

Western  Graduate&PostdoctoralStudies

Western University  
Scholarship@Western

---

Electronic Thesis and Dissertation Repository

---

7-28-2011 12:00 AM

## Diagnostic Checking, Time Series and Regression

Esam Mahdi

*The University of Western Ontario*

Supervisor

A.I. McLeod

*The University of Western Ontario*

Graduate Program in Statistics and Actuarial Sciences

A thesis submitted in partial fulfillment of the requirements for the degree in Doctor of Philosophy

© Esam Mahdi 2011

Follow this and additional works at: <https://ir.lib.uwo.ca/etd>



Part of the [Applied Statistics Commons](#), [Longitudinal Data Analysis and Time Series Commons](#), and the [Multivariate Analysis Commons](#)

---

### Recommended Citation

Mahdi, Esam, "Diagnostic Checking, Time Series and Regression" (2011). *Electronic Thesis and Dissertation Repository*. 244.

<https://ir.lib.uwo.ca/etd/244>

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact [wlsadmin@uwo.ca](mailto:wlsadmin@uwo.ca).

# DIAGNOSTIC CHECKING, TIME SERIES AND REGRESSION

(Spine title: Diagnostic Checking, Time Series and Regression)

(Thesis format: Integrated-Article)

by

Esam Mahdi

Graduate Program  
in  
Statistics

A thesis submitted in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy

The School of Graduate and Postdoctoral Studies  
The University of Western Ontario  
London, Ontario, Canada

© Esam Mahdi 2011

THE UNIVERSITY OF WESTERN ONTARIO  
SCHOOL OF GRADUATE AND POSTDOCTORAL STUDIES

**CERTIFICATE OF EXAMINATION**

Supervisor

\_\_\_\_\_  
Dr. A.I. McLeod

Examiners

\_\_\_\_\_  
Dr. Rogemar Mamon

\_\_\_\_\_  
Dr. Serge B. Provest

\_\_\_\_\_  
Dr. John Knight

\_\_\_\_\_  
Dr. Yulia Gel

The thesis by

**Esam Mahdi**

entitled:

**DIAGNOSTIC CHECKING, TIME SERIES AND REGRESSION**

is accepted in partial fulfillment of the  
requirements for the degree of  
Doctor of Philosophy

Date : **July 28, 2011**

**Andrew Watson**  
Chair of the Thesis Examination Board

## ABSTRACT

In this thesis, a new univariate-multivariate portmanteau test is derived. The proposed test statistic can be used for diagnostic checking ARMA, VAR, FGN, GARCH, and TAR time series models as well as for checking randomness of series and goodness-of-fit VAR models with stable Paretian errors. The asymptotic distribution of the test statistic is derived as well as a chi-square approximation. However, the Monte-Carlo test is recommended unless the series is very long. Extensive simulation experiments demonstrate the usefulness of this test and its improved power performance compared to widely used previous multivariate portmanteau diagnostic check.

The contributed R package **portes** is also introduced. This package can utilize multi-core CPUs often found in modern personal computers as well as a computer cluster or grid. The proposed package includes the most important univariate and multivariate diagnostic portmanteau tests with the new test statistic given in this thesis. It is also useful for simulating univariate/multivariate data from nonseasonal ARIMA /VARIMA process with finite or infinite variances, testing for stationarity and invertibility, and estimating parameters from stable distributions. Many illustrative applications are given.

In this thesis, it has been shown that the classical ordinary least squares regression may produce smaller p-values than it should due to the lack of statistical independency in the fitted model which may invalidate the statistical inferences. The Poincaré plots are suggested to check for such hidden positive correlations.

KEY WORDS: Diagnostic check, Portmanteau test, Monte-Carlo significance test, Residual autocorrelation function, VARMA models, FGN models, GARCH models, Stable distributions, Parallel computing and multi-core systems, Poincaré plots, R.

## ACKNOWLEDGEMENTS

I reserve my greatest appreciation for my thesis supervisor, Professor A.I. McLeod, for suggesting the research topics, his extremely helpful guidance, encouragement and patience throughout the course of my research work. I would like also to thank my thesis examiners, Professors Rogemar Mamon, Serge B. Provest, John Knight, and Yulia Gel for all helpful comments and suggestions. Finally, I would like to thank my family for their patience during my study period.

# CONTENTS

<b>ABSTRACT</b>	<b>iii</b>
<b>ACKNOWLEDGEMENTS</b>	<b>iv</b>
<b>LIST OF TABLES</b>	<b>vii</b>
<b>LIST OF FIGURES</b>	<b>ix</b>
<b>INTRODUCTION</b>	<b>1</b>
<b>1 IMPROVED MULTIVARIATE PORTMANTEAU TEST</b>	<b>5</b>
1.1 INTRODUCTION . . . . .	5
1.1.1 Multivariate portmanteau tests . . . . .	6
1.1.2 Univariate generalized variance portmanteau test . . . . .	8
1.2 NEW MULTIVARIATE PORTMANTEAU TEST . . . . .	10
1.2.1 Asymptotic distribution and approximation . . . . .	12
1.2.2 Monte-Carlo significance test . . . . .	16
1.3 SIMULATION RESULTS . . . . .	19
1.3.1 Comparison of type 1 error rates . . . . .	19
1.3.2 Power comparisons . . . . .	21
1.4 ILLUSTRATIVE APPLICATIONS . . . . .	26
1.4.1 IBM and S&P index . . . . .	26
1.4.2 Investment, income and consumption time series . . . . .	26
1.5 CONCLUDING REMARKS . . . . .	29

<b>2</b>	<b>PORTMANTEAU TESTS FOR TIME SERIES MODELS: R PACK- AGE portes</b>	<b>35</b>
2.1	INTRODUCTION . . . . .	35
2.2	THE MAIN FUNCTION: portest . . . . .	38
2.2.1	Monte-Carlo goodness-of-fit test . . . . .	39
2.2.2	Monte-Carlo testing for randomness . . . . .	43
2.2.3	Asymptotic distribution significance test . . . . .	45
2.3	COMPUTATION PORTMANTEAU TESTS . . . . .	46
2.3.1	Box and Pierce portmanteau tests . . . . .	46
2.3.2	Ljung and Box portmanteau tests . . . . .	47
2.3.3	Hosking portmanteau tests . . . . .	47
2.3.4	Li and McLeod portmanteau tests . . . . .	48
2.3.5	Generalized variance portmanteau test . . . . .	49
2.4	APPLICATIONS . . . . .	51
2.4.1	Canadian Lynx trappings data . . . . .	51
2.4.2	Monthly simple returns of the CRSP value-weighted index data	54
2.4.3	Monthly log stock returns of intel corporation data . . . . .	56
2.4.4	U.S. inflation data . . . . .	58
2.4.5	Nile annual minima data . . . . .	60
2.4.6	Canadian labor market data . . . . .	61
2.4.7	Prey data . . . . .	63
2.5	SOME USEFUL FUNCTIONS . . . . .	64
2.5.1	Stationary and invertibility of VARMA models . . . . .	64
2.5.2	Simulation from nonseasonal ARMA or VARMA models . . . . .	67
2.5.3	Fit parameters to stable distribution . . . . .	73
2.6	CONCLUDING REMARKS . . . . .	74
<b>3</b>	<b>POINCARÉ PLOTS AND REGRESSION WITH HIDDEN COR- RELATIONS</b>	<b>79</b>
3.1	INTRODUCTION . . . . .	79
3.2	ILLUSTRATIVE EXAMPLES . . . . .	81
3.2.1	Simulated data with hidden correlations . . . . .	81
3.2.2	Generalized linear modeling example . . . . .	83
3.2.3	Loess fitting example . . . . .	84
3.3	LINEAR REGRESSION MODEL . . . . .	88
3.4	SIMULATION RESULTS . . . . .	91
3.5	CONCLUDING REMARKS . . . . .	99
3.6	FUTURE WORK . . . . .	99
	<b>CURRICULUM VITAE</b>	<b>101</b>

## LIST OF TABLES

1.1	The empirical 1% and 5% significance levels, in percent, comparing approximation, $a\chi_b^2$ , and Monte-Carlo, MC, for the portmanteau test statistic $\mathfrak{D}_m$ . . . . .	20
1.2	Empirical power in percent comparison of $\mathfrak{D}_m$ and $\tilde{Q}_m$ for nominal 1% and 5% tests. $10^4$ simulations with $N = 10^3$ . . . . .	25
1.3	IBM and S&P 500 Index Data. $a\chi_b^2$ : approximation. MC: Monte-Carlo $N = 10^3$ . NA: not applicable. The p-values are in percent. The * indicates a p-value less than 0.1%. . . . .	27
1.4	Trivariate West German Macroeconomic Series. $a\chi_b^2$ : approximation. MC: Monte-Carlo using $10^3$ replications. The p-values are in percent and * indicates a p-value less than 0.1%. . . . .	27
1.5	The residuals of the fitted VAR(2) model on West German Macroeconomic series are tested for heteroscedastic effects. $a\chi_b^2$ : approximation. MC: Monte-Carlo using $10^3$ replications. The p-values are in percent and * indicates a p-value less than 0.1%. . . . .	28
2.1	CPU time in seconds. The R functions <code>FitAR()</code> , <code>ar()</code> , <code>arima()</code> , <code>arima0()</code> , <code>Arima()</code> , and <code>auto.arima()</code> are used to fit the AR(1) model to univariate time series of lengths $n = 100, 500, 1000$ and the <code>portest()</code> function is applied on the fitted model based on $10^3$ replications of Monte-Carlo test. <code>nslaves</code> denotes the number of CPU's are used. . . . .	43
2.2	CPU time in seconds. The R functions <code>VAR()</code> and <code>ar.ols()</code> are used to fit the VAR(1) model to series of lengths $n = 100, 500, 1000$ with dimensions $k = 2, 3, 4$ . The <code>portest()</code> function is applied on the fitted model based on $10^3$ replications of Monte-Carlo test. <code>nslaves</code> denotes the number of CPU's are used. . . . .	43
2.3	CPU time in seconds. Univariate time series of lengths $n = 100, 500, 1000$ are generated from normal process and R functions <code>garch()</code> and <code>garch-Fit()</code> are used to fit the GARCH(1,1) model, whereas the function <code>FitFGN()</code> is used to fit FGN model. The <code>portest()</code> function is applied on the fitted model based on $10^3$ replications of Monte-Carlo test. <code>nslaves</code> denotes the number of CPU's are used. . . . .	44
3.1	Classical ordinary least squares model (OLS). . . . .	81
3.2	Generalized least squares model (GLS) with exact covariance. . . . .	82
3.3	Distance vector generated from $t_5$ ; $k=1$ , range = 3, 6, 9, and $n=30$ . . . . .	93

3.4	Distance vector generated from $U(0, 1)$ ; $k=1$ , range = 3, 6, 9, and $n=60$ . . .	94
3.5	Distance vector generated from $N(0, 1)$ ; $k=1$ , range = 3, 6, 9, and $n = 90$ . . .	95
3.6	Distance vector generated from $N(0, 1)$ ; $k=2$ , range = 3, 6, 9, and $n=30$ . . .	96
3.7	Distance vector generated from $t_5$ ; $k=2$ , range = 3, 6, 9, and $n=60$ . . . .	97
3.8	Distance vector generated from $N(0, 1)$ ; $k=3$ , range = 3, 6, 9, and $n=30$ . . .	98

## LIST OF FIGURES

2.1	A simulated time series of example 2.5.2.2. . . . .	71
2.2	A simulated time series of example 2.5.2.3. . . . .	72
3.1	Poincaré plot of the residuals from OLS fitted model with a loess smooth.	82
3.2	Poincaré plot of the residuals from GLS fitted model with a loess smooth.	83
3.3	Poincaré plot of deviance residuals in the logistic regression of low birth weight on 9 explanatory variables and 8 parametric bootstrap simulations.	85
3.4	Residual dependency plot of residuals vs. age. . . . .	86
3.5	Residuals plot in the robust loess fit of Cleveland (1993, section 3.6) to the polarization data. . . . .	87
3.6	Poincaré plot of residuals in the robust loess fit to the polarization data. .	87
3.7	Poincaré plot of residuals in the robust loess fit to the polarization data with ties in both variables removed. . . . .	88

# INTRODUCTION

After identification and estimation of the parameters in a fitted model, the goodness of fit test is the next most important step for testing the selected model. In time series analysis, we assume that the series is stationary with white noise innovations. This implies that a good fitted model must produce residuals that are approximately uncorrelated in time. [Box and Pierce \(1970\)](#) show that the asymptotic distribution of the residual autocorrelations can be utilized to check the validity of this assumption under the ARMA models. They introduced to the literature the overall goodness-of-fit test on the residuals autocorrelations up to lag  $m$ . This test statistic is called *portmanteau test*. Since then there evolves many literature on portmanteau tests for ARMA and GARCH models ([Ljung and Box, 1978](#); [McLeod and Li, 1983](#); [Peña and Rodriguez, 2002](#); [Rodríguez and Ruiz, 2005](#); [Peña and Rodriguez, 2006](#)). The portmanteau test is extended to the multivariate VARMA models by [Chitturi \(1974, 1976\)](#); [Hosking \(1980\)](#); [Li and McLeod \(1981\)](#); [Francq and Raïsi \(2007\)](#) and to the MGARCH models by ([Li and Mak, 1994](#); [Ling and Li, 1997](#)). [Lin and McLeod \(2006\)](#) introduce the Monte-Carlo portmanteau test and show that this test provides a test with the correct size. They show that the Monte-Carlo version of [Peña and Rodriguez \(2002\)](#) is often more powerful than its competitors. [Lin and McLeod \(2008\)](#) extends the Monte-Carlo test of [Peña and Rodriguez \(2002\)](#) to the ARMA models with stable Paretian errors.

In Chapter 1 of this thesis we introduce a new univariate-multivariate portmanteau test based on the approximation chi-square distribution and the Monte-Carlo procedures. The new test statistic may be considered an extension of the univariate portmanteau test statistics suggested by [Peña and Rodriguez \(2002\)](#); [Lin and McLeod](#)

(2006, 2008). Simulation experiments demonstrate that the proposed test statistic is the most powerful test with the correct size.

In Chapter 2, the portmanteau tests for time series models, **portes**, R package is introduced. The Simple Network of Workstations, **snow**, R package for parallel computing is implemented in this package. Many applications with this package are given.

In Chapter 3, the Poincaré plots are suggested as a tool for checking the possibility of lack of statistical independence in the observations due to positive hidden correlations. It is found via simulations that the test statistics based on the classical ordinary least squares regression may overstate the significance level due to lack of statistical independence in the observations.

## REFERENCES

- Box, G. and Pierce, D. (1970). Distribution of residual autocorrelation in autoregressive-integrated moving average time series models. *Journal of American Statistical Association* 65(332), 1509–1526.
- Chitturi, R. V. (1974). Distribution of residual autocorrelations in multiple autoregressive schemes. *Journal of the American Statistical Association* 69(348), 928–934.
- Chitturi, R. V. (1976). Distribution of multivariate white noise autocorrelations. *Journal of the American Statistical Association* 71(353), 223–226.
- Francq, C. and Raïsi, H. (2007). Multivariate portmanteau test for autoregressive models with uncorrelated but nonindependent errors. *Journal of Time Series Analysis* 28(3), 454–470.
- Hosking, J. R. M. (1980). The multivariate portmanteau statistic. *Journal of American Statistical Association* 75(371), 602–607.
- Li, W. K. and Mak, T. K. (1994). On the squared residual autocorrelations in non linear time series with conditional heteroskedasticity. *The Journal of Time Series Analysis* 15, 627–636.
- Li, W. K. and McLeod, A. I. (1981). Distribution of the residual autocorrelation in multivariate arma time series models. *Journal of the Royal Statistical Society, Series B* 43(2), 231–239.
- Lin, J.-W. and McLeod, A. I. (2006). Improved Peña-Rodríguez portmanteau test. *Computational Statistics and Data Analysis* 51(3), 1731–1738.
- Lin, J.-W. and McLeod, A. I. (2008). Portmanteau tests for ARMA models with infinite variance. *Journal of Time Series Analysis* 29(3), 600–617.
- Ling, S. and Li, W. K. (1997). Diagnostic checking of nonlinear multivariate time series with multivariate ARCH errors. *Journal of Time Series Analysis* 18(5), 447–464.
- Ljung, G. M. and Box, G. E. P. (1978). On a measure of lack of fit in time series models. *Biometrika* 65, 297–303.
- McLeod, A. I. and Li, W. K. (1983). Diagnostic checking ARMA time series models using squared-residual autocorrelations. *The Journal of Time Series Analysis* 4, 269–273.

- Peña, D. and Rodríguez, J. (2002). A powerful portmanteau test of lack of test for time series. *Journal of American Statistical Association* 97(458), 601–610.
- Peña, D. and Rodríguez, J. (2006). The log of the determinant of the autocorrelation matrix for testing goodness of fit in time series. *Journal of Statistical Planning and Inference* 8(136), 2706–2718.
- Rodríguez, J. and Ruiz, E. (2005). A powerful test for conditional heteroscedasticity for financial time series with highly persistent volatilities. *Statistica Sinica* 15, 505–525.

## Chapter 1

### IMPROVED MULTIVARIATE PORTMANTEAU TEST

#### 1.1 INTRODUCTION

The VARMA  $(p, q)$  model for a  $k$ -dimensional mean zero time series  $\mathbf{Z}_t = (Z_{1,t}, \dots, Z_{k,t})'$  can be written as

$$\Phi(\mathbf{B})\mathbf{Z}_t = \Theta(\mathbf{B})\mathbf{a}_t, \quad (1.1)$$

where  $\Phi(\mathbf{B}) = \mathbb{I}_k - \Phi_1\mathbf{B} - \dots - \Phi_p\mathbf{B}^p$ ,  $\Theta(\mathbf{B}) = \mathbb{I}_k - \Theta_1\mathbf{B} - \dots - \Theta_q\mathbf{B}^q$ ,  $\mathbb{I}_k$  is the identity matrix of order  $k$ , the coefficient matrices are,  $\Phi_\ell = (\phi_{i,j,\ell})_{k \times k}$ ,  $\ell = 1, \dots, p$ ;  $\Theta_\ell = (\theta_{i,j,\ell})_{k \times k}$ ,  $\ell = 1, \dots, q$  and  $\mathbf{B}$  is the backshift operator on  $t$ . Let  $\boldsymbol{\beta} = (\text{vec } \Phi_1, \dots, \text{vec } \Phi_p, \text{vec } \Theta_1, \dots, \text{vec } \Theta_q)$  be the vector of true parameters, where  $\text{vec}$  denotes the matrix vectorization function. We assume that an efficient estimation algorithm such as maximum likelihood is used to produce the corresponding estimate  $\hat{\boldsymbol{\beta}}$  so that  $\hat{\boldsymbol{\beta}} - \boldsymbol{\beta} = O_p(n^{-1/2})$ . The white noise process,  $\mathbf{a}_t = (a_{1,t}, \dots, a_{k,t})'$ , is assumed independent normal with mean zero and covariance matrix,  $E(\mathbf{a}_t\mathbf{a}'_{t-\ell}) = \delta_\ell\boldsymbol{\Gamma}_0$ , where  $\boldsymbol{\Gamma}_0$  is the innovation covariance matrix and  $\delta_\ell = 1$  or  $0$  according as  $\ell = 0$  or  $\ell \neq 0$ . The assumption of normality may be relaxed to that of strong white noise so that  $\mathbf{a}_t, t = 1, \dots, n$  are assumed to be independent and identically distributed with mean zero and constant covariance matrix,  $\boldsymbol{\Gamma}_0$ . The model is assumed to be stationary, invertible, and identifiable (Box et al., 2008, §14.2). After fitting this model to a series of length  $n$ , the residuals,  $\hat{\mathbf{a}}_t = (\hat{a}_{1,t}, \dots, \hat{a}_{k,t})'$ ,  $t = 1, \dots, n$  may be estimated and used to check the model assumption that the innovations are white

noise, that is, to test the null hypothesis that

$$\mathcal{H}_0 : \mathbf{\Gamma}_\ell = 0, \ell = 1, \dots, m, \quad (1.2)$$

where  $\mathbf{\Gamma}_\ell = \text{Cov}\{\mathbf{a}_t, \mathbf{a}_{t-\ell}\}$  and  $m$  is chosen large enough to cover all lags,  $\ell$ , of interest. Several versions of the multivariate portmanteau test have been developed for this purpose (Li, 2004).

In the next two subsections, brief reviews are given of previous multivariate portmanteau tests as well as the univariate versions of the generalized variance test of Peña and Rodriguez (2002, 2006). In Section 1.2, the multivariate extension of the generalized variance test of Peña and Rodriguez (2002) is discussed and its asymptotic distribution is derived. As in the univariate case (Peña and Rodriguez, 2002, Equation 9), it is shown in Equation 1.18 that the stronger the multivariate autocorrelation, the smaller the generalized variance. A chi-square approximation is suggested but for most purposes it is recommended to use the Monte-Carlo testing procedure that is described in Section 1.2.2. Simulation experiments in Section 1.3, demonstrate the improvement in power over the widely used previous multivariate portmanteau test. Illustrative applications are discussed in Section 1.4.

### 1.1.1 Multivariate portmanteau tests

The portmanteau test statistics,  $Q_m$  and  $\tilde{Q}_m$  and others, discussed in this section are all asymptotically  $\chi^2_{k^2(m-p-q)}$  as  $n \rightarrow \infty$ . It is also assumed that  $m > p + q$  is fixed and that  $m$  large enough so that Theorem 5 in Li and McLeod (1981) holds.

Hosking (1980) defined the residual autocorrelation matrix,

$$\hat{\mathbf{R}}_\ell = \hat{\mathbf{L}}' \hat{\mathbf{\Gamma}}_\ell \hat{\mathbf{L}}, \quad (1.3)$$

where  $\hat{\Gamma}_\ell = n^{-1} \sum_{t=\ell+1}^n \hat{\mathbf{a}}_t \hat{\mathbf{a}}_t'$ ,  $\hat{\Gamma}_{-\ell} = \hat{\Gamma}_\ell'$ ,  $\ell \geq 0$  and  $\hat{\mathbf{L}}$  is the lower triangular Cholesky decomposition of  $\hat{\Gamma}_0^{-1}$ . The multivariate portmanteau test statistic may be written,

$$Q_m = n \sum_{\ell=1}^m \hat{\mathbf{r}}_\ell' (\hat{\mathbf{R}}_0^{-1} \otimes \hat{\mathbf{R}}_0^{-1}) \hat{\mathbf{r}}_\ell, \quad (1.4)$$

where  $\hat{\mathbf{r}}_\ell = \text{vec } \hat{\mathbf{R}}_\ell'$  is a row vector of length  $k^2$  formed by stacking the rows of  $\hat{\mathbf{R}}_\ell$ ,  $\otimes$  is the Kronecker product, and  $m$  represents the number of lags being tested. In the univariate case,  $Q_m$  is identical to Box-Pierce portmanteau statistic (Box and Pierce, 1970) and both statistics are asymptotically  $\chi_{k^2(m-p-q)}^2$  (Hosking, 1980, 1981a).

Li and McLeod (1981) defined,

$$\hat{\mathbf{R}}_\ell^{(\dagger)} = (\hat{r}_{i,j}(\ell))_{k \times k}, \quad (1.5)$$

where  $\hat{r}_{i,j}(\ell) = \hat{\gamma}_{i,j}(\ell) / \sqrt{(\hat{\gamma}_{i,i}(0) \hat{\gamma}_{j,j}(0))}$ ,  $i, j = 1, \dots, k$ ,  $\hat{\gamma}_{i,j}(\ell) = n^{-1} \sum_{t=\ell+1}^n \hat{a}_{i,t} \hat{a}_{j,t-\ell}$ ,  $\hat{\gamma}_{i,j}(-\ell) = \hat{\gamma}_{j,i}(\ell)$ ,  $\ell \geq 0$ . Replacing  $\hat{\mathbf{R}}$  by  $\hat{\mathbf{R}}^{(\dagger)}$  in Equation 1.4, another portmanteau test statistic  $Q_m^{(\dagger)}$  is obtained. The null distribution of  $Q_m^{(\dagger)}$  is also asymptotically  $\chi_{k^2(m-p-q)}^2$ . The definition of residual autocorrelations used in Equation 1.3 is equivalent to the residual autocorrelations in Equation 1.5 if the residuals used Equation 1.5,  $\hat{\mathbf{a}}_t$ , are replaced by the standardized residuals,  $\hat{\mathbf{L}}' \hat{\mathbf{a}}_t$ .

Chitturi (1974) defined the residual autocorrelation matrix at lag  $\ell$ ,

$$\hat{\mathbf{R}}_\ell^{(\ddagger)} = \hat{\Gamma}_\ell \hat{\Gamma}_0^{-1}, \quad (1.6)$$

and another portmanteau test statistic  $Q_m^{(\ddagger)}$  is obtained by replacing  $\hat{\mathbf{R}}$  by  $\hat{\mathbf{R}}^{(\ddagger)}$  in Equation 1.4, and its null distribution is also asymptotically  $\chi_{k^2(m-p-q)}^2$ .

Hosking (1981a) noted that  $Q_m = Q_m^{(\dagger)} = Q_m^{(\ddagger)}$  and the portmanteau test statistic may be expressed simply in terms of the residual autocovariances,

$$Q_m = n \sum_{\ell=1}^m \text{tr}(\hat{\Gamma}'_{\ell} \hat{\Gamma}_0^{-1} \hat{\Gamma}_{\ell} \hat{\Gamma}_0^{-1}), \quad (1.7)$$

where  $\text{tr}(\cdot)$  denotes trace of a matrix. The multivariate portmanteau test statistic is equivalent to a test based on the Lagrange multiplier (Hosking, 1981b; Poskitt and Tremayne, 1982).

Hosking (1980) and Li and McLeod (1981) suggested modified versions of  $Q_m$  so that the expected value of the modified portmanteau statistic under the null hypothesis is equal to  $k^2(m-p-q) + O_p(1/n)$  and showed that both of these modifications are satisfactory when  $n$  and  $m$  are large enough. Simulation experiments suggest that both these modified portmanteau tests work about equally well (Li, 2004, §3).

The modified portmanteau test of Hosking (1980) is given by,

$$\tilde{Q}_m = n^2 \sum_{\ell=1}^m \hat{\mathbf{r}}'_{\ell} (\hat{\mathbf{R}}_0^{-1} \otimes \hat{\mathbf{R}}_0^{-1}) \hat{\mathbf{r}}_{\ell} / (n - \ell). \quad (1.8)$$

In the univariate time series, the  $\tilde{Q}_m$  test statistic approximately equal the Ljung-Box statistic (Ljung and Box, 1978) and both statistics are asymptotically  $\chi_{k^2(m-p-q)}^2$  (Hosking, 1980, 1981a).

### 1.1.2 Univariate generalized variance portmanteau test

Peña and Rodriguez (2002) proposed a univariate portmanteau test statistic,

$$\hat{D}_m = n \left( 1 - |\hat{\mathcal{R}}_m|^{1/m} \right), \quad (1.9)$$

where  $|\cdot|$  denotes the determinant and  $\hat{\mathcal{R}}_m$  is the residual correlation matrix of order

$m + 1$ ,

$$\hat{\mathcal{R}}_m = \begin{pmatrix} 1 & \hat{r}_{11}(1) & \dots & \hat{r}_{11}(m) \\ \hat{r}_{11}(1) & 1 & \dots & \hat{r}_{11}(m-1) \\ \vdots & \dots & \ddots & \vdots \\ \hat{r}_{11}(m) & \hat{r}_{11}(m-1) & \dots & 1 \end{pmatrix}. \quad (1.10)$$

Peña and Rodriguez (2002) derived the asymptotic distribution of  $\hat{D}_m$  as gamma using the standardized values of residual autocorrelations. Li (2004, §2.7) noted several interesting interpretations for this statistic. It was shown in simulation experiments (Peña and Rodriguez, 2002) that the  $\hat{D}_m$  statistic had better power than the test of Ljung and Box (1978) in many situations. One problem noted by Lin and McLeod (2006) is that the test statistic  $\hat{D}_m$  may not exist because, with the modified version of the residual autocorrelations used, the residual autocorrelation sequence is not always positive-definite or even non-negative definite. Furthermore, the size of the test may not be accurate due to the asymptotic approximation (Li, 2004, p. 19). To overcome these difficulties Lin and McLeod (2006) suggested using a Monte-Carlo significance test and demonstrated that this approach provides a test with the correct size and is often more powerful than the usual Ljung-Box test (Lin and McLeod, 2006, Table 6).

Peña and Rodriguez (2006) suggested taking the log of the  $(m + 1)$ th root of the determinant in Equation 1.10,

$$\tilde{D}_m = -n(m + 1)^{-1} \log | \hat{\mathcal{R}}_m | \quad (1.11)$$

and they derived a gamma distribution approximation for this test statistic.

In the portmanteau tests based on the asymptotic distribution (Ljung and Box, 1978; Peña and Rodriguez, 2002, 2006) not only is the size of the test inaccurate if the series length  $n$  is not large enough but there is also a problem if  $m$ , the number of

lags, is not large enough as well. The Monte-Carlo significance test approach does not require any such assumption about  $m$  and has much better finite-sample properties than tests based on the asymptotic distribution.

## 1.2 NEW MULTIVARIATE PORTMANTEAU TEST

The univariate residual autocorrelations in the Toeplitz matrix in Equation 1.10 are replaced by,  $\hat{\mathbf{R}}_\ell$ ,  $\ell = 1, \dots, m$  in Equation 1.3,

$$\hat{\mathfrak{R}}_m = \begin{pmatrix} \mathbb{I}_k & \hat{\mathbf{R}}_1 & \dots & \hat{\mathbf{R}}_m \\ \hat{\mathbf{R}}_1' & \mathbb{I}_k & \dots & \hat{\mathbf{R}}_{m-1}' \\ \vdots & \dots & \ddots & \vdots \\ \hat{\mathbf{R}}_m' & \hat{\mathbf{R}}_{m-1}' & \dots & \mathbb{I}_k \end{pmatrix}, \quad (1.12)$$

where  $\mathbb{I}_k = \hat{\mathbf{R}}_0$ . The proposed multivariate portmanteau test statistic is

$$\mathfrak{D}_m = -n \log |\hat{\mathfrak{R}}_m|. \quad (1.13)$$

From Hadamard's inequality for the determinant of a positive definite matrix,  $|\hat{\mathfrak{R}}_m| \leq 1$ . When there is no significant autocorrelation in the residuals,  $\hat{\mathbf{R}}_\ell = O_p(n^{-\frac{1}{2}})$  so  $\hat{\mathfrak{R}}_m$  is approximately block diagonal and hence  $|\hat{\mathfrak{R}}_m| \approx 1$ .

On the other hand, when there is autocorrelation present,  $|\hat{\mathfrak{R}}_m|$  will be expected to be smaller than 1. To see this we repeatedly apply the formula for the determinant of a partitioned matrix (Seber, 2008, §14.1),

$$|\hat{\mathfrak{R}}_m| = \prod_{\ell=1}^m |\mathbb{I}_k - \hat{\mathfrak{R}}_{(\ell)} \hat{\mathfrak{R}}_{\ell-1}^{-1} \hat{\mathfrak{R}}_{(\ell)}'|, \quad (1.14)$$

where  $\hat{\mathfrak{R}}_{(\ell)} = [\hat{\mathbf{R}}_1 : \dots : \hat{\mathbf{R}}_\ell]$  is the  $k$ -by- $\ell k$  block partitioned matrix. Then  $\hat{\Sigma}_\ell = \mathbb{I}_k - \hat{\mathfrak{R}}_{(\ell)} \hat{\mathfrak{R}}_{\ell-1}^{-1} \hat{\mathfrak{R}}_{(\ell)}'$  corresponds to the error covariance matrix when a linear predictor of

order  $\ell$  is fit to  $\hat{\mathbf{L}}'\hat{\mathbf{a}}_t$  using the previous  $\ell$  values (Reinsel, 1997, Equation 3.15). Thus, Equation 1.14 is a direct multivariate generalization of the well known univariate decomposition of generalized variance into the product of the one-step ahead variances of the linear minimum-mean-square error predictors (McLeod, 1977, p. 532),

$$|\hat{\mathcal{R}}_m| = \prod_{\ell=1}^m \hat{\sigma}_\ell^2, \quad (1.15)$$

where  $\hat{\sigma}_\ell^2$  is the mean-square error for a fitted linear predictor of order  $\ell$ . In this case,  $R_\ell^2 = 1 - \hat{\sigma}_\ell^2$ , where  $R_\ell^2$  is the square of the multiple correlation for the order  $\ell$  linear predictor, and so (Peña and Rodriguez, 2002, Equation 7),

$$|\hat{\mathcal{R}}_m| = \prod_{\ell=1}^m (1 - R_\ell^2). \quad (1.16)$$

In the multivariate case,

$$\hat{\eta}_\ell^2 = 1 - |\mathbb{I}_k - \hat{\mathfrak{R}}_{(\ell)} \hat{\mathfrak{R}}_{\ell-1}^{-1} \hat{\mathfrak{R}}'_{(\ell)}| \quad (1.17)$$

is the proportion of the generalized variance that is accounted for by a linear predictor of order  $\ell$ . From Equations 1.14 and 1.17, the corresponding multivariate equivalent of Equation 1.16 is

$$|\hat{\mathfrak{R}}_m| = \prod_{\ell=1}^m (1 - \hat{\eta}_\ell^2). \quad (1.18)$$

It follows from Equation 1.18 that  $|\hat{\mathfrak{R}}_m| < 1$  and that the smaller the value of  $|\hat{\mathfrak{R}}_m|$ , the more strongly autocorrelated the normalized residuals,  $\hat{\mathbf{L}}'\hat{\mathbf{a}}_t$ , are.

Using the Chitturi (1974) multivariate residual autocorrelations, Equation 1.6, the correlation matrix corresponding to Equation 2.10,  $\hat{\mathfrak{R}}_m^{(\ddagger)}$ , is defined by the block matrix with  $(i, j)$ -block,  $\hat{\mathbf{R}}_{i-j}^{(\ddagger)}$  for  $i, j = 1, \dots, m + 1$ . This matrix is not symmetric but  $|\hat{\mathfrak{R}}_m| = |\hat{\mathfrak{R}}_m^{(\ddagger)}|$ , so these multivariate autocorrelations could also be used.

Multivariate autocorrelations are often defined as in Equation 1.5 (Box et al., 2008, Equation 14.1.2). Using this definition, the residual autocorrelation matrix may be written,

$$\hat{\mathbf{R}}_\ell^{(\dagger)} = \hat{\mathbf{D}}^{-1/2} \hat{\mathbf{\Gamma}}_\ell \hat{\mathbf{D}}^{-1/2}, \quad (1.19)$$

where  $\hat{\mathbf{D}}^{-1/2} = \text{diag}(\hat{\gamma}_{1,1}^{-1/2}(0), \dots, \hat{\gamma}_{k,k}^{-1/2}(0))$ . The correlation matrix corresponding to Equation 2.10 obtained by replacing  $\hat{\mathbf{R}}_\ell$  by  $\hat{\mathbf{R}}_\ell^{(\dagger)}$  may be denoted by  $\hat{\mathfrak{R}}_\ell^{(\dagger)}$  and the corresponding generalized variance portmanteau statistic,  $|\hat{\mathfrak{R}}_m^{(\dagger)}|$ . A similar decomposition as given in Equation 1.18 shows that small values  $|\hat{\mathfrak{R}}_m^{(\dagger)}|$  correspond to positive autocorrelation. On the other hand, when there is no autocorrelation present, the off-block diagonal entries in the matrix  $\hat{\mathfrak{R}}_m^{(\dagger)}$  are  $O_p(n^{-1/2})$ . So,  $|\hat{\mathfrak{R}}_m^{(\dagger)}| \approx |\hat{\mathbf{R}}_0^{(\dagger)}|^{m+1}$ . When the innovation variance matrix,  $\mathbf{\Gamma}_0$ , has large off-diagonal elements,  $|\hat{\mathbf{R}}_0^{(\dagger)}| < 1$ . Hence again  $|\hat{\mathfrak{R}}_m^{(\dagger)}| = O_p(r^m)$  for some  $r \in (0, 1)$ . So, in both cases, autocorrelation or no autocorrelation,  $|\hat{\mathfrak{R}}_m^{(\dagger)}|$  tends to be small provided the innovation covariance matrix is not diagonal. Numerical experiments confirmed that the test using  $\mathfrak{D}_m^{(\dagger)}$  and  $\mathfrak{D}_m$  are essentially equivalent when  $\mathbf{\Gamma}_0$  is diagonal but in the non-diagonal case,  $\mathfrak{D}_m^{(\dagger)}$  does not provide a useful test.

### 1.2.1 Asymptotic distribution and approximation

In this section, the asymptotic distribution for  $\mathfrak{D}_m$  in Equation 1.13 is derived and an approximation to this distribution is suggested. Since, as shown in Lin and McLeod (2006, Figure 2) in the univariate case by simulation, the actual finite-sample distribution for  $\mathfrak{D}_m$  converges slowly, the asymptotic distribution for  $\mathfrak{D}_m$  is not expected to be of much use in diagnostic checking multivariate time series models unless  $n$  is very large.

We use the following notation as in Hosking (1980, §4),  $\Psi(\mathbf{B}) = \Phi(\mathbf{B})^{-1} \Theta(\mathbf{B}) =$

$\sum_{i=0}^{\infty} \Psi_i B^i$  and  $\mathbf{\Pi}(B) = \mathbf{\Theta}(B)^{-1} = \sum_{i=0}^{\infty} \Pi_i B^i$  are matrix power series such that the elements  $\Psi_i$  and  $\Pi_i$  converge exponentially to zero as  $i \rightarrow \infty$ . Define

$$\mathbf{G} = \begin{pmatrix} G_0 & 0 & \dots & 0 \\ G_1 & G_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ G_{m-1} & G_{m-2} & \dots & G_{m-p} \end{pmatrix}, \quad (1.20)$$

and

$$\mathbf{H} = \begin{pmatrix} H_0 & 0 & \dots & 0 \\ H_1 & H_0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ H_{m-1} & H_{m-2} & \dots & H_{m-q} \end{pmatrix}, \quad (1.21)$$

where  $G_r = \sum_{i=0}^{\infty} \mathbf{\Gamma}_0 \Psi_i' \otimes \Pi_{r-i}$  and  $H_r = \mathbf{\Gamma}_0 \otimes \Pi_r$ .

**Theorem 1.** *Assume that the model specified in Equation 1.1 has independent and identically distributed innovations with mean zero and constant covariance matrix. The model is fit to a series of length  $n$  using an  $n^{-1/2}$ -consistent algorithm. After obtaining the residuals defined in Equation 1.3 and the test statistic,  $\mathfrak{D}_m$ , in Equation 1.13,  $\mathfrak{D}_m$  is asymptotically distributed as*

$$\sum_{i=1}^{k^2 m} \lambda_i \chi_{1,i}^2,$$

where  $\chi_{1,i}^2$ ,  $i = 1, \dots, k^2 m$  are independent  $\chi_1^2$  random variables and  $\lambda_1, \dots, \lambda_{k^2 m}$  are the eigenvalues of  $(\mathbb{I}_{k^2} - \mathbf{Q})\mathbf{M}$ , where  $\mathbf{M}$  is  $k^2 m \times k^2 m$  diagonal matrix

$$\mathbf{M} = \begin{pmatrix} m\mathbb{I}_{k^2} & \mathbf{O} & \dots & \mathbf{O} \\ \mathbf{O} & (m-1)\mathbb{I}_{k^2} & \dots & \mathbf{O} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{O} & \mathbf{O} & \dots & \mathbb{I}_{k^2} \end{pmatrix}, \quad (1.22)$$

and

$$\mathbf{Q} = \mathbf{X}(\mathbf{X}'\mathbf{W}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}^{-1} \quad (1.23)$$

is an idempotent matrix with rank  $k^2(p+q)$ ,  $\mathbf{X}$  is defined as  $k^2m \times k^2(p+q)$  matrix  $(\mathbf{G} - \mathbf{H})$ , and  $\mathbf{W} = \mathbb{I}_m \otimes \mathbf{\Gamma}_0 \otimes \mathbf{\Gamma}_0$  is positive-definite symmetric.

*Proof.* From the decomposition in Equation 1.14, it follows that,

$$-n \log |\hat{\mathfrak{R}}_m| = -n \sum_{\ell=1}^m \log |\mathbb{I}_k - \mathbf{A}_\ell|, \quad (1.24)$$

where  $\mathbf{A}_\ell = \hat{\mathfrak{R}}_{(\ell)} \hat{\mathfrak{R}}_{\ell-1}^{-1} \hat{\mathfrak{R}}'_{(\ell)}$ . Using the fact that  $|\mathbb{I}_k - \mathbf{A}_\ell| = \prod_{i=1}^k (1 - \lambda_i(\ell))$ , where  $0 < \lambda_i(\ell) < 1$  are the eigenvalues of  $\mathbf{A}_\ell$ ,  $\ell = 1, \dots, m$ ,

$$-n \log |\hat{\mathfrak{R}}_m| = -n \sum_{\ell=1}^m \sum_{i=1}^k \log(1 - \lambda_i(\ell)). \quad (1.25)$$

Expanding  $\log(1 - \lambda_i(\ell)) = -\sum_{r=1}^{\infty} r^{-1} \lambda_i^r(\ell)$  and  $\text{tr}(\mathbf{A}_\ell) = \sum_{i=1}^k \lambda_i(\ell)$ ,

$$\mathfrak{D}_m = n \sum_{\ell=1}^m \text{tr}(\mathbf{A}_\ell) + O_p(n^{-1}). \quad (1.26)$$

One can verify that

$$\begin{aligned} \text{tr}(\mathbf{A}_1) &= \text{tr}(\hat{\mathbf{R}}'_1 \hat{\mathbf{R}}_1) \\ \text{tr}(\mathbf{A}_2) &\approx \text{tr}(\hat{\mathbf{R}}'_1 \hat{\mathbf{R}}_1) + \text{tr}(\hat{\mathbf{R}}'_2 \hat{\mathbf{R}}_2) \\ &\vdots \\ \text{tr}(\mathbf{A}_m) &\approx \text{tr}(\hat{\mathbf{R}}'_1 \hat{\mathbf{R}}_1) + \dots + \text{tr}(\hat{\mathbf{R}}'_m \hat{\mathbf{R}}_m), \end{aligned} \quad (1.27)$$

so that,

$$\mathfrak{D}_m \approx n \sum_{\ell=1}^m (m - \ell + 1) \text{tr}(\hat{\mathbf{R}}'_\ell \hat{\mathbf{R}}_\ell). \quad (1.28)$$

Using the commutative property of trace,

$$\mathfrak{D}_m \approx n \sum_{\ell=1}^m (m - \ell + 1) \operatorname{tr} (\hat{\Gamma}'_{\ell} \hat{\Gamma}_0^{-1} \hat{\Gamma}_{\ell} \hat{\Gamma}_0^{-1}). \quad (1.29)$$

It follows from [Neudecker \(1969, Equation 2.12\)](#),

$$\begin{aligned} \mathfrak{D}_m &\approx n \sum_{\ell=1}^m (m - \ell + 1) (\operatorname{vec} \hat{\Gamma}_{\ell})' (\hat{\Gamma}_0^{-1} \otimes \hat{\Gamma}_0^{-1}) \operatorname{vec} \hat{\Gamma}_{\ell}, \\ &= n (\operatorname{vec} \hat{\Gamma})' (\mathbb{I}_m \otimes \hat{\Gamma}_0^{-1} \otimes \hat{\Gamma}_0^{-1}) \mathbf{M} (\operatorname{vec} \hat{\Gamma}), \end{aligned} \quad (1.30)$$

where  $\operatorname{vec} \hat{\Gamma} = (\operatorname{vec} \hat{\Gamma}_1 \dots \operatorname{vec} \hat{\Gamma}_m)$  is  $k^2 m \times 1$  column vector and  $\mathbf{M}$  is  $k^2 m \times k^2 m$  diagonal matrix defined in [Equation 1.22](#).

[Hosking \(1980, Theorem 1\)](#) showed that

$$\sqrt{n} \operatorname{vec} \hat{\Gamma} \sim N_{k^2 m}(\mathbf{0}, (\mathbb{I}_{k^2 m} - \mathbf{Q}) \mathbf{W}), \quad (1.31)$$

where  $\mathbf{W}^{-1}$  can be replaced by a consistent estimator  $\hat{\mathbf{W}}^{-1} = \mathbb{I}_m \otimes \hat{\Gamma}_0^{-1} \otimes \hat{\Gamma}_0^{-1}$ , and  $\mathbf{Q}$  is the idempotent matrix of rank  $k^2(p + q)$  in [Equation 1.23](#).

From the theorem on quadratic forms given by [Box \(1954, Theorem 2.1\)](#), and [Equations 1.30, 1.31](#), the asymptotic distribution of  $\mathfrak{D}_m$  is given by,

$$\mathfrak{D}_m \rightarrow \sum_{i=1}^{k^2 m} \lambda_i \chi_1^2, \quad (1.32)$$

where  $\rightarrow$  stands for convergence in distribution as  $n \rightarrow \infty$  and  $\lambda_1, \dots, \lambda_{k^2 m}$  are the eigenvalues of  $(\mathbb{I}_{k^2 m} - \mathbf{Q}) \mathbf{M}$ . □

### 1.2.1.1 Approximation

The upper percentiles of the cumulative distribution function in [Equation 1.32](#) could be evaluated by the [Imhof \(1961\)](#) algorithm. For the univariate case, [Lin and McLeod](#)

(2006, Table 2) showed that the convergence to the asymptotic distribution is very slow. In the case of large-samples, an approximation based on Box (1954, Theorem 3.1) works well. Using this result, the test statistic in Equation 1.32 can be approximated by  $a\chi_b^2$ , where  $a$  and  $b$  are chosen to make the first two moments agree with those of exact distribution of  $\mathfrak{D}_m$ . Hence,  $a = \sum \lambda_i^2 / \sum \lambda_i$  and  $b = (\sum \lambda_i)^2 / \sum \lambda_i^2$ , where,

$$\begin{aligned} \sum_{i=1}^{k^2 m} \lambda_i &= \text{tr}[(\mathbb{I}_{k^2 m} - \mathbf{Q})\mathbf{M}], \\ \sum_{i=1}^{k^2 m} \lambda_i^2 &= \text{tr}[(\mathbb{I}_{k^2 m} - \mathbf{Q})\mathbf{M}(\mathbb{I}_{k^2 m} - \mathbf{Q})\mathbf{M}]. \end{aligned} \tag{1.33}$$

When  $p = q = 0$ ,  $a = (2m + 1)/3$  and  $b = 1.5k^2m(m + 1)/(2m + 1)$ . In the VARMA  $(p, q)$  case, one degree of freedom is lost for each parameter so  $\mathfrak{D}_m$  is approximately distributed as  $a\chi_b^2$ , where

$$\begin{aligned} a &= \frac{2m + 1}{3}, \\ b &= \frac{3k^2m(m + 1)}{2(2m + 1)} - k^2(p + q). \end{aligned} \tag{1.34}$$

### 1.2.2 Monte-Carlo significance test

Monte-Carlo significance tests, originally suggested by George Barnard (Barnard, 1963), are feasible for many small-sample problems (Marriott, 1979) and with modern computing facilities these types of tests are increasingly feasible for larger samples and more complex problems (Dufour and Khalaf, 2001). For a pure significance test with no nuisance parameters, as is the case, for example, for simply testing a time series for randomness, the accuracy of the Monte-Carlo procedure depends only on the number of simulations (Dufour, 2006, Proposition 2.1).

In the case of diagnostic checking, the model parameters must be estimated

and [Dufour \(2006, Proposition 5.1\)](#) has shown that, provided consistent estimators are used, Monte-Carlo tests remain asymptotically valid. Since we assume  $n^{-1/2}$ -consistent estimators are used, the requirements for [Dufour \(2006, Proposition 5.1\)](#) are met.

Simulations for  $\mathfrak{D}_m$  in the univariate case ([Lin and McLeod, 2006, Table 3](#)) as well as our simulations for the multivariate case in [Section 1.3.1](#), suggest the impact of nuisance parameters is negligible. The p-value for all of the portmanteau test statistics presented in this paper may be obtained using the Monte-Carlo method outlined below. We use the statistic  $\mathfrak{D}_m$  in the description but  $\tilde{Q}_m$  could be used instead.

**Step 1:** Set  $N$ , the number of simulations. Usually,  $N \leftarrow 1000$  but smaller values may be used if necessary. By choosing  $N$  large enough, an accurate estimate of the p-value may be obtained.

**Step 2:** After fitting the model and obtaining the residuals, compute the portmanteau test statistic for lag  $m$  or possibly a set of lags such as  $\ell = 1, \dots, m$ , where  $m \geq 1$ . Typically  $m$  is chosen large enough to allow for possible high-order autocorrelations. Denote the observed value of the test statistics by  $\mathfrak{D}_\ell^{(o)}, \ell = 1, \dots, m$ .

**Step 3:** For each  $i = 1, \dots, N$ , simulate the fitted model, refit it, obtain the residuals from this model, compute the test statistic,  $\mathfrak{D}_\ell^{(i)}, \ell = 1, \dots, m$ .

**Step 4:** For each  $\ell, \ell = 1, \dots, m$ , the estimated p-value is given by,

$$\hat{p} = \frac{\#\{\mathfrak{D}_\ell^{(i)} \geq \mathfrak{D}_\ell^{(o)}, i = 1, 2, \dots, N\} + 1}{N + 1}. \quad (1.35)$$

The approximate 95% margin of error for the p-value is,  $1.96\sqrt{\hat{p}(1-\hat{p})/N}$ .

The above algorithm is simply a restatement of the Monte-Carlo testing algorithm given by [Lin and McLeod \(2006, §3\)](#) for the univariate case. [Lin and McLeod \(2006, Table 3\)](#) demonstrate that the Monte-Carlo testing procedure has the correct size for an AR(1) and this is verified for some VAR(1) models in [Section 1.3.1](#).

*Remark 1.* In the Monte-Carlo test procedure it is assumed that the innovations used in our simulations in Step 3 are normally distributed but any distribution with constant covariance matrix could be used. In particular, using the empirical joint distribution is equivalent to bootstrapping the multivariate residuals. Using bootstrapped residuals is implemented in our software ([Mahdi and McLeod, 2011](#)).

*Remark 2.* A limitation of the Monte-Carlo diagnostic check is the assumption of constant variance. Many financial time series exhibit conditional heteroscedasticity. In practice this means that our test may overstate the significance level ([Duchesne and Lalancette, 2003](#)). This means that when used for constructing a VAR or VARMA model, the final fitted model may not be as parsimonious as a model developed using a portmanteau test which takes into account conditional heteroscedasticity ([Francq and Raïsi, 2007](#); [Duchesne, 2006](#)). Our Monte-Carlo portmanteau test can also be used to test for the presence of multivariate conditional heteroscedasticity simply by replacing the residuals by squared or absolute residuals. An illustration of this procedure is given later in [Section 1.4.2](#).

*Remark 3.* [Francq and Raïsi \(2007\)](#) discuss a more general asymptotic multivariate portmanteau diagnostic test that is valid assuming only that the innovations are uncorrelated. This test requires a large sample though.

*Remark 4.* [Lin and McLeod \(2008\)](#) discuss the Monte-Carlo portmanteau test for univariate ARMA with infinite variance. The Monte-Carlo method of [Lin and McLeod](#)

(2008) for infinite-variance ARMA has been extended to the multivariate case as well and is available in our R package (Mahdi and McLeod, 2011).

## 1.3 SIMULATION RESULTS

The purpose of our simulations is to demonstrate the improved power as well as the correct size of the Monte-Carlo (MC) test using  $\mathfrak{D}_m$ . We also compare the empirical Type 1 error rates for the  $a\chi_b^2$ -approximation discussed in Section 1.2.1.1.

### 1.3.1 Comparison of type 1 error rates

The empirical error rates have been evaluated under the Gaussian bivariate VAR(1) process  $\mathbf{Z}_t = \Phi_i \mathbf{Z}_{t-1} + \mathbf{a}_t$ ,  $i = 1, \dots, 4$  for the portmanteau test statistic  $\mathfrak{D}_m$  using the MC and  $a\chi_b^2$ -approximation to evaluate the p-value. The covariance matrix of  $\mathbf{a}_t$  has unit variances and covariance 1/2 and the coefficient matrices are taken from Hosking (1980) and Li and McLeod (1981),

$$\Phi_1 = \begin{pmatrix} 0.9 & 0.1 \\ -0.6 & 0.4 \end{pmatrix}, \Phi_2 = \begin{pmatrix} -1.5 & 1.2 \\ -0.9 & 0.5 \end{pmatrix}, \Phi_3 = \begin{pmatrix} 0.4 & 0.1 \\ -1.0 & 0.5 \end{pmatrix}, \Phi_4 = \begin{pmatrix} 0.3 & 0.5 \\ 0.0 & 0.3 \end{pmatrix}.$$

The empirical error 1% and 5% rates are shown in Table 1.1. For each entry in Table 1.1,  $10^3$  simulations were done. The MC test also used  $N = 10^3$ .

The 95% confidence interval assuming a 5% rejection rate for each test is (3.6, 6.4). At the 5% rejection rate, there are 17 entries outside this interval with the  $a\chi_b^2$  approximation and only one 1 with the Monte-Carlo test. In conclusion, size-distortion with the Monte-Carlo test appears to be negligible but is sometimes present when the  $a\chi_b^2$  approximation is used.

In Section 1.4, we found that there is a much larger discrepancy between the p-values using the  $a\chi_b^2$  approximation and those using the Monte-Carlo test.

		$\alpha = 1\%$						$\alpha = 5\%$					
		$n = 100$		$n = 200$		$n = 500$		$n = 100$		$n = 200$		$n = 500$	
$m$		$a\chi_b^2$	MC	$a\chi_b^2$	MC	$a\chi_b^2$	MC	$a\chi_b^2$	MC	$a\chi_b^2$	MC	$a\chi_b^2$	MC
$\Phi_1$													
5		1.2	0.8	1.2	0.9	1.0	0.7	5.9	4.6	5.1	4.7	4.8	4.8
10		1.2	0.7	1.0	0.8	0.8	0.9	5.2	4.5	4.4	5.2	3.7	4.2
15		1.3	0.7	1.1	0.7	0.8	0.9	5.7	5.4	4.5	4.4	3.6	3.8
20		1.6	0.7	1.1	0.7	0.8	0.9	6.8	5.8	4.8	4.0	3.8	3.8
25		2.0	0.8	1.2	0.7	0.9	0.8	7.8	4.9	5.3	4.1	4.0	4.0
30		2.5	1.1	1.4	0.7	0.9	0.8	9.0	4.8	5.8	3.7	4.4	4.1
$\Phi_2$													
5		0.9	0.9	0.8	1.2	0.7	1.0	4.7	4.8	4.0	4.8	3.5	4.7
10		1.0	0.7	0.9	1.3	0.6	1.4	4.8	3.8	3.8	4.0	3.5	4.8
15		1.2	1.0	1.0	0.8	0.7	1.5	5.7	3.9	4.3	3.9	3.6	5.0
20		1.7	0.8	1.0	0.8	0.7	1.2	6.9	4.2	4.9	4.2	3.8	4.8
25		2.0	0.7	1.0	0.7	0.7	1.0	8.2	4.0	5.3	3.9	4.1	5.3
30		2.4	0.7	1.3	0.6	0.7	1.0	9.5	4.3	5.8	4.0	4.5	5.4
$\Phi_3$													
5		0.7	0.9	0.9	1.2	0.6	1.0	4.0	4.6	3.6	5.7	3.2	5.2
10		1.0	0.7	0.8	1.5	0.7	0.8	4.5	4.8	3.8	6.5	3.1	5.3
15		1.1	0.9	0.8	0.8	0.6	1.0	5.1	4.2	4.1	6.3	3.3	5.1
20		1.5	0.9	0.9	0.7	0.7	1.3	6.6	4.3	4.6	6.2	3.6	5.2
25		1.9	0.9	1.0	0.7	0.8	1.0	7.7	4.5	5.3	5.4	4.0	5.3
30		2.3	0.8	1.2	0.8	0.8	1.3	9.0	4.2	5.9	5.5	4.3	5.0
$\Phi_4$													
5		0.6	0.8	0.6	1.1	0.5	1.0	2.9	4.3	2.6	4.7	2.5	5.2
10		0.8	0.8	0.7	1.1	0.6	1.0	3.9	4.6	3.2	4.9	3.0	4.5
15		1.0	0.9	0.9	1.0	0.6	1.0	4.9	4.1	3.9	4.6	3.2	5.0
20		1.3	1.2	0.9	1.0	0.7	1.0	6.1	4.4	4.5	5.3	3.6	4.9
25		1.8	0.9	1.1	1.0	0.7	1.0	7.3	3.9	5.0	5.0	3.9	4.8
30		2.2	1.0	1.3	1.0	0.8	0.9	8.7	3.9	5.6	5.2	4.3	4.7

Table 1.1: The empirical 1% and 5% significance levels, in percent, comparing approximation,  $a\chi_b^2$ , and Monte-Carlo, MC, for the portmanteau test statistic  $\mathfrak{D}_m$ .

### 1.3.2 Power comparisons

Only Monte-Carlo significance tests are used to compare the empirical power of 1% and 5% level tests with  $\tilde{Q}_m$  and  $\mathfrak{D}_m$ . Possible size-distortion sometimes makes power comparisons between asymptotic tests and Monte-Carlo tests invalid. In our comparisons, VAR models are fitted to various multivariate models. The power of diagnostic tests using  $\mathfrak{D}_m$  versus  $\tilde{Q}_m$  are compared using simulation. In all comparisons, the p-values were evaluated using the Monte-Carlo (MC) method with  $N = 10^3$ . We consider a VAR (1) model fitted to simulated data generated from eight VARMA models selected from well-known textbooks as cited below.

#### Model 1

[Lütkepohl \(2005, p. 17\)](#).

$$\begin{bmatrix} Z_{1,t} \\ Z_{2,t} \end{bmatrix} - \begin{bmatrix} 0.5 & 0.1 \\ 0.4 & 0.5 \end{bmatrix} \begin{bmatrix} Z_{1,t-1} \\ Z_{2,t-1} \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0.3 & 0 \end{bmatrix} \begin{bmatrix} Z_{1,t-2} \\ Z_{2,t-2} \end{bmatrix} = \begin{bmatrix} a_{1,t} \\ a_{2,t} \end{bmatrix}$$

$$\mathbf{\Gamma}_0 = \begin{pmatrix} 1.00 & 0.71 \\ 0.71 & 1.00 \end{pmatrix}$$

#### Model 2

[Brockwell and Davis \(1991, p. 428\)](#).

$$\begin{bmatrix} Z_{1,t} \\ Z_{2,t} \end{bmatrix} - \begin{bmatrix} 0.7 & 0 \\ 0 & 0.6 \end{bmatrix} \begin{bmatrix} Z_{1,t-1} \\ Z_{2,t-1} \end{bmatrix} = \begin{bmatrix} a_{1,t} \\ a_{2,t} \end{bmatrix} - \begin{bmatrix} 0.5 & 0.6 \\ -0.7 & 0.8 \end{bmatrix} \begin{bmatrix} a_{1,t-1} \\ a_{2,t-1} \end{bmatrix}$$

$$\mathbf{\Gamma}_0 = \begin{pmatrix} 1.00 & 0.71 \\ 0.71 & 2.00 \end{pmatrix}$$

### Model 3

Reinsel (1997, p. 81).

$$\begin{bmatrix} Z_{1,t} \\ Z_{2,t} \end{bmatrix} - \begin{bmatrix} 1.2 & -0.5 \\ 0.6 & 0.3 \end{bmatrix} \begin{bmatrix} Z_{1,t-1} \\ Z_{2,t-1} \end{bmatrix} = \begin{bmatrix} a_{1,t} \\ a_{2,t} \end{bmatrix} - \begin{bmatrix} -0.6 & 0.3 \\ 0.3 & 0.6 \end{bmatrix} \begin{bmatrix} a_{1,t-1} \\ a_{2,t-1} \end{bmatrix}$$

$$\mathbf{\Gamma}_0 = \begin{pmatrix} 1.00 & 0.50 \\ 0.50 & 1.25 \end{pmatrix}$$

### Model 4

Tsay [2005 2nd ed, p. 371].

$$\begin{bmatrix} Z_{1,t} \\ Z_{2,t} \end{bmatrix} - \begin{bmatrix} 0.8 & -2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Z_{1,t-1} \\ Z_{2,t-1} \end{bmatrix} = \begin{bmatrix} a_{1,t} \\ a_{2,t} \end{bmatrix} - \begin{bmatrix} -0.5 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} a_{1,t-1} \\ a_{2,t-1} \end{bmatrix}$$

$$\mathbf{\Gamma}_0 = \begin{pmatrix} 1.00 & 0.71 \\ 0.71 & 1.00 \end{pmatrix}$$

### Model 5

Reinsel (1997, p. 25).

$$\begin{bmatrix} Z_{1,t} \\ Z_{2,t} \end{bmatrix} = \begin{bmatrix} a_{1,t} \\ a_{2,t} \end{bmatrix} - \begin{bmatrix} 0.8 & 0.7 \\ -0.4 & 0.6 \end{bmatrix} \begin{bmatrix} a_{1,t-1} \\ a_{2,t-1} \end{bmatrix}$$

$$\mathbf{\Gamma}_0 = \begin{pmatrix} 4 & 1 \\ 1 & 2 \end{pmatrix}$$

### Model 6

Tsay [2005 2nd ed, p. 350].

$$\begin{bmatrix} Z_{1,t} \\ Z_{2,t} \end{bmatrix} = \begin{bmatrix} a_{1,t} \\ a_{2,t} \end{bmatrix} - \begin{bmatrix} 0.2 & 0.3 \\ -0.6 & 1.1 \end{bmatrix} \begin{bmatrix} a_{1,t-1} \\ a_{2,t-1} \end{bmatrix}$$

$$\mathbf{\Gamma}_0 = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$$

### Model 7

Lütkepohl (2005, p. 445).

$$\begin{bmatrix} Z_{1,t} \\ Z_{2,t} \end{bmatrix} - \begin{bmatrix} 0.5 & 0.1 \\ 0.4 & 0.5 \end{bmatrix} \begin{bmatrix} Z_{1,t-1} \\ Z_{2,t-1} \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 0.25 & 0 \end{bmatrix} \begin{bmatrix} Z_{1,t-2} \\ Z_{2,t-2} \end{bmatrix} = \\ \begin{bmatrix} a_{1,t} \\ a_{2,t} \end{bmatrix} - \begin{bmatrix} 0.6 & 0.2 \\ 0 & 0.3 \end{bmatrix} \begin{bmatrix} a_{1,t-1} \\ a_{2,t-1} \end{bmatrix}$$

$$\mathbf{\Gamma}_0 = \begin{pmatrix} 1.0 & 0.3 \\ 0.3 & 1.0 \end{pmatrix}$$

### Model 8

Reinsel et al. (1992, p. 141).

$$\begin{aligned}
\begin{bmatrix} Z_{1,t} \\ Z_{2,t} \\ Z_{3,t} \end{bmatrix} &= \begin{bmatrix} 0.4 & 0.3 & -0.6 \\ 0.0 & 0.8 & 0.4 \\ 0.3 & 0.0 & 0.0 \end{bmatrix} \begin{bmatrix} Z_{1,t-1} \\ Z_{2,t-1} \\ Z_{3,t-1} \end{bmatrix} + \\
&\begin{bmatrix} a_{1,t} \\ a_{2,t} \\ a_{3,t} \end{bmatrix} = \begin{bmatrix} 0.7 & 0.0 & 0.0 \\ 0.1 & 0.2 & 0.0 \\ -0.4 & 0.5 & -0.1 \end{bmatrix} \begin{bmatrix} a_{1,t-1} \\ a_{2,t-1} \\ a_{3,t-1} \end{bmatrix} \\
\mathbf{\Gamma}_0 &= \begin{pmatrix} 1.0 & 0.5 & 0.4 \\ 0.5 & 1.0 & 0.7 \\ 0.4 & 0.7 & 1.0 \end{pmatrix}
\end{aligned}$$

The power of the portmanteau statistics  $\mathcal{D}_m$  and  $\tilde{Q}_m$  for nominal 1% and 5% tests using the MC test are shown in Table 1.2. The power is evaluated for  $10^4$  simulations for each parameter setting and  $N = 10^3$  is used in the MC algorithm. It is clear from Table 1.2 that the  $\mathcal{D}_m$  test is often substantially more powerful than the  $\tilde{Q}_m$ . Only when  $n = 50$  and  $m = 30$  is the  $\tilde{Q}_m$  test more powerful and this only occurs for Models 2 and 4.

Model	$m$	$\alpha = 1\%$						$\alpha = 5\%$					
		$n = 50$		$n = 100$		$n = 200$		$n = 50$		$n = 100$		$n = 200$	
		$\mathfrak{D}_m$	$\tilde{Q}_m$										
1	5	14	10	44	25	87	71	35	24	68	53	96	90
1	10	8	6	32	13	77	47	24	16	55	36	90	73
1	15	6	5	22	10	68	32	18	14	46	30	85	61
1	20	4	4	18	9	58	25	13	13	39	26	80	52
1	30	3	3	12	7	45	18	10	12	30	23	68	43
2	5	43	22	98	78	100	100	70	48	100	94	100	100
2	10	35	15	97	52	100	99	60	38	99	82	100	100
2	15	25	13	95	44	100	95	50	35	99	75	100	100
2	20	19	12	90	37	100	91	43	34	97	70	100	99
2	30	10	12	81	31	100	82	28	37	93	64	100	97
3	5	94	57	100	100	100	100	99	84	100	100	100	100
3	10	88	26	100	94	100	100	96	64	100	99	100	100
3	15	80	15	100	78	100	100	93	48	100	97	100	100
3	20	70	13	100	61	100	100	88	39	100	91	100	100
3	30	49	10	99	37	100	98	73	36	100	77	100	100
4	5	26	10	80	36	100	90	51	27	93	62	100	98
4	10	15	8	62	21	99	68	37	24	84	48	100	89
4	15	9	6	47	17	97	52	27	22	74	40	99	81
4	20	7	7	38	14	93	44	20	22	65	37	98	73
4	30	4	6	25	10	86	34	13	22	53	33	95	65
5	5	99	49	100	100	100	100	99	68	100	100	100	100
5	10	95	25	100	84	100	100	95	46	100	93	100	100
5	15	88	19	100	66	100	100	90	36	100	81	100	100
5	20	80	16	100	50	100	100	83	32	100	72	100	100
5	30	60	15	100	35	100	96	69	30	100	60	100	97
6	5	79	23	100	83	100	100	83	45	100	90	100	100
6	10	67	13	100	49	100	99	74	32	100	69	100	100
6	15	55	11	100	36	100	95	62	28	99	57	100	96
6	20	43	9	99	26	100	85	54	28	98	52	100	92
6	30	25	8	96	17	100	69	40	27	95	44	100	84
7	5	12	8	44	24	90	72	29	21	65	49	97	91
7	10	7	4	30	11	80	45	19	14	53	33	92	74
7	15	5	4	22	8	70	30	14	12	43	27	86	61
7	20	3	3	17	6	62	24	13	11	35	22	82	53
7	30	3	4	12	5	47	17	11	11	27	19	72	41
8	5	55	12	85	58	100	99	77	28	96	85	100	100
8	10	40	7	76	33	100	91	65	19	92	61	100	99
8	15	26	6	63	21	100	79	52	17	84	48	100	94
8	20	16	5	51	16	100	67	38	14	76	40	100	90
8	30	5	5	31	11	98	47	15	13	55	33	100	78

Table 1.2: Empirical power in percent comparison of  $\mathfrak{D}_m$  and  $\tilde{Q}_m$  for nominal 1% and 5% tests.  $10^4$  simulations with  $N = 10^3$ .

## 1.4 ILLUSTRATIVE APPLICATIONS

### 1.4.1 IBM and S&P index

[Tsay \(2010, Chapter 8\)](#) uses the portmanteau diagnostic test in constructing a VAR model for the monthly log returns of IBM stock and the S&P 500 index for January 1926 to December 2008. So here,  $n = 996$ . Univariate analysis for both of these series indicates the presence of conditional heteroscedasticity ([Tsay, 2010, p. 408](#)) but for forecasting purposes, we may consider a VAR model rather than a more complex VAR/GARCH model ([Weiss, 1984; Francq and Raïsi, 2007](#)). The AIC selects a VAR(5) model. We found that the BIC selects a VAR(1) model. [Table 1.3](#) compares the p-values for the portmanteau tests for the VAR( $p$ ) for  $p = 1, 3, 5$ .

These portmanteau tests suggest that the VAR(5) is adequate and that the VAR(1) and VAR(3) both exhibit lack of fit. The VAR(4) is not shown but the results for this model are similar to the VAR(3). As noted in Remark 2, the presence of conditional heteroscedasticity means that the p-values in [Table 1.3](#) are too small and this implies that, possibly, a lower order model than the VAR(5) may be adequate. This possibility could be investigated using the multivariate portmanteau test of [Francq and Raïsi \(2007\)](#).

[Table 1.3](#) also shows that  $a\chi_b^2$  approximation for the p-value of  $\mathfrak{D}_m$  is inaccurate whereas for  $\tilde{Q}_m$  the asymptotic approximation agrees quite well with the Monte-Carlo result.

### 1.4.2 Investment, income and consumption time series

The trivariate quarterly time series, 1960–1982, of West German investment, income, and consumption was discussed by [Lütkepohl \(2005, §3.2.3\)](#). For this series,  $n = 92$  and  $k = 3$ . As in [Lütkepohl \(2005, §4.3.1\)](#) we model the logarithms of the first

m	VAR(1)				VAR(3)				VAR(5)			
	$a\chi_b^2$		MC		$a\chi_b^2$		MC		$a\chi_b^2$		MC	
	$\mathfrak{D}_m$	$\tilde{Q}_m$										
5	0.2	*	*	*	10.4	0.6	1.7	0.6	NA	NA	91.2	89.9
10	0.1	0.3	*	0.2	13.5	6.1	2.8	4.0	77.4	50.3	59.4	50.2
15	0.3	2.1	*	2.2	20.4	22.3	6.4	22.1	84.0	61.2	63.1	61.4
20	0.2	*	*	*	15.4	2.6	5.0	2.2	71.8	11.3	45.2	9.9
25	0.1	*	*	*	8.7	1.1	2.3	0.7	53.0	7.6	27.5	7.1
30	0.2	*	*	*	7.3	2.7	2.3	2.2	46.2	13.7	23.3	12.0

Table 1.3: IBM and S&P 500 Index Data.  $a\chi_b^2$ : approximation. MC: Monte-Carlo  $N = 10^3$ . NA: not applicable. The p-values are in percent. The \* indicates a p-value less than 0.1%.

differences. Using the AIC, Lütkepohl (2005, Table 4.5) selected a VAR(2) for this data. Only lags  $m = 5, 10, 15$  are used in the diagnostic checks since  $n$  is relatively short. All diagnostic tests reject simple randomness, VAR(0). The Monte-Carlo tests for VAR(1) suggests model inadequacy at lag 5. Table 1.4 supports the choice of the VAR(2) model.

m	VAR(0)				VAR(1)				VAR(2)			
	$a\chi_b^2$		MC		$a\chi_b^2$		MC		$a\chi_b^2$		MC	
	$\mathfrak{D}_m$	$\tilde{Q}_m$										
5	*	*	0.1	0.1	3.1	4.7	2.2	4.8	33.1	29.8	31.2	38.0
10	*	0.6	0.3	0.5	4.0	14.7	7.0	12.7	49.5	48.0	54.2	50.6
15	*	0.2	0.4	0.6	4.1	13.7	17.7	12.4	32.8	34.6	56.2	35.5

Table 1.4: Trivariate West German Macroeconomic Series.  $a\chi_b^2$ : approximation. MC: Monte-Carlo using  $10^3$  replications. The p-values are in percent and \* indicates a p-value less than 0.1%.

As pointed out in Remark 2, we may test for multivariate heteroscedasticity by using the squared residuals and Table 1.5 gives the p-values with this test for the VAR(2) model. In this case,  $a\chi_b^2$  approximation for  $\mathfrak{D}_m$  as well as the asymptotic  $\chi^2$

approximation for  $\tilde{Q}_m$  are quite inaccurate. Based on the Monte-Carlo tests there is little evidence to reject that null hypothesis of constant variance.

m	$a\chi_b^2$		MC	
	$\mathfrak{D}_m$	$\tilde{Q}_m$	$\mathfrak{D}_m$	$\tilde{Q}_m$
5	0.2	15.2	31.9	81.3
10	0.3	6.3	24.4	37.9
15	*	*	12.2	1.6

Table 1.5: The residuals of the fitted VAR(2) model on West German Macroeconomic series are tested for heteroscedastic effects.  $a\chi_b^2$ : approximation. MC: Monte-Carlo using  $10^3$  replications. The p-values are in percent and \* indicates a p-value less than 0.1%.

## 1.5 CONCLUDING REMARKS

[Box et al. \(2008\)](#) stress the importance of constructing an adequate and parsimonious model in which the residuals pass a suitable portmanteau diagnostic check. In forecasting experiments with monthly riverflow time series, [Noakes et al. \(1985\)](#) found that simply using a criterion such as the AIC or BIC may provide a model that either does not pass a suitable diagnostic check for randomness of the residuals or that may have more parameters than necessary. Monthly riverflow time series models chosen with the fewest number of parameters that pass the portmanteau diagnostic check for periodic autocorrelation ([McLeod, 1994](#)) tend to produce better one-step ahead forecasts ([Noakes et al., 1985](#)). [McLeod \(1993\)](#) suggested formulating the principle of parsimony as an optimization problem: minimize model complexity subject to model adequacy. In any case, in the overall approach suggested many years ago and presented in their recent book ([Box et al., 2008](#)), portmanteau diagnostic checks play a crucial role in constructing time series models.

In Section 1.2.2, Remark 2, it was pointed out the Monte-Carlo test with  $\mathfrak{D}_m$  may also be useful in diagnostic checking for multivariate conditional heteroscedasticity when used with squared or absolute residuals. This test is implemented in [Mahdi and McLeod \(2011\)](#). There is an extensive literature on testing residuals in AR and ARMA models for conditional heteroscedasticity ([Ling and Li, 1997](#); [Duchesne and Lalancette, 2003](#); [Duchesne, 2004](#); [Rodríguez and Ruiz, 2005](#); [Duchesne, 2006](#); [Chabot-Hall and Duchesne, 2008](#)). The power study presented Section 1.3.2 suggests that the  $\mathfrak{D}_m$  with squared or absolute residuals may be useful. [Peña and Rodríguez \(2002\)](#) also suggested that using squared-residuals with their generalized-variance portmanteau test would outperform the usual diagnostic check ([McLeod and Li, 1983](#)). Other tests designed for particular alternatives might be expected to perform better than an

omnibus portmanteau test such as  $\mathcal{D}_m$  or  $\tilde{Q}_m$  when these alternatives hold. For example, [Rodríguez and Ruiz \(2005\)](#) developed a diagnostic check for heteroscedasticity for the case of small autocorrelations.

The multivariate portmanteau diagnostic test developed by [Francq and Raïsi \(2007\)](#) does not require independent and identically innovations but only uncorrelated innovations. This test would be appropriate for the bivariate example in [Section 1.4.1](#).

Scripts for reproducing all tables in this paper are available with our freely available software ([Mahdi and McLeod, 2011](#)). This package can utilize multicore CPUs often found in modern personal computers as well as a computer cluster or grid ([Schmidberger et al., 2009](#)). On a modern eight core personal computer, the computations for [Tables 1.4](#) and [1.5](#) take about one minute. [Table 1.3](#) takes about six minutes due to the longer series length and increased number of lags. The simulations reported in [Section 1.3](#) were run on a computer cluster.

## REFERENCES

- Barnard, G. A. (1963). Discussion of "The spectral analysis of point processes" by M. S. Bartlett. *Journal of the Royal Statistical Society, B* 25, 264–296.
- Box, G. (1954). Some theorems on quadratic forms applied in the study of analysis of variance problems, I. effect of inequality of variance in the one-way classification. *The Annals of Mathematical Statistics* 25(2), 290–302.
- Box, G. and Pierce, D. (1970). Distribution of residual autocorrelation in autoregressive-integrated moving average time series models. *Journal of American Statistical Association* 65(332), 1509–1526.
- Box G., Jenkins G., and Reinsel, G. C. (2008). *Time Series Analysis: Forecasting and Control* (4th ed.). New York: Wiley.
- Brockwell, P. J. and Davis, R. A. (1991). *Time series: Theory and Methods* (2nd ed.). New York: Springer-Verlag.
- Chabot-Hall, D. and Duchesne, P. (2008). Diagnostic checking of multivariate nonlinear time series models with martingale difference errors. *Statistics and Probability Letters* 78(8), 997–1005.
- Chitturi, R. V. (1974). Distribution of residual autocorrelations in multiple autoregressive schemes. *Journal of the American Statistical Association* 69(348), 928–934.
- Chitturi, R. V. (1976). Distribution of multivariate white noise autocorrelations. *Journal of the American Statistical Association* 71(353), 223–226.
- Duchesne, P. (2004). On robust testing for conditional heteroscedasticity in time series models. *Computational Statistics and Data Analysis* 46, 227–256.
- Duchesne, P. (2006). Testing for multivariate autoregressive conditional heteroskedasticity using wavelets. *Computational Statistics and Data Analysis* 51(353), 2142–2163.
- Duchesne, P. and Lalancette, S. (2003). On testing for multivariate ARCH effects in vector time series models. *The Canadian Journal of Statistics* 31(3), 275–292.
- Dufour, J.-M. (2006). Monte Carlo tests with nuisance parameters: A general approach to finite-sample inference and non-standard asymptotics. *Journal of Econometrics* 133(2), 443–477.

- Dufour, J.-M. and Khalaf, L. (2001). Monte-Carlo test methods in econometrics. In B. Baltagi (Ed.), *Companion to Theoretical Econometrics*, Chapter 23, pp. 494–519. Oxford: Blackwell.
- Francq, C. and Raïsi, H. (2007). Multivariate portmanteau test for autoregressive models with uncorrelated but nonindependent errors. *Journal of Time Series Analysis* 28(3), 454–470.
- Hosking, J. R. M. (1980). The multivariate portmanteau statistic. *Journal of American Statistical Association* 75(371), 602–607.
- Hosking, J. R. M. (1981a). Equivalent forms of the multivariate portmanteau statistic. *Journal of the Royal Statistical Society, Series B* 43(2), 261–262.
- Hosking, J. R. M. (1981b). Lagrange-multiplier tests of multivariate time-series models. *Journal of the Royal Statistical Society, Series B (Methodological)* 43(2), 219–230.
- Imhof, J. P. (1961). Computing the distribution of quadratic forms in normal variables. *Biometrika* 48, 419–426.
- Li, W. K. (2004). *Diagnostic checks in time series*. New York: Chapman and Hall/CRC.
- Li, W. K. and McLeod, A. I. (1981). Distribution of the residual autocorrelation in multivariate arma time series models. *Journal of the Royal Statistical Society, Series B* 43(2), 231–239.
- Lin, J.-W. and McLeod, A. I. (2006). Improved Peña-Rodríguez portmanteau test. *Computational Statistics and Data Analysis* 51(3), 1731–1738.
- Lin, J.-W. and McLeod, A. I. (2008). Portmanteau tests for ARMA models with infinite variance. *Journal of Time Series Analysis* 29(3), 600–617.
- Ling, S. and Li, W. K. (1997). Diagnostic checking of nonlinear multivariate time series with multivariate ARCH errors. *Journal of Time Series Analysis* 18(5), 447–464.
- Ljung, G. M. and Box, G. E. P. (1978). On a measure of lack of fit in time series models. *Biometrika* 65, 297–303.
- Lütkepohl, H. (2005). *New Introduction to multiple time series analysis*. New York: Springer-Verlag.
- Mahdi, E. and McLeod, A. I. (2011). *portes: Portmanteau tests for time series models*.

- Marriott, F. H. C. (1979). Barnard's Monte Carlo tests: How many simulations? *Applied Statistics* 28(1), 75–77.
- McLeod, A. I. (1977). Improved Box-Jenkins estimators. *Biometrika* 64(3), pp. 531–534.
- McLeod, A. I. (1993). Parsimony, model adequacy and periodic correlation in forecasting time series. *International Statistical Review* 61(3), 387–393.
- McLeod, A. I. (1994). Diagnostic checking periodic autoregression models with application. *The Journal of Time Series Analysis* 15(3), 221–233. Addendum, JTSA 16, 647–648.
- McLeod, A. I. and Li, W. K. (1983). Diagnostic checking ARMA time series models using squared-residual autocorrelations. *The Journal of Time Series Analysis* 4, 269–273.
- Neudecker, H. (1969). Some theorems on matrix differentiation with special reference to kronecker matrix products. *Journal of American Statistical Association* 64(327), 953–962.
- Noakes, D. J., McLeod, A. I., and Hipel, K. W. (1985). Forecasting seasonal hydrological time series. *The International Journal of Forecasting* 1(1), 179–190.
- Peña, D. and Rodríguez, J. (2002). A powerful portmanteau test of lack of test for time series. *Journal of American Statistical Association* 97(458), 601–610.
- Peña, D. and Rodríguez, J. (2006). The log of the determinant of the autocorrelation matrix for testing goodness of fit in time series. *Journal of Statistical Planning and Inference* 8(136), 2706–2718.
- Poskitt, D. S. and Tremayne, A. R. (1982). Diagnostic tests for multiple time series models. *Annals of Statistics* 6(1), 114–120.
- Reinsel, G. C. (1997). *Elements of multivariate time series analysis* (2nd ed.). New York: Springer-Verlag.
- Reinsel, G. C., Basu, S. and Yap, S. F. (1992). Maximum likelihood estimators in the multivariate autoregressive moving average model from a generalized least squares viewpoint. *Journal of Time Series Analysis* 13(2), 133–145.
- Rodríguez, J. and Ruiz, E. (2005). A powerful test for conditional heteroscedasticity for financial time series with highly persistent volatilities. *Statistica Sinica* 15, 505–525.

- Schmidberger, M., Morgan, M., Eddelbuettel, D., Yu, H., Tierney, L., and Mansmann, U. (2009). State of the art in parallel computing with R. *Journal of Statistical Software* 31(1), 1–27.
- Seber, G. A. F. (2008). *A matrix handbook for statisticians*. New York: Wiley.
- Tsay, R. S. (2010). *Analysis of financial time series* (3rd ed.). (2nd ed. 2005). New York: Wiley.
- Weiss, A. A. (1984). ARMA models with ARCH errors. *Journal of Time Series Analysis* 5(2), 129–143.

## Chapter 2

### PORTMANTEAU TESTS FOR TIME SERIES MODELS: R PACKAGE portes

#### 2.1 INTRODUCTION

The multivariate vector integrated autoregressive moving average, VARIMA  $(p, \mathbf{d}, q)$ , model with mean vector  $\boldsymbol{\mu}$  and deterministic equation  $\mathbf{a} + \mathbf{b}t$  for a  $k$ -dimensional time series  $\mathbf{Z}_t = (Z_{1,t}, \dots, Z_{k,t})'$  can be written as

$$\boldsymbol{\Phi}(\mathbf{B})\mathbf{D}(\mathbf{B})(\mathbf{Z}_t - \boldsymbol{\mu}) = \mathbf{a} + \mathbf{b}t + \boldsymbol{\Theta}(\mathbf{B})\mathbf{e}_t, \quad (2.1)$$

where  $\boldsymbol{\Phi}(\mathbf{B}) = \mathbb{I}_k - \boldsymbol{\Phi}_1\mathbf{B} - \dots - \boldsymbol{\Phi}_p\mathbf{B}^p$ ,  $\boldsymbol{\Theta}(\mathbf{B}) = \mathbb{I}_k - \boldsymbol{\Theta}_1\mathbf{B} - \dots - \boldsymbol{\Theta}_q\mathbf{B}^q$  and  $\mathbb{I}_k$  is a  $k \times k$  identity matrix,  $\boldsymbol{\Phi}_\ell = (\phi_{ij,\ell})_{k \times k}$ ,  $\boldsymbol{\Theta}_\ell = (\theta_{ij,\ell})_{k \times k}$  are coefficient matrices, and  $\mathbf{B}$  is a backshift operator such as  $\mathbf{B}^j\mathbf{Z}_t = \mathbf{Z}_{t-j}$ .  $\mathbf{D}(\mathbf{B}) = \text{diag}[(1-\mathbf{B})^{d_1}, \dots, (1-\mathbf{B})^{d_k}]$  is a diagonal  $k \times k$  matrix, where  $\mathbf{d} = (d_1, \dots, d_k)$ ,  $d_i \geq 0$ . This states that each individual series  $Z_i, i = 1, \dots, k$  is differenced  $d_i$  times to reduce to a stationary VARMA  $(p, q)$  series. It is assumed that the VARMA model is stationary, invertible, and identifiable (Reinsel, 1997; Box et al., 2008). The coefficients  $\mathbf{a}$  and  $\mathbf{b}$  represent the constant drift and the deterministic time trend respectively and the white noise process  $\mathbf{e}_t = (e_{1,t}, \dots, e_{k,t})'$  is assumed to be uncorrelated in time with mean zero; that is,  $E(\mathbf{e}_t) = \mathbf{0}$  and  $E(\mathbf{e}_t\mathbf{e}'_{t-\ell}) = \boldsymbol{\Gamma}_0\delta_\ell$ , where  $\boldsymbol{\Gamma}_0$  is the  $k \times k$  positive definite variance covariance matrix and  $\delta_\ell$  is the usual Kronecker delta with unity at  $\ell = 0$  and zero elsewhere.

In univariate time series, i.e. when  $k = 1$ , the model in Equation 2.1 reduces to be an integrated autoregressive moving average, ARIMA  $(p, d, q)$ , model

$$\phi(B)D(B)(Z_t - \mu) = a + bt + \theta(B)e_t, \quad (2.2)$$

where  $a$  and  $b$ , are the drift and the trend terms respectively,  $\phi(B) = 1 - \phi_1 B - \dots - \phi_p B^p$ ,  $\theta(B) = 1 - \theta_1 B - \dots - \theta_q B^q$ ,  $D(B) = d$  is the differencing order, and  $e_t$  is the white noise series with mean zero and variance  $\sigma^2$  (Hannan, 1969; Pfaff, 2006; Box et al., 2008).

After fitting the ARIMA or VARIMA model using efficient estimators, the residuals,  $\hat{e}_{i,t}$  where  $i = 1, \dots, k$  and  $t = 1, \dots, n$  can be obtained. We may then use portmanteau goodness-of-fit test to test the adequacy of the fitted model by checking whether the residuals are approximately white noise (Li, 2004).

The most popular portmanteau tests that were introduced by Box and Pierce (1970); Ljung and Box (1978); Hosking (1980); Li and McLeod (1981), with a new portmanteau generalized variance test based on the determinant of the standardized multivariate residual autocorrelations (Peña and Rodriguez, 2002, 2006; Lin and McLeod, 2006; Mahdi and McLeod, 2011) are implemented in the R package **portes**. In addition, the portmanteau tests for nonlinear structure in time series as proposed by McLeod and Li (1983) and for Fractional Gaussian Noise, FGN, stochastic model (McLeod et al., 2007b) that possesses long memory, are also implemented in this package. Adler et al. (1998) studied ARMA models with identical and independent innovations from stable distribution with infinite variances and Lin and McLeod (2008) developed a generalized variance portmanteau test for such models using Monte-Carlo techniques and this test is also included in this package. Any other fitted model, such as threshold autoregressive models, TAR, can be also tested for adequacy using this

proposed package (Cryer and Chan, 2010). Brief descriptions for different portmanteau tests are given in Section 2.3.

The concern of using the portmanteau tests is that the accuracy of the asymptotic distributions requires  $n$  and  $m$  large, where  $n$  denotes the length of the series and  $m$  is the lag time value. The development of powerful computers, specially those with multicore systems, in which the R packages **Rmpi** (Yu, 2002) and **snow** (Tierney et al., 2011) work efficiently with parallel computing, has made Monte-Carlo tests more affordable and recommended in such cases as simulation experiments show that the Monte-Carlo tests are usually more accurate and powerful than asymptotic distributions (Lin and McLeod, 2006; Mahdi and McLeod, 2011).

In Sections 2.2 and 2.4 we describe the main function, `portest()`, with some illustrative applications explaining the Monte-Carlo version of Peña and Rodriguez (2002); Mahdi and McLeod (2011) portmanteau test with implementation to **snow** package. This function can be used for testing whiteness of series as well as for testing adequacy of fitted ARMA and VARMA models with finite or infinite variances, GARCH effects, FGN models, or any other fitted model. To reproduce the results in these sections, one should install the packages **fGarch**, **FGN**, **snow**, **TSA**, and **vars**, on a computer with a minimum requirement of dual core. These packages are available from the Comprehensive R Archive Network website, CRAN.

```
R> library("fGarch")
```

```
R> library("FGN")
```

```
R> library("snow")
```

```
R> library("vars")
```

The proposed R package **portes** is developed so one can simulate time series from univariate ARIMA process or multivariate VARIMA process with innovations of finite

or infinite variance from stable distribution (see Section 2.5.2). The simulated data may have a deterministic constant drift and time trend term with non-zero mean. It can be used for testing stationarity and invertibility and for estimating stable parameters of data from stable distribution. The package **portes** is available from the Comprehensive R Archive Network, CRAN, at <http://CRAN.R-project.org/package=portes> version 1.08 and can be installed in the usual ways and is ready to use after typing

```
R> library("portes")
```

## 2.2 THE MAIN FUNCTION: portest

In this section we describe the main function, `portest()`. This function implements the portmanteau test statistics as given in [Box and Pierce \(1970\)](#); [Ljung and Box \(1978\)](#); [Hosking \(1980\)](#); [Li and McLeod \(1981\)](#); [McLeod and Li \(1983\)](#) with a new portmanteau test statistic based on the determinant of the standardized univariate or multivariate residual autocorrelations ([Peña and Rodriguez, 2002, 2006](#); [Lin and McLeod, 2006, 2008](#); [Mahdi and McLeod, 2011](#)). The p-values can be evaluated using the Monte-Carlo techniques and the approximate asymptotic chi-square distribution. The syntax of `portest()` is listed below:

```
portest(obj,lags=seq(5,30,5),order=0,test=c("gvtest","BoxPierce",  
      "LjungBox","Hosking","LiMcLeod"),MonteCarlo=TRUE,nslaves=1,  
      NREP=1000,InfiniteVarianceQ=FALSE,SquaredQ=FALSE,  
      func=list(SimModel=NULL,FitModel=NULL),pkg=NULL,SetSeed=TRUE)
```

### 2.2.1 Monte-Carlo goodness-of-fit test

The minimal required input of the function `portest()` is `portest(obj)`. By selecting this argument, the generalized variance portmanteau statistic, `gvtest()`, that is described in Section 2.3.5 will be implemented using the Monte-Carlo approach with one thousand replications on a single CPU at lags 5, 10, 15, 20, 25, and 30.

As an illustrative example, consider the univariate series `DEXCAUS`. Data is available from our R Package `portes` and refer to daily Canada/US foreign exchanges rates from January 04, 1971 to September 05, 1996. A complete description of the data can be retrieved simply by typing `?DEXCAUS` in R session. In this example, the `portest()` function implements the Monte-Carlo version of the statistic `gvtest` with  $10^3$  replications using a single CPU and test for randomness of returns. The results suggest that the `returns` series behaves like random series. With respect to computer time, the CPU time equals to 49.61 seconds to get the output of this example.

```
R> data("DEXCAUS")
R> returns <- log(DEXCAUS[-1]/DEXCAUS[-length(DEXCAUS)])
R> portest(returns)
```

Lags	Statistic	df	p-value
5	5.726436	4.090909	0.2237762
10	12.002784	7.857143	0.1478521
15	16.798015	11.612903	0.1598402
20	20.847948	15.365854	0.1718282
25	23.805231	19.117647	0.2227772
30	27.248101	22.868852	0.2477522

For the Monte-Carlo significance test, the function `portest()` with an object `obj` of class `"list"` requires an additional two arguments, `func` and `pkg` as a minimal

required inputs. The argument `func` is a list with two functions, `SimModel()` and `FitModel()`. In this case, users should write their own R code of these two functions and make sure that the class output of the `FitModel()` function is a "list" with at least one components, `res`, where `res` stands for the residuals of the fitted model. This list of outputs is then passes the argument of the function `SimModel()`. The output of the function `SimModel()` should be a simulated univariate or multivariate time series from the fitted model. The name of the R library that contains the function used in the fitted model should be entered via the argument `pkg` (see the simple example given in Section 2.4.7).

In general, the input argument `obj` must be an object with class "ar", "Arima", "arima0", "varest", "FitAR", "FitFGN", "garch", "fGARCH", "numeric", "ts", "matrix", "mts & ts", or "list". The classes "Arima", "arima0", "FitAR", "FitFGN", "garch", "fGARCH", "numeric", and "ts" are used with univariate time series, whereas the classes "matrix", and "mts & ts" are used with multivariate time series. The functions `ar()`, `ar.yw()`, and `ar.ols()` with a returned object of class "ar" work with univariate or multivariate time series while the function `ar.burg()` returns an object with a class "ar" that can be used with the univariate time series only. The class object "varest" is associated with the output of the function `VAR()` from the R `vars` package under the multivariate time series case (Pfaff, 2010). Note that the function `arima()` in the `stats` package and the functions `auto.arima()`, `Arima()` in the `forecast` package have output object with class "Arima", whereas the function `arima0()` has output object with class "arima0". Note also that the functions `FitAR()`, `FitARp()`, and `FitARz()` in the R package `FitAR` have output objects with class "FitAR" (McLeod and Zhang, 2011). The function `FitFGN()` in the R package `FGN` has output object with class "FitFGN" (McLeod et al., 2011) whereas the functions `garch()` and `garchFit()` return objects with classes "garch" and "fGARCH"

respectively (Trapletti, 2011; Wuertz and core team members, 2011). By default, the argument `MonteCarlo = TRUE` in the main function, `portest()`, implements the Monte-Carlo version of the portmanteau test statistic, whereas the approximation asymptotic distribution of the portmanteau test statistic will be implemented by selecting the argument `MonteCarlo = FALSE`. The argument `nslaves = 1` is used with one single CPU, whereas `nslaves > 1` is used with multiple CPU's, provided that the `snow` package is loaded and the argument `MonteCarlo = TRUE` is selected (Tierney et al., 2011),

For objects with classes `"ar"`, `"Arima"`, `"arima0"`, `"varest"`, `"FitAR"`, `"FitFGN"`, `"garch"`, `"fgARCH"`, and `"list"`, the Monte-Carlo goodness-of-fit test checks the adequacy of the fitted model under the null hypothesis using the following steps:

1. Get the residuals from the input object `obj` and calculate the observed value of the portmanteau test at lag  $m$ , say  $\mathfrak{D}_m^0$ . If the argument `SquaredQ = TRUE` is selected then the observed value of the portmanteau test will be calculated for the square of residuals to test for nonlinearity structure in time series (McLeod and Li, 1983).
2. Apply the function `arima.sim()`, `SimulateFGN()`, `garchSim()`, or `varima.sim()` on the extracted estimated parameters from the ARMA, FGN, GARCH, or VAR models respectively and simulate a new time series where the exact distribution of the extracted residuals are used for bootstrapping the simulated innovations series. Users may write their own fitted and simulated functions and pass these two functions via the argument `func`, provided that the output of the fitted has an object `obj` with class `"list"`. The argument `InfiniteVarianceQ = TRUE` is used for models with innovations from stable distribution with infinite variance, while the default one is used with innovations with finite variance

(Adler et al., 1998; Lin and McLeod, 2008).

3. Use the same function used before to fit the same model to the simulated series.
4. Extract the residuals from this fitted model and calculate the simulated portman-teau test as described in step 1, say  $\mathfrak{D}_m^i, i = 1, 2, \dots$
5. Repeat Steps 2 to 4 NREP times, where NREP is an argument represents the number of replications needed to use in Monte-Carlo test.

The Monte-Carlo p-values associated with the different lag values (Dufour, 2006; Davison and Hinkley, 1997, Chapter 4) are calculated using the following formula

$$\text{p-value} = \frac{\#\{\mathfrak{D}_m^i \geq \mathfrak{D}_m^0, i = 1, 2, \dots, \text{NREP}\} + 1}{\text{NREP} + 1} \quad (2.3)$$

We investigate the timing for the function `portest()` based on the Monte-Carlo significance test for univariate and multivariate time series of lengths  $n = 100, 500, 1000$ . In the univariate case, we generate AR(1) series and then compute the elapsed AR(1) fitted time based on the contributed R functions, `FitAR()`, `ar()`, `arima()`, `arima0()`, `Arima()`, and `auto.arima()`. In the multivariate series we consider the series dimensions  $k = 2, 3, 4$  generated from the VAR(1) process and the computed time is compared between the fitted VAR(1) model using the functions `VAR()` and `ar.ols()`.

In addition to this, we compute the elapsed time of fitting GARCH models using the functions `garch()` and `garchFit()` as well as FGN models using the function `FitFGN()`.

The times reported in Tables 2.1, 2.2, and 2.3 were achieved on a 2.1 GHz intel(R) Core(TM) i3 CPU running Windows 7.

nslaves	$n$	FitAR()	ar()	arima()	arima0()	Arima()	auto.arima()
1	100	72.93	68.59	82.11	82.37	82.53	82.82
1	500	73.21	68.94	91.16	91.46	91.61	91.11
1	1000	74.96	69.59	104.8	105.11	105.16	210.83
4	100	19.54	17.39	22.27	21.01	22.13	21.83
4	500	18.78	17.87	24.43	22.98	24.40	25.06
4	1000	19.39	17.91	27.51	26.15	27.44	52.15

Table 2.1: CPU time in seconds. The R functions `FitAR()`, `ar()`, `arima()`, `arima0()`, `Arima()`, and `auto.arima()` are used to fit the AR (1) model to univariate time series of lengths  $n = 100, 500, 1000$  and the `portest()` function is applied on the fitted model based on  $10^3$  replications of Monte-Carlo test. `nslaves` denotes the number of CPU's are used.

nslaves	$n$	VAR()			ar.ols()		
		$k = 2$	$k = 3$	$k = 4$	$k = 2$	$k = 3$	$k = 4$
1	100	117.94	127.58	139.29	109.39	112.78	117.16
1	500	329.39	344.34	361.16	319.20	327.06	334.57
1	1000	359.97	378.68	401.02	346.50	355.69	365.16
4	100	31.70	34.34	35.63	25.65	26.66	27.82
4	500	77.19	81.75	87.01	68.76	71.11	72.70
4	1000	85.23	93.54	101.65	74.91	77.43	80.05

Table 2.2: CPU time in seconds. The R functions `VAR()` and `ar.ols()` are used to fit the VAR (1) model to series of lengths  $n = 100, 500, 1000$  with dimensions  $k = 2, 3, 4$ . The `portest()` function is applied on the fitted model based on  $10^3$  replications of Monte-Carlo test. `nslaves` denotes the number of CPU's are used.

### 2.2.2 Monte-Carlo testing for randomness

When the object `obj` has a class `"numeric"`, `"ts"`, `"matrix"`, or `"mts & ts"` provided that `MonteCarlo = TRUE` is selected, the function `portest()` implements the Monte-Carlo techniques for testing the randomness behaviors of the object `obj` as follows,

1. Treat the object `obj` as residuals and calculate the observed value of the portman-teau as described in Step 1 before. The argument `SquaredQ = TRUE` maybe selected if one need to check for nonlinearity structure.

nslaves	$n$	garch()	garchFit()	FitFGN()
1	100	137.27	240.39	77.08
1	500	164.24	346.9	90.14
1	1000	198.08	458.43	121.14
4	100	38.83	62.35	20.39
4	500	45.35	83.53	22.59
4	1000	53.77	119.73	29.98

Table 2.3: CPU time in seconds. Univariate time series of lengths  $n = 100, 500, 1000$  are generated from normal process and R functions `garch()` and `garchFit()` are used to fit the GARCH(1,1) model, whereas the function `FitFGN()` is used to fit FGN model. The `portest()` function is applied on the fitted model based on  $10^3$  replications of Monte-Carlo test. `nslaves` denotes the number of CPU's are used.

2. If `InfiniteVarianceQ = TRUE` then simulate an innovation series with infinite variance from stable distribution. Otherwise, simulate a white noise series from normal distribution with mean zero and estimated covariance obtained from the object `obj`.

Step 2 will be repeated `NREP` times as described before. Each time, the simulated portmanteau test is calculated for the simulated series from step 2 and compared to the observed one. The Monte-Carlo p-values associated with the different lag values are calculated as in Equation 2.3 before.

Note that the argument `lags = seq(5, 30, 5)` is a vector of lag auto-cross correlation coefficients used for test statistic that can be changed to any desirable vector of positive integers. Note also that the argument `order` is not used in Monte-Carlo version of the test statistic and used only for asymptotic distribution approach with object class `"numeric"`, `"ts"`, `"matrix"`, or `"mts & ts"`.

### 2.2.3 Asymptotic distribution significance test

By setting the argument `MonteCarlo = FALSE` in the main function, `portest()`, the portmanteau test statistic based on the asymptotic chi-distribution as we will describe in Section 2.3 will be implemented.

The argument `order` is needed to determine the degrees of freedom for the chi-distribution in such a case, but it is automatically determined if the object `obj` is a fitted time series model with class `"ar"`, `"Arima"`, `"arima0"`, `"varest"`, `"FitAR"`, `"FitFGN"`, `"garch"`, `"fGARCH"`, or `"list"`.

In general `order = p + q` represents the orders of the ARMA /VARMA models.

The default argument `order = 0` is used for the case of random series, garch effects, or fractional gaussian noise. The argument `lags` is described as before while the other arguments are not used in this case.

Note that for both types, Monte-Carlo and asymptotic distribution, when `SquaredQ = TRUE`, then apply the test on the squared values. This checks for Autoregressive Conditional Heteroscedastic, ARCH, effects. When `SquaredQ = FALSE`, then apply the test on the usual residuals.

As an illustrative example, consider again the returns of the DEXCAUS data. We apply the `portest()` function using the asymptotic chi-square method, and the results support the claim that the `returns` series behaves like random series.

```
R> portest(returns, MonteCarlo = FALSE)
```

Lags	Statistic	df	p-value
5	5.726436	4.090909	0.2302146
10	12.002784	7.857143	0.1431200
15	16.798015	11.612903	0.1395903

20	20.847948	15.365854	0.1566299
25	23.805231	19.117647	0.2090410
30	27.248101	22.868852	0.2396022

## 2.3 COMPUTATION PORTMANTEAU TESTS

### 2.3.1 Box and Pierce portmanteau test

In the univariate time series, [Box and Pierce \(1970\)](#) introduced the portmanteau statistic

$$Q_m = n \sum_{\ell=1}^m \hat{r}_\ell^2 \quad (2.4)$$

where  $\hat{r}_\ell = \sum_{t=\ell+1}^n \hat{e}_t \hat{e}_{t-\ell} / \sum_{t=1}^n \hat{e}_t^2$ , and  $\hat{e}_1, \dots, \hat{e}_n$  are the residuals. This test statistic is implemented in the R function `BoxPierce()` and can be used in the multivariate case as well. It is approximately chi-square distribution with  $k^2(m - (p + q))$  degrees of freedom where  $k$  represents the dimension of the time series. The usage of this function is extremely simple:

```
BoxPierce(obj, lags = seq(5, 30, 5), order = 0, SquaredQ = FALSE),
```

where the arguments of this function are as previously described.

Note that the function `portest()` with the arguments `test = "BoxPierce"`, `MonteCarlo = FALSE`, and `order = 0` will the same results as the function `BoxPierce()`. The Monte-Carlo version of this test statistic is implemented in the function `portest()` as an argument `test = "BoxPierce"` provided that `MonteCarlo = TRUE` is selected.

### 2.3.2 Ljung and Box portmanteau test

Ljung and Box (1978) modified the Box and Pierce (1970) test statistic as follows:

$$\hat{Q}_m = n(n+2) \sum_{\ell=1}^m (n-\ell)^{-1} \hat{r}_\ell^2. \quad (2.5)$$

This test statistic is also asymptotically chi-square with degrees of freedom  $k^2(m-p-q)$  and implemented in the R function `LjungBox()`,

```
LjungBox(obj, lags = seq(5, 30, 5), order = 0, SquaredQ = FALSE),
```

where the arguments of this function are previously described.

In R, the function `Box.test()` was built to compute the Box and Pierce (1970) and Ljung and Box (1978) test statistics only in the univariate case where we cannot use more than one single lag value at a time. The functions `BoxPierce()` and `LjungBox()` are more general than `Box.test()` and can be used in the univariate or multivariate time series in conjunction with a vector of different lag values; as well, they can be applied on an output object from a fitted model.

Note that the function `portest()` with the arguments `test="LjungBox"`, `MonteCarlo=FALSE`, and `order=0` will the same results as the function `LjungBox()`. The Monte-Carlo version of this test statistic is implemented in the function `portest()` as an argument `test = "LjungBox"` provided that `MonteCarlo = TRUE` is selected.

### 2.3.3 Hosking portmanteau test

Hosking (1980) generalized the univariate portmanteau test statistics given in Equations 2.4 and 2.5 to the multivariate case. He suggested the modified multivariate portmanteau test statistic

$$\tilde{Q}_m = n^2 \sum_{\ell=1}^m (n - \ell)^{-1} \hat{\mathbf{r}}'_\ell (\hat{\mathbf{R}}_0^{-1} \otimes \hat{\mathbf{R}}_0^{-1}) \hat{\mathbf{r}}_\ell \quad (2.6)$$

where  $\hat{\mathbf{r}}_\ell = \text{vec } \hat{\mathbf{R}}'_\ell$  is a  $1 \times k^2$  row vector with rows of  $\hat{\mathbf{R}}_\ell$  stacked one next to the other, and  $m$  is the lag order. The symbol  $\otimes$  denotes the Kronecker product ([http://en.wikipedia.org/wiki/Kronecker\\_product](http://en.wikipedia.org/wiki/Kronecker_product)),  $\hat{\mathbf{R}}_\ell = \hat{\mathbf{L}}' \hat{\mathbf{\Gamma}}_\ell \hat{\mathbf{L}}$ ,  $\hat{\mathbf{L}} \hat{\mathbf{L}}' = \hat{\mathbf{\Gamma}}_0^{-1}$  where  $\hat{\mathbf{\Gamma}}_\ell = n^{-1} \sum_{t=\ell+1}^n \hat{\mathbf{e}}_t \hat{\mathbf{e}}'_{t-\ell}$  is the lag  $\ell$  residual autocovariance matrix.

The asymptotic distributions of  $\tilde{Q}_m$  is chi-squared with  $k^2(m - p - q)$  degrees of freedom. In **portest** package, this statistic is implemented in the function `Hosking()`:

```
Hosking(obj, lags = seq(5, 30, 5), order = 0, SquaredQ = FALSE),
```

where the arguments of this function is described as before. Note that the function `portest()` with the arguments `test = "Hosking"`, `MonteCarlo = FALSE`, and `order = 0` will gives the same results of the function `Hosking()`. The Monte-Carlo version of this test statistic is implemented in the function `portest()` as an argument `test = "Hosking"` provided that `MonteCarlo = TRUE` is selected.

### 2.3.4 Li and McLeod portmanteau test

[Li and McLeod \(1981\)](#) suggested the multivariate modified portmanteau test statistic

$$\tilde{Q}_m^{(L)} = n \sum_{\ell=1}^m \hat{\mathbf{r}}'_\ell (\hat{\mathbf{R}}_0^{-1} \otimes \hat{\mathbf{R}}_0^{-1}) \hat{\mathbf{r}}_\ell + \frac{k^2 m(m+1)}{2n} \quad (2.7)$$

which is asymptotically distributed as chi-squared with  $k^2(m - p - q)$  degrees of freedom. In **portes** package, the test statistic  $\tilde{Q}_m^{(L)}$  is implemented in the function `LiMcLeod()`,

```
LiMcLeod(obj, lags = seq(5, 30, 5), order = 0, SquaredQ = FALSE),
```

where the arguments of this function is described as before.

Note that the function `portest()` with the arguments `test= "LiMcLeod"`, `MonteCarlo=FALSE`, and `order=0` will the same results as the function `LiMcLeod()`. The Monte-Carlo version of this test statistic is implemented in the function `portest()` as an argument `test="LiMcLeod"` provided that `MonteCarlo = TRUE` is selected.

### 2.3.5 Generalized variance portmanteau test

Peña and Rodriguez (2002) proposed a univariate portmanteau test of goodness-of-fit test based on the  $m$ -th root of the determinant of the Toeplitz residual autocorrelation matrix of order  $m + 1$ ,

$$\hat{\mathcal{R}}_m = \begin{pmatrix} \hat{r}_0 & \hat{r}_1 & \dots & \hat{r}_m \\ \hat{r}_{-1} & \hat{r}_0 & \dots & \hat{r}_{m-1} \\ \vdots & \dots & \ddots & \vdots \\ \hat{r}_{-m} & \hat{r}_{-m+1} & \dots & \hat{r}_0 \end{pmatrix}, \quad (2.8)$$

where  $\hat{r}_0 = 1$  and  $\hat{r}_{-\ell} = \hat{r}_\ell$ , for all  $\ell$ . They approximated the distribution of their proposed test statistic by the gamma distribution and provided simulation experiments to demonstrate the improvement of their statistic in comparison with the one that is given in Equation 2.5.

Lin and McLeod (2006) introduced the Monte-Carlo version of this test as they noted that it is quite often that the generalized variance portmanteau test does not agree with the suggested Gamma approximation. They show that the Monte-Carlo is more powerful than its competitors with the correct size level.

Peña and Rodriguez (2006) suggested to modify the generalized variance test by taking the log of the  $(m + 1)$ -th root of the determinant of  $\hat{\mathcal{R}}_m$  given in Equation 2.8. They proposed two approximations by using the Gamma and Normal distributions to the asymptotic distribution of this test and indicated that the performance of both

approximations for checking the goodness-of-fit in linear models is similar and more powerful for small sample size than the previous one.

Mahdi and McLeod (2011) extended this portmanteau test to the multivariate time series and their simulation experiments illustrated and demonstrated the usefulness of the Monte-Carlo test as well as its improved power performance compared to the previously used multivariate portmanteau diagnostic checks.

Their new multivariate generalized portmanteau test statistic is

$$\mathfrak{D}_m = -3n(2m + 1)^{-1} \log |\hat{\mathfrak{R}}_m| \quad (2.9)$$

where

$$\hat{\mathfrak{R}}_m = \begin{pmatrix} \mathbb{I}_k & \hat{\mathbf{R}}_1 & \dots & \hat{\mathbf{R}}_m \\ \hat{\mathbf{R}}_1' & \mathbb{I}_k & \dots & \hat{\mathbf{R}}_{m-1}' \\ \vdots & \dots & \ddots & \vdots \\ \hat{\mathbf{R}}_m' & \hat{\mathbf{R}}_{m-1}' & \dots & \mathbb{I}_k \end{pmatrix}, \quad (2.10)$$

The asymptotic null distribution is chi-square with  $k^2[1.5m(m+1)(2m+1)^{-1} - p - q]$  degrees of freedom, and it is implemented in the R function `gvtest()`,

```
gvtest(obj, lags = seq(5, 30, 5), order = 0, SquaredQ = FALSE),
```

where the arguments of this function are described as before.

Note that the function `portest()` with the arguments `test="gvtest"`, `MonteCarlo=FALSE`, and `order=0` will the same results as the function `gvtest()`. The Monte-Carlo version of this test statistic is implemented in the function `portest()` as an argument `test="gvtest"` provided that `MonteCarlo = TRUE` is selected.

More simulation results including the simulation R code that we used to compare the asymptotic distribution with the Monte-Carlo significance tests are available on the online vignette documentation of the **portes** R package.

## 2.4 APPLICATIONS

In this section we apply the generalized variance portmanteau test, `gvttest`, on some real data using the approximation asymptotic distribution and the Monte-Carlo significance test. For the Monte-Carlo significance test, we use the argument `nslaves = 4` on a personal computer with 4 CPU's. Users may change this number to meet the number of CPU's that they want. We also include the R code used for calculating the elapsed time in seconds for each Monte-Carlo test.

```
R> nslaves <- 4
```

### 2.4.1 Canadian Lynx trappings data

The first example make use of the univariate series of the logarithms of Canadian lynx trappings from 1821 to 1934. Data is available from the R package `datasets` under the name `lynx`. The AR(2) fitted model was selected based on the BIC criterion. The asymptotic distribution and the Monte-Carlo version of  $\mathfrak{D}_m$  statistic are given in the following R code for lags  $m = 5, 10, 15, 20, 25, 30$ .

```
R> lynxData <- log(lynx)
R> p <- SelectModel(lynxData, ARModel = "AR", Criterion = "BIC",
+   Best = 1)
R> fitlynx <- FitAR::FitAR(lynxData, p, ARModel = "AR")
R> gvttest(fitlynx)
```

Lags	Statistic	df	p-value
5	5.984989	2.090909	0.054687987
10	10.036630	5.857143	0.115222212
15	21.447021	9.612903	0.014964682

```

20 31.810564 13.365854 0.003100578
25 38.761595 17.117647 0.002040281
30 43.936953 20.868852 0.002252062

R> startTime <- proc.time()
R> portest(fitlynx, test = "gvtest", nslaves = nslaves)

      4 slaves are spawned successfully. 0 failed.
Lags Statistic      df      p-value
  5  5.984989  2.090909 0.142857143
 10 10.036630  5.857143 0.172827173
 15 21.447021  9.612903 0.018981019
 20 31.810564 13.365854 0.006993007
 25 38.761595 17.117647 0.002997003
 30 43.936953 20.868852 0.002997003

R> endTime <- proc.time()
R> (endTime - startTime)[3]

elapsed
17.87

```

For lags  $m \geq 15$ , the  $\mathfrak{D}_m$  statistic based on the asymptotic distribution and the Monte-Carlo approach suggests model inadequacy. Fitting a subset autoregressive using the function `FitARp()` with the BIC criterion ([McLeod and Zhang, 2011, 2008b](#)), the Monte-Carlo version of the portmanteau test more decisively suggests model adequacy, whereas the asymptotic chi-square suggests inadequacy at lags 15 and adequacy otherwise.

```
R> SelectModel(log(lynx), lag.max = 15, ARModel = "ARp", Criterion =  
+ "BIC", Best = 1)
```

```
[1] 1 2 4 10 11
```

```
R> fitsubsetAR <- FitAR::FitARp(log(lynx), c(1, 2, 4, 10, 11))
```

```
R> portest(fitsubsetAR, test = "gvtest", MonteCarlo = FALSE)
```

Lags	Statistic	df	p-value
5	2.374225	0.0000000	NA
10	3.598248	0.0000000	NA
15	5.661285	0.6129032	0.008190694
20	8.590962	4.3658537	0.090004731
25	11.462473	8.1176471	0.184353957
30	13.900470	11.8688525	0.297764350

```
R> startTime <- proc.time()
```

```
R> portest(fitsubsetAR, test = "gvtest", nslaves = nslaves)
```

4 slaves are spawned successfully. 0 failed.

Lags	Statistic	df	p-value
5	2.374225	0.0000000	0.5534466
10	3.598248	0.0000000	0.8401598
15	5.661285	0.6129032	0.8821179
20	8.590962	4.3658537	0.8491508
25	11.462473	8.1176471	0.8341658
30	13.900470	11.8688525	0.8461538

```
R> endTime <- proc.time()
```

```
R> (endTime - startTime)[3]
```

elapsed

20.14

### 2.4.2 Monthly simple returns of the CRSP value-weighted index data

In the second example, we consider the monthly simple returns of the CRSP value-weighted index univariate data from January 1926 to December 1997. This data was discussed by [Lin and McLeod \(2008\)](#) and available in our Package with the name CRSP. They implemented the Monte-Carlo and the asymptotic test of [Peña and Rodriguez \(2002\)](#) and [Ljung and Box \(1978\)](#) on the fitted AR(5) model assuming that the innovations have finite and infinite variance.

```
R> data("CRSP")
R> CRSP.AR5 <- stats::arima(CRSP, c(5, 0, 0))
R> lags <- c(10, 20, 30)
```

We may reproduce the output results of [Lin and McLeod \(2008, Tables II, V\)](#) using the function `portest()`.

First we apply the Monte-Carlo test of `gvttest` on the fitted model assuming that the innovations have finite variance.

```
R> startTime <- proc.time()
R> portest(CRSP.AR5, lags = lags, nslaves = nslaves)
```

```
4 slaves are spawned successfully. 0 failed.
```

Lags	Statistic	df	p-value
10	4.838733	2.857143	0.062937063
20	18.807192	10.365854	0.013986014
30	34.519726	17.868852	0.004995005

```
R> endTime <- proc.time()
R> (endTime - startTime)[3]
```

```
elapsed
58.39
```

Then we follow the suggestion of [Lin and McLeod \(2008\)](#), by applying the quantile estimation method of [McCulloch \(1986\)](#) on the residuals of the fitted mode (see Section 2.5.3),

```
R> resCRSP.AR5 <- CRSP.AR5$resid
R> fitstable(resCRSP.AR5)

      alpha      beta      scale      location
1.650244 -0.3204106 0.02779931 0.005089577
```

The estimated parameter  $\alpha = 1.65$  suggests that the infinite variance hypothesis is plausible for this data. Now we apply the Monte-Carlo version of `gvttest` test assuming that innovations have infinite variance,

```
R> startTime <- proc.time()
R> portest(CRSP.AR5, lags = lags, nslaves = nslaves,
+ InfiniteVarianceQ = TRUE)
```

```
4 slaves are spawned successfully. 0 failed.
```

Lags	Statistic	df	p-value
10	4.838733	2.857143	0.06093906
20	18.807192	10.365854	0.03596404
30	34.519726	17.868852	0.01898102

```
R> endTime <- proc.time()
R> (endTime - startTime)[3]
```

```
elapsed
57.72
```

As noticed by [Lin and McLeod \(2008\)](#), the Monte-Carlo version of the [Peña and Rodriguez \(2002\)](#) test based on the assumption that innovations have finite variance is not different from that assumption of innovations with infinite variance.

### 2.4.3 Monthly log stock returns of intel corporation data

Our next example is given as an illustrative application of testing Autoregressive Conditional Heteroscedastic, ARCH, effects. We consider the monthly log stock returns of Intel Corporation data which is available in our package with the names `monthintel`. First we apply the asymptotic distribution and the Monte-Carlo version of the statistic `gvttest` directly on the returns, which suggests no significant serial correlations.

```
R> data("monthintel")
R> monthintel <- as.ts(monthintel)
R> startTime <- proc.time()
R> portest(monthintel, lags = seq(10, 40, 10), MonteCarlo = FALSE,
+         SquaredQ = FALSE)
```

Lags	Statistic	df	p-value
10	9.85822	7.857143	0.2632960
20	20.38081	15.365854	0.1738108
30	28.17267	22.868852	0.2039194
40	34.86373	30.370370	0.2627128

```
R> portest(monthintel, lags = seq(10, 40, 10), MonteCarlo = TRUE,  
+         nslaves = nslaves, SquaredQ = FALSE)
```

```
4 slaves are spawned successfully. 0 failed.
```

Lags	Statistic	df	p-value
10	9.85822	7.857143	0.2667333
20	20.38081	15.365854	0.1728272
30	28.17267	22.868852	0.1848152
40	34.86373	30.370370	0.2217782

```
R> endTime <- proc.time()
```

```
R> (endTime - startTime)[3]
```

```
elapsed
```

```
26.3
```

After that we apply both methods of `gvttest` on the squared returns. The result suggests that the monthly returns are not serially independent and the return series may suffers of ARCH effects.

```
R> startTime <- proc.time()
```

```
R> portest(monthintel, lags = seq(10, 40, 10), MonteCarlo = FALSE,
```

```
+         SquaredQ = TRUE)
```

Lags	Statistic	df	p-value
10	34.67590	7.857143	2.718240e-05
20	52.04751	15.365854	7.097281e-06
30	64.65689	22.868852	7.331587e-06
40	72.26830	30.370370	2.856747e-05

```
R> portest(monthintel, lags = seq(10, 40, 10), MonteCarlo = TRUE,
+         nslaves = nslaves, SquaredQ = TRUE)
```

```
4 slaves are spawned successfully. 0 failed.
```

Lags	Statistic	df	p-value
10	34.67590	7.857143	0.007992008
20	52.04751	15.365854	0.005994006
30	64.65689	22.868852	0.004995005
40	72.26830	30.370370	0.003996004

```
R> endTime <- proc.time()
R> (endTime - startTime)[3]
```

```
elapsed
26.3
```

#### 2.4.4 U.S. inflation data

In this example, we fit an ARMA/GARCH model to the U.S. inflation ([Bollerslev, 1986](#)). We used the GNP deflator for 1947-01-01 to 2010-04-01. Data is available from our package with the name `GNPDEF`. There are  $n = 254$  observations which are denoted by  $z_t, t = 1, \dots, n$ . The inflation rate may be estimated by the logarithmic difference,  $r_t = \log(z_t) - \log(z_{t-1})$ .

The function `garchFit()` from the R package **fGarch** is used to fit the ARMA(4,0)-GARCH(1,1) model and the Monte-Carlo `gvtest` test does not show any GARCH effects in the fitted residuals.

```
R> z <- ts(GNPDEF[, 2], start = 1947, freq = 4)
R> r <- 100 * diff(log(z))
```

```
R> GarchFit <- garchFit(formula = ~arma(4, 0) + garch(1, 1), data = r,  
+   trace = FALSE)
```

```
R> startTime <- proc.time()
```

```
R> portest(GarchFit, nslaves = nslaves, SquaredQ = FALSE)
```

4 slaves are spawned successfully. 0 failed.

Lags	Statistic	df	p-value
5	4.32453	4.090909	0.05294705
10	8.18047	7.857143	0.10889111
15	11.57489	11.612903	0.16183816
20	14.09171	15.365854	0.25374625
25	16.28966	19.117647	0.34165834
30	18.80426	22.868852	0.41658342

```
R> portest(GarchFit, nslaves = nslaves, SquaredQ = TRUE)
```

4 slaves are spawned successfully. 0 failed.

Lags	Statistic	df	p-value
5	2.282153	4.090909	0.5464535
10	8.514656	7.857143	0.2397602
15	11.523989	11.612903	0.3046953
20	14.299126	15.365854	0.3436563
25	16.510697	19.117647	0.4255744
30	19.908227	22.868852	0.4265734

```
R> endTime <- proc.time()
```

```
R> (endTime - startTime)[3]
```

elapsed

317.59

### 2.4.5 Nile annual minima data

For testing Fractional Gaussian Noise, FGN, we consider the NileMin data. This data is given in the **FGN** package with the name NileMin (McLeod et al., 2011). We use the function FitFGN to fit FGN to this data and then apply the Monte-Carlo version of gvttest test to the fitted model,

```
R> data("NileMin")
R> NILE.FGN <- FitFGN(NileMin)
R> startTime <- proc.time()
R> portest(NILE.FGN, nslaves = nslaves)
```

```
4 slaves are spawned successfully. 0 failed.
```

Lags	Statistic	df	p-value
5	0.8061659	4.090909	0.9030969
10	2.8027767	7.857143	0.9010989
15	5.6961553	11.612903	0.8831169
20	7.9541034	15.365854	0.9070929
25	11.7901068	19.117647	0.8571429
30	16.4277997	22.868852	0.7682318

```
R> endTime <- proc.time()
R> (endTime - startTime)[3]
```

```
elapsed
```

```
111.57
```

Results suggest no inadequacy in the fitted model.

## 2.4.6 Canadian labor market data

This example uses the multivariate macro economic data set for Canada. Data starts at the first quarter of 1980 and ends at the fourth quarter of 2000 and available from the software **JMulTi** (Breitung et al., 2004) and the R Package **vars** (Pfaff, 2010).

```
R> data("Canada")
```

According to Breitung et al. (2004); Pfaff (2008a), the AIC and FPE determine the optimal lag number of an unrestricted VAR model with a maximal lag length of eight to be  $p = 3$ , whereas the HQ criterion indicates  $p = 2$  and the SC criterion indicates an optimal lag length of  $p = 1$ . After that, they estimated for all three lag orders a VAR including a drift constant and a deterministic trend and conducted diagnostic tests using the multivariate version of the BoxPierce portmanteau test statistic. See results in (Pfaff, 2008a, Table 3).

```
R> p1ct <- vars::VAR(Canada, p = 1, type = "both")
R> BoxPierce(p1ct, lags = seq(4, 16, 4), order = 1)
```

Lags	Statistic	df	p-value
4	96.77914	48	3.873632e-05
8	140.53171	112	3.525332e-02
12	182.80955	176	3.468634e-01
16	233.49989	240	6.059988e-01

As BoxPierce statistic suggests an adequacy in the fitted VAR(1) model at lag  $m = 12$  and 16, it shows inadequacy at lower lags. The Monte-Carlo version of BoxPierce portmanteau test confirmed these results.

```
R> startTime <- proc.time()
R> portest(p1ct, test = "BoxPierce", lags = seq(4, 16, 4), nslaves = nslaves)
```

4 slaves are spawned successfully. 0 failed.

Lags	Statistic	df	p-value
4	96.77914	48	0.004995005
8	140.53171	112	0.070929071
12	182.80955	176	0.318681319
16	233.49989	240	0.393606394

```
R> endTime <- proc.time()
```

```
R> (endTime - startTime)[3]
```

elapsed

16.86

Next, we fit a VAR(3) with a constant and a trend using the function VAR and implement the goodness of fit tests based on the Monte-Carlo version of `gvtest` statistic,

```
R> p3ct <- vars::VAR(Canada, p = 3, type = "both")
```

```
R> startTime <- proc.time()
```

```
R> portest(p3ct, lags = seq(4, 16, 4), nslaves = nslaves)
```

4 slaves are spawned successfully. 0 failed.

Lags	Statistic	df	p-value
4	19.70053	5.333333	0.8281718
8	57.65003	53.647059	0.9640360
12	109.50000	101.760000	0.9870130
16	175.40899	149.818182	0.9960040

```
R> endTime <- proc.time()
```

```
R> (endTime - startTime)[3]
```

elapsed

26.69

The Monte-Carlo version of `gvtest` statistics suggests model adequacy.

### 2.4.7 Prey data

The final example make use the stationary prey *Didinium* time series in the `veilleux` data. Data is available from the **TSA** R package with the name `prey.eq` ([Cryer and Chan, 2010](#)). We apply the Monte-Carlo version of `gvtest` on the fitted threshold AR model taken from the package **TSA** ([Cryer and Chan, 2010](#)). In this example we coded the two functions `FitModel()` and `SimModel()` needed for the argument `func` as follows,

```
R> library("TSA")
R> FitModel <- function(data) {
+   fit <- TSA::tar(y = log(data), p1 = 4, p2 = 4, d = 3, a = 0.1,
+     b = 0.9, print = FALSE)
+   res <- ts(fit$std.res)
+   parSpec <- list(res = res, fit = fit)
+   parSpec
+ }
R> SimModel <- function(parSpec) {
+   fit <- parSpec$fit
+   exp(tar.sim(fit)$y)
+ }
R> data(pre.eq)
```

```
R> portest(FitModel(preym.eq), nslaves = nslaves, func = list(SimModel,
+   FitModel), pkg = "TSA")
```

```
4 slaves are spawned successfully. 0 failed.
```

Lags	Statistic	df	p-value
5	2.588669	4.090909	0.4355644
10	4.125518	7.857143	0.7112887
15	5.792759	11.612903	0.8331668
20	7.421699	15.365854	0.9000999
25	9.195862	19.117647	0.9410589
30	10.633459	22.868852	0.9730270

```
R> detach(package:TSA)
```

The results suggest that the model is an adequate model. Note that we can modify this simple code replacing the fitted TAR model and `tar.sim()` function by any other fitted model and simulation code.

## 2.5 SOME USEFUL FUNCTIONS

### 2.5.1 Stationary and invertibility of VARMA models

The conditions for stationary and invertibility of the VARMA  $(p, q)$  process are the same as in the pure VAR  $(p)$  and pure VMA  $(q)$  cases, respectively (Reinsel, 1997).

The multivariate VAR  $(p)$  model of  $k$ -dimensional time series  $\mathbf{Z}_t = (Z_{1,t}, \dots, Z_{k,t})'$  with mean vector  $\boldsymbol{\mu}$  and no deterministic equation is given by

$$\Phi(\mathbf{B})(\mathbf{Z}_t - \boldsymbol{\mu}) = \mathbf{e}_t, \quad (2.11)$$

where  $\Phi(\mathbf{B}) = \mathbb{I}_k - \Phi_1\mathbf{B} - \dots - \Phi_p\mathbf{B}^p$ , can always be expressed in the state-space  $kp$ -dimensional VAR(1) model in terms of  $\mathbf{Z}_t^\star = (\mathbf{Z}'_t, \dots, \mathbf{Z}'_{t-p+1})'$  as  $\mathbf{Z}_t^\star = \boldsymbol{\mu} + \Phi^\star\mathbf{Z}_{t-1}^\star + \mathbf{e}_t^\star$ , with  $\mathbf{e}_t^\star = (\mathbf{e}'_t, \mathbf{0}', \dots, \mathbf{0}')$  and  $\Phi^\star$  equal to the  $kp \times kp$  companion matrix associated with the VAR( $p$ ) operator  $\Phi(\mathbf{B})$ , that is,

$$\Phi^\star = \begin{pmatrix} \Phi_1 & \Phi_2 & \dots & \dots & \Phi_p \\ \mathbb{I}_k & \mathbf{0} & \dots & \dots & \mathbf{0} \\ \mathbf{0} & \mathbb{I}_k & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbb{I}_k & \mathbf{0} \end{pmatrix}_{kp \times kp} . \quad (2.12)$$

Similar to this, the  $k$ -dimensional VMA( $q$ ) process can be represented in the state-space  $kq$ -dimensional VMA(1) model as  $\mathbf{Z}_t^\star = \boldsymbol{\mu} + \mathbf{e}_t^\star - \Theta^\star\mathbf{e}_{t-1}^\star$ , where  $\Theta^\star$  is defined by Equation 2.12 after replacing  $\Phi_i$  by  $\Theta_j$  and  $i = 1, \dots, p$  by  $j = 1, \dots, q$ ,  $\mathbf{e}_t^\star$  by  $\mathbf{Z}_t^\star$  and  $\mathbf{Z}_t^\star$  by  $\mathbf{e}_t^\star$ . In this light, the stationary condition that all roots of  $\det[\Phi(\mathbf{B})] = 0$  are greater than one in absolute value in the VAR( $p$ ) model is equivalent to the condition in the state-space VAR(1) representation that all eigenvalues of the  $kp \times kp$  companion matrix  $\Phi^\star$  be less than one in absolute value, and the invertibility condition that all roots of  $\det[\Theta(\mathbf{B})] = 0$  are greater than one in absolute value in the VMA( $q$ ) model is equivalent to the condition in the state-space VMA(1) representation that all eigenvalues of the  $kq \times kq$  companion matrix  $\Theta^\star$  be less than one in absolute value. The function `InvertQ()` in our package uses this technique and checks the validity of the stationary and invertibility assumptions in the process. The syntax of `InvertQ` is:

`InvertQ(coef)`.

It takes a numeric, matrix, or array of coefficients of AR, MA, VAR, or VMA process as an argument and returns a warning message, "check stationary/invertibility

condition !", only if the process is not stationary or not invertible.

For checking stationarity of a process with dimension  $k = 2$  from VAR(2) with coefficients  $\Phi_1 = \begin{pmatrix} 0.5 & 0.1 \\ 0.4 & 0.5 \end{pmatrix}$  and  $\Phi_2 = \begin{pmatrix} 0.0 & 0.0 \\ 0.3 & 0.0 \end{pmatrix}$ , type:

```
R> phi <- array(c(0.5, 0.4, 0.1, 0.5, 0, 0.3, 0, 0), dim = c(2,
+ 2, 2))
R> InvertQ(phi)
```

and for checking invertibility of a process with dimension  $k = 3$  from VMA(1) with coefficient  $\Theta = \begin{pmatrix} 0.5 & 0.0 & 0.0 \\ 0.1 & 0.1 & 0.3 \\ 0.0 & 0.2 & 0.3 \end{pmatrix}$ , type:

```
R> theta <- array(c(0.5, 0.1, 0, 0, 0.1, 0.2, 0, 0.3, 0.3), dim = c(3,
+ 3, 1))
R> InvertQ(theta)
```

while for checking stationarity and invertibility of process from VARMA(1,1) with coefficients  $\Phi = \begin{pmatrix} 0.9 & 0.4 \\ 0.3 & 0.1 \end{pmatrix}$  and  $\Theta = \begin{pmatrix} 0.5 & 0.7 \\ 0.1 & 0.5 \end{pmatrix}$ , type:

```
R> phi <- array(c(0.9,0.3,0.4,0.1),dim=c(2,2,1))
R> InvertQ(phi)
```

Warning message:

```
In InvertQ(phi) : check stationary/invertibility condition !
```

```
R> theta <- array(c(0.5, 0.1, 0.7, 0.5), dim = c(2, 2, 1))
R> InvertQ(phi)
```

The warning message in the output means that the stationary condition in the process is not valid.

### 2.5.2 Simulation from nonseasonal ARIMA or VARIMA models

We may use the function `varima.sim()` in the proposed package to simulate non-seasonal data from VARIMA, ARIMA process described in Equations 2.1 and 2.2 respectively. The simulated data may have a deterministic constant drift and time trend term with non-zero mean. To simulate time series from models with infinite variance innovations, the argument `StableParameters` with stable parameters, `ALPHA`, `BETA`, `GAMMA`, and `DELTA` must be entered in the function `varima.sim()`, provided that the arguments `innov = NULL` and `innov.dist = "stable"` are selected. The parameters `ALPHA` represent the index parameter of the stable distribution and must be entered as a numeric or vector with values in the range  $(0, 2]$ . The skewness parameters `BETA` values must be in the range  $[-1, 1]$ . The scale parameters, and the location parameters, `GAMMA`, `DELTA`, can have `NULL` values or real values in the range  $(-\infty, \infty)$ .

The `varima.sim()` function is:

```
varima.sim(phi=NULL,theta=NULL,d=NA,sigma,n,constant=NA,trend=NA,  
           demean=NA,innov=NULL,innov.dist=c("normal","t","stable"),  
           df=1,StableParameters=NA,Trunc.Series=NA),
```

where `phi` and `theta` are the autoregressive and moving average parameters respectively and should have class `"array"` or `"NULL"` in case of  $k > 1$ . The arguments `constant` and `trend` represent coefficients of the deterministic equation and the argument `demean` stands for the mean of the series. For  $k = 1$ , these parameters can be entered with class `"numeric"`, `"array"`, or `"NULL"`. Note that also for  $k = 1$ , the class of the argument `sigma` is `"numeric"` or `"matrix"`, whereas for  $k > 1$ , the argument `sigma` must be entered as a matrix with class `"matrix"`.  $\mathbf{d} = (d_1, \dots, d_k)$ ,  $d_i \geq 0$

is the differencing order. The argument `innov` maybe used to include a univariate or multivariate innovations series, otherwise the innovations series maybe generated from a distribution selected from the argument `innov.dist`. More information about this function can be found on the **portes** manual documentation.

For given values of the argument `StableParameters`, the function `varima.sim()` calls the function `rStable()` to generate innovations from stable distributions,  $S(\alpha, \beta, \gamma, \delta)$ . The syntax of `rStable()` in our package is defined as follows:

```
rStable(n, ALPHA, BETA, GAMMA = NULL, DELTA = NULL).
```

The arguments of this functions are defined as before. Our function `rStable()` may be considered an extension to the function `rStable()` in the R **fBasics** package ([Wuertz and core team members, 2010](#)) as it can be used for generating univariate or multivariate data from independent stable distributions.

The function `varima.sim()` checks the conditions of stationary or invertibility or both in the simulated process by calling the function `InvertQ()`, then it determines recursively the impulse response coefficients,  $\psi_l$  or  $\Psi_l$ ,  $l = 1, 2, \dots$ , by solving the equation  $\psi(B) = \phi^{-1}(B)\theta(B)$  or  $\Psi(B) = \Phi^{-1}(B)\Theta(B)$  ([Reinsel, 1997](#)). Finally it represents this process in terms of an infinite MA or VMA filter, as follows

$$Z_t = \mu + e_t + \psi_1 e_{t-1} + \psi_2 e_{t-2} + \dots \quad (2.13)$$

$$\mathbf{Z}_t = \boldsymbol{\mu} + \mathbf{e}_t + \boldsymbol{\Psi}_1 \mathbf{e}_{t-1} + \boldsymbol{\Psi}_2 \mathbf{e}_{t-2} + \dots \quad (2.14)$$

The impulse response coefficients  $\psi_l$  or  $\Psi_l$  are calculated from the function `ImpulseVMA()` in our package at sufficiently large number. That is, we set a large value to the argument `Trunc.Series` in the functions `ImpulseVMA()` or `varima.sim()` at which the models in Equation 2.13 or Equation 2.14 need to be truncated, so that the

weight of the impulse response coefficient at this truncated value is negligible. If `Trunc.Series` is NA is used in the function `varima.sim()` then the infinite MA or VMA representation will be truncated at the minimum value of 100 or least integer value of  $n/3$ , where  $n$  is the length of the simulated series. If `Trunc.Series` is NA is used in the function `ImpulseVMA()` then the truncation value will be  $p + q$ , where  $p$  and  $q$  represent the order of autoregressive and moving average respectively.

### 2.5.2.1 Simulation example 1

To generate a univariate time series with length 100 from AR(2,1) with drift equation  $2 + 0.01t$ , mean  $\mu = 0$ , ARMA coefficients  $\phi_1 = 0.7, \phi_2 = 0.2, \theta = -0.5$  and t-distribution innovations, type:

```
R> Trunc.Series <- 30
R> n <- 100
R> sigma <- 1
R> phi <- array(c(0.7, 0.2), dim = c(1, 1, 2))
R> theta <- -0.5
R> constant <- 2
R> trend <- 0.01
R> demean <- 0
R> d <- 0
R> z1 <- varima.sim(phi, theta, d, sigma, n, constant, trend, demean,
+   innov.dist = "t", Trunc.Series = Trunc.Series)
```

In this example the infinite MA series is truncated at the value 30, however users can use any desirable truncation value.

### 2.5.2.2 Simulation example 2

In this example we simulate a bivariate time series of length 200 from a VARMA (1,1) model with mean  $\mu = (2, 5)'$ , drift equation  $\mathbf{a} + \mathbf{b}t = (1, 4)' + (0, 0.04)'t$ , and coefficient matrices  $\Phi = \begin{pmatrix} 0.5 & 0.1 \\ 0.4 & 0.5 \end{pmatrix}$ , and  $\Theta = \begin{pmatrix} 0.5 & 0.6 \\ -0.7 & 0.3 \end{pmatrix}$ , where  $\mathbf{a}_t$  are generated from multivariate normal distribution with mean vector zero and covariance matrix  $\Gamma_0 = \begin{pmatrix} 1.00 & 0.71 \\ 0.71 & 1.00 \end{pmatrix}$ . The infinite VMA series is truncated at the least integer value of  $200/3 = 67$ .

```
R> set.seed(123)
R> n <- 200
R> phi <- array(c(0.5, 0.4, 0.1, 0.5), dim = c(2, 2, 1))
R> theta <- array(c(0.5, -0.7, 0.6, 0.3), dim = c(2, 2, 1))
R> sigma <- matrix(c(1, 0.71, 0.71, 1), 2, 2)
R> constant <- c(1, 4)
R> trend <- c(0, 0.04)
R> demean <- c(2, 5)
R> d <- NA
R> z2 <- varima.sim(phi, theta, d, sigma, n, constant, trend, demean)
```

### 2.5.2.3 Simulation example 3

Finally, to simulate an VAR (2) process of length 600 with coefficient matrices  $\Phi_1 = \begin{pmatrix} 0.8 & 0 \\ 0.0 & -2 \end{pmatrix}$ ,  $\Phi_2 = \begin{pmatrix} -0.5 & 0.0 \\ 0.0 & 0.0 \end{pmatrix}$ , covariance matrix  $\Gamma_0 = \begin{pmatrix} 1.0 & 0.5 \\ 0.5 & 1.0 \end{pmatrix}$ , and error term from a stable distribution with  $\alpha = (1.3, 1.6)'$ ,  $\beta = (0, 0.2)'$ ,  $\gamma = (1, 1)'$ ,  $\delta = (0, 0.2)'$ , type:

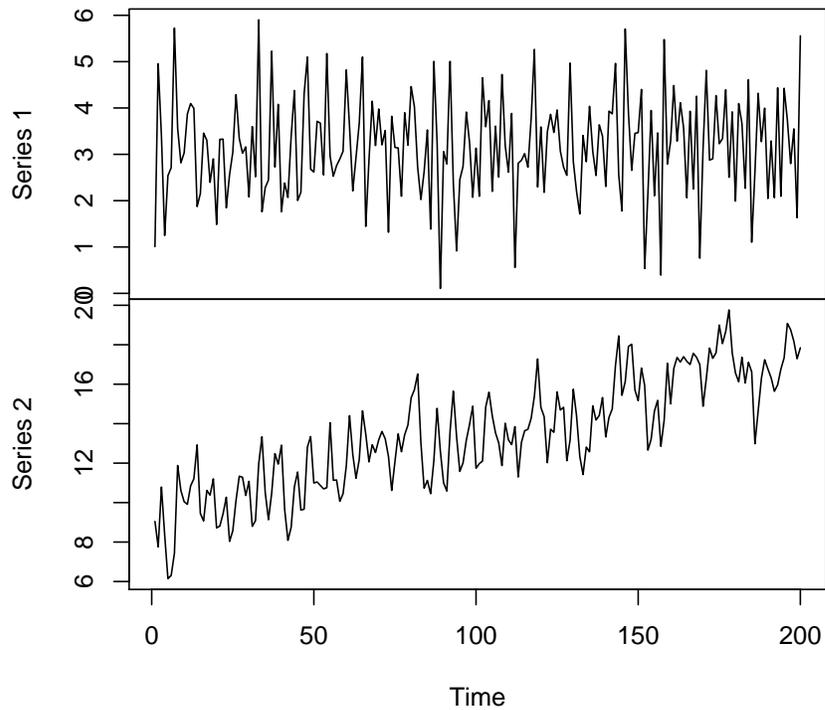


Figure 2.1: A simulated time series of example [2.5.2.2](#).

```
R> set.seed(1234)
R> k <- 2
R> n <- 600
R> phi <- array(c(0.8, 0, -2, 0), dim = c(k, k, 1))
R> theta <- array(c(-0.5, 0, 0, 0), dim = c(k, k, 1))
R> sigma <- matrix(c(1, 0.5, 0.5, 1), k, k)
R> ALPHA <- c(1.3, 1.6)
R> BETA <- c(0, 0.2)
R> GAMMA <- c(1, 1)
```

```
R> DELTA <- c(0, 0.2)
R> StableParameters <- c(ALPHA, BETA, GAMMA, DELTA)
R> d <- NA
R> z3 <- varima.sim(phi, theta, d, sigma, n, innov.dist = "stable",
+   StableParameters = StableParameters)
```

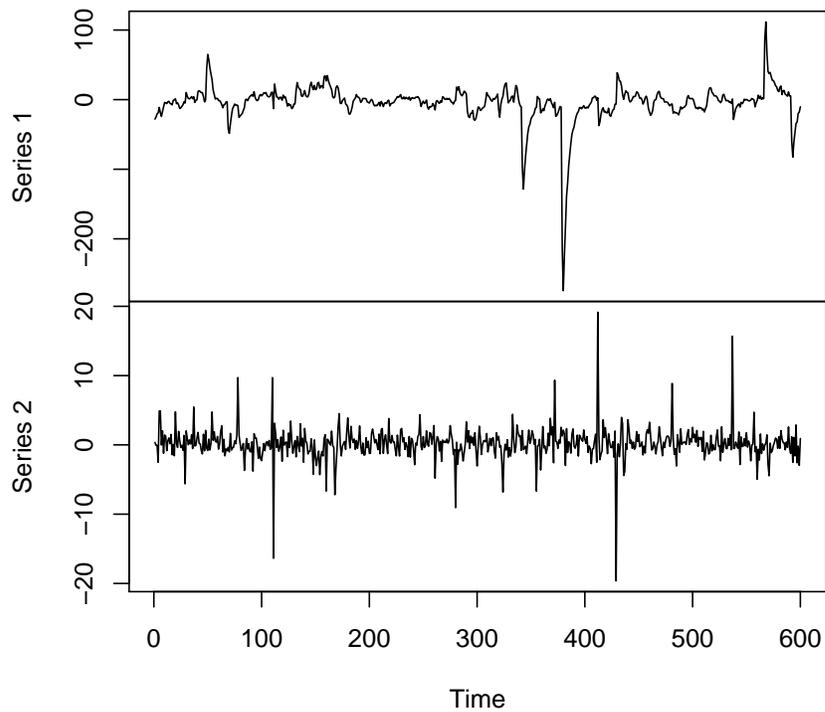


Figure 2.2: A simulated time series of example [2.5.2.3](#).

In this example the series is truncated at default value:  $\text{Trunc.Series} = \min(100, 600/3) = 100$

### 2.5.3 Fit parameters to stable distribution

The quantile estimation method of [McCulloch \(1986\)](#) is implemented in the R function `fitstable()`. This method is highly reliable, fast and reasonably efficient especially bearing in mind that in most applications there is a lot of data. The function `fitstable()` is:

```
fitstable(x),
```

where `x` represents the data frame. This function is used under the assumption of that the variables of the vector `x` are independents. The output of this function is a  $k$  rows represents the number of variables in the vector `x`, and 4 columns with named components `alpha`, `beta`, `scale`, and `location` associated to the stable parameters, `ALPHA`, `BETA`, `GAMMA`, and `DELTA` respectively.

In the following example we use the function `fitstable()` to estimate the stable parameters for a simulated data from ARMA(2,1) model with errors from stable distribution.

```
R> set.seed(54368)
R> n <- 1000
R> phi <- c(1.3, -0.35)
R> theta <- 0.1
R> sigma <- 1
R> ALPHA <- 1.5
R> BETA <- 0
R> GAMMA <- 1
R> DELTA <- 0
R> StableParameters <- c(ALPHA, BETA, GAMMA, DELTA)
```

```

R> d <- NA
R> x <- varima.sim(phi, theta, d, sigma, n, innov.dist = "stable",
+   StableParameters = StableParameters)
R> fitstable(x)

      alpha      beta    scale location
1.436127 -0.2743106 6.330361 5.219509

```

As another example we use the function `rStable()` to simulate a bivariate independent data of size 100 from a stable distribution with heavy tail then we use the function `fitstable()` to estimate the stable parameters as follows

```

R> set.seed(54368)
R> ALPHA <- c(1.3, 1.6)
R> BETA <- c(0, 0.2)
R> GAMMA <- c(1, 1)
R> DELTA <- c(0, 0.2)
R> x <- rStable(100, ALPHA, BETA, GAMMA, DELTA)
R> fitstable(x)

      alpha      beta    scale  location
1.550165 -0.2760850 0.9314066 0.08272251
1.705539  1.0000000 0.9158591 0.57164890

```

## 2.6 CONCLUDING REMARKS

Our package **portes** is available from the Comprehensive R Archive Network, CRAN, at <http://CRAN.R-project.org/package=portes>. It implements the Monte-Carlo test in a very convenient way, specially if we are running R in batch mode using

the **snow** package with cluster computers. We believe that many R users may find that the **portes** package is convenient for simulating time series from nonseasonal ARIMA  $(p, d, q)$  or VARIMA  $(p, \mathbf{d}, q)$  models with or without deterministic equation where innovations have finite or infinite variances.

## REFERENCES

- Adler, R., Feldman, R., and Gallagher, C. (1998). *Analysing stable time series. In a practical guide to heavy tails: Statistical techniques and applications* (eds R. J. Adler, R. E. Feldman and M. S. Taqqu). Number 133–158. Boston: Birkhäuser.
- Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics* 31(3), 307–327.
- Box, G. and Pierce, D. (1970). Distribution of residual autocorrelation in autoregressive-integrated moving average time series models. *Journal of American Statistical Association* 65(332), 1509–1526.
- Box, G., Jenkins, G., and Reinsel, G. C. (2008). *Time series analysis: Forecasting and control* (4th ed.). New York: Wiley.
- Breitung, J., Brüggemann, R., and Lütkepohl, H. (2004). *Structural vector autoregressive modeling and impulse responses. In Lütkepohl, H. and Krätzig, M. Applied time series econometrics, pp. 159-196.* Cambridge University Press: Cambridge.
- Cryer, J., and Chan, Kung-Sik. (2010). *TSA: Time Series Analysis*. URL <http://CRAN.R-project.org/package=TSA>.
- Davison, A. C. and Hinkley, D. V. (1997). *Bootstrap methods and their application*. Cambridge University Press: Cambridge.
- Dufour, J.-M. (2006). Monte Carlo tests with nuisance parameters: A general approach to finite-sample inference and non-standard asymptotics. *Journal of Econometrics* 133(2), 443–477.
- Hannan, E. J. (1969). The estimation of mixed moving average autoregressive systems. *Biometrika* 56(3), 579–593.
- Hosking, J. R. M. (1980). The multivariate portmanteau statistic. *Journal of American Statistical Association* 75(371), 602–607.
- Leisch, F. (2002). *Dynamic generation of statistical reports using literate data analysis. In Wolfgang Härdle and Bernd Rönz. COMPSTAT 2002 – Proceedings in Computational Statistics, pp. 575–580.* Physica-Verlag: Heidelberg.
- Leisch, F. (2003). *Sweave and beyond: Computations on text documents. In Hornik, Kurt and Leisch, Friedrich and Zeileis, Achim. Proceedings*

- of the 3rd International Workshop on Distributed Statistical Computing.. <http://www.ci.tuwien.ac.at/Conferences/DSC-2003/Proceedings>. Vienna, Austria.
- Li, W. K. (2004). *Diagnostic checks in time series*. New York: Chapman and Hall/CRC.
- Li, W. K. and McLeod, A. I. (1981). Distribution of the residual autocorrelation in multivariate arma time series models. *Journal of the Royal Statistical Society, Series B* 43(2), 231–239.
- Lin, J.-W. and McLeod, A. I. (2006). Improved Peña-Rodríguez portmanteau test. *Computational Statistics and Data Analysis* 51(3), 1731–1738.
- Lin, J.-W. and McLeod, A. I. (2008). Portmanteau tests for ARMA models with infinite variance. *Journal of Time Series Analysis* 29(3), 600–617.
- Ljung, G. M. and Box, G. E. P. (1978). On a measure of lack of fit in time series models. *Biometrika* 65, 297–303.
- Lütkepohl, H. (2005). *New Introduction to multiple time series analysis*. New York: Springer-Verlag.
- Mahdi, E. and McLeod, A. I. (2011, accepted). *Improved multivariate portmanteau test*. *The Journal of Time Series Analysis JTSA*(3192).
- McCulloch, J. H. (1986). Simple consistent estimator of stable distribution parameters. *Commun. Statist.-Simula* 4(15), 1109–1136.
- McLeod, A. I. and Li, W. K. (1983). Diagnostic checking ARMA time series models using squared-residual autocorrelations. *The Journal of Time Series Analysis* 4, 269–273.
- McLeod, A. I., Yu, H., and Krougly, Z. (2011). *FGN: Fractional Gaussian noise, estimation and simulaton*. URL <http://CRAN.R-project.org/package=FGN>.
- McLeod, A. I., Yu, H., and Krougly, Z. (2007b). Algorithms for linear time series analysis: With R package. *Journal of Statistical Software* 23(5), 1–26. <http://www.jstatsoft.org/v23/i05>.
- McLeod, A. I. and Zhang, Y.(2011). *FitAR: Subset AR model fitting*. <http://CRAN.R-project.org/package=FitAR>.
- McLeod, A. I. and Zhang, Y. (2008b). Improved subset autoregression: With R package. *Journal of Statistical Software* 28(2), 1–28. <http://www.jstatsoft.org/v28/i02>.
- Peña, D. and Rodríguez, J. (2002). A powerful portmanteau test of lack of test for time series. *Journal of American Statistical Association* 97(458), 601–610.

- Peña, D. and Rodríguez, J. (2006). The log of the determinant of the autocorrelation matrix for testing goodness of fit in time series. *Journal of Statistical Planning and Inference* 8(136), 2706–2718.
- Pfaff, B. (2006). *Analysis of integrated and cointegrated time series with R*. New York: Springer.
- Pfaff, B. (2008a). Var, svar and svec models: Implementation within R package vars. *Journal of Statistical Software* 27(4), 1–32. <http://www.jstatsoft.org/v27/i04>
- Pfaff, B. (2010). *vars: VAR modelling*. <http://CRAN.R-project.org/package=vars>.
- R Development Core Team (2010). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. <http://www.R-project.org>.
- Reinsel, G. C. (1997). *Elements of multivariate time series analysis* (2nd ed.). Springer-Verlag: New York.
- Schmidberger, M., Morgan, M., Eddelbuettel, D., Yu, H., Tierney, L., and Mansmann, U. (2009). State of the art in parallel computing with R. *Journal of Statistical Software* 31(1), 1–27.
- Tierney L., Rossini, A. J., Li, N., and Sevcikova, H. (2011). *snow: Simple Network of Workstations*. <http://CRAN.R-project.org/package=snow>
- Trapletti, A. (2011). *tseries: Time series analysis and computational finance*. <http://CRAN.R-project.org/package=tseries>.
- Wuertz, D. Rmetrics core team members (2010). *fBasics: Rmetrics - Markets and basic statistics*. <http://CRAN.R-project.org/package=fBasics>.
- Wuertz, D. and Rmetrics core team members (2011). *fGarch: Rmetrics - Autoregressive conditional heteroskedastic modelling*. <http://CRAN.R-project.org/package=fGarch>.
- Yu, H. (2002). *Rmpi: Parallel statistical computing in R*. *R News* 2(2), 10–14. <http://CRAN.R-project.org/doc/Rnews/>.
- Zivot, E. and Wang, J. (2006). *Modeling financial time series with S-PLUS* (2nd ed.). Springer Science+Business Media, Inc: New York.

## Chapter 3

# POINCARÉ PLOTS AND REGRESSION WITH HIDDEN CORRELATIONS

### 3.1 INTRODUCTION

There are many possible threats to the validity of a statistical model, but one of the most potentially serious in many situations is the possibility of lack of statistical independence in the observations. When positive correlation exists and is not taken into account then the estimators will not be fully efficient in many situations. An even more serious problem is that statistical inferences from the model may be completely wrong. More specifically, under positive autocorrelation, it is well known that in the usual regression model the variances are inflated. This means that the usual confidence limits will be too narrow and the p-values will overstate the statistical significance of the results ([Wold and Jureen, 1966](#), §13.4). This effect of positive correlation applies potentially to all statistical models and not just to models involving time series data.

Indeed [McCullagh and Nelder \(1989, §2.2, p.26\)](#) voice this concern with respect to generalized linear models when they state: For the random part we assume independence and constant variance of the errors. These assumptions are strong and need checking.

In the case of time series data or when the order of collection of the observations is known, the residual autocorrelation function is often used as well as related statistical tests such as the Durbin-Watson test. For time series regression models, [Draper and](#)

[Smith \(1998\)](#) recommend plotting the residual for the  $t^{th}$  observation denoted by  $\hat{e}_t$  as a lagged one scatter plot of  $\hat{e}_{t+1}$  versus  $\hat{e}_t$ . Plotting  $\hat{e}_t$  vs.  $t$  may also be useful but since dependency relationships are usually strongest at lag one, the lagged one scatter plot often best reveals problems due to lack of independence in the data or residuals.

Scatter plots of  $\hat{e}_{t+1}$  versus  $\hat{e}_t$ , referred to as Poincaré return maps, are used in non-linear time series analysis ([Tong, 1998](#)) and in non-linear dynamics for identifying limit cycles ([Kaplan and Glass, 1995](#), §6.6, p.304). When applied for the purpose of diagnostic checking of a statistical model we will refer to this type of plot as a Poincaré Plot. Many statistical models assume that given the model specification the residuals are statistically independent. Violation of this assumption indicates that the model is misspecified and this misspecification may result in incorrect statistical inferences. Assuming that the observations are statistically independent, the observations may be ordered in various ways. For example, the observations could be ordered according to some covariate.

Poincaré plots may reveal non-linear forms of dependence or features not well summarized by the correlation coefficient. An informal method of statistical inference is to use a parametric bootstrap of the model to examine a sequence of Poincaré plots simulated when the independence assumption is known to hold. The significance of the residual plot can be informally judged by comparing with these plots. More formally, the Kendall rank correlation between  $\hat{e}_{t+1}$  and  $\hat{e}_t$  provides a statistical test for monotone dependency which may be helpful in some cases.

[Cleveland \(1979\)](#) introduced the residual dependency plot by plotting the residuals versus a covariate along with a loess smooth to help visualize whether there is a relationship. The Poincaré plot is recommended as a complement not a replacement to this plot. In the following examples, the residual dependency plot suggests that the

fitted model is an adequate model, but the Poincaré plot indicates a strong hidden positive dependency in the observations.

## 3.2 ILLUSTRATIVE EXAMPLES

### 3.2.1 Simulated data with hidden correlations

In this example, we simulate data with three covariate variables,  $X_1$ ,  $X_2$ ,  $X_3$ , and a response variable  $Y$  using the linear relationship  $y_i = \beta_0 + \sum_{j=1}^3 \beta_j x_{ji} + e_i$ ,  $i = 1, \dots, n$ . The sample size is 100. The error terms,  $e_i$ , exhibit hidden positive correlations according to the order values of  $X_1$  (see eqn. 3.3). First, we fit the classical ordinary least squares model, OLS, and test the overall adequacy of this model ( $H_0 : \beta_1 = \beta_2 = \beta_3 = 0$ ; vs.  $H_1 : \beta_j \neq 0$ , for at least one  $j = 1, 2, 3$ ).

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-0.7459	0.0765	-9.7454	0.0000
X1	0.0188	0.0052	3.6473	0.0005
X2	-0.0307	0.0647	-0.4751	0.6363
X3	-0.0198	0.0754	-0.2623	0.7939

Table 3.1: Classical ordinary least squares model (OLS).

The p-value corresponding to the F-Statistic, 0.47%, is very small which suggests that at least one of the covariates contributes significantly to the linear model. The residual plots (Draper and Smith, 1998; Cleveland, 1993) did not indicate any violation to the usual assumptions of the linear models. So we may think that the classical OLS model is a acceptable fitted model!

Now things get disquieting if we look at the Poincaré Plot. When the observations are ordered according to  $X_1$ , inspection of Poincaré plot in Figure 3.1 and the Kendall rank correlation test statistic, 7.33, with p-value =  $2.23 \times 10^{-12}\%$ , suggest a strong

positive dependence due to many nearly tied values and hence the classical OLS model is no longer the true fitted model for this data.

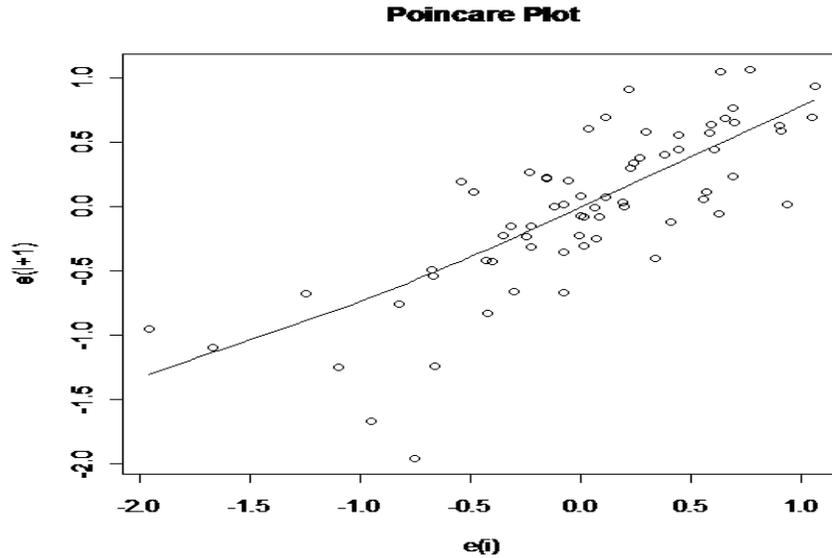


Figure 3.1: Poincaré plot of the residuals from OLS fitted model with a loess smooth.

Next we fit the true generalized least squares model, GLS, for the ordered data using the exact variance covariance matrix of the error term. The p-value for testing the hypothesis  $H_0 : \beta_1 = \beta_2 = \beta_3 = 0$  is 84%. Although the GLS model is the true model for this data, the large p-value suggests that none of the covariates contributes significantly to the model. The Poincaré Plot in Figure 3.2 as well as Kendall rank statistic, -0.41 with p-value = 68%, do not show any hidden correlation left.

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	-0.7311	0.5145	-1.42	0.1600
x1	-0.0046	0.0254	-0.18	0.8560
x2	-0.0098	0.0276	-0.35	0.7241
x3	0.0543	0.0377	1.44	0.1542

Table 3.2: Generalized least squares model (GLS) with exact covariance.

Comparing the estimated standard errors of the regression coefficients in Table

3.1 with those in Table 3.2, one can see that the standard errors of the intercept coefficient and the first covariate coefficient in the classical OLS are less than those in GLS model.

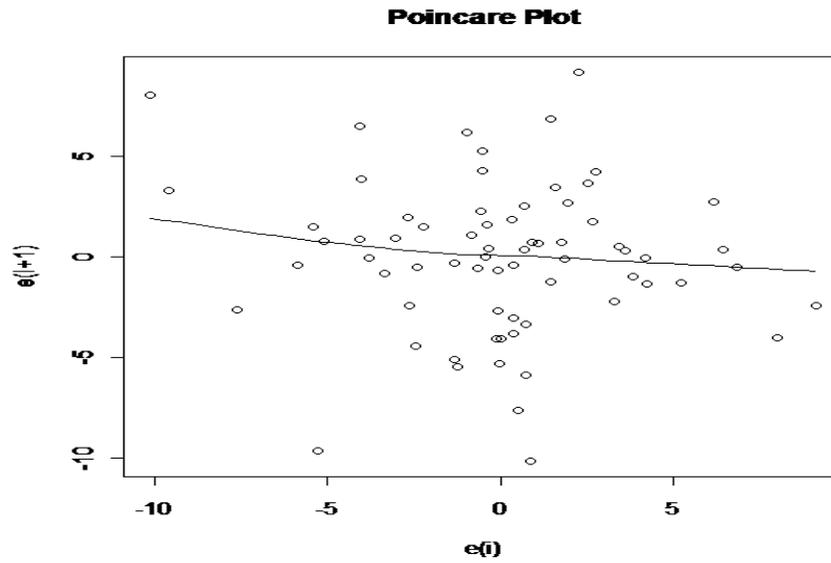


Figure 3.2: Poincaré plot of the residuals from GLS fitted model with a loess smooth.

In this chapter we will show by simulation experiments that this is the case in most cases. That is, when the error terms exhibit hidden positive autocorrelations according to the  $j^{th}$  covariate,  $X_j$ , then the wrong classical OLS model will often yield a small value of variance for the corresponding  $j^{th}$  coefficient in comparison with the true GLS model. Also, we will show that it is always the case that the variance of the intercept estimator in the OLS model has the lower value comparing to that true one.

### 3.2.2 Generalized linear modeling example

Deviance residuals (McCullagh and Nelder, 1989, §2.4.3) are frequently used for diagnostic checking with generalized linear models. Under the usual assumptions, the

observations in a specified model are statistically independent. This implies that the deviance residuals should also be approximately statistically independent. Consider the logistic regression of 189 births fitted by [Venables and Ripley \(2002, P.194-197\)](#). In this model a response variable, low birth weight, is fitted to 9 explanatory variables. One of the explanatory variable is age, which represents the age of the mother in years. Using this age variable to order the data, the resulting Poincaré plot of the deviance residuals shown in the lower left panel of [Figure 3.3](#) indicates a very strong positive residual dependence. For comparison, the Poincaré plots for 8 bootstrap simulations of deviance residuals are shown in the other panels. To aid visualization of the dependence relationship or lack thereof, a robust linear loess smooth with span equal to 1 is shown in each panel. [Figure 3.3](#) clearly reveals that there is significant strong positive dependence in the residuals and so statistical inferences from the fitted model may not be correct. [Figure 3.4](#) shows the residual dependency plot of the residuals vs. age. In this plot there is no apparent correlation in the residuals.

### **3.2.3 Loess fitting example**

[Cleveland \(1993, §3.6, p.122-127\)](#) fits a loess curve to some sunlight polarization data. The response variable is the Babinet point and the explanatory variable is the concentration of particulate matter in the atmosphere. In Cleveland's final fit, the Babinet point is regressed on the cube-root of concentration using a robust loess linear regression with a span of  $1/3$ . The data are ordered according to the concentration variable. The resulting Poincaré plot for this fit indicates very strong positive dependency in this data exists. It should be noted that [Cleveland \(1993, p.126, Figure 3.37\)](#) found the usual residual dependence plot satisfactory and there is no indication of positive correlation and dependency in this plot (see [Figure 3.5](#)). An improved fit cannot be obtained simply by changing the loess smoothing parameters in this case.

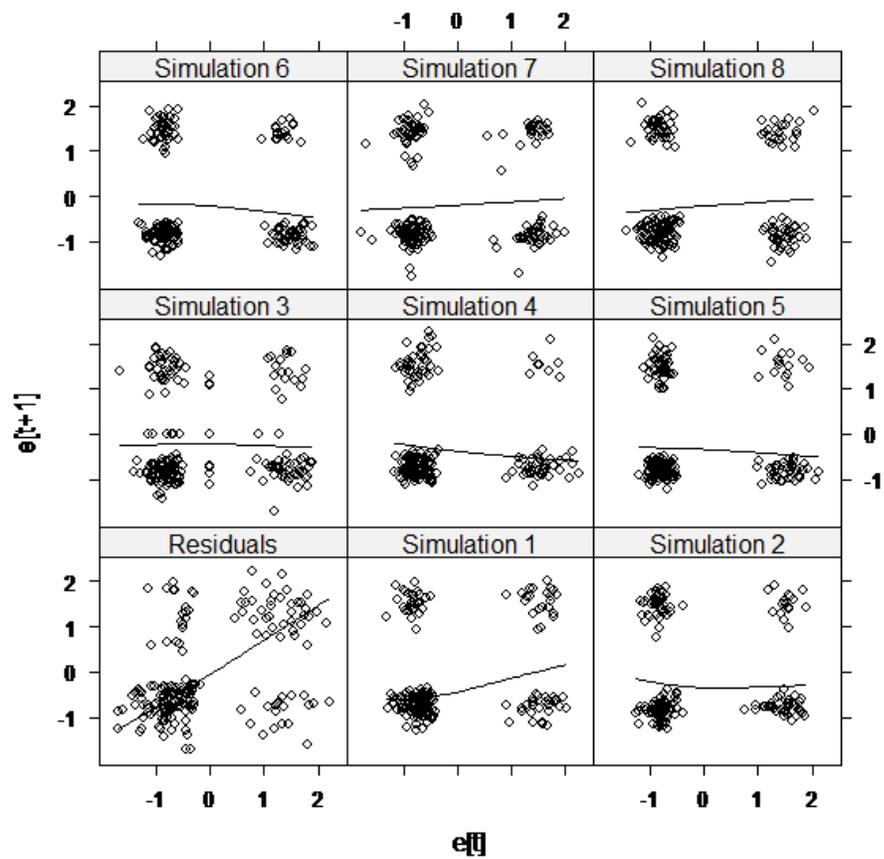


Figure 3.3: Poincaré plot of deviance residuals in the logistic regression of low birth weight on 9 explanatory variables and 8 parametric bootstrap simulations.

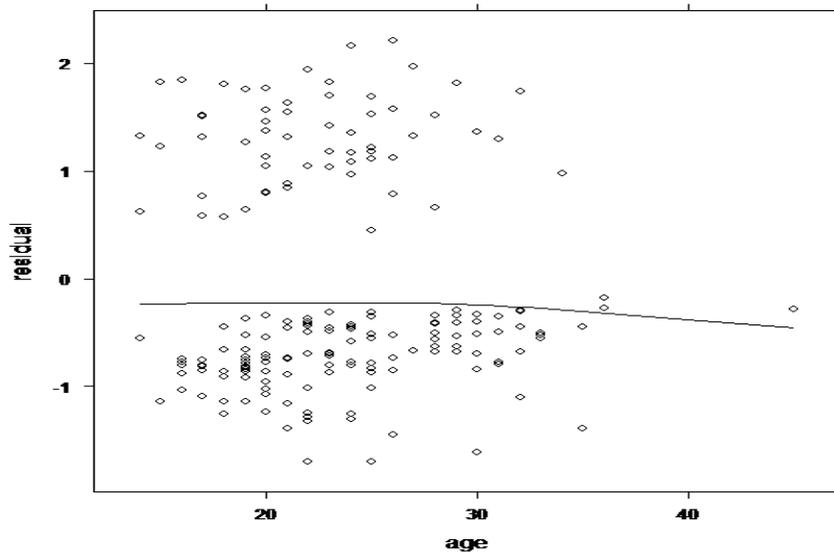


Figure 3.4: Residual dependency plot of residuals vs. age.

Choosing the span to be zero or close to zero can remove the positive dependence in the Poincaré plot but at the expense of increasing the variance and degrading the overall fit. Close inspection of Figure 3.6 shows that many points follow the 45 degree line. This means they are exactly equal and hence that both the dependent and independent variable are tied. Such ties are not consistent with the hypothesis of independent and continuously distributed data. Removing data values corresponding to ties in both variables and refitting the loess curve, as shown in Figure 3.7, does improve the Poincaré plot a little but there still remains strong positive dependence due to many nearly tied values.

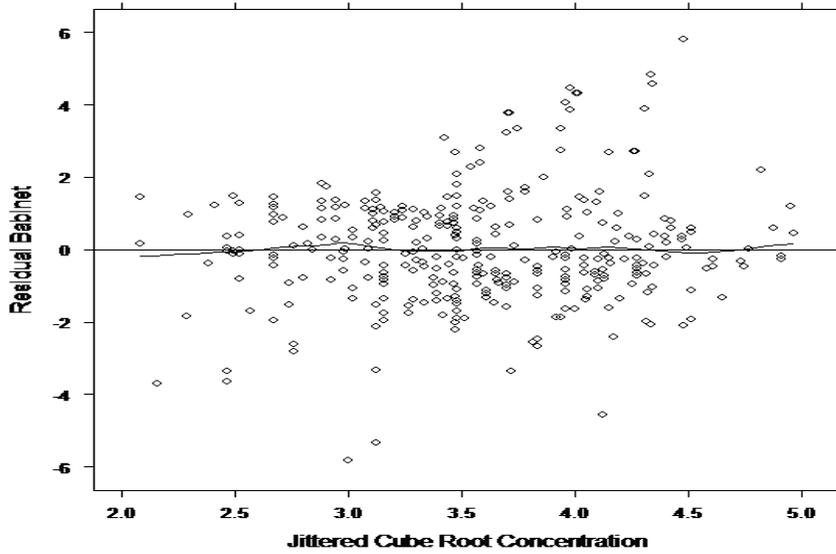


Figure 3.5: Residuals plot in the robust loess fit of Cleveland (1993, section 3.6) to the polarization data.

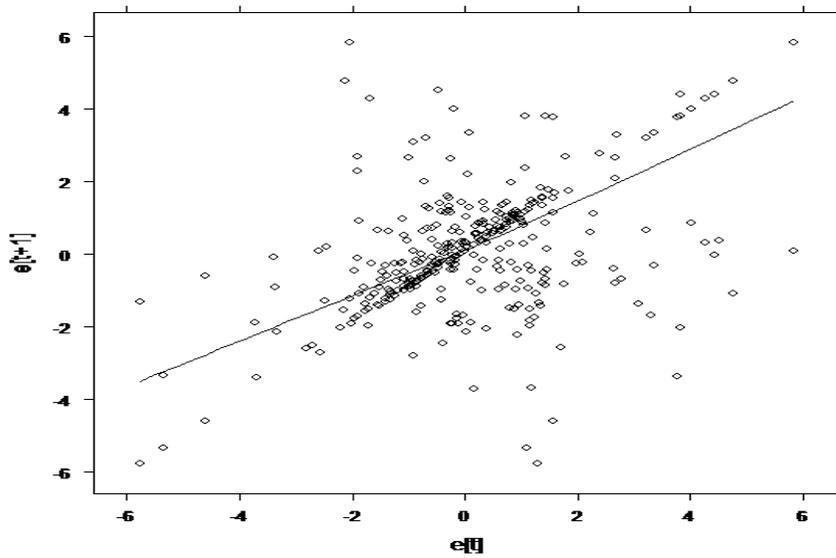


Figure 3.6: Poincaré plot of residuals in the robust loess fit to the polarization data.

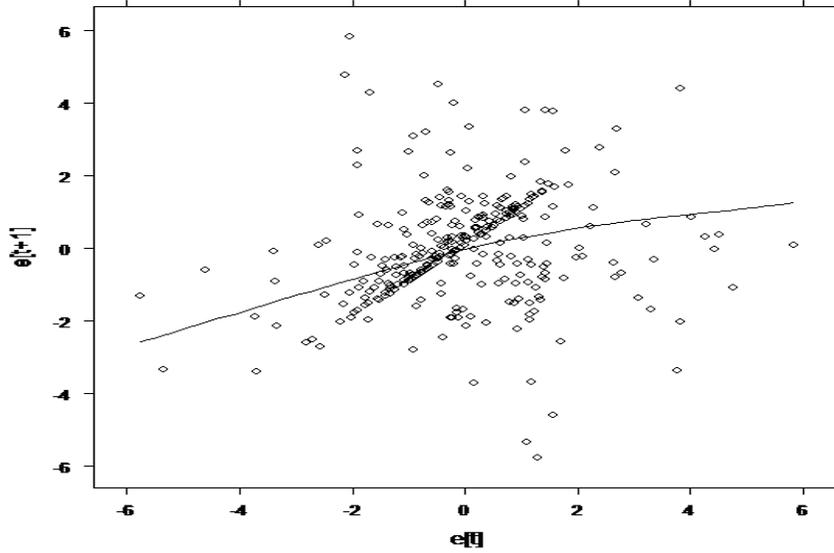


Figure 3.7: Poincaré plot of residuals in the robust loess fit to the polarization data with ties in both variables removed.

### 3.3 LINEAR REGRESSION MODEL

Let  $Y$  denotes the  $n \times 1$  vector of a response variable where  $y_i, i = 1, \dots, n$  can be modeled as a linear combination of  $k$  covariate variables,  $X_1, \dots, X_k$ ,

$$y_i = \beta_0 + \beta_1(x_{1i} - \bar{x}_1) + \dots + \beta_k(x_{ki} - \bar{x}_k) + e_i, \quad (3.1)$$

where  $\mathbf{e} = (e_1, e_2, \dots, e_n)'$  is a  $n \times 1$  vector of random components. Assume that  $\mathbf{e}$  has a normal distribution with mean zero and positive definite covariance matrix,

$$\mathbf{\Omega} = \sigma^2 \mathbb{I}_n - \gamma(\mathbf{h}), \quad (3.2)$$

where  $\sigma^2$  is the variance of  $e_i$ , for all  $i$ ,  $\gamma(\mathbf{h})$  is the semivariogram (Cressie, 1993),  $\mathbf{h}$  is the vector of the spacing between the values of a covariate variable, say  $X_p = (x_{p1}, \dots, x_{pn})'$ ,  $h_{ij} = \|x_{pi} - x_{pj}\|$  for some  $p = 1, 2, \dots, k$  and  $i, j = 1, 2, \dots, n$ .

In our simulation, the exponential covariance is used to create the hidden correlation in the error terms,

$$\boldsymbol{\Omega}_{(\mathbf{h};\sigma^2,r)} = \sigma^2 \exp(-\mathbf{h}/r), \quad (3.3)$$

where  $r$  is the covariance parameter, denotes the range of the semivariogram.

The conditional log-likelihood function with the parameters  $\boldsymbol{\theta} = (\boldsymbol{\beta}, \sigma^2, r)'$  may be written, after dropping the constant term as,

$$L(\boldsymbol{\theta}|\mathbf{X}, \mathbf{y}) = -\frac{1}{2} \ln |\boldsymbol{\Omega}| - \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'\boldsymbol{\Omega}^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}). \quad (3.4)$$

The simultaneous joint MLE of  $\boldsymbol{\theta}$  may be obtained by the iterative algorithm described in (McLeod et al., 2007) and the hypotheses about  $\boldsymbol{\theta}$  can be tested by the likelihood ratio test. Under the null hypothesis, there is no hidden correlation, the change in  $-2(\log \text{likelihood})$  between the independent error model and the model with hidden correlation should follow a chi-square distribution with 1 degree of freedom.

Recall that the Gauss-Fisher regression theorem is assuming that the random variables  $\mathbf{e}$  are identical and mutually independent with zero mean and finite equal variances,  $\boldsymbol{\Omega} = \sigma^2\mathbb{I}$ , as well as they are independent of the covariate observations.

**Theorem 2.** *On the Gauss-Fisher specification, the least-squares regression coefficients have the smallest variance among all unbiased estimates of the regression coefficients.*

**Note 1:** In the case that  $\boldsymbol{\Omega} = \sigma^2\mathbb{I}$ , the classical ordinary least squares, OLS, estimation of the regression coefficients is given by  $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \dots, \hat{\beta}_k)'$ ,

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}, \quad (3.5)$$

where  $\hat{\boldsymbol{\beta}}$  are unbiased estimators,  $E(\hat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$ ,  $\mathbf{X} = [\mathbf{1} : \mathbf{X}_c]$  is the design matrix, and  $\mathbf{X}_c$  is defined as follows

$$\mathbf{X}_c = \begin{pmatrix} x_{11} - \bar{x}_1 & x_{12} - \bar{x}_2 & \cdots & x_{1k} - \bar{x}_k \\ x_{21} - \bar{x}_1 & x_{22} - \bar{x}_2 & \cdots & x_{2k} - \bar{x}_k \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} - \bar{x}_1 & x_{n2} - \bar{x}_2 & \cdots & x_{nk} - \bar{x}_k \end{pmatrix}_{n \times k}. \quad (3.6)$$

The corresponding classical variance-covariance matrix of  $\hat{\boldsymbol{\beta}}$  is

$$\text{Cov}(\hat{\boldsymbol{\beta}}) = \hat{\sigma}^2 (\mathbf{X}'\mathbf{X})^{-1}, \quad (3.7)$$

and  $\hat{\sigma}^2$  is the OLS maximum likelihood estimator of  $\sigma^2$ .

**Note 2:** When the true variance-covariance of  $\mathbf{e}$ ,  $\boldsymbol{\Omega}$ , is not equal to  $\sigma^2\mathbb{I}$  then the estimates  $\hat{\boldsymbol{\beta}}$  are still unbiased but their true variance-covariance is

$$(\mathbf{X}'\mathbf{X})^{-1} \cdot \mathbf{X}'\boldsymbol{\Omega}\mathbf{X} \cdot (\mathbf{X}'\mathbf{X})^{-1}. \quad (3.8)$$

The intercept estimator equals  $\bar{\mathbf{y}} = \bar{\mathbf{e}}$  (Wold and Jureen, 1966); hence its variance is

$$\underbrace{\frac{\hat{\sigma}^2}{n}}_{\text{classical ols Var}(\hat{\beta}_0)} + \underbrace{\frac{2}{n^2} \sum_{i=1}^{n-1} \sum_{j=i+1}^n \text{Cov}(\hat{e}_i, \hat{e}_j)}_{> 0}. \quad (3.9)$$

**Note 3:** The generalized least squares, GLS, estimation of the regression coefficients,  $\tilde{\boldsymbol{\beta}}$ , (Draper and Smith, 1998; Thisted, 1998),

$$\tilde{\boldsymbol{\beta}} = (\mathbf{X}'\boldsymbol{\Omega}^{-1}\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\Omega}^{-1}\mathbf{Y}, \quad (3.10)$$

which also are unbiased estimators with variance-covariance matrix,

$$\text{Cov}(\tilde{\boldsymbol{\beta}}) = (\mathbf{X}'\boldsymbol{\Omega}^{-1}\mathbf{X})^{-1}. \quad (3.11)$$

In such a case, the intercept estimator equals  $\bar{\mathbf{y}}^{\star} = \bar{\mathbf{e}}^{\star}$ , where  $\mathbf{e}^{\star} = K^{-1}\mathbf{e}$ , and  $\sigma K$  is the Cholesky decomposition of  $\boldsymbol{\Omega}$  so that  $\boldsymbol{\Omega} = \sigma^2 K K'$ . The variance of the intercept estimator is

$$\underbrace{\frac{\hat{\sigma}^2}{n}}_{\text{classical ols Var}(\hat{\beta}_0)} + \underbrace{\frac{2}{n^2} \sum_{i=1}^{n-1} \sum_{j=i+1}^n \text{Cov}(\hat{e}^{\star}_i, \hat{e}^{\star}_j)}_{> 0}. \quad (3.12)$$

Equations 3.9 and 3.12 clearly indicate that when  $\boldsymbol{\Omega}$  is positive-definite  $\neq \sigma^2\mathbb{I}$ , then the classical OLS variance of the intercept estimator is always less than it should be and this implies that the corresponding p-value may overstates the statistical significance of the results.

The simulation experiments in the next section confirm this fact. In addition, when the error terms exhibit hidden positive autocorrelations according to the  $j^{\text{th}}$  covariate  $X_j$ , then the classical OLS model will often yield a smaller value of variance compared to the true and GLS models.

### 3.4 SIMULATION RESULTS

In this section, we consider fitting three different regression models - (a) the classical ordinary least squares regression as in Equation 3.5, (b) the true ordinary least squares using the estimated variance-covariance matrix, True OLS, as in Equation 3.8, (c) and the generalized least squares regression, GLS, as in Equation 3.10.

The results are based on 1000 simulations using three different sample sizes, 30,

60, and 90. In each simulation we generate the random errors from the multivariate normal distribution with zero mean and variance-covariance function  $\mathbf{\Omega}$  as given in Equation 3.3, where  $\sigma^2$  is assumed to be 1. We choose the variogram parameters  $r = 3$ ,  $r = 6$ , and  $r = 9$  with a design matrix of  $k = 1, 2, 3$  covariates, where  $X_1$  is used to create the hidden correlation. We generate the irregular distances vector  $\mathbf{h}$  from the  $U(0, 1)$  distribution,  $N(0, 1)$  distribution, and  $t_5$  distribution. Finally we calculate the variances of the least squares parameter estimates using Equations 3.7, 3.8, and 3.11 and compare them to each other. Tables 3.3-3.8 reports some of these results.

The results suggest that the classical OLS variances of the estimators of  $\beta_0$  and  $\beta_1$  are always less than those variances of true OLS and GLS. On the other hand, the classical OLS variances of the estimators of  $\beta_2$  and  $\beta_3$  are greater than those variances of true OLS and GLS.

	Classical OLS	True OLS & C.I	GLS & C.I
<b>r = 3</b>			
Var( $\hat{\beta}_0$ )	0.0333	0.2291 (0.2089, 0.2493)	0.2242 (0.2044, 0.2440)
Var( $\hat{\beta}_1$ )	0.0007	0.0039 (0.0036, 0.0043)	0.0036 (0.0032, 0.0039)
<b>r = 6</b>			
Var( $\hat{\beta}_0$ )	0.0333	0.3846 (0.3496, 0.4196)	0.3757 (0.3415, 0.4100)
Var( $\hat{\beta}_1$ )	0.0007	0.0045 (0.0041, 0.0049)	0.0039 (0.0035, 0.0043)
<b>r = 9</b>			
Var( $\hat{\beta}_0$ )	0.0333	0.4889 (0.4442, 0.5336)	0.4745 (0.4311, 0.5178)
Var( $\hat{\beta}_1$ )	0.0007	0.0044 (0.0040, 0.0048)	0.0038 (0.0035, 0.0042)

Table 3.3: Distance vector generated from  $t_5$ ; k=1, range = 3, 6, 9, and n=30.

	Classical OLS	True OLS & C.I	GLS & C.I
<b>r = 3</b>			
Var( $\hat{\beta}_0$ )	0.0167	0.2141 (0.1947, 0.2335)	0.2115 (0.1926, 0.2305)
Var( $\hat{\beta}_1$ )	0.0002	0.0019 (0.0017, 0.0020)	0.0016 (0.0014, 0.0017)
<b>r = 6</b>			
Var( $\hat{\beta}_0$ )	0.0167	0.3285 (0.2982, 0.3588)	0.3287 (0.2985, 0.3589)
Var( $\hat{\beta}_1$ )	0.0002	0.0026 (0.0023, 0.0028)	0.0022 (0.0020, 0.0024)
<b>r = 9</b>			
Var( $\hat{\beta}_0$ )	0.0167	0.4240 (0.3845, 0.4636)	0.4230 (0.3834, 0.4626)
Var( $\hat{\beta}_1$ )	0.0002	0.0027 (0.0024, 0.0029)	0.0023 (0.0021, 0.0025)

Table 3.4: Distance vector generated from  $U(0, 1)$ ;  $k=1$ , range = 3, 6, 9, and  $n=60$ .

	Classical OLS	True OLS & C.I	GLS & C.I
<b>r = 3</b>			
Var( $\hat{\beta}_0$ )	1.11e-02	0.0917 (0.0839, 0.0995)	0.0921 (0.0842, 0.1001)
Var( $\hat{\beta}_1$ )	3.44e-05	0.00025 (0.0002, 0.0003)	0.00023 (0.0002, 0.0003)
<b>r = 6</b>			
Var( $\hat{\beta}_0$ )	1.11e-02	0.1771 (0.1613, 0.1929)	0.1771 (0.1613, 0.1929)
Var( $\hat{\beta}_1$ )	3.44e-05	0.0004 (0.0004, 0.0004)	0.0004 (0.0003, 0.0004)
<b>r = 9</b>			
Var( $\hat{\beta}_0$ )	1.11e-02	0.2492 (0.2272, 0.2712)	0.2402 (0.2191, 0.2613)
Var( $\hat{\beta}_1$ )	3.44e-05	0.0005 (0.0004, 0.0005)	0.0004 (0.0004, 0.0005)

Table 3.5: Distance vector generated from  $N(0, 1)$ ;  $k=1$ , range = 3, 6, 9, and  $n = 90$ .

	Classical OLS	True OLS & C.I	GLS & C.I
<b>r = 3</b>			
Var( $\hat{\beta}_0$ )	0.0333	0.2448 (0.2228, 0.2668)	0.2415 (0.2202, 0.2629)
Var( $\hat{\beta}_1$ )	0.0010	0.0050 (0.0045, 0.0055)	0.0044 (0.0040, 0.0048)
Var( $\hat{\beta}_2$ )	0.0502	0.0235 (0.0212, 0.0258)	0.0077 (0.0069, 0.0085)
<b>r = 6</b>			
Var( $\hat{\beta}_0$ )	0.0333	0.4065 (0.3701, 0.4429)	0.3961 (0.3610, 0.4312)
Var( $\hat{\beta}_1$ )	0.0010	0.0056 (0.0051, 0.0061)	0.0049 (0.0045, 0.0054)
Var( $\hat{\beta}_2$ )	0.0502	0.0150 (0.0136, 0.0165)	0.00382 (0.0034, 0.0042)
<b>r = 9</b>			
Var( $\hat{\beta}_0$ )	0.0333	0.5136 (0.4675, 0.5597)	0.4977 (0.4536, 0.5417)
Var( $\hat{\beta}_1$ )	0.0010	0.0053 (0.0048, 0.0058)	0.0047 (0.0043, 0.0052)
Var( $\hat{\beta}_2$ )	0.0502	0.0114 (0.0103, 0.0125)	0.0026 (0.0023, 0.0028)

Table 3.6: Distance vector generated from  $N(0, 1)$ ;  $k=2$ , range = 3, 6, 9, and  $n=30$ .

	Classical OLS	True OLS & C.I	GLS & C.I
<b>r = 3</b>			
Var( $\hat{\beta}_0$ )	1.67e-02	0.1225 (0.1111, 0.1339)	0.1196 (0.1086, 0.1306)
Var( $\hat{\beta}_1$ )	9.16e-05	0.0006 (0.0005, 0.0006)	0.0005 (0.0005, 0.0006)
Var( $\hat{\beta}_2$ )	1.02e-02	0.0084 (0.0076, 0.0093)	0.0025 (0.0022, 0.0027)
<b>r = 6</b>			
Var( $\hat{\beta}_0$ )	1.67e-02	0.2228 (0.2028, 0.2429)	0.2154 (0.1961, 0.2347)
Var( $\hat{\beta}_1$ )	9.16e-05	0.0009 (0.0008, 0.0010)	0.0008 (0.0007, 0.0009)
Var( $\hat{\beta}_2$ )	1.02e-02	0.0069 (0.0062, 0.0077)	0.0013 (0.0012, 0.0014)
<b>r = 9</b>			
Var( $\hat{\beta}_0$ )	1.67e-02	0.3082 (0.2811, 0.3352)	0.2955 (0.2700, 0.3210)
Var( $\hat{\beta}_1$ )	9.16e-05	0.0011 (0.0010, 0.0012)	0.0009 (0.0008, 0.0010)
Var( $\hat{\beta}_2$ )	1.02e-02	0.0054 (0.0048, 0.0060)	0.0008 (0.0007, 0.0009)

Table 3.7: Distance vector generated from  $t_5$ ; k=2, range = 3, 6, 9, and n=60.

	Classical OLS	True OLS & C.I	GLS & C.I
<b>r = 3</b>			
Var( $\hat{\beta}_0$ )	0.0333	0.2658 (0.2441, 0.2874)	0.2642 (0.2420, 0.2865)
Var( $\hat{\beta}_1$ )	0.0010	0.0049 (0.0045, 0.0054)	0.0044 (0.0040, 0.0048)
Var( $\hat{\beta}_2$ )	0.0413	0.0249 (0.0223, 0.0274)	0.0078 (0.0071, 0.0086)
Var( $\hat{\beta}_3$ )	0.0374	0.0244 (0.0220, 0.0268)	0.0081 (0.0073, 0.0089)
<b>r = 6</b>			
Var( $\hat{\beta}_0$ )	0.0333	0.4267 (0.3904, 0.4631)	0.4086 (0.3730, 0.4443)
Var( $\hat{\beta}_1$ )	0.0010	0.0055 (0.0050, 0.0060)	0.0047 (0.0043, 0.0051)
Var( $\hat{\beta}_2$ )	0.0413	0.0172 (0.0150, 0.0194)	0.0040 (0.0035, 0.0045)
Var( $\hat{\beta}_3$ )	0.0374	0.0161 (0.0144, 0.0179)	0.0040 (0.0035, 0.0046)
<b>r = 9</b>			
Var( $\hat{\beta}_0$ )	0.0333	0.5384 (0.4924, 0.5843)	0.5126 (0.4676, 0.5575)
Var( $\hat{\beta}_1$ )	0.0010	0.0053 (0.0048, 0.0057)	0.0045 (0.0041, 0.0049)
Var( $\hat{\beta}_2$ )	0.0413	0.0127 (0.0111, 0.0143)	0.0027 (0.0024, 0.0030)
Var( $\hat{\beta}_3$ )	0.0374	0.0117 (0.0104, 0.0130)	0.0028 (0.0024, 0.0032)

Table 3.8: Distance vector generated from  $N(0, 1)$ ;  $k=3$ , range = 3, 6, 9, and  $n=30$ .

### 3.5 CONCLUDING REMARKS

After some suitable ordering of the fitted residuals,  $\hat{e}_i, i = 1, 2, \dots, n$ , it is suggested to make use of Poincaré plots that is, scatter plots of  $\hat{e}_{i+1}$  versus  $\hat{e}_i$  along with a robust loess trend to check for lack of statistical independence. Poincaré plots are helpful in detecting hidden positive correlations in the fitted model which may invalidate statistical inferences due to incorrect small p-values in classical ordinary least squares regression. One can use Kendall rank correlation test as well.

One should avoid using Poincaré residual plots for data that has been ordered by the response variable, since positive dependence is expected in the Poincaré residuals plot even when the assumption of independence holds in this case.

### 3.6 FUTURE WORK

Many published statistical results in medicine seem to overstate their statistical significance ([Matthews, 1998](#)). One of the possible reason for this apparent lack of robustness in medical statistics, at least in some cases, could simply be model misspecification due to lack of statistical independence. Further research is needed on ways to improve statistical modeling when statistical dependence is found. The exponential variogram model maybe useful for describing and modeling such a correlation in some of these data.

## REFERENCES

- Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association* 74, 829–836.
- Cleveland, W. S. (1993). *Visualizing data*. Hobart Press, Summit: New Jersey.
- Cressie, N. A. C. (1993). *Statistics for spatial data* (revised ed.) New York: John Wiley and Sons, Inc.
- Draper, N. R. and Smith, H. (1998). *Applied regression analysis* (3rd ed). New York: Wiley.
- Jones, R. H. and Boadi-Boateng, F. (1991). Unequally spaced longitudinal data with AR (1) serial correlation. *Biometrika* 47(11), 161–175.
- Kaplan, R. and Glass, L. (1995). *Understanding nonlinear dynamics* (3rd ed). New York: Springer-Verlag.
- Kendall, M. (1938). A New Measure of Rank Correlation. *Biometrika* 30, 81–89.
- Matthews, R. (1998). *The Great Health Hoax*. The Sunday Telegraph, September 13.
- McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models* (2nd ed). New York: Hall/CRC.
- McLeod, A.I., Yu, H., and Krougly, Z. (2007). Algorithms for Linear Time Series Analysis: With R Package. *Journal of Statistical Software* 32(5), 1–26.
- Thisted, R. A. (1998). *Elements of statistical computing*. London: Chapman and Hall.
- Tong, H. (1990). *Non-linear time series*. Oxford University Press: Oxford.
- Venables, W. N. and Ripley B.(2002). *Modern applied statistics with S*. Springer-Verlag: New York.
- Wilfredo, P. (2007). *Long-Memory Time Series: Theory and Methods*. Wiley Series in Probability and Statistics.
- Wold, H. and Jureen, L. (1966). *Demand analysis; a study in econometrics* (4th ed). New York: John Wiley and Sons, Inc.

# CURRICULUM VITAE

ESAM MAHDI

## Post-secondary Education and Degrees

- **Ph.D.** (*Statistics*), The University of Western Ontario, London, Canada, 2011.
- **M.Sc.** (*Mathematical Statistics*), IUG, Gaza Strip, Palestine, 2000.
- **B.Sc.** (*Mathematics*), IUG, Gaza Strip, Palestine, 1998.

## Honours and Awards

- Western Graduate Research Scholarship, The University of Western Ontario, London Ontario, Canada, 2007 – 2011.
- Western Graduate Scholarship, The University of Western Ontario, London Ontario, Canada, 2007 – 2011.
- Joint Statistical Meeting (JSM), Vancouver, Canada, (Poster winner) 2010.
- Robert and Ruth Lumsden Graduate Awards in Science. The University of Western Ontario, London Ontario, Canada, 2008 – 2009.
- Ontario Graduate Scholarship in Science and Technology, Canada. (Awarded) 2011 – 2012.

## Articles Accepted for Publications or Under Review by Academic Journal

- Mahdi, E. and McLeod, A. I. (2011, accepted). Improved multivariate portmanteau test. *Journal of Time Series Analysis*. (JTSA - 3192)
- McLeod, A. I., Yu, H. and Mahdi, E. (2011, accepted). *Time Series Analysis with R*.
- Mahdi, E. and McLeod, A. I. (2011, to be submitted). Portmanteau test for time series models: R package **portes**.