Universidade Federal de Pernambuco
Centro de Ciências Exatas e da Natureza
Programa de Pós-Graduação em Estatística

Vinícius Teodoro Scher

Portmanteau testing inference in beta autoregressive moving average models

Recife
2017

Vinícius Teodoro Scher

Portmanteau testing inference in beta autoregressive moving average models

Master's thesis submitted to the Graduate Program in Statistics, Department of Statistics, Universidade Federal de Pernambuco as a requirement to obtain a Master's degree in Statistics.

Advisor: *Francisco Cribari-Neto*
Co-advisor: *Fábio Mariano Bayer*

Recife
2017

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

S326p Scher, Vinícius Teodoro
Portmanteau testing inference in beta autoregressive moving average models / Vinícius Teodoro Scher. – 2017.
69 f.: il., fig., tab.

Orientador: Francisco Cribari Neto.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CCEN, Estatística, Recife, 2017.
Inclui referências e apêndices.

1. Análise de regressão. 2. Regressão beta. I. Cribari Neto, Francisco (orientador). II. Título.

519.536 CDD (23. ed.) UFPE- MEI 2017-189

VINÍCIUS TEODORO SCHER

PORTMANTEAU TESTING INFERENCE IN BETA AUTOREGRESSIVE MOVING AVERAGE MODELS

Dissertação apresentada ao Programa de Pós-Graduação em Estatística da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Estatística.

Aprovada em: 02 de agosto de 2017.

BANCA EXAMINADORA

Prof. Francisco Cribari Neto
UFPE

Prof. Abraão David Costa do Nascimento
UFPE

Prof. Aluísio de Souza Pinheiro
Unicamp

To my father, with love.

Acknowledgements

Initially, I would like to thank my family for their support and care, especially my mother, Maria, who has always encouraged me to pursue my dreams.
To my father Olavo, wherever you are, I share this realization with you. I miss you every single day.
Special thanks to my advisor, Francisco Cribari-Neto, for his constant guidance, for teaching me new things, for believing in my ability, for the confidence he placed on my work and for being a role model of teacher and professional. He has always received me with the patience and humility that are typical of a true mentor.
I would also like to thank my co-advisor, Fábio Bayer, who has always encouraged me to follow an academic career, and showed me that science is the pathway to understanding the world.
My thanks to the professors at the UFPE Statistics Department, especially to professors Alex Dias, Gauss Cordeiro, Francisco Cysneiros and Audrey Cysneiros. I would also like to thank Valéria Bittencourt for her dedication to her job and for helping me with bureaucratic matters.
I am also very grateful to Thainá, for the companionship and love during the time we were together in heart, on this long road.
I am grateful to all my friends who have always supported me, especially Vagner, for many years of friendship and partnership.
To my colleagues and almost brothers Jucelino and Yuri, thanks for the support and for our true friendship during my stay in Recife.
To my colleagues Alejandro, Olívia, Rodney, Cristina and César, for always having the right word and for the examplary fellowship.
Finally, I thank Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) and Centro Nacional de Supercomputação of Universidade Federal do Rio Grande do Sul (CESUP/UFRGS).

Open your mind to what I shall reveal and
seal it in, for to have understood and not
retained, as knowledge does not count.

—DANTE ALIGHIERI (Divine Comedy,1300)

Abstract

The class of beta autoregressive moving average (βARMA) models is useful for modeling time series data that assume values in the standard unit interval, such as rates and proportions. This thesis is composed of two main and independent chapters. In the first part, we consider portmanteau testing inference in the class of βARMA models. To that end, we use tests that have been developed for Gaussian models, such as the Ljung and Box, Monti, Dufour and Roy, Kwan and Sim, and Lin and McLeod tests. We also consider bootstrap variants of the Ljung and Box, Monti, Dufour and Roy, and Kwan and Sim tests. Moreover, we propose two new test statistics which, like the Monti statistic, are based on residual partial autocorrelations. Additionally, we present and discuss results from Monte Carlo simulations and an empirical application. The second part of the thesis focuses on the recursive nature of βARMA log-likelihood derivatives under moving average dynamics. We provide closed form expressions for the relevant derivatives by considering errors in the predictor scale.

Keywords: Bootstrap. βARMA. Model diagnostic. Monte Carlo simulation. Portmanteau test. Residual analysis.

Resumo

A classe de modelos beta autorregressivos de médias móveis (βARMA) é útil para modelar dados que assumem valores no intervalo unitário padrão, como taxas e proporções. A presente dissertação tem como tema tal classe de models e é composta por dois capítulos principais e independentes. Na primeira parte, consideramos inferências baseadas em testes portmanteau na classe de modelos βARMA. Para tanto, utilizamos testes que foram desenvolvidos para modelos gaussianos, como os testes de Ljung e Box, Monti, Dufour e Roy, Kwan e Sim, e Lin e McLeod. Também consideramos variantes bootstrap dos testes de Ljung e Box, Monti, Dufour e Roy and Kwan e Sim. Adicionalmente, propomos duas novas estatísticas de testes que, tal qual a estatística de Monti, são baseadas em autocorrelações parciais dos resíduos. Apresentamos e discutimos resultados de simulações de Monte Carlo e uma aplicação empírica. A segunda parte da dissertação aborda a natureza recursiva das derivadas da função de log-verossimilhança βARMA sob dinâmica de médias móveis. Nós fornecemos expressões em forma fechada para as derivadas relevantes considerando erros na escala do preditor.

Palavras-chave: Análise de resíduos. Bootstrap. βARMA. Diagnóstico. Simulação de Monte Carlo. Teste portmanteau.

List of Figures

2.1 Null rejection rates of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}, βAR(1) with $\varphi = 0.2$. 25
2.2 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βAR(1) and the true data generating process is βARMA$(1,1)$, $\varphi = 0.2$ and $\theta = 0.2$ 33
2.3 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βAR(1) and the true data generating process is βARMA$(1,1)$, $\varphi = 0.2$ and $\theta = 0.5$ 34
2.4 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βAR(1) and the true data generating process is βARMA$(1,1)$, $\varphi = 0.2$ and $\theta = 0.8$ 34
2.5 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βAR(1) and the true data generating process is βARMA$(1,1)$, $\varphi = 0.5$ and $\theta = 0.2$ 35
2.6 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βAR(1) and the true data generating process is βARMA$(1,1)$, $\varphi = 0.5$ and $\theta = 0.5$ 35
2.7 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βAR(1) and the true data generating process is βARMA$(1,1)$, $\varphi = 0.5$ and $\theta = 0.8$ 36
2.8 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βAR(1) and the true data generating process is βARMA$(1,1)$, $\varphi = 0.8$ and $\theta = 0.2$ 36
2.9 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βAR(1) and the true data generating process is βARMA$(1,1)$, $\varphi = 0.8$ and $\theta = 0.5$ 37
2.10 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βAR(1) and the true data generating process is βARMA$(1,1)$, $\varphi = 0.8$ and $\theta = 0.8$ 37
2.11 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4},Q_{KW4p}; the fitted model is βAR(1) and the true model is βARMA$(1,1)$, $\varphi = 0.2$. 38
2.12 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βMA(1) and the true model is βARMA$(1,1)$, $\varphi = 0.2$ and $\theta = 0.2$ 38
2.13 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βMA(1) and the true model is βARMA$(1,1)$, $\varphi = 0.2$ and $\theta = 0.5$ 39
2.14 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βMA(1) and the true model is βARMA$(1,1)$, $\varphi = 0.2$ and $\theta = 0.8$ 39
2.15 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βMA(1) and the true model is βARMA$(1,1)$, $\varphi = 0.5$ and $\theta = 0.2$ 40
2.16 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βMA(1) and the true model is βARMA$(1,1)$, $\varphi = 0.5$ and $\theta = 0.5$ 40
2.17 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βMA(1) and the true model is βARMA$(1,1)$, $\varphi = 0.5$ and $\theta = 0.8$ 41
2.18 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βMA(1) and the true model is βARMA$(1,1)$, $\varphi = 0.8$, $\theta = 0.2$ and $\varphi = 0.8$, $\theta = 0.8$ 42

2.19 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βARMA$(1,1)$ and the true model is βARMA$(2,1)$, $\varphi = 0.2, 0.2$ and $\theta = 0.2$ 42
2.20 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βARMA$(1,1)$ and the true model is βARMA$(2,1)$, $\varphi = 0.2, 0.2$ and $\theta = 0.5$ 43
2.21 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βARMA$(1,1)$ and the true model is βARMA$(2,1)$, $\varphi = 0.2, 0.2$ and $\theta = 0.8$ 43
2.22 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βARMA$(1,1)$ and the true model is βARMA$(2,1)$, $\varphi = 0.2, 0.5$ and $\theta = 0.2$ 44
2.23 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βARMA$(1,1)$ and the true model is βARMA$(2,1)$, $\varphi = 0.2, 0.5$ and $\theta = 0.5$ 44
2.24 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βARMA$(1,1)$ and the true model is βARMA$(2,1)$, $\varphi = 0.2, 0.5$ and $\theta = 0.8$ 45
2.25 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βARMA$(1,1)$ and the true model is βARMA$(2,1)$, $\varphi = 0.2, 0.8$ and $\theta = 0.2$ 45
2.26 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βARMA$(1,1)$ and the true model is βARMA$(2,1)$, $\varphi = 0.2, 0.8$, $\theta = 0.5$ and $\varphi = 0.2, 0.8$, $\theta = 0.8$ 46
2.27 Average rates of stocked energy in the South of Brazil. 47
2.28 Correlogram and partial correlogram. 47
2.29 Standardized residuals from the fitted βARMA$(1,1)$ model. 48
2.30 Residual correlogram and partial correlogram. 48
2.31 Portmanteau tests p-values. 49
2.32 Energy stored rates (solid lines) and predict values (dashed lines) from the fitted model. 50

3.1 Numerical partial derivative (solid line), analytic partial derivative with recursion (dashed line) and analytic partial derivative without recursion (dot line) for the βARMA$(1,1)$ model, $\varphi = 0.2$. 55
3.2 Numerical partial derivative (solid line), analytic partial derivative with recursion (dashed line) and analytic partial derivative without recursion (dot line) for the βARMA$(1,1)$ model, $\varphi = 0.2$; first range of values for θ. 56
3.3 Numerical partial derivative (solid line), analytic partial derivative with recursion (dashed line) and analytic partial derivative without recursion (dot line) for the βARMA$(1,1)$ model, $\varphi = 0.2$; second range of values for θ. 56
3.4 Numerical partial derivative (solid line), analytic partial derivative with recursion (dashed line) and analytic partial derivative without recursion (dot line) for the βARMA$(1,1)$ model, $\varphi = 0.2$; third range of values for θ. 57

List of Tables

2.1 Mean size distortion rates of portmanteau tests without bootstrap resampling, $\gamma = 5\%$. 25
2.2 Null rejection rates of portmanteau tests, βAR(1) model, $\gamma = 5\%$. 27
2.3 Null rejection rates of portmanteau tests for a βMA(1) model; $\gamma = 5\%$. 28
2.4 Null rejection rates of portmanteau tests, βARMA$(1,1)$ model, $\varphi = 0.2$, $\gamma = 5\%$. 29
2.5 Null rejection rates of portmanteau tests, βARMA$(1,1)$ model, $\varphi = 0.5$, $\gamma = 5\%$. 30
2.6 Null rejection rates of portmanteau tests, βARMA$(1,1)$ model, $\varphi = 0.8$, $\gamma = 5\%$. 31
2.7 Mean size distortion (MSD) rates of bootstrap portmanteau tests, $\gamma = 5\%$. 32
2.8 Null rejection rates of bootstrap portmanteau tests, βAR(1) model, $\gamma = 5\%$. 32
2.9 Null rejection rates of bootstrap portmanteau tests, βMA(1) model, $\gamma = 5\%$. 33
2.10 Descriptive statistics on the average rates of stocked energy in the South of Brazil. 46
2.11 Parameter estimates, standard errors and p-values; βARMA$(1,1)$ model. 47
2.12 Mean absolute forecasting errors, βARMA$(1,1)$, ARMA$(1,1)$ and AR(2). 50

Contents

1 Preliminaries **13**

2 Portmanteau testing inference in beta autoregressive moving average models **14**
2.1 Introduction 14
2.2 The βARMA model 15
2.3 Portmanteau Tests 17
2.3.1 Standard portmanteau tests 17
2.3.2 New portmanteau tests 22
2.4 Numerical evidence 23
2.5 Empirical Application 46
2.6 Conclusion 50

3 Recursion in Partial Derivatives **52**
3.1 Introduction 52
3.2 βARMA log-likelihood recursive partial derivatives 52

References **58**

Appendix A **61**
A.1 Null rejection rates without resorting to bootstrap resampling 61
A.2 Empirical powers without resorting to bootstrap resampling 65

Appendix B **68**
B.1 The modified *tsdiag* function 68

CHAPTER 1

Preliminaries

In this chapter we present a brief outline of the dissertation, which is composed of two main and independent chapters. The subject of the dissertation is portmanteau testing inference in the class of beta autoregressive moving average models proposed by Rocha and Cribari-Neto (2009). Since the seminal work of Box and Pierce (1970), many different portmanteau tests have been proposed in the literature for different time series data that assume values in the real line. We evaluate the accuracy of testing inferences based on such criteria when the time series data assume values in the standard unit interval, $(0,1)$, and are modeled using a beta autorregressive moving average model. We also propose two new portmanteau test statistics.

In Chapter 2 we address the issue of performing portmanteau testing inference using time series data that assume values in the standard unit interval. Our focus lies in the class of beta autoregressive moving average (βARMA) models. We consider different portmanteau goodness of fit tests such as Ljung and Box (1978), Monti (1994), Dufour and Roy (1986), Kwan and Sim (1996a,b) e Lin and McLeod (2006). The numerical evidence we provide show that the bootstrap test proposed in Lin and McLeod (2006). We also consider bootstrap va deliveres accurate inferences in small sample sizes when used with βARMA models. We consider bootstrap variants of the tests introduced by Ljung and Box (1978), Monti (1994), Dufour and Roy (1986) and Kwan and Sim (1996a,b). Moreover, we propose two new test statistics which, like the statistic introduced by Monti (1994), are based on residual partial autocorrelations. The simulation results we present show that the new tests can be powerful than that of Monti (1994). We also present and discuss the results of an empirical application.

Chapter 3 is devoted to the recursive nature of βARMA log-likelihood derivatives under moving averages dynamics. Closed form expressions for the relevant derivatives are provided by considering errors in the predictor scale.

The notation and terminology used is consistent throughout the thesis. The programming routines for Monte Carlo simulations and all figures in this thesis were developed for the R statistical computing environment (R Core Team, 2017).

CHAPTER 2

Portmanteau testing inference in beta autoregressive moving average models

2.1 Introduction

The beta autoregressive moving average (βARMA) model (Rocha and Cribari-Neto, 2009) is a dynamic model based on the class of beta regression models (Ferrari and Cribari-Neto, 2004). In the βARMA model the variable of interest (y) is assumed to follow the beta law, its mean being impacted by a set of covariates and also subject to autoregressive and moving average dynamics. The model was developed to be used with time series that assume values in standard unit interval $(0,1)$, such as rates and proportions. Such variables are typically asymmetrically distributed and inferences based on the Gaussian assumption may be quite inaccurate. Additionally, a novel feature of the βARMA model is that it requires no data transformation; fitted values and out-of-sample forecasts will always belong to the standard unit interval.

Diagnostic analyses are of paramount importance in time series modeling. They are performed after the model has been identified and fitted. Different model validation strategies can be used. Perhaps the most commonly used validation strategy involves portmanteau testing inference (Wei, 1994). Such tests are based on statistics that use residual autocorrelations. They seek to detect any existing serial correlation in the residuals obtained from the fitted model.

Since the seminal article by Box and Pierce (1970), which introduced the first portmanteau test, considerable attention has been devoted to tests that use residual autocorrelations to assess goodness of fit. Goodness of fit assessment should not only determine whether there is evidence of lack of fit, but should also suggest ways in which the model may be modified when that proves necessary (Box and Pierce, 1970). Ljung and Box (1978) showed that a simple modification to the test statistic proposed by Box and Pierce (1970) considerably reduces the distribution location bias and improves the quality of the asymptotic approximation used in the test. Monti (1994) proposed to base portmanteau testing inference on a test statistic that uses residual partial autocorrelations rather than residual autocorrelations. The Ljung and Box and Monti tests are asymptotically equivalent under correct model specification, but their performances under the alternative hypothesis can be substantially different in small samples. Dufour and Roy (1986) introduced a non-parametric portmanteau test based on rank autocorrelations. Their test is particularly useful when the underlying distribution of the time series is unknown and tends to deliver accurate inferences under non-normality. Dufour and Roy (1986) noted that the distribution of their statistic is the same for all samples whenever the observations are continuous and exchangeable, irrespective of the distributional form. Kwan and Sim

(1996a) proposed three modified portmanteau test statistics that make use of the Fisher (1921) and Hotelling (1953) transformations to sample autocorrelations. Kwan and Sim (1996b) introduced a test statistic that uses the Jenkins (1954) variance-stabilizing transformation applied to residual autocorrelations. The simulations evidence presented by Kwan and Sim (1996a,b) showed that these tests they proposed typically deliver more accurate inferences than those of tests introduced by Ljung and Box (1978) and Dufour and Roy (1986). Peña and Rodriguez (2002) introduced a test based on the determinant of the residual autocorrelation matrix. They approximated the test statistic null distribution using the gamma distribution. Such an approximation can be poor in some situations. To circumvent such a shortcoming, Lin and McLeod (2006) proposed using bootstrap resampling to estimate the test statistic null distribution.

All portmanteau tests listed above were developed for standard ARMA models, i.e., for models used with variables that assume values in the real line. How do such tests perform when used with βARMA models? To the best of our knowledge, this question remains unanswered. Our chief goal in this dissertation is to investigate the accuracy of portmanteau testing inference in the class of βARMA models. We also consider bootstrap variants of the tests. Additionally, we introduce two new portmanteau test statistics which are obtained by modifying two test statistics proposed by Kwan and Sim (1996a,b) to use residual partial autocorrelations instead of ordinary residual autocorrelations. The simulation results we present show that such tests are typically more powerful than the other tests considered in this thesis.

We present an empirical application where we use portmanteau testing inference with a fitted βARMA model. The interest lies in modeling the proportion of stocked hydroelectric energy in the South of Brazil. The data range from January 2001 through October 2016. We produce out-of-sample forecasts using both the βARMA model and the standard ARMA model. It is shown that the former yields more accurate short term forecasts. Prior to using the fitted βARMA model for forecasting we validate it on the basis of portmanteau testing inference. We shall return to this application in Section 2.5.

This chapter unfolds as follows. Section 2.2 presents the βARMA model and its main properties. In Section 2.3 we review some portmanteau tests for model adequacy that have been used in the literature and propose two new tests. Monte Carlo simulation evidence is presented in Section 2.4. An empirical application is presented and discussed in Section 2.5. Finally, Section 2.6 contains some concluding remarks.

2.2 The βARMA model

Let $\mathbf{y} = (y_1, \ldots, y_n)^\top$ be a vector containing n random variables, where each y_t, for $t = 1, \ldots, n$, given the previous information set $\mathscr{F}_{t-1}$, where $\mathscr{F}_t$ is an increasing sequence of σ-fields, i.e., $\mathscr{F}_0 \subset \mathscr{F}_1 \subset \mathscr{F}_2 \ldots$, follows the beta law with conditional mean μ_t and precision parameter ϕ. The conditional density of y_t given $\mathscr{F}_{t-1}$ is

$$f(y_t \mid \mathscr{F}_{t-1}) = \frac{\Gamma(\phi)}{\Gamma(\mu_t\phi)\Gamma((1-\mu_t)\phi)} y_t^{\mu_t\phi-1}(1-y_t)^{(1-\mu_t)\phi-1}, \qquad 0 < y_t < 1, \tag{2.1}$$

$0 < \mu_t < 1$ and $\phi > 0$, where $\Gamma(\cdot)$ is the gamma function. The conditional mean and the conditional variance of the y_t are, respectively,

$$\mathbb{E}(y_t \mid \mathscr{F}_{t-1}) = \mu_t \quad \text{and} \quad \text{var}(y_t \mid \mathscr{F}_{t-1}) = V(\mu_t)/(1+\phi),$$

where $V(\mu_t) = \mu_t(1-\mu_t)$ is the variance function. Note that μ_t is the mean of y_t and ϕ is a precision parameter, in the sense that for a fixed μ_t the variance of y_t decreases as ϕ increases.

By assuming that the variable of interest follows the above beta law, Rocha and Cribari-Neto (2009) proposed the following βARMA(p,q) model:

$$g(\mu_t) = \alpha + x_t'\beta + \sum_{i=1}^{p} \varphi_i \left\{ g(y_{t-i}) - x_{t-i}'\beta \right\} + \sum_{j=1}^{q} \theta_j r_{t-j},$$

where $g : (0,1) \to \mathbb{R}$ is a link function which is strictly monotonic and twice differentiable. Here, $\alpha \in \mathbb{R}$ is a scalar parameter and $p, q \in \mathbb{N}$ are, respectively, the autoregressive and moving average orders. Additionally, r_t is the moving average error term and the φ's and the θ's are the autoregressive and moving average parameters, respectively. There are different specifications for r_t, e.g., (i) error on the original scale: $y_t - \mu_t$, and (ii) error on the predictor scale: $g(y_t) - \eta_t$. In what follows, we shall consider the latter.

The estimation of the parameters that index the βARMA model is typically performed by conditional maximum likelihood (Andersen, 1970). The conditional log-likelihood function (given the first K observations) is

$$\ell = \sum_{t=K+1}^{n} \log f(y_t \mid \mathscr{F}_{t-1}) = \sum_{t=K+1}^{n} \ell_t(\mu_t, \phi),$$

where $K = \max\{p,q\}$ and $f(y_t \mid \mathscr{F}_{t-1})$ is given in Equation (2.1).

The conditional maximum likelihood estimators of the model parameters cannot be expressed in closed form. They are typically computed by numerically maximizing the conditional log-likelihood function. In what follows we shall perform log-likelihood maximizations using the BFGS quasi-Newton nonlinear optimization algorithm; for details, see Press et al. (1992). When the model contains moving average components, it is necessary to take into account the recursive structure of log-likelihood derivatives (Benjamin et al., 1998). With predictor scale error, such derivatives are given by Rocha and Cribari-Neto (2017):

$$\frac{\partial \eta_t}{\partial \alpha} = 1 - \sum_{j=1}^{q} \theta_j \frac{\partial \eta_{t-j}}{\partial \alpha},$$

$$\frac{\partial \eta_t}{\partial \beta_l} = x_t' - \sum_{i=1}^{p} \varphi_i x_{t-i}' - \sum_{j=1}^{q} \theta_j \frac{\partial \eta_{t-j}}{\partial \beta},$$

$$\frac{\partial \eta_t}{\partial \varphi_i} = g(y_{t-i}) - x_{t-i}'\beta - \sum_{j=1}^{q} \theta_j \frac{\partial \eta_{t-j}}{\partial \varphi_i}, \quad i = 1, \ldots, p,$$

$$\frac{\partial \eta_t}{\partial \theta_l} = g(y_{t-l}) - \eta_{t-l} - \sum_{j=1}^{q} \theta_j \frac{\partial \eta_{t-j}}{\partial \theta_l}, \quad l = 1, \ldots, q.$$

Starting values for η_t can be obtained by setting $\eta_t = g(y_t)$ and the derivatives of η with respect to the model parameters equal to zero for $t = 1, \ldots, q$ (Benjamin et al., 1998).

The model residuals measure the distance between the observed responses and the fitted conditional means. In what follows we shall use the standardized ordinary residual defined by (Ferrari and Cribari-Neto, 2004)

$$\hat{r}_t = \frac{y_t - \hat{\mu}_t}{\sqrt{\widehat{\text{var}}(y_t)}},$$

where $\widehat{\text{var}}(y_t) = \hat{\mu}_t(1-\hat{\mu}_t)/(1+\hat{\phi})$. Here, $\hat{\mu}_t$ is obtained by evaluating μ_t at the maximum likelihood estimates and $\hat{\phi}$ is the maximum likelihood estimate of ϕ.

2.3 Portmanteau Tests

Portmanteau tests are commonly used in time series analysis to assess goodness of fit (Wei, 1994). Portmanteau test statistics are based on residual autocorrelations and the fitted model is taken as a good representation of the data when such autocorrelations are jointly negligible. According to Box et al. (2015), time series model building consists of three separate stages, namely: identification, estimation and validation. Portmanteau tests focus on the latter. Box and Pierce (1970) argue that model adequacy assessment should not only determine whether there is evidence of lack of fit, but also indicate ways in which the fitted model may be modified when that proves necessary.

The null and alternative hypotheses are

$$H_0 : \rho_1 = \rho_2 = \cdots = \rho_m = 0$$
$$H_1 : \text{at least one } \rho_i \neq 0.$$

Rejection of the null hypothesis indicates that there is evidence of model misspecification.

In what follows we shall briefly present some well known portmanteau tests (Subsection 2.3.1) and then propose two new tests (Subsection 2.3.2).

2.3.1 Standard portmanteau tests

The kth order residual (sample) autocorrelation is given by

$$\hat{\rho}_k = \frac{\sum_{t=k+1}^{n} \hat{r}_t \hat{r}_{t-k}}{\sum_{t=1}^{n} \hat{r}_t^2}, \quad k = 1, \ldots, m,$$

where $\hat{r}_t$ is the tth residual. When the model is correctly specified $\hat{r}_t$ is expected to be nearly uncorrelated.

Box and Pierce (1970) showed that when the model is correctly specified

$$\mathrm{var}(\hat{\rho}_k) = \frac{n-k}{n(n+2)} \approx \frac{1}{n}$$

and $\mathrm{Cov}(\hat{\rho}_k, \hat{\rho}_h) \approx 0$ for $(k \neq h)$. We carried out extensive numerical evaluations using different values of n and m, and their results indicate that such a also holds for βARMA models.

The authors then proposed the following portmanteau test statistic:

$$Q_{BP} = n \sum_{k=1}^{m} \hat{\rho}_k^2.$$

Box and Pierce (1970) proved that the asymptotic null distribution of their test statistic is χ^2_{m-p-q} (for $m > p-q$). The result follows from the asymptotic normality of the residual autocorrelations, which hold for a wide range of stochastic processes. Numerical evidence not presented here for brevity indicates that this is the case for time indexed beta-distributed random variables. The alternative test statistics considered in this thesis (see below) are asymptotically equivalent to that of Box and Pierce.

Ljung and Box (1978) showed that Q_{BP} suffers from a location bias of the null distribution of the test statistic, which happens even when the sample size is large $(n > 100)$ (Dufour and Roy, 1986). Ljung and Box (1978) then proposed the following test statistic:

$$Q_{LB} = n(n+2) \sum_{k=1}^{m} \frac{\hat{\rho}_k^2}{(n-k)}.$$

The authors showed that the null distribution of Q_{LB} is better approximated by the χ^2_{m-p-q} reference distribution than that of the Box and Pierce (1970) test statistic.

Monti (1994) proposed a portmanteau test statistic based on the sum of the square residual partial autocorrelations. Let $\hat{\pi}_k$ be the kth residual partial autocorrelation. The Monti test statistic can be written as

$$Q_M = n(n+2) \sum_{k=1}^{m} \frac{\hat{\pi}_k^2}{(n-k)}.$$

It is also asymptotically distributed as χ^2_{m-p-q} under the null hypothesis. Notice that the Monti test statistics is obtained by replacing residual autocorrelations with residual partial autocorrelations in the Ljung-Box test statistic. The test is particularly sensitive to model misspecification that involves the omission of moving average terms (Monti, 1994). Under the null hypothesis, Q_M and Q_{LB} are asymptotically equivalent, but their performances under the alternative hypothesis can be quite different. The Monte Carlo simulation evidence presented by Monti (1994) shows that the Q_M and Q_{LB} tests behave similarly when the null hypothesis is true and that the Q_M test tends to be more powerful than the Q_{LB} test when the fitted model fails to include important moving average terms.

Dufour and Roy (1986) introduced a non-parametric portmanteau test based on rank autocorrelations. Let R_t be the rank of $\hat{r}_t$. The kth residual rank autocorrelation is

$$\tilde{\rho}_k = \frac{\sum_{t=1}^{n-k}(R_t - \bar{R})(R_{t+k} - \bar{R})}{\sum_{t=1}^{n}(R_t - \bar{R})^2}, \qquad 1 \leq k \leq n-1,$$

where $\bar{R} = n^{-1}\sum_{t=1}^{n} R_t = (n+1)/2$ and $\sum_{t=1}^{n}(R-\bar{R})^2 = n(n^2-1)/12$ if all ranks are distinct. Since $R_1,\ldots,R_n$ are interchangeable when $r_1,\ldots,r_n$ are interchangeable and continuous, the mean of $\tilde{\rho}_k$ is

$$\mathbb{E}(\tilde{\rho}_k) = \mu_k = -\frac{(n-k)}{n(n-1)}, \qquad 1 \le k \le n-1.$$

It is possible to show that

$$\mathrm{var}(\tilde{\rho}_k) = \tilde{\sigma}_k^2 = \frac{5n^4-(5k+9)n^3+9(k-2)n^2+2k(5k+8)n+16k^2}{5(n-1)^2n^2(n+1)}, \qquad 1 \le k \le n-1.$$

Dufour and Roy (1986) proposed the following portmanteau test statistic:

$$Q_{DR2} = \sum_{k=1}^{m} \frac{(\tilde{\rho}_k-\mu_k)^2}{\tilde{\sigma}_k^2},$$

where $\tilde{\rho} = (\tilde{\rho}_1,\ldots,\tilde{\rho}_m)^\top$. Under the null hypothesis, Q_{DR2} is asymptotically distributed as χ^2_{m-p-q}.

Kwan and Sim (1996a) considered the situation where a portmanteau test is applied to a time series, and not to residuals from a fitted model. They showed that it is possible to reduce the dispersion bias of Q_{LB} by applying a variance-stabilizing transformation to the autocorrelations such as those introduced by Fisher (1921) and Hotelling (1953). Fisher (1921) proposed transforming of $\hat{\rho}_k$ as

$$z_{1k} = \frac{1}{2}\log\left\{\frac{(1+\hat{\rho}_k)}{(1-\hat{\rho}_k)}\right\}, \quad k = 1,\ldots,m,$$

whereas the two Hotelling (1953) transformations are

$$z_{2k} = z_{1k} - \frac{1}{4}\frac{(3z_{1k}+\hat{\rho}_k)}{(n-k)},$$

$$z_{3k} = z_{2k} - \frac{(23z_{1k}+33\hat{\rho}_k-5\hat{\rho}_k^3)}{96(n-k)^2},$$

where z_{ik}, for $i = 1,2,3,4$, are normally distributed with $\mathbb{E}(z_{ik}) \approx 0$, $\mathrm{var}(z_{1k}) \approx (n-k-3)^{-1}$, $\mathrm{var}(z_{2k}) \approx (n-k-1)^{-1}$ and $\mathrm{var}(z_{3k}) \approx (n-k-1)^{-1}$. Using these approximations, the authors proposed three modified portmanteau test statistics:

$$Q_{KWi} = \sum_{k=1}^{m}(n-k-\tau_i)z_{ik}^2, \quad i = 1,2,3,$$

where $\tau_1 = 3$ and $\tau_2 = \tau_3 = 1$. Under the null hypothesis, all three statistics are asymptotically distributed as χ^2_m. (The number of degrees of freedom is m because the test is, as noted above, applied to a time series, and not to residuals from a fitted model.) The Monte Carlo evidence in Kwan and Sim (1996a) indicates that the dispersion bias of the null distribution of Q_{LB} is considerably reduced when a variance-stabilizing transformation is used.

Kwan and Sim (1996b) introduced a fourth test statistic based on a variance-stabilizing transformation proposed by Jenkins (1954). The transformation is given by

$$z_{4k} = \sin^{-1}(\hat{\rho}_k),$$

where $\mathbb{E}(z_{4k}) \approx 0$ and $\text{var}(z_{4k}) \approx ((n-k)^{-2}(n-k-1))$. The new portmanteau test statistic is

$$Q_{KW4} = \sum_{k=1}^{m} \frac{(n-k)^2}{(n-k-1)} z_{4k}^2.$$

Under the null hypothesis, it is asymptotically distributed as χ^2_m. Simulation results reported by Kwan and Sim (1996b) show that the dispersion bias of Q_{LB} can be considerably reduced by applying the transformation suggested by Jenkins (1954).

When the sample size is large relative to m, it can be show that (Kwan and Sim, 1996a,b)

$$\mathbb{E}(Q_{KW1}) \simeq \mathbb{E}(Q_{KW2}) \simeq \mathbb{E}(Q_{KW3}) \simeq m - \frac{m(m+4)}{n}, \tag{2.2}$$

$$\mathbb{E}(Q_{KW4}) \simeq m - \frac{m(m+1)}{n}. \tag{2.3}$$

Kwan and Sim (1996a) noticed that (2.2) and (2.3) suggest that, for a fixed n, the means of Q_{KW1}, Q_{KW2}, Q_{KW3} and Q_{KW4} are smaller than m. Additionally, $\mathbb{E}(Q_{KW1})$, $\mathbb{E}(Q_{KW2})$, $\mathbb{E}(Q_{KW3})$ and $\mathbb{E}(Q_{KW4})$ are not integers. The authors then proposed to modify the tests critical values using

$$\mathbb{E}(Q_{KWi}) = \sum_{k=1}^{m} (n-k-\tau_i) \left\{ \mathbb{E}(\hat{\rho}_k^2) + \frac{2}{3}\mathbb{E}(\hat{\rho}_k^4) \right\}, \quad (i = 1,2,3), \tag{2.4}$$

$$\mathbb{E}(Q_{KW4}) = \sum_{k=1}^{m} \frac{(n-k)^2}{(n-k-1)} \left\{ \mathbb{E}(\hat{\rho}_k^2) + \frac{1}{3}\mathbb{E}(\hat{\rho}_k^4) \right\}. \tag{2.5}$$

Expressions for $\mathbb{E}(\hat{\rho}_k^2)$ and $\mathbb{E}(\hat{\rho}_k^4)$ in Equations (2.4) and (2.5) are given in Davies et al. (1977) and Ljung and Box (1978).

According to Kwan and Sim (1996a,b), the tests can be performed as follows:

1. Compute the first m residual autocorrelations: $\hat{\rho}_k (k = 1, \ldots, m)$.
2. Compute z_{1k}, z_{2k}, z_{3k} and z_{4k}.
3. Compute $\mathbb{E}(Q_{KW1})$, $\mathbb{E}(Q_{KW2})$, $\mathbb{E}(Q_{KW3})$ and $\mathbb{E}(Q_{KW4})$ using (2.4) and (2.5).
4. Reject the null hypothesis at the $\gamma \times 100\%$ significance level if $Q_{KW1} \geq \chi^2_{1-\gamma \mathbb{E}(Q_{KW1})}$, $Q_{KW2} \geq \chi^2_{1-\gamma, \mathbb{E}(Q_{KW2})}$, $Q_{KW3} \geq \chi^2_{1-\gamma, \mathbb{E}(Q_{KW3})}$ or $Q_{KW4} \geq \chi^2_{1-\gamma, \mathbb{E}(Q_{KW4})}$.

As noted above, Kwan and Sim (1996a,b) focused on testing whether a given time series behaves as white noise. In contrast, our interest lies in applying portmanteau tests to residuals from a fitted model. It is thus necessary to take into account the loss in degrees of freedom that follows from model fitting. We thus reject the null hypothesis if

$$Q_{KWi} \geq \chi^2_{1-\gamma,\mathbb{E}(Q_{KWi})-p-q}, \quad i = 1,2,3,4.$$

Notice the the critical value is now obtained from $\chi^2_{\mathbb{E}(Q_{KWi})-p-q}$ instead of $\chi^2_{\mathbb{E}(Q_{KWi})}$.

Peña and Rodriguez (2002) proposed a different portmanteau test statistic which is based on the determinant of the residual autocorrelation matrix, being given by

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

where

$$\hat{R}_m = \begin{pmatrix} 1 & \hat{\rho}_1 & \cdots & \hat{\rho}_m \\ \hat{\rho}_1 & 1 & \cdots & \hat{\rho}_{m-1} \\ \vdots & \cdots & \ddots & \vdots \\ \hat{\rho}_m & \cdots & \hat{\rho}_1 & 1 \end{pmatrix}.$$

Peña and Rodriguez (2002) noted that $|\hat{R}_m|$ is the estimated generalized variance of the standardized residuals. The authors proposed approximating the asymptotic null distribution of $\hat{D}_m$ by the gamma distribution with shape parameter

$$\lambda = \frac{3(m+1)[m-2(p+q)]^2}{2[2m(2m+1)-12(m+1)(p+q)]},$$

and scale parameter

$$r = \frac{3(m+1)[m-2(p+q)]}{2m(2m+1)-12(m+1)(p+q)}.$$

When m is small, however, it is possible to obtain $\lambda \leq 0$ and/or $r \leq 0$ which is numerically unfeasible. A modified test statistic was the proposed by the authors:

$$D_m = n\left(1 - |\ddot{R}_m|^{1/m}\right),$$

where $\ddot{R}_m$ denotes the residual autocorrelation matrix obtained by replacing $\hat{\rho}_k^2$ with $\ddot{\rho}_k^2$, where

$$\ddot{\rho}_k^2 = (n+2)(n-k)^{-1}\hat{\rho}_k^2.$$

They showed that the gamma approximation may not work well when the number of lags is small even for the modified test statistic. McLeod and Jimenez (1984) noted a serious shortcoming of D_m: $\ddot{R}_m$ is not always positive definite.

Lin and McLeod (2006) proposed using bootstrap resampling when performing the Peña and Rodriguez (2002) test; for details on the bootstrap method, see Davison and Hinkley (1997). Such a test can be adapted for βARMA models as follows:

1. Fit the $\beta\text{ARMA}(p,q)$ model and compute $\hat{D}_m$.

2. Select the number of bootstrap replications, B. (Typically, $100 \leq B \leq 1{,}000$.)

3. Simulate an $\beta\text{ARMA}(p,q)$ series after replacing the model's parameters by their corresponding estimates, fit the model using the simulated time series and compute $\hat{D}_m^*$.

4. Execute Step 3 B times and count the number of times k such that $\hat{D}_m^* \geq \hat{D}_m$.

5. Compute the bootstrap p-value: $(k+1)/(B+1)$.

6. Reject the null hypothesis if the bootstrap p-value is smaller than γ, the selected significance level.

We adapted all the portmanteau tests described above for the βARMA model. In the next subsection we shall propose two new portmanteau tests.

2.3.2 New portmanteau tests

We shall now propose two new portmanteau test statistics. Since they are based on two of the test statistics introduced by Kwan and Sim (1996a,b) we shall denote them by Q_{KW1p} and Q_{KW4p}.

As before, let $\hat{\pi}_k$ denote the kth residual partial autocorrelation. We shall explore the use of variance-stabilizing transformations applied to residual partial autocorrelations much in the same way Kwan and Sim (1996a,b) did it with standard residual autocorrelations. We shall them use such transformed partial autocorrelations to construct portmanteau test statistics much in the same way Monti (1994) did it with standard partial autocorrelations. At the outset, we consider the transformation introduced by Fisher (1921):

$$\hat{z}_{1k} = \frac{1}{2}\log\left\{\frac{(1+\hat{\pi}_k)}{(1-\hat{\pi}_k)}\right\}, \quad k = 1,\ldots,m.$$

The corresponding modified test statistic can be written as

$$Q_{KW1p} = \sum_{k=1}^{m}(n-k-\tau_1)\hat{z}_{1k}^2,$$

where $\tau_1 = 3$. Under the null hypothesis, Q_{KW1p} is asymptotically distributed as χ^2_{m-p-q}.

The second test statistic we propose makes use of the transformation introduced by Jenkins (1954), namely:

$$\hat{z}_{4k} = \sin^{-1}(\hat{\pi}_k).$$

Using it, we arrive at the following portmanteau test statistic:

$$Q_{KW4p} = \sum_{k=1}^{m} \frac{(n-k)^2}{(n-k-1)} \hat{z}^2_{4k}.$$

Under the null hypothesis, Q_{KW4p} follows the χ^2_{m-p-q} distribution asymptotically.

Following Kwan and Sim (1996a,b), we suggest that the test critical values be corrected using the approach outlined in the previous subsection. The numerical evidence we shall present in the next section shows that Q_{KW1p} and Q_{KW4p} are more powerful than the test proposed by Monti (1994) in βARMA models. Recall that the latter also uses partial autocorrelations.

The finite sample performances of the two tests presented above can be improved with the aid of bootstrap resampling. A similar approach can be used to other portmanteau tests. In what follows we shall use the bootstrap method to improve the finite sample performances of the following tests: Q_{LBb}, Q_{Mb}, Q_{DRb}, Q_{KW1b}, Q_{KW2b}, Q_{KW3b}, Q_{KW4b}, Q_{KW1pb} and Q_{KW4pb}. The bootstrap tests can be performed as follows:

1. Fit the βARMA(p,q) model and compute the test statistic of interest, Q.
2. Select the number of bootstrap replications, B. (Typically, $100 \leq B \leq 1{,}000$.)
3. Simulate an βARMA(p,q) series after replacing the model's parameters by the corresponding estimates, fit the model using the simulated time series and compute Q^*.
4. Execute Step 3 B times and count the number of times k such that $Q^* \geq Q$.
5. Compute the bootstrap p-value: $(k+1)/(B+1)$.
6. Reject the null hypothesis if the bootstrap p-value is smaller than γ, the test significance level.

2.4 Numerical evidence

Several simulation experiments were carried out to investigate the finite sample performances of the different portmanteau tests in the class of βARMA models. All simulations were performed using the R statistical computing environment (R Core Team, 2017). In the first set of simulations we examine the tests null rejection rates without resorting to bootstrap resampling. We consider the tests proposed by Ljung and Box (1978), Monti (1994), Dufour and Roy (1986), Kwan and Sim (1996a,b) and also the two portmanteau tests introduced in Subsection 2.3.2. All tests were performed at the 10%, 5% and 1% significance levels. In order to save space, however, we shall only present results for $\gamma = 5\%$. The sample sizes are

$n = 50, 250, 500$, the values of m we used are $m = 5, 10, 15, 20, 25$ and the number of Monte Carlo replications is $5,000$. All log-likelihood maximizations were performed using the BFGS quasi-Newton method with analytic first derivatives. Starting values for the parameters were selected as follows: (i) all moving average parameters were set equal to zero, and (ii) the values for the autoregressive parameters were selected by regressing $g(y_t)$ on $g(y_{t-1}), \ldots, g(y_{t-p})$ using ordinary least squares. There were no convergence failures. Beta random number generation was performed based on uniform random generation that used the Mersenne Twister algorithm. The hardware used was a cluster of computers with 64 blades of processing, 15.97 Tflops and 174TB RAM running on Linux and also a personal computer with an Intel i7 processor and 16GB RAM running the Linux operating system. The computing cluster we used belongs to the National Supercomputing Center at Federal University of Rio Grande do Sul (CESUP/UFRGS).

In the first part of the experiment we generate data from the βAR(1) model with the following values for the autoregressive parameter: $\varphi = 0.2, 0.5, 0.8$. The tests null rejection rates (%) are presented in Table 2.2. They reveal that the finite sample performances of the Q_{LB}, Q_M, Q_{DR}, Q_{KW1p} and Q_{KW4p} tests are highly sensitive to the choice of m in small samples. When the value of m is large relative to the sample size (e.g., $n = 50$ and $m = 25$) Q_{LB} and Q_{DR} are considerably liberal whereas Q_{KW1p} and Q_{KW4p} are conservative. When $\varphi = 0.8$, $n = 50$ and $m = 25$, the Q_{LB} and Q_{DR} null rejection reach 8.0% and 7.9%, respectively. This behavior had already been noted by Kwan et al. (2005) using normally distributed data. Even for $n = 50$, the Q_M null rejection rate decreases when m goes from 20 (4.4%) to 25 (2.8%). Kwan and Wu (1997) had noticed that the size of Q_M decreases whenever the value of m exceeds 14 and $\varphi \geq 0.7$, i.e., Q_M tends to reject the null hypothesis less frequently. Additionally, the null rejection rates of Q_{KW1p} and Q_{KW4p} decrease with m. Such tests become considerably undersized when m is large relative to n. It is noteworthy that tests whose statistics use partial autocorrelations (Q_M, Q_{KW1p} and Q_{KW4p}) behave similarly. The tests proposed by Kwan and Sim (1996a,b) (Q_{KW1}, Q_{KW2}, Q_{KW3}, and Q_{KW4}) display good control of the type I error probability even when m is large relative to n. All tests perform well when $n = 250$ and $n = 500$. For instance, when $n = 250$ and $m = 15$, the tests null rejection rates range from 5.1% to 5.8%.

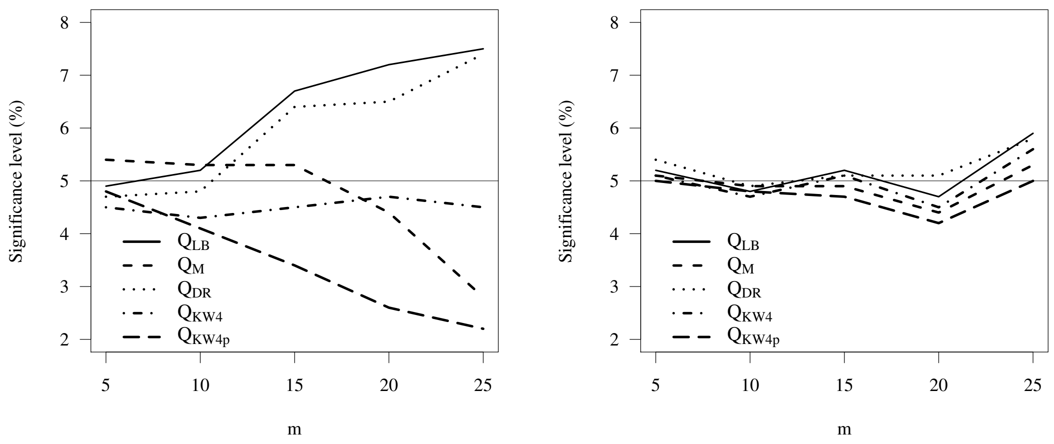
In order to compare the performances of the different tests for each sample ($n = 50, 250, 500$), we defined a measure which we call 'mean size distortion': $MSD = s^{-1}\sum_{i=1}^{s} |\text{nrr}_i - \gamma|$, where nrr_i is the test null rejection rate for the ith value of m. Since we use five different values for m, it follows that $s = 5$. Table 2.1 presents the MSD values for the tests considered in this dissertation. We note that, the Q_{KW4} test proposed by Kwan and Sim (1996b) displays the smallest MSD for all sample sizes when the data generating process is βAR(1). It is thus the most accurate as far as size distortions are concerned in the βAR(1) model. The test proposed by Ljung and Box (1978) is the worst performer.

In Figure 2.1 we plot the tests null rejection rates against m for two samples sizes ($n = 50, 500$) and five test statistics (Q_{LB}, Q_M, Q_{DR}, Q_{KW4} and Q_{KW4p}). Notice that when n is small (left panel), the null behavior of the Q_{LB}, Q_M, Q_{DR} tests are sensitive to the choice of m. In contrast, the null behavior of Q_{KW4} is nearly insensitive to the value of m used in the test. When the sample is large (right panel) all tests behave similarly.

Table 2.3 contains the tests null rejection rates (%) for the βMA(1) model with moving

Table 2.1 Mean size distortion rates of portmanteau tests without bootstrap resampling, $\gamma = 5\%$.

Test	βAR(1)			βMA(1)			βARMA(1,1), $\varphi = 0.2$			βARMA(1,1), $\varphi = 0.5$			βARMA(1,1), $\varphi = 0.8$		
	$n=50$	$n=250$	$n=500$	$n=50$	$n=250$	$n=500$	$n=50$	$n=250$	$n=500$	$n=50$	$n=250$	$n=500$	$n=50$	$n=250$	$n=500$
Q_{LB}	1.61	0.59	0.35	1.18	0.51	0.27	0.81	0.35	0.27	0.61	0.22	0.27	0.97	0.33	0.53
Q_M	0.80	0.31	0.33	0.83	0.39	0.23	1.28	0.26	0.28	0.99	0.25	0.29	1.27	0.32	0.60
Q_{DR}	1.39	0.61	0.36	1.07	0.55	0.37	0.81	0.42	0.35	0.71	0.39	0.46	1.00	0.50	0.65
Q_{KW1}	0.38	0.31	0.29	0.86	0.23	0.25	1.03	0.39	0.34	1.30	0.33	0.25	1.04	0.29	0.56
Q_{KW2}	0.38	0.31	0.29	0.85	0.23	0.25	1.03	0.39	0.34	1.30	0.33	0.25	1.04	0.29	0.56
Q_{KW3}	0.38	0.31	0.29	0.85	0.23	0.25	1.03	0.39	0.34	1.30	0.33	0.25	1.04	0.29	0.56
Q_{KW4}	0.30	0.30	0.28	0.77	0.23	0.24	0.91	0.41	0.35	1.30	0.33	0.26	0.98	0.28	0.54
Q_{KW1p}	1.57	0.32	0.29	1.67	0.42	0.33	1.67	0.43	0.25	1.89	0.33	0.37	1.95	0.53	0.90
Q_{KW4p}	1.53	0.31	0.34	1.73	0.40	0.33	1.57	0.23	0.23	1.83	0.31	0.37	1.93	0.53	0.90



(a) $n = 50$ (b) $n = 500$

Figure 2.1 Null rejection rates of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}, βAR(1) with $\varphi = 0.2$.

average parameter $\theta = 0.2, 0.5, 0.8$. They show again that the size distortions of Q_{LB}, Q_M, Q_{DR}, Q_{KW1p} and Q_{KW4p} are affected by the choice of m. The Q_{LB} test is the worst performer when $\theta = 0.8$, $n = 50$ and m is large relative to n ($m = 20, 25$). Again, when the sample is small ($n = 50$), the null rejection rates of Q_M decrease as m increases. The Q_{KW1p} and Q_{KW4p} tests display the largest *MSD* in small samples. Overall, the tests introduced by Kwan and Sim (1996a,b) are the best performers. When $n = 250$ and $n = 500$ all tests display reasonably good control of the type I error probability and behave similarly.

Table 2.4 presents results obtained using the βARMA(1,1) data generating process with $\varphi = 0.2$ and $\theta = 0.2, 0.5, 0.8$. Similarly to the previously results, all portmanteau tests yield reliable testing inferences when $n = 250$ or $n = 500$. When the sample size is small ($n = 50$) and $m = 5$, the tests performances are impacted by the moving average dynamics. In particular, Q_{KW1p} becomes considerably liberal, its null rejection reaching 9.1% when $\theta = 0.8$. The Q_{DR} test performs well when $\theta = 0.2$ and $\theta = 0.5$, being, however, a bit oversized when $\theta = 0.8$.

Table 2.5 shows the tests null rejection rates for the βARMA(1,1) model with autoregressive parameter $\varphi = 0.5$ and $\theta = 0.2, 0.5, 0.8$. When the sample size is small ($n = 50$) and $m = 5$ or $m = 10$, all tests are affected by the moving average dynamics and tend to reject the null hypothesis more frequently as the value of θ increases. The figures in Table 2.1 show that the Q_{KW1}, Q_{KW2}, Q_{KW3} and Q_{KW4} tests display good control of the type I error probability when the size sample is large ($n = 500$). As in the previous results, Q_{KW1p} and Q_{KW4p} display the largest *MSD* when the sample size is small.

Table 2.6 contains the tests null rejection rates for the βARMA$(1,1)$ data generating process with $\varphi = 0.8$ and $\theta = 0.2, 0.5, 0.8$. Again, the Q_M, Q_{KW1p} and Q_{KW4p} test were sensitive to changes in the value m, the worst performances taking place when $m = 25$ and $n = 50$. When $m = 5$ and $\theta = 0.8$, all tests are liberal. Hence, in a $\beta ARMA(1,1)$ model twith strong moving average dynamics the tests do not perform well when m is small and $n = 50$; see Tables 2.4 and 2.5.

When the model contains both autoregressive and moving averages parameters, the test proposed by Ljung and Box (1978) tends to display MSDs that are smaller than those of the competing tests, closely followed by the test introduced by Kwan and Sim (1996b). Q_{KW1p} is the worst performer.

The results presented so far indicate that portmanteau tests can be considerably size-distorted in small samples when used with βARMA models. The same happens with Gaussian-based models. As noted earlier, Lin and McLeod (2006) proposed using bootstrap resampling when performing the portmanteau test proposed by Peña and Rodriguez (2002) with $n < 1,000$. Following up on their proposal, we shall now investigate the effectiveness of bootstrap resampling when coupled with portmanteau tests in the class of βARMA models. We shall consider the following bootstrap tests: Q_{LBb}, Q_{Mb}, Q_{DRb}, Q_{KW1b}, Q_{KW2b}, Q_{KW3b}, Q_{KW4b}, Q_{KW1pb} and Q_{KW4pb}. Such tests are the bootstrap variants of Q_{LB}, Q_M, Q_{DR}, Q_{KW1}, Q_{KW2}, Q_{KW3}, Q_{KW4}, Q_{KW1p} and Q_{KW4p}, respectively. Since the Q_{KW1b}, Q_{KW2b} and Q_{KW3b} behave similarly, we shall only report results for Q_{KW1b}. All results are based on $1,000$ bootstrap samples, i.e., $B = 1,000$. The tests significance level is $\gamma = 5\%$ and the sample size is $n = 50$. We shall not report results for $n = 250, 500$ because in large samples the bootstrap tests behave similarly to the corresponding standard tests.

Table 2.8 contains the null rejection rates of the bootstrap portmanteau tests, including the bootstrap variant of the Peña-Rodriguez (Lin and McLeod, 2006) for the βAR(1) model with $\varphi = 0.2, 0.5, 0.8$. It is noteworthy that all tests are now nearly size distortion free, i.e., their null rejection rates are quite close to the selected significance level (5%). It is thus clear the importance of using bootstrap resampling when performing portmanteau testing inferences in small samples.

From the results presented in Table 2.7, we also note that the Q_{DRb} and Q_{KW4pb} tests display the smallest MSDs, $\hat{D}_m$ being the worst performer.

Table 2.9 presents the bootstrap tests null rejection rates obtained using the βMA(1) model with $\theta = 0.2, 0.5, 0.8$ for data generation. Again, all tests yield accurate inferences and are nearly size distortion-free. In particular, the Dufour-Roy (Q_{DR}) becomes considerably more accurate when couple with a bootstrap resampling scheme when the value of θ is large ($\theta = 0.8$). We note that Q_{DRb} has the smallest MSD value, Q_M being the worst performer.

A second set of Monte Carlo simulations was carried out to evaluate the tests nonnull behavior, i.e., to evaluate the tests powers. Data generation is now carried out under the alternative hypothesis and the interest lies in measuring the tests ability to detect that the model specifi-

Table 2.2 Null rejection rates of portmanteau tests, $\beta\mathrm{AR}(1)$ model, $\gamma = 5\%$.

φ	Test	$n=50$					$n=250$					$n=500$				
		$m=5$	$m=10$	$m=15$	$m=20$	$m=25$	$m=5$	$m=10$	$m=15$	$m=20$	$m=25$	$m=5$	$m=10$	$m=15$	$m=20$	$m=25$
	Q_{LB}	4.9	5.2	6.7	7.2	7.5	4.5	5.2	5.6	5.3	6.0	5.2	4.8	5.2	4.7	5.9
	Q_M	5.4	5.3	5.3	4.4	2.8	4.6	5.3	5.7	5.0	5.4	5.1	4.9	4.9	4.4	5.3
	Q_{DR}	4.7	4.8	6.4	6.5	7.4	4.9	5.5	5.6	5.8	6.4	5.4	4.9	5.1	5.1	5.8
	Q_{KW1}	4.5	4.2	4.5	4.5	4.3	4.4	4.9	5.1	4.5	5.2	5.1	4.7	5.1	4.5	5.5
0.2	Q_{KW2}	4.5	4.2	4.5	4.5	4.3	4.4	4.9	5.1	4.5	5.2	5.1	4.7	5.1	4.5	5.5
	Q_{KW3}	4.5	4.2	4.5	4.5	4.3	4.4	4.9	5.1	4.5	5.2	5.1	4.7	5.1	4.5	5.5
	Q_{KW4}	4.5	4.3	4.5	4.7	4.5	4.4	4.9	5.1	4.6	5.2	5.1	4.7	5.1	4.5	5.6
	Q_{KW1p}	5.0	4.1	3.3	2.5	2.2	4.6	4.9	5.3	4.5	4.7	5.0	4.8	4.7	4.2	4.9
	Q_{KW4p}	4.8	4.1	3.4	2.6	2.2	4.6	5.0	5.3	4.6	4.7	5.0	4.8	4.7	4.2	5.0
	Q_{LB}	5.3	5.6	6.9	7.5	7.5	4.8	4.9	5.5	6.0	5.5	5.1	4.8	5.6	4.9	5.9
	Q_M	5.4	5.5	5.3	4.4	2.8	4.7	4.8	5.5	5.6	5.0	5.3	4.8	5.4	4.4	5.7
	Q_{DR}	5.0	5.8	7.1	6.8	7.3	5.0	4.7	5.8	5.7	5.8	5.5	4.7	5.4	5.1	5.7
	Q_{KW1}	4.7	4.5	4.7	4.8	4.7	4.7	4.7	5.1	5.3	4.9	5.1	4.6	5.4	4.6	5.6
0.5	Q_{KW2}	4.7	4.5	4.7	4.8	4.7	4.7	4.7	5.1	5.3	4.9	5.1	4.6	5.4	4.6	5.6
	Q_{KW3}	4.7	4.5	4.7	4.8	4.7	4.7	4.7	5.1	5.3	4.9	5.1	4.6	5.4	4.6	5.6
	Q_{KW4}	4.8	4.6	4.8	5.0	5.0	4.7	4.8	5.1	5.4	4.9	5.1	4.7	5.4	4.6	5.6
	Q_{KW1p}	5.1	4.2	3.3	2.7	2.2	4.6	4.6	5.1	4.7	4.3	5.3	4.7	5.1	4.1	5.3
	Q_{KW4p}	5.1	4.3	3.5	2.7	2.3	4.7	4.6	5.1	4.7	4.3	5.3	4.7	5.2	4.1	5.3
	Q_{LB}	5.5	6.4	7.0	7.8	8.0	4.6	5.4	5.7	6.3	6.2	5.3	5.3	4.8	5.2	5.5
	Q_M	5.3	5.4	4.6	4.2	2.7	4.7	5.4	5.1	5.3	5.2	5.3	5.1	4.4	4.7	5.3
	Q_{DR}	5.5	6.0	6.5	7.1	7.9	4.9	5.6	5.6	6.2	5.6	5.7	5.1	5.4	5.3	5.4
	Q_{KW1}	5.1	4.9	5.2	5.2	4.5	4.6	5.3	5.3	5.6	5.4	5.3	5.2	4.7	5.0	5.1
0.8	Q_{KW2}	5.1	4.9	5.2	5.2	4.5	4.6	5.3	5.3	5.6	5.4	5.3	5.2	4.7	5.0	5.1
	Q_{KW3}	5.1	4.9	5.2	5.2	4.5	4.6	5.3	5.3	5.6	5.4	5.3	5.2	4.7	5.0	5.1
	Q_{KW4}	5.1	5.1	5.3	5.4	4.7	4.6	5.3	5.3	5.6	5.4	5.3	5.2	4.7	5.0	5.1
	Q_{KW1p}	4.9	4.1	3.2	2.9	1.9	4.6	5.0	4.8	4.8	4.5	5.3	5.0	4.2	4.5	4.8
	Q_{KW4p}	4.9	4.2	3.3	3.0	1.9	4.6	5.1	4.8	4.8	4.5	5.3	5.0	4.2	4.5	4.8

Table 2.3 Null rejection rates of portmanteau tests for a βMA(1) model; $\gamma = 5\%$.

θ	Test	$n=50$					$n=250$					$n=500$				
		$m=5$	$m=10$	$m=15$	$m=20$	$m=25$	$m=5$	$m=10$	$m=15$	$m=20$	$m=25$	$m=5$	$m=10$	$m=15$	$m=20$	$m=25$
	Q_{LB}	4.3	5.0	5.7	6.3	6.9	5.3	5.8	5.2	5.6	5.8	4.9	4.8	5.4	5.1	5.6
	Q_{M}	4.9	5.4	4.4	3.8	2.8	5.8	5.9	4.9	5.3	5.1	5.0	4.9	4.9	4.7	5.2
	Q_{DR}	4.4	5.0	5.1	6.1	6.4	5.3	5.8	5.6	5.6	5.8	4.8	4.9	5.7	5.4	5.9
	Q_{KW1}	3.8	3.6	3.7	3.8	3.7	5.2	5.5	4.8	5.1	5.1	4.9	4.6	5.1	4.9	5.3
0.2	Q_{KW2}	3.8	3.6	3.7	3.8	3.7	5.2	5.5	4.8	5.1	5.1	4.9	4.6	5.1	4.9	5.3
	Q_{KW3}	3.8	3.6	3.7	3.8	3.7	5.2	5.5	4.8	5.1	5.1	4.9	4.6	5.1	4.9	5.3
	Q_{KW4}	3.8	3.8	3.7	4.0	3.8	5.2	5.5	4.8	5.1	5.1	4.9	4.6	5.1	4.9	5.2
	Q_{KW1p}	4.7	3.9	2.9	1.9	2.7	5.7	5.7	4.5	4.6	4.6	5.0	4.8	4.6	4.6	4.9
	Q_{KW4p}	4.6	3.9	2.9	2.0	1.8	5.7	5.7	4.5	4.7	4.6	5.0	4.8	4.6	4.6	4.9
	Q_{LB}	4.3	4.8	5.9	6.6	6.9	4.9	5.4	4.7	5.5	5.9	5.0	5.0	4.8	5.2	4.9
	Q_{M}	5.2	5.0	5.2	4.3	2.7	5.1	5.3	4.3	5.2	5.4	5.2	4.7	5.0	5.1	4.7
	Q_{DR}	4.7	5.0	6.1	6.3	6.7	5.2	5.5	4.9	5.0	6.0	5.3	5.3	5.2	5.0	5.1
	Q_{KW1}	4.0	3.6	4.4	4.0	4.0	4.7	5.2	4.2	4.9	5.1	4.9	4.9	4.7	4.9	4.5
0.5	Q_{KW2}	4.0	3.6	4.5	4.0	4.0	4.7	5.2	4.2	4.9	5.1	4.9	4.9	4.7	4.9	4.5
	Q_{KW3}	4.0	3.6	4.5	4.0	4.0	4.7	5.2	4.2	4.9	5.1	4.9	4.9	4.7	4.9	4.5
	Q_{KW4}	4.0	3.8	4.5	4.3	4.2	4.7	5.2	4.2	4.9	5.2	4.9	4.9	4.7	4.9	4.5
	Q_{KW1p}	4.8	4.0	3.6	2.4	2.0	5.1	5.1	3.9	4.6	4.5	5.1	4.6	4.7	4.7	4.3
	Q_{KW4p}	4.8	3.9	3.5	2.5	2.1	5.0	5.1	3.9	4.6	4.5	5.1	4.6	4.7	4.7	4.3
	Q_{LB}	5.2	5.2	6.3	7.7	8.4	5.2	5.5	5.5	5.4	6.1	5.6	4.5	5.2	5.5	5.4
	Q_{M}	6.5	5.0	5.0	4.3	2.7	5.2	5.3	5.5	4.5	5.4	5.4	4.5	4.8	5.4	4.7
	Q_{DR}	5.7	5.7	6.7	7.5	7.8	5.6	5.1	5.8	5.4	6.5	5.6	5.0	5.2	5.7	5.9
	Q_{KW1}	4.9	4.3	4.8	5.1	5.4	5.1	5.1	5.1	4.8	5.3	5.5	4.4	4.9	5.3	4.9
0.8	Q_{KW2}	4.9	4.3	4.8	5.1	5.4	5.1	5.1	5.1	4.8	5.3	5.5	4.4	4.9	5.3	4.9
	Q_{KW3}	4.9	4.3	4.8	5.1	5.4	5.1	5.1	5.1	4.8	5.3	5.5	4.4	4.9	5.3	4.9
	Q_{KW4}	5.0	4.4	4.9	5.2	5.6	5.1	5.1	5.1	4.8	5.3	5.6	4.4	4.9	5.2	4.9
	Q_{KW1p}	6.4	4.3	3.4	3.2	2.5	5.2	5.0	5.0	4.2	4.6	5.3	4.4	4.6	5.1	4.4
	Q_{KW4p}	6.4	4.3	3.6	3.2	2.4	5.2	5.0	5.0	4.2	4.7	5.3	4.4	4.6	5.1	4.4

Table 2.4 Null rejection rates of portmanteau tests, βARMA$(1,1)$ model, $\varphi = 0.2$, $\gamma = 5\%$.

θ	Test	$n=50$					$n=250$					$n=500$				
		$m=5$	$m=10$	$m=15$	$m=20$	$m=25$	$m=5$	$m=10$	$m=15$	$m=20$	$m=25$	$m=5$	$m=10$	$m=15$	$m=20$	$m=25$
	Q_{LB}	5.2	4.6	5.6	5.4	5.9	4.8	4.9	5.0	5.1	5.7	5.1	4.6	4.5	4.6	5.2
	Q_{M}	6.5	5.7	5.5	4.3	2.8	5.0	5.1	5.3	5.1	5.2	5.3	4.7	4.7	4.7	5.1
	Q_{DR}	5.3	4.5	5.2	5.2	5.8	5.2	5.1	4.9	5.6	5.6	5.3	4.6	4.8	5.1	5.3
	Q_{KW1}	5.5	3.6	4.2	3.7	3.5	4.8	4.7	4.7	4.5	4.7	5.1	4.4	4.4	4.3	4.9
0.2	Q_{KW2}	5.5	3.6	4.2	3.7	3.5	4.8	4.7	4.7	4.5	4.7	5.1	4.4	4.4	4.3	4.9
	Q_{KW3}	5.5	3.6	4.2	3.7	3.5	4.8	4.7	4.7	4.5	4.7	5.1	4.4	4.4	4.3	4.9
	Q_{KW4}	5.1	3.5	4.2	3.8	3.8	4.7	4.7	4.6	4.5	4.8	5.1	4.4	4.3	4.3	4.9
	Q_{KW1p}	7.5	5.3	4.3	3.4	2.7	5.0	5.5	5.3	4.9	4.9	5.4	4.9	5.1	4.9	5.2
	Q_{KW4p}	6.9	5.2	4.2	3.4	2.7	5.0	5.5	5.2	4.9	4.9	5.4	4.9	5.1	4.8	5.2
	Q_{LB}	5.4	4.5	5.0	6.0	5.2	4.8	5.0	4.4	4.6	5.4	4.4	5.2	4.9	4.8	5.5
	Q_{M}	7.0	5.7	5.4	4.2	3.1	5.1	5.1	4.8	4.6	5.2	4.6	5.3	5.5	5.2	4.7
	Q_{DR}	5.4	4.3	5.2	6.1	5.3	5.2	5.3	4.6	4.7	5.6	5.1	5.5	5.1	5.6	5.7
	Q_{KW1}	5.8	3.7	3.8	3.9	3.2	4.8	4.8	4.1	4.2	4.5	4.4	5.1	4.9	4.6	5.1
0.5	Q_{KW2}	5.8	3.7	3.8	3.9	3.2	4.8	4.8	4.1	4.2	4.5	4.4	5.1	4.9	4.6	5.1
	Q_{KW3}	5.8	3.7	3.8	3.9	3.2	4.8	4.8	4.1	4.2	4.5	4.4	5.1	4.9	4.6	5.1
	Q_{KW4}	5.3	3.9	3.8	4.0	3.1	4.8	4.8	4.1	4.2	4.4	4.4	5.1	4.9	4.6	5.1
	Q_{KW1p}	6.5	4.5	3.6	2.6	2.4	5.1	5.4	5.0	4.7	4.8	5.5	4.9	5.3	5.2	5.3
	Q_{KW4p}	6.1	4.4	3.6	2.7	2.3	5.0	5.3	5.0	4.7	4.9	5.4	4.9	5.2	5.2	5.3
	Q_{LB}	6.9	5.9	6.0	6.3	7.4	5.8	4.7	5.5	5.2	5.7	5.5	5.0	5.1	4.9	4.9
	Q_{M}	8.5	6.7	5.6	4.5	3.5	6.1	5.0	5.6	5.0	5.5	5.6	4.9	5.3	5.1	4.9
	Q_{DR}	6.7	5.7	6.4	6.2	7.5	6.2	5.1	5.6	5.7	5.3	5.8	5.2	5.8	5.2	5.0
	Q_{KW1}	7.4	4.8	4.5	4.4	5.0	5.9	4.5	5.0	4.7	5.0	5.5	4.8	4.8	4.7	4.5
0.8	Q_{KW2}	7.4	4.8	4.5	4.4	5.0	5.9	4.5	5.0	4.7	5.0	5.5	4.8	4.8	4.7	4.5
	Q_{KW3}	7.4	4.8	4.5	4.4	5.0	5.9	4.5	5.0	4.7	5.0	5.5	4.8	4.8	4.7	4.5
	Q_{KW4}	7.0	4.7	4.5	4.5	5.1	5.9	4.5	5.0	4.7	5.0	5.5	4.8	4.8	4.7	4.5
	Q_{KW1p}	9.1	5.9	4.3	3.5	2.9	5.7	5.5	5.0	4.6	4.6	5.9	5.4	5.0	5.0	5.1
	Q_{KW4p}	8.4	5.8	4.3	3.5	2.8	5.7	5.4	5.0	4.7	4.5	5.8	5.4	5.0	5.0	5.1

Table 2.5 Null rejection rates of portmanteau tests, βARMA$(1,1)$ model, $\varphi = 0.5$, $\gamma = 5\%$.

θ	Test	$n=50$					$n=250$					$n=500$				
		$m=5$	$m=10$	$m=15$	$m=20$	$m=25$	$m=5$	$m=10$	$m=15$	$m=20$	$m=25$	$m=5$	$m=10$	$m=15$	$m=20$	$m=25$
	Q_{LB}	3.9	3.9	4.7	4.6	5.3	5.0	4.6	5.0	5.0	5.1	4.9	5.5	4.6	5.2	5.6
	Q_M	5.0	5.3	4.7	3.9	3.2	5.2	5.1	4.8	5.0	5.0	4.8	5.6	5.1	5.3	5.7
	Q_{DR}	4.5	4.0	4.6	5.3	5.5	4.9	5.0	5.3	5.5	5.3	5.3	5.2	5.1	5.4	5.8
	Q_{KW1}	4.3	3.2	3.3	2.9	2.8	5.0	4.5	4.5	4.6	4.3	4.9	5.3	4.5	4.9	5.2
0.2	Q_{KW2}	4.3	3.2	3.3	2.9	2.8	5.0	4.5	4.5	4.6	4.3	4.9	5.3	4.5	4.9	5.2
	Q_{KW3}	4.3	3.2	3.3	2.9	2.8	5.0	4.5	4.5	4.6	4.3	4.9	5.3	4.5	4.9	5.2
	Q_{KW4}	3.9	3.0	3.3	2.9	2.8	4.9	4.4	4.5	4.6	4.3	4.8	5.3	4.5	4.9	5.2
	Q_{KW1p}	5.9	4.0	2.9	2.2	1.9	5.1	5.3	4.7	5.0	4.5	5.0	5.2	4.6	4.6	4.5
	Q_{KW4p}	5.5	3.9	2.8	2.2	1.9	5.1	5.3	4.7	5.0	4.5	4.9	5.2	4.7	4.6	4.5
	Q_{LB}	5.0	5.0	4.8	5.0	5.4	4.6	5.1	5.0	4.9	5.6	5.1	5.1	4.5	5.2	5.5
	Q_M	6.0	5.4	5.1	3.9	2.5	4.9	5.4	5.4	4.5	5.3	5.1	5.4	4.9	5.4	5.5
	Q_{DR}	5.2	4.7	4.7	5.1	5.4	5.3	5.2	5.4	5.1	6.1	5.5	5.2	4.9	5.9	5.8
	Q_{KW1}	5.2	4.3	3.3	2.8	3.3	4.6	4.9	4.7	4.3	4.8	5.1	5.1	4.4	5.0	5.2
0.5	Q_{KW2}	5.2	4.3	3.3	2.8	3.3	4.6	4.9	4.7	4.3	4.8	5.1	5.1	4.4	5.0	5.2
	Q_{KW3}	5.2	4.3	3.3	2.8	3.3	4.6	4.9	4.7	4.3	4.8	5.1	5.1	4.4	5.0	5.2
	Q_{KW4}	4.8	4.2	3.4	2.9	3.3	4.6	4.9	4.7	4.3	4.9	5.0	5.1	4.4	4.9	5.2
	Q_{KW1p}	6.0	4.0	2.9	2.4	1.9	5.2	5.1	4.5	4.7	4.4	4.9	5.3	4.6	4.6	4.4
	Q_{KW4p}	5.4	3.8	2.9	2.4	1.8	5.1	5.0	4.5	4.7	4.4	4.9	5.3	4.6	4.6	4.4
	Q_{LB}	6.5	5.3	5.7	6.2	6.6	5.5	5.1	5.2	5.2	5.6	5.3	4.5	5.0	5.0	5.1
	Q_M	7.2	5.7	4.8	3.8	3.0	5.5	5.5	5.2	5.1	5.3	5.4	5.1	5.1	5.0	4.7
	Q_{DR}	6.7	5.5	6.3	6.4	6.7	5.8	5.3	5.5	5.2	5.7	6.3	5.4	5.3	5.5	5.1
	Q_{KW1}	7.0	4.7	4.3	4.1	4.4	5.6	4.9	4.8	4.8	5.0	5.3	4.4	4.8	4.8	4.7
0.8	Q_{KW2}	7.0	4.7	4.3	4.1	4.4	5.6	4.9	4.8	4.8	5.0	5.3	4.4	4.8	4.8	4.7
	Q_{KW3}	7.0	4.7	4.3	4.1	4.4	5.6	4.9	4.8	4.8	5.0	5.3	4.4	4.8	4.8	4.7
	Q_{KW4}	6.6	4.6	4.4	4.2	4.5	5.5	4.8	4.8	4.8	5.0	5.3	4.4	4.8	4.8	4.7
	Q_{KW1p}	7.9	5.2	3.8	3.0	2.6	5.7	5.1	4.7	4.8	4.3	5.5	5.5	4.8	4.7	4.2
	Q_{KW4p}	7.3	5.3	3.9	3.0	2.5	5.7	5.1	4.7	4.8	4.3	5.5	5.5	4.8	4.7	4.2

Table 2.6 Null rejection rates of portmanteau tests, βARMA$(1,1)$ model, $\varphi = 0.8$, $\gamma = 5\%$.

θ	Test	$n=50$					$n=250$					$n=500$				
		$m=5$	$m=10$	$m=15$	$m=20$	$m=25$	$m=5$	$m=10$	$m=15$	$m=20$	$m=25$	$m=5$	$m=10$	$m=15$	$m=20$	$m=25$
	Q_{LB}	4.1	5.0	5.3	4.8	5.3	4.6	5.2	4.7	5.0	5.5	4.8	5.4	5.9	4.6	5.2
	Q_M	5.4	5.5	5.1	3.6	2.4	5.0	5.1	4.5	4.5	5.4	5.0	5.2	5.8	5.0	5.1
	Q_{DR}	5.0	5.1	5.4	5.3	4.9	4.9	5.4	5.0	5.2	5.6	5.3	5.8	6.2	5.0	5.3
	Q_{KW1}	4.3	4.0	3.5	2.7	3.1	4.6	4.8	4.3	4.4	5.0	4.8	5.2	5.6	4.4	4.8
0.2	Q_{KW2}	4.3	4.0	3.5	2.7	3.1	4.6	4.8	4.3	4.4	5.0	4.8	5.2	5.6	4.4	4.8
	Q_{KW3}	4.3	4.0	3.5	2.7	3.1	4.6	4.8	4.3	4.4	5.0	4.8	5.2	5.6	4.4	4.8
	Q_{KW4}	4.0	4.0	3.6	2.8	3.2	4.6	4.8	4.3	4.4	5.0	4.8	5.2	5.6	4.4	4.8
	Q_{KW1p}	5.1	3.8	2.8	2.1	1.8	5.5	5.0	4.5	4.7	4.3	5.3	4.8	4.9	4.8	4.8
	Q_{KW4p}	4.7	3.8	2.7	2.1	1.7	5.5	5.0	4.5	4.8	4.3	5.3	4.8	4.9	4.8	4.8
	Q_{LB}	6.9	5.9	6.0	6.3	7.4	5.7	5.2	5.4	4.9	5.5	5.5	4.5	4.7	4.9	5.1
	Q_M	8.5	6.7	5.6	4.5	3.5	5.5	5.2	5.4	4.9	4.5	5.5	4.9	4.7	4.8	4.8
	Q_{DR}	6.7	5.7	6.4	6.2	7.5	6.2	5.5	5.8	5.3	5.9	5.9	5.1	5.3	5.1	5.3
	Q_{KW1}	7.4	4.8	4.5	4.4	5.0	5.7	5.1	5.0	4.7	5.2	5.4	4.4	4.6	4.7	4.9
0.5	Q_{KW2}	7.4	4.8	4.5	4.4	5.0	5.7	5.1	5.0	4.7	5.2	5.4	4.4	4.6	4.7	4.9
	Q_{KW3}	7.4	4.8	4.5	4.4	5.0	5.7	5.1	5.0	4.7	5.2	5.4	4.4	4.6	4.7	4.9
	Q_{KW4}	7.0	4.7	4.5	4.5	5.1	5.6	5.1	5.0	4.7	5.2	5.4	4.4	4.6	4.6	4.9
	Q_{KW1p}	6.0	4.0	2.9	2.0	1.6	5.7	4.7	4.5	4.5	4.3	5.1	4.8	4.7	4.5	4.7
	Q_{KW4p}	5.3	3.9	2.8	1.9	1.5	5.6	4.7	4.4	4.5	4.3	5.1	4.8	4.7	4.5	4.7
	Q_{LB}	6.5	5.3	5.7	6.2	6.6	5.5	5.1	5.2	5.2	5.6	5.3	4.5	5.0	5.0	5.1
	Q_M	7.2	5.7	4.8	3.8	3.0	5.5	5.5	5.2	5.1	5.3	5.4	5.1	5.1	5.0	4.7
	Q_{DR}	6.7	5.5	6.3	6.4	6.7	5.8	5.3	5.5	5.2	5.7	6.3	5.4	5.3	5.5	5.1
	Q_{KW1}	7.0	4.7	4.3	4.1	4.4	5.6	4.9	4.8	4.8	5.0	5.3	4.4	4.8	4.8	4.7
0.8	Q_{KW2}	7.0	4.7	4.3	4.1	4.4	5.6	4.9	4.8	4.8	5.0	5.3	4.4	4.8	4.8	4.7
	Q_{KW3}	7.0	4.7	4.3	4.1	4.4	5.6	4.9	4.8	4.8	5.0	5.3	4.4	4.8	4.8	4.7
	Q_{KW4}	6.6	4.6	4.4	4.2	4.5	5.5	4.8	4.8	4.8	5.0	5.3	4.4	4.8	4.8	4.7
	Q_{KW1p}	8.3	5.4	4.1	2.9	2.5	6.0	4.7	4.4	4.4	4.2	5.3	5.1	5.1	5.1	5.0
	Q_{KW4p}	8.0	5.3	4.1	3.0	2.5	6.0	4.7	4.4	4.4	4.2	5.3	5.1	5.1	5.1	5.0

Table 2.7 Mean size distortion (MSD) rates of bootstrap portmanteau tests, $\gamma = 5\%$.

Test	$n = 50$					
	βAR(1)			βMA(1)		
	$\varphi = 0.2$	$\varphi = 0.5$	$\varphi = 0.8$	$\theta = 0.2$	$\theta = 0.5$	$\theta = 0.8$
$\hat{D}_m$	0.20	0.28	0.30	0.26	0.20	0.10
Q_{LBb}	0.10	0.20	0.18	0.06	0.18	0.38
Q_{Mb}	0.12	0.22	0.18	0.16	0.26	0.24
Q_{DRb}	0.22	0.10	0.08	0.16	0.24	0.08
Q_{KW1b}	0.18	0.20	0.08	0.14	0.14	0.34
Q_{KW4b}	0.18	0.20	0.08	0.14	0.14	0.32
Q_{KW1pb}	0.14	0.20	0.12	0.14	0.22	0.20
Q_{KW4pb}	0.14	0.18	0.12	0.14	0.24	0.16

Table 2.8 Null rejection rates of bootstrap portmanteau tests, βAR(1) model, $\gamma = 5\%$.

Test	$n = 50$														
	$\varphi = 0.2$					$\varphi = 0.5$					$\varphi = 0.8$				
	$m = 5$	$m = 10$	$m = 15$	$m = 20$	$m = 25$	$m = 5$	$m = 10$	$m = 15$	$m = 20$	$m = 25$	$m = 5$	$m = 10$	$m = 15$	$m = 20$	$m = 25$
$\hat{D}_m$	4.8	4.9	5.3	5.4	5.0	4.1	5.0	5.1	5.0	4.6	4.7	4.8	4.6	4.6	4.8
Q_{LBb}	5.0	5.0	4.8	4.8	4.9	4.7	5.0	5.0	5.5	5.2	5.1	4.8	4.9	4.7	4.8
Q_{Mb}	5.0	4.9	5.2	5.2	5.1	4.6	5.2	5.0	5.0	5.5	5.1	5.2	4.7	5.1	4.8
Q_{DRb}	5.3	4.8	5.0	4.6	4.8	5.1	5.1	5.1	4.8	5.0	5.0	4.9	4.7	5.0	5.0
Q_{KW1b}	5.0	5.0	4.7	4.5	4.9	4.7	5.0	5.1	5.3	5.3	5.0	4.9	4.7	5.0	5.0
Q_{KW4b}	5.0	5.0	4.7	4.6	4.8	4.7	5.0	5.1	5.3	5.3	5.0	4.9	4.7	5.0	5.0
Q_{KW1pb}	5.1	4.9	5.2	4.9	5.2	4.6	4.9	5.2	5.0	5.3	4.9	5.0	4.8	5.1	5.2
Q_{KW4pb}	5.1	4.9	5.2	5.0	5.3	4.6	4.9	5.2	5.0	5.2	4.9	5.0	4.9	5.2	5.2

cation is in error. The true data generating process is βARMA$(1,1)$, and the fitted model is βAR(1). The sample sizes are $n = 50, 250$, the values of m range from 3 to 25, and all results are based on $5,000$ Monte Carlo replications. Since the tests proposed by Kwan and Sim (1996a,b) behave similarly, we shall only report results on Q_{KW4}. The Q_{KW1p} and Q_{KW4p} proposed in this paper also behave similarly, and for that reason we shall only consider Q_{KW4p}. Since some of the tests are liberal, we shall base all tests on exact (estimated from the size simulations) critical values rather than on asymptotic critical values. By doing so, we force all tests to have correct size.

Figure 2.2 displays the empirical powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4} and Q_{KW4p} for βARMA$(1,1)$ model at $\varphi = 0.2$ and $\theta = 0.2$. We notice that when $n = 50$ (left panel) all tests display unstable behavior, making it difficult to single out the best performing test. Q_{DR} is the worst performer for both sample sizes. When the sample size is large ($n = 250$, right panel), the tests behave similarly when $m < 5$. Again, Q_{DR} is the least capable of detecting model misspecification. It is important to note two characteristics of all tests. First, all tests performances are affected by increasing the value of lag m: power decreases as m increases. Second, the powers of all tests for a given value of m increase with the sample size.

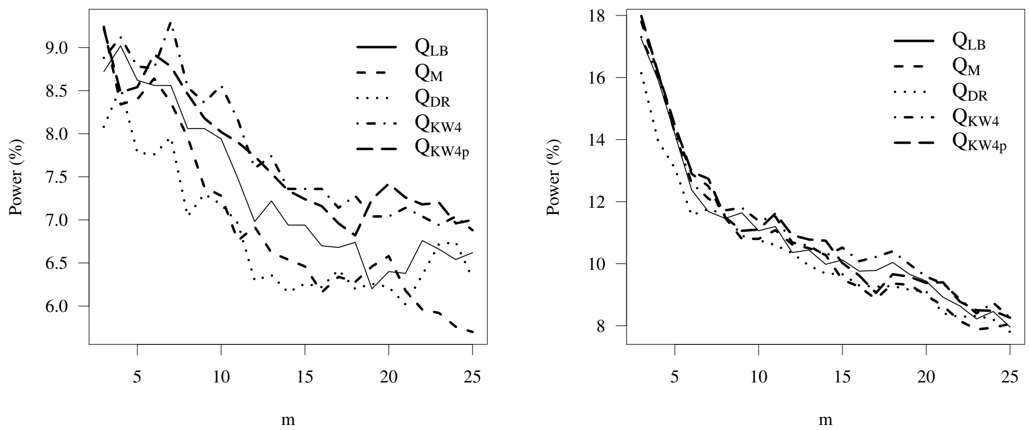
Figure 2.3 presents results obtained using $\varphi = 0.2$ and $\theta = 0.5$. Notice that when $n = 50$ the Q_{KW4p} test outperforms the competing tests for all values of m. The superiority of Q_{KW4p} also holds for $n = 250$. The Q_{DR} test is the worst performing test in both sample sizes.

Figure 2.4 shows the results obtained using $\varphi = 0.2$ and $\theta = 0.8$. The conclusions are similar to those obtained from the results presented in Figure 2.3, with Q_{KW4p} outperforming the competition. When $n = 250$, the Q_M and Q_{KW4p} tests behave similarly, both tests displaying power close to 100%.

The empirical powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4} and Q_{KW4p} for $\varphi = 0.5$ and $\theta = 0.2$ are

Table 2.9 Null rejection rates of bootstrap portmanteau tests, βMA(1) model, $\gamma = 5\%$.

Test	$n=50$														
	$\theta=0.2$					$\theta=0.5$					$\theta=0.8$				
	$m=5$	$m=10$	$m=15$	$m=20$	$m=25$	$m=5$	$m=10$	$m=15$	$m=20$	$m=25$	$m=5$	$m=10$	$m=15$	$m=20$	$m=25$
$\hat{D}_m$	4.8	5.2	5.4	5.3	5.2	4.5	4.9	4.7	5.0	4.9	4.8	5.2	5.1	5.0	5.0
Q_{LBb}	5.1	5.0	5.0	5.0	4.8	5.4	4.8	5.1	5.0	5.2	5.3	5.1	5.3	5.6	5.6
Q_{Mb}	5.1	4.8	5.0	5.3	5.2	5.4	4.9	5.3	5.3	5.2	5.3	4.9	4.9	5.4	5.3
Q_{DRb}	5.1	5.1	4.8	5.0	4.6	5.0	4.8	4.8	4.6	4.6	4.9	5.3	5.0	5.0	5.0
Q_{KW1b}	5.3	5.0	5.2	5.2	5.0	5.4	4.9	4.9	5.1	5.0	5.4	5.1	5.2	5.4	5.6
Q_{KW4b}	5.3	5.0	5.2	5.2	5.0	5.4	4.9	5.0	5.1	5.1	5.4	5.1	5.1	5.4	5.6
Q_{KW1pb}	5.1	4.8	5.1	5.3	5.0	5.4	4.8	4.9	5.2	5.2	5.2	4.7	4.9	5.2	5.2
Q_{KW4pb}	5.1	4.8	5.1	5.3	5.0	5.4	4.8	4.9	5.3	5.2	5.2	4.7	4.9	5.2	5.0



(a) $n = 50$ (b) $n = 250$

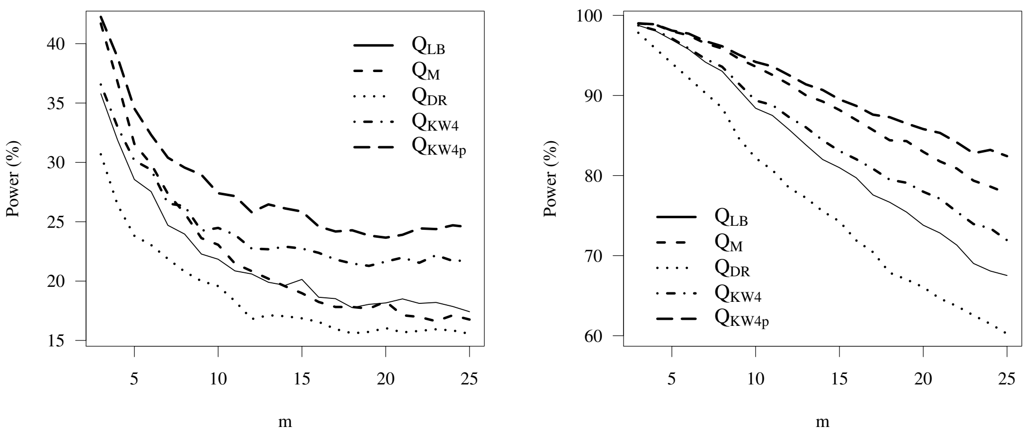
Figure 2.2 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βAR(1) and the true data generating process is βARMA$(1,1)$, $\varphi = 0.2$ and $\theta = 0.2$

presented in Figure 2.5. Again, when $n = 50$ all tests display unstable behavior. When $m > 6$, we note in Figure 2.5a that the Q_{KW4} test outperforms the competition. Figure 2.5b shows that all tests display similar behavior, except for Q_{DR}, which displays low ability to detect model misspecification.

Figure 2.6 presents results obtained using $\varphi = 0.5$ and $\theta = 0.5$. Notice that when $n = 50$ (left panel) Q_{KW4p} outperforms the remaining tests for all values of m. When the sample size is large ($n = 250$, right panel), the Q_{KW4p} test considerably outperforms all other tests for all values of m. The Q_{DR} test is the worst performer in both cases ($n = 50$ and $n = 250$). Visual inspection of Figure 2.3 reveals that the choice of m considerably impacts the powers of the tests. In particular, the tests ability to detect model misspecification weakens as m grows.

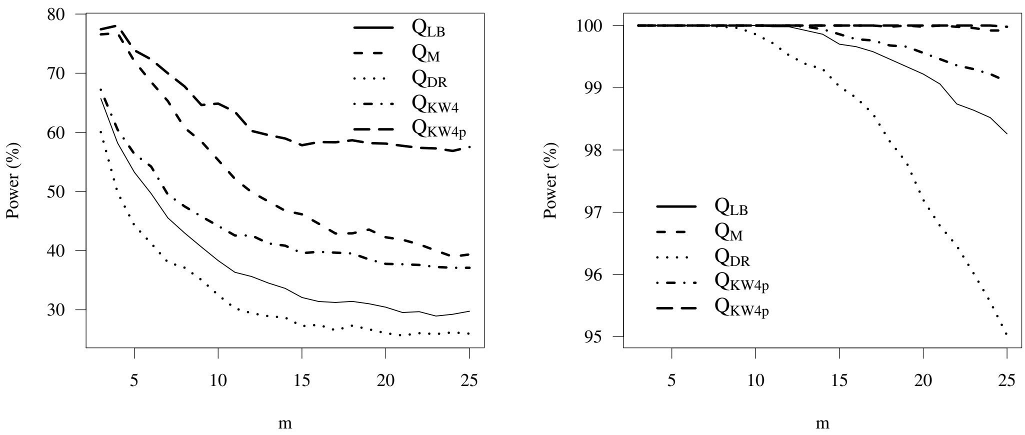
Figure 2.7 presents results obtained using $\varphi = 0.5$ and $\theta = 0.8$ in the βARMA data generating process. The Q_{KW4p} test has superior ability to detect misspecification for all values of lag m when $n = 50$. When $n = 250$, Figure 2.7b shows that the Q_M and Q_{KW4p} behave similarly, both with power close to 100%.

Figure 2.8 displays the empirical powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4} and Q_{KW4p} for $\varphi = 0.8$



(a) $n = 50$ (b) $n = 250$

Figure 2.3 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βAR(1) and the true data generating process is βARMA$(1,1)$, $\varphi = 0.2$ and $\theta = 0.5$

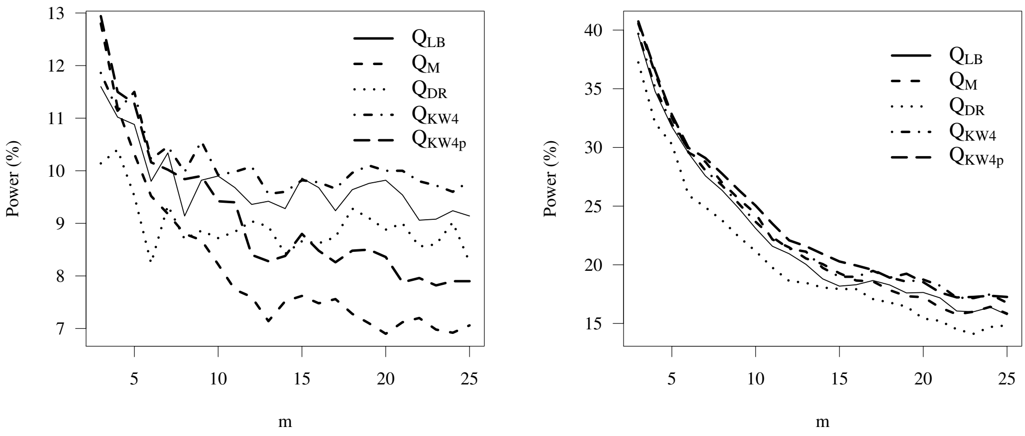


(a) $n = 50$ (b) $n = 250$

Figure 2.4 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βAR(1) and the true data generating process is βARMA$(1,1)$, $\varphi = 0.2$ and $\theta = 0.8$

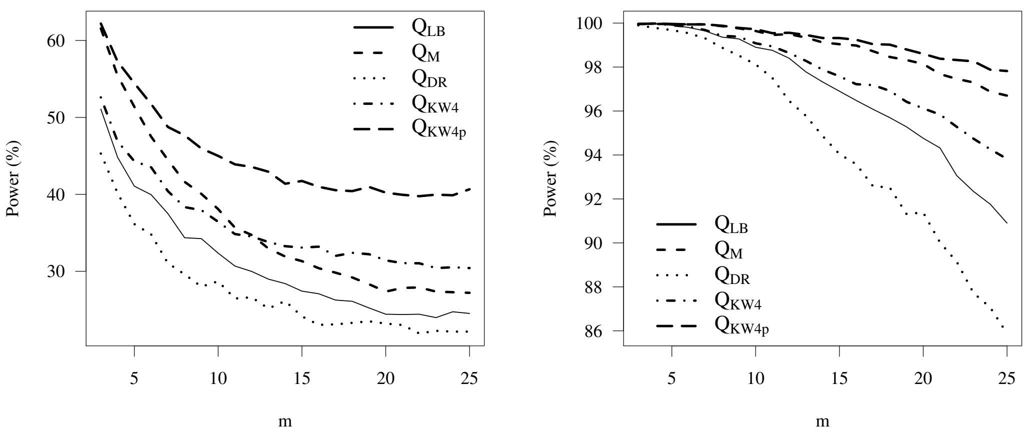
and $\theta = 0.2$. When $m > 16$, Q_{KW4p} has the test most capable of detecting model misspecification. The Q_{DR} test is the worst performer with both sample sizes.

Figure 2.9 and Figure 2.10 present results obtained using $\varphi = 0.8$, and $\theta = 0.5$ and $\theta = 0.8$ respectively. The conclusions are similar to those drawn from the previously results, with the Q_{KW4p} being the most powerful test for both sample sizes and Q_{DR} displaying the worst performance.



(a) $n = 50$ (b) $n = 250$

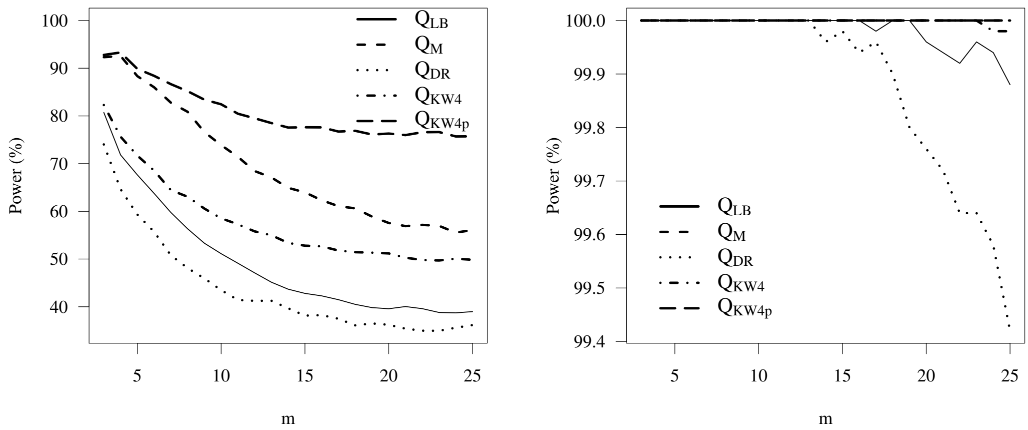
Figure 2.5 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βAR(1) and the true data generating process is βARMA$(1,1)$, $\varphi = 0.5$ and $\theta = 0.2$



(a) $n = 50$ (b) $n = 250$

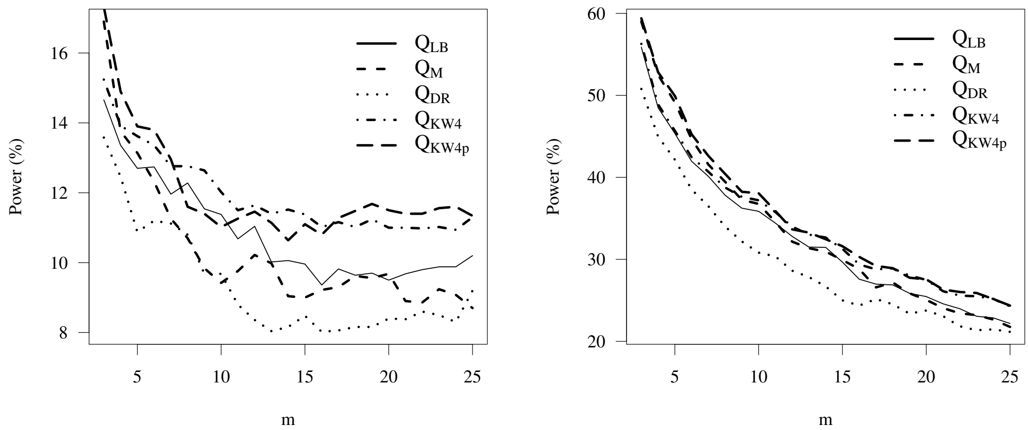
Figure 2.6 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βAR(1) and the true data generating process is βARMA$(1,1)$, $\varphi = 0.5$ and $\theta = 0.5$

How does the value of θ impact the tests powers? In order to answer that question we ran simulations using different values of the moving average parameter when generating the data. The value of m is fixed at 5, the value of the autoregressive parameter (φ) is fixed at 0.2, and two sample sizes are used: $n = 50, 250$. The tests estimated powers are displayed in Figure 2.11 (left panel for $n = 50$ and right panel for $n = 250$). The Q_{KW4} and Q_M tests are the clear winners when the sample size is small, especially when the value of θ is large; Q_{DR}



(a) $n = 50$ (b) $n = 250$

Figure 2.7 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βAR(1) and the true data generating process is βARMA$(1,1)$, $\varphi = 0.5$ and $\theta = 0.8$

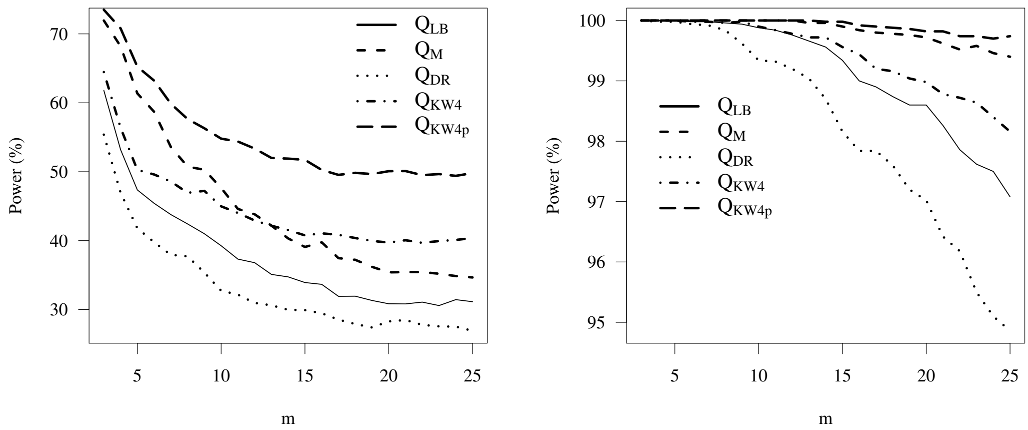


(a) $n = 50$ (b) $n = 250$

Figure 2.8 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βAR(1) and the true data generating process is βARMA$(1,1)$, $\varphi = 0.8$ and $\theta = 0.2$

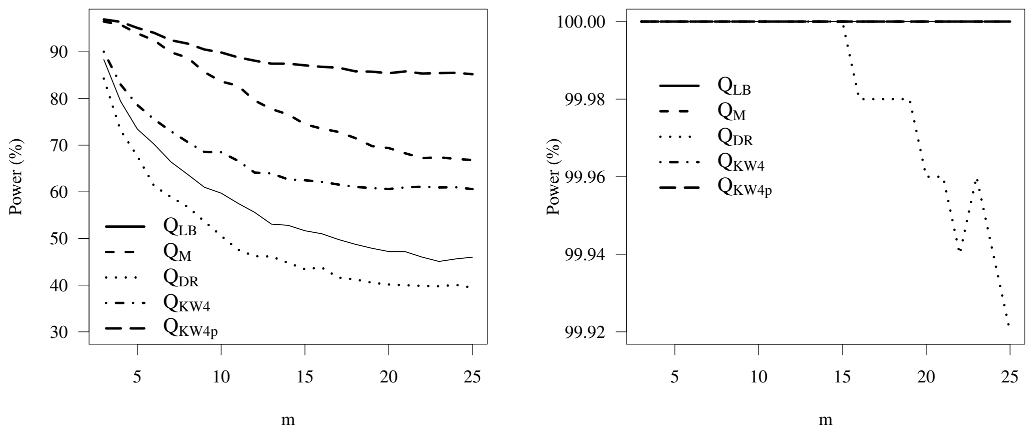
is the test with smallest power. When the sample size is large, Q_{KW4} and Q_M remain the most powerful tests, but not by much, and Q_{KW4} becomes the worst performing test.

We now move to the situation where the true data generating process is βARMA$(1,1)$ but the fitted model is βMA(1). In the previous case the fitted model was incorrectly specified because it failed to take into account relevant moving average dynamics. In contrast, model misspecification now stems from failing to account for important autoregressive dynamics. The



(a) $n = 50$ (b) $n = 250$

Figure 2.9 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βAR(1) and the true data generating process is βARMA$(1,1)$, $\varphi = 0.8$ and $\theta = 0.5$

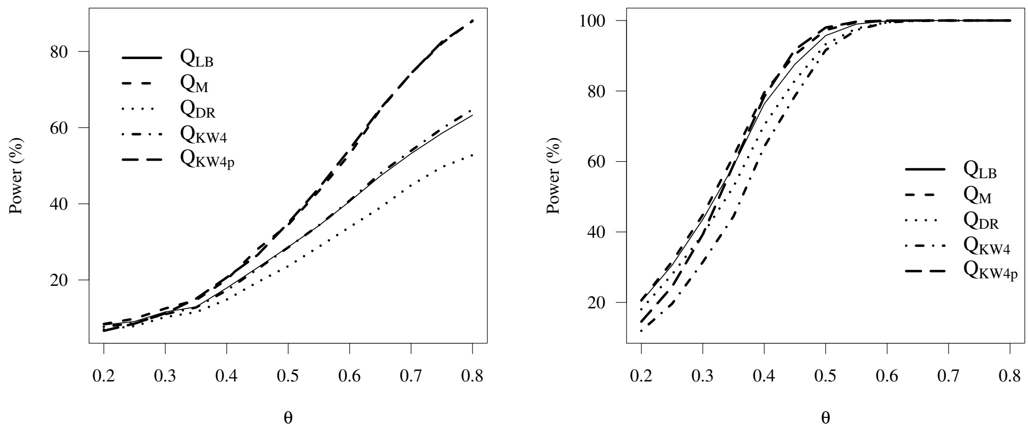


(a) $n = 50$ (b) $n = 250$

Figure 2.10 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βAR(1) and the true data generating process is βARMA$(1,1)$, $\varphi = 0.8$ and $\theta = 0.8$

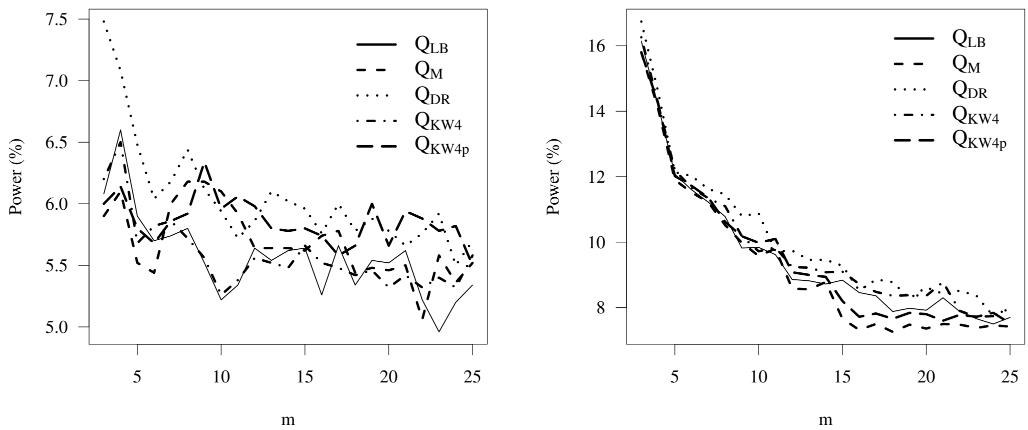
tests powers are displayed in Figure 2.12 which shows the estimated powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4} and Q_{KW4p} for $\varphi = 0.2$ and $\theta = 0.2$. The tests ability to detect model misspecification increase with the sample size, as expected, and decrease with m. It is important to notice that when $n = 50$ and $m = 3$ the Q_{DR} test displays better ability to detect misspecification model than the other tests.

Figure 2.13 displays results obtained using $\varphi = 0.2$ for $\theta = 0.5$. When $n = 50$ (left panel)



(a) $n = 50$ (b) $n = 250$

Figure 2.11 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4},Q_{KW4p}; the fitted model is βAR(1) and the true model is βARMA$(1,1)$, $\varphi = 0.2$.



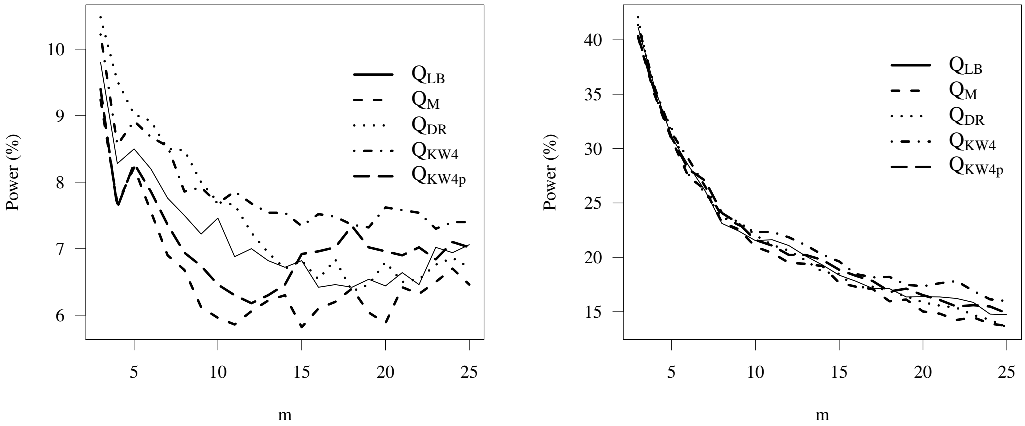
(a) $n = 50$ (b) $n = 250$

Figure 2.12 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βMA(1) and the true model is βARMA$(1,1)$, $\varphi = 0.2$ and $\theta = 0.2$

and $m > 10$ the Q_{KW4} test outperforms the other tests, but has low power. Figure 2.13b shows that when $m < 10$ the tests behave similarly.

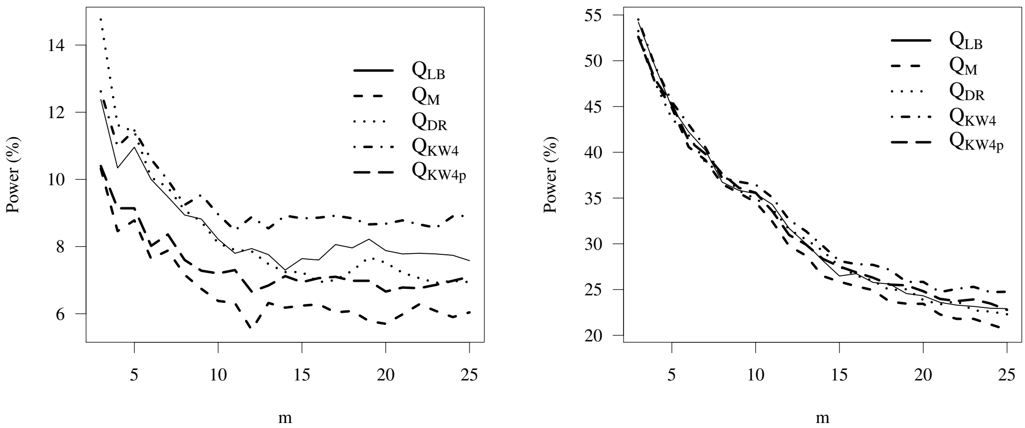
Figure 2.14 displays results obtained using $\varphi = 0.2$ for $\theta = 0.8$. The conclusions are similar to those drawn from Figure 2.13, with the Q_{KW4p} being the most powerful test for $m > 7$ and when $n = 250$ all test performing similarly.

Figure 2.15 presents the empirical powers of the Q_{LB}, Q_M, Q_{DR}, Q_{KW4} and Q_{KW4p} tests



(a) $n = 50$ (b) $n = 250$

Figure 2.13 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βMA(1) and the true model is βARMA$(1,1)$, $\varphi = 0.2$ and $\theta = 0.5$

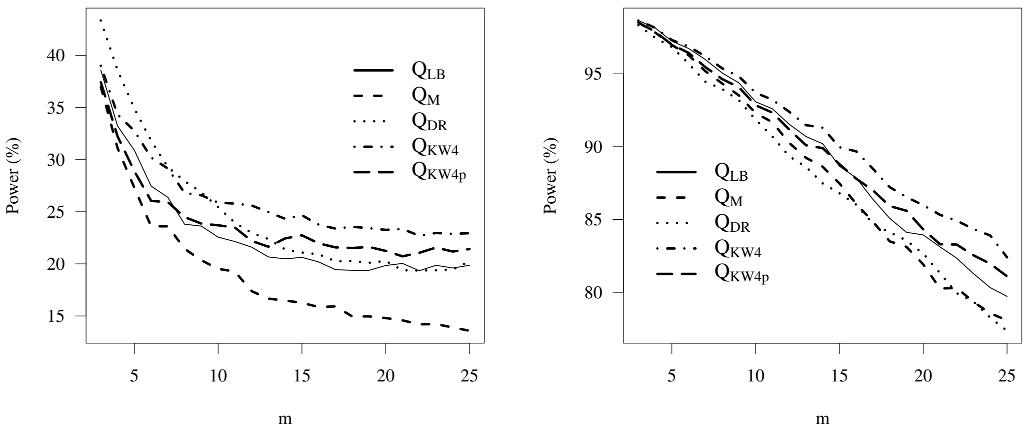


(a) $n = 50$ (b) $n = 250$

Figure 2.14 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βMA(1) and the true model is βARMA$(1,1)$, $\varphi = 0.2$ and $\theta = 0.8$

for $\varphi = 0.5$ and $\theta = 0.2$. When $n = 50$ the Q_M test is the worst performer for all values of m. The Q_{KW4} test outperforms the competitors when $m > 10$. When $n = 250$, Q_{KW4} is the most powerful test for all values of m.

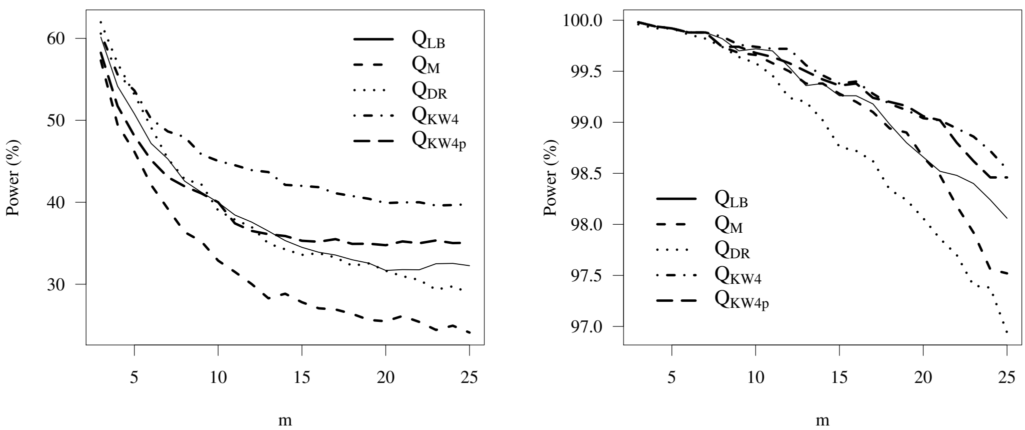
Figure 2.16 presents of empirical powers using $\varphi = 0.5$ for $\theta = 0.5$. When the sample size is small (left panel), Q_{KW4} is the most powerful test, followed by Q_{KW4p} when m is large. When the sample size is large, Q_{KW4} and Q_{KW4p} are the winners, but not by much when m is



(a) $n = 50$ (b) $n = 250$

Figure 2.15 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βMA(1) and the true model is βARMA$(1,1)$, $\varphi = 0.5$ and $\theta = 0.2$

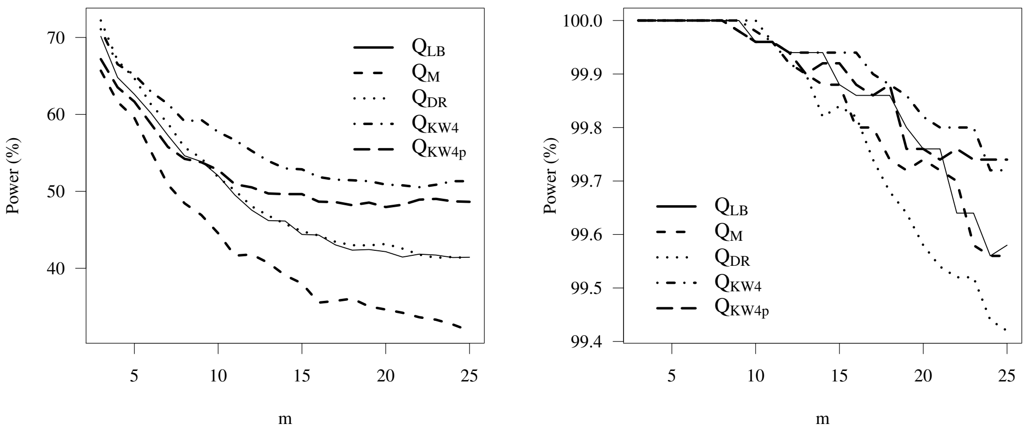
small; when m is large, the distance between the powers of Q_{KW4} and Q_{KW4p} and those of the remaining tests becomes more pronounced.



(a) $n = 50$ (b) $n = 250$

Figure 2.16 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βMA(1) and the true model is βARMA$(1,1)$, $\varphi = 0.5$ and $\theta = 0.5$

Figure 2.17 displays the tests empirical powers obtained using $\varphi = 0.5$ for $\theta = 0.8$. As in the previous result, the Q_{KW4} test is the most capable of detecting model misspecification when $n = 50$. When $n = 250$, the Q_{DR} test is the worst performer.



(a) $n = 50$ (b) $n = 250$

Figure 2.17 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βMA(1) and the true model is βARMA$(1,1)$, $\varphi = 0.5$ and $\theta = 0.8$

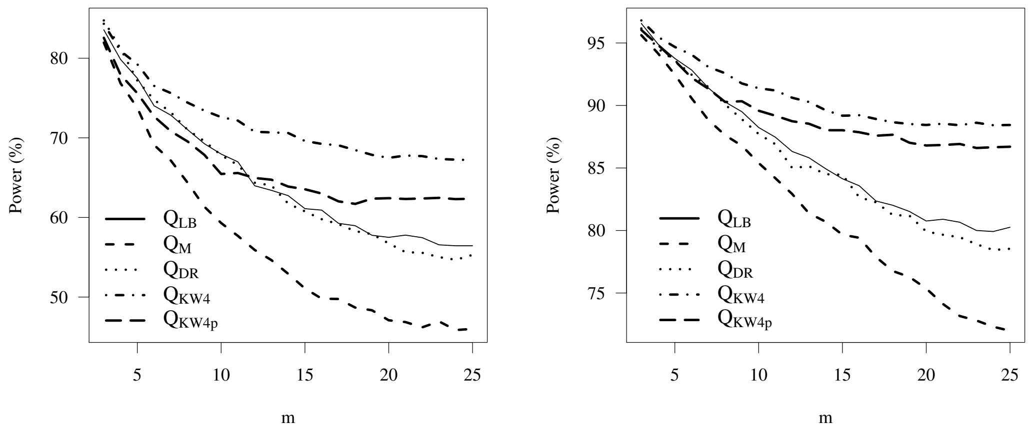
Figure 2.18a presents the empirical powers of the Q_{LB}, Q_M, Q_{DR}, Q_{KW4} and Q_{KW4p} tests for $\varphi = 0.8$ and $\theta = 0.2$. The results obtained using $\varphi = 0.8$ and $\theta = 0.8$ are displayed in Figure 2.18b. In both plots the sample size is $n = 50$. With both data generating processes, Q_{KW4} is the winner. In the right panel Q_{KW4p} is the runner-up and in the left panel Q_{KW4p} is the second best performing test when m is large. Interestingly, in both cases Q_M is the worst performer. Recall that the Q_M and Q_{KW4p} test statistics use residual partial partial autocorrelations; the Q_{KW4p} test is, however, considerably more powerful than the Q_M test. Even though we do not present results for $n = 250$, we note that when $\varphi = 0.8$ all tests displayed power of 100%.

We shall now consider a different model specification error, namely: the true data generating process is βARMA$(2,1)$ but the fitted model is βARMA$(1,1)$, i.e., some existing autoregressive dynamics is not accounted for. Figure 2.19 displays the empirical powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4} and Q_{KW4p} for $\varphi = 0.2$, 0.2 and $\theta = 0.2$. When $n = 50$, it is important to notice the instable behavior of all tests, the Q_M test being the worst performer. The empirical powers of tests when $n = 250$ are similar when $m = 3$.

Figure 2.20 presents the tests empirical powers obtained using $\varphi = 0.2, 0.2$ for $\theta = 0.5$. It is noteworthy the unstable behavior of the tests when $n = 50$ for all values of m. When $n = 250$, all tests display lower empirical powers and loss of power as the value of m increases.

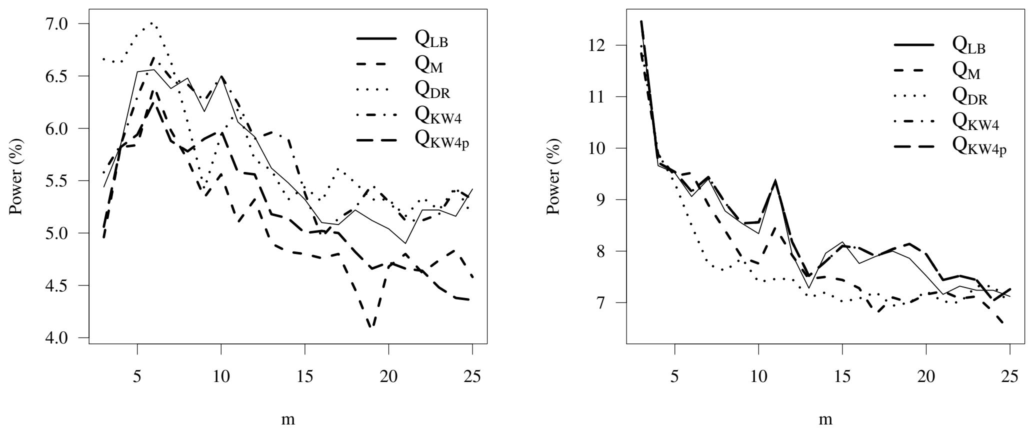
Figure 2.21 displays results obtained using $\varphi = 0.2, 0.2$ and $\theta = 0.8$, respectively. The conclusions are similar to those obtained from previously set of results, with the tests displaying unstable behavior for $n = 50$ and having loss of empirical power when $n = 250$.

Figure 2.22 displays the empirical powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4} and Q_{KW4p} for $\varphi = 0.2$, 0.5 and $\theta = 0.2$. It is noteworthy the influence of the sample size on the tests ability to detect that the model specification is in error: all the tests become considerably more powerful when



(a) $\varphi = 0.8$ and $\theta = 0.2$ (b) $\varphi = 0.8$ and $\theta = 0.8$

Figure 2.18 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βMA(1) and the true model is βARMA$(1,1)$, $\varphi = 0.8$, $\theta = 0.2$ and $\varphi = 0.8$, $\theta = 0.8$

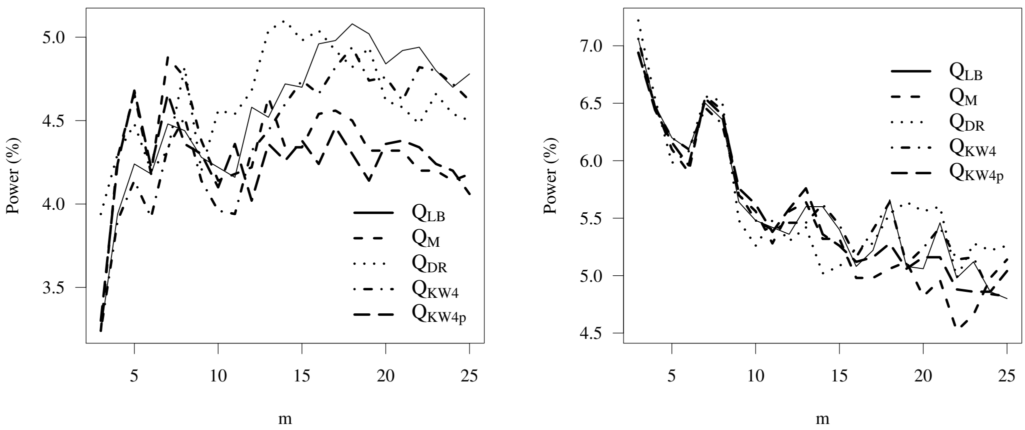


(a) $n = 50$ (b) $n = 250$

Figure 2.19 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βARMA$(1,1)$ and the true model is βARMA$(2,1)$, $\varphi = 0.2, 0.2$ and $\theta = 0.2$

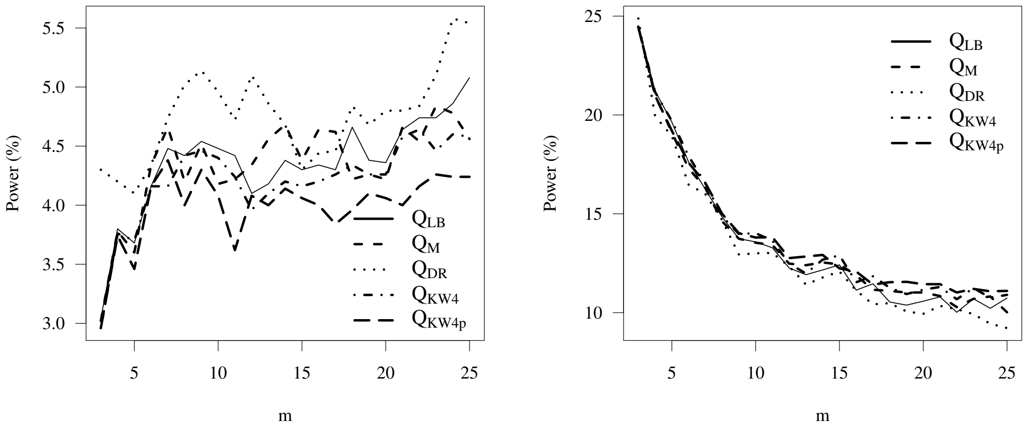
n goes from 50 to 250. Again, the tests powers decrease with m. In both cases, the Q_{KW4} is the clear winner.

Figure 2.23 presents the tests empirical powers obtained using $\varphi = 0.2, 0.5$ and $\theta = 0.5$, respectively. Again, all tests display unstable behavior when $n = 50$, the Q_{KW4} test having superior empirical power when $m > 6$. When $m > 9$, again the Q_{KW4} test is the most capable of detecting model misspecification.



(a) $n = 50$ (b) $n = 250$

Figure 2.20 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βARMA$(1,1)$ and the true model is βARMA$(2,1)$, $\varphi = 0.2, 0.2$ and $\theta = 0.5$

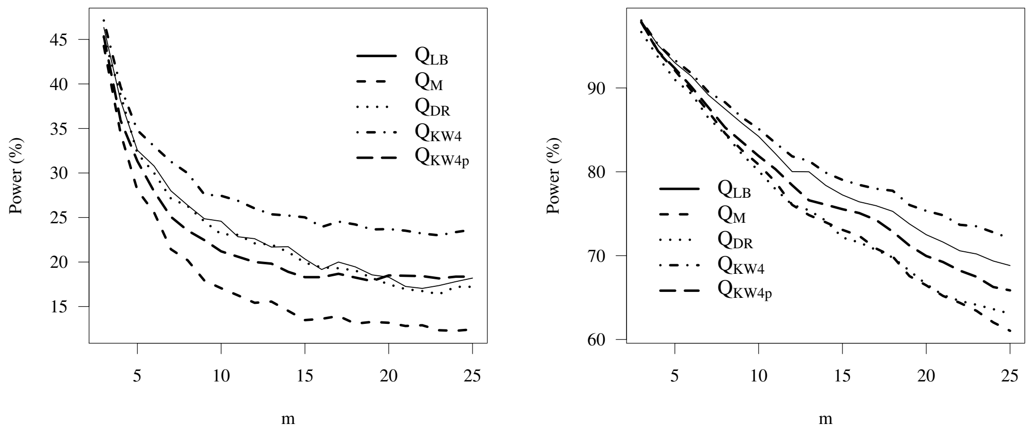


(a) $n = 50$ (b) $n = 250$

Figure 2.21 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βARMA$(1,1)$ and the true model is βARMA$(2,1)$, $\varphi = 0.2, 0.2$ and $\theta = 0.8$

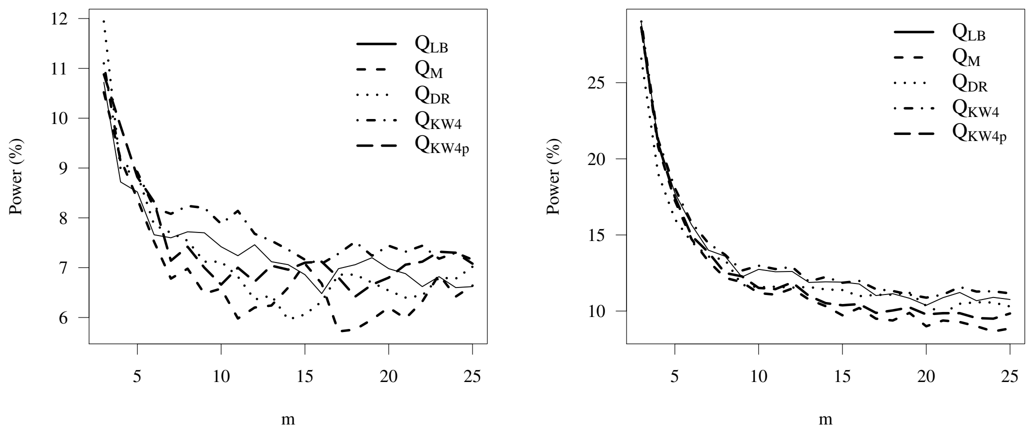
Figure 2.24 display results obtained using $\varphi = 0.2, 0.5$ and $\theta = 0.8$, respectively. The conclusions are similar to those drawn from previous results, with the tests displaying unstable behavior for $n = 50$ and having loss of empirical power when $n = 250$, being difficult to single out the test with the best performance.

Figure 2.25 displays the empirical powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4} and Q_{KW4p} for $\varphi = 0.2, 0.8$ and $\theta = 0.2$. The test proposed by Kwan and Sim (1996b) again outperformed the



(a) $n = 50$ (b) $n = 250$

Figure 2.22 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βARMA$(1,1)$ and the true model is βARMA$(2,1)$, $\varphi = 0.2, 0.5$ and $\theta = 0.2$

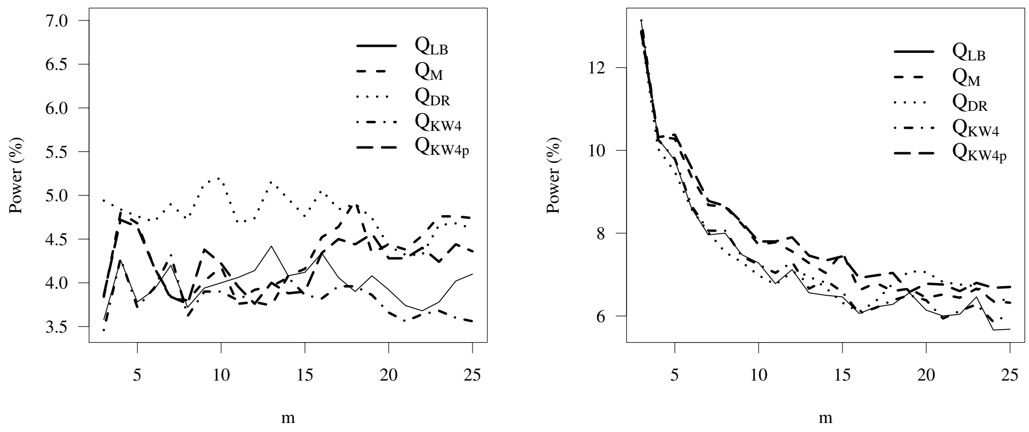


(a) $n = 50$ (b) $n = 250$

Figure 2.23 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βARMA$(1,1)$ and the true model is βARMA$(2,1)$, $\varphi = 0.2, 0.5$ and $\theta = 0.5$

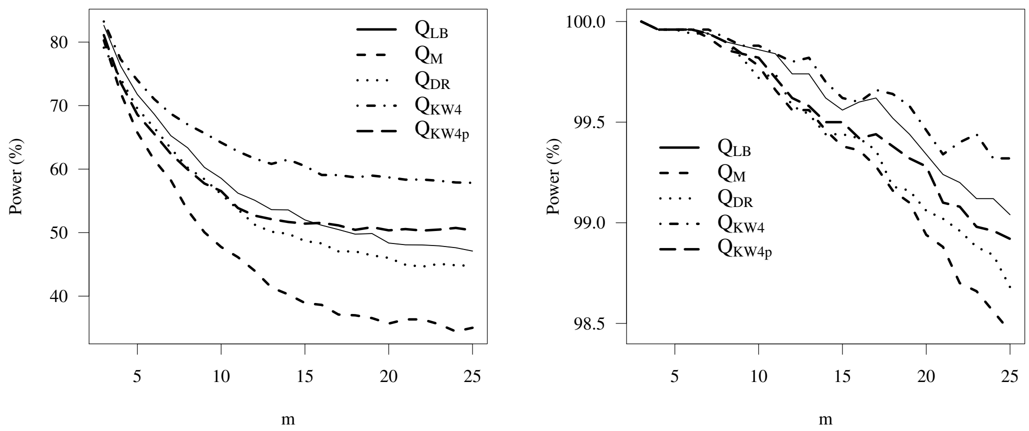
competition. It is noteworthy that Q_M (which, like Q_{KW4p}, is based on residual partial autocorrelations) is the worst performer. We also note that Q_{KW4p} is the second best performer when the sample size is small and m is large.

Figure 2.26a presents the empirical powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4} and Q_{KW4p} for $\varphi = 0.2, 0.8$ and $\theta = 0.5$. The results were obtained using $\varphi = 0.2, 0.8$ and $\theta = 0.8$ and are displayed in Figure 2.26b. In both plots, the results correspond to $n = 50$. In the left panel, Q_{KW4}



(a) $n = 50$ (b) $n = 250$

Figure 2.24 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βARMA$(1,1)$ and the true model is βARMA$(2,1)$, $\varphi = 0.2, 0.5$ and $\theta = 0.8$

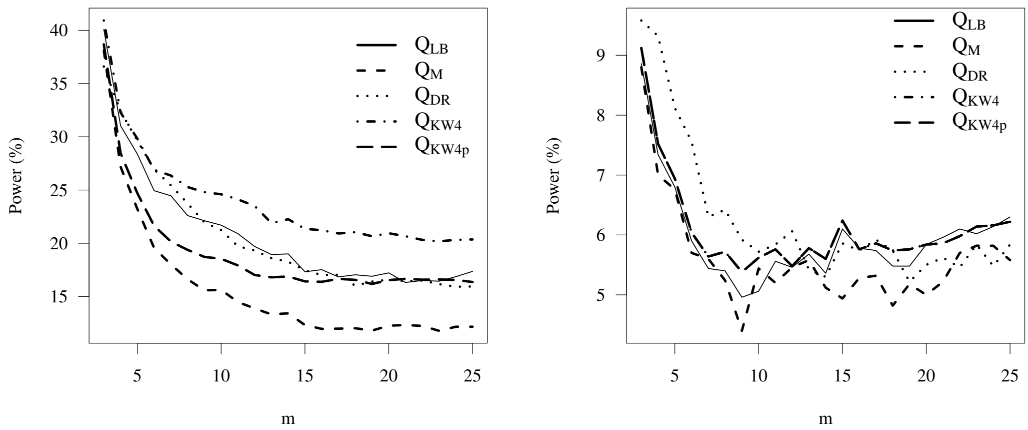


(a) $n = 50$ (b) $n = 250$

Figure 2.25 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βARMA$(1,1)$ and the true model is βARMA$(2,1)$, $\varphi = 0.2, 0.8$ and $\theta = 0.2$

is the winner. In the right panel, the Q_{DR} test presents superior performance relative to the competition.

Even though we do not present results for $n = 250$, we note that when $\theta = 0.5$ or $\theta = 0.8$ all tests powers equal 100%.



(a) $\varphi = 0.8$ and $\theta = 0.5$ (b) $\varphi = 0.8$ and $\theta = 0.8$

Figure 2.26 Powers of Q_{LB}, Q_M, Q_{DR}, Q_{KW4}, Q_{KW4p}; the fitted model is βARMA$(1,1)$ and the true model is βARMA$(2,1)$, $\varphi = 0.2, 0.8$, $\theta = 0.5$ and $\varphi = 0.2, 0.8$, $\theta = 0.8$

2.5 Empirical Application

We shall now turn to the empirical application briefly described in Section 2.1. The variable of interest, which assumes values in the standard unit interval, is the proportion of stocked hydroelectric energy (ONS, 2016) in South Brazil. The data are monthly averages from January 2001 to October 2016, thus covering 190 months ($n = 190$). The next six observations (November 2016 through April 2017) were used for evaluating the three models' forecasting accuracy. Table 2.10 contains some descriptive statistics on the data. Notice the negative skewness and also the negative excess kurtosis, and recall that the beta density easily accommodates such features.

Table 2.10 Descriptive statistics on the average rates of stocked energy in the South of Brazil.

min	max	median	mean	variance	asymmetry	excess kurtosis
0.2977	0.9862	0.7323	0.7069	0.0403	−0.3270	−1.1644

According to Shumway and Stoffer (2010), visual inspection of a time plot is recommended for detecting data anomalies, such as the presence of outliers or non-constant variability over time. A time series plot of the data is given by Figure 2.27. Notice that there are frequent returns to the overall mean which is indicative of stationarity. The series correlogram and partial correlogram are presented in Figure 2.28. The sample autocorrelation function shows fast decay, which is also indicative of stationarity. The augmented Dickey-Fuller (ADF) test (with drift) was applied to the data. The null hypothesis was rejected at the 1% nominal level. Again, there is evidence of stationarity.

Model selection was performed using the Akaike Information Criterion (AIC). We considered all models with autoregressive and moving average dynamics up to the fourth order and

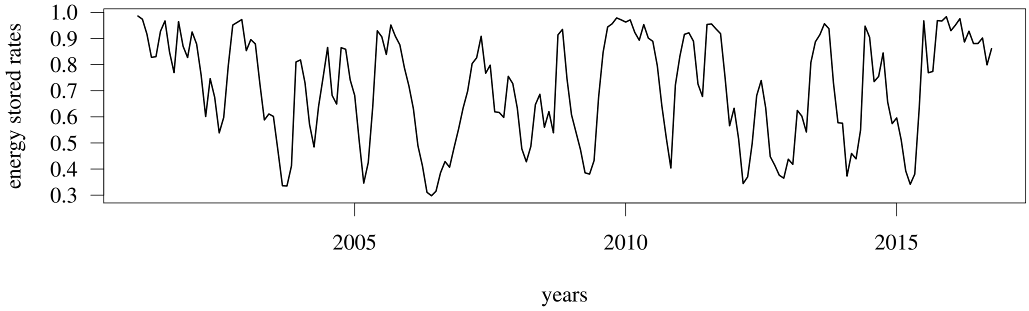
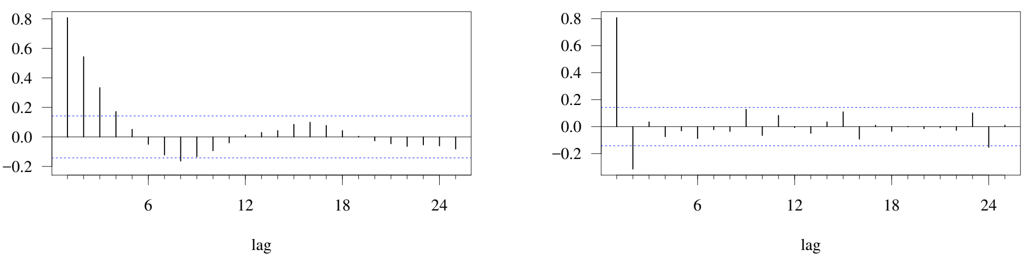


Figure 2.27 Average rates of stocked energy in the South of Brazil.



(a) ACF (b) PACF

Figure 2.28 Correlogram and partial correlogram.

logit link function. The selected model was the βARMA$(1,1)$ model, whose AIC was equal to -307.9635. Parameter estimation was carried out by numerically maximizing the conditional log-likelihood function using the BFGS quasi-Newton algorithm with analytical first derivatives; for details on the BFGS algorithm, see Press et al. (1992). Parameter estimates, standard errors and z-tests p-values are presented in Table 2.11. It is noteworthy that all parameters are significantly different from zero at the 1% significance level. (Recall that α is the model intercept.)

Table 2.11 Parameter estimates, standard errors and p-values; βARMA$(1,1)$ model.

	estimate	standard error	p-value
α	0.3452	0.0787	< 0.0001
φ_1	0.5235	0.0412	< 0.0001
θ_1	0.3588	0.0502	< 0.0001
ϕ	11.7593	1.1910	< 0.0001

Figure 2.29 contains a time series plot of the standardized residuals. One of the residu-

als exceeds 3 in absolute value but the corresponding observation is not an outlier at the 5% significance level according to the Bonferroni outlier criterion.

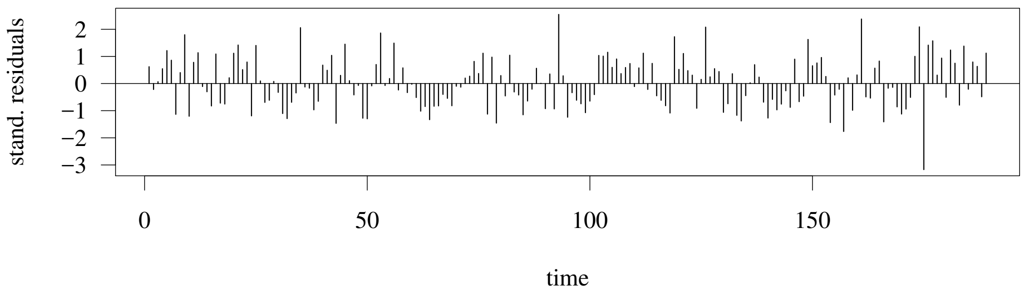
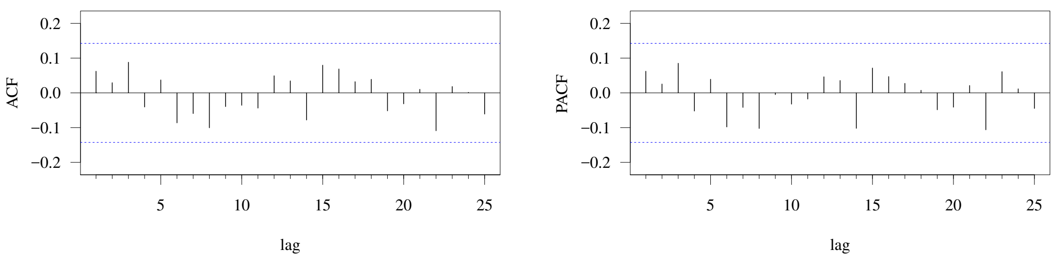


Figure 2.29 Standardized residuals from the fitted βARMA$(1,1)$ model.

Figure 2.30 contains the residuals correlogram (left panel) and partial correlogram (right panel). The dotted lines indicate the corresponding 95% confidence intervals. Since all residual autocorrelations and partial autocorrelations lie inside such intervals they can be taken to be statistically equal to zero and, as consequence, there is no evidence of model misspecification.



(a) ACF (b) PACF

Figure 2.30 Residual correlogram and partial correlogram.

We shall now consider portmanteau testing inference, i.e., we shall use the different portmanteau tests to assess whether the fitted model adequately represents the time series data. The tests p-values for different values of m are presented in Figure 2.31. Each plot includes a dashed line at 0.05. It is noteworthy that all p-values computed using $7 \leq m \leq 30$ lie above such a dashed line, thus indicating that the null hypothesis of correct model specification is not rejected at the 5% significance level (for each of m individually). Indeed, the only test that yields rejection of the null hypothesis is the test proposed by Dufour and Roy (1986) with small values of m.

The final step in our empirical analysis involves forecasting. Indeed, stocked energy forecasting is quite important for all institutions responsible for energy distribution. We produced

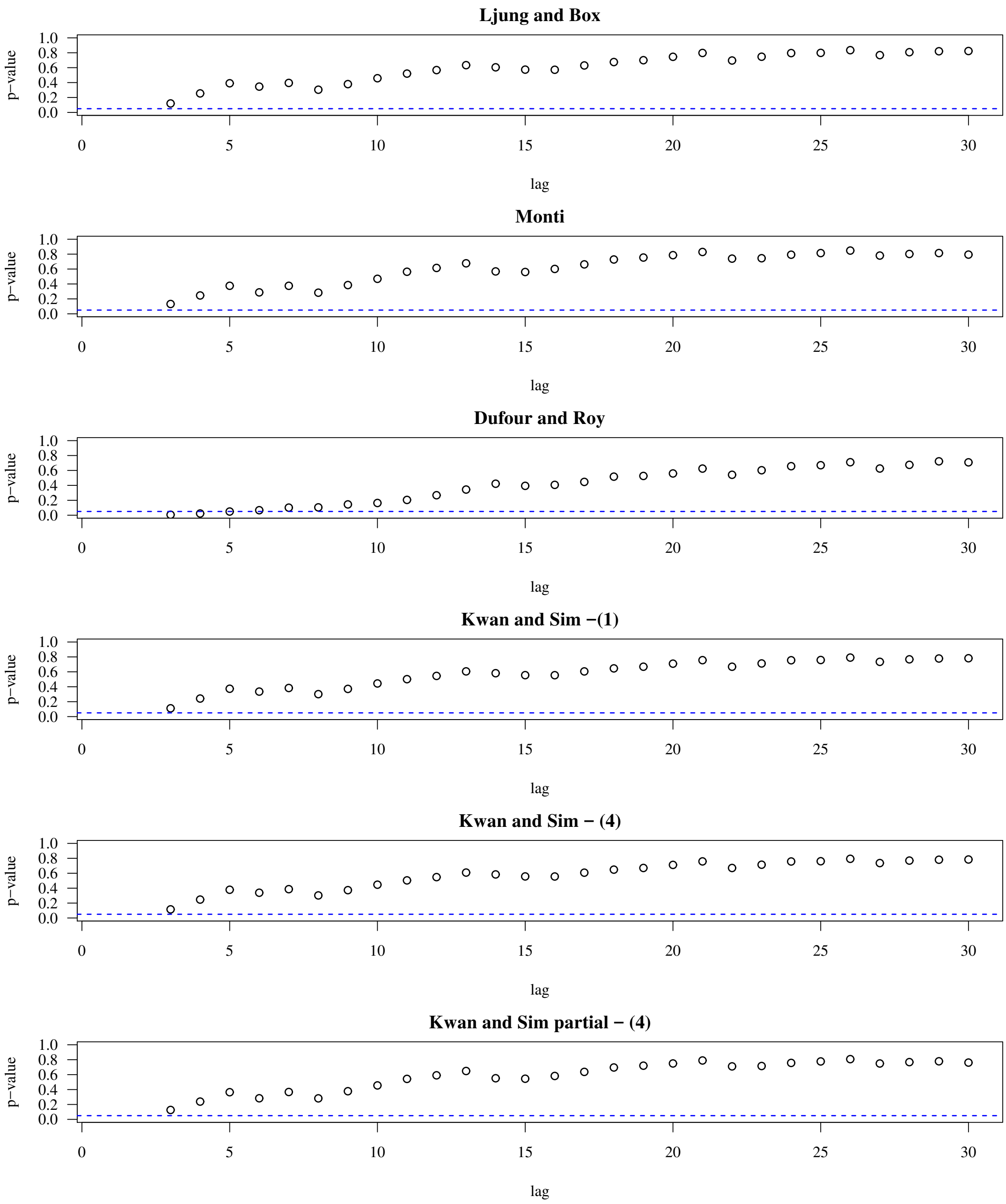


Figure 2.31 Portmanteau tests p-values.

forecasts using three different time series models, namely: the βARMA$(1,1)$ model, the Gaussian ARMA$(1,1)$, and the Gaussian AR(2) model which was selected by the AIC; the latter was selected using the `auto.arima` function of the *forecast* package of the R statistical computing environment (R Core Team, 2017).

The observed time series and the predicted values from the fitted βARMA$(1,1)$ model are presented in Figure 2.32. It is noteworthy that that the βARMA$(1,1)$ model is able to satisfactorily capture the data dynamics.

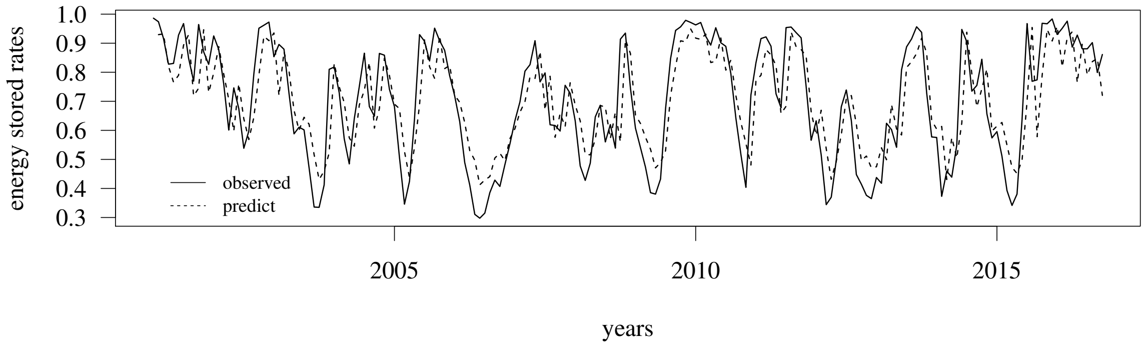


Figure 2.32 Energy stored rates (solid lines) and predict values (dashed lines) from the fitted model.

We now move from in-sample to out-of-sample forecasting. We consider a horizon of 6 months, i.e., we wish to forecasts the time series next six values. Forecasting accuracy is measures using the mean absolute error (MAE), i.e., the mean value of the absolute differences between observed and predicted values. The results are presented in Table 2.12 for $h = 1, \dots, 6$, h denoting the forecasting horizon. We note that the βARMA$(1,1)$ model yields forecasts that are more accurate than those obtained from the two competing models in all cases, i.e., for $h = 1, \dots, 6$. For instance, when forecasting the next three observations ($h = 3$), the βARMA$(1,1)$ MAE equals 0.1364 which is considerably smaller than the MAEs of the two competing models (0.1820 and 0.1680).

Table 2.12 Mean absolute forecasting errors, βARMA$(1,1)$, ARMA$(1,1)$ and AR(2).

	MAE					
	$h=1$	$h=2$	$h=3$	$h=4$	$h=5$	$h=6$
βARMA$(1,1)$	0.1244	0.1444	0.1364	0.1484	0.1694	0.1839
ARMA$(1,1)$	0.1518	0.1828	0.1820	0.1982	0.2211	0.2364
AR(2)	0.1345	0.1690	0.1680	0.1830	0.2050	0.2198

2.6 Conclusion

The βARMA model is particularly useful for modeling and forecasting time series data that assume values in the standard unit interval. The model naturally accommodates distributional asymmetry and nonconstant variance. It will always yield fitted values and out-of-sample forecasts that are positive and smaller than one. Additionally, no data transformation is needed prior to the analysis. The fitted model must be validated before it is used for forecasting. This is where our interest lies. Can the standard portmanteau tests be used with βARMA models?

If so, how to they behave in finite samples? What is the impact of the choice of the truncation lag (m) on the tests null and nonnull behaviors? We reviewed several portmanteau tests that are available in the literature and proposed two new tests. The test statistics we propose use residual partial autocorrelations instead of residual autocorrelations. We presented Monte Carlo simulation results on the finite sample behaviors of the different portmanteau tests in the class of βARMA models. Our results showed that some of the tests can be considerably size-distorted when the sample size is small. We then considered bootstrap variants of the tests and numerically evaluated their small sample performances. The evidence we provided showed that all tests became nearly size distortion-free when they were coupled with a bootstrap resampling scheme, especially when m is not very small.

The most interesting evidence from our numerical evaluations, however, relate to the tests powers. First, the choice of m impacts such powers: they typically decrease with m. Second, a portmanteau test that proved to be robust under Gaussian data — the test proposed by Dufour and Roy (1986) — did not perform well when used with βARMA models. Third, overall, the most powerful tests were that proposed by Kwan and Sim (1996b) and one the tests we proposed in this paper. (The two new tests displayed similar nonnull behaviors, hence we only presented results for one of them.) Our tests were the best performers in pure autoregressive and moving average models and the Kwan-Sim test was the most powerful test when the model included both autoregressive and moving average dynamics. It is noteworthy that whenever our tests were not the most powerful ones they were the next more powerful tests as long as the value of m was not small. Fourth, overall, the tests we proposed proved to be more powerful that of Monti (1994). This is interesting all three test statistics make use of residual partial autocorrelations.

We also presented and discussed an empirical application. Our focus was on modeling and forecasting the proportion of stocked hydroelectric energy in the southern region of Brazil. Such an empirical application showed the usefulness of portmanteau testing inference for model validation and also the usefulness of the class of βARMA time series models. It is noteworthy that the $\beta\text{ARMA}(1,1)$ used in the application yielded out-of-sample forecasts that were more accurate than those obtained using traditional time series models, i.e., models that are not based on the beta law.

CHAPTER 3

Recursion in Partial Derivatives

3.1 Introduction

The βARMA model (Rocha and Cribari-Neto, 2009) is a dynamic model that, in full generality, contains both autoregressive and moving averages dynamics. When the model only includes autoregressive dynamics the log-likelihood derivatives can be easily computed. In contrast, when the the model includes one or more moving average components a recursive structure in the log-likelihood derivatives must be accounted for. In this chapter, we provide closed form expressions for βARMA$(1,1)$ log-likelihood derivatives by considering errors in the predictor scale. Additionally, we perform a numerical evaluation to assess the differences between derivatives that take (corrrect) and do not take (incorrect) recursion into consideration. We also consider derivatives that are computed numerically.

3.2 βARMA log-likelihood recursive partial derivatives

The conditional maximum likelihood estimators of the βARMA model cannot be expressed in closed form. Focusing on a different class of dynamic models Benjamin et al. (1998) noted that when the time series model includes moving average parameters it is necessary to consider a recursive structure present in log-likelihood partial derivatives. As noted in the Section 2.2, Rocha and Cribari-Neto (2017) provided general expressions for such partial derivatives in the class of βARMA models. Starting values for η_t can be obtained by setting $\eta_t = g(y_t)$ and the partial derivativess of η with respect to the model parameters equal to zero for $t = 1, \ldots, q$ (Benjamin et al., 1998).

For βARMA models, the partial derivative of the linear predictor with respect to α is

$$\frac{\partial \eta_t}{\partial \alpha} = 1 - \sum_{j=1}^{q} \theta_j \frac{\partial \eta_{t-j}}{\partial \alpha}.$$

For example, for the βARMA$(1,1)$ model we obtain

$$\begin{aligned} 1 - \sum_{j=1}^{1} \theta_j \frac{\partial \eta_{t-j}}{\partial \alpha} &= 1 - \theta_1 \frac{\partial \eta_{t-1}}{\partial \alpha} \\ &= 1 - \theta_1 \left\{ 1 - \theta_1 \frac{\partial \eta_{(t-1)-1}}{\partial \alpha} \right\} \\ &= 1 - \theta_1 \left\{ 1 - \theta_1 \left[1 - \theta_1 \frac{\partial \eta_{(t-2)-1}}{\partial \alpha} \right] \right\} \end{aligned}$$

$$= 1 - \theta_1 \left\{ 1 - \theta_1 \left[1 - \theta_1 \left(1 - \theta \frac{\partial \eta_{(t-3)-1}}{\partial \alpha} \right) \right] \right\}$$

$$\vdots$$

$$= 1 + \sum_{k=1}^{t-2} \left\{ (-\theta_1)^k \right\} + (-\theta_1)^{t-1} \frac{\partial \eta_1}{\partial \alpha}.$$

The βARMA model may also include a set of regressors in the linear predictor. The vector of parameters associated with such regressors is $\beta = (\beta_1, \ldots, \beta_k)'$. The linear predictor partial derivative with respect to the lth component of such a vector is

$$\frac{\partial \eta_t}{\partial \beta_l} = x_t' - \sum_{i=1}^{p} \varphi_i x_{t-i}' - \sum_{j=1}^{q} \theta_j \frac{\partial \eta_{t-j}}{\partial \beta}.$$

As illustration, for the βARMA$(1,1)$ model, the above derivative reduces to

$$\begin{aligned}
x_t' - \sum_{i=1}^{1} \varphi_i x_{t-i}' - \sum_{j=1}^{1} \theta_j \frac{\partial \eta_{t-j}}{\partial \beta} &= x_t' - \varphi_1 x_{t-1}' - \theta_1 \frac{\partial \eta_{t-1}}{\partial \beta} \\
&= x_t' - \varphi_1 x_{t-1}' - \theta_1 \left\{ x_{(t-1)}' - \varphi_1 x_{(t-1)-1}' - \theta_1 \frac{\partial \eta_{(t-1)-1}}{\partial \beta} \right\} \\
&= x_t' - \varphi_1 x_{t-1}' - \theta_1 \left\{ x_{(t-1)}' - \varphi_1 x_{(t-1)-1}' - \theta_1 \left[x_{(t-2)}' - \varphi_1 x_{(t-2)-1}' \right.\right. \\
&\quad \left.\left. - \theta_1 \frac{\partial \eta_{(t-2)-1}}{\partial \beta} \right] \right\} \\
&= x_t' - \varphi_1 x_{t-1}' - \theta_1 \left\{ x_{(t-1)}' - \varphi_1 x_{(t-1)-1}' - \theta_1 \left[x_{(t-2)}' - \varphi_1 x_{(t-2)-1}' \right.\right. \\
&\quad \left.\left. - \theta_1 \left(x_{(t-3)}' - \varphi_1 x_{(t-3)-1}' - \theta_1 \frac{\partial \eta_{(t-3)-1}}{\partial \beta} \right) \right] \right\} \\
&\quad \vdots \\
&= x_t' + \sum_{k=1}^{t-2} \left\{ (-1)^k x_{t-k}' \theta^k (\varphi_1 \theta_1^{-1} + 1) \right\} - (-\theta_1)^{t-2} \varphi_1 x_1' \\
&\quad + (-\theta_1)^{t-1} \frac{\partial \eta_1}{\partial \beta}.
\end{aligned}$$

It is important to note that the partial derivatives with respect to the βARMA autoregressive parameters entail no recursion. The general expression of the partial derivative with respect to the ith autoregressive parameter (φ_i) is given by

$$\frac{\partial \eta_t}{\partial \varphi_i} = g(y_{t-i}) - x_{t-i}' \beta - \sum_{j=1}^{q} \theta_j \frac{\partial \eta_{t-j}}{\partial \varphi_i}, \quad i = 1, \ldots, p.$$

For the βARMA$(1,1)$ model, we obtain

$$g(y_{t-i}) - x_{t-i}' \beta - \sum_{j=1}^{1} \theta_j \frac{\partial \eta_{t-j}}{\partial \varphi_i} = g(y_{t-1}) - x_{t-1}' \beta - \theta_1 \frac{\partial \eta_{t-1}}{\partial \varphi_1}$$

$$
\begin{aligned}
&= g(y_{t-1}) - x'_{t-1}\beta - \theta_1 \left\{ g(y_{(t-1)-1}) - x'_{(t-1)-1}\beta - \theta_1 \frac{\partial \eta_{(t-1)-1}}{\partial \varphi_1} \right\} \\
&= g(y_{t-1}) - x'_{t-1}\beta - \theta_1 \left\{ g(y_{(t-1)-1}) - x'_{(t-1)-1}\beta - \theta_1 \left[g(y_{(t-2)-1}) \right.\right. \\
&\left.\left. - x'_{(t-2)-1}\beta - \theta_1 \frac{\partial \eta_{(t-2)-1}}{\partial \varphi_1} \right] \right\} \\
&= g(y_{t-1}) - x'_{t-1}\beta - \theta_1 \left\{ g(y_{(t-1)-1}) - x'_{(t-1)-1}\beta - \theta_1 \left[g(y_{(t-2)-1}) \right.\right. \\
&\left.\left. - x'_{(t-2)-1}\beta - \theta_1 \left(g(y_{(t-3)-1}) - x'_{(t-3)-1}\beta - \theta_1 \frac{\partial \eta_{(t-3)-1}}{\partial \varphi_1} \right) \right] \right\} \\
&\;\;\vdots \\
&= \sum_{k=0}^{t-2} \left\{ (-\theta_1)^k \left[g(y_{(t-k)-1} - x'_{t-1}\beta) \right] \right\} + (-\theta_1)^{t-1} \frac{\partial \eta_1}{\partial \varphi_1}.
\end{aligned}
$$

When the βARMA model only contains a single, first-order moving average component, the general expression for the partial derivative with respect to θ, the parameter associated with such a moving average term, is given by

$$
\frac{\partial \eta_t}{\partial \theta_l} = g(y_{t-l}) - \eta_{t-l} - \sum_{j=1}^{q} \theta_j \frac{\partial \eta_{t-j}}{\partial \theta_l}, \quad l = 1, \ldots, q.
$$

This derivative for the βARMA$(1,1)$ is

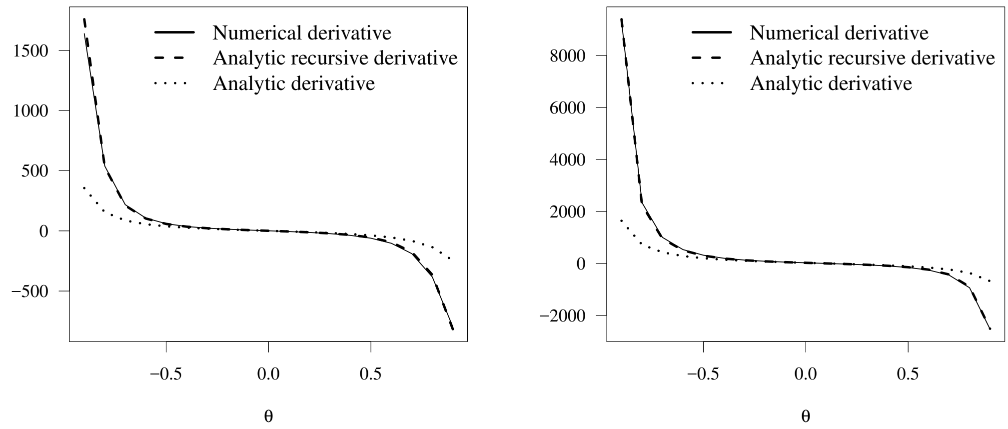
$$
\begin{aligned}
g(y_{t-l}) - \eta_{t-l} - \sum_{j=1}^{1} \theta_j \frac{\partial \eta_{t-j}}{\partial \theta_l} &= g(y_{t-1}) - \eta_{t-1} - \theta_1 \frac{\partial \eta_{t-1}}{\partial \theta_1} \\
&= g(y_{t-1}) - \eta_{t-1} - \theta_1 \left\{ g(y_{(t-1)-1}) - \eta_{(t-1)-1} - \theta_1 \frac{\partial \eta_{(t-1)-1}}{\partial \theta_1} \right\} \\
&= g(y_{t-1}) - \eta_{t-1} - \theta_1 \left\{ g(y_{(t-1)-1}) - \eta_{(t-1)-1} - \theta_1 \left[g(y_{(t-2)-1}) \right.\right. \\
&\left.\left. - \eta_{(t-2)-1} - \theta_1 \frac{\partial \eta_{(t-2)-1}}{\partial \theta_1} \right] \right\} \\
&= g(y_{t-1}) - \eta_{t-1} - \theta_1 \left\{ g(y_{(t-1)-1}) - \eta_{(t-1)-1} - \theta_1 \left[g(y_{(t-2)-1}) \right.\right. \\
&\left.\left. - \eta_{(t-2)-1} - \theta_1 \left(g(y_{(t-3)-1}) - \eta_{(t-3)-1} - \theta_1 \frac{\partial \eta_{(t-3)-1}}{\partial \theta_1} \right) \right] \right\} \\
&\;\;\vdots \\
&= \sum_{k=0}^{t-2} \left\{ (-\theta_1)^k \left[g(y_{(t-k)-1} - \eta_{t-1}) \right] \right\} + (-\theta_1)^{t-1} \frac{\partial \eta_1}{\partial \theta_1}.
\end{aligned}
$$

In what follows, we shall present the the results of a set of numerical evaluations that were carried out to compare partial derivatives computed: (i) numerically, (ii) analytically by taking recursion into account, (iii) as presented in Rocha and Cribari-Neto (2009), i.e., without recursion. [The corrected, recursively-based derivatives are presented in Rocha and Cribari-Neto

(2017)]. We aim at proving the following question with an answer: How much precision is lost when recursion is not accounted for?

All numerical evaluations were carried out using the R statistical environment. The numerical derivatives were computed using *grad* function of the *numDeriv* package.

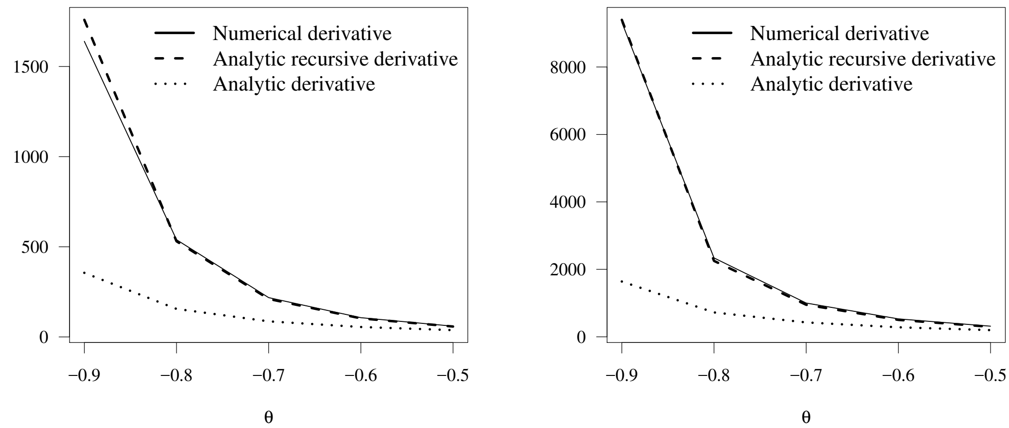
We consider the $\beta\text{ARMA}(1,1)$ model with the autoregressive parameter equal to 0.2 and with the value of the moving average parameter varying in the closed interval $[-0.9, 0.9]$. The sample sizes considered are $n = 50$ and $n = 250$. In Figure 3.1 we plot the three derivatives of η_t with respect to θ against θ. We note that, for both sample sizes ($n = 50$ and $n = 250$) the numerical value of the partial derivatives that consider recursion are very close for all values of θ. These two derivatives are similar to the analytic derivative computed without taking recursion into consideration for $-0.5 \leq \theta \leq 0.5$; when $\theta \notin [-0.5, 0.5]$, the no-recursion derivative becomes considerably imprecise.



(a) $n = 50$ (b) $n = 250$

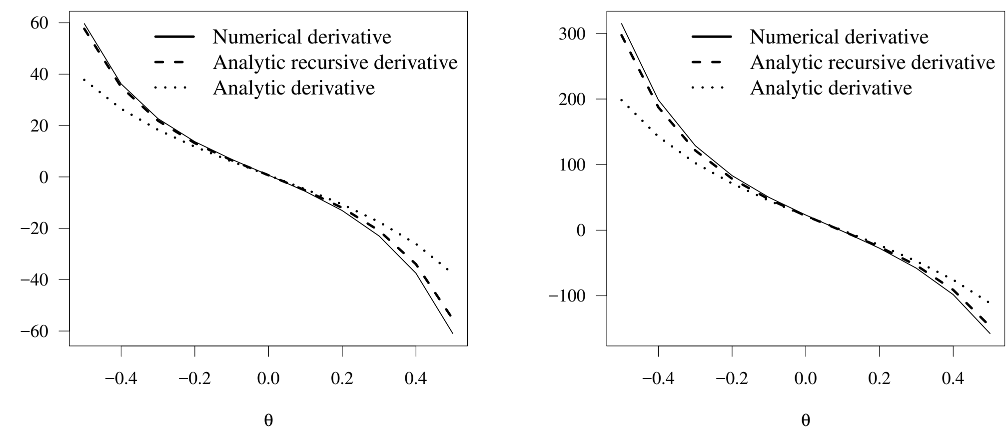
Figure 3.1 Numerical partial derivative (solid line), analytic partial derivative with recursion (dashed line) and analytic partial derivative without recursion (dot line) for the $\beta\text{ARMA}(1,1)$ model, $\varphi = 0.2$.

For better visualization, we present plots for smaller ranges of variation of θ: (i) θ assumes values in $[-0.9, -0.5]$ (Figure 3.2), (ii) θ assumes values in $[-0.5, 0.5]$ (Figure 3.3), and (iii) θ assumes values in $[0.5, 0.9]$ (Figure 3.4). Notice the large difference between the two analytic derivatives for $\theta < 0.5$ and for $\theta > 0.5$. It is now clear that good agreement between the two analytic derivatives only occurs when $-0.2 < \theta < \theta$; see Figure 3.3. In conclusion, the derivative that does not account for recursion becomes progressively more inaccurate as $|\theta|$ moves away from zero, more so for values of $|\theta|$ in excess of 0.5.



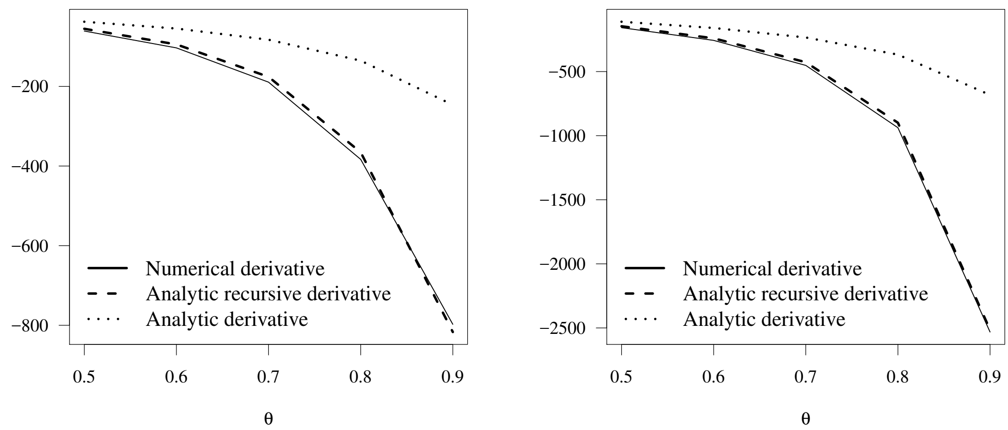
(a) $n = 50$ (b) $n = 250$

Figure 3.2 Numerical partial derivative (solid line), analytic partial derivative with recursion (dashed line) and analytic partial derivative without recursion (dot line) for the βARMA$(1,1)$ model, $\varphi = 0.2$; first range of values for θ.



(a) $n = 50$ (b) $n = 250$

Figure 3.3 Numerical partial derivative (solid line), analytic partial derivative with recursion (dashed line) and analytic partial derivative without recursion (dot line) for the βARMA$(1,1)$ model, $\varphi = 0.2$; second range of values for θ.



(a) $n = 50$ (b) $n = 250$

Figure 3.4 Numerical partial derivative (solid line), analytic partial derivative with recursion (dashed line) and analytic partial derivative without recursion (dot line) for the βARMA$(1,1)$ model, $\varphi = 0.2$; third range of values for θ.

References

Akaike, H. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

Andersen, E. B. Asymptotic properties of conditional maximum-likelihood estimators. *Journal of the Royal Statistical Society B*, pages 283–301, 1970.

Benjamin, M. A., Rigby, R. A., and Stasinopoulos, M. D. Fitting non-Gaussian time series models. In *COMPSTAT*, pages 191–196, Great Britain, 1998. Springer.

Box, G. E. and Pierce, D. A. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American Statistical Association*, 65(332):1509–1526, 1970.

Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2015.

Brockwell, P. J. and Davis, R. A. *Introduction to Time Series and Forecasting*. Springer, Switzerland, 2016.

Carbon, M. and Francq, C. Portmanteau goodness-of-fit test for asymmetric power garch models. *Austrian Journal of Statistics*, 40(1&2):55–64, 2016.

Chang, I., Tiao, G. C., and Chen, C. Estimation of time series parameters in the presence of outliers. *Technometrics*, 30(2):193–204, 1988.

Cryer, J. and Chan, K. *Time Series Analysis: With Applications in R*. Springer, New York, 2008. ISBN 9780387759593.

Davies, N., Triggs, C., and Newbold, P. Significance levels of the Box-Pierce portmanteau statistic in finite samples. *Biometrika*, 64(3):517–522, 1977.

Davison, A. C. and Hinkley, D. V. *Bootstrap Methods and Their Application*. Cambridge University Press, New York, 1997.

Dufour, J.-M. and Roy, R. Generalized portmanteau statistics and tests of randomness. *Communications in Statistics-Theory and Methods*, 15(10):2953–2972, 1986.

Ferrari, S. L. P. and Cribari-Neto, F. Beta regression for modelling rates and proportions. *Journal of Applied Statistics*, 31(7):799–815, 2004.

Fisher, R. On the “probable error”of a coefficient of correlation deduced from a small sample. *Metron*, 1:332, 1921.

Fox, A. J. Outliers in time series. *Journal of the Royal Statistical Society B*, 34(3):350–363, 1972.

Hotelling, H. New light on the correlation coefficient and its transforms. *Journal of the Royal Statistical Society B*, 15(2):193–232, 1953.

Jenkins, G. An angular transformation for the serial correlation coefficient. *Biometrika*, 41 (1/2):261–265, 1954.

Kwan, A. C. and Sim, A.-B. On the finite-sample distribution of modified portmanteau tests for randomness of a gaussian time series. *Biometrika*, 83(4):938–943, 1996a.

Kwan, A. C. and Sim, A.-B. Portmanteau tests of randomness and jenkins’ variance-stabilizing transformation. *Economics Letters*, 50(1):41–49, 1996b.

Kwan, A. C., Sim, A.-B., and Wu, Y. A comparative study of the finite-sample performance of some portmanteau tests for randomness of a time series. *Computational Statistics & Data Analysis*, 48(2):391–413, 2005.

Kwan, C. K. and Wu, Y. Further results on the finite-sample distribution of Monti’s portmanteau test for the adequacy of an ARMA (p, q) model. *Biometrika*, 84(3):733–736, 1997.

Lin, J.-W. and McLeod, A. I. Improved Peña–Rodriguez portmanteau test. *Computational Statistics & Data Analysis*, 51(3):1731–1738, 2006.

Ljung, G. M. Diagnostic testing of univariate time series models. *Biometrika*, 73(3):725–730, 1986.

Ljung, G. M. and Box, G. E. On a measure of lack of fit in time series models. *Biometrika*, 65(2):297–303, 1978.

McLeod, A. On the distribution of residual autocorrelations in Box-Jenkins models. *Journal of the Royal Statistical Society B*, 40(3):296–302, 1978.

McLeod, A. I. and Jimenez, C. Nonnegative definiteness of the sample autocovariance function. *The American Statistician*, 38(4):297–298, 1984.

Monti, A. C. A proposal for a residual autocorrelation test in linear models. *Biometrika*, 81 (4):776–780, 1994.

Morettin, P. A. and Toloi, C. *Análise de Séries Temporais*. Blucher, 2006.

ONS. Operador nacional do sistema elétrico - energia armazenada., 2016. URL `http://www.ons.org.br/historico/energia_armazenada.aspx`.

Peña, D. and Rodriguez, J. A powerful portmanteau test of lack of fit for time series. *Journal of the American Statistical Association*, 97:601–610, 2002.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. *Numerical Recipes in C: The Art of Scientific Computing*, volume 2. Cambridge University Press, New York, 1992.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017. URL `https://www.R-project.org/`.

Rocha, A. V. and Cribari-Neto, F. Beta autoregressive moving average models. *Test*, 18(3): 529–545, 2009.

Rocha, A. V. and Cribari-Neto, F. Erratum to: Beta autoregressive moving average models. *TEST*, pages 1–9, 2017.

Shumway, R. H. and Stoffer, D. S. *Time Series Analysis and its Applications: With R Examples*. Springer, 2010.

Wei, W. W.-S. *Time Series Analysis*. Addison-Wesley Publishing Company, 1994.

APPENDIX A

Appendix A

In this Appendix we present the R source codes of two Monte Carlo simulations programs that were used to evaluate the finite sample performances of different portmanteau tests in the class of βARMA models. The simulation results obtained using such programs were presented and discussed in Chapter 2.

A.1 A.1 Null rejection rates without resorting to bootstrap resampling

```
###################################################################################################
PROGRAM: Simu_dissert1p.r

USE: The program computes the null rejection rates of the portmanteau tests in the BARMA model.

AUTHOR: Vinícius Teodoro Scher

VERSION: 1.10

LAST MODIFIED: 08/12/2016
##################################################################################################

source("simu_barma.r") #function that generates observations of a BARMA model.
source("barma.r")      #function that estimates the parameters of the model.
source("barma.fit.r")  #function taht estimates the parameters using recursive derivatives.
source("ljung_box.r")  #function that performs the Ljung-Box test.
source("dufour.r")     #function that performs the Dufour and Roy test.
source("monti.r")      #function that performs the Ana Cristina Monti test.
source("kwan.r")       #function that performs the 1 de Kwan and Sim test.
source("kwan2.r")      #function that performs the 2 de Kwan and Sim test.
source("kwan3.r")      #function that performs the 3 de Kwan and Sim test.
source("kwan4.r")      #function that performs the 4 de Kwan and Sim test.
source("Kwan_Chest.R") #function that performs the Kwan and Sim tests with partial autocorrelation.

library(moments)       #package that allows the calculation of curtosis and skewness.
library(doMC)          #package that realize parallel simulation.
library(doRNG)         #package that allows the selection of seed in parallel simulation.
library(matrixStats)   #package that allows calculate descriptive measures of matrix.
registerDoMC(24)       #selection of cluster number in parallel.


############################## Simulations ######################################################
R<-5000                                       #number of Monte Carlo replications.
vm<- c(5,10,15,20,25)                         #number of lag considered.
vn<- c(50,250,500)                            #sizes of samples.
va<-c(0.01,0.05,0.1)                          #significance levels.
results<-matrix(rep(NA,2160),nrow=216,ncol=10) #matrix that stores null rejection rates.
results2<-matrix(rep(NA,360),nrow=36,ncol=10)  #matrix that stores descriptive measures.
results3<-matrix(rep(NA,1890),nrow=189,ncol=10)#matrix that stores quantiles.
j<-1                                          #index control.
b<-1                                          #index control.
t<-1                                          #index control.
```

```
f<-1                                            #degress of freedom for chi-square distribution.
vp<-c(0.2,0.5,0.8)                              #vector for autoregressive parameters.
vq<-c(NA)                                       #vector for moving average parameters.
prec<-120                                       #precision parameter.
typ<-"partial"                                  #type of autocorrelation for Kwan_Chest function.

#Monte Carlo start
 for(p in vp)
 {
  for(q in vq)
  {
   for(n in vn)
   {
      #null rejections rates
      rej1<-matrix(rep(0,R),nrow=R, ncol=15) #partial results matrix I.
      rej2<-matrix(rep(0,R),nrow=R, ncol=15) #partial results matrix II.
      rej3<-matrix(rep(0,R),nrow=R, ncol=15) #partial results matrix III.
      rej4<-matrix(rep(0,R),nrow=R, ncol=15) #partial results matrix IV.
      rej5<-matrix(rep(0,R),nrow=R, ncol=15) #partial results matrix V.
      rej6<-matrix(rep(0,R),nrow=R, ncol=15) #partial results matrix VI.
      rej7<-matrix(rep(0,R),nrow=R, ncol=15) #partial results matrix VII.

      #parameters of model
      vectA<-rep(NA,R)                       #initializing alpha vector
      vectP<-rep(NA,R)                       #initializing phi vector
      vectT<-rep(NA,R)                       #initializing theta vector
      vectPR<-rep(NA,R)                      #initializing precision vector

      #p-values of tests
      LBPV<-matrix(rep(0,R),nrow=1, ncol=5)  #partial results matrix I.
      MOPV<-matrix(rep(0,R),nrow=1, ncol=5)  #partial results matrix II.
      DRPV<-matrix(rep(0,R),nrow=1, ncol=5)  #partial results matrix III.
      KS1PV<-matrix(rep(0,R),nrow=1, ncol=5) #partial results matrix IV.
      KS2PV<-matrix(rep(0,R),nrow=1, ncol=5) #partial results matrix V.
      KS3PV<-matrix(rep(0,R),nrow=1, ncol=5) #partial results matrix VI.
      KS4PV<-matrix(rep(0,R),nrow=1, ncol=5) #partial results matrix VII.

      #statistics of tests
      LBPVE<-matrix(rep(0,R),nrow=1, ncol=5) #partial results matrix I.
      MOPVE<-matrix(rep(0,R),nrow=1, ncol=5) #partial results matrix II.
      DRPVE<-matrix(rep(0,R),nrow=1, ncol=5) #partial results matrix III.
      KS1PVE<-matrix(rep(0,R),nrow=1, ncol=5)#partial results matrix IV.
      KS2PVE<-matrix(rep(0,R),nrow=1, ncol=5)#partial results matrix V.
      KS3PVE<-matrix(rep(0,R),nrow=1, ncol=5)#partial results matrix VI.
      KS4PVE<-matrix(rep(0,R),nrow=1, ncol=5)#partial results matrix VII.

      #combine results by rows
      comb <- function(...) {
      mapply('rbind', ..., SIMPLIFY=FALSE)
      }
#parallel simulation start
        res<-foreach(i = 1:R,.combine = 'comb', .multicombine=TRUE, .options.RNG=1955) %dorng% {
        z<-simu.barma(n,phi=p,theta=q,prec=120,alpha=0.0) #generate observations of BARMA
        fit<-barma(z,ar=c(1),ma=c(NA),h=12,diag=0)        #fit model
        x<-fit$resid1                                     #residuals
        vectA<-fit$alpha                                  #alpha estimate
        vectP<-fit$phi                                    #phi estimate
        #vectT<-fit$theta                                 #theta estimate
        vectPR<-fit$prec                                  #precision estimate

        w<-1                                              #index control
        for(m in vm)                                      #lag values
        {
        #Ljung Box Test
        teste1<-Ljung.Box(x, lag=m, fitdf =f)
        LBPV[,w]<-teste1$p.value
```

```
  LBPVE[,w]<-teste1$statistic

  #Monti Test
  teste2<-Monti.test(x, lag=m, fitdf = f)
  MOPV[,w]<-teste2$p.value
  MOPVE[,w]<-teste2$statistic

  #Dufour and Roy Test
  teste3<-Dufour.test(x, lag=m, fitdf = f)
  DRPV[,w]<-teste3$p.value
  DRPVE[,w]<-teste3$statistic

  #Kwan and Sim(1) Test
  teste4<-Kwan.sim.test1(x, lag=m, fitdf = f)
  KS1PV[,w]<-teste4$p.value
  KS1PVE[,w]<-teste4$statistic

  #Kwan and Sim(4) Test
  teste5<-Kwan.sim.test4(x, lag=m, fitdf = f)
  KS2PV[,w]<-teste5$p.value
  KS2PVE[,w]<-teste5$statistic

  #Kwan and Sim(1) with partial autocorrelation
  teste6<-Kwan.sim.chest(x, lag=m, fitdf = f,type=typ,test=1)
  KS3PV[,w]<-teste6$p.value
  KS3PVE[,w]<-teste6$statistic

  #Kwan and Sim(4) with partial autocorrelation
  teste7<-Kwan.sim.chest(x, lag=m, fitdf = f,type=typ,test=4)
  KS4PV[,w]<-teste7$p.value
  KS4PVE[,w]<-teste7$statistic
  w<-w+1
  }

  list(LBPV,MOPV,DRPV,KS1PV,KS2PV,KS3PV,KS4PV,LBPVE,MOPVE,DRPVE,KS1PVE,KS2PVE,KS3PVE,KS4PVE,
       vectA,vectP,vectPR)
  }

  #conditional vectors
  for(i in 1:R)
  {
    for(w in 1:5)
    {
     rej1[i,w+5]<-as.integer(res[[1]][i,w]<va[2])
     rej2[i,w+5]<-as.integer(res[[2]][i,w]<va[2])
     rej3[i,w+5]<-as.integer(res[[3]][i,w]<va[2])
     rej4[i,w+5]<-as.integer(res[[4]][i,w]<va[2])
     rej5[i,w+5]<-as.integer(res[[5]][i,w]<va[2])
     rej6[i,w+5]<-as.integer(res[[6]][i,w]<va[2])
     rej7[i,w+5]<-as.integer(res[[7]][i,w]<va[2])
    }
  }

#output
#null rejection rates
#significance level - 5%
results[(16+((b-1)*24)-7),6:10]<-colSums(rej1[,6:10])/R #null rejection rates for ljung-box
results[(16+((b-1)*24)-6),6:10]<-colSums(rej2[,6:10])/R #null rejection rates for Monti
results[(16+((b-1)*24)-5),6:10]<-colSums(rej3[,6:10])/R #null rejection rates for Dufour and Roy
results[(16+((b-1)*24)-4),6:10]<-colSums(rej4[,6:10])/R #null rejection rates for Kwan and Sim(1)
results[(16+((b-1)*24)-3),6:10]<-colSums(rej5[,6:10])/R #null rejection rates for Kwan and Sim(4)
results[(16+((b-1)*24)-2),6:10]<-colSums(rej6[,6:10])/R #null rejection rates for Kwan and Sim(1p)
results[(16+((b-1)*24)-1),6:10]<-colSums(rej7[,6:10])/R #null rejection rates for Kwan and Sim(4p)

#Descriptive measures for vector alpha's
results2[(4*b-3),6]<-round(colMeans(res[[15]]),digits = 4)                          #mean
```

```
  results2[(4*b-3),7]<-round(colVars(res[[15]])+(colMeans(res[[15]])-0)^2,digits = 4) #MSE
  results2[(4*b-3),8]<-round(kurtosis(c(res[[15]])),digits=4)                           #kutosis
  results2[(4*b-3),9]<-round(skewness(c(res[[15]])),digits=4)                           #skewness

  #Descriptive measures for vector phi's
  results2[(4*b-2),6]<-round(colMeans(res[[16]]),digits=4)                              #mean
  results2[(4*b-2),7]<-round(var(c(res[[16]]))+(colMeans(res[[16]])-p)^2,digits = 4)   #MSE
  results2[(4*b-2),8]<-round(kurtosis(res[[16]]),digits=4)                              #kutosis
  results2[(4*b-2),9]<-round(skewness(res[[16]]),digits=4)                              #skewness
  results2[(4*b-2),10]<-round(abs((colMeans(res[[16]])-p))*100/p,digits=4)              #relative bies

  #Descriptive measures for vector theta's
  #results2[(4*b-1),6]<-round(colMeans(res[[16]]),digits=4)                             #mean
  #results2[(4*b-1),7]<-round(var(c(res[[16]]))+(colMeans(res[[16]])-q)^2,digits = 4)  #MSE
  #results2[(4*b-1),8]<-round(kurtosis(res[[16]]),digits=4)                             #kutosis
  #results2[(4*b-1),9]<-round(skewness(res[[16]]),digits=4)                             #skewness
  #results2[(4*b-1),10]<-round(abs((colMeans(res[[16]])-p))*100/p,digits=4)             #relative bies

  #Descriptive measures for vector precision's
  results2[(4*b-0),6]<-round(colMeans(res[[17]]),digits=4)                              #mean
  results2[(4*b-0),7]<-round(var(c(res[[17]]))+(colMeans(res[[17]])-120)^2,digits = 4)#MSE
  results2[(4*b-0),8]<-round(kurtosis(res[[17]]),digits=4)                              #kutosis
  results2[(4*b-0),9]<-round(skewness(res[[17]]),digits=4)                              #skewness
  results2[(4*b-0),10]<-round(abs((colMeans(res[[17]])-120))*100/120,digits=4)          #relative bies

  #quantiles
  for(k in 1:5)
  {
  #Ljung-Box
  results3[(21*b-20),k+5]<-colQuantiles(res[[8]],probs = c(0.99,0.95,0.90))[k,1]
  results3[(21*b-19),k+5]<-colQuantiles(res[[8]],probs = c(0.99,0.95,0.90))[k,2]
  results3[(21*b-18),k+5]<-colQuantiles(res[[8]],probs = c(0.99,0.95,0.90))[k,3]
  #Monti
  results3[(21*b-17),k+5]<-colQuantiles(res[[9]],probs = c(0.99,0.95,0.90))[k,1]
  results3[(21*b-16),k+5]<-colQuantiles(res[[9]],probs = c(0.99,0.95,0.90))[k,2]
  results3[(21*b-15),k+5]<-colQuantiles(res[[9]],probs = c(0.99,0.95,0.90))[k,3]
  #Dufour and Roy
  results3[(21*b-14),k+5]<-colQuantiles(res[[10]],probs = c(0.99,0.95,0.90))[k,1]
  results3[(21*b-13),k+5]<-colQuantiles(res[[10]],probs = c(0.99,0.95,0.90))[k,2]
  results3[(21*b-12),k+5]<-colQuantiles(res[[10]],probs = c(0.99,0.95,0.90))[k,3]
  #Kwan and Sim(1)
  results3[(21*b-11),k+5]<-colQuantiles(res[[11]],probs = c(0.99,0.95,0.90))[k,1]
  results3[(21*b-10),k+5]<-colQuantiles(res[[11]],probs = c(0.99,0.95,0.90))[k,2]
  results3[(21*b-9),k+5]<-colQuantiles(res[[11]],probs = c(0.99,0.95,0.90))[k,3]
  #Kwan and Sim(4)
  results3[(21*b-8),k+5]<-colQuantiles(res[[12]],probs = c(0.99,0.95,0.90))[k,1]
  results3[(21*b-7),k+5]<-colQuantiles(res[[12]],probs = c(0.99,0.95,0.90))[k,2]
  results3[(21*b-6),k+5]<-colQuantiles(res[[12]],probs = c(0.99,0.95,0.90))[k,3]
  #Kwan and Sim(1p)
  results3[(21*b-5),k+5]<-colQuantiles(res[[13]],probs = c(0.99,0.95,0.90))[k,1]
  results3[(21*b-4),k+5]<-colQuantiles(res[[13]],probs = c(0.99,0.95,0.90))[k,2]
  results3[(21*b-3),k+5]<-colQuantiles(res[[13]],probs = c(0.99,0.95,0.90))[k,3]
  #Kwan and Sim(4p)
  results3[(21*b-2),k+5]<-colQuantiles(res[[14]],probs = c(0.99,0.95,0.90))[k,1]
  results3[(21*b-1),k+5]<-colQuantiles(res[[14]],probs = c(0.99,0.95,0.90))[k,2]
  results3[(21*b-0),k+5]<-colQuantiles(res[[14]],probs = c(0.99,0.95,0.90))[k,3]
  }

print(results)
print(results2)
print(results3)
j<-1
b<-b+1
t<-t+1
}
}
```

```
}
```

A.2 A.2 Empirical powers without resorting to bootstrap resampling

```
####################################################################################################
PROGRAM: poderm1K.r

USE: The program computes the empirical powers of the portmanteau tests in the BARMA model.

AUTHOR: Vinícius Teodoro Scher

VERSION: 1.2

LAST MODIFIED: 23/03/2017
####################################################################################################

source("simu_barma.r") #function that generates observations of a BARMA model.
source("barma.r")      #function that estimates the parameters of the model.
source("barma.fit.r")  #function taht estimates the parameters using recursive derivatives.
source("ljung_box.r")  #function that performs the Ljung-Box test.
source("dufour.r")     #function that performs the Dufour and Roy test.
source("monti.r")      #function that performs the Ana Cristina Monti test.
source("kwan.r")       #function that performs the 1 de Kwan and Sim test.
source("kwan2.r")      #function that performs the 2 de Kwan and Sim test.
source("kwan3.r")      #function that performs the 3 de Kwan and Sim test.
source("kwan4.r")      #function that performs the 4 de Kwan and Sim test.
source("Kwan_Chest.R") #function that performs the Kwan and Sim tests with partial autocorrelation.

set.seed(1955)                                  #selection seed
R<-5000                                         #number of Monte Carlo replications.
vm<- c(seq(from = 3, to = 25, by = 1))          #number of lag considered.
vn<- c(50,250)                                  #sizes of samples.
va<-c(0.05)                                     #significance levels.
results<-matrix(rep(NA,2996),nrow=107,ncol=28)#matrix that stores empirical powers rates.
b<-1                                            #index control.
t<-1                                            #index control.
f<-1                                            #degress of freedom for chi-square distribution.
vp<-c(0.2,0.5,0.8)                              #vector for autoregressive parameters.
vq<-c(0.2,0.5,0.8)                              #vector for moving average parameters.
typ<-"partial"                                  #type of autocorrelation for Kwan_Chest function.

#Simulation start
  for(n in vn)
  {
    for(p in vp)
    {
      for(q in vq)
      {
      #empirical powers rates
      j<-1                                     #index control
      rej1<-matrix(rep(0,R),nrow=R, ncol=15) #partial results matrix I.
      rej2<-matrix(rep(0,R),nrow=R, ncol=15) #partial results matrix II.
      rej3<-matrix(rep(0,R),nrow=R, ncol=15) #partial results matrix III.
      rej4<-matrix(rep(0,R),nrow=R, ncol=15) #partial results matrix IV.
      rej5<-matrix(rep(0,R),nrow=R, ncol=15) #partial results matrix V.
      rej6<-matrix(rep(0,R),nrow=R, ncol=15) #partial results matrix VI.
      rej7<-matrix(rep(0,R),nrow=R, ncol=15) #partial results matrix VII.

      #p-values of tests
      LBPV<-matrix(rep(0,R),nrow=1, ncol=5)  #partial results matrix I.
      MOPV<-matrix(rep(0,R),nrow=1, ncol=5)  #partial results matrix II.
      DRPV<-matrix(rep(0,R),nrow=1, ncol=5)  #partial results matrix III.
      KS1PV<-matrix(rep(0,R),nrow=1, ncol=5) #partial results matrix IV.
      KS2PV<-matrix(rep(0,R),nrow=1, ncol=5) #partial results matrix V.
```

```
KS3PV<-matrix(rep(0,R),nrow=1, ncol=5) #partial results matrix VI.
KS4PV<-matrix(rep(0,R),nrow=1, ncol=5) #partial results matrix VII.

#Monte Carlo start
for(i in 1:R)
  {
    z<-simu.barma(n,phi=c(p),theta=c(q),prec=120,alpha=0.0)  #generate observations of BARMA
    fit<-barma(z,ar=c(1),ma=c(NA),h=12,diag=0)               #fit model
    x<-fit$resid1                                            #residuals

      w<-1                                                   #index control
      for(m in vm)                                           #lag values
      {
      #Ljung Box Test
      teste1<-Ljung.Box(x, lag=m, fitdf =f)
      LBPV[,w]<-teste1$p.value

      #Monti Test
      teste2<-Monti.test(x, lag=m, fitdf = f)
      MOPV[,w]<-teste2$p.value

      #Dufour and Roy Test
      teste3<-Dufour.test(x, lag=m, fitdf = f)
      DRPV[,w]<-teste3$p.value

      #Kwan and Sim(1) Test
      teste4<-Kwan.sim.test1(x, lag=m, fitdf = f)
      KS1PV[,w]<-teste4$p.value

      #Kwan and Sim(4) Test
      teste5<-Kwan.sim.test4(x, lag=m, fitdf = f)
      KS2PV[,w]<-teste5$p.value

      #Kwan and Sim(1) with partial autocorrelation
      teste6<-Kwan.sim.chest(x, lag=m, fitdf = f,type=typ,test=1)
      KS3PV[,w]<-teste6$p.value

      #Kwan and Sim(4) with partial autocorrelation
      teste7<-Kwan.sim.chest(x, lag=m, fitdf = f,type=typ,test=4)
      KS4PV[,w]<-teste7$p.value
      w<-w+1
      }

      rej1<-rej1+as.integer(LBPV<va)
      rej2<-rej2+as.integer(MOPV<va)
      rej3<-rej3+as.integer(DRPV<va)
      rej4<-rej4+as.integer(KS1PV<va)
      rej5<-rej5+as.integer(KS2PV<va)
      rej6<-rej6+as.integer(KS3PV<va)
      rej7<-rej7+as.integer(KS4PV<va)

    }

    #output
    #empirical powers rates
    #significance level - 5%
    for(j in 1:23)
    {
    results[(8+((b-1)*8)-7),j+5]<-sum(rej1[j])/R #empirical powers for ljung-box
    results[(8+((b-1)*8)-6),j+5]<-sum(rej2[j])/R #empirical powers for Monti
    results[(8+((b-1)*8)-5),j+5]<-sum(rej3[j])/R #empirical powers for Dufour and Roy
    results[(8+((b-1)*8)-4),j+5]<-sum(rej4[j])/R #empirical powers for Kwan and Sim(1)
    results[(8+((b-1)*8)-3),j+5]<-sum(rej5[j])/R #empirical powers for Kwan and Sim(4)
    results[(8+((b-1)*8)-2),j+5]<-sum(rej6[j])/R #empirical powers for Kwan and Sim(1p)
    results[(8+((b-1)*8)-1),j+5]<-sum(rej7[j])/R #empirical powers for Kwan and Sim(4p)
    }
```

```
      print(results)

   #columns labels
   for(g in 1:5)
   {
     results[6+((b-1)*6)-g,1]<-n
     results[6+((b-1)*6)-g,2]<-p
     results[6+((b-1)*6)-g,3]<-q
     results[6+((b-1)*6)-g,4]<-va
   }
      b<-b+1
  }
 }
}
```

APPENDIX B

Appendix B

In this appendix we modity the *tsdiag* function available in the *stats* package of the statistical software R. In our empirical application, presented in Section 2.5, we noted that such a function cannot be directly used with residuals from a fitted time series model because the number of degrees of freedom of the chi-square reference distribution is not correctly defined.

B.1 B.1 The modified *tsdiag* function

The *tsdiag* function obtains sample autocorrelations of the series by calling another function, namely: *Box.Test*. Such a function can be applied to residuals from a fitted time series model since it allows users to specify how many degrees of freedom should be subtracted from *m*. However, *tsdiag* uses *Box.Test* in its default mode, which perform any adjustment to the number of degrees of freedom of the χ^2 reference distribution. The *tsdiag* source code is:

```
function (object, gof.lag = 10, ...)
{
    oldpar <- par(mfrow = c(3, 1))
    on.exit(par(oldpar))
    rs <- object$residuals
    stdres <- rs/sqrt(object$sigma2)
    plot(stdres, type = "h", main = "Standardized Residuals", ylab = "")
    abline(h = 0)
    acf(object$residuals, plot = TRUE, main = "ACF of Residuals", na.action = na.pass)
    nlag <- gof.lag
    pval <- numeric(nlag)
    for (i in 1L:nlag) pval[i] <- Box.test(rs, i, type = "Ljung-Box")$p.value #HERE
    plot(1L:nlag, pval, xlab = "lag", ylab = "p value", ylim = c(0, 1),
    main = "p values for Ljung-Box statistic")
    abline(h = 0.05, lty = 2, col = "blue")
}
```

We marked the line that needs to be corrected '#HERE'. The *Box.text* function receives four arguments: first, the series (rs); second, number of sample autocorrelations to be used when computing the test statistic; third: the test statistic to be computed ("Ljung-Box"); fourth: how many degrees of freedom must be subtracted from *m* when the series is comprised of residuals from a fited model (empty, default=0). We recommend that the signaled line be altered so that the proper correction to the number of degrees of freedom is made. The correct code is:

```
function (object, gof.lag = 10, fitdf=0) #MODIFIED
{
    oldpar <- par(mfrow = c(3, 1))
    on.exit(par(oldpar))
    rs <- object$residuals
```

```
    stdres <- rs/sqrt(object$sigma2)
    plot(stdres, type = "h", main = "Standardized Residuals", ylab = "")
    abline(h = 0)
    acf(object$residuals, plot = TRUE, main = "ACF of Residuals", na.action = na.pass)
    nlag <- gof.lag
    pval <- numeric(nlag)
    for (i in 1L:nlag) pval[i] <- Box.test(rs, i, type = "Ljung-Box", fitdf)$p.value #MODIFIED
    plot(1L:nlag, pval, xlab = "lag", ylab = "p value", ylim = c(0, 1),
    main = "p values for Ljung-Box statistic")
    abline(h = 0.05, lty = 2, col = "blue")
}
```

With the proposed adjustment the *tsdiag* function can be used with residuals from a fitted time series model.