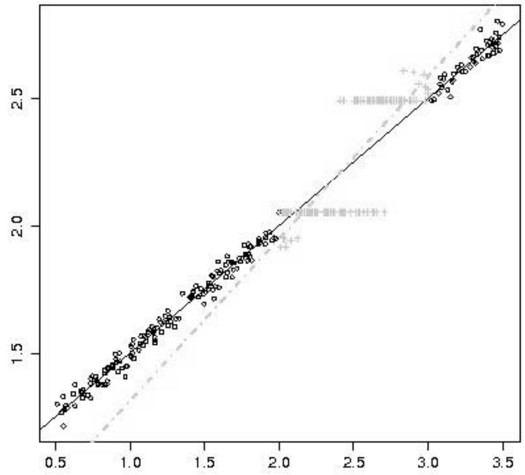

Figure 3.2: Observed and PMM-imputed values (grey) for truncated data (and reestimated regression line)

In a broader sense truncation can be seen as a special case of poor coverage of parts of the variable space of Y. As already pointed out, these situations may arise in particular if the sample size (more precisely: the donor pool) is small. The simulations study in chapter 5 features a data small set with $n = 200$, in order to investigate such effects.

Our a priori belief in the positive aspects of PMM (plausible values and robustness towards model misspecification) is expressed by the omittance of PMM variants for ordinal-scale variables – despite the fact that this variable type is often prevalent in survey data. We assume that applying models for (continuous) metric-scale variables in combination with PMM will still yield consistent estimates for partially incomplete variables. The test design in chapter 6 evaluates this hypothesis, since some of the variables used in the study are ordinal-scale. Our hypothesis is backed up by the results from Münnich & Rässler (2005), where a binary variable was multiply imputed using a linear regression model in conjunction with PMM, and other studies, where PMM was used for MI in survey data settings (e.g. Landerman et al. 1997).

4

Multiple Imputation for metric-scale discrete data

4.1 Parametric MI variants

The following multiple imputation algorithms are all 'classical' MI algorithms in the sense that (1) we perform random draws for θ from an observed-data posterior distribution $\pi(\theta|\mathbf{Y_{obs}})$, followed by (2) random draws $\tilde{Y}_{mis}$ for $\mathbf{Y_{mis}}$ from their conditional predictive distribution $f(\mathbf{Y_{mis}}|\theta, \mathbf{Y_{obs}})$. Since both steps rely on distributional assumptions for $\mathbf{Y}$ and θ, we label these algorithms as *Parametric MI* variants. However, since we are discussing random draws from continuous posterior predictive distributions for metric-scale *discrete* data, another step (3) is needed to modify $\tilde{Y}_{mis}$ into genuine draws for $\mathbf{Y_{mis}}$, in order to avoid *zero-probability* measures. Steps (1) to (3) are carried out M times to generate multiple imputations for $\mathbf{Y_{mis}}$. Two variants of step (3) propose techniques which are related to a nearest-neighbor technique called *Predictive Mean Matching* (PMM) (Rubin 1986, Little 1988*a*). Note that, although PMM is a non-parametric imputation method (which appears to contradict the *Parametric MI* categorization), we merely use it to modify draws from the posterior predictive distribution.

Suppose we have a data situation, where only one integer random variable $Y = [y_i]$ with $i = 1, \ldots, n$ has some missing values. Let $\mathbf{X}$ denote the $n \times p$ design matrix of the completely observed variables that are used as predictors in the standard Bayesian linear regression model

$$Y|\mathbf{X} \sim N(\mathbf{X}\beta, \sigma^2), \tag{4.1}$$

with a noninformative Jeffrey's prior distribution,

$$\pi(\beta, \sigma^2) \propto \sigma^{-2}. \tag{4.2}$$

Furthermore let $\mathbf{X_{obs}}$ ($\mathbf{X_{mis}}$) denote the rows for which Y is observed (missing). The first two steps of the following algorithms are all identical:

(1) Draw σ^2 and β from their respective observed data posterior distribution as given in e.g. Box & Tiao (1992).

$$\sigma^2|\mathbf{X_{obs}}, Y_{obs} \sim (Y_{obs} - \mathbf{X_{obs}}\hat{\beta})^T(Y_{obs} - \mathbf{X_{obs}}\hat{\beta})\chi^{-2}_{n-p} \tag{4.3}$$

and

$$\beta|\mathbf{X_{obs}}, Y_{obs}, \sigma^2 \sim mvN_p(\hat{\beta}, (\mathbf{X_{obs}}^T\mathbf{X_{obs}})^{-1}\sigma^2), \tag{4.4}$$

where $\hat{\beta} = (\mathbf{X_{obs}}^T\mathbf{X_{obs}})^{-1}\mathbf{X_{obs}}^T Y_{obs}$ is the (unweighted) OLS estimate for the vector of the model parameters β.

(2) Draw from the conditional predictive distribution of the missing data given the observed data

$$\tilde{Y}_{mis}|\beta, \sigma^2 \sim N(\mathbf{X_{mis}}\beta, \sigma^2|\mathbf{X_{mis}}). \tag{4.5}$$

If not stated otherwise the subindices j, k denote those cases of Y that are observed, and the subindex i denotes missing units of Y. Step (3) varies among the different algorithms, but all methods are in one way or the other linked to the observed values. This guarantees a final value for $y_{mis,i}$ which belongs to the value space of Y. The three steps are carried out M times to create multiple imputations.

4.1.1 Rounding to the Nearest Observed Value

Rounding to the Nearest Observed Value (ROV) is a straightforward way to turn the draws from a continuous – in this case the normal – distribution $\tilde{Y}_{mis}$ into discretely distributed random draws.[1]

(3) identify $y_{obs,j} \leq |\tilde{y}_{mis,i} - y_{obs,k}| \, \forall k$ and impute $y_{obs,j}$ for $y_{mis,i}$

Note that ROV is implemented in some MI software packages such as Schafer's NORM (Schafer 1999).

4.1.2 Inverse Probability Rounding

A stochastic variant of ROV is *Inverse Probability Rounding* (IPR), sometimes also called *stochastic rounding* or *round-random*. The draws from the continuous distribution are not deterministically rounded to the nearest observed value, but instead the final value depends on a random draw with inverse probability to the relative distance of $\tilde{Y}_{mis,i}$ to the respective nearest lower and higher observed values.

(3) Round $Y_{lb,i} \leq \tilde{y}_{mis,i} \leq Y_{ub,i}$ to the lower-bound plausible value $Y_{lb,i}$ with probability $p = \frac{\tilde{y}_{i,mis} - Y_{lb,i}}{Y_{ub,i} - y_{lb,i}}$ and round to the upper-bound plausible value $Y_{lb,i}$ with probability $q = 1 - p$. If $\tilde{y}_{i,mis}$ is less or greater than any $y_{k,obs}$, IPR is identical to ROV.

4.1.3 PRIMA - Predictive Imputation Matching

Predictive Imputation Matching (PRIMA), as proposed by Münnich & Rässler (2005), was first applied to binary data as part of the DACSEIS project[2] and can be considered as 'Proper MI plus original PMM', because the PMM-step is simply

[1] Rounding to the nearest integer was not considered, because this approach might yield implausible values.

[2] More information about this project can be retrieved at *http://www.dacseis.de/*

carried out for the draws from the conditional predictive distribution, i.e. we have to generate draws from the conditional predictive distribution for the observed units as well:

(2) Draw $\tilde{y}_{obs,k}|\beta, \sigma^2 \sim N(\mathbf{X_{obs}}\beta, \sigma^2)$.

(3) Identify $\tilde{y}_{obs,j} \leq |\tilde{y}_{mis,i} - \tilde{y}_{obs,k}| \, \forall k$ and impute $y_{obs,j}$ for $y_{mis,i}$.

4.1.4 Semi-PRIMA

We labeled this approach 'Semi-PRIMA', because the draws from the conditional predictive distribution for $\tilde{y}_{mis,i}$ in step (2) are replaced by using $\hat{y}_{mis,i}$, but – analogously to PRIMA – $\tilde{y}_{obs,k}$ is drawn from the conditional predictive distribution:

(2) Compute $\hat{y}_{mis,i} = \mathbf{x_{mis,i}}^T\hat{\beta}$ and draw $\tilde{y}_{obs,k}|\beta, \sigma^2 \sim N(\mathbf{X_{obs}}\beta, \sigma^2)$.

(3) Identify $\tilde{y}_{obs,j} \leq |\hat{y}_{mis,i} - \tilde{y}_{obs,k}| \, \forall k$ and impute $y_{obs,j}$ for $y_{mis,i}$.

This approach is already a hybrid between the parametric MI variants and the semi-parametric variants introduced in the next section. It is also the already mentioned suggestion by Little (1988*a*) to transform PMM into an MI algorithm.

4.2 Semi-parametric variants based on Predictive Mean Matching

This section contains those MI variants, where at least the draw from the posterior predictive distribution is replaced by a non-parametric alternative (PMM).

4.2.1 M nearest neighbors

The simplest strategy of creating multiple imputation not only from PMM, but from any nearest-neighbor approach in general is to leave the algorithm unchanged from the single imputation variant, and to take over not only the nearest, but the $M > 1$ nearest neighbors. This approach and the next one differ from all the other presented algorithms fundamentally, because multiple imputations are not obtained by carrying out the corresponding steps M times.

4.2.2 Rounded Predictive Mean Matching

Rounded Predictive Mean Matching (RPMM) is an MI PMM variant that only works for discrete data. The algorithm starts just like classical PMM by obtaining the complete vector $\mathbf{\hat{y}} = [\hat{y}_1, \ldots, \hat{y}_n]$

(1) Compute $\mathbf{\hat{y}} = \mathbf{X}\hat{\beta}$.

(2) Round the predictive means of all observed units $\mathbf{\hat{y}_{obs}}$ to the nearest observed value to get a new vector $\mathbf{y^*_{obs}}$.

(3) For every $\hat{y}_{mis,i}$ identify $y^{*,(i)}_{obs}$ (denoting the value of $\mathbf{y^*_{obs}}$ that is nearest to $\hat{y}_{mis,i}$).

(4) Sample M times with replacement from the units $n_{obs}(y^{*,(i)}_{obs})$ whose predictive means were rounded to $y^{*,(i)}_{obs}$, and take over their actually observed values.

4.2.3 Posterior Predictive Mean Matching

Whereas RPMM can still be seen as an extension to a stand-alone hot deck single imputation technique, Posterior Predictive Mean Matching (PPMM) is more of a hybrid between classical multiple imputation and Predictive Mean Matching. More precisely, we follow the steps from (4.3) and (4.4), getting parameter draws which are identical to 'classical' linear model-based MI. But instead of drawing values from the conditional predictive distribution in (4.5), we replace this step by Predictive Mean Matching:

(1) Draw $\sigma^2 | \mathbf{X_{obs}}, Y_{obs} \sim (Y_{obs} - \mathbf{X_{obs}}\hat{\beta})^T (Y_{obs} - \mathbf{X_{obs}}\hat{\beta}) \chi^{-2}_{n-p}$ and
$\beta | \mathbf{X_{obs}}, Y_{obs}, \sigma^2 \sim mvN_p(\hat{\beta}, (\mathbf{X_{obs}}^T \mathbf{X_{obs}})^{-1} \sigma^2)$

(2) Compute $\hat{\mathbf{y}} = \mathbf{X}\hat{\beta}$.

(3) Identify $\hat{y}_{obs,j} \leq |\hat{y}_{mis,i} - \hat{y}_{obs,k}| \, \forall k$ and impute $y_{obs,j}$ for $y_{mis,i}$.

Multiple imputations are obtained by replicating the above three steps $M > 1$ times.

4.2.4 Bayesian Bootstrap Predictive Mean Matching

Further relaxing distributional assumptions about the distributions of σ^2 and β leads to replacing equations (4.3) and (4.4) by Bayesian Bootstrap (Rubin 1981) draws for β. The main difference between 'classical' Bootstrapping and its Bayesian counterpart is that we administer sampling weights to each observation. These weights are also randomized draws and sum up to 1. Due to this additional random step Bayesian Bootstrapping is using fewer observations on average than Bootstrapping.

The imputation step is again replaced by Predictive Mean Matching.

(1) Generate a BB sample n^*_{obs} as defined in section 2.3.3.

(2) Estimate β^* based on n^*_{obs}.

(3) Compute $\hat{\mathbf{y}} = \mathbf{X}\hat{\beta}^*$.

(4) Identify $\hat{y}_{obs,j} \leq |\hat{y}_{mis,i} - \hat{y}_{obs,k}| \, \forall k$ and impute $y_{obs,j}$ for $y_{mis,i}$.

The idea behind this modification to PPMM is to make the procedure less sensitive to distributional assumptions regarding the model parameters. We label this approach 'Bayesian Bootstrap Predictive Mean Matching' (BBPMM). Multiple imputations are, again, obtained by carrying out the described steps $M > 1$ times.

4.3 Performance assessment

Before we decided which algorithms should eventually be used in the experimental design described in the next chapter, we conducted some pre-tests to evaluate the general applicability of the algorithms using a completely observed variable X and an incomplete variable Y (50% MCAR) that was a linear function of X with a normal error component, such that $\rho_{x,y} = 0.5$. All values were rounded to the next integer. The pre-tests did not investigate MI variance estimators, but were primarily directed at biases of three point estimators: mean and median of Y, as well as the correlation between X and Y.

A surprising result is the relatively bad performance of PRIMA for the correlation estimator. Apparently, drawing from the posterior predictive distribution and combining it with PMM creates too much randomness.

'col 1' is a completely observed integer variable, and 'col 3' is an integer variable with 50% of the values being MCAR. A single imputation step with PRIMA yields the result given in the sunflower plot in figure 4.1.

The correlations are biased towards zero, when PRIMA is used for MI, and the same effects can be observed to some lesser extent for Semi-PRIMA which is why both variants were not considered for the experimental design.

Taking the M nearest neighbors from a 'classical' PMM induces biasing effects on the correlation estimates as well. This effect is caused by not considering the relative distances between the M values of $\hat{y}_{obs}$ and the matched $\hat{y}_{mis,i}$. The effect is like super-imposing a uniform distribution over the distribution of Y. RPMM is also mildly affected by this, but was still considered for the experimental design in chapter 5.

ROV and IPR were also pre-tested, and it turns out that IPR introduces too much noise into the assignment process of the final value – similar to PRIMA and Semi-PRIMA.

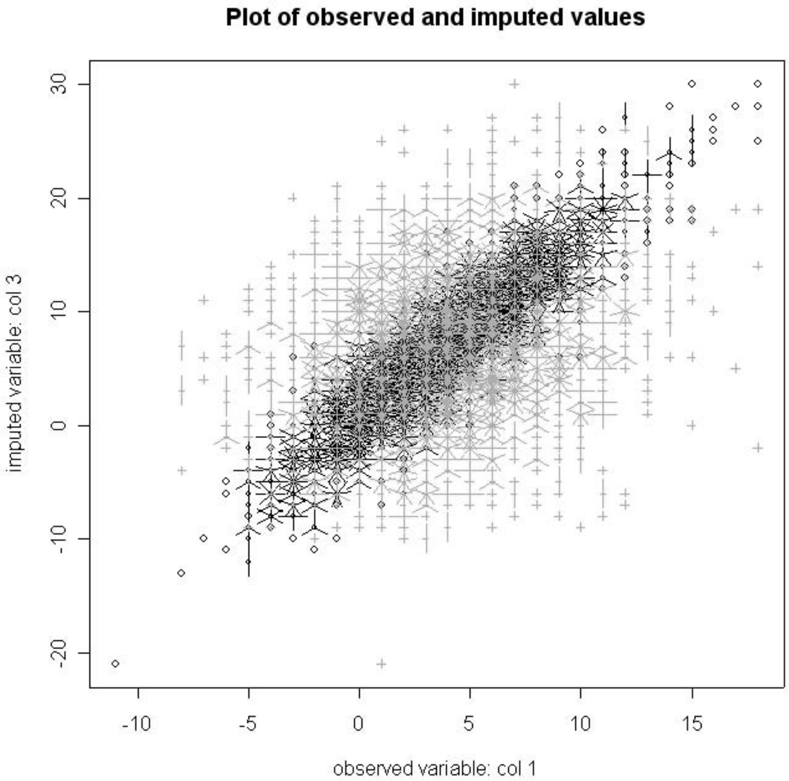

Figure 4.1: Observed and PRIMA-imputed MCAR values (grey) for two discrete variables (only y partially incomplete)

As a result of the pre-tests, ROV, RPMM, PPMM and BBPMM were chosen for the experimental design in the next chapter, whereas ROV was additionally needed to investigate a side effect of rounding on variance estimators.

5

Multiple Imputation for metric-scale discrete data: an experimental design

5.1 Description of the experimental design

Surveys collecting information using questionnaires may result in many types of variables. Rounded variables or genuine integer value variables are common. ′Age′, for instance, becomes ′Age in Years′ or ′Year of Birth′ in a questionnaire and therefore rounded. Further, there is a group of variables which are ordered-categorical, like grading-type variables (″On a scale from X to Y, how many...″) that can be considered close enough to genuine metric-scale variables to use analysis and imputation techniques for metric integer variables.

This experimental design evaluates several strategies for imputing missing values in such variables. The goal of this design is to compare different Multiple Imputation algorithms, and to investigate their applicability to rounded variables under a variety of data situations. All algorithms used in the experimental design draw from a continuous distribution and imputations are then "coarsened" to rounded values. This approach can be seen as reverse to the work of Heitjan & Rubin (1990, 1991), who had coarsened data such as 'Heaped Age' at their disposal and wanted to get precise values.

As stated in the previous chapter we expect the proposed methods to have specific strengths and weaknesses, we test them under different data situations. These data situations are set up within a multi-factorial Monte Carlo study which is explained in full detail in section three. The focus of the experimental design in this Monte Carlo study is twofold: The main focus of the Monte Carlo study is on robustness towards model misspecification, as the variants of Predictive Mean Matching are assumed to add robustness to the imputation models. Additionally, we want to investigate the effects of rounding on variance and covariance estimates – primarily under the correctly specified model (in order to avoid contaminating effects of the deliberate model misspecification). As derived by Sheppard (1898), rounding overestimates the variance by $(u-l)^2/12$ (the variance of the Uniform distribution) with u and l being the upper and lower bound values (see e.g. Dempster & Rubin 1983).[1] The rounding problem in (multiple) imputation was first mentioned by Horton et al. (2003), who used a different technique to derive the bias for a dichotomous variable if it was treated as continuous in the imputation process.

5.2 MI variants used for the experimental design

Throughout the complete experimental design a linear model is used for imputation – irrespective of the real data generating function. For the sake of coherence we repeat shortened versions of the MI variant descriptions from the previous chapters.

1. *Rounding to the nearest observed value* (ROV) in this context is the 'classical' MI as previously described with $\hat{y}_{mis}$ rounded to the nearest observed y_{obs}. Rounding to the nearest observed value ensures that only plausible values are imputed.

2. *Rounded Predictive Mean Matching* (RPMM) rounds any missing (observed) $\hat{y}_{mis}$ ($\hat{y}_{obs}$) to the nearest observed value y_{obs} and performs PMM. Since rounding to the nearest observed value almost certainly ensures zero-distances to

[1] unless the domain of the continuous variable is $[k-0.5; k+0.5[$.

several potential donors, multiple imputations are simply created by drawing M times (with replacement) from the pool of equally nearest neighbors.

3. *Posterior Predictive Mean Matching* (PPMM) takes over the posterior steps from (4.3) and (4.4), but we replace the random draw from the posterior predictive distribution in (4.5) by Predictive Mean Matching.

4. *Bayesian Bootstrap Predictive Mean Matching* (BBPMM): The Posterior-draws are replaced by a Bayesian Bootstrap (BB) draws. The Posterior Predictive-draws are replaced by PMM.

The three PMM variants are the remainders of an originally larger pool of MI-PMM algorithms that we developed for this design. But some of these algorithms were too seriously flawed to be considered for the experimental design.

5.3 Simulation study

5.3.1 Objective

The goal of this Monte Carlo study is to examine the performance of the four imputation methods for rounded variables under different conditions. A potential drawback of nearest neighbor approaches (and also of ROV) is that they only impute *observed* rather than *plausible* values. This clearly is a problem in settings with censored or truncated data, and around predicted means when data are scarce. Generally, this could pose a greater problem for small data sets (with a smaller number of potential donors).

5.3.2 Data generation

All simulated data sets considered in our Monte Carlo study consist of three variables – two completely observed variables X_1 and X_2 and one variable Y with missing values. In total we create twelve different data sets: Three different data generating functions for Y, two sample sizes, $n = 200$ and $n = 2000$ (in the analyses abbreviated as 'small' and 'big' data set), and two missing-data mechanisms: *missing completely at random* (MCAR) and *missing at random* (MAR).[2] This study can be considered as a 3×2×2 factorial design.

Throughout all data sets X_1 and X_2 are generated using

$$X_1 \sim U(0,3)\text{ ; and } x_2 = -x_1 + \varepsilon\text{ , with } \varepsilon \sim N(0,4).$$

The following three distributions were used for generating Y:

1. $y_1 = [1.75 + x_1 - 0.5x_2 + u_1]$, with $u_1 \sim N(0, \frac{11}{48})$
2. $y_2 = [1.75 + x_1 - 0.5x_2 + (u_2 - \frac{107}{96})]$, with $u_2 \sim \chi^2_{\frac{107}{96}}$
3. $y_3 = [4 + 1.5(x_1 - 1.5)^3 + 0.25 \cdot log(abs(x_2 + 9)) + u_3]$,
   with $u_3 \sim N(0, 0.2)$.

We defined the error terms in (1) and (2), such that the expectational value and the variance of Y are in both cases an integer; $E(Y) = 4$ for both data sets, $Var(Y) = 3$ for data set (1) and $Var(Y) = 4$ for data set (2). The parametric and semi-parametric imputations for missing values in the first data set are under a correctly specified model. The remaining two are misspecified models: The second data set violates the normality assumption, and the third data set violates the linearity assumption of the imputation model.

We expect the fully-parametric imputation algorithm (ROV) to yield best results for data set (1), since it incorporates all information available from the data, whereas for data sets (2) and (3) the semi-parametric algorithms might give less biased inference. Furthermore we expect the PMM variants to yield worse results for the small data sets, since a *decreased* pool of potential donors means an *increased* average distance between nearest neighbors.

[2] NMAR was not included, since MI is assuming *ignorability*.

5.3.3 Missing-data mechanisms

We set the rate of missing values to 60% for both the 'big' and the 'small' data set in order to create enough missingness to identify differences in performance for the tested algorithms which leaves only 80 potential donors for the small data sets. The *missing at random* (Rubin 1976, 1987*a*) mechanism is related to X_1, and defined by

$$y_i = \begin{cases} \text{missing,} & \text{if } F_Z(z_i) > 0.4 \\ y_i & \text{if } F_Z(z_i) \leq 0.4 \end{cases}, \forall i = 1, \dots, n$$

where $F_Z(z)$ is the empirical distribution function of Z, and

$$z = g(x_1) = \frac{1}{1 + \exp(0.2 x_1 \phi + \varepsilon)}, \text{ with } \phi \sim N(0, 16) \text{ and } \varepsilon \sim N(0, 36).$$

Bias, MSE and coverage are all based on the values given in table5.1.

Table 5.1: True values for all three data sets and quantities of interest

	DS1	DS2	DS3
$E(Y)$	4	4	4.491
$Var(Y)$	3.001	5.001	3.841
$p(Y < 3)$	0.2033	0.2565	0.1334
$p(Y < 4)$	0.3944	0.4358	0.2185
$p(Y < 6)$	0.7967	0.7821	0.7836
$\rho(X_1, Y)$	0.75	0.5812	0.8783
$\rho(X_2, Y)$	-0.8279	-0.6415	-0.3138
α	1.75	1.749	3.997
β_1	1	1	1.5
β_2	-0.5	-0.5002	0.2513

5.3.4 Multiple Imputation analysis

We consider the following ten estimands of interest:

- mean: $E(Y)$
- variance: $Var(Y)$

- quantiles: $P(Y < 3)$, $P(Y < 4)$ and $P(Y < 6)$
- correlations: $\rho(X_1, Y)$ and $\rho(X_2, Y)$
- 'true model' parameter estimates (three parameters).

The considered imputation techniques were evaluated based on bias, MSE, and 95% confidence intervals. We use the normal approximation for the proportions and the log-variance. For the correlations, however, we need to apply the z transformation, also suggested by Schafer (1997), which gives us $z = 0.5 \cdot ln[(1 + \rho)(1 - \rho)^{-1}]$ and calculate CIs for z with variance $(n - 3)^{-1}$. Taking the inverse $z^{-1} = tanh(z)$ for the lower and upper bounds of z gives us the lower and upper bound estimates for ρ.

Although the analytical derivation of the true values for some of the quantities is straightforward, we generated all true values – for the sake of continuance – by generating 20,000 data sets upfront, and took the mean over these data generations. These values were then used to derive the estimated bias, MSE and coverage.

5.3.5 Computational consequences of the factorial design

We set the number of multiple imputations to $M = 15$ for all twelve data sets (three different data generators, two different sample sizes, two different missingness mechanisms). We want to compare the four different imputation algorithm results with the complete data sets and the *complete cases*. For each of the 12 conditions, we generated 5,000 replicates for the small data set and 500 replicates for the large data set. In total the design amounts to 2,029,500 data set generations.

5.4 Analysis of the imputed data

The objective of this section is to examine the results by starting with a very aggregated level, and becoming more small-grained with every subsection. The vast amount of analyses does not allow us to show all tables and results, but we try to distil the most relevant findings by applying this funneled approach.

5.4.1 Abbreviations

Throughout this section we are using the following short forms: BD (before deletion – with Y completely observed), CC (complete cases – the incomplete data set after removing 60% of all values of Y), ROV (fully-parametric MI with rounding to the nearest observed value), PPMM (Predictive Mean Matching with random draws from the observed-data posterior for the parameters) BBPMM (Predictive Mean Matching with a Bayesian Bootstrap step for the parameter draws), RPMM (Rounded Predictive Mean Matching), MCAR (60% *missing completely at random*), MAR (60% *missing at random*), 'big' data set (for the simulated data sets with $n = 2,000$), 'small' data set (for the simulated data sets with $n = 200$), and data set (1), (2), and (3), referring to the three different functions we used for generating Y in our simulated data sets.

5.4.2 Averaged effects

Given the factorial nature of the experimental design, we analyzed the data with tables that average the results for bias and coverage over the different factors 'sample size', 'missing-data mechanism' and 'data generating model'. We use the relative bias $(\widehat{E}(\hat{\theta}) - \theta)/\theta$ to compare the bias of the complete cases and the four imputation procedures. Table 5.2 gives the overall mean and the cell specific means for every factor.

Table 5.2: Average relative bias (in %) by various factors

bias	CC	ROV	PPMM	BBPMM	RPMM
overall	4.02	6.17	0.58	0.58	1.39
big	3.98	6.12	0.70	0.67	1.32
small	4.07	6.21	0.46	0.48	1.45
MCAR	*0.15*	6.19	0.59	0.53	1.24
MAR	7.90	6.14	0.57	0.62	1.54
DS(1)	3.65	0.23	0.20	0.21	0.36
DS(2)	3.17	0.81	0.24	0.26	0.29
DS(3)	5.26	**17.46**	1.30	1.26	3.51

The complete cases (CC) of the MCAR data represent a general reference point for the imputation methods. For the (relative) bias no imputation method achieves the value of the complete cases under MCAR (0.15% in *italic* font), although BBPMM and PPMM get close for data sets (1) and (2). Overall, Posterior Predictive Mean Matching and Bayesian Bootstrap Predictive Mean Matching have the smallest average bias.

The results for the coverage in table 5.3 show the percentages of a 95 % confidence interval containing the true value.

Table 5.3: Average coverage (in %) by various factors

coverage	CC	ROV	PPMM	BBPMM	RPMM
overall	82.02	84.09	90.80	92.37	83.75
big	75.80	77.71	91.82	93.19	80.77
small	88.23	90.47	89.79	91.55	86.73
MCAR	*94.81*	83.92	91.23	92.75	84.09
MAR	69.23	84.26	90.37	91.99	83.41
DS(1)	82.82	95.58	91.37	92.75	86.24
DS(2)	82.53	90.71	88.96	90.48	86.46
DS(3)	80.70	**65.99**	92.08	93.88	78.54

Again, the average MCAR effect of the complete cases ((94.81% in *italic* font) serves as a reference figure. Overall BBPMM is getting closest to the nominal 95% (with 92.37%), and it is also the only method that achieves a coverage of 90% or more for every effect.

ROV makes use of the full information under a correctly specified model in data set (1), and it outperforms PPMM and BBPMM with respect to coverage (95.58%), but not for the bias (0.23% vs 0.20 and 0.21% respectively). The good performance for data sets (1) and (2) is outweighed by the extremely bad results for data set (3) (in bold font), making ROV the method with the most biased estimates overall. In return, this indicates the relative strength of PMM methods over purely model-based techniques with respect to model misspecification.

The bias estimates of RPMM are better than the ROV results, but the coverage is on a similar overall level. In particular for data set (3) RPMM yields higher bias estimates and lower coverage for the imputation under a misspecified model, compared with the other two PMM variants. An interesting result is that the analysis figures are on average less biased for both PPMM and BBPMM for the small data sets compared with the big data set counterparts (whereas the coverage is higher for the big data sets).

Note that the detailed tables – which are the basis for these aggregated tables – are listed in section B of the appendix.

5.4.3 Best and worst performing method

Displaying results for the best- and worst-performing method gives us slightly more detailed information, since we now also consider the ten analysis quantities which helps us to identify potential causes for the shortcomings displayed in the previous tables.

Overview

We (further) abbreviate the considered imputation methods ROV, PPMM, BBPMM and RPMM by using numbers 1 to 4. Table 5.4 gives results for the best/worst performing method in terms of bias, where the figure before (after) the 'slash' symbol displays the best (worst) method for a particular condition and analysis figure. For instance, the two figures at the bottom right corner show that method '2' (PPMM) is the best method with the smallest bias for the true model parameter β_2, whereas method '1' (ROV) is the worst.

Table 5.4: best/worst performing method for relative bias (in %) for all 12 conditions

data set	miss.	b/s	$E(Y)$	$Var(Y)$	$p(y<3)$	$p(y<4)$	$p(y<6)$	$\rho(x_1,y)$	$\rho(x_2,y)$	α	β_1	β_2
DS(1)	MCAR	big	1/2	4/1	4/1	4/1	4/1	3/4	2/4	1/4	1/4	2/4
DS(1)	MCAR	small	1/3	1/3	4/1	4/1	2/1	3/4	2/4	2/4	2/4	2/4
DS(1)	MAR	big	1/4	2/1	2/1	2/4	3/1	3/4	2/4	2/4	3/4	2/4
DS(1)	MAR	small	1/4	1/3	4/1	2/4	2/1	2/4	2/4	1/4	1/4	2/4
DS(2)	MCAR	big	4/1	4/1	3/1	4/1	4/1	2/4	2/4	2/4	2/4	2/4
DS(2)	MCAR	small	4/1	4/3	1/3	2/1	3/1	3/4	2/4	2/1	2/3	2/4
DS(2)	MAR	big	1/4	2/4	2/1	2/1	2/1	1/4	1/4	1/4	1/4	3/4
DS(2)	MAR	small	1/4	1/4	2/1	2/1	2/1	3/4	2/4	2/4	1/4	1/4
DS(3)	MCAR	big	2/4	2/1	4/1	3/1	3/1	2/4	1/4	3/1	2/1	3/1
DS(3)	MCAR	small	3/4	3/1	2/1	4/1	2/1	2/4	1/4	3/1	2/1	3/1
DS(3)	MAR	big	2/4	2/1	3/1	3/1	3/1	2/4	1/4	3/1	2/1	3/1
DS(3)	MAR	small	1/4	1/4	2/1	2/1	3/1	2/4	1/4	3/1	2/1	2/1

By focusing on the second number (worst method) we can identify two clear patterns:

1) ROV performs worst for data set (3), if the analysis figure does incorporate a distributional assumption, thus verifying the results from the previous table and our assumption that the PMM variants are less sensitive to model misspecification. The mean aside, this comprises all quantiles and the 'true model' parameter estimates.

2) RPMM performs worst for correlations and to a lesser extent for the model parameter estimates. One possible explanation is that the predicted values are rounded and a donor is randomly chosen from the pool of potential donors, all of which have zero-distances to the rounded predictive mean of the missing value. Pooling donors through rounding, however, is blind to the original distances between predictive means, and information on bi- or multivariate associations is lost.

   The best method figures do not show any comparably distinct patterns, especially because PPMM and BBPMM are very similar, and tend to 'cannibalize' each other. ROV seems to yield very good mean estimates (seven out of twelve times best method). BBPMM and PPMM share the 'best method' counts for the quantiles and 'true model' parameters of data set (3), except for two quantile figures, where RPMM was 'best method'.

The best/worst figures for the coverage in table 5.5 show a slightly different picture to their counterparts for bias: ROV gets a lot more 'best method' counts. In some situations the picture has reversed completely, where ROV switched from 'worst method' for bias to 'best method' for coverage.

Table 5.5: best/worst performing method for coverage (in %) for all 12 conditions

data set	miss.	b/s	$E(Y)$	$Var(Y)$	$p(y < 3)$	$p(y < 4)$	$p(y < 6)$	$\rho(x_1, y)$	$\rho(x_2, y)$	α	β_1	β_2
DS(1)	MCAR	big	1/4	1/2	1/2	1/4	1/3	3/4	3/4	1/4	1/4	1/4
DS(1)	MCAR	small	1/4	1/4	1/2	1/2	1/2	3/4	3/4	1/2	1/4	1/4
DS(1)	MAR	big	3/1	1/2	1/3	1/4	1/2	3/4	3/4	1/4	1/4	1/4
DS(1)	MAR	small	1/2	1/2	1/2	1/2	1/2	1/4	3/4	1/2	1/4	1/4
DS(2)	MCAR	big	1/4	2/1	1/3	2/1	3/1	3/4	3/4	1/4	3/4	1/4
DS(2)	MCAR	small	1/2	1/2	1/4	1/2	1/2	1/4	1/4	1/4	3/4	1/4
DS(2)	MAR	big	1/4	1/2	1/4	1/4	2/1	3/4	3/4	1/4	3/4	1/4
DS(2)	MAR	small	1/4	1/4	1/4	1/4	1/4	1/4	1/4	1/2	1/4	1/4
DS(3)	MCAR	big	3/1	1/4	3/1	2/1	2/1	2/4	1/4	3/1	3/1	3/1
DS(3)	MCAR	small	1/4	1/2	3/1	3/1	3/1	1/4	1/4	3/4	3/1	3/4
DS(3)	MAR	big	1/4	1/4	3/1	3/1	3/1	3/4	1/4	3/1	3/1	3/1
DS(3)	MAR	small	1/4	1/4	3/1	4/1	3/1	1/4	1/4	3/4	3/1	3/4

By looking at the aggregated overview for 'best method'/'worst method' counts in table 5.6, we can confirm our impression that ROV and RPMM are overall the methods with the most 'worst method' counts.

Table 5.6: Totals for best/worst occurrences for relative bias and coverage (in %)

	n(best)	n(worst)
ROV	26	54
PPMM	54	1
BBPMM	25	6
RPMM	15	59

	n(best)	n(worst)
ROV	74	26
PPMM	6	21
BBPMM	39	3
RPMM	1	70

All six 'worst method' counts for BBPMM in the overview for bias occur for small data sets. But why does BBPMM perform more poorly for small data sets? The PMM variants rely on a donor pool that allows for small distances between matches. Especially in the case of the MAR data sets the average between-match distances increase for imputations in areas with disproportionate missing values of Y given X_1 and X_2. Additionally for BBPMM, the bootstrap step discards some observations for the parameter estimation, due to sampling with replacement. The remaining observations are much more influential, and therefore the parameter estimates can strongly deviate from the 'correct' model in some extreme bootstrap samples. The long-run properties of the bootstrap should adjust for the outliers, but it is possible, that even with 5,000 repetitions, traces of this effect can be found in the results. As a consequence one should choose M large, if the sample size is small. In the simulation study, all BBPMM 'worst method' counts were still close to the other methods, which explains why according to table 5.2 BBPMM is the method with the joint smallest overall bias together with PPMM, although the latter has more 'best method' counts and only one 'worst method' count.

Especially the table for coverage also shows the flaw of just looking at the methods in terms of 'best' and 'worst', because they lack information with respect to 'how good' and 'how bad'. The 'ANOVA-type' tables do not carry any information about the ten analysis figures, but the best/worst tables do not show the relative quality of each method's estimates. By just looking at the coverage totals in table 5.6, ROV looks like the most favorable method (74 out of 120 possible 'best method' counts), but table 5.3 showed that – averaged over all twelve data situations and all ten analysis figures – BBPMM outperforms ROV by more than eight percentage points, indicating that ROV has an extremely low coverage for some situations that outweighs the many 'best method' counts. Only the *combined* information of tables 5.2 through 5.6 leads us to a comprehensive picture of the imputation quality of the considered methods. To gain additional micro-level insight, we focus on two diametral data situations in the next subsection, where the corresponding tables not only give the name of the best/worst method, but also the corresponding estimates.

Most and least favorable data situation

Data set (1) (correctly specified model) with an MCAR mechanism and big sample size can be considered the *most favorable* data situation for imputation, as the imputation model is correctly specified, and all algorithms benefit from bigger data sets. Table 5.7 displays the values for this particular data situations showing results for the best and worst performing method with respect to *bias* and *coverage* over all ten analysis figures.

Table 5.7: *best* and *worst* method for the big data set (1), MCAR

	big DS1, MCAR	$Bias$	$Coverage$
$E(Y)$	best	ROV (0.00076)	ROV (0.95)
	worst	PPMM (0.00121)	RPMM (0.93)
$Var(Y)$	best	RPMM (-0.0061)	ROV (0.96)
	worst	ROV **(0.0423)**	PPMM (0.95)
$p(y < 3)$	best	RPMM (-5.9e-05)	ROV (0.98)
	worst	ROV (1.5e-03)	PPMM (0.92)
$p(y < 4)$	best	RPMM (7.5e-05)	ROV (0.97)
	worst	ROV (4.0e-04)	RPMM (0.91)
$p(y < 6)$	best	RPMM (-0.00039)	ROV (0.96)
	worst	ROV (-0.00179)	BBPMM (0.91)
$\rho(x_1, y)$	best	BBPMM (0.00034)	BBPMM (0.97)
	worst	RPMM (-0.01252)	RPMM **(0.80)**
$\rho(x_2, y)$	best	PPMM (-0.00018)	BBPMM (0.98)
	worst	RPMM (0.01509)	RPMM **(0.54)**
α	best	ROV (0.0032)	ROV (0.97)
	worst	RPMM **(0.0418)**	RPMM **(0.76)**
β_1	best	ROV (-0.00091)	ROV (0.97)
	worst	RPMM (-0.01738)	RPMM (0.84)
β_2	best	PPMM (0.00075)	ROV (0.96)
	worst	RPMM (0.01024)	RPMM **(0.77)**

An interesting finding is that ROV is generally doing well in this data situation, but rounding to the nearest observed value seems to distort the quantiles $p(y < 3)$, $p(y < 4)$ and $p(y < 6)$, and the variance is overestimated by slightly more than $1/12$.[3] The coverage for the variance is not affected by this, because the impact

[3] Since $Var(Y) = 3$, the relative bias would have been $(3\frac{1}{12} - 3)/3 = 0.02\bar{7}$.

of an average bias of 0.0423 is rather small. This table also confirms the previous result that RPMM seems to affect multivariate estimators like correlations or model parameters.

As mentioned earlier the quality of the results of nearest neighbor approaches depends on the distances for identified matches. Therefore, the small data sets tend to penalize the PMM variants, as the 'choice' of potential donors is much smaller ($n_{obs} = 80$). Therefore, the *least favorable* data situation – in particular for the PMM variants – is the MAR version of the small data set (3). MAR in combination with small samples is 'thinning out' parts of the domain considerably, thus tending to yield larger distances for nearest neighbor approaches.

The results from table 5.8 confirm our ex-ante hypothesis regarding the relative superiority of the PMM variants over fully-parametric approaches, when the imputation model is misspecified.

Table 5.8: *best* and *worst* method for the small data set (3), MAR

	small DS3, MAR	$Bias$	$Coverage$
$E(Y)$	best	ROV (-0.0052)	ROV (0.96)
	worst	RPMM (-0.0271)	RPMM (0.91)
$Var(Y)$	best	ROV (0.060)	ROV (0.90)
	worst	RPMM (-0.101)	PPMM (0.85)
$p(y < 3)$	best	PPMM (0.0016)	BBPMM (0.94)
	worst	ROV (0.0252)	ROV (0.91)
$p(y < 4)$	best	PPMM (0.0018)	RPMM (0.93)
	worst	ROV **(0.0644)**	ROV **(0.61)**
$p(y < 6)$	best	BBPMM (0.00022)	BBPMM (0.92)
	worst	ROV **(-0.07444)**	ROV **(0.58)**
$\rho(x_1, y)$	best	PPMM (-0.0029)	ROV (0.97)
	worst	RPMM (-0.0199)	RPMM (0.83)
$\rho(x_2, y)$	best	ROV (0.0017)	BBPMM (0.95)
	worst	RPMM (0.0111)	RPMM (0.88)
α	best	BBPMM (0.0047)	BBPMM (0.96)
	worst	ROV **(0.4594)**	RPMM (0.83)
β_1	best	PPMM (-0.040)	BBPMM (0.87)
	worst	ROV **(-0.172)**	ROV **(0.57)**
β_2	best	PPMM (0.0025)	BBPMM (0.96)
	worst	ROV **(-0.2329)**	RPMM (0.82)

In particular the 'true model' parameters are extremely biased: the ROV estimate for α deviates on average by almost 46% from the true value. Since RPMM (not ROV!) has the worst coverage with 83%, this also indicates that ROV-based MI yields very inefficient estimates in this data situation (the findings for β_2 can be interpreted in a similar way). The relative weakness of RPMM with respect to correlation estimates is also clearly visible.

5.4.4 Detail analysis

In a way the above results already exhibit the inherent weaknesses of ROV and RPMM, but in order to confirm the key findings, we will investigate some of the 24 basic analysis tables that can also be found in the appendix B of this thesis.

We decided to display results for the relative bias of the small data set (1) with an MAR mechanism (see table 5.9), because they are characteristic for all small data set results regarding the underestimation of variances for the PMM variants.[4] Another pattern of minor biases occurs for the estimate of α for all imputation strategies. Only the complete cases – in spite of the MAR mechanism – are virtually unbiased. One reason for the occurrence of biases for the intersect estimate might be that it is more sensitive to the effects induced by rounding/matching to the nearest neighbor.

Table 5.9: Relative bias: DS1 MAR small

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	-0.33	0.011	0.012	0.012	0.0037
$Var(Y)$	-0.1	-0.018	**-0.058**	**-0.057**	**-0.064**
$p(Y < 3)$	0.054	-0.0013	-0.0038	-0.0036	-0.002
$p(Y < 4)$	0.084	-0.0012	-0.0019	-0.0017	0.00014
$p(Y < 6)$	0.053	-0.0054	-0.0041	-0.0044	-0.0027
$\rho(X_1, Y)$	-0.012	-0.0063	-0.0016	-0.0016	-0.013
$\rho(X_2, Y)$	0.0013	0.012	0.0035	0.0041	0.019
α	0.0057	0.027	0.029	0.03	0.057
β_1	0.002	-0.0055	-0.0086	-0.0085	-0.022
β_2	3.8e-05	0.011	0.009	0.0095	0.019

[4] Biases exceeding 4% are in bold – apart from CC figures.

The second detailed bias table 5.10 shows the results for the big data set (3), again with MAR mechanism. It reflects the overall poor performance of ROV for misspecified imputation models. The way we generated the MAR values apparently benefitted the CC estimates figures of the 'true model' parameter estimates, but also PPMM and BBPMM yield reasonably good estimates.

Table 5.10: Relative bias: DS3 MAR BIG

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	-0.47	0.0029	-0.0022	-0.0023	-0.022
$Var(Y)$	-0.17	**0.077**	-0.0087	-0.0091	**-0.04**
$p(Y < 3)$	0.055	0.025	-0.00011	-7.8e-05	0.0014
$p(Y < 4)$	0.079	**0.063**	-5e-04	-0.00049	0.0015
$p(Y < 6)$	0.076	**-0.076**	0.00038	0.00033	0.0028
$\rho(X_1, Y)$	-0.0069	-0.0056	-0.00088	-0.0011	-0.023
$\rho(X_2, Y)$	0.012	-0.00082	0.0091	0.0088	0.012
α	-0.0064	**0.46**	**-0.09**	**-0.084**	**0.1**
β_1	5e-04	**-0.16**	-0.0022	-0.0036	**-0.1**
β_2	0.0033	**-0.23**	**0.045**	**0.042**	**-0.064**

By looking at the coverage for the same data situation (table 5.11) the consequences of the biased estimates are even more evident, as for ROV 3 out of 10 figures are zero, and for RPMM 2 out of 10 are below 10%. For comparison: the minimum coverage of BBPMM over 116 analysis figures is 81.4%.[5]

[5] the BBPMM coverages for the variance estimates of data set (2) are smaller, but we exclude them from our analysis, because they seem to be biased for all estimates (even the CC estimates under MCAR are around 82%, instead of the expected 95%).

Table 5.11: Coverage: DS3 MAR BIG

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	0	96.0	93.0	93.4	88.8
$Var(Y)$	80.8	88.4	87.4	87.4	84.8
$p(Y < 3)$	1.4	**30.0**	93.4	94.0	93.2
$p(Y < 4)$	0	**0**	94.2	94.8	93.6
$p(Y < 6)$	0	**0**	92.4	93.2	89.6
$\rho(X_1, Y)$	91.4	94.0	95.0	96.0	**7.6**
$\rho(X_2, Y)$	95.8	98.0	94.4	96.6	92.2
α	95.6	**44.4**	90.6	98.0	82.8
β_1	95.2	**0**	89.2	90.4	**3.2**
β_2	95.8	**42.6**	90.4	97.6	81.2

5.4.5 Analysis overview

The results from the preceding sections are summarized tables, based on an original total of 72 tables: twelve data situations à four imputation methods as well as the ′Before Deletion′ data sets and the ′Complete Cases′. Although we focused on bias and coverage in the analysis discussion, we also calculated the MSE and fraction of missing information (see section 2.2). Since the 72 tables contained four diagnostics and 10 analysis figures each, we can only provide an extract of all analyses that were carried out within this experimental design. In the remainder of the section we will therefore give an overview of our findings, grouped by the four evaluated diagnostics.

Bias:

In terms of bias all methods perform well for data sets (1) and (2) for most diagnostics. ROV slightly overestimates variances due to the rounding bias, and for data set (2) the bias for the three percentiles is ten times higher than for the PMM variants. The PMM variants tend to underestimate variances, especially for the ′small′ data sets, and RPMM also underestimates correlations throughout all data situations. For data set (3) the results are more clear-cut: All ten analysis figures for ROV are heavily biased, whereas the bias of the PMM variants, especially for BBPMM and PPMM , is on a similar level compared with the figures from data sets (1) and (2).

MSE:

The MSEs are in general similar, but less distinctive than the results for bias, and for data sets (1) and (2) all four imputation methods yield very similar figures.

Fraction of missing information:

For the fraction of missing information the results are less encouraging for the PMM variants. Multiple imputations under the correctly specified model *without correction* are used to calculate benchmarks for λ. ROV matches these benchmark values, but all three PMM variants have lower values. BBPMM comes close for the 'big' data sets, but PPMM and RPMM have a much lower fraction of missing information. The main reason for this result is a smaller *between* variance for the PMM variants. B is further reduced, if the number of potential donors is reduced (hence, λ is lower for the 'small' data set). Choosing M large for small data sets might yield better estimates for the fraction of missing information.

Coverage:

Throughout all analysis quantities for data set (1), ROV has a coverage between 92% and 98% and an average coverage for data set (1) of 95.58%. BBPMM has a slightly lower coverage rate (92.75%), followed by PPMM (91.37%) and RPMM with 86.24% (see also table 5.3). The only outlier among the analysis quantities from data set (1) is RPMM's coverage of $\rho(X_2, Y)$ for 'big' data sets with 53.6% (MCAR) and 51% (MAR). The small coverage in this case is mainly caused by a small *between* variance. In general, the coverage is influenced by previously identified effects for fraction of missing information *and* bias, which is why the coverage for ROV drops heavily for data set (3). For instance, the coverage of ROV for the 'big' data sets (MCAR and MAR) is zero in six out of twenty analysis quantities, and only the coverage for mean, variance and correlations is close to or higher than 90%. In comparison, 18 of the same 20 analysis figures for BBPMM and PPMM have coverages of more than 90%.

5.5 Summary of results

ROV is the best method for data set (1), but the problems for imputations under a misspecified model (data set (3)) are severe. RPMM yields relatively good results in terms of bias, but variances and correlations are underestimated. The underestimation of correlations makes also clear, why the naïve MI extension for PMM – taking the M nearest neighbors as donors – should not be considered, because biasing effects on bi- and multivariate associations are even more extreme for the M-nearest-neighbors strategy than for RPMM (see section 4.3). Both methods are super-imposing a uniform distribution over the predicted values for the imputation step by ignoring the different distances of potential donors. However, the main shortcoming of RPMM is the small *between* variance and the resulting low coverage for many analysis quantities.

Since it is not easy to avoid at least slight misspecifications of imputation models for survey data, we favor the two other PMM variants: PPMM and BBPMM. The analysis of RPMM demonstrates how difficult it is to utilize the robust properties of PMM within MI. Some diagnostics for the PMM variants show a relative decrease in performance (e.g. the fraction of missing information) for the 'small' data sets compared with the 'big' 2,000 sample data sets, but overall the results are satisfying, and the MAR results mirror the MCAR results. But overall analysis results suggest that PPMM and BBPMM yield very good results – not only in terms of consistency, but also with respect to coverage rates. Both are very similar methods, the only difference being that BBPMM is even 'less-parametric', because the posterior step for the model parameters is replaced by a Bayesian Bootstrap step. This seems to have a positive effect on the variation of results, since the fraction of missing information is closer to the benchmark results for data set (1). The higher between variance also yields higher coverage throughout all data situations and analysis figures. The more precise fraction of missing information and higher coverage for BBPMM aside, both methods yield very similar results.

A side aspect of this simulation study was the investigation of the effects of rounding continuously distributed random draws to discrete values. We tested rounding to the nearest *integer* within the Monte Carlo study for the 'big' data set (1) with 60 % of the values MCAR and almost exactly reproduced the predicted variance increase due to rounding (the expected variance bias is $\frac{1}{12}0.6 = 0.05$, and the simulation results yielded 0.0496 for the estimated bias). But as this method does not guarantee plausible values, it was not considered in the actual design. However, the overestimation of variances and covariances inflicted on the imputed data by rounding, is slightly decreased when values are (deterministically) rounded to the nearest observed value. This result seems intuitive, since extreme values which are out of bounds of the observed value range can not be imputed with ROV. The overestimation is constant and not depending on the (true) variance, which is why the rounding bias component will be more problematical, if the (true) variance is small, and the bias increase relatively higher. Therefore, we consider ROV to be a suitable Multiple Imputation technique for metric-scale discrete variables, *if* the imputation model is correctly specified, and the variance of the rounded variable is large.

Our conclusion is that, overall, BBPMM can be considered the 'best' method. ROV's coverage is better for data set (1), but building a correctly specified model in empirical survey data settings is difficult, and the decrease in coverage due to using BBPMM is small. On the other hand, if errors are non-normal or the imputation model is misspecified, ROV yields biased results and bad coverage, whereas the BBPMM imputations seem to be very robust to model misspecification. Because of higher coverages and better preservation of λ, BBPMM has the edge over PPMM in terms of correct MI inference. It has been rightfully stated by Duda et al.

(2001) that there is no free lunch in the context of general algorithm superiority, meaning that no algorithm can outperform any other algorithm for all analysis objectives. This is also true for our tested imputation algorithms (as displayed by the results), but since BBPMM was never ‘much worse’ than the best algorithm in this simulation study, and over all analysis figures better than any other technique over all tested diagnostics, we think that the findings imply that BBPMM has no inherent weakness and can be safely recommended for application to empirical data. Therefore, the next chapter describes the implementation of BBPMM into an FCS algorithm as basis for a fully-functional MI algorithm that can be applied to large data sets.

6

A comparison of MI algorithms using simulated multi-variable data sets and real survey data

Imputation techniques are used to retrieve efficient and consistent estimates based on incomplete data, but this should not happen at the expense of introducing bias due to imputation model misspecification. In the previous chapter we compared MI algorithms for (metric-scale) discrete survey data, and investigated their properties under various data conditions. The Bayesian Bootstrap Predictive Mean Matching (BBPMM) variant turned out to yield the most consistent estimates in this Monte Carlo experimental design. In a next step, we implement this strategy into a fully-functional algorithm for Multiple Imputation. Instead of applying the algorithm to a simulated data set, where only one variable is partially incomplete, we analyze an incomplete empirical survey data set, and compare the results of the BBPMM algorithm with counterparts based on the imputation software IVEware (Raghunathan et al. 2002), as well as to the complete-data estimates. Since the missing data are real, there is no way of comparing the results in terms of consistency. Therefore, we additionally impose an artificial missingness structure on 'jackknife' samples from the complete cases of the original data set. This procedure allows us to further analyze the imputation methods with respect to bias, MSE and coverage.

6.1 Imputation algorithm

As already mentioned, more recently the useful properties of PMM have been rediscovered (see e.g. Little & An 2004), and it was implemented in some Multiple Imputation algorithms such as MICE (van Buuren & Oudshoorn 1999) or the Hmisc library in R by Harrell (2006).

6.1.1 Chained Equations regression

The BBPMM method is combined with a so-called sequential regression or chained equation approach, and belongs to the FCS class described in section 2.4.2.

Starting solution

First, variables are sorted in ascending order according to their percentage of missing values. A starting solution is obtained by regressing each variable with missing values on the complete variables only (see fig. 6.1).

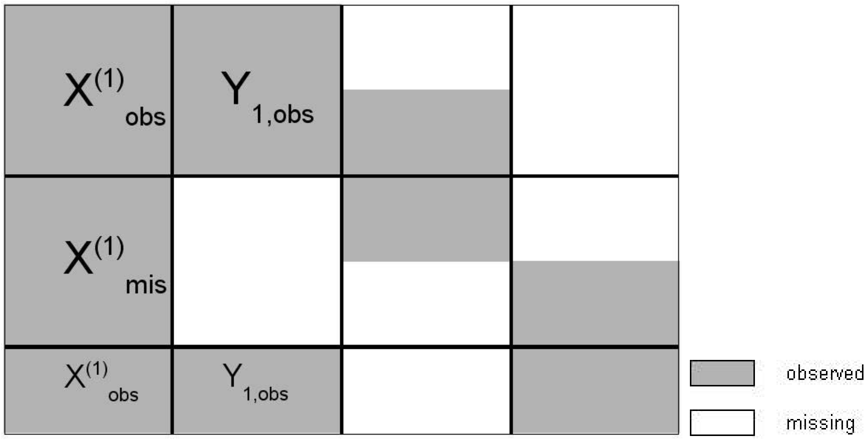

Figure 6.1: Starting solution for Y_1 based on completely observed variables

Let $\mathbf{X}_{\mathbf{obs}}^{(\mathbf{k})}$ denote the rows of the design matrix for the observed units of regressand $\mathbf{Y}_k$, with $k = 1 \ldots K$. Then

$$\widehat{\beta_k} = (\mathbf{X}_{obs}^{(k)\ T}\mathbf{X}_{obs}^{(k)})^{-1}\mathbf{X}_{obs}^{(k)\ T}Y_{k,obs}, \tag{6.1}$$

and estimates for the missing parts of variable $\mathbf{Y}_k$ are obtained by calculating

$$\hat{\mathbf{y}}_{k,mis} = \mathbf{X}^{(k)}_{mis}\widehat{\beta}_k. \tag{6.2}$$

Imputations are generated by additionally applying PMM as described in chapter 3.

In case all variables have missing values, the starting solution is generated using hot deck random draws to impute for missing values in variable Y_1, before variables Y_2 to Y_K are regressed on Y_1 (with the imputation model consisting of observed values for Y_1, where both Y_1 and Y_k are observed, and random draws for Y_1, where Y_k is observed, but Y_1 is not). Since one of the problems of chained equation approaches is that we do not know whether a joint distribution over all variables exists, we sort the variables in the data set by their respective number of missing values. Ideally, this would create a monotone missingness pattern (see section 1.2), for which the joint distribution exists, if the imputation model conditions on the variables to the 'left'. It is of course more likely that the pattern after sorting will be non-monotone, but the sorting routine at least approximates the monotone pattern.

Alternatively, a starting solution for the sequential regression can be generated using an ECM (Expectation - Conditional Maximization) algorithm (Meng & Rubin 1993). The difference to a classical EM-algorithm is that the expectation and maximization steps are carried out variable by variable through conditioning on all other variables. Note that the ECM step does not contain a PMM component. The reason for this is that a regression with PMM might never achieve convergence, although no stochastic element is involved. This effect is caused by outliers or model misspecification, which might catch the algorithm in an endless cycle. The ECM is iterated, until convergence for all model parameters is established. Let $\hat{\beta}^{[r]}$ denote a vector that includes the combined regression parameters of all K partially imputed variables within cycle r. Then convergence is assumed, if $|\hat{\beta}^{[r]} - \hat{\beta}^{[r-1]}| < 10^{-9}$. One might want to chose different random starting values for initial imputations, and compare the converged parameter results in order to check for convergence to local instead of global maxima. In the latter case the different starting values should converge to identical parameter estimates.

Chained equations

After a starting solution is created, Y_1 is regressed again on the completely observed variables *and* the partially imputed variables Y_2 to Y_K (see fig. 6.2). The imputation of Y_1 in cycle r is therefore based on the completely observed parts of variables Y_2 to Y_K as well as on the parts of these variables which were imputed in cycle $r-1$.

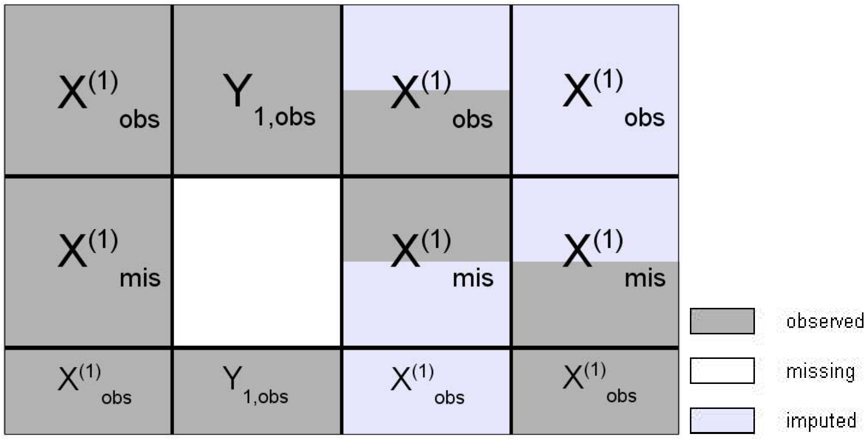

Figure 6.2: Sequential regression: Imputation of Y_1 based on completely observed and partially imputed variables

By letting $\mathbf{X}^{(\mathbf{k})}$ denote the $n \times p$ design matrix for regressand $\mathbf{Y}_k$, we get

$$\widehat{\beta}_k = (\mathbf{X}^{(k)\ T}\mathbf{X}^{(k)})^{-1}\mathbf{X}^{(k)\ T}Y_k, \tag{6.3}$$

and the predictors for variable Y_k are obtained by

$$\hat{y}_k = \mathbf{X}^{(k)}\widehat{\beta}_k. \tag{6.4}$$

The Predictive Mean Matching step is applied right after obtaining predictors for a particular variable with missing values rather than at the end of a cycle collectively.

6.1.2 Multiple imputations via Bayesian Bootstrap parameter draws

Our key objective is to integrate Predictive Mean Matching into an approximately proper multiple imputation algorithm. One of the difficulties in doing so lies with the nearest neighbor step of PMM. The approach we propose is to replace the P-step of Data Augmentation by a Bayesian Bootstrap step (Rubin 1981). Instead of drawing from a posterior distribution for all model parameters θ, we draw a sample $x_1^*, \ldots, x_n^*$ from the original data $X = (x_1, \ldots, x_n)$ as described in chapter 2.3.3. This new sample is used to obtain $\hat{\theta}^* = \hat{\theta}(x_1^*, \ldots, x_n^*)$ which is replacing the distributional draws. Thus, equations (6.3) and (6.4) are further modified to

$$\widehat{\beta}_k^* = (\mathbf{X}^{(k),*\ T}\mathbf{X}^{(k),*})^{-1}\mathbf{X}^{(k),*\ T}Y_k^*, \tag{6.5}$$

and

$$\hat{y}_k = \mathbf{X}^{(k)}\widehat{\beta}_k^*. \tag{6.6}$$

The advantage of this approach is that it does not rely on a normal distribution assumption for the estimated error terms. The I-step is also replaced by Predictive Mean Matching as described above. Bayesian Bootstrapping and PMM are finally embedded in the sequential regression approach, and we have the means to conduct BBPMM on large data sets with non-monotone missing-data patterns. We have already mentioned that the existence of a joint (posterior) distribution cannot be verified, which is why convergence in distribution can usually not be shown either. However, the PMM properties seem to create solutions within one or two cycles which show no signs of autocorrelation to the imputation model parameters of the starting solution.

6.2 Description of the empirical data set

A data set based on a survey of drinking habits among Michigan-licensed adults (see Bingham et al. 2007) is used to examine the effects of the different imputation methods. The corresponding telephone interviews were conducted in 1999 and in 2000 as part of a longitudinal study. The data set has a total of 4,199 interviews, out of which n_{cc}=2,278 are complete and 1,921 have missing information for at least one of eight analysis variables considered for the imputation analysis.

The eight analysis variables:

- AQF/AQF_2 measure the *alcohol quantity frequency* in 1999 and 2000 respectively. The variable is the product of periodicity (0 = never drinking to 4 = four or more times a week drinking) and quantity (0 = not drinking to 5 = 10 or more drinks). The integer values of the two variables therefore range from 0 to 20.

- AUDIT/AUDIT2 are the variables based on the 10-item **A**lcohol **U**se **D**isorders **I**dentification **T**est (Saunders et al. 1993) obtained in 1999 and 2000. The underlying items measure alcohol dependence, consumption patterns and personal/social problems. Each item can take a value between 0 and 3, and the AUDIT score is the sum over all 10 items. Although 30 is the theoretical maximum value, the highest empirical score is 29 (0 being the mode with 358 observations)

- DRK_DRI1/DRK_DRIV describes perceived risks of drink/driving, like getting stopped for drink/driving or being involved in a car crash. In total there are six such items, all of them ranging from 'very likely' (=1) to 'very unlikely' (=4). The final variable is a scale score with values between 0 and 12.8 (the minimum step of 0.2 between two scores indicates that probably only five variables were used to construct the score). The variable names do not follow the same logic as their predecessors, but again, they describe the results from the 1999 survey and the follow-up in 2000.

- COMPETE comprises competitive attitudes towards driving. The original number of five statements with four different categories is condensed into one single index by averaging over all five statements. The range for this variable is 1 to 4.

- CONSEQNC, finally, is a z-transformed quasi-continuous variable consisting of several indicators. It measures awareness of the consequences of drink/driving.

The number of missing observations varies over the eight analysis variables:

- AQF: $n_{mis} = 10$
- AUDIT: $n_{mis} = 15$
- COMPETE: $n_{mis} = 30$
- DRK_DRI1: $n_{mis} = 60$
- CONSEQNC: $n_{mis} = 1857$
- AQF_2: $n_{mis} = 1861$
- DRK_DRIV: $n_{mis} = 1867$
- AUDIT2: $n_{mis} = 1876$

6.3 IVEware

IVEware is a SAS-based imputation algorithm[1] that performs multiple imputations of missing values using a sequential regression approach and conditional draws from the posterior predictive distribution (Raghunathan et al. 2002). Unlike the sequential BBPMM imputation algorithms described in the section 6.1.2 it is fully parametric. Different variable types are imputed with different GLM variants and error assumptions: for continuous variables a linear model with normally distributed errors is used, whereas count data are imputed via a Poisson link model. If the data set features categorical variables, a multinomial logit model

[1] Srcware is a stand-alone version of IVEware that runs on MS-Windows or Linux/Unix platforms.

is used to impute missing values in those categorical variables (if a variable is binary the multinomial logit is automatically reduced to a binomial logit model). A special case is the occurrence of mixed-type variables. Some variables have both, a categorical and a continuous component. For instance the questions “do you smoke?”, “if yes: how many per day” can be expressed within one mixed-type variable.[2] Typically these variables have at least one discrete value that captures a considerable margin of all observations. IVEware imputes mixed-type variables by adopting a two-step procedure: In a first step the binary aspect is imputed (categorical vs continuous), followed by the second step: imputation of the pre-identified continuous cases using the normal model.

While the Bayesian Bootstrap Predictive Mean Matching algorithm will automatically impute observed (and therefore plausible) values, IVEware provides the option to specify ranges for valid values. Additionally, all variables in the data set are subject to classification into ‘categorical’, ‘count’, ‘mixed’ or ‘continuous’. Although AQF/AQF_2 and AUDIT/AUDIT2 are count variables which can be imputed via the Poisson link (at least, if we add 1 to all values in order to get rid of the zeros), these variables were treated as ’continuous’ (with the observed minima and maxima to define the corresponding bounds). The reason for this deliberate mis-classification was the relatively high number of categories and the shape of the empirical distributions. CONSEQNC and COMPETE were likewise considered to be ’continuous’, whereas DRK_DRIV and DRK_DRI1 were considered ’mixed’. One consequence of the decision to impute count variables with an identity link and normal errors is the occurrence of implausible non-integer values, but this would have been the case for COMPETE and DRK_DRIV/DRK_DRI1 anyway, since IVEware has no means to preserve the domain created by the constructed indices. For the considered analysis quantities this is only a minor drawback, since we are primarily interested in moments and similar aggregated figures.

[2] Although the ’continuous’ component is never genuinely continuous, if the data stem from a questionnaire-based survey.

6.4 Analysis based on the original data set

6.4.1 Data preparation

IVEware assumes normally distributed error terms for all continuous variables. We examined histograms and Q-Q plots to identify suitable transformation functions. $y = x^{1/3}$ for AQF/AQF_2 and $y = 4 - x$ for COMPETE established approximate normality for these variables, and all IVEware imputations were performed on this partially transformed data set. Figure 6.3 illustrates the effects of the transformation for COMPETE.[3]

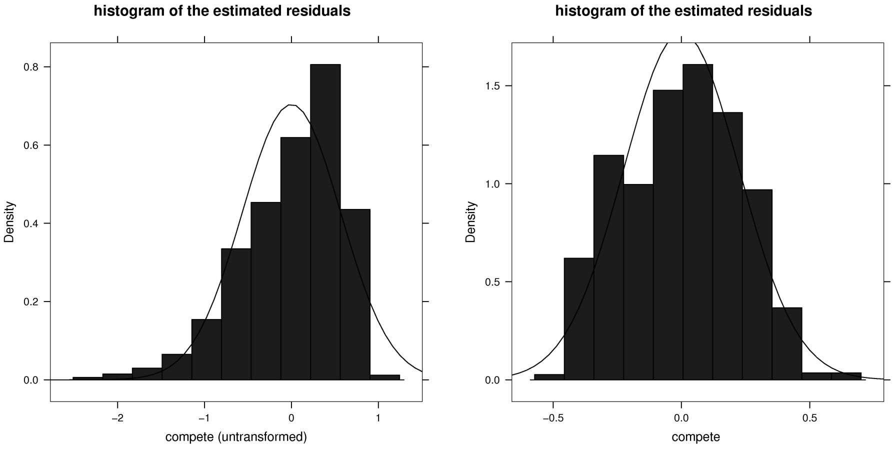

Figure 6.3: The variable COMPETE before and after transformation

Since BBPMM is supposed to be robust to model misspecification, it does not rely on normally distributed error terms, and so there is no need to apply the above transformations. Then again, analysis results are more difficult to compare if a different data base is used, which is why we decided to run BBPMM on both, the transformed and the original (untransformed) data set.

All MI results are based on $M = 20$ partially imputed data sets. For IVEware we stored every tenth iteration as imputed data set for the MI analysis. A finding from prior work with sequential Predictive Mean Matching algorithms is that results do typically not converge – even if no stochastic component is involved – and that no

[3] The diagnostic plots can also be found in appendix A.

autocorrelation can be identified for two subsequent iterations. As a compromise between computational speed and synchronism to IVEware we decided to store every fifth iteration for the BBPMM imputations.

6.4.2 Methods and quantities of interest

We compare results for the complete cases of the original data set, the BBPMM results based on the original and the transformed data set and the IVEware results based on imputations of the transformed data set. Throughout the analysis the following abbreviations are used to distinguish between the four methods:

- Methods: CC (Complete Cases), BBPMMO (MI results of the sequential Bayesian Bootstrap Predictive Mean Matching using the untransformed original data set), BBPMMT (MI results of the sequential Bayesian Bootstrap Predictive Mean Matching using the transformed data set), IVE (MI IVEware results using the transformed data set)
- Variables: AQF (AQF), AUDIT (AUD), COMPETE (COM), DRK_DRI1 (DR1), CONSEQNC (CON), AQF_2 (AQ2), DRK_DRIV (DRK), AUDIT2 (AU2)

We investigated means for all eight variables, but due to the mixed-type character of the drink/drive variables, we compared the estimates for $p(X = 0)$ and $E(X|X > 0)$. Additionally we compared the parameter estimates for two regressions. The first one is an ordered logit model, where the variable *DRK_DRIV* was recoded into an ordinal scale variably Y with five categories ranging from 1 to 5:

$$\ln \frac{\pi(Y_i \leq j)}{\pi(Y_i > j)} = (\alpha - \alpha_j) + \beta_1 COM_i + \beta_2 CON_i + \epsilon_{1,i}, \forall i = 1, \ldots, n \qquad (6.7)$$

and $j = 1, ..., 5$.

The second regression model estimated the ratio of DRK_DRIV and DRK_DRI1

$$\frac{DR1_i + 1}{DRK_i + 1} = \gamma + \delta_1 \frac{AQ2_i + 1}{AQF_i + 1} + \delta_2 \frac{AU2_i + 1}{AUD_i + 1} + \epsilon_{2,i} \forall i = 1, \ldots, n.^{4} \qquad (6.8)$$

6.4.3 Results

Table 6.1 shows the results for the proportion and mean estimates as well as the lower and upper bound of the central 95% confidence interval.

Table 6.1: Proportion and mean estimates based on the different methods

		estimate	lower bound	upper bound
$p(DR1 = 0)$	CC	0.4807	0.4602	0.5012
	BBPMMO	0.4755	0.4603	0.4908
	BBPMMT	0.4759	0.4608	0.4911
	IVE	0.4757	0.4605	0.4909
$p(DRK = 0)$	CC	0.5136	0.4931	0.5341
	BBPMMO	0.4926	0.4739	0.5113
	BBPMMT	0.4958	0.4759	0.5157
	IVE	0.501	0.4826	0.5195
$E(AQF)$	CC	3.236	3.107	3.364
	BBPMMO	3.388	3.349	3.427
	BBPMMT	3.387	3.348	3.426
	IVE	3.386	3.346	3.425
$E(AUD)$	CC	4.09	3.928	4.252
	BBPMMO	4.218	4.17	4.265
	BBPMMT	4.215	4.168	4.263
	IVE	4.216	4.169	4.263
$E(COM)$	CC	3.443	3.42	3.466
	BBPMMO	3.442	3.435	3.448
	BBPMMT	3.441	3.434	3.448
	IVE	3.44	3.433	3.447
$E(DR1 \vert DR1 > 0)$	CC	2.466	2.323	2.608
	BBPMMO	2.504	2.47	2.537
	BBPMMT	2.504	2.473	2.536
	IVE	2.512	2.48	2.543
$E(CON)$	CC	0.001749	-0.03083	0.03433
	BBPMMO	0.1245	0.1136	0.1354
	BBPMMT	0.1356	0.1226	0.1487
	IVE	0.05464	0.02818	0.08109
$E(AU2)$	CC	3.723	3.575	3.872
	BBPMMO	3.954	3.852	4.056
	BBPMMT	4.034	3.953	4.114
	IVE	4.173	4.095	4.251
$E(DRK \vert DRK > 0)$	CC	2.008	1.884	2.133
	BBPMMO	1.971	1.868	2.073
	BBPMMT	2.011	1.937	2.085
	IVE	2.375	2.3	2.449
$E(AQ2)$	CC	2.917	2.803	3.031
	BBPMMO	3.142	3.063	3.22
	BBPMMT	3.178	3.114	3.241
	IVE	3.195	3.087	3.303

The small percentage of missing data equates to very similar results for the mean estimates of AQF, AUDIT, COMPETE and DRK_DRI1 for IVEware and the BBPMM variants. The CC results for COMPETE are close to the imputation variants, but the deviations for the other three 'wave one' variables are bigger. Since

[4] $i = 1, \ldots, n_{cc}$ for the CC variant in (6.7) and (6.8).

the CC estimates are based on hardly more than 50%, and since there are strong indications that the missing information is not MCAR, the imputation variants are likely to be closer to the (unavailable) complete-data estimates. Among the 'wave two' variables with higher percentages of missing data, the deviations of the mean estimates are generally bigger. The mean estimates for CONSEQNC vary so strongly that the confidence intervals are non-overlapping, except for the two BBPMM variants.[5] But deviations can also be found for AUDIT2, AQF_2 and the mean estimate for non-zero values of DRK_DRIV.

At first glance, the deviations between the four methods seem to be bigger for the model parameter estimates in table 6.2 than for the proportion and mean estimates. But under closer investigation this can be explained by the (relatively) larger standard errors. The generally higher overlap among the method-specific confidence intervals confirms this assumption.

Apart from mean and parameter estimates we also examined bivariate correlations among the eight variables in the data set. The 28 correlations and the corresponding 95% confidence intervals are plotted in figure 6.4.

The bounds for ρ were calculated using Fisher's transformation $z = 0.5 \cdot ln[(1+\rho)(1-\rho)^{-1}]$. Using variance $(n-3)^{-1}$ and taking the inverse $z^{-1} = tanh(z)$ allows us to estimate approximate MI confidence intervals. Generally the CIs are wider if the correlations are close to zero. For instance, $\hat{\rho}(AUD, AQF)$ is quite close to one, and therefore all methods yield small CIs. The correlation estimates between CONSEQNC and DRK_DRIV show a heavy 'outlier' for IVEware. Another interesting finding are the relatively large differences between the two BBPMM variants for $\hat{\rho}(AU2, AQ2)$ and $\hat{\rho}(DRK, AQ2)$. The results from chapter 5 suggested that BBPMM is fairly robust to model misspecification, yet two different imputation models (original/transformed) yield considerably different estimates for these two correlations. The detailed tables for correlations are given in the appendix section of this thesis.

[5] Which are almost always very close to each other.

Table 6.2: Parameter estimates based on the different methods

		estimate	lower bound	upper bound
$\alpha_{1\|2}$	CC	-1.107	-1.632	-0.5822
	BBPMMO	-1.191	-1.683	-0.6989
	BBPMMT	-1.365	-1.865	-0.8648
	IVE	-1.079	-1.586	-0.5716
$\alpha_{2\|3}$	CC	-0.4021	-0.9253	0.1211
	BBPMMO	-0.5083	-0.9977	-0.019
	BBPMMT	-0.7079	-1.203	-0.2134
	IVE	-0.6128	-1.12	-0.1056
$\alpha_{3\|4}$	CC	0.1861	-0.3373	0.7095
	BBPMMO	0.05902	-0.4248	0.5429
	BBPMMT	-0.1592	-0.661	0.3426
	IVE	-0.2878	-0.7986	0.223
$\alpha_{4\|5}$	CC	1.238	0.7093	1.767
	BBPMMO	1.066	0.5817	1.549
	BBPMMT	0.8211	0.3258	1.316
	IVE	0.5578	0.02944	1.086
β_1	CC	-0.4626	-0.6101	-0.3151
	BBPMMO	-0.4744	-0.617	-0.3318
	BBPMMT	-0.516	-0.6554	-0.3767
	IVE	-0.4114	-0.5567	-0.2661
β_2	CC	2.973	2.655	3.292
	BBPMMO	2.515	2.182	2.848
	BBPMMT	2.233	1.938	2.528
	IVE	2.509	2.111	2.908
γ	CC	1.704	1.625	1.783
	BBPMMO	1.753	1.68	1.827
	BBPMMT	1.72	1.64	1.801
	IVE	1.772	1.673	1.871
δ_1	CC	-0.2	-0.2995	-0.1005
	BBPMMO	-0.1968	-0.2963	-0.09734
	BBPMMT	-0.1399	-0.2148	-0.065
	IVE	-0.2645	-0.3331	-0.1959
δ_2	CC	-0.1008	-0.1825	-0.01924
	BBPMMO	-0.1	-0.1782	-0.02189
	BBPMMT	-0.1395	-0.2056	-0.07339
	IVE	-0.105	-0.1458	-0.0643

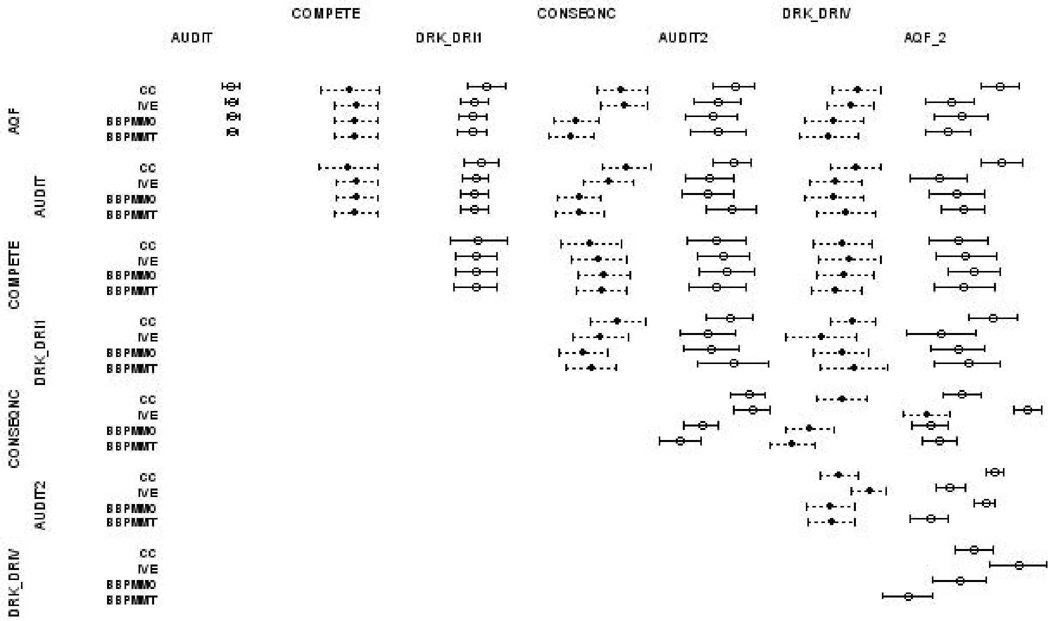

Figure 6.4: Bivariate correlations and confidence intervals for all eight variables

6.4.4 Summary

The analysis results for the four different methods vary partially so strongly that the confidence intervals have very small or no intersections. The differences between the CC results and the MI variants can be explained, if we assume that MCAR does not hold (propensity scores based on AQF, AUDIT, COMPETE and DRK_DRI1 calculated for the overall complete cases as well as the complete cases of these four variables support this assumption). However, the differences between the results based on IVEware and on the BBPMM variants lead to the conclusion that the selection of an imputation method has an influence on the analysis results as well. This analysis faces the general dilemma that we do not know the 'true values' of the analyzed quantities of interest. Therefore we can not make any statement with respect to gain in efficiency or reduction of biases. The following section tries to overcome this by using the complete cases as reference data set in a Monte Carlo study. Missingness is artificially induced and afterwards imputed again.

6.5 Analysis based on Monte Carlo simulations

6.5.1 Design of the simulation study

We use the 2,278 complete cases and create a subsample consisting of 280 random draws without replacement. Then we superimpose an MAR pattern on this data set by applying $\binom{8}{2} = 28$ different logit models (similar to those from the previous chapter), and eliminate 30% of the 280 observations for each variable. We repeat this procedure 500 times and obtain 500 data sets à 280 observations with different MAR patterns. Analogously to the previous section we compare the results based on the complete cases (i.e. ′complete cases′ after artificially removing parts of the data) with the MI results using IVEware and the two BBPMM variants ($M = 20$ for all MI methods), and focus again on the ten proportion/mean and the nine parameter estimates from the regression models in (6.7) and (6.8). The original CC data set from which the 'jackknife'[6] samples are taken, provides the 'true values' that allows us to estimate bias, MSE and coverage for the 19 analysis quantities, and are given in table 6.3.

Table 6.3: True values for all proportions/means and model parameters

	Proportions/Means		Parameters
$p(DR1 = 0)$	0.4807	α_1	-1.107
$p(DRK = 0)$	0.5136	α_2	-0.4021
$E(AQF)$	3.236	α_3	0.1861
$E(AUD)$	4.09	α_4	1.238
$E(COM)$	3.443	β_1	-0.4626
$E(DR1\|DR1 > 0)$	2.466	β_2	2.973
$E(CON)$	0.001749	γ	1.704
$E(AU2)$	3.723	δ_1	-0.2
$E(DRK\|DRK > 0)$	2.008	δ_2	-0.1008
$E(AQ2)$	2.917		

[6] Note that genuine jackknifing means that we leave out only one unit per iteration run.

6.5.2 Results

We calculate two different measures for all methods to compare the different mean and regression estimates: $Measure1 = bias^2/MSE \times 100\%$ can be loosely described as a 'standardized bias share', and $Measure2 = var(BD)/MSE \times 100\%$ reflects some kind of 'efficiency' measure, where $var(BD)$ is the 'Before Deletion' variance averaged over all jackknife runs. Since the MSE can be decomposed into variance plus squared bias, and since, generally, $var(BD)$ should be smaller than the variance of any estimator that is partially multiply imputed,[7] Measure2 can take values between zero and one. The 'Coverage' contains the information, whether the 'true value' lies within the central 95% confidence interval of the given method. Table 6.4 shows the averaged values over all 500 Monte Carlo runs – separated for proportions/means (M) and the regression parameters (R). Again, more detailed results can be found in appendix B 2.5 and 2.6.

Table 6.4: Analysis figures for the Jackknife simulations

		Measure1	Measure2	Coverage
CC	M	23.52	29.28	86.92
CC	R	24.06	16.65	76.71
BBPMMO	M	1.25	42.48	93.48
BBPMMO	R	9.557	28.8	92.13
BBPMMT	M	0.978	42.98	93.18
BBPMMT	R	15.33	30.98	91.27
IVE	M	9.995	37.49	92.7
IVE	R	20.7	31.82	87.4

Ideally, values should be close to zero for Measure1, close to 100% for Measure2 and close to 95% for the coverage. We can see that all MI methods outperform the complete cases. Using the available information in the observed part of the data allows the MI methods to yield better estimates than a simple analysis of the remaining complete cases. Especially the BBPMM variants reduce the 'M' values of Measure1 almost back to zero, but also the values for the parameter estimates are more than halved for BBPMMO, and considerably reduced for BBPMMT com-

[7] A notable exception are cases of 'superefficiency' (Rubin 1996).

pared with the CC figures. IVEware yields slightly worse results for Measure1. This is also true in a less distinct way for the proportion/mean estimates of Measure2. The parameter estimates of Measure2, however, are almost on the same level for all methods, with IVEware even edging the BBPMM variants. The coverages for the 'M' estimates are quite close to the ideal 95% for all MI methods. But IVEware has a slightly lower value for 'R' than the two BBPMM variants. Figure 6.5 displays the respective coverage figures in a more detailed way.

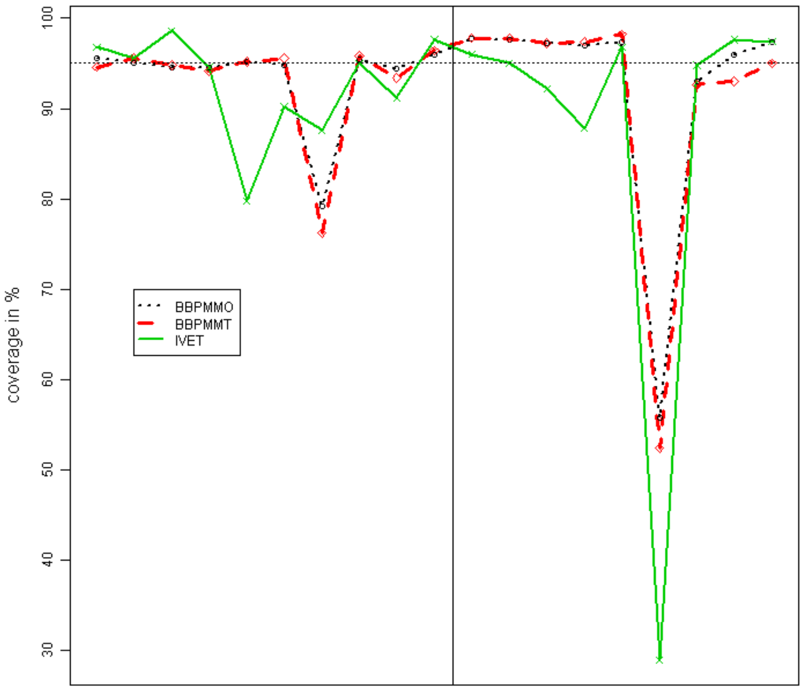

Figure 6.5: Coverages for all three methods over 19 analysis quantities

The vertical line in the middle of the plot marks the intersection between 'M' and 'R' estimates, and the dotted horizontal line the benchmark 95%. If we consider only 'outliers' with values below 90%, IVEware performs worse for the mean estimate of COMPETE (79.8% vs. 95.2% for both BBPMM variants) and better for the mean estimate of CONSEQNC (87.6% coverage vs. 79.2/76.2% for BBPMMO/T). The estimate for IVEware's α_4 of (6.7) is below 90% (87.8%), but the overall most striking outlier is the coverage for β_2 of the first regression model: 55.8/52.4% for BBPMMO/T and only 29.0% for IVEware. This parameter estimate for COMPETE is the only heavy outlier, and although the BBPMM variants yield higher coverages than IVEware, the tendency is the same. In contrast, all methods give very high coverages for β_1.

Unlike the MAR mechanism used in the Monte Carlo experimental design with simulated data, the mechanism used in this study yields partially very low coverages for the complete cases, as table 6.5 demonstrates.

Table 6.5: CC: Average parameter estimates from 500 jackknife samples

	$E(Y)$	$Bias$	MSE	lower bound	upper bound	coverage	λ
α_1	-1.2	-0.073	2	-3.2	0.84	0.96	NA
α_2	-0.73	-0.32	2.1	-2.7	1.3	0.95	NA
α_3	-0.43	-0.62	2.4	-2.4	1.6	0.91	NA
α_4	-0.077	-1.3	3.8	-2.1	1.9	0.78	NA
β_1	-0.25	0.21	0.21	-0.82	0.33	0.91	NA
β_2	0.97	-2	4.2	0.45	1.5	0.012	NA
γ	1.5	-0.25	0.22	1	1.9	0.55	NA
δ_1	-0.12	0.082	0.27	-0.75	0.52	0.9	NA
δ_2	-0.14	-0.037	0.21	-0.69	0.41	0.93	NA

In comparison the BBPMM results for the untransformed data yield much higher average coverages, and smaller bias and MSE estimates (see table 6.6.

Table 6.6: BBPMMO: Average parameter estimates from 500 jackknife samples

	$E(Y)$	$Bias$	MSE	lower bound	upper bound	coverage	λ
α_1	-1.3	-0.15	1.8	-3.3	0.79	0.98	0.45
α_2	-0.59	-0.19	1.8	-2.6	1.5	0.98	0.45
α_3	-0.041	-0.23	1.8	-2.1	2	0.97	0.45
α_4	0.93	-0.31	1.9	-1.1	3	0.97	0.45
β_1	-0.45	0.012	0.15	-1	0.13	0.97	0.44
β_2	2	-0.94	1.3	0.98	3.1	0.56	0.54
γ	1.7	0.019	0.064	1.4	2.1	0.93	0.5
δ_1	-0.16	0.042	0.086	-0.57	0.26	0.96	0.58
δ_2	-0.16	-0.056	0.063	-0.5	0.19	0.97	0.55

An interesting finding is that the corresponding IVEware table 6.7 sometimes have a slightly lower estimated fraction of missing information $\widehat{\lambda}$. Usually, the IVEware confidence intervals are slightly bigger than the BBPMM counterparts, but not in all cases: After analyzing the results from the experimental design in section 5.4 of the previous chapter, we assumed that BBPMM tends to underestimate the between variance, when applied to small data sets. But for some cases in table 6.6 (e.g. δ_1), BBPMM's CI length and $\hat{\lambda}$ are bigger than IVEware's counterpart. This suggests that at least sometimes BBPMM yields higher between variances than fully-parametric MI algorithms.

Table 6.7: IVEware: Average parameter estimates from 500 jackknife samples

	$E(Y)$	$Bias$	MSE	lower bound	upper bound	coverage	λ
α_1	-1.5	-0.35	2.2	-3.6	0.68	0.96	0.45
α_2	-0.93	-0.53	2.4	-3.1	1.2	0.95	0.45
α_3	-0.57	-0.75	2.6	-2.7	1.6	0.92	0.45
α_4	0.27	-0.97	3	-1.9	2.4	0.88	0.46
β_1	-0.5	-0.037	0.18	-1.1	0.12	0.97	0.46
β_2	1.9	-1	1.4	1.1	2.8	0.29	0.52
γ	1.8	0.059	0.064	1.4	2.1	0.95	0.49
δ_1	-0.15	0.053	0.034	-0.43	0.14	0.98	0.49
δ_2	-0.18	-0.082	0.033	-0.43	0.059	0.97	0.41

The discrepancy between the IVEware and BBPMM results is more obvious for the estimated MSE and bias, but even without the extremely low value for β_2, BBPMM has higher average coverages than IVEware, although the sample size is similar to the small data sets from the MC experimental design.

6.5.3 Summary

The two BBPMM variants ('transformed'/'original' data set) generally yielded very similar results. This can be interpreted as a confirmation of the hypothesis that the robustifying properties of PMM alleviate the biasing effects of misspecified imputation models, since the results of BBPMMO/T are almost indistinguishable. The biased results for β_2 show that this hypothesis does not always hold, but we believe it to be true for all cases, where the misspecified model is a monotonous function of the 'true' model.

An obvious handicap of any nearest neighbor-approach is a small number of potential nearest neighbors, because this increases the average distances of the matched pairs. Considering that the data sets in the simulation study comprised only $n_{obs} = 196$ (70% of 280) potential donors per variable, the overall results for the BBPMM variants are quite encouraging.

6.6 Discussion

Previous results from chapter 5 with simulated data have shown that the combination of Bayesian Bootstrap and Predictive Mean Matching works well as a replacement for the corresponding P- and I-steps of Data Augmentation, and that the *fraction of missing information* (Rubin 1987*a*) is nearly identical to the expected fraction (based on a parametric MI under a correctly specified model). The findings from this study with respect to coverages and their interpretation draw upon these analyses, and therefore we can conclude that the higher coverage rates of the BBPMM variants compared with IVEware are not caused by overly conservative confidence intervals.

We tested means and regression parameter estimates, and the results suggest that the BBPMM variants outperform the purely parametric IVEware algorithm, but on the other hand, the test design was based on a particular data set, and the tested quantities of interest are a small fraction of all potential analysis figures. More extensive analysis has to be carried out to validate the findings from this study – both with simulated data (e.g. using deliberately misspecified imputation models) as well as with empirical data.

7

BBPMMimpute: An *R* package for handling missing data in large survey data sets

7.1 Introduction

R is an open source statistical software based on the programming language *S* that has become wide-spread in the statistical community, with a number of contributed packages – by the date this thesis was submitted – approaching the 2000 threshold.

The packages *Hmisc* (Harrell 2006) and *mi* (Gelman et al. 2008) as well as ports of Joseph Schafer's NORM and MIX software already feature Multiple Imputation algorithms. The *aregImpute* function from Frank Harrell's package is even very similar to our algorithms, since it combines Bootstrapping and Predictive Mean Matching (and additive regression splines), where PMM can be applied to metric-scale variables. *aregImpute* was the second software after MICE to implement PMM into an MI algorithm.

The innovation of the MI algorithms in the *BBPMMimpute* package is that PMM is used for any variable type, including unordered categorical variables (called 'factors' in *R*). The source code of version 0.1-1 can be found in appendix C.

7.2 BBPMM.col – ’column-wise’ Multiple Imputation

7.2.1 Description

The *BBPMM.col* function is based on the algorithm described in section 6.1: Because missing-data imputation is carried out variable-by-variable, we will refer to this function as ’column-wise’ multiple imputation.

The algorithm re-groups partially incomplete variables by their number of missing values in ascending order. Initial imputations are generated by . In order to emulate a monotone missing-data pattern as well as possible, variables are sorted by rate of missingness (in ascending order). If no complete variables exist, the least incomplete variable is imputed via hot-deck. The starting solution then builds the imputation model using the observed values of a particular variable y_t, and the corresponding observed or already imputed values of the variables $[y_1 \ldots y_{t-1}]$ (i.e. all variables with fewer missing values than y_t).

The iterations of the chained equations algorithm re-start from the starting solution, after an imputed data set was stored. Due to the PMM element in the algorithm, the auto-correlation of the iterations is virtually zero. Therefore, a burn-in period is not required, and there is no need to administer ’high’ values (>20) to the parameter that governs the number of iterations either.

Unordered categorical variables are imputed via a multinomial logit model (the *multinom* function of the *nnet* library). Predictive Mean Matching for these variables is carried out using the method described in section 3.3.

7.2.2 Arguments

data A partially incomplete data frame or matrix.

M Number of multiple imputations. If $M = 1$, no Bayesian Bootstrap step is carried out.

n.iter The Number of iterations of the chained equations algorithm before the imputed data set is stored.

out.file A character string that specifies the path and file name for the imputed data sets. If M>1, the extension '_<m>' is added before the file extension to mark the number of the imputed data set. If *out.file*=NULL (default), no data set is stored.

ignore A character or numerical vector that specifies either column positions or variable names that are to be excluded from the imputation model and process, e.g. an ID variable. If *ignore*=NULL (default), all variables in *data* are used in the imputation model.

var.type A character vector that flags the class of each variable in *data* (without the variables defined by the *ignore* argument), with either 'M' for metric-scale or 'C' for categorical. The default (NULL) takes over the classes of *data*. Overruling these classes can sometimes make sense: e.g. an ordinal-scale variable is originally classified as 'factor', but treating it as metric-scale variable within the imputation process might be a better choice (considering the robust properties of PMM to model misspecification)

eff.measure Calculates the Goodness-of-Fit measure described in section 3.4 to monitor the quality of the nearest neighbor matches.

maxit Imported argument from the *nnet* package that specifies the maximum number of iterations for the multinomial logit model estimation.

verbose The algorithm prints information on imputation and iteration numbers.

... Further arguments passed to or from other functions.

7.2.3 Values

impdata The imputed data set, if M=1, or a list containing M imputed data sets.

mis.overview The percentage of missing values per incomplete variable.

eff.measure A matrix containing efficiency measures for all M data sets (rows) and incomplete metric-scale variables. Completely observed variables and factors are set to 'NA'.

ind.matrix A matrix with the same dimensions as *data* minus *ignore* containing flags for missing values.

7.2.4 Troubleshooting

A tricky consequence of the Bayesian Bootstrap step is the possibility of incompatible imputation models. This event occurs, if:

- one or several factor levels (i.e. categories of an unordered categorical variable) of one or several left-hand-side variables are missing in the bootstrapped data set – irrespective of the variable type of the right-hand-side variable

- the right-hand-side variable is a factor, and one or several of its levels are missing in the bootstrapped data set

In either case the solution to this problem is setting the corresponding parameters to zero. Since factors are imputed with a multinomial logit model, the complete parameter vector of a missing level has to be treated accordingly.

A final issue that needs to be addressed is the ′conundrum′ of chained equations approaches: When we run a regression of y_t on $[y_1 \ldots y_{t-1}, y_{t+1} \ldots y_T]$, y_t ideally is a linear combination of the regressors. But when we run the subsequent regression of y_{t+1}, the regressors ideally are completely independent of each other. Currently there is no ′safety catch′ for perfect multi-collinearity, but updates of the algorithm will contain the option to use ′least angle regression′ for the imputation models to keep them estimable. For the time being we can but refer to Goldberger (1991), who coined the mock-expression *micronumerosity*, meaning that multi-collinearity – just like small data sets – tends to increase the standard errors of the model parameters, but the estimators themselves are still consistent.

7.2.5 Logical inconsistencies

Unlike IVEware the *BBPMM.col* function does not (yet) handle filter variables. Ignorance of filters can lead to logical inconsistencies in the data, e.g. non-smokers with a daily cigarette consumption or underage driver's licence holders. There are two ways to deal with such occurrences:

1) we impute the data irrespective of filters, and 'repair' logical inconsistencies manually after the imputation step (e.g. by re-entering filter-generated missing values)

2) in a first step we omit all filter-dependent variables, and impute subsequently the subsets of the data set according to the imputed values of the filter variables.

7.3 BBPMM.row – 'row-wise' Multiple Imputation

7.3.1 Description

Unlike the sequential regression-based algorithm described in the previous section, the *BBPMM.row* function (multiply) imputes not only one variable at a time, but all variables with identical missingness patterns simultaneously. This property predestines the algorithm for Multiple Imputation of missing-by-design patterns as described in section 1.2, where usually a large number of variables have identical missingness patterns. Data preparation steps comprise the identification of unique patterns ('blocks'), and the exclusion of constant columns.

The challenging part of a PMM over more than one variable is how to relate the distances of predicted means of different variables to each other. The *BBPMM.row* function follows closely the algorithm described in (Little 1988*a*). A Mahalanobis distance metric is used to identify a 'global' nearest neighbor over all Y variables within a block. If several elements are minimal, a random draw among the nearest neighbors selects the final donor. Note that this function is suited for monotone and data fusion patterns as described in section 1.2, but not for SQS designs (because the algorithm is not based on chained equations).

The covariance matrix of the residuals from the regression of $\mathbf{Y}$ on the completely observed variables $\mathbf{X}$ is transformed into a diagonal matrix by using conditional regressions. This simplifies the usage of manual weights.

Unlike *BBPMM.col* this function is only suited for metric-scale variables. Theoretically, an unordered factor in y with L levels could be recoded into $L-1$ dummies, and we could use a linear model – the incompatible value space would not matter, as an overall nearest neighbor will 'donate' its complete (plausible) vector y_i. But replacing a single unordered factor by $L-1$ variables would exaggerate the influence on the selection of a nearest neighbor of this variable. Clearly, choosing only one category dummy is not a sensible option either.

The benefits of imputing a complete vector of missing values over variable-by-variable imputation are:

- The distribution of the 'Y' variables is potentially better preserved.
- Particularly, we avoid logical inconsistencies (among Y variables) as described in the above section.
- Data that are nearly impossible to be directly imputed can be imputed via (artificial) replacement variables (e.g. factors or indices). An example for such a data situation are measured purchases from a scanner-based consumer tracking panel. Purchases differ in number and description among households, and every single purchase can be described by a large number of variables (point of sale, date, category, brand, prize,...). Instead of imputing single purchases, we create variables consisting of aggregated purchase behavior per household, and define the 'household' as observational unit for imputation (thus also avoiding dependencies among the observational units). After identifying nearest neighbors for the households, the disaggregated purchase behavior can be matched via the data keys, and we are free to choose our analysis objective.

The drawback of the 'row-wise approach' is that finding a nearest neighbor over several Y's always means to compromise: The more numerous or heterogenous the Y variables, or the smaller the donor pool (i.e. the bigger the average distance of the matched pairs), the more 'watered-down' the quality of the nearest neighbor-match will be.

Both algorithms, *BBPMM.col* and *BBPMM.row*, will yield asymptotically identical results, if the missing-data pattern is monotonous and if each variable has a unique missing-data pattern.

7.3.2 Arguments

mis.data.pat An object created by *fusion.prep* that contains information on all identified missing-data patterns. [1]

block.imp A scalar or vector containing the number(s) of the block(s) considered for imputation. Per default only the last block is imputed.

M The number of multiple imputations. If M=1, no Bayesian Bootstrap step is carried out.

out.file A character string that specifies the path and file name for the imputed data sets. If M>1, the extension '_<m>' is added before the file extension to mark the number of the imputed data set. If *out.file*=NULL (default), no data set is stored.

mod.sav A character string that specifies the path and file name for the model parameters of all variables per blocks and imputed data set. If *mod.sav*=NULL (default), the parameters are not written to an external file.

verbose The algorithm prints information on weighting matrices and imputation numbers. Default=TRUE.

man.weights Optional argument containing manual weights for the PMM step. *man.weights* can either be a list containing a vector for each missingness pattern, or just a vector, if only one missingness pattern/block exists. In either case, the number of elements in the vector(s) must match the number of variables in the corresponding block. Note that the higher the weight the lower the importance of a good match for the corresponding variable's predictive means.

tol An imported argument from function *qr* that specifies the tolerance level for linear dependencies among the complete variables, and defaults to 0.25 within *BBPMM.row*.

... Further arguments passed to or from other functions.

7.3.3 Values

impdata The imputed data set, if M=1, or a list containing M imputed data sets.

BB.impdata A list containing the M bootstrapped data sets (for diagnostic purposes – only available if M>1).

weight.matrix A list containing weight matrices for all blocks and imputations.

model A list containing the imputation models for all blocks and imputations.

pairlist A list containing the donor/recipient pairlist data frames for all blocks and imputations.

dist A list containing the PMM distance vectors for all blocks and imputations.

7.3.4 Troubleshooting

Let $\mathbf{X^k_{obs}}$ be those values of the completely observed variables which are also observed for a set of incomplete variables $\mathbf{Y_k} = [y_{k,1} \ldots y_{k,K}]$ with identical missingness patterns. Furthermore let n^k_{obs} denote the number of observed values for any y_k. If $n^k_{obs} < K$, the algorithm returns a warning and proceeds with the imputation of the next block (missing-data pattern). Additionally, a QR decomposition for $\mathbf{X^k_{obs}}$ is carried out to check for serious multi-collinearity among the imputation model variables, and again, the algorithm skips the imputation of $\mathbf{Y_k}$, if the matrix does not have full rank at the specified tolerance level. Since *BBPMM.row* also uses a Bayesian Bootstrap step, missing factor levels of variables that are part of $\mathbf{X_{obs}}$ need to be dealt with analogously to *BBPMM.col*.

8

Concluding remarks

Multiple Imputation in conjunction with Predictive Mean Matching solves many of the problems that arise in imputation tasks with discrete survey data. Where parametric approaches have to specify zero-inflated Poisson models (because of the 'hyperbolic' character of some variables in the data set), lower and upper bounds (to avoid implausible values), or where the empirical distribution does not asymptotically resemble any statistical distribution, PMM will still give relatively consistent unbiased estimates as the results from the experimental designs in chapters 5 and 6 have demonstrated – even when the imputation model is (moderately) misspecified. Robustness to model misspecification is an important issue for statistical analysis of incomplete survey data, because the empirical probability density functions of many variables do not resemble theoretical statistical distributions, and thus inferences are prone to being biased, if the variable distributions are not examined carefully.

The gist of this summary is *not* to encourage 'imputers' to be sloppy with the specification of their imputation models – because PMM will 'right the wrongs' anyway. But for very large incomplete survey data sets (with several hundred or even thousand variables), the specification of tailor-made imputation models for every incomplete variable is not economic. The introduced MI algorithms can be applied to those mass imputation problems, while still yielding relatively consistent and efficient estimators based on multiply imputed data.

Overall, we think that the benefits – robustness to model misspecification and imputation of plausible values – of PMM easily outweigh the aforementioned drawbacks for imputing incomplete survey data. Especially the observed value / plausible value issue is hardly a handicap, since most survey data variables have observations on the complete domain of the variable[1], unless the survey data suffer from censoring problems as a particular form of missing-data problem.

So far, we have only introduced a PMM variant for (unordered) categorical variables. In a next step we will investigate the relative performance of it in comparison with other proposed methods for MI of categorical variables. Additionally, extension to ordered probit models can also be explored and be implemented into another experimental design. This will also investigate, how legitimate it is to treat ordered categorical variables as discrete metric-scale variables in combination with PMM. Remember: we assumed that the situation for ordered-categorical variables can be compared to the imputation of genuine metric-scale variables under a (slightly) misspecified imputation model.

One problem we will tackle in the future is the trade-off that arises in FCS approaches: When we impute variable Y_k, we would like the other variables to explain this variable as well as possible. When we impute variable Y_{k+1}, the former regressand Y_k and all but Y_{k+1} now form the set of regressors. But ideally, this new set of regressors should not be multicollinear! An interesting alternative to (forward) stepwise regression, could be least-angle regression (Efron et al. 2004).

Another aspect of potential improvement is the flexibility of the imputation model. We have stated that tailor-made imputation models for every partially incomplete variable are hardly feasible in large survey data sets, but there are non-parametric regression approaches available that can automatically adjust our models. Generalized additive models (GAMs) as described by Hastie & Tibshirani (1990) could be used for mass imputation models, where only the hyperparameter λ that governs the 'wriggliness of the tails' of the model has to be specified upfront. First tests have shown that careless specification of λ leads to very large between vari-

[1] Admittedly, for a few typical survey data variables like 'age' this is theoretically not true.

ances for MI data sets, but sensible application could coalesce imputation model robustness with PMM's robustifying properties.

Appendix A

Diagnostic plots for the alcoholism study

This appendix section displays all histograms and Q-Q plots of the variables used in the alcoholism study.

A.1 Transformed variables

A.1.1 Histograms

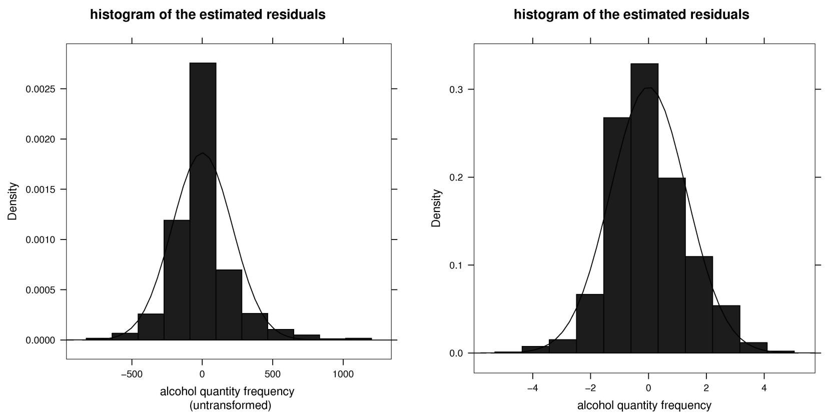

Figure A.1: Histogram: ALC_QF before and after transformation

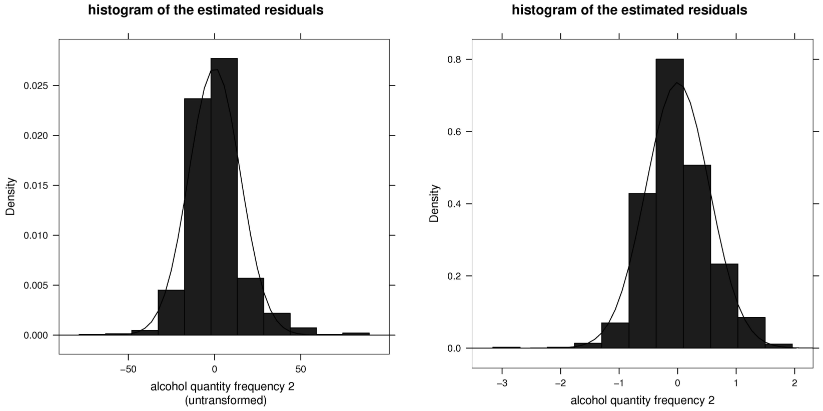

Figure A.2: Histogram: AQF2 before and after transformation

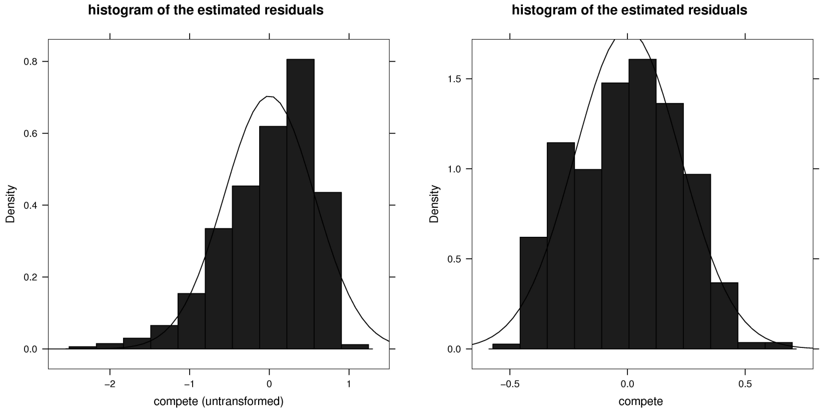

Figure A.3: Histogram: COMPETE before and after transformation

A.1.2 Q-Q plots

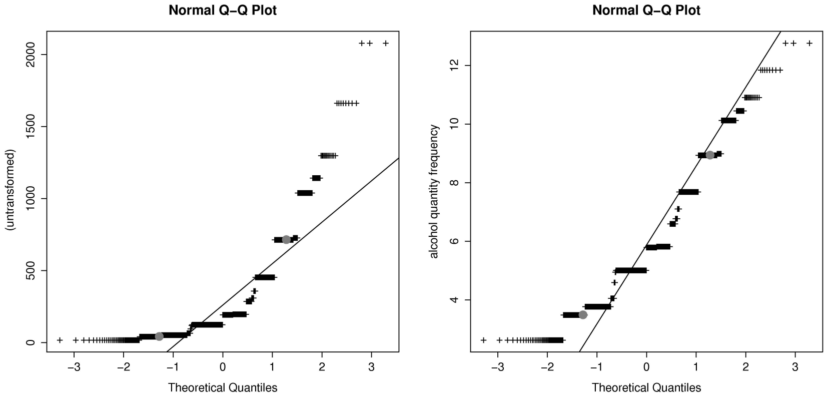

Figure A.4: Q-Q plot: ALC_QF before and after transformation

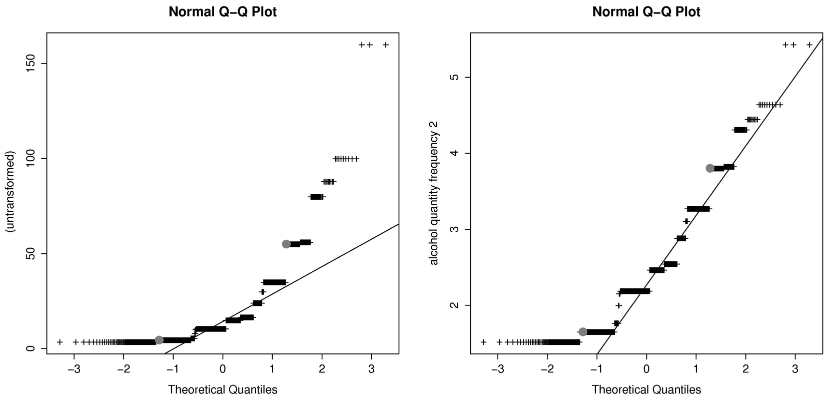

Figure A.5: Q-Q plot: AQF2 before and after transformation

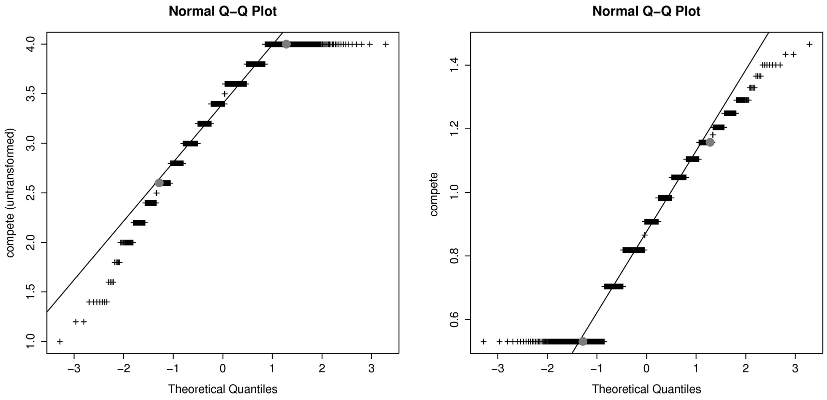

Figure A.6: Q-Q plot: COMPETE before and after transformation

A.2 Untransformed variables

A.2.1 Histograms

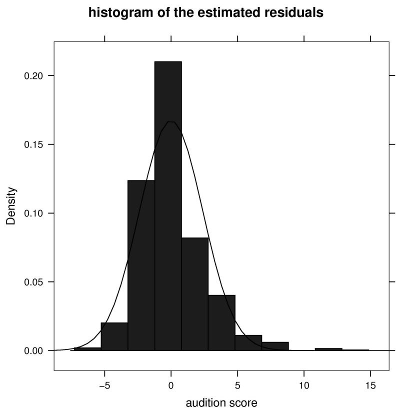

Figure A.7: Histogram: AUDIT

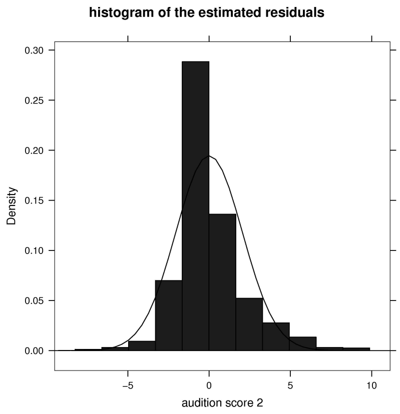

Figure A.8: Histogram: AUDIT2

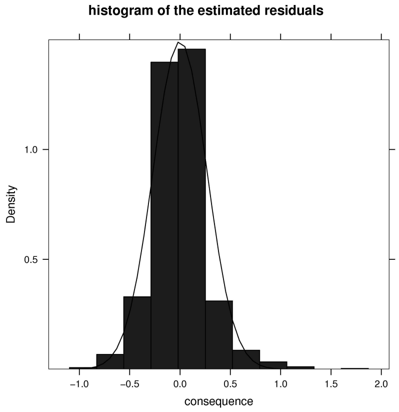

Figure A.9: Histogram: CONSEQNC

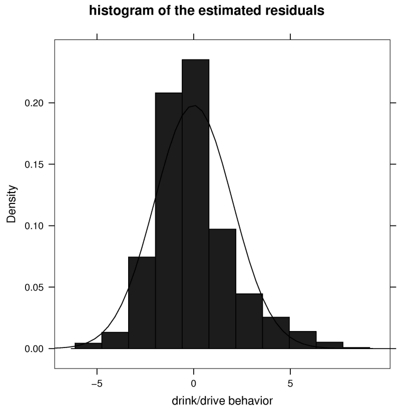

Figure A.10: Histogram: DRK_DRIV

A.2.2 Q-Q plots

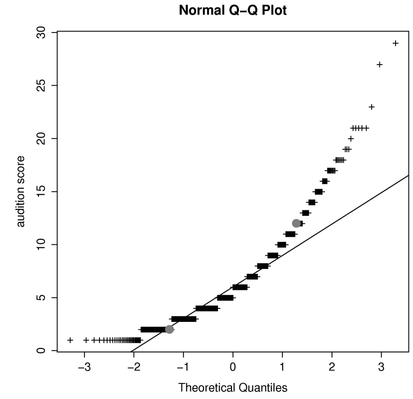

Figure A.11: Q-Q plot: AUDIT

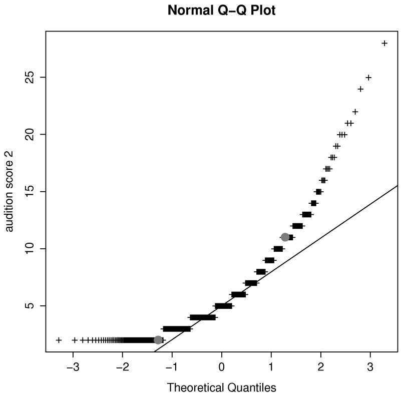

Figure A.12: Q-Q plot: AUDIT2

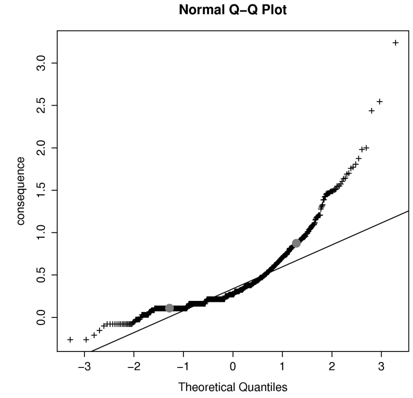

Figure A.13: Q-Q plot: CONSEQNC

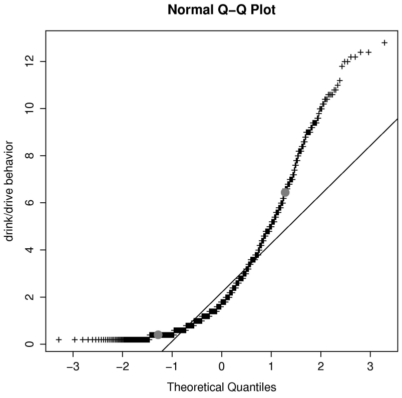

Figure A.14: Q-Q plot: DRK_DRIV

Appendix B

Tables

B.1 Basic tables from the discrete-data MCMC experimental design

In the following we list all basic tables for bias and coverage over all twelve data situations:

B.1.1 Bias tables: five methods and ten quantities of interest

Table B.1: Bias: DS1 MAR BIG

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	-0.35	2.8e-05	-0.00016	-0.00034	-0.011
$Var(Y)$	-0.12	**0.038**	-0.0085	-0.0087	-0.013
$p(Y < 3)$	0.059	0.0015	-0.00027	-0.00035	0.0012
$p(Y < 4)$	0.088	0.0011	6e-04	0.00065	0.0029
$p(Y < 6)$	0.058	-0.0016	-0.00029	-0.00025	0.0016
$\rho(X_1, Y)$	-0.011	-0.005	0.00097	0.00091	-0.012
$\rho(X_2, Y)$	0.00086	0.0076	0.00055	0.00062	0.016
α	-0.00012	0.0019	0.0014	0.0015	**0.032**
β_1	0.0011	0.00082	0.00057	0.00042	-0.017
β_2	0.00091	0.0024	0.0018	0.0019	0.012

Table B.2: Bias: DS1 MAR small

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	-0.33	0.011	0.012	0.012	0.0037
$Var(Y)$	-0.1	-0.018	**-0.058**	**-0.057**	**-0.064**
$p(Y < 3)$	0.054	-0.0013	-0.0038	-0.0036	-0.002
$p(Y < 4)$	0.084	-0.0012	-0.0019	-0.0017	0.00014
$p(Y < 6)$	0.053	-0.0054	-0.0041	-0.0044	-0.0027
$\rho(X_1, Y)$	-0.012	-0.0063	-0.0016	-0.0016	-0.013
$\rho(X_2, Y)$	0.0013	0.012	0.0035	0.0041	0.019
α	0.0057	0.027	0.029	0.03	0.057
β_1	0.002	-0.0055	-0.0086	-0.0085	-0.022
β_2	3.8e-05	0.011	0.009	0.0095	0.019

Table B.3: Bias: DS1 MCAR BIG

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	0.0012	0.00076	0.0012	0.00096	0.00082
$Var(Y)$	-0.001	**0.042**	-0.0068	-0.0074	-0.0061
$p(Y < 3)$	-0.00048	0.0015	-0.00011	-0.00016	-5.9e-05
$p(Y < 4)$	-0.00037	4e-04	1e-04	0.00015	7.5e-05
$p(Y < 6)$	-0.00014	-0.0018	-0.00055	-0.00051	-0.00039
$\rho(X_1, Y)$	-0.00066	-0.0056	4e-04	0.00034	-0.013
$\rho(X_2, Y)$	4.2e-05	0.0068	-0.00018	-0.00018	0.015
α	0.0019	0.0032	0.0038	0.004	**0.042**
β_1	-0.0011	-0.00091	-0.0013	-0.0016	-0.017
β_2	-0.00018	0.00099	0.00075	0.00079	0.01

Table B.4: Bias: DS1 MCAR small

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	-0.00059	0.0061	0.0074	0.0065	0.0059
$Var(Y)$	-0.012	-0.017	**-0.061**	**-0.06**	**-0.06**
$p(Y < 3)$	0.00046	0.00092	-6e-04	-0.00033	-0.00022
$p(Y < 4)$	0.0016	0.00036	-0.0017	-0.0011	-0.00058
$p(Y < 6)$	0.00068	-0.0022	0.00021	0.00037	0.00033
$\rho(X_1, Y)$	-0.00032	-0.0061	0.00024	0.00032	-0.011
$\rho(X_2, Y)$	0.0022	0.011	0.0037	0.0046	0.019
α	-0.0022	0.021	0.02	0.019	**0.052**
β_1	0.00099	-0.0053	-0.0047	-0.0042	-0.017
β_2	0.0015	0.011	0.0099	0.011	0.02

Table B.5: Bias: DS2 MAR BIG

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	-0.36	-0.0018	-0.0039	-0.0051	-0.014
$Var(Y)$	-0.14	0.01	-0.008	-0.011	-0.025
$p(Y < 3)$	0.066	0.0049	0.00074	0.00098	0.002
$p(Y < 4)$	0.079	-0.0074	0.0011	0.0012	0.0028
$p(Y < 6)$	0.044	**-0.025**	0.00023	0.00048	0.0018
$\rho(X_1, Y)$	-0.011	-0.00047	0.0024	0.0021	-0.0058
$\rho(X_2, Y)$	0.0029	0.0012	-0.0031	-0.0031	0.0068
α	-0.0076	0.00037	-0.0074	-0.0065	0.017
β_1	0.00095	-0.00048	0.00072	-0.00059	-0.014
β_2	-0.00072	0.0015	-0.0011	-0.0011	0.0076

Table B.6: Bias: DS2 MAR small

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	-0.34	0.015	0.0028	0.00034	-0.0033
$Var(Y)$	-0.15	**-0.073**	**-0.13**	**-0.14**	**-0.12**
$p(Y < 3)$	0.063	0.004	-0.00013	2e-05	0.00037
$p(Y < 4)$	0.074	-0.0094	-0.0027	-0.0022	-0.0014
$p(Y < 6)$	0.042	**-0.027**	0.00024	8.7e-05	0.00067
$\rho(X_1, Y)$	-0.0071	-0.00087	0.0012	-0.0011	-0.0045
$\rho(X_2, Y)$	-0.0024	0.0011	-0.0046	0.00021	0.0068
α	0.0049	**0.043**	**0.045**	**0.055**	**0.056**
β_1	-0.003	-0.015	-0.026	-0.029	-0.027
β_2	-0.00028	0.01	0.0092	0.015	0.019

Table B.7: Bias: DS2 MCAR BIG

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	0.003	0.0072	0.0044	0.0045	0.0031
$Var(Y)$	0.012	0.025	0.012	0.012	0.00012
$p(Y < 3)$	-0.00063	-0.00091	-0.00058	-5e-04	-0.00056
$p(Y < 4)$	-0.00017	-0.013	-2e-04	-0.00026	-6.8e-05
$p(Y < 6)$	-7e-04	-0.023	-0.0011	-0.0011	-0.00063
$\rho(X_1, Y)$	-0.00061	-0.0028	-0.00035	-5e-04	-0.0082
$\rho(X_2, Y)$	0.00012	0.0033	0.00038	0.00054	0.01
α	0.0033	0.016	0.0073	0.0084	**0.038**
β_1	-0.00098	-0.004	-0.0017	-0.0022	-0.015
β_2	-0.00038	0.0021	0.00068	0.00086	0.0092

Table B.8: Bias: DS2 MCAR small

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	0.013	**0.036**	0.021	0.023	0.021
$Var(Y)$	0.098	**0.031**	-0.0034	0.0085	0.013
$p(Y < 3)$	-0.0014	-0.001	-0.003	-0.0029	-0.0026
$p(Y < 4)$	6e-04	-0.014	-0.00069	-0.00035	-0.0015
$p(Y < 6)$	-0.00081	**-0.027**	-0.0016	-0.0016	-0.0019
$\rho(X_1, Y)$	0.0026	-0.0021	0.004	0.00056	-0.003
$\rho(X_2, Y)$	-0.00076	0.0068	0.00057	0.0048	0.011
α	-0.005	**0.049**	**0.025**	**0.038**	**0.044**
β_1	0.01	-0.004	0.00044	-0.0034	-0.0051
β_2	-0.0012	0.011	0.0095	0.013	0.016

Table B.9: Bias: DS3 MAR BIG

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	-0.47	0.0029	-0.0022	-0.0023	-0.022
$Var(Y)$	-0.17	**0.077**	-0.0087	-0.0091	**-0.04**
$p(Y < 3)$	0.055	0.025	-0.00011	-7.8e-05	0.0014
$p(Y < 4)$	0.079	**0.063**	-5e-04	-0.00049	0.0015
$p(Y < 6)$	0.076	**-0.076**	0.00038	0.00033	0.0028
$\rho(X_1, Y)$	-0.0069	-0.0056	-0.00088	-0.0011	-0.023
$\rho(X_2, Y)$	0.012	-0.00082	0.0091	0.0088	0.012
α	-0.0064	**0.46**	**-0.09**	**-0.084**	**0.1**
β_1	5e-04	**-0.16**	-0.0022	-0.0036	**-0.1**
β_2	0.0033	**-0.23**	**0.045**	**0.042**	**-0.064**

Table B.10: Bias: DS3 MAR small

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	-0.46	0.01	-0.00052	-0.0045	-0.0082
$Var(Y)$	-0.21	0.016	**-0.12**	**-0.12**	**-0.12**
$p(Y < 3)$	0.052	0.021	-0.002	-0.0017	-0.001
$p(Y < 4)$	0.078	**0.06**	-0.0019	-0.0015	-0.00051
$p(Y < 6)$	0.074	**-0.076**	-0.0031	-0.0023	-0.00053
$\rho(X_1, Y)$	-0.0079	-0.0096	-0.0039	-0.0062	-0.021
$\rho(X_2, Y)$	0.012	0.0039	0.0085	0.0078	0.012
α	-0.0083	**0.44**	-0.021	-2e-04	**0.1**
β_1	-0.0024	**-0.17**	**-0.041**	**-0.05**	**-0.11**
β_2	0.007	**-0.22**	0.0083	-0.0039	**-0.058**

Table B.11: Bias: DS3 MCAR BIG

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	-0.0032	-4e-04	0.00016	-0.00026	0.00077
$Var(Y)$	0.0082	**0.053**	-0.008	-0.0081	-0.0089
$p(Y < 3)$	0.001	**0.027**	0.00031	0.00028	0.00019
$p(Y < 4)$	0.00043	**0.069**	-0.00055	-0.00053	-0.00055
$p(Y < 6)$	2.8e-05	**-0.069**	-0.00026	-0.00019	-3e-04
$\rho(X_1, Y)$	0.00029	-0.0058	-0.00082	-0.0011	-0.022
$\rho(X_2, Y)$	0.00077	0.00087	0.011	0.011	0.013
α	-0.0041	**0.44**	**-0.099**	**-0.096**	**0.12**
β_1	-0.00017	**-0.16**	-0.0023	-0.0035	**-0.095**
β_2	0.0026	**-0.23**	**0.051**	**0.049**	**-0.058**

Table B.12: Bias: DS3 MCAR small

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	0.0044	0.0053	-0.00052	0.0018	-0.001
$Var(Y)$	-0.048	0.0027	**-0.12**	**-0.084**	**-0.083**
$p(Y < 3)$	-0.0014	0.024	-0.002	-0.00041	0.00075
$p(Y < 4)$	-0.00097	**0.067**	-0.0019	0.00077	0.00027
$p(Y < 6)$	-0.00011	**-0.068**	-0.0031	-0.00093	0.00041
$\rho(X_1, Y)$	-0.0013	-0.0092	-0.0039	-0.0053	-0.02
$\rho(X_2, Y)$	0.0049	0.0071	0.0085	0.012	0.016
α	-0.012	**0.42**	-0.021	**-0.035**	**0.078**
β_1	-0.00014	**-0.17**	**-0.041**	**-0.04**	**-0.096**
β_2	0.0033	**-0.21**	0.0083	0.016	**-0.042**

B.1.2 Coverage tables: five methods and ten quantities of interest

Table B.13: Coverage: DS1 MAR BIG

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	0	93.6	93.6	93.8	93.6
$Var(Y)$	90.8	95.0	93.2	94.2	94.2
$p(Y < 3)$	4.4	96.6	92.0	91.6	93.4
$p(Y < 4)$	0	97.2	92.0	92.2	91.2
$p(Y < 6)$	0.8	97.6	88.4	88.4	88.8
$\rho(X_1, Y)$	92.2	95.4	95.6	96.0	81.6
$\rho(X_2, Y)$	95.8	91.4	96.2	97.0	**51.0**
α	95.2	96.8	91.2	93.0	80.8
β_1	95.4	96.0	90.8	93.8	83.2
β_2	95.2	97.0	89.0	92.0	**72.4**

Table B.14: Coverage: DS1 MAR small

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	57.2	95.6	92.2	93.2	93.2
$Var(Y)$	96.8	97.0	92.4	92.8	94.2
$p(Y < 3)$	82.6	96.2	89.4	90.4	92.0
$p(Y < 4)$	67.0	97.4	91.2	92.0	92.8
$p(Y < 6)$	69.4	97.2	91.0	90.2	91.8
$\rho(X_1, Y)$	93.8	95.2	93.8	95.4	91.8
$\rho(X_2, Y)$	95.2	94.0	92.0	93.4	87.4
α	95.8	96.8	88.0	90.0	87.2
β_1	95.4	95.0	88.8	92.8	87.2
β_2	93.6	92.8	84.8	88.8	82.0

Table B.15: Coverage: DS1 MCAR BIG

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	94.6	94.8	93.6	93.8	93.2
$Var(Y)$	97.8	95.6	94.6	94.8	94.8
$p(Y < 3)$	96.2	97.8	91.8	92.8	92.0
$p(Y < 4)$	95.0	96.8	92.8	93.6	91.4
$p(Y < 6)$	95.2	96.2	92.6	91.4	91.4
$\rho(X_1, Y)$	96.6	94.2	95.6	97.0	**79.8**
$\rho(X_2, Y)$	95.4	91.6	96.4	97.6	**53.6**
α	96.0	96.6	89.8	92.0	**76.2**
β_1	96.2	96.8	91.8	94.0	84.0
β_2	95.6	95.6	89.8	94.4	**76.8**

Table B.16: Coverage: DS1 MCAR small

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	93.8	95.6	93.8	93.8	94.4
$Var(Y)$	95.8	93.4	91.8	91.8	90.6
$p(Y < 3)$	95.6	96.6	91.6	92.2	91.8
$p(Y < 4)$	94.4	96.4	90.8	91.8	91.0
$p(Y < 6)$	93.4	98.0	90.0	90.0	90.6
$\rho(X_1, Y)$	96.6	94.6	94.0	95.4	91.0
$\rho(X_2, Y)$	95.2	93.0	92.2	93.4	87.2
α	93.4	93.0	82.2	86.0	83.0
β_1	94.6	95.2	89.0	92.2	86.2
β_2	92.2	93.4	84.2	86.0	82.4

Table B.17: Coverage: DS2 MAR BIG

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	0.6	91.2	87.0	88.4	86.0
$Var(Y)$	78.6	**82.0**	**75.2**	**75.8**	**74.8**
$p(Y < 3)$	3.2	96.6	91.2	90.8	90.4
$p(Y < 4)$	0.8	95.8	94.0	94.6	91.0
$p(Y < 6)$	12.0	**55.0**	89.4	88.2	84.6
$\rho(X_1, Y)$	90.8	90.6	89.6	91.6	87.8
$\rho(X_2, Y)$	89.8	89.6	87.6	91.4	85.2
α	96.2	96.2	94.2	94.6	92.0
β_1	95.2	93.8	93.0	94.6	85.4
β_2	95.0	95.2	90.8	93.2	87.0

Table B.18: Coverage: DS2 MAR small

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	69.2	92.6	88.0	88.6	85.8
$Var(Y)$	82.2	**81.6**	**72.4**	**71.0**	**69.2**
$p(Y < 3)$	75.8	99.0	92.6	93.2	92.0
$p(Y < 4)$	74.8	98.8	90.0	90.8	90.0
$p(Y < 6)$	80.6	92.2	88.0	88.8	84.4
$\rho(X_1, Y)$	91.0	89.4	86.4	89.8	85.8
$\rho(X_2, Y)$	87.0	85.0	80.6	81.4	**79.0**
α	95.2	93.4	89.4	91.0	89.2
β_1	94.6	95.0	90.4	95.0	86.8
β_2	95.0	94.8	87.6	91.6	80.6

Table B.19: Coverage: DS2 MCAR BIG

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	93.8	93.6	91.6	90.4	88.8
$Var(Y)$	**82.4**	**81.4**	**82.0**	**81.4**	**78.8**
$p(Y < 3)$	95.2	98.2	90.8	90.4	91.2
$p(Y < 4)$	95.4	89.4	94.4	94.0	90.2
$p(Y < 6)$	96.0	**56.4**	90.4	90.6	88.0
$\rho(X_1, Y)$	92.4	89.0	89.0	92.2	86.6
$\rho(X_2, Y)$	86.6	87.2	87.4	91.8	80.6
α	95.0	94.4	91.2	92.0	88.4
β_1	96.2	94.4	93.0	97.2	88.6
β_2	95.6	95.6	92.8	94.8	86.6

Table B.20: Coverage: DS2 MCAR small

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	93.6	94.2	90.0	89.8	89.6
$Var(Y)$	**83.4**	**82.0**	**77.2**	**77.8**	**77.6**
$p(Y < 3)$	94.6	99.0	90.4	90.4	90.2
$p(Y < 4)$	95.4	97.4	90.6	91.2	90.4
$p(Y < 6)$	93.2	93.6	90.6	91.8	90.6
$\rho(X_1, Y)$	91.4	89.4	86.8	89.8	82.8
$\rho(X_2, Y)$	89.0	89.4	85.4	88.4	83.2
α	95.2	95.4	86.4	90.2	86.0
β_1	96.8	96.6	94.6	97.2	87.6
β_2	97.8	97.0	92.4	95.2	87.8

Table B.21: Coverage: DS3 MAR BIG

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	0	96.0	93.0	93.4	88.8
$Var(Y)$	80.8	88.4	87.4	87.4	84.8
$p(Y < 3)$	1.4	**30.0**	93.4	94.0	93.2
$p(Y < 4)$	0	**0**	94.2	94.8	93.6
$p(Y < 6)$	0	**0**	92.4	93.2	89.6
$\rho(X_1, Y)$	91.4	94.0	95.0	96.0	**7.6**
$\rho(X_2, Y)$	95.8	98.0	94.4	96.6	92.2
α	95.6	**44.4**	90.6	98.0	82.8
β_1	95.2	**0**	89.2	90.4	**3.2**
β_2	95.8	**42.6**	90.4	97.6	81.2

Table B.22: Coverage: DS3 MAR small

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	43.2	96.2	93.8	93.6	91.8
$Var(Y)$	86.8	89.2	83.2	84.8	86.4
$p(Y < 3)$	75.6	93.6	91.8	92.8	92.4
$p(Y < 4)$	66.8	**64.6**	92.0	93.8	91.6
$p(Y < 6)$	46.4	**55.6**	91.8	92.2	90.0
$\rho(X_1, Y)$	98.2	97.6	95.4	96.6	80.4
$\rho(X_2, Y)$	96.0	95.2	90.0	93.8	87.6
α	95.0	96.0	93.0	97.4	82.0
β_1	95.2	**56.8**	84.4	87.4	**60.2**
β_2	95.2	96.2	93.2	96.8	81.4

Table B.23: Coverage: DS3 MCAR BIG

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	92.2	92.2	93.4	93.4	92.8
$Var(Y)$	92.8	91.2	90.4	89.8	89.4
$p(Y < 3)$	95.0	**25.6**	95.2	95.8	94.2
$p(Y < 4)$	94.4	**0**	95.0	94.6	91.4
$p(Y < 6)$	95.0	**0**	96.0	95.0	93.4
$\rho(X_1, Y)$	97.6	93.4	97.4	97.0	**4.6**
$\rho(X_2, Y)$	97.8	96.8	93.2	95.0	89.4
α	96.2	**44.4**	91.2	97.0	80.4
β_1	95.2	**0**	90.6	93.2	**3.8**
β_2	95.2	**42.2**	90.8	96.6	**79.0**

Table B.24: Coverage: DS3 MCAR small

	CC	ROV	PPMM	BBPMM	RPMM
$E(Y)$	94.2	95.0	93.8	95.0	93.2
$Var(Y)$	89.2	89.6	83.2	89.4	87.4
$p(Y < 3)$	91.6	93.0	91.8	93.4	90.8
$p(Y < 4)$	92.4	**59.6**	92.0	93.2	92.4
$p(Y < 6)$	93.2	**57.2**	91.8	93.0	91.6
$\rho(X_1, Y)$	97.8	96.8	95.4	94.0	84.6
$\rho(X_2, Y)$	94.2	94.4	90.0	92.8	87.0
α	91.8	94.4	93.0	97.0	82.8
β_1	92.0	**53.8**	84.4	88.8	**66.2**
β_2	91.2	93.8	93.2	97.4	83.2

B.2 Tables from the analysis of the alcoholism study

B.2.1 Original data set: proportion and mean estimates

Table B.25: Proportions and means based on the different imputation methods

		estimate	lower bound	upper bound
$p(DR1 = 0)$	CC	0.4807	0.4602	0.5012
	BBPMMO	0.4755	0.4603	0.4908
	BBPMMT	0.4759	0.4608	0.4911
	IVE	0.4757	0.4605	0.4909
$p(DRK = 0)$	CC	0.5136	0.4931	0.5341
	BBPMMO	0.4926	0.4739	0.5113
	BBPMMT	0.4958	0.4759	0.5157
	IVE	0.501	0.4826	0.5195
$E(AQF)$	CC	3.236	3.107	3.364
	BBPMMO	3.388	3.349	3.427
	BBPMMT	3.387	3.348	3.426
	IVE	3.386	3.346	3.425
$E(AUD)$	CC	4.09	3.928	4.252
	BBPMMO	4.218	4.17	4.265
	BBPMMT	4.215	4.168	4.263
	IVE	4.216	4.169	4.263
$E(COM)$	CC	3.443	3.42	3.466
	BBPMMO	3.442	3.435	3.448
	BBPMMT	3.441	3.434	3.448
	IVE	3.44	3.433	3.447
$E(DR1 \mid DR1 > 0)$	CC	2.466	2.323	2.608
	BBPMMO	2.504	2.47	2.537
	BBPMMT	2.504	2.473	2.536
	IVE	2.512	2.48	2.543
$E(CON)$	CC	0.001749	-0.03083	0.03433
	BBPMMO	0.1245	0.1136	0.1354
	BBPMMT	0.1356	0.1226	0.1487
	IVE	0.05464	0.02818	0.08109
$E(AU2)$	CC	3.723	3.575	3.872
	BBPMMO	3.954	3.852	4.056
	BBPMMT	4.034	3.953	4.114
	IVE	4.173	4.095	4.251
$E(DRK \mid DRK > 0)$	CC	2.008	1.884	2.133
	BBPMMO	1.971	1.868	2.073
	BBPMMT	2.011	1.937	2.085
	IVE	2.375	2.3	2.449
$E(AQ2)$	CC	2.917	2.803	3.031
	BBPMMO	3.142	3.063	3.22
	BBPMMT	3.178	3.114	3.241
	IVE	3.195	3.087	3.303

B.2.2 Original data set: variance estimates

Table B.26: Variances based on the different imputation methods

		estimate	variance	lower bound	upper bound
$Var(AQF)$	CC	9.76	0.002319	8.881	10.73
	BBPMMO	10.63	0.001193	9.939	11.38
	BBPMMT	10.62	0.001187	9.929	11.36
	IVE	10.63	0.001188	9.939	11.38
$Var(AUD)$	CC	15.52	0.002331	14.12	17.06
	BBPMMO	16.35	0.00131	15.23	17.55
	BBPMMT	16.32	0.001311	15.2	17.52
	IVE	16.31	0.001309	15.19	17.51
$Var(COM)$	CC	0.3176	0.00141	0.2951	0.3419
	BBPMMO	0.3201	0.0007787	0.3031	0.3381
	BBPMMT	0.3204	0.0007788	0.3033	0.3384
	IVE	0.3196	0.0007737	0.3026	0.3375
$Var(DR1)$	CC	4.749	0.003388	4.237	5.322
	BBPMMO	4.783	0.00186	4.395	5.205
	BBPMMT	4.783	0.001831	4.398	5.201
	IVE	4.764	0.00179	4.385	5.176
$Var(CON)$	CC	0.6293	0.001486	0.5835	0.6787
	BBPMMO	0.4261	0.001721	0.3928	0.4623
	BBPMMT	0.4269	0.001817	0.3926	0.4642
	IVE	0.6046	0.000945	0.5691	0.6423
$Var(AU2)$	CC	13.11	0.002986	11.78	14.59
	BBPMMO	12.04	0.003375	10.73	13.52
	BBPMMT	12.55	0.002528	11.37	13.86
	IVE	12.25	0.001388	11.38	13.18
$Var(DR1)$	CC	3.186	0.004633	2.788	3.641
	BBPMMO	3.17	0.004444	2.777	3.618
	BBPMMT	3.286	0.00355	2.922	3.696
	IVE	3.498	0.002211	3.188	3.838
$Var(AQ2)$	CC	7.767	0.002854	6.995	8.624
	BBPMMO	7.222	0.003134	6.461	8.073
	BBPMMT	7.271	0.0023	6.614	7.994
	IVE	10.76	0.002878	9.668	11.98

B.2.3 Original data set: model parameter estimates

Table B.27: Regression parameters based on the different imputation methods

		estimate	lower bound	upper bound
α_1	CC	-1.107	-1.632	-0.5822
	BBPMMO	-1.191	-1.683	-0.6989
	BBPMMT	-1.365	-1.865	-0.8648
	IVE	-1.079	-1.586	-0.5716
α_2	CC	-0.4021	-0.9253	0.1211
	BBPMMO	-0.5083	-0.9977	-0.019
	BBPMMT	-0.7079	-1.203	-0.2134
	IVE	-0.6128	-1.12	-0.1056
α_3	CC	0.1861	-0.3373	0.7095
	BBPMMO	0.05902	-0.4248	0.5429
	BBPMMT	-0.1592	-0.661	0.3426
	IVE	-0.2878	-0.7986	0.223
α_4	CC	1.238	0.7093	1.767
	BBPMMO	1.066	0.5817	1.549
	BBPMMT	0.8211	0.3258	1.316
	IVE	0.5578	0.02944	1.086
β_1	CC	-0.4626	-0.6101	-0.3151
	BBPMMO	-0.4744	-0.617	-0.3318
	BBPMMT	-0.516	-0.6554	-0.3767
	IVE	-0.4114	-0.5567	-0.2661
β_2	CC	2.973	2.655	3.292
	BBPMMO	2.515	2.182	2.848
	BBPMMT	2.233	1.938	2.528
	IVE	2.509	2.111	2.908
γ	CC	1.704	1.625	1.783
	BBPMMO	1.753	1.68	1.827
	BBPMMT	1.72	1.64	1.801
	IVE	1.772	1.673	1.871
δ_1	CC	-0.2	-0.2995	-0.1005
	BBPMMO	-0.1968	-0.2963	-0.09734
	BBPMMT	-0.1399	-0.2148	-0.065
	IVE	-0.2645	-0.3331	-0.1959
δ_2	CC	-0.1008	-0.1825	-0.01924
	BBPMMO	-0.1	-0.1782	-0.02189
	BBPMMT	-0.1395	-0.2056	-0.07339
	IVE	-0.105	-0.1458	-0.0643

B.2.4 Original data set: correlation estimates

Table B.28: Correlation estimates based on the complete cases

	estimate	variance	lower bound	upper bound
$\rho(AQF, AUD)$	0.86	0.00044	0.85	0.87
$\rho(AQF, COM)$	-0.2	0.00044	-0.24	-0.16
$\rho(AUD, COM)$	-0.23	0.00044	-0.27	-0.19
$\rho(AQF, DR1)$	0.63	0.00044	0.6	0.65
$\rho(AUD, DR1)$	0.65	0.00044	0.63	0.67
$\rho(COM, DR1)$	-0.2	0.00044	-0.24	-0.16
$\rho(AQF, CON)$	0.42	0.00044	0.38	0.45
$\rho(AUD, CON)$	0.41	0.00044	0.38	0.45
$\rho(COM, CON)$	-0.11	0.00044	-0.15	-0.072
$\rho(DR1, CON)$	0.29	0.00044	0.26	0.33
$\rho(AQF, AU2)$	0.56	0.00044	0.54	0.59
$\rho(AUD, AU2)$	0.61	0.00044	0.58	0.63
$\rho(COM, AU2)$	-0.13	0.00044	-0.17	-0.094
$\rho(DR1, AU2)$	0.46	0.00044	0.43	0.49
$\rho(CON, AU2)$	0.64	0.00044	0.62	0.67
$\rho(AQF, DR1)$	0.44	0.00044	0.41	0.47
$\rho(AUD, DR1)$	0.43	0.00044	0.39	0.46
$\rho(COM, DR1)$	-0.15	0.00044	-0.19	-0.11
$\rho(DR1, DR1)$	0.49	0.00044	0.46	0.52
$\rho(CON, DR1)$	0.4	0.00044	0.36	0.43
$\rho(AU2, DR1)$	0.62	0.00044	0.59	0.64
$\rho(AQF, AQ2)$	0.6	0.00044	0.57	0.62
$\rho(AUD, AQ2)$	0.57	0.00044	0.54	0.6
$\rho(COM, AQ2)$	-0.13	0.00044	-0.17	-0.094
$\rho(DR1, AQ2)$	0.44	0.00044	0.4	0.47
$\rho(CON, AQ2)$	0.6	0.00044	0.57	0.62
$\rho(AU2, AQ2)$	0.84	0.00044	0.83	0.85
$\rho(DR1, AQ2)$	0.61	0.00044	0.58	0.63

Table B.29: Correlation estimates based on BBPMMO

	estimate	variance	lower bound	upper bound
$\rho(AQF, AUD)$	0.86	0.00024	0.85	0.87
$\rho(AQF, COM)$	-0.19	0.00024	-0.22	-0.16
$\rho(AUD, COM)$	-0.22	0.00024	-0.25	-0.19
$\rho(AQF, DR1)$	0.61	0.00025	0.59	0.63
$\rho(AUD, DR1)$	0.64	0.00025	0.62	0.66
$\rho(COM, DR1)$	-0.21	0.00024	-0.24	-0.18
$\rho(AQF, CON)$	0.36	0.00032	0.32	0.39
$\rho(AUD, CON)$	0.35	3e-04	0.32	0.38
$\rho(COM, CON)$	-0.094	0.00034	-0.13	-0.058
$\rho(DR1, CON)$	0.25	0.00032	0.21	0.28
$\rho(AQF, AU2)$	0.53	0.00062	0.5	0.57
$\rho(AUD, AU2)$	0.58	0.00069	0.54	0.61
$\rho(COM, AU2)$	-0.12	0.00036	-0.16	-0.084
$\rho(DR1, AU2)$	0.44	0.00054	0.4	0.47
$\rho(CON, AU2)$	0.58	0.00032	0.56	0.6
$\rho(AQF, DR1)$	0.41	0.00057	0.37	0.45
$\rho(AUD, DR1)$	0.4	0.00057	0.36	0.44
$\rho(COM, DR1)$	-0.15	0.00039	-0.19	-0.11
$\rho(DR1, DR1)$	0.47	6e-04	0.44	0.51
$\rho(CON, DR1)$	0.35	0.00037	0.32	0.39
$\rho(AU2, DR1)$	0.6	0.00066	0.57	0.64
$\rho(AQF, AQ2)$	0.54	0.00067	0.5	0.58
$\rho(AUD, AQ2)$	0.51	0.00063	0.47	0.54
$\rho(COM, AQ2)$	-0.11	0.00034	-0.15	-0.077
$\rho(DR1, AQ2)$	0.39	0.00047	0.35	0.43
$\rho(CON, AQ2)$	0.55	0.00033	0.53	0.58
$\rho(AU2, AQ2)$	0.83	0.00051	0.81	0.84
$\rho(DR1, AQ2)$	0.59	0.00077	0.55	0.63

Table B.30: Correlation estimates based on BBPMMT

	estimate	variance	lower bound	upper bound
$\rho(AQF, AUD)$	0.86	0.00024	0.85	0.87
$\rho(AQF, COM)$	-0.19	0.00024	-0.22	-0.16
$\rho(AUD, COM)$	-0.22	0.00024	-0.25	-0.19
$\rho(AQF, DR1)$	0.61	0.00025	0.59	0.62
$\rho(AUD, DR1)$	0.64	0.00026	0.62	0.66
$\rho(COM, DR1)$	-0.21	0.00024	-0.24	-0.18
$\rho(AQF, CON)$	0.35	0.00032	0.32	0.38
$\rho(AUD, CON)$	0.35	0.00036	0.32	0.38
$\rho(COM, CON)$	-0.098	3e-04	-0.13	-0.064
$\rho(DR1, CON)$	0.26	0.00035	0.22	0.29
$\rho(AQF, AU2)$	0.54	0.00071	0.5	0.58
$\rho(AUD, AU2)$	0.61	0.00075	0.57	0.64
$\rho(COM, AU2)$	-0.13	0.00038	-0.17	-0.096
$\rho(DR1, AU2)$	0.47	0.00094	0.42	0.51
$\rho(CON, AU2)$	0.55	4e-04	0.52	0.58
$\rho(AQF, DR1)$	0.4	0.00059	0.36	0.44
$\rho(AUD, DR1)$	0.41	0.00058	0.37	0.45
$\rho(COM, DR1)$	-0.16	0.00033	-0.2	-0.13
$\rho(DR1, DR1)$	0.49	9e-04	0.45	0.54
$\rho(CON, DR1)$	0.33	3e-04	0.3	0.36
$\rho(AU2, DR1)$	0.61	0.00062	0.57	0.64
$\rho(AQF, AQ2)$	0.52	0.00044	0.49	0.55
$\rho(AUD, AQ2)$	0.51	0.00041	0.48	0.54
$\rho(COM, AQ2)$	-0.13	0.00045	-0.17	-0.084
$\rho(DR1, AQ2)$	0.4	0.00069	0.36	0.45
$\rho(CON, AQ2)$	0.57	0.00032	0.54	0.59
$\rho(AU2, AQ2)$	0.75	0.00087	0.73	0.78
$\rho(DR1, AQ2)$	0.52	0.00054	0.48	0.55

Table B.31: Correlation estimates based on IVEware

	estimate	variance	lower bound	upper bound
$\rho(AQF, AUD)$	0.86	0.00024	0.85	0.87
$\rho(AQF, COM)$	-0.19	0.00024	-0.22	-0.16
$\rho(AUD, COM)$	-0.22	0.00024	-0.24	-0.19
$\rho(AQF, DR1)$	0.61	0.00024	0.59	0.63
$\rho(AUD, DR1)$	0.64	0.00025	0.62	0.66
$\rho(COM, DR1)$	-0.21	0.00024	-0.24	-0.18
$\rho(AQF, CON)$	0.42	0.00037	0.39	0.45
$\rho(AUD, CON)$	0.39	0.00044	0.35	0.42
$\rho(COM, CON)$	-0.1	0.00037	-0.14	-0.065
$\rho(DR1, CON)$	0.27	0.00041	0.23	0.31
$\rho(AQF, AU2)$	0.54	0.00049	0.51	0.57
$\rho(AUD, AU2)$	0.58	6e-04	0.54	0.61
$\rho(COM, AU2)$	-0.12	0.00034	-0.16	-0.089
$\rho(DR1, AU2)$	0.43	0.00053	0.39	0.47
$\rho(CON, AU2)$	0.65	0.00047	0.62	0.67
$\rho(AQF, DR1)$	0.43	4e-04	0.4	0.46
$\rho(AUD, DR1)$	0.4	0.00046	0.36	0.43
$\rho(COM, DR1)$	-0.14	0.00046	-0.19	-0.1
$\rho(DR1, DR1)$	0.45	0.00087	0.4	0.49
$\rho(CON, DR1)$	0.51	0.00049	0.48	0.55
$\rho(AU2, DR1)$	0.66	0.00044	0.63	0.68
$\rho(AQF, AQ2)$	0.53	0.00054	0.49	0.56
$\rho(AUD, AQ2)$	0.48	0.00063	0.44	0.52
$\rho(COM, AQ2)$	-0.12	0.00044	-0.17	-0.083
$\rho(DR1, AQ2)$	0.37	0.00071	0.32	0.41
$\rho(CON, AQ2)$	0.69	3e-04	0.67	0.7
$\rho(AU2, AQ2)$	0.78	0.00064	0.76	0.8
$\rho(DR1, AQ2)$	0.67	0.0012	0.63	0.71

B.2.5 Jackknife simulations: proportion and mean estimates

Table B.32: CC: Average mean and proportion estimates from 500 jackknife samples

	$E(Y)$	$Bias$	MSE	lower bound	upper bound	coverage	λ
AQF	3	-0.24	0.14	2.6	3.4	0.79	NA
AUDIT	3.8	-0.27	0.21	3.3	4.3	0.81	NA
COMPETE	3.5	0.015	0.0032	3.4	3.5	0.93	NA
DR1|DRI1 > 0	2.4	-0.075	0.12	1.9	2.9	0.91	NA
DR1|DRI1 $= 0$	0.5	0.021	0.0029	0.43	0.57	0.92	NA
CONSEQNC	-0.045	-0.046	0.0083	-0.16	0.068	0.91	NA
AUDIT2	3.5	-0.27	0.18	3	3.9	0.82	NA
DRIV|DRIV > 0	1.9	-0.11	0.098	1.5	2.3	0.91	NA
DRIV|DRIV $= 0$	0.54	0.025	0.0031	0.47	0.61	0.92	NA
AQF_2	2.7	-0.2	0.11	2.3	3.1	0.77	NA

Table B.33: BBPMMO: Average mean and proportion estimates from 500 jackknife samples

	$E(Y)$	$Bias$	MSE	lower bound	upper bound	coverage	λ
AQF	3.2	-0.041	0.083	2.8	3.6	0.95	0.12
AUDIT	4	-0.051	0.13	3.5	4.5	0.95	0.12
COMPETE	3.4	0.0034	0.0032	3.4	3.5	0.95	0.25
DR1|DRI1 > 0	2.5	0.0049	0.12	2	3	0.95	0.22
DR1|DRI1 $= 0$	0.49	0.0078	0.0023	0.42	0.55	0.96	0.2
CONSEQNC	-0.0035	-0.0053	0.0033	-0.065	0.058	0.79	0.27
AUDIT2	3.7	-0.039	0.11	3.2	4.2	0.95	0.13
DRIV|DRIV > 0	2	-0.013	0.092	1.6	2.4	0.94	0.24
DRIV|DRIV $= 0$	0.52	0.0051	0.0023	0.45	0.58	0.95	0.21
AQF_2	2.9	-0.034	0.066	2.5	3.3	0.96	0.13

Table B.34: BBPMMT: Average mean and proportion estimates from 500 jackknife samples

	$E(Y)$	$Bias$	MSE	lower bound	upper bound	coverage	λ
AQF	3.2	-0.053	0.086	2.8	3.6	0.95	0.16
AUDIT	4.1	-0.02	0.13	3.6	4.6	0.94	0.14
COMPETE	3.4	0.0031	0.0032	3.4	3.5	0.95	0.25
DR1|DRI1 > 0	2.5	0.005	0.12	2	3	0.96	0.22
DR1|DRI1 $= 0$	0.49	0.0075	0.0023	0.42	0.55	0.95	0.2
CONSEQNC	0.0049	0.0032	0.0029	-0.051	0.061	0.76	0.14
AUDIT2	3.7	-0.023	0.11	3.2	4.2	0.96	0.15
DRIV|DRIV > 0	2	-0.008	0.093	1.6	2.4	0.93	0.25
DRIV|DRIV $= 0$	0.52	0.0047	0.0023	0.45	0.58	0.96	0.21
AQF_2	2.9	-0.032	0.067	2.5	3.3	0.96	0.15

Table B.35: IVEware: Average mean and proportion estimates from 500 jackknife samples

	$E(Y)$	$Bias$	MSE	lower bound	upper bound	coverage	λ
AQF	3.3	0.036	0.089	2.8	3.7	0.99	0.27
AUDIT	4.2	0.12	0.14	3.7	4.7	0.95	0.11
COMPETE	3.4	-0.046	0.0051	3.3	3.5	0.8	0.24
DR1$\mid$DRI1 > 0	2.6	0.16	0.14	2.2	3.1	0.9	0.2
DR1$\mid$DRI1 $= 0$	0.48	0.0022	0.0022	0.42	0.55	0.97	0.21
CONSEQNC	0.00058	-0.0012	0.0039	-0.077	0.078	0.88	0.23
AUDIT2	3.8	0.11	0.12	3.4	4.3	0.95	0.12
DRIV$\mid$DRIV > 0	2.1	0.14	0.11	1.7	2.5	0.91	0.23
DRIV$\mid$DRIV $= 0$	0.51	-0.001	0.0022	0.45	0.58	0.96	0.23
AQF_2	2.9	0.03	0.072	2.6	3.3	0.98	0.22

B.2.6 Jackknife simulations: model parameter estimates

Table B.36: CC: Average parameter estimates from 500 jackknife samples

	$E(Y)$	$Bias$	MSE	lower bound	upper bound	coverage	λ
α_1	-1.2	-0.073	2	-3.2	0.84	0.96	NA
α_2	-0.73	-0.32	2.1	-2.7	1.3	0.95	NA
α_3	-0.43	-0.62	2.4	-2.4	1.6	0.91	NA
α_4	-0.077	-1.3	3.8	-2.1	1.9	0.78	NA
β_1	-0.25	0.21	0.21	-0.82	0.33	0.91	NA
β_2	0.97	-2	4.2	0.45	1.5	0.012	NA
γ	1.5	-0.25	0.22	1	1.9	0.55	NA
δ_1	-0.12	0.082	0.27	-0.75	0.52	0.9	NA
δ_2	-0.14	-0.037	0.21	-0.69	0.41	0.93	NA

Table B.37: BBPMMO: Average parameter estimates from 500 jackknife samples

	$E(Y)$	$Bias$	MSE	lower bound	upper bound	coverage	λ
α_1	-1.3	-0.15	1.8	-3.3	0.79	0.98	0.45
α_2	-0.59	-0.19	1.8	-2.6	1.5	0.98	0.45
α_3	-0.041	-0.23	1.8	-2.1	2	0.97	0.45
α_4	0.93	-0.31	1.9	-1.1	3	0.97	0.45
β_1	-0.45	0.012	0.15	-1	0.13	0.97	0.44
β_2	2	-0.94	1.3	0.98	3.1	0.56	0.54
γ	1.7	0.019	0.064	1.4	2.1	0.93	0.5
δ_1	-0.16	0.042	0.086	-0.57	0.26	0.96	0.58
δ_2	-0.16	-0.056	0.063	-0.5	0.19	0.97	0.55

Table B.38: BBPMMT: Average parameter estimates from 500 jackknife samples

	$E(Y)$	$Bias$	MSE	lower bound	upper bound	coverage	λ
α_1	-1.3	-0.18	1.8	-3.3	0.75	0.98	0.44
α_2	-0.62	-0.22	1.8	-2.6	1.4	0.98	0.44
α_3	-0.079	-0.26	1.8	-2.1	1.9	0.97	0.44
α_4	0.88	-0.35	1.9	-1.1	2.9	0.97	0.44
β_1	-0.46	-0.0011	0.14	-1	0.11	0.98	0.44
β_2	2	-0.92	1.3	1.1	3	0.52	0.47
γ	1.7	-0.012	0.057	1.4	2	0.93	0.48
δ_1	-0.068	0.13	0.063	-0.41	0.27	0.93	0.51
δ_2	-0.22	-0.12	0.054	-0.51	0.066	0.95	0.5

Table B.39: IVEware: Average parameter estimates from 500 jackknife samples

	$E(Y)$	$Bias$	MSE	lower bound	upper bound	coverage	λ
α_1	-1.5	-0.35	2.2	-3.6	0.68	0.96	0.45
α_2	-0.93	-0.53	2.4	-3.1	1.2	0.95	0.45
α_3	-0.57	-0.75	2.6	-2.7	1.6	0.92	0.45
α_4	0.27	-0.97	3	-1.9	2.4	0.88	0.46
β_1	-0.5	-0.037	0.18	-1.1	0.12	0.97	0.46
β_2	1.9	-1	1.4	1.1	2.8	0.29	0.52
γ	1.8	0.059	0.064	1.4	2.1	0.95	0.49
δ_1	-0.15	0.053	0.034	-0.43	0.14	0.98	0.49
δ_2	-0.18	-0.082	0.033	-0.43	0.059	0.97	0.41

Appendix C

R code

C.1 BBPMM.row – Bayesian Bootstrap Predictive Mean Matching for missing-by-design patterns

```
BBPMM.row <- function(mis.data.pat,
                      block.imp = length(mis.data.pat$blocks),
                      M=10,
                      out.file = NULL,
                      mod.sav = NULL,
                      man.weights = NULL,
                      verbose = T,
                      tol=0.25,
                      ...)
{
  pairlist <- list()
  weight.matrix <- list()
  model <- list()
  dist <- list()
  y.hat <- list()
  impdata <- list()
  BB.impdata <- list()
  data.set <- mis.data.pat$data
  key <- mis.data.pat$key
  block <- mis.data.pat$blocks
  comp.names <- mis.data.pat$comp.names
  if (!is.null(key) && ncol(key) == 1) {
    Donid <- Recid <- names(key)[1]
  } else if (!is.null(key) && ncol(key) == 2) {
    Recid <- names(key)[1]
    Donid <- names(key)[2]
  } else if (is.null(key)) {
    pairlist <- NULL}
  org.names <- var.names <- names(data.set)
```

```
w.model <- NULL
if(!is.null(man.weights)) {
  if(is.vector(man.weights)) {
    if(length(block.imp) > 1) {
      stop(paste("Only one vector with manual weights, but more than one",
                 "block specified for imputation!\n"))}
    if (any(unlist(man.weights) < 0)) {
      stop(paste("man.weights contains negative value(s)!\n"))}
    man.weights2 <- man.weights
    man.weights <- list()
    man.weights[[block.imp]] <- man.weights2 }
  if(length(setdiff(block.imp, 1:length(block))) > 0) {
    stop(paste(as.character(block.imp),"is not a subset of the number of",
               "different missing-data patterns (blocks)!\n"))}
  for (i in 1:length(man.weights)) {
    man.weights[[i]][man.weights[[i]]==0] <- 1e-16
  }
}
n <- nrow(data.set)
l <- ncol(data.set)
### first loop for MI----------------------------------------------------
for (m in 1:M) {
  if(!is.null(key)) {
    pairlist[[m]] <- list()
  }
  weight.matrix[[m]] <- list()
  model[[m]] <- list()
  dist[[m]] <- list()
  y.hat[[m]] <- list()
  impdata[[m]] <- data.set
  for (j in 1:length(block)) { # second loop for different blocks
    model[[m]][[j]] <- NULL
    k.model <- NULL
    miss <- function(x) {any(is.na(x)) }
    mrow <- apply(as.matrix(data.set[ ,block[[j]]]), 1, miss)
    mis.pos <- (1:n) [mrow == TRUE]
    obs.pos <- (1:n) [mrow == FALSE]
    S.xy <- NULL
    ## Test for available degress of freedom in the model
    if((length(obs.pos)-1) <= length(comp.names)) {
      warning(paste("Ratio between completely observed values for block",
                    j, "and imputation model variables is too small!\n"))
      next
    }
    ## Test for multicollinearity
    ## QR decomposition
    mc.test <- qr(as.matrix(data.set[obs.pos, comp.names]))
    if (mc.test$rank != length(comp.names)) {
      warning(paste("Multicollinearity in imputation model for block",
                    j,"too strong!\n"))
      next
    }
```

```
if (length(comp.names > 1)) {
  xvars <- paste(comp.names,collapse= ’ + ’)
} else if (length(comp.names == 1)) {
  xvars <- as.character(comp.names) }
y.hat[[m]][[j]] <- matrix(nrow=n,ncol=length(block[[j]]))
co2 <- 0
dist[[m]][[j]] <- vector()
if(M > 1){
  BB.data <- data.set
  BB.ind <- BayesBoot(ind.obs = obs.pos)
  BB.data[obs.pos, ] <- data.set[BB.ind, ]
}
for (k in block[[j]]) {
  co2 <- co2+1
  if (co2 == 1) {
    s.model <- as.formula(paste(var.names[k],’ ˜ ’,xvars))
    if(M == 1) {
      regmod <- lm(s.model, data=data.set, na.action=na.exclude)
    } else if(M > 1) {
      BB.stab <- BB.mod.stab.glm(data=data.set,BB.data=BB.data,
                                 s.model=s.model)
      regmod <- BB.stab$model
      if (any(BB.stab$mislevpos == T) && co2 == 1) {
        warning(paste("Imputation ",m,": Bayesian Bootstrap dropped ",
                      "at least one category of a factor variable!\n",
                      sep=""))
      }
    }
    var.T <- var(data.set[ ,k], na.rm=T)
    var.U <- var(regmod$residuals)
  } else if (co2 > 1) {
    if (length(var.names[block[[j]]][1:(co2-1)]) > 1) {
      lside <- var.names[block[[j]]][1:(co2-1)]
      xvars.e <- paste(lside, collapse=’ + ’)
    } else if (length(var.names[block[[j]]][1:(co2-1)]) == 1) {
      xvars.e <- as.character(var.names[block[[j]]][1:(co2-1)]) }
    s.model.e <- as.formula(paste(var.names[k],’ ˜ ’,xvars.e))
    if(M == 1) {
      regmod.e <- lm(s.model.e, data=data.set, na.action=na.exclude)
    } else if(M > 1) {
      regmod.e <- lm(s.model.e, data=BB.data, na.action=na.exclude)
    }
    y.c <- rep(NA, n)
    y.c[obs.pos] <- regmod.e$residuals
    data.set.2 <- as.data.frame(cbind(y.c,data.set[ ,comp.names]))
    s.model <- as.formula(paste(names(data.set.2[1]),’ ˜ ’,xvars))
    if(M == 1) {
      regmod <- lm(s.model, data=data.set.2, na.action=na.exclude)
    } else if(M > 1) {
      BB.data.2 <- as.data.frame(cbind(y.c,BB.data[ ,comp.names]))
      BB.stab <- BB.mod.stab.glm(data=data.set.2,BB.data=BB.data.2,
                                 s.model=s.model)
      regmod <- BB.stab$model
    }
    var.T <- var(y.c, na.rm=T)
    var.U <- var(regmod$residuals)
  }
  if (M > 1 && any(BB.stab$mislevpos == T)) {
    BB.regmod <- lm(s.model, data=BB.data, na.action=na.exclude)
    c.namen <- names(BB.stab$c.model$coefficients)
    paranames <- list()
    paranames[[1]] <- rownames(coef(summary(BB.stab$c.model)))
    paranames[[2]] <- colnames(coef(summary(BB.stab$c.model)))
    para <- matrix(nrow=length(c.namen),ncol=4,dimnames=paranames)
    para[BB.stab$mislevpos==F,] <- signif(coef(summary(BB.regmod)),3)
    para[BB.stab$mislevpos==T,] <- c(0,NA,NA,NA)
  } else { para <- signif(coef(summary(regmod)),3) }
  k.model <- rbind(k.model,
                   c(var.names[k],colnames(para)),
                   cbind(rownames(para), para))
  y.hat[[m]][[j]][ ,co2] <- predict(regmod,newdata=data.set,
                                    na.action="na.fail")
  ## multicollinearity among ys
  if (is.na(var.T) || var.T < 1e-16) {
    S.xy[co2] <- 1e16
  } else {
    S.xy[co2] <- var.U }
} ## end of loop k (incomplete variables)
if (length(S.xy) > 1) {
  weight.matrix[[m]][[j]] <- diag(S.xy)
} else {
  weight.matrix[[m]][[j]] <- S.xy}
if (!is.null(man.weights) && length(man.weights[[j]]) > 0) {
  if (any(man.weights[[j]] <= 0)) {
    cat("At least one weight is non-positive!","\n")
    break }
  if (length(man.weights[[j]]) != length(S.xy)) {
    cat(paste("manual weight vector ",as.character(j),
              " does not match number of variables with missing data",
              " in block ", as.character(j),"\n", sep=""))
    break }
  weight.matrix[[m]][[j]] <- man.weights[[j]]^(-1)*
    weight.matrix[[m]][[j]] }
y.hat.obs <- y.hat[[m]][[j]][obs.pos, ]
if (verbose) {
  cat(paste("Imputation ",m,": weight matrix for block ",j,
           ":\n",sep = ""))
  print(weight.matrix[[m]][[j]])
}
if (!is.null(key)) {
  pairlist[[m]][[j]] <- matrix(nrow=length(mis.pos),ncol=2)}
model[[m]][[j]] <- k.model
if (!is.null(mod.sav)) {
  model.header <- c(paste("Parameter estimates for imputation ",m,
                              ", block ",j,":",sep=""),rep("",4))
      m.verbose <- rbind(model.header, k.model)
      w.model <- rbind(w.model, m.verbose)}
    co3 <- 0
    for (i in mis.pos) # third loop b) for the unobserved ys
      {
        co3 <- co3+1
        index <- obs.pos[apply(t(y.hat.obs),2,
                               FUN = function(x) {
                                 t(y.hat[[m]][[j]][i, ] - x) %*%
                                   weight.matrix[[m]][[j]] %*%
                                     (y.hat[[m]][[j]][i, ] - x)})
                         == min(apply(t(y.hat.obs),2,
                                FUN = function(x) {
                                  t(y.hat[[m]][[j]][i, ] - x) %*%
                                    weight.matrix[[m]][[j]] %*%
                                      (y.hat[[m]][[j]][i, ] - x)}))]
        if (length(index) > 1) {
          index <- sample(index, 1)
        } # random selection in case of several nearest neighbours
        dist[[m]][[j]][co3] <- t(matrix(y.hat[[m]][[j]][i, ] -
                                        y.hat.obs[index])) %*%
                                          weight.matrix[[m]][[j]] %*%
                                            (y.hat[[m]][[j]][i, ] -
                                             y.hat.obs[index])
        if (!is.null(key)) {
          pairlist[[m]][[j]][co3, ] <- c(key[i,Recid],key[index,Donid]) }
        data.set[i, block[[j]]] <- data.set[index, block[[j]]]
      } ## end of i loop (missing values)
  } ## end of j loop (blocks)
  if (!is.null(key)) data.set <- cbind(key, data.set)
  impdata[[m]] <- data.set
  if (M > 1) {
    BB.impdata[[m]] <- BB.data
  }
  if (!is.null(out.file))
    {
      if (M > 1) {
        dot.pos <- which(strsplit(out.file,"")[[1]]==".")
        out.file2 <-  paste(substr(out.file,1,dot.pos-1),"_",m,
                             substring(out.file,dot.pos),sep="")
      } else if (M == 1) {out.file2 <- out.file}
      write.table (data.set, file = out.file2, sep = "\t", row.names = F)
    }
  if (!is.null(mod.sav))
    {write.table (w.model, file = mod.sav, sep = "\t", row.names = F,
                  col.names=F, quote=F)}
} ## end of m loop (MI)
if (M > 1) {
  list(impdata = impdata, BB.impdata = BB.impdata,
       weight.matrix = weight.matrix, model = model,
       pairlist = pairlist, dist = dist)
  } else if (M == 1) {
    list(impdata = impdata, weight.matrix = weight.matrix, model = model,
         pairlist = pairlist, dist = dist)
  }
}
```

C.2 BBPMM.col – Bayesian Bootstrap Predictive Mean Matching based on Sequential Regression

```
BBPMM.col <- function(data,
                      M = 10,
                      n.iter = 10,
                      out.file = NULL,
                      ignore = NULL,
                      var.type = NULL,
                      eff.measure = TRUE,
                      maxit = 20,
                      verbose=TRUE,
                      ...)
  {
    impdata <- list()
    M.data <- list()
    data <- as.data.frame(data)
    orgnames <- varnames <- names(data)
    org.l <- ncol(data)
    if (!is.null(ignore)) {
      if (is.character(ignore)) {
        ig.pos <- is.element(varnames, ignore)
      } else {
        ig.pos <- is.element(1:ncol(data), ignore)
      }
      not.inc <- as.data.frame(data[ ,ignore])
      varnames <- varnames[-ignore]
      data <- data[ ,-ignore]}
    n <- nrow(data)
    l <- ncol(data)
    if (eff.measure) {
      e.meas <- matrix(nrow=M, ncol=l)
      colnames(e.meas) <- varnames
    } else { e.meas <- NULL}
    ## take over class from data.frame or administer classes
    if (!is.null(var.type)) {
      if (length(var.type) != l) {
        stop(paste("Error: Number of flagged variables in ’var.type’",
                   "does not match number of (remaining) variables in",
                   "data set!\n"))
      } else if (any(var.type != "C" & var.type != "M")) {
        stop(paste("Error: ’var.type’ contains wrong character(s)!\n"))
      }
      f.pos <- which(var.type == "C")
      if (length(f.pos) > 0) {
        data[,f.pos] <- lapply(data[,f.pos], as.factor)}
      m.pos <- which(var.type == "M")
      if (length(m.pos) > 0) {
        data[,m.pos] <- lapply(data[,m.pos], as.numeric)}
    }
```

```
## indicator matrix for missing values
R <- matrix(is.na(data), nrow=n)
mis.num <- apply(is.na(data), 2, sum)
mis.overview <- paste("number of missing values ", names(data),": ",
                      mis.num, sep="")
if (verbose) print(mis.overview)
## new variable order
n.order <- order(mis.num)
o.order <- order(n.order)
o.data <- data[ ,n.order]
varnames <- varnames[n.order]
mvar <- apply(o.data, 2, FUN = function(x) {any(is.na(x)) })
p.impvar <- (1:l)[mvar == T]
p.comp <- (1:l)[mvar == F]
i.mis <- list()
i.obs <- list()
co1 <- 0
for (j in p.impvar) {
  co1 <- co1+1
  i.mis[[co1]] <- (1:n)[is.na(o.data[ ,j]) == T]
  i.obs[[co1]] <- (1:n)[is.na(o.data[ ,j]) == F]
}
## starting solution
co2 <- 0
MI.data <- o.data
for (j in p.impvar) {
  ## stepwise imputation of y_t based on y_1 to y_t-1.
  co2 <- co2+1
  if (length(p.comp) == 0) {
    MI.data[i.mis[[co2]],j] <- sample(MI.data[i.obs[[co2]],j],
                                      length(i.mis[[co2]]),replace = T)
    p.comp <- j
  }
  xvars <- paste(c(varnames[p.comp],varnames[p.impvar[0:(co2-1)]]),
                 collapse=’ + ’)
  s.model <- as.formula(paste(varnames[j], ’ ˜ ’, xvars,
                              sep=""))
  y <- MI.data[ ,j]
  if (is.numeric(y)) {
    regmod <- lm(s.model, data=MI.data, na.action=na.exclude)
    y.pred <- predict(regmod, newdata=MI.data, na.action="na.fail")
    y.pred.mis <- y.pred[i.mis[[co2]]]
    y.pred.obs <- y.pred[i.obs[[co2]]]
    allDif <- outer(y.pred.mis, y.pred.obs, FUN="-")
    allDif[allDif==0] <- 1e-09
    nextlist <- y[i.obs[[co2]]][max.col(as.matrix(abs(allDif)ˆ(-1)),
                                        ties.method="random")]
  } else if (is.factor(y)) {
    if (length(table(y)) > 2) {
      options(warn=-1)
      regmod <- multinom(s.model,data=MI.data,trace=F,
                         na.action=na.exclude)
      options(warn=0)
      y.pred <- predict(regmod, newdata=MI.data, type = "probs")
      y.pred[y.pred > 0.999] <- 0.999
      y.pred[y.pred < 0.001] <- 0.001
      l.y.pred <- log(y.pred/(1-y.pred))
      y.pred.mis <- l.y.pred[i.mis[[co2]],]
      y.pred.obs <- l.y.pred[i.obs[[co2]],]
      ## calculate outer product for all obs/mis columns
      dist <- matrix(rep(0,nrow(y.pred.mis)*nrow(y.pred.obs)),
                     ncol=nrow(y.pred.obs))
      for (i in 1:ncol(y.pred)){
        dist <- dist + outer(y.pred.mis[,i],y.pred.obs[,i],FUN="-")^2
      }
      m.dist <- matrix(dist,ncol=nrow(y.pred.obs),byrow=F)
      m.dist[m.dist==0] <- 1e-09
      nextlist <- y[i.obs[[co2]]][max.col(as.matrix(m.dist^(-1)),
                                          ties.method="random")]
    } else if (length(table(y)) == 2) {
      regmod <- glm(s.model, data=MI.data,
                    family = binomial(link="logit"),
                    na.action=na.exclude)
      y.pred <- predict(regmod, newdata=MI.data, na.action="na.fail")
      y.pred.mis <- y.pred[i.mis[[co2]]]
      y.pred.obs <- y.pred[i.obs[[co2]]]
      allDif <- outer(y.pred.mis, y.pred.obs, FUN="-")
      allDif[allDif==0] <- 1e-09
      nextlist <- y[i.obs[[co2]]][max.col(as.matrix(abs(allDif)^(-1)),
                                          ties.method="random")]
    }
  }
  MI.data[i.mis[[co2]],j] <- nextlist
} ## end of j cycle
##+++++++++++++++++++++++++PMM++++++++++++++++++++++++++++++++++
## Sequential Regression with Predictive Mean Matching
for (m in 1:M) {
  co <- 0
  iterate <- T
  while (iterate) {
    ##---------------first loop for iterations----------------------
    co <- co + 1
    co2 <- 0
    if (verbose) {
      cat(paste("Imputation ", m," of ",M ,": iteration ", co,
                sep=""), "\n") }
    ##--------------- Bayesian Bootstrap -------------------------
    if (M > 1) {
      ind1 <- BayesBoot(ind.obs = 1:n)
      ## Bayesian Bootstrap: draw n times with replacement as basis for
      ## imputation model parameter estimates
      BB.data <- MI.data[ind1, ]
    }
    ##----second loop for every variable with missing values------
```

```
for (j in p.impvar) {
  co2 <- co2 + 1
  xvars <- paste(varnames[-j], collapse = ’ + ’)
  y <- MI.data[ ,j]
  s.model <- as.formula(paste(varnames[j],’˜’,xvars))
  if (is.numeric(y)) {
    if (M == 1) {
      regmod <- lm(s.model, data=MI.data, na.action=na.exclude)
    } else if (M > 1) {
      BB.stab <- BB.mod.stab.glm(data=MI.data, BB.data=BB.data,
                                 s.model=s.model)
      regmod <- BB.stab$model}
    y.pred <- predict(regmod, newdata=MI.data,
                      na.action="na.fail")
    y.pred.mis <- y.pred[i.mis[[co2]]]
    y.pred.obs <- y.pred[i.obs[[co2]]]
    ## find nearest observed neighbour for y.hat.mis
    allDif <- outer(y.pred.mis, y.pred.obs, FUN="-")
    allDif[allDif==0] <- 1e-09
    nextlist <- y[i.obs[[co2]]][max.col(as.matrix(abs(allDif)ˆ(-1)),
                                        ties.method="random")]
    ## distance metric
    if ((eff.measure == T) & (co == n.iter)) {
      n.mis <- length(i.mis[[co2]])
      ## actual squared distances mean
      D.mean <- mean(apply(allDif, 1, FUN = function(x) min(xˆ2)))
      ## artificial randomized squared distances mean and variance
      d <- sample(y.pred.obs,n.mis)-sample(y.pred.mis,n.mis)
      mu.h.d <- mean(d)
      sig2.h.d <- var(d)
      e.meas[m,j] <- (sig2.h.d/n.mis + mu.h.dˆ2 -
                      D.mean)/(sig2.h.d/n.mis + mu.h.dˆ2)
    }
  } else if (is.factor(y) & length(table(y)) > 2) {
    if (M == 1) {
      options(warn = -1)
      regmod <- multinom(s.model,data=MI.data,trace=F,
                         na.action=na.exclude)
      options(warn = 0)
    } else if ( M > 1) {
      BB.stab <- BB.mod.stab.mlog(data=MI.data,
                                  BB.data=BB.data,
                                  s.model=s.model)
      regmod <- BB.stab$model
    }
    y.pred <- predict(regmod, newdata=MI.data,
                      type="probs",na.action="na.fail")
    y.pred[y.pred > 0.999] <- 0.999
    y.pred[y.pred < 0.001] <- 0.001
    l.y.pred <- log(y.pred/(1-y.pred))
    y.pred.mis <- l.y.pred[i.mis[[co2]],]
    y.pred.obs <- l.y.pred[i.obs[[co2]],]
```

```
      ## calculate outer product for all obs/mis columns
      dist <- matrix(rep(0,nrow(y.pred.mis)*nrow(y.pred.obs)),
                     ncol=nrow(y.pred.obs))
      for (i in 1:ncol(y.pred)){
        dist <- dist + outer(y.pred.mis[,i],y.pred.obs[,i],FUN="-")^2
      }
      m.dist <- matrix(dist,ncol=nrow(y.pred.obs),byrow=F)
      m.dist[m.dist==0] <- 1e-09
      nextlist <- y[i.obs[[co2]]][max.col(as.matrix(m.dist^(-1)),
                                          ties.method="random")]
    } else if (is.factor(y) & length(table(y)) == 2) {
      if (M == 1) {
        regmod <- glm(s.model, data=MI.data,
                      family = binomial(link="logit"),
                      na.action=na.exclude)
      } else if (M > 1) {
        BB.stab <- BB.mod.stab.glm(data=MI.data, BB.data=BB.data,
                                   s.model=s.model, model="binomial")
        regmod <- BB.stab$model}
      y.pred <- predict(regmod, newdata=MI.data,
                        na.action="na.fail")
      y.pred.mis <- y.pred[i.mis[[co2]]]
      y.pred.obs <- y.pred[i.obs[[co2]]]
      ## find nearest observed neighbour for y.hat.mis
      allDif <- outer(y.pred.mis, y.pred.obs, FUN="-")
      allDif[allDif==0] <- 1e-09
      nextlist <- y[i.obs[[co2]]][max.col(as.matrix(abs(allDif)^(-1)),
                                          ties.method="random")]
    }
    MI.data[i.mis[[co2]],j] <- nextlist
  }
  if (co == n.iter) iterate <- F
  MI.data2 <- MI.data[ ,o.order]
  if (is.null(ignore)) {
    M.data[[m]] <- MI.data2
  } else {
      M.data[[m]] <- as.data.frame(matrix(nrow=n,ncol=org.l))
      M.data[[m]][ ,ig.pos==F] <- MI.data2
      M.data[[m]][ ,ig.pos==T] <- not.inc
      names(M.data[[m]]) <- orgnames
    }
  if (!is.null(out.file))
    {
      if (M > 1) {
        dot.pos <- which(strsplit(out.file,"")[[1]]==".")
        out.file2 <-  paste(substr(out.file,1,dot.pos-1),"_",m,
                            substring(out.file,dot.pos),sep="")
      } else if (M == 1) {out.file2 <- out.file}
      write.table(M.data[[m]], file = out.file2, sep = "\t",
                  row.names = F)
    }
}
```

```
      if (M == 1) M.data <- M.data[[m]]
    }
    list(impdata=M.data, mis.overview=mis.overview, eff.measure=e.meas,
         ind.matrix=R)
  }
```

C.3 Additional functions

C.3.1 Bayesian Bootstrap

```
BayesBoot <- function(ind.obs,...)
  {
    n.obs <- length(ind.obs)
    draw <- runif(n.obs-1,0,1)
    ## n.obs-1 random draws from a [0,1]uniform distribution
    diff.a <- diff.b <- c()
    diff.a[1:(n.obs-1)] <- sort(draw)
    diff.a[n.obs] <- 1
    diff.b[1] <- 0
    diff.b[2:n.obs] <- diff.a[1:(n.obs-1)]
    ## this creates two lists: list A has 1 as n.obs-th observation and
    ## list B has 0 as first observation. The differences give a list of
    ## n.obs probabilities which sum up to 1.
    p.draw <- diff.a - diff.b
    d.w.repl.obs <- rmultinom(1, size = n.obs, prob = p.draw)
    BB.ind.obs <- rep(ind.obs, d.w.repl.obs)
    return(BB.ind.obs)
  }
```

C.3.2 Model stabilizer for bootstrapped data

```
## Bootstrap model stabilizer for linear models
BB.mod.stab.glm <- function(data, BB.data, s.model, model="linear")
  {
    if (model == "binomial") {
      regmod <- glm(s.model, data=BB.data, family = binomial(link="logit"),
                    na.action=na.exclude)
      c.regmod <- glm(s.model, data=data, family = binomial(link="logit"),
                      na.action=na.exclude)
    } else if (model == "linear") {
      regmod <- lm(s.model, data=BB.data, na.action=na.exclude)
      c.regmod <- lm(s.model, data=data, na.action=na.exclude)
    }
    c.namen <- names(c.regmod$coefficients)
    BB.namen <- names(regmod$coefficients)
    mislevpos <- !is.element(c.namen, BB.namen)
    if (any(mislevpos == T)) {
      help.coeff <- regmod$coefficients
      regmod$coefficients <- c.regmod$coefficients
      regmod$coefficients[mislevpos==T] <- 0
      regmod$coefficients[mislevpos==F] <- help.coeff
      regmod$xlevels <- c.regmod$xlevels
      regmod$rank <- c.regmod$rank
      regmod$assign <- c.regmod$assign
      regmod$qr$pivot <- c.regmod$qr$pivot
      regmod$qr$rank <- c.regmod$qr$rank
    }
    list(model=regmod, c.model=c.regmod, mislevpos=mislevpos)
  }
#######################################################################
## Bootstrap model stabilizer for multinomial logit models
BB.mod.stab.glm <- function(data, BB.data, s.model, model="linear")
  {
    if (model == "binomial") {
      regmod <- glm(s.model, data=BB.data, family = binomial(link="logit"),
                    na.action=na.exclude)
      c.regmod <- glm(s.model, data=data, family = binomial(link="logit"),
                      na.action=na.exclude)
    } else if (model == "linear") {
      regmod <- lm(s.model, data=BB.data, na.action=na.exclude)
      c.regmod <- lm(s.model, data=data, na.action=na.exclude)
    }
    c.namen <- names(c.regmod$coefficients)
    BB.namen <- names(regmod$coefficients)
    mislevpos <- !is.element(c.namen, BB.namen)
    if (any(mislevpos == T)) {
      help.coeff <- regmod$coefficients
      regmod$coefficients <- c.regmod$coefficients
      regmod$coefficients[mislevpos==T] <- 0
      regmod$coefficients[mislevpos==F] <- help.coeff
      regmod$xlevels <- c.regmod$xlevels
      regmod$rank <- c.regmod$rank
      regmod$assign <- c.regmod$assign
      regmod$qr$pivot <- c.regmod$qr$pivot
      regmod$qr$rank <- c.regmod$qr$rank
    }
    list(model=regmod, c.model=c.regmod, mislevpos=mislevpos)
  }
```

References

Abayomi, K., Gelman, A. & Levy, M. (2008), 'Diagnostics for multivariate imputations', *Journal of the Royal Statistical Society Series C* **57**, 273–291.

Aerts, M., Claeskens, G., Hens, N. & Molenberghs, G. (2002), 'Local multiple imputation', *Biometrika* **89**, 375–388.

Allison, P. (2001), *Missing Data*, Vol. P.D. Allison of *Quantitative Applications in the Social Sciences*, 136 edn, Sage University Papers, Thousand Oaks.

Barnard, J. & Rubin, D. (1999), 'Small-sample degrees of freedom with multiple imputation', *Biometrika* **86**, 948–955.

Baum, L. & Petrie, T. (1966), 'Statistical inference for probabilistic functions of finite state markov chains', *Annals of Mathematical Statistics* **37**, 1554–1563.

Beissel-Durrant, G. & Skinner, C. (2004), 'Estimation of the distribution of hourly pay from household survey data: The use of missing data methods to handle measurement error', *S3RI Methodology Working Papers, M04/08* .

Bernaards, C., Farmer, M., Qi, K., Dulai, G., Ganz, P. & Kahn, K. (2003), 'Comparison of two multiple imputation procedures in a cancer screening survey', *Journal of Data Science* **1**, 293–312.

Bingham, C., Elliott, M. & Shope, J. (2007), 'Social and behavioral characteristics of young adult drink/drivers adjusted for level of alcohol use', *Alcoholism: Clinical and Experimental Research* **31**, 655–664.

Box, G. & Tiao, G. (1992), *Bayesian Inference in Statistical Analysis*, Wiley, New York.

Carpenter, J., Kenward, M. & Vansteelandt, S. (2006), 'A comparison of multiple imputation and doubly robust estimation for analyses with missing data', *Journal of the Royal Statistical Society, Series A* **169**, 571–584.

Chen, H. & Little, R. (1999), 'A test of missing completely at random for generalised estimating equations with missing data', *Biometrika* **86**, 1–13.

Cohen, M. (1997), The bayesian bootstrap and multiple imputation for unequal probability sample designs, *in* 'Proceedings of the Section on Survey Research Methods', 635–638.

Dempster, A., Laird, N. & Rubin, D. (1977), 'Maximum likelihood from incomplete data via the EM algorithm', *Journal of the Royal Statistical Society Series B* **39**, 1–38.

Dempster, A. & Rubin, D. (1983), 'Rounding in regression estimation: The appropriateness of sheppard's correction', **45**, 51–59.

Drechsler, J. & Rässler, S. (2008), *Recent Advances in Linear Models and Related Areas*, Physica-Verlag HD, chapter Does Convergence Really Matter?, 341–355.

Duda, R., Hart, P. & Stork, D. (2001), *Pattern Classification*, Wiley, New York.

Efron, B. (1979), 'Bootstrap methods: Another look at the jackknife', *Annals of Statistics* **7**, 1–26.

Efron, B. (1994), 'Missing data, imputation, and the bootstrap', **89**, 463–479.

Efron, B., Hastie, T., Johnstone, I. & Tibshirani, R. (2004), 'Least angle regression', *Annals of Statistics* **33**, 402–429.

Gelfand, A. & Smith, A. (1990), 'Sampling based approaches to calculating marginal densities', **85**, 398–409.

Gelman, A., Hill, J., M.Yajima, Su, J. & Pittau, M. (2008), *'mi' package*, http://www.cran.r-project.org/.

Gelman, A., King, G. & Liu, C. (1998), 'Not asked and not answered: Multiple imputation for multiple surveys', **93**, 846–857. with discussion.

Geman, S. & Geman, D. (1984), 'Stochastic relaxation, gibbs distributions and the bayesian restoration of images', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**, 721–741.

Geweke, J. (1992), *Bayesian Statistics*, Oxford University Press, 169–193.

Goldberger, A. (1991), *A Course in Econometrics*, Harvard University Press, Cambridge, MA.

Gould, S. (1985), *The Flamingo's Smile. Reflections in Natural History*, W.W. Norton & Co, New York.

Harrell, F. (2006), *The Hmisc Package*, http://www.cran.r-project.org/.

Hartley, H. (1958), 'Maximum likelihood estimation from incomplete data', *Biometrics* **14**, 174–194.

Hastie, T. & Tibshirani, R. (1990), *Generalized additive models*, Chapman & Hall/CRC.

Hastings, W. (1970), 'Monte carlo sampling methods using markov chains and their applications', *Biometrika* **57**, 97–109.

Heeringa, S., Little, R. & Raghunathan, T. (2002), *Multivariate imputation of coarsened survey data on household wealth*, Wiley, New York, 357–372.

Heitjan, D. & Basu, S. (1996), 'Distinguishing "missing at random" and "missing completely at random"', *The American Statistician* **50**, 207–213.

Heitjan, D. & Landis, J. (1994), 'Assessing secular trends in blood pressure: A multiple imputation approach', **89**, 750–759.

Heitjan, D. & Rubin, D. (1990), 'Inference from coarse data via multiple imputation with application to age heaping', *Journal of the American Statistical Association* **85**, 304–314.

Heitjan, D. & Rubin, D. (1991), 'Ignorability and coarse data', *Annals of Statistics* **19**, 2244–2253.

Herring, A., Ibrahim, J. & Lipsitz, S. (2004), 'Non-ignorable missing covariate data in survival analysis: a case-study of an international breast cancer study group trial', *Journal of the Royal Statistical Society Series C* **53**, 293–310.

Horton, N., Lipsitz, S. & Parzen, M. (2003), 'A potential for bias when rounding in multiple imputation', *The American Statistician* **57**, 229–232.

Kamakura, W. & Wedel, M. (1997), 'Statistical data fusion for cross-tabulation', *Journal of Marketing Research* **34**, 485–498.

Kim, J. (2002), 'A note on approximate bayesian bootstrap imputation', *Biometrika* **89**, 470–477.

Landerman, L., Land, K. & Pieper, C. (1997), 'An empirical evaluation of the predictive mean matching method for imputing missing values', *Sociological Methods & Research* **26**, 3–33.

Li, K. (1988), 'Imputation using markov chains', *Journal of Statistical Computation and Simulation* **30**, 57–79.

Li, K., Raghunathan, T. & Rubin, D. (1991), 'Large-sample significance levels from multiply-imputed data using moment-based statistics and an F reference distribution', **86**, 1065–1073.

Little, R. (1986), 'Survey nonresponse adjustments for estimates of means', *International Statistical Review* **54**, 139–157.

Little, R. (1988*a*), 'Missing-data adjustments in large surveys', *Journal of Business and Economic Statistics* **6**, 287–296.

Little, R. (1988*b*), 'A test of missing completely at random for multivariate data with missing values', **83**, 1198–1202.

Little, R. (1992), 'Regression with missing x's: A review', **87**, 1227–1237.

Little, R. & An, H. (2004), 'Robust likelihood-based analysis of multivariate data with missing values', *Statistica Sinica* **14**, 949–968.

Little, R. & Rubin, D. (2002), *Statistical Analysis with Missing Data*, 2 edn, Wiley, New York.

Meng, X. (2000), 'Missing data: Dial M for ???', **95**, 1325–1330.

Meng, X. & Rubin, D. (1991), 'Using EM to obtain asymptotic variance-covariance matrices: The SEM algorithm', **86**, 899–909.

Meng, X. & Rubin, D. (1993), 'Maximum likelihood estimation via the ECM algorithm: A general framework', *Biometrika* **80**, 267–278.

Metropolis, N., Rosenbluth, A., Teller, A. & Teller, E. (1953), 'Equations of state calculations by fast computing machines', *The Journal of Chemical Physics* **21**, 1087–1092.

Münnich, R. & Rässler, S. (2005), 'PRIMA: A new multiple imputation procedure for binary variables', *Journal of Official Statistics* **21**, 325–341.

Neal, T., Raghunathan, T., Schenker, N., Katzoff, M. & Johnson, C. (2006), 'An evaluation of matrix sampling methods using data from the national health and nutrition examination survey', *Survey Methodology* **32**(2), 217–231.

Raghunathan, T. (1993), 'A quasi-empirical bayes method for small area estimation', **88**, 1444–1448.

Raghunathan, T. & Grizzle, J. (1995), 'A split questionnaire survey design', *JASA* **90**, 54–63.

Raghunathan, T., Lepkowski, J., Hoewyk, J. & Solenberger, P. (2001), 'A multivariate technique for multiply imputing missing values using a sequence of regression models', *Survey Methodology* **27**, 85–95.

Raghunathan, T., Solenberger, P. & Hoewyk, J. (2002), IVEware: Imputation and variance estimation software, User guide, Institue for Social Research, University of Michigan.

Rao, J. (1996), 'On variance estimation with imputed survey data', **91**, 499–506.

Rässler, S. (2002), *Statistical Matching - Lecture Notes in Statistics*, Springer, New York.

Rassler, S., Mäenpää, C. & Koller, F. (2002), A split questionnaire survey design applied to german media and consumer surveys, *in* 'Proceedings of the International Conference on Improving Surveys, ICIS 2002'.

Robins, J. & Rotnitzky, A. (1995), 'Semiparametric effciency in multivariate regression models with missing data', **90**, 122–129.

Robins, J., Rotnitzky, A. & Zhao, L. (1995), 'Analysis of semiparametric regression models for repeated outcomes in the presence of missing data', **90**, 109–121.

Rodgers, W. (1984), 'An evaluation of statistical matching', **2**, 91–102.

Rosenbaum, P. & Rubin, D. (1983), 'The central role of the propensity score in observational studies for causal effects', *Biomektrika* **70**, 41–55.

Rubin, D. (1976), 'Inference and missing data', *Biometrika* **63**, 581–592.

Rubin, D. (1978), Multiple imputation in sample surveys – a phenomological bayesian approach to nonresponse, *in* 'Proceedings of the Survey Research Method Section of the American Statistical Association', 20–40.

Rubin, D. (1981), 'The bayesian bootstrap', *Annals of Statistics* **9**, 130–134.

Rubin, D. (1986), 'Statistical matching using file concatenation with adjusted weights and multiple imputation', *Journal of Economic and Business Statistics* **4**, 87–94.

Rubin, D. (1987*a*), *Multiple Imputation for Nonresponse in Surveys*, Wiley, New York.

Rubin, D. (1987*b*), 'A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The sir algorithm. discussion of tanner and wong (1987)', **82**, 543–546.

Rubin, D. (1994), 'Missing data, imputation, and the bootstrap: Comment', **89**, 475–478.

Rubin, D. (1996), 'Multiple imputation after 18+ years', **91**, 473–489.

Rubin, D. & Schenker, N. (1986), 'Multiple imputation for interval estimation from simple random samples with ignorable nonresponse', **81**, 366–374.

Saunders, J., Aasland, O., Babor, T., Fuente, J. & Grant, M. (1993), 'Development of the alcohol use disorders identification test (AUDIT): WHO collaborative project on early detection of persons with harmful alcohol consumption-ii', *Addiction* **88**, 791–804.

Schafer, J. (1997), *Analysis of Incomplete Multivariate Data*, Chapman and Hall, London.

Schafer, J. (1999), 'NORM multiple imputation under a normal model, version 2.03', MI software, http://www.stat.psu.edu/ jls/misoftwa.html.

Scharfstein, D., Robins, J. & Rotnitzky, A. (1994), 'Adjusting for nonignorable drop-out using semi-parametric nonresponse models (with comments)', **94**, 1096–1146.

Shao, J. & Sitter, R. (1996), 'Bootstrap for imputed survey data', **91**, 1278–1288.

Sheppard, W. (1898), 'On the calculation of the most probable values of frequency constants for data arranged according to equidistant divisions of a scale', *Proceedings of the London Mathematical Society* **29**, 353–380.

Tang, G., Little, R. & Raghunathan, T. (2003), 'Analysis of multivariate missing data with nonignorable nonresponse', *Biometrika* **90**, 747–764.

Tanner, M. & Wong, W. (1987), 'The calculation of posterior distributions by data augmentation (with discussion)', **82**, 528–550.

van Buuren, S., Brand, J., Groothuis-Oudshoorn, C. & Rubin, D. (2006), 'Fully conditional specification in multivariate imputation', *Journal of Statistical Computation and Simulation* **76**, 1049–1064.

van Buuren, S. & Oudshoorn, K. (1999), Flexible multivariate imputation by MICE, TNO report PG/VGZ/99.054, TNO Prevention and Health.