

Universidade Federal de Pernambuco  
Centro de Ciências Exatas e da Natureza  
Departamento de Estatística

Sadraque Eneas de Figueiredo Lucena

## **Testes de Hipóteses Não-Encaixadas em Regressão Beta**

*Trabalho apresentado ao Programa de Pós-Graduação em Estatística do Departamento de Estatística da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Estatística.*

Orientador: *Prof. Dr. Francisco Cribari Neto*

Recife  
2013

**Catálogo na fonte**  
**Bibliotecária Jane Souto Maior, CRB4-571**

**Lucena, Sadraque Eneas de Figueiredo**  
**Testes de hipóteses não-encaixadas em regressão**  
**beta. / Sadraque Eneas de Figueiredo Lucena. - Recife: O**  
**Autor, 2013.**  
**xi, 92 folhas: fig., tab.**

**Orientador: Francisco Cribari Neto.**  
**Dissertação (mestrado) - Universidade Federal de**  
**Pernambuco. CCEN, Estatística, 2013.**

**Inclui bibliografia e apêndice.**

**1. Análise de regressão. 2. Teste de hipóteses estatístico. 3.**  
**Regressão beta. 4. Bootstrap. I. Cribari Neto, Francisco**  
**(orientador). II. Título.**

**519.536**

**CDD (23. ed.)**

**MEI2013 – 044**

**Recife, 22 de fevereiro de 2013.**

Nós recomendamos que a dissertação de mestrado de autoria de

**Sdraque Eneias de Figueiredo Lucena**

Intitulada

**“Testes de Hipóteses Não-Encaixadas em Regressão Beta”**

Seja aceita como cumprimento parcial dos requerimentos para o grau de Mestre em Estatística.

---

Coordenador da Pós-Graduação em Estatística

Banca Examinadora:

---

Francisco Cribari Neto	Orientador / UFPE
------------------------	-------------------

---

Raydonal Ospina Martínez	UFPE
--------------------------	------

---

Silvia Lopes de Paula Ferrari	USP
-------------------------------	-----

Este documento será anexado à versão final da tese.

*Aos meus pais, Severino e Dorcas,  
e à minha irmã, Stefanny,  
dedico.*

# Agradecimentos

Ao professor Cribari, pela oportunidade concedida, por seus valiosos ensinamentos, paciência e excelente orientação. Ademais, por ser um exemplo de profissional, professor e pesquisador.

A Ana Hermínia, por estar comigo nessa jornada desde a graduação. Por partilhar todos os momentos de tensão e de alegria vivenciados no Residencial Laranjeiras. A Heloisa, Diêgo e Thiago, sem os quais as tardes de estudo não teriam o mesmo brilho. A Mariana, por suas contribuições imprescindíveis para o andamento de nossos estudos em probabilidade e inferência.

A Valéria, pela dedicação e exímia eficiência ao atender os alunos da pós-graduação. Aos professores do Departamento de Estatística da UFPE, em especial aos professores Raydonal Martínez, Alex Ramos, Audrey Cysneiros e Francisco Cysneiros, pelo conhecimento transmitido nas disciplinas. Aos professores do Departamento de Estatística da UFPB, sobretudo a Marcelo e Juliana, pelo apoio.

A Leyde, por estar sempre disposta a ouvir minhas explicações sobre esta dissertação. A Jobson, por estar presente em todos os momentos de alívio de estresse. A Mônica, por sempre me fazer sorrir com seu jeito *cult*. A Luana Caroline e ao pequeno Guilherme, por receber as visitas dos amigos e nos deixar perceber o surgimento de algo tão inspirador. A Edjackson, que, mesmo do outro lado do Atlântico, deu-me apoio incondicional. A Carolina, Pedro e Allyne, pelos momentos únicos.

A CAPES, pelo apoio financeiro.

Por fim, o agradecimento mais que especial aos meus pais, Severino e Dorcas, e à minha irmã, Stefanny, por serem toda minha estrutura.

*Não leia apenas com a mente.*  
—ECKHART TOLLE

# Resumo

Em determinados estudos, a modelagem de dados por meio de regressão pode levar a dois ou mais modelos com ajustes semelhantes, mas com especificações distintas. Se qualquer um dos modelos avaliados não pode ser obtido do outro por meio de restrições sobre os parâmetros que os indexam, estes são ditos não-encaixados. Em se tratando de modelos de regressão beta, que são adequados a dados cuja variável resposta varia no intervalo  $(0, 1)$ , os mesmos são não-encaixados quando os modelos diferem nos regressores e/ou nos casos em que há distinção na função de ligação associada ao preditor linear utilizado no submodelo da média ou da precisão. Nessas situações, os testes propostos para avaliar qual dos modelos está corretamente especificado não podem ser aplicados, pois são adequados apenas a modelos lineares de regressão. Neste sentido, a presente dissertação tem por objetivo apresentar uma adaptação no teste  $J$  – o mais utilizado em regressões na avaliação de hipóteses não-encaixadas – e em sua modificação, o teste  $MJ$ . Foram avaliados cenários em que os modelos diferem nos regressores e nas funções de ligação considerando-se pequenos tamanhos de amostra, uma vez que em amostras grandes as aproximações assintóticas são válidas. A avaliação foi baseada nas taxas de rejeição nula dos testes, sendo comparados os resultados dos testes com suas versões *bootstrap*. De acordo com os resultados, as taxas de rejeição dos testes tenderam aos valores desejados de acordo com o aumento do tamanho da amostra e a utilização de uma esquema *bootstrap* reduziu consideravelmente as distorções de tamanho.

**Palavras-chave:** Bootstrap, hipóteses não-encaixadas, regressão beta, teste  $J$ , teste  $MJ$ .

# Abstract

Oftentimes, practitioners have at their disposal two or more competing models with different parametric structures. If no model can be obtained as a particular case of the others when parametric restrictions are imposed on them we say the models are nonnested. Tests for choosing between nonnested linear regression models were proposed in the literature. Particularly useful tests are the  $J$  and  $MJ$  tests. In this thesis, we extend these two tests to the class of beta regression models, which is useful for modeling responses that assume values in the standard unit interval,  $(0, 1)$ . The tests can be used, for example, to choose between two beta regressions that use different link functions. We report the results of several Monte Carlo experiments which are designed to evaluate the tests finite sample performances. Bootstrap-based testing inference is also considered. Overall, the best performing test is the bootstrap  $MJ$  test. We also evaluate the accuracy of model selection using the  $MJ$  statistic.

**Keywords:** Beta regression, bootstrap,  $J$  test,  $MJ$  test, nonnested hypothesis tests.

# Lista de Figuras

2.1	Densidades beta para diferentes valores de $(p, q)$ .	6
2.2	Densidades beta para diferentes valores de $(\mu, \phi)$ .	7
4.1	Taxas de rejeição não-nula do teste $J$ e do teste $J bootstrap$ ao nível $\alpha = 5\%$ para modelos com diferença nos regressores.	30
4.2	Taxas de rejeição não-nula do teste $J$ e do teste $J bootstrap$ ao nível $\alpha = 5\%$ para modelos com diferença nas funções de ligação.	31
4.3	Taxas de rejeição não-nula do teste $MJ$ e do teste $MJ bootstrap$ ao nível $\alpha = 5\%$ para modelos com diferença nos regressores.	32
4.4	Taxas de rejeição não-nula do teste $MJ$ e do teste $MJ bootstrap$ ao nível $\alpha = 5\%$ para modelos com diferença nas funções de ligação.	33
5.1	Gráficos dos resíduos deviance com envelopes simulados para os modelos 1 e 2.	36
5.2	Gráficos dos resíduos deviance com envelopes simulados para os modelos $A$ e $B$ utilizados na modelagem de dados de habilidade de leitura em crianças disléxicas e não-disléxicas.	39

## Lista de Tabelas

4.1	Taxas de rejeição nula (%) no modelo com erro no regressor da média.	24
4.2	Taxas de rejeição nula (%) nos modelos com erro no regressor da precisão.	25
4.3	Taxas de rejeição nula (%) no modelo com erro no regressor da média e da precisão.	25
4.4	Taxas de rejeição nula (%) no modelo com erro na função de ligação da média.	26
4.5	Taxas de rejeição nula (%) no modelo com erro na função de ligação da precisão.	27
4.6	Taxas de rejeição nula (%) no modelo com erro na função de ligação da média e da precisão.	28
4.7	Percentuais em que o modelo correto foi selecionado via estatística $MJ$ ( $\hat{m} = m^*$ ) quando a hipótese nula não foi rejeitada.	29
5.1	Estimativas dos parâmetros dos modelos 1 e 2 utilizados na modelagem de dados de produção de gasolina.	35
5.2	Medidas de avaliação dos modelos 1 e 2.	36
5.3	$p$ -valores do teste $J$ usando as estatísticas da razão de verossimilhança ( $RV$ ) e de Wald e suas versões <i>bootstrap</i> .	37
5.4	Estimativas dos parâmetros dos modelos $A$ e $B$ utilizados na modelagem de dados de habilidade de leitura em crianças disléxicas e não-disléxicas.	38
5.5	Medidas de avaliação dos modelos $A$ e $B$ utilizados na modelagem de dados de habilidade de leitura em crianças disléxicas e não-disléxicas.	38
5.6	$p$ -valores dos testes $J$ e $MJ$ usando as estatísticas da razão de verossimilhança ( $RV$ ) e de Wald e suas versões <i>bootstrap</i> para os modelos com dados de habilidade de leitura em crianças disléxicas e não-disléxicas.	39

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Organização da dissertação	2
1.2	Suporte computacional	3
<b>2</b>	<b>Modelo de Regressão Beta</b>	<b>4</b>
2.1	Introdução	4
2.2	Distribuição Beta	4
2.3	Modelo de Regressão Beta	5
2.3.1	Reparametrização da distribuição beta	5
2.3.2	Regressão beta	6
<b>3</b>	<b>Testes de Hipóteses em Modelos Não-Encaixados</b>	<b>14</b>
3.1	Introdução	14
3.2	Seleção de modelos não-encaixados	14
3.3	Teste $J$	15
3.4	Teste $MJ$	17
3.5	Adaptação para regressão beta	18
3.6	Esquema bootstrap nos teste $J$ e $MJ$	19
<b>4</b>	<b>Avaliações Numéricas</b>	<b>22</b>
4.1	Introdução	22
4.2	Especificação incorreta de regressores	23
4.2.1	Regressores da média	23
4.2.2	Regressores da precisão	24
4.2.3	Regressores da média e da precisão	24
4.3	Especificação incorreta da função de ligação	26
4.3.1	Função de ligação da média	26
4.3.2	Função de ligação da precisão	26
4.3.3	Função de ligação da média e da precisão	27
4.3.4	Seleção do modelo correto pelo teste $MJ$	27
4.3.5	Poderes dos testes	29
<b>5</b>	<b>Aplicações</b>	<b>34</b>
5.1	Dados de Produção de Gasolina	34
5.2	Habilidade de leitura em crianças disléxicas e não-disléxicas	37

## SUMÁRIO

<b>6</b>	<b>Considerações Finais</b>	<b>40</b>
	<b>Referências Bibliográficas</b>	<b>42</b>
<b>A</b>	<b>Códigos em Ox</b>	<b>46</b>
A.1	Modelos com diferença nos regressores da média	46
A.2	Modelos com diferença nos regressores da dispersão	52
A.3	Modelos com diferença nos regressores da média e da dispersão	58
A.4	Modelos com diferença na função de ligação da média	64
A.5	Modelos com diferença na função de ligação da dispersão	71
A.6	Modelos com diferença nas funções de ligação da média e da dispersão	78
<b>B</b>	<b>Códigos em R</b>	<b>86</b>
B.1	Dados de Produção de Gasolina	86
B.2	Habilidade de leitura em crianças disléxicas e não-disléxicas	89

## CAPÍTULO 1

# Introdução

Modelos de regressão são utilizados para avaliar a relação entre variáveis, permitindo descrever essa relação por meio de uma equação. Embora seja amplamente usada em aplicações, a modelagem clássica é inapropriada em situações cujas suposições de normalidade e homoscedasticidade não são adequadas. Adicionalmente, esta modelagem não é aplicável a situações cujo suporte da variável resposta está restrito a um intervalo na reta, pois as estimativas obtidas podem fornecer valores ajustados fora do intervalo de variação da variável resposta. Neste sentido, Ferrari & Cribari-Neto (2004) propuseram a classe de modelos de regressão beta, adequada à modelagem de taxas e proporções, cujo suporte da variável resposta está restrito ao intervalo  $(0, 1)$ . Tais modelos são naturalmente heteroscedásticos.

A classe de modelos de regressão beta é baseada na suposição de que a variável resposta segue uma distribuição beta. Na especificação do modelo é utilizada uma parametrização alternativa que possibilita a modelagem direta da média da variável resposta ao invés dos parâmetros que indexam a distribuição. A especificação do modelo é semelhante à dos modelos lineares generalizados (MLGs), em que a média é relacionada a um preditor linear por meio de uma função de ligação. Pode-se utilizar ainda um modelo com dispersão variável, em que é considerado um submodelo para a média e outro para a precisão, fornecendo assim maior flexibilidade ao ajuste dos dados (Simas et al. 2010).

Após estimado o modelo de regressão, geralmente são aplicados testes para determinar se o modelo é compatível com os dados observados. Uma das possibilidades é realizar testes de hipóteses sobre modelos encaixados, isto é, modelos cuja representação da hipótese nula é um caso particular de um modelo mais geral especificado na hipótese alternativa. Pode ainda haver interesse em testar hipóteses envolvendo modelos não-encaixados. Nesses casos nenhum dos modelos pode ser obtido a partir da imposição de restrições sobre os parâmetros que indexam o outro modelo.

A literatura sobre testes não-encaixados tem origem nos trabalhos de Cox (1961, 1962), que abordou três ramos gerais (Pesaran & Weeks 2001): (i) o procedimento da razão de verossimilhanças modificadas (centrada) ou teste de Cox; (ii) a abordagem dos modelos abrangentes (*comprehensive models*), defendida por Atkinson (1970) e utilizada por Davidson & MacKinnon (1981) e o (iii) procedimento abrangente (*encompassing procedure*), inicialmente considerado por Deaton (1982) e Dastoor (1983) e ampliado por Gouriéroux et al. (1983) e Mizon & Richard (1986).

O teste  $J$ , proposto por Davidson & MacKinnon (1981), é atualmente o procedimento mais utilizado para testar modelos de regressão não-encaixados (McAleer 1995, Godfrey 2011). Esse teste se baseia em um modelo construído artificialmente de modo que cada modelo em consideração possa ser obtido deste modelo geral como um caso particular (Davidson & Mac-

## 1.1 ORGANIZAÇÃO DA DISSERTAÇÃO

Kinnon 2009). Algumas adaptações do teste  $J$  podem ser encontradas na literatura. MacKinnon et al. (1983) adaptaram o teste  $J$  para diferentes transformações na variável resposta. Wooldridge (1990) modificou o teste de modo a torná-lo robusto sob heterogeneidade de forma desconhecida. Sapa (2008) apresentou uma modificação que torna mais robusto o teste  $J$  na presença de outliers. Ramalho et al. (2011) avaliaram o desempenho de testes de hipóteses não-encaixadas para escolha da função de ligação em modelos com variável resposta binária ( $y = 0$  ou  $y = 1$ ).

Um ponto negativo do teste  $J$  é que na presença de vários modelos não-encaixados surge a necessidade de uma sequência de testes em que cada hipótese nula (o  $m$ -ésimo modelo está corretamente especificado) é considerada. Neste sentido, Hagemann (2012) propôs o teste  $MJ$ , utilizado para verificar se o modelo corretamente especificado está presente entre os modelos testados, evitando assim uma sequência de testes e possíveis distorções de tamanho. Adicionalmente, o teste  $MJ$  tende a selecionar o modelo corretamente especificado consoante aumenta o tamanho da amostra.

Os modelos de regressão beta não-encaixados podem apresentar diferenças quanto à presença de regressores, à função de ligação ou ambos simultaneamente. Nesses casos, os testes tradicionalmente usados para testar hipóteses encaixadas, tais como os testes de Wald, escore e da razão de verossimilhanças, não são aplicáveis. Esta dissertação tem por objetivo apresentar uma adaptação nos testes  $J$  e  $MJ$  para regressão beta, bem como avaliar os desempenhos dos mesmos. Inferência *bootstrap* também é considerada.

### 1.1 Organização da dissertação

Esta dissertação é constituída por seis capítulos. No segundo capítulo é apresentada uma parametrização alternativa da distribuição beta, a classe de modelos de regressão beta, como são estimados os parâmetros do modelo e como são construídos os intervalos de confiança. Adicionalmente, é apresentada uma medida de ajuste global, o pseudo- $R^2$ .

No Capítulo 3 são introduzidos os modelos não-encaixados. É abordado o teste  $J$  para modelos lineares e o teste  $MJ$  proposto por Hagemann (2012). É apresentada ainda a adaptação para a classe de modelos de regressão beta e as respectivas versões *bootstrap*.

No Capítulo 4 são avaliados os desempenhos dos testes em modelos que diferem nos regressores ou nas funções de ligações da média, da precisão e de ambos conjuntamente. A avaliação é realizada com base nas taxas de rejeição nula dos testes obtidos via simulação de Monte Carlo. É também avaliado o desempenho do teste  $MJ$  na seleção do modelo correto, além dos resultados de simulação de poder dos dois testes.

No Capítulo 5 são apresentadas aplicações a dados reais. Na primeira aplicação é utilizado um conjunto de dados em que os modelos avaliados diferem na função de ligação da média e na segunda são abordados modelos que distinguem nos regressores do submodelo da precisão. Por fim, no Capítulo 6 são apresentadas as principais conclusões do estudo.

## 1.2 Suporte computacional

As avaliações numéricas apresentadas ao longo deste trabalho foram realizadas em linguagem Ox versão 6.21 para sistema operacional Linux. Ox é uma linguagem matricial de programação desenvolvida por Jurgen Doornik e se encontra disponível para uso acadêmico no sítio <http://www.doornik.com>. Maiores detalhes sobre Ox podem ser encontrados em Doornik (2009).

As análises realizadas nas aplicações foram produzidas em ambiente R versão 2.14.1 para sistema operacional Linux. R é um sistema integrado para análise estatística e produção de gráficos, estando disponível para *download* gratuito em <http://www.r-project.org>. Para mais detalhes, ver Venerables et al. (2009).

A presente dissertação foi produzida usando o sistema tipográfico L<sup>A</sup>T<sub>E</sub>X (Lamport 1986), que consiste em uma série de macros e rotinas baseadas no sistema T<sub>E</sub>X, criado por Donald Knuth. Especificamente, este trabalho foi tipografado na classe UFPETthesis, desenvolvida por Paulo G. S. Fonseca e disponível em <http://www.cin.ufpe.br/~paguso/ufpethe> sis. Detalhes sobre o sistema tipográfico L<sup>A</sup>T<sub>E</sub>X podem ser encontrados em Mittelbach et al. (2004) ou no sítio <http://www.tex.ac.uk/CTAN/latex>.

## Modelo de Regressão Beta

### 2.1 Introdução

Quando se deseja modelar variáveis aleatórias contínuas restritas ao intervalo  $(0, 1)$ , tais como taxas e proporções, o modelo de regressão linear clássico não é adequado, uma vez que o ajuste de tal modelo pode fornecer valores preditos fora do suporte da variável resposta (Kieschnick & McCullough 2003). Adicionalmente, medidas como taxas e proporções tendem a apresentar heteroscedasticidade e assimetria, de modo que as inferências resultantes do modelo linear clássico podem levar a conclusões incorretas por terem suas suposições violadas.

Uma tentativa de resolver os problemas citados anteriormente consiste em transformar a variável dependente para que esta assuma valores na reta e então estimar o modelo de regressão linear. Entretanto, com esta abordagem os parâmetros do modelo são interpretáveis em função da média da variável transformada, e não da variável original. Ainda, utilizar uma transformação nem sempre garante a estabilização da variância (Cribari-Neto & Zeileis 2010).

Outra abordagem foi introduzida por Ferrari & Cribari-Neto (2004) e se baseia em um modelo cuja variável resposta tem distribuição beta. A classe de modelos de regressão beta utiliza uma reparametrização da distribuição beta indexada pela média e por um parâmetro de precisão. Esta apresenta vantagens sobre a modelagem clássica, uma vez que a densidade pode assumir diferentes formas, tendo, inclusive, a distribuição uniforme como um caso particular (Vasconcellos & Cribari-Neto 2005). Bury (1999) listou diversas aplicações da distribuição beta na engenharia e Johnson et al. (1995) apresentaram aplicações em que o uso da distribuição beta era o mais adequado para ajustar taxas e proporções. A seguir é apresentada a distribuição beta, a reparametrização utilizada por Ferrari & Cribari-Neto (2004), o modelo de regressão beta e algumas de suas propriedades.

### 2.2 Distribuição Beta

Seja  $Y$  uma variável aleatória (v.a.) com distribuição beta de parâmetros  $p$  e  $q$ , denotado por  $Y \sim \mathfrak{B}(p, q)$ . A função de densidade de  $Y$  é definida por

$$f(y; p, q) = \frac{\Gamma(p+q)}{\Gamma(p)\Gamma(q)} y^{p-1} (1-y)^{q-1}, \quad 0 < y < 1, \quad (2.1)$$

em que  $p, q > 0$  e  $\Gamma(\cdot)$  é a função gama. A função de distribuição de  $Y$  é dada por

$$F(y; p, q) = \frac{\Gamma(p+q)}{\Gamma(p)\Gamma(q)} \int_0^y t^{p-1} (1-t)^{q-1} dt.$$

## 2.3 MODELO DE REGRESSÃO BETA

O  $r$ -ésimo momento não-central é

$$\mu_r = \mathbb{E}(Y^r) = \int_0^1 y^r \frac{\Gamma(p+q)}{\Gamma(p)\Gamma(q)} y^{p-1} (1-y)^{q-1} = \frac{B(p+r, q)}{B(p, q)} = \frac{\Gamma(p+q)\Gamma(p+r)}{\Gamma(p)\Gamma(p+r+q)},$$

em que  $B(\cdot, \cdot)$  é a função beta. Logo, a média de  $Y \sim \mathfrak{B}(p, q)$  é dada por

$$\mathbb{E}(Y) = \frac{p}{p+q} \quad (2.2)$$

e a variância é dada por

$$\text{Var}(Y) = \frac{pq}{(p+q)^2(p+q+1)}. \quad (2.3)$$

Se  $p, q > 1$ , a moda existe e é dada por

$$\text{Moda}(Y) = \frac{p-1}{p+q-2}.$$

De acordo com os valores de  $p$  e  $q$ , a densidade beta pode assumir diferentes formas no intervalo  $(0, 1)$ . A Figura 2.1 contém densidades beta para alguns valores de  $(p, q)$ . Note que para  $p, q < 1$  a densidade tem forma de “U”, quando  $p < q$  há assimetria à esquerda e quando  $p > q$  há assimetria à direita. Quando  $p = q = 1$ , obtém-se a distribuição uniforme padrão.

## 2.3 Modelo de Regressão Beta

### 2.3.1 Reparametrização da distribuição beta

Ao introduzir o modelo de regressão beta, Ferrari & Cribari-Neto (2004) consideraram uma reparametrização da densidade apresentada em (2.1) que permite modelar a média da variável resposta envolvendo parâmetros da regressão e um parâmetro de precisão (ou dispersão), de forma similar aos modelos lineares generalizados (McCullagh & Nelder 1989). Sejam  $\mu = p/(p+q)$  e  $\phi = p+q$ , então  $p = \mu\phi$  e  $q = (1-\mu)\phi$ . Assim, a densidade de  $Y$  pode ser reescrita da seguinte forma:

$$f(y; \mu, \phi) = \frac{\Gamma(\phi)}{\Gamma(\mu\phi)\Gamma((1-\mu)\phi)} y^{\mu\phi-1} (1-y)^{(1-\mu)\phi-1}, \quad (2.4)$$

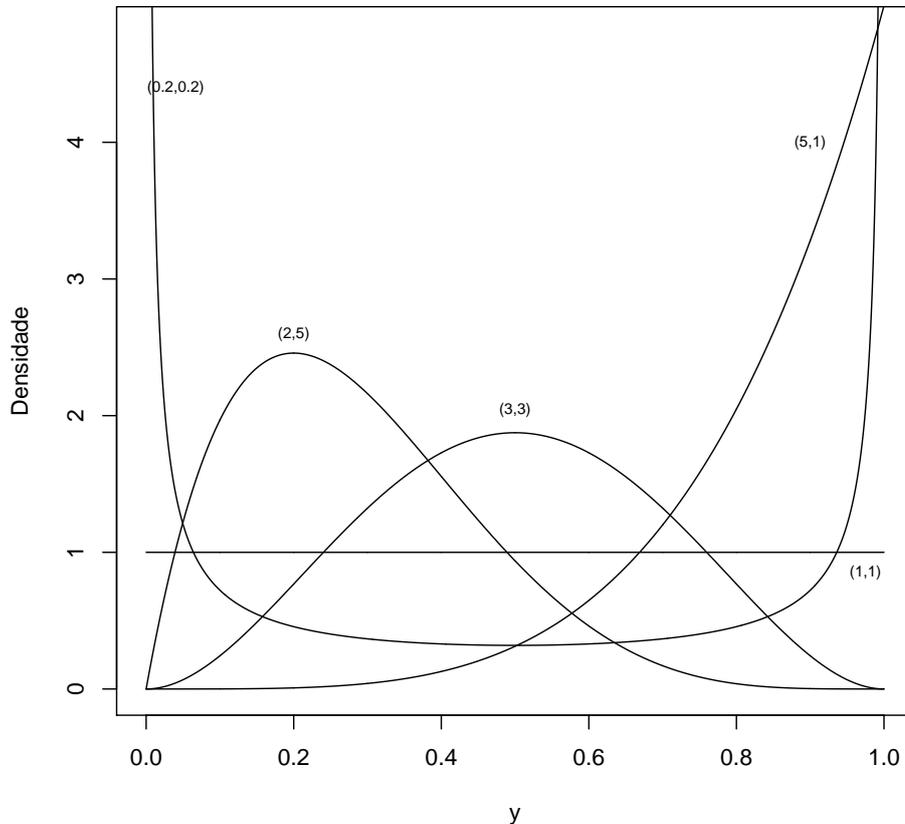
em que  $y \in (0, 1)$ ,  $\mu \in (0, 1)$  e  $\phi > 0$ . A partir de (2.2) e (2.3) segue que

$$\mathbb{E}(Y) = \mu$$

e

$$\text{Var}(Y) = \frac{V(\mu)}{1+\phi},$$

## 2.3 MODELO DE REGRESSÃO BETA



**Figura 2.1** Densidades beta para diferentes valores de  $(p, q)$ .

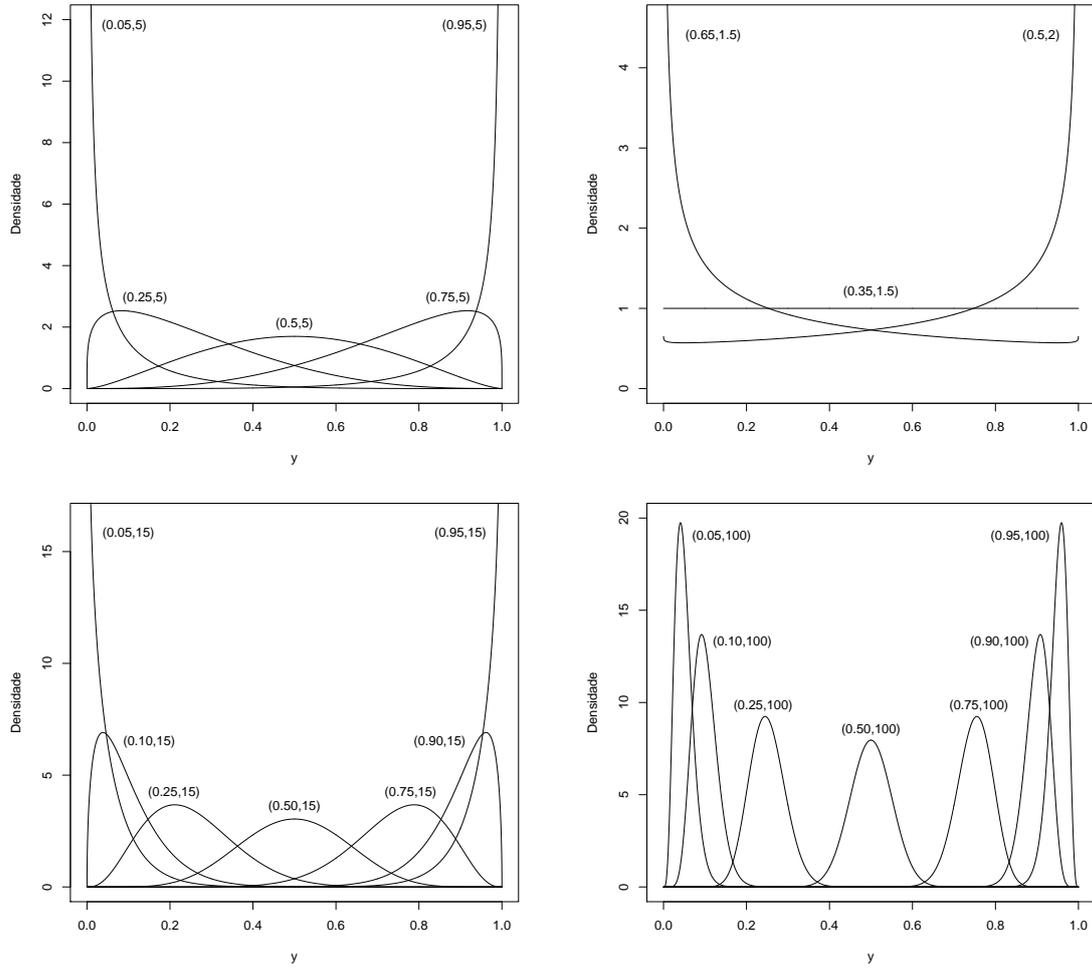
em que  $V(\mu) = \mu(1 - \mu)$  é a função de variância. Portanto,  $\mu$  é a média da variável aleatória  $Y$  e  $\phi$  pode ser interpretado como um parâmetro de precisão, uma vez que, para  $\mu$  fixo, quanto maior o valor de  $\phi$ , menor a variância de  $Y$ .

Na Figura 2.2 são apresentadas densidades beta considerando a nova parametrização. Como se pode observar, a densidade pode assumir forma de “J”, “J” invertido e de “U”. Adicionalmente, ela é simétrica quando  $\mu = 1/2$  e assimétrica quando  $\mu \neq 1/2$ . Ainda, para  $\mu$  fixo, a dispersão da distribuição diminui à medida que  $\phi$  aumenta e para  $\mu = 1/2$  e  $\phi = 2$  a densidade se reduz à uniforme padrão (Ferrari & Cribari-Neto 2004).

### 2.3.2 Regressão beta

Sejam  $y_1, \dots, y_n$  variáveis aleatórias independentes, de modo que cada  $y_t$  tenha distribuição beta reparametrizada,  $\mathfrak{B}(\mu_t, \phi)$ , isto é, com média  $\mu_t$ ,  $t = 1, \dots, n$ , e parâmetro de precisão  $\phi$  (desconhecido e constante para todo  $t$ ). Ferrari & Cribari-Neto (2004) propuseram a modelagem da média  $\mu_t$  por meio de uma função de ligação associada ao preditor linear  $\eta_t$ , de modo

## 2.3 MODELO DE REGRESSÃO BETA



**Figura 2.2** Densidades beta para diferentes valores de  $(\mu, \phi)$ .

a satisfazer a seguinte relação funcional:

$$g(\mu_t) = \eta_t = \sum_{i=1}^k x_{ti} \beta_i = x_t^\top \beta, \quad (2.5)$$

em que  $\beta = (\beta_1, \dots, \beta_k)^\top$  é o vetor de parâmetros desconhecidos ( $\beta \in \mathbb{R}^k$ ),  $x_t = (x_{t1}, \dots, x_{tk})^\top$  é o vetor de observações de  $k$  ( $k < n$ ) variáveis explicativas (covariáveis), cujos valores são assumidos fixos e conhecidos, e  $g: (0, 1) \mapsto \mathbb{R}$  é uma função de ligação estritamente monótona e duas vezes diferenciável.

Com base no modelo (2.5), temos que

$$\mu_t = g^{-1}(\eta_t) \quad (2.6)$$

e

$$\text{Var}(y_t) = \frac{\mathbb{V}(g^{-1}(\eta_t))}{1 + \phi},$$

### 2.3 MODELO DE REGRESSÃO BETA

ou seja, a variância de  $y_t$  depende de  $\mu_t$  e, portanto, não é constante para todo  $t$ . Logo, o modelo de regressão beta incorpora naturalmente a heteroscedasticidade característica de dados de taxas e proporções.

Há diversas possibilidades para escolha da função de ligação  $g(\cdot)$ . Na prática, são comumente utilizadas a função logística:  $g(\mu_t) = \log[\mu_t/(1 - \mu_t)]$ ; a função probit ou função normal inversa:  $g(\mu_t) = \Phi^{-1}(\mu_t)$ , em que  $\Phi(\cdot)$  é a função de distribuição da normal padrão; a função log-log:  $g(\mu_t) = -\log\{-\log(\mu_t)\}$ ; a função log-log complementar:  $g(\mu) = \log\{-\log(1 - \mu)\}$  e a função Cauchy:  $g(\mu) = \tan\{\phi(\mu_t - 0.5)\}$ .

Uma vez que a variável aleatória  $y_t$  pertence ao intervalo  $(0, 1)$ , sua média,  $\mu_t$ , está restrita ao mesmo intervalo. Logo, o emprego da função de ligação  $g(\cdot)$  permite que cada  $\beta_j$  possa assumir qualquer valor em  $\mathbb{R}$ , sem restrições. Isto é,  $\mu_t \in (0, 1)$ , mas  $g(\mu_t) = \eta_t$ ,  $\eta_t \in \mathbb{R}$ . Aqui, não há problemas em relação à interpretação dos parâmetros em função da variável resposta, uma vez que a função  $g$  é aplicada sobre  $\mu_t$  e não sobre  $y_t$ .

No modelo proposto por Ferrari & Cribari-Neto (2004), o parâmetro de precisão,  $\phi$ , é assumido constante. Entretanto, em algumas aplicações a dispersão (recíproco da precisão) pode ser variável e, se não considerada esta característica no modelo, as inferências resultantes podem estar incorretas. Neste contexto, Simas et al. (2010) propuseram uma extensão do modelo de regressão beta ao adicionar uma estrutura de regressão para o parâmetro de precisão juntamente à modelagem da média. Aqui, para uma amostra  $y_1, \dots, y_n$ , assume-se que cada  $y_t$  tem distribuição beta com média  $\mu_t$  e parâmetro de precisão  $\phi_t$ , ou seja,  $y_t \sim \mathfrak{B}(\mu_t, \phi_t)$ . Supõe-se então que  $\phi_t$  satisfaz a seguinte relação funcional:

$$h(\phi_t) = \vartheta_t = \sum_{j=1}^r z_{tj} \gamma_j = z_t^\top \gamma, \quad (2.7)$$

em que  $\vartheta_t$  é o preditor linear,  $\gamma = (\gamma_1, \dots, \gamma_r)^\top$  é um vetor de parâmetros desconhecidos ( $\gamma \in \mathbb{R}^r$ ),  $z_t = (z_{t1}, \dots, z_{tr})^\top$  é o vetor de  $r$  ( $r < n - k$ ) covariáveis, cujos valores são fixos e conhecidos, e  $h: (0, +\infty) \mapsto \mathbb{R}$  é uma função estritamente monótona e duas vezes diferenciável. Usualmente as funções de ligação utilizadas são a função logarítmica,  $h(\phi_t) = \log(\phi_t)$ , e a raiz quadrada,  $h(\phi_t) = \sqrt{\phi_t}$ .

Os desenvolvimentos abordados a seguir estão apresentados em Simas et al. (2010). No modelo de regressão beta, os parâmetros são estimados conjuntamente por máxima verossimilhança. O logaritmo da função de verossimilhança baseado em  $n$  observações independentes da distribuição beta, dado por

$$\ell(\beta, \gamma) = \sum_{t=1}^n \ell_t(\mu_t, \phi_t), \quad (2.8)$$

em que

$$\begin{aligned} \ell_t(\mu_t, \phi_t) = & \log \Gamma(\phi_t) - \log \Gamma(\mu_t \phi_t) - \log \Gamma\{(1 - \mu_t) \phi_t\} + \\ & (\mu_t \phi_t - 1) \log y_t + \{(1 - \mu_t) \phi_t - 1\} \log(1 - y_t). \end{aligned}$$

Como foi definido em (2.5) e (2.7),  $\mu_t = g^{-1}(\eta_t)$  e  $\phi_t = h^{-1}(\vartheta_t)$  são funções de  $\beta$  e  $\gamma$ , respectivamente, e as estimativas de máxima verossimilhança são únicas sob condições de regularidade (Rocha & Simas 2011). Os componentes do vetor escore são obtidos a partir da

### 2.3 MODELO DE REGRESSÃO BETA

diferenciação da função de log-verossimilhança com respeito aos parâmetros  $(\beta, \gamma)$  e são dados por  $U = (U_\beta(\beta, \gamma), U_\gamma(\beta, \gamma))^\top$ . Para  $i = 1, \dots, k$ , temos

$$U_\beta(\beta, \gamma) = \frac{\partial \ell(\beta, \gamma)}{\partial \beta_i} = \sum_{t=1}^n \frac{\partial \ell(\mu_t, \phi_t)}{\partial \mu_t} \frac{\partial \mu_t}{\partial \eta_t} \frac{\partial \eta_t}{\partial \beta_i}, \quad (2.9)$$

em que  $\partial \mu_t / \partial \eta_t = 1/g'(\mu_t)$ , com  $g'(\mu_t)$  sendo a primeira derivada da função  $g$  avaliada em  $\mu_t$ ,  $\partial \eta_t / \partial \beta_i = x_{ti}$  e

$$\frac{\partial \ell(\mu_t, \phi_t)}{\partial \mu_t} = \phi_t \left[ \log \left( \frac{y_t}{1-y_t} \right) - \{ \psi(\mu_t \phi_t) - \psi((1-\mu_t)\phi_t) \} \right], \quad (2.10)$$

em que  $\psi(\cdot)$  é a função digama. Tomando  $y_t^* = \log\{y_t/(1-y_t)\}$  e  $\mu_t^* = \psi(\mu_t \phi_t) - \psi((1-\mu_t)\phi_t)$ , pode-se reescrever (2.10) da seguinte forma:

$$\frac{\partial \ell(\mu_t, \phi_t)}{\partial \mu_t} = \phi_t (y_t^* - \mu_t^*).$$

Assim, de (2.9) obtemos

$$U_\beta(\beta, \gamma) = \sum_{t=1}^n \phi_t (y_t^* - \mu_t^*) \frac{1}{g'(\mu_t)} x_{ti}. \quad (2.11)$$

Matricialmente temos

$$U_\beta(\beta, \gamma) = X^\top \Phi T (y^* - \mu^*), \quad (2.12)$$

em que  $X$  é uma matriz  $n \times k$ , com  $x_t$  sendo a  $t$ -ésima linha da matriz  $X$ ,  $T = \text{diag}\{1/g'(\mu_1), \dots, 1/g'(\mu_n)\}$ ,  $y^* = (y_1^*, \dots, y_n^*)^\top$ ,  $\mu^* = (\mu_1^*, \dots, \mu_n^*)^\top$  e

$$\Phi = \text{diag}\{\phi_1, \dots, \phi_n\}. \quad (2.13)$$

De modo semelhante, para  $j = 1, \dots, q$ , temos

$$U_\gamma(\beta, \gamma) = \frac{\partial \ell(\beta, \gamma)}{\partial \gamma_j} = \sum_{t=1}^n \frac{\partial \ell_t(\mu_t, \phi_t)}{\partial \phi_t} \frac{\partial \phi_t}{\partial \vartheta_t} \frac{\partial \vartheta_t}{\partial \gamma_j}, \quad (2.14)$$

em que  $\partial \phi_t / \partial \vartheta_t = 1/h'(\phi_t)$ ,  $h'(\phi_t)$  sendo a primeira derivada da função  $h$  avaliada em  $\phi_t$ , para  $t = 1, \dots, n$ ,  $\partial \vartheta_t / \partial \gamma_j = z_{tj}$  e

$$\frac{\partial \ell_t(\mu_t, \phi_t)}{\partial \phi_t} = \mu_t (y_t^* - \mu_t^*) + \log(1-y_t) - \psi((1-\mu_t)\phi_t) + \psi(\phi_t).$$

Portanto, a equação (2.14) se torna

$$U_\gamma(\beta, \gamma) = \sum_{t=1}^n [\mu_t (y_t^* - \mu_t^*) + \log(1-y_t) - \psi((1-\mu_t)\phi_t) + \psi(\phi_t)] \frac{1}{h'(\phi_t)} z_{tj}. \quad (2.15)$$

### 2.3 MODELO DE REGRESSÃO BETA

Matricialmente temos

$$U_\gamma(\beta, \gamma) = Z^\top H a, \quad (2.16)$$

em que  $Z$  é uma matriz  $n \times r$  cuja  $t$ -ésima linha contém o vetor de observações  $z_t^\top$ ,  $H = \text{diag}\{1/h'(\phi_1), \dots, 1/h'(\phi_n)\}$  e  $a = (a_1, \dots, a_n)^\top$ , com  $a_t = \partial \ell_t(\mu_t, \phi_t) / \partial \phi_t$ .

A matriz de informação observada,  $O$ , é obtida da segunda derivada da função de log-verossimilhança com relação aos parâmetros desconhecidos. Ela é definida como

$$O = O(\beta, \gamma) = \begin{pmatrix} O_{\beta\beta} & O_{\beta\gamma} \\ O_{\gamma\beta} & O_{\gamma\gamma} \end{pmatrix},$$

em que

$$O_{\beta\beta} = X^\top \Phi S X,$$

sendo  $\Phi$  definida em (2.13),  $S = \text{diag}\{s_1, \dots, s_n\}$ , em que

$$s_t = \left\{ \phi_t [\psi'(\mu_t \phi_t) + \psi'((1 - \mu_t) \phi_t)] + (y^* - \mu^*) \frac{g''(\mu_t)}{g'(\mu_t)} \right\} \frac{1}{[g'(\mu_t)]^2},$$

$\psi'(\cdot)$  sendo a função trigama e  $g''(\mu_t)$  denotando a segunda derivada da função  $g$  avaliada em  $\mu_t$ ;

$$O_{\beta\gamma} = O_{\gamma\beta}^\top = X^\top F T H Z,$$

em que  $F = \text{diag}\{f_1, \dots, f_n\}$ , com

$$f_t = c_t - (y^* - \mu^*)$$

e

$$c_t = \phi_t \{ \psi'(\mu_t \phi_t) \mu_t - \psi'((1 - \mu_t) \phi_t) (1 - \mu_t) \}; \quad (2.17)$$

$T = \text{diag}\{1/g'(\mu_1), \dots, 1/g'(\mu_n)\}$  e  $H = \text{diag}\{1/h'(\phi_1), \dots, 1/h'(\phi_n)\}$ ;

$$O_{\gamma\gamma} = Z^\top \mathcal{V} Z,$$

em que  $\mathcal{V} = \text{diag}\{v_1, \dots, v_n\}$ , com

$$v_t = d_t + a_t \{ h''(\phi_t) / [h'(\phi_t)]^3 \}$$

e

$$d_t = [\psi'(\mu_t \phi_t) \mu_t^2 + \psi'((1 - \mu_t) \phi_t) (1 - \mu_t)^2 - \psi'(\phi_t)] / \{h'(\phi_t)\}^2. \quad (2.18)$$

A matriz de informação de Fisher é então obtida tomando-se a esperança da matriz observada,  $O$ , e é dada por

$$K = K(\beta, \gamma) = \begin{pmatrix} K_{\beta\beta} & K_{\beta\gamma} \\ K_{\gamma\beta} & K_{\gamma\gamma} \end{pmatrix},$$

em que

### 2.3 MODELO DE REGRESSÃO BETA

$$K_{\beta\beta} = X^\top \Phi W X,$$

sendo  $W = \text{diag}\{w_1, \dots, w_n\}$ , com

$$w_t = \phi_t \{ \psi'(\mu_t \phi_t) + \psi'((1 - \mu_t) \phi_t) \} \frac{1}{[g'(\mu_t)]^2}; \quad (2.19)$$

$$K_{\beta\gamma} = X^\top C T H Z,$$

com  $C = \text{diag}\{c_1, \dots, c_n\}$  e  $c_t$  definido em (2.17);

$$K_{\gamma\gamma} = Z^\top D Z,$$

em que  $D = \text{diag}\{d_1, \dots, d_n\}$  e  $d_t$  definido em (2.18).

Quando a dispersão do modelo é constante, isto é,  $\phi_1 = \dots = \phi_n = \phi$ , as expressões (2.17), (2.18) e (2.19) se reduzem, respectivamente, a

$$c_t = \phi \{ \psi'(\mu_t \phi) \mu_t - \psi'((1 - \mu_t) \phi) (1 - \mu_t) \},$$

$$d_t = \psi'(\mu_t \phi) \mu_t^2 + \psi'((1 - \mu_t) \phi) (1 - \mu_t)^2 - \psi'(\phi)$$

e

$$w_t = \phi \{ \psi'(\mu_t \phi) + \psi'((1 - \mu_t) \phi) \} \frac{1}{[g'(\mu_t)]^2}.$$

Os estimadores de máxima verossimilhança,  $\hat{\beta}$  e  $\hat{\gamma}$ , são então obtidos da solução do sistema

$$\begin{cases} U_\beta(\beta, \gamma) = 0 \\ U_\gamma(\beta, \gamma) = 0 \end{cases}.$$

Este sistema não possui solução em forma fechada e as estimativas de máxima verossimilhança são obtidas por meio da maximização numérica da função de log-verossimilhança. A maximização deve ser realizada usando um algoritmo de otimização não linear, tais como algoritmos baseados no método de Newton ou algoritmos quasi Newton, e.g., BFGS; mais detalhes podem ser encontrados em Nocedal & Wright (1999).

Os algoritmos de otimização requerem a especificação de valores iniciais a serem usados no esquema iterativo. Ferrari & Cribari-Neto (2004) propuseram utilizar como valores iniciais para  $\beta$  as estimativas de mínimos quadrados obtidas da regressão linear entre as respostas transformadas  $g(y_1), \dots, g(y_n)$  e  $X$ , isto é, utilizar  $(X^\top X)^{-1} X^\top z$ , em que  $z = (g(y_1), \dots, g(y_n))^\top$ . Para o caso em que a dispersão do modelo é constante, o chute inicial para  $\phi$  é dado por

$$\frac{1}{n} \sum_{t=1}^n \frac{\check{\mu}_t (1 - \check{\mu}_t)}{\check{\sigma}_t^2} - 1, \quad (2.20)$$

em que  $\check{\mu}_t$  é obtido aplicando-se  $g^{-1}(\cdot)$  ao  $t$ -ésimo valor ajustado da regressão linear entre  $g(y_1), \dots, g(y_n)$  e  $X$ , isto é,

$$\check{\mu}_t = g^{-1}(x_t^\top (X^\top X)^{-1} X^\top z)$$

### 2.3 MODELO DE REGRESSÃO BETA

e

$$\hat{\sigma}_t^2 = \frac{\check{e}^\top \check{e}}{(n-k)\{g'(\check{\mu}_t)\}^2},$$

onde  $\check{e} = z - X(X^\top X)^{-1}X^\top z$  é o vetor de resíduos de mínimos quadrados ordinários da regressão linear estimada usando a resposta transformada.

Para o caso em que a dispersão não é considerada constante, Cribari-Neto & Zeileis (2010) sugeriram utilizar a expressão (2.20) como estimativa para o intercepto (levando-se em conta a função de ligação) e os demais coeficientes de  $\hat{\gamma}^{(0)}$  iguais a zero. Isto é, utilizar o chute inicial considerando a dispersão constante.

É importante salientar que, diferentemente dos modelos lineares generalizados, na classe de modelos de regressão beta os parâmetros  $\beta$  e  $\gamma$  não são ortogonais.

Sob condições de regularidade e para amostras de tamanho grande, os estimadores de máxima verossimilhança têm distribuição aproximadamente normal:

$$\begin{pmatrix} \hat{\beta} \\ \hat{\gamma} \end{pmatrix} \sim N_{k+r} \left( \begin{pmatrix} \beta \\ \gamma \end{pmatrix}, K^{-1} \right),$$

em que

$$K^{-1} = K^{-1}(\beta, \gamma) = \begin{pmatrix} K^{\beta\beta} & K^{\beta\gamma} \\ K^{\gamma\beta} & K^{\gamma\gamma} \end{pmatrix}$$

é a inversa da informação de Fisher e foi obtida por Simas et al. (2010) por meio de uma expressão padrão para a inversa de matrizes particionadas (Rao 1973), cujos blocos são dados por

$$K^{\beta\beta} = (X^\top \Phi W X - X^\top C T H Z (Z^\top D Z)^{-1} Z^\top H T C^\top X)^{-1}, \quad (2.21)$$

$$K^{\beta\gamma} = (K^{\gamma\beta})^\top = -K^{\beta\beta} X^\top C T H Z (Z^\top D Z)^{-1}$$

e

$$K^{\gamma\gamma} = (Z^\top D Z)^{-1} \{I_r + (Z^\top H T C^\top X) K^{\beta\beta} X^\top C T H Z (Z^\top D Z)^{-1}\}, \quad (2.22)$$

onde  $I_r$  é uma matriz identidade de ordem  $r$ .

Os limites do intervalo de confiança de cobertura  $(1 - \alpha)$ ,  $0 < \alpha < 1/2$ , para  $\beta_i$ ,  $i = 1, \dots, k$ , e  $\gamma_j$ ,  $j = 1, \dots, r$  são dados, respectivamente, por

$$\hat{\beta}_i \pm \Phi^{-1}(1 - \alpha/2) \text{ep}(\hat{\beta}_i)$$

e

$$\hat{\gamma}_j \pm \Phi^{-1}(1 - \alpha/2) \text{ep}(\hat{\gamma}_j)$$

em que  $\Phi^{-1}(\cdot)$  é a função quantil normal padrão e  $\text{ep}(\hat{\beta}_i)$  e  $\text{ep}(\hat{\gamma}_j)$  são, respectivamente, os erros-padrão assintóticos de  $\hat{\beta}_i$  e  $\hat{\gamma}_j$ . Estes são obtidos, respectivamente, da raiz quadrada do  $i$ -ésimo elemento da diagonal da matriz  $\hat{K}^{\beta\beta}$ , definida em (2.21) e da raiz quadrada do  $j$ -ésimo elemento da diagonal da matriz  $\hat{K}^{\gamma\gamma}$ , definida em (2.22). Assim, um intervalo de confiança de

## 2.3 MODELO DE REGRESSÃO BETA

cobertura aproximadamente  $(1 - \alpha) \times 100\%$  para a média da variável resposta dado um vetor de covariáveis  $x_t^\top$ , isto é, um intervalo de confiança para  $\mu_t$ , é dado por

$$[g^{-1}(\hat{\eta}_t - \Phi^{-1}(1 - \alpha/2)\text{ep}(\hat{\eta}_t)), \quad g^{-1}(\hat{\eta}_t + \Phi^{-1}(1 - \alpha/2)\text{ep}(\hat{\eta}_t))],$$

em que  $\hat{\eta}_t = x_t^\top \hat{\beta}$  e o erro-padrão assintótico de  $\hat{\eta}_t$  é  $\text{ep}(\hat{\eta}_t) = \{x_t^\top \hat{K}^{\beta\beta}(\hat{\beta})x_t\}^{\frac{1}{2}}$ . Detalhes sobre o comportamento das estimativas em pequenas amostras e estratégias de estimação pontual e intervalar podem ser encontrados em Ospina et al. (2006).

Uma medida para avaliar o ajuste global de modelos de regressão beta é o pseudo- $R^2$  ( $R_p^2$ ), definido como o quadrado do coeficiente de correlação entre  $\hat{\eta}$  e  $g(y)$ . Note que  $R_p^2 \in [0, 1]$  e quanto mais próximo de 1, melhor o ajuste do modelo (Vasconcellos & Cribari-Neto 2005).

No que diz respeito à análise de resíduos e de diagnóstico, Rocha & Simas (2011) estenderam resultados apresentados por Ferrari & Cribari-Neto (2004) e Espinheira et al. (2008) ao considerar modelos com estrutura de regressão para o parâmetro de precisão. Simas et al. (2010) consideraram modelos de regressão beta não-lineares e correção de viés para as estimativas do modelo.

A distribuição beta apresentada na Seção 2.3.1 não é adequada para dados que assumem valores no intervalo  $(0, 1]$ ,  $[0, 1)$  ou  $[0, 1]$ . Neste contexto, Ospina & Ferrari (2010) propuseram a distribuição beta inflacionada, que é uma composição da distribuição beta e a distribuição de Bernoulli degenerada no ponto  $\zeta$ , com  $\zeta = 0$  ou  $\zeta = 1$ . Ospina & Ferrari (2012) apresentaram modelos de regressão beta inflacionados, seus aspectos inferenciais e medidas de diagnóstico.

# Testes de Hipóteses em Modelos Não-Encaixados

## 3.1 Introdução

Modelos de regressão não-encaixados são tais que um não pode ser obtido como caso particular do outro a partir da imposição de restrições sobre os parâmetros do modelo. Nesses casos o teste  $J$  proposto por Davidson & MacKinnon (1981) é adequado para decidir qual dos modelos está corretamente especificado. A seguir são apresentados o teste  $J$  para modelos lineares de regressão, uma modificação sugerida por Hagemann (2012), a adaptação para a classe de modelos de regressão beta e suas versões *bootstrap*.

## 3.2 Seleção de modelos não-encaixados

Suponha que são observadas  $M$  ( $M \geq 2$ ) matrizes de regressores que dão origem a  $M$  modelos de regressão linear distintos associados a  $y = (y_1, \dots, y_n)^\top$ , isto é,

$$y = X_m \theta_m + u_m, \quad m \in \mathcal{M} = \{1, \dots, M\}, \quad (3.1)$$

em que  $X_m$  é a matriz de regressores do  $m$ -ésimo modelo e tem dimensão  $n \times k_m$ ;  $\theta_m$  é um vetor contendo  $k_m$  parâmetros desconhecidos e  $u_m$  é um vetor de erros identicamente distribuídos com média zero e matriz de covariâncias  $\sigma_m^2 I$ , isto é,  $u_m \sim \text{IID}(0, \sigma_m^2 I)$ . Uma vez que os modelos são não-encaixados, para quaisquer duas matrizes  $X_m$  e  $X_l$ ,  $m \neq l$ , uma não pode ser obtida por meio de uma transformação linear da outra, o que implica que há pelo menos um regressor em  $X_m$  que não pertence a  $X_l$  e vice-versa.

Com base nos modelos supracitados, há interesse em identificar a existência de uma matriz  $X_{m^*}$  que corresponde ao modelo correto  $m^*$ . Formalmente, são testadas as hipóteses

$$\begin{aligned} H_1 : & \quad y = X_1 \theta_1 + u_1 \\ H_2 : & \quad y = X_2 \theta_2 + u_2 \\ & \quad \vdots \\ H_M : & \quad y = X_M \theta_M + u_M \end{aligned} \quad (3.2)$$

e uma forma de testá-las consiste em utilizar um modelo artificial ampliado que compreenda todos os modelos em (3.2). O modelo ampliado pode ser escrito como

$$y = \left( 1 - \sum_{\substack{l=1 \\ l \neq m}}^M \lambda_l \right) X_m \theta_m + \sum_{\substack{l=1 \\ l \neq m}}^M \lambda_l X_l \theta_l + u, \quad (3.3)$$

### 3.3 TESTE $J$

em que  $\lambda_l$ ,  $l = 1, \dots, M$ ,  $\lambda_l \in \mathbb{R}^{M-1}$ , é um escalar. No caso de serem testadas apenas duas hipóteses,  $H_1$  e  $H_2$ , o modelo artificial é

$$y = (1 - \lambda)X_1\theta_1 + \lambda X_2\theta_2 + u.$$

Quando  $\lambda = 0$ , temos  $H_1$  e quando  $\lambda = 1$ , temos  $H_2$ . Assim, para testar  $H_1$  basta estimar o modelo ampliado e testar  $\lambda = 0$ . Analogamente, para  $M$  modelos,  $H_m$  é o modelo correto se  $\lambda_l = 0$ ,  $\forall l \neq m$ , na Equação (3.3). No entanto, estimar este modelo não é possível, uma vez que nem todos os parâmetros são identificáveis. No modelo artificial há  $k_1 + \dots + k_M + M - 1$  parâmetros a serem estimados, mas o número de parâmetros que podem ser identificados é igual à dimensão do subespaço gerado por  $X_1, \dots, X_M$ , que é no máximo  $k_1 + \dots + k_M$ .

Uma alternativa a (3.3) é utilizar a regressão inclusiva, na qual se utiliza o seguinte modelo:

$$y = X_m\theta'_m + \sum_{\substack{l=1 \\ l \neq m}}^M X'_l\theta'_l + u, \quad (3.4)$$

em que  $X'_l$ ,  $l = 1, \dots, M$ ,  $l \neq m$ , é uma matriz  $n \times k'_l$  que contém  $k'_l$  colunas de regressores ou quaisquer possíveis combinações lineares que não pertencem a qualquer outra matriz  $X_m$ . Assim, a dimensão desse espaço é  $k'_1 + \dots + k'_m + \dots + k'_M$ , os parâmetros do modelo podem ser estimados por mínimos quadrados ordinários (MQO) e a hipótese de que  $\theta'_l = 0$  pode ser testada por meio de um teste  $F$ .

Para o caso não-linear, as hipóteses são dadas por

$$\begin{aligned} H_1 : & y = x_1(\theta_1) + u_1 \\ H_2 : & y = x_2(\theta_2) + u_2 \\ & \vdots \\ H_M : & y = x_M(\theta_M) + u_M \end{aligned} \quad (3.5)$$

e, para quaisquer  $\theta_m$  e  $\theta_l$ ,  $m \neq l$ , não é possível determinar valores tais que  $x_m(\theta_m) = x_l(\theta_l)$ . Em outras palavras, um modelo não pode ser escrito como caso particular do outro a menos que sejam impostas restrições em ambos. Aqui, o modelo artificial ampliado é

$$y = \left( 1 - \sum_{\substack{l=1 \\ l \neq m}}^M \lambda_l \right) x_m(\theta_m) + \sum_{\substack{l=1 \\ l \neq m}}^M \lambda_l x_l(\theta_l) + u.$$

Embora o teste  $F$  para testar  $\theta'_l = 0$  na regressão inclusiva (3.4) seja recomendado, o mesmo não vale para testar hipóteses relacionadas a modelos não lineares, pois a generalização não é trivial ou satisfatória. Adicionalmente, esse teste tem menor poder que o teste  $J$  (Davidson & MacKinnon 2009), que será apresentado a seguir.

### 3.3 Teste $J$

No caso linear, Davidson & MacKinnon (1981) propuseram uma alternativa que consiste em substituir por estimativas consistentes os parâmetros do Modelo (3.3) que não estão sendo

### 3.3 TESTE $J$

testados. Assim, para testar  $H_m$ , basta substituir  $\theta_l$  por  $\hat{\theta}_l$ , o vetor de parâmetros estimados por MQO sob o modelo  $H_l$ , isto é,  $\hat{\theta}_l = (X_l^\top X_l)^{-1} X_l^\top y$ . Redefinindo os erros apropriadamente, obtém-se o modelo

$$y = \left( 1 - \sum_{\substack{l=1 \\ l \neq m}}^M \lambda_l \right) X_m \theta_m + \sum_{\substack{l=1 \\ l \neq m}}^M \lambda_l X_l \hat{\theta}_l + u. \quad (3.6)$$

No caso de se testar duas hipóteses,  $H_1$  e  $H_2$ , o modelo (3.6) fica

$$y = X_1 \theta_1 + \lambda X_2 \hat{\theta}_2 + u. \quad (3.7)$$

Após estimado o modelo (3.6), é realizado um teste em busca de evidências de que  $\theta_l = 0$ . Então, o teste  $J$  para testar a validade do modelo  $m$  na presença de  $M - 1$  modelos alternativos nada mais é que o teste  $t$  para testar  $\theta_l = 0$ ,  $l = 1, \dots, M$ ,  $l \neq m$ , cuja estatística é

$$J_m = \frac{y' P_l M_m y}{\hat{\sigma}_J \sqrt{y' P_l M_m P_l y}}, \quad (3.8)$$

em que  $P_l = X_m (X_m^\top X_m)^{-1} X_m^\top$ ,  $M_m = I_n - P_m$ . Aqui,  $I_n$  é a matriz identidade de ordem  $n$  e  $\hat{\sigma}_J$  é o estimador de MQO do desvio-padrão do erro na regressão (Michelis 1999).

No caso não-linear, a Equação (3.6) é substituída por

$$y = \left( 1 - \sum_{\substack{l=1 \\ l \neq m}}^M \lambda_l \right) x_m(\theta_m) + \sum_{\substack{l=1 \\ l \neq m}}^M \lambda_l \hat{x}_l + u,$$

em que  $\hat{x}_l = x_l(\hat{\theta}_l)$ ,  $\hat{\theta}_l$  é o vetor de estimativas de mínimos quadrados ponderados obtidas sob o modelo  $H_l$  e o procedimento utilizado é semelhante ao caso linear.

Ao aplicar o teste  $J$ , o sucesso de um modelo ocorre quando os modelos concorrentes não fornecem melhorias em seu poder de explicação. Caso haja aumento notável no poder explicativo do modelo, este é rejeitado. Assim, o método adiciona variáveis artificiais, obtidas dos valores ajustados pelos modelos concorrentes, e testa a significância da adição (Fingleton 2005). Entretanto, rejeitar a hipótese de que  $H_m$  é correto não é suficiente, pois não significa que os modelos concorrentes são adequados. Desta forma, deve-se aplicar o teste  $J$  de forma sequencial para testar cada modelo. No caso de duas hipóteses,  $H_1$  e  $H_2$ , o procedimento se dá em duas partes e pode ser estendido para  $M > 2$  modelos:

#### 1. Teste $H_1$ contra $H_2$

- Estime  $H_2$  e obtenha os valores ajustados ( $\hat{y}_2 = X_2 \hat{\theta}_2$ );
- Inclua  $\hat{y}_2$  como regressor em  $H_1$ ;
- Estime o modelo ampliado (3.7) e teste a exclusão da variável adicionada. A rejeição da hipótese  $\lambda = 0$  implica a rejeição de  $H_1$ .

### 3.4 TESTE $MJ$

#### 2. Teste $H_2$ contra $H_1$

- Estime  $H_1$  e obtenha os valores ajustados ( $\hat{y}_1 = X_1 \hat{\theta}_1$ );
- Inclua  $\hat{y}_1$  como regressor em  $H_2$ ;
- Estime o modelo ampliado dado por

$$y = X_2 \theta_2 + \lambda' \hat{y}_1 + u,$$

em que  $\lambda' = 1 - \lambda$ , e teste a exclusão da variável adicionada. A rejeição da hipótese  $\lambda' = 0$  implica a rejeição de  $H_2$ .

A partir desse procedimento é possível rejeitar todos os modelos, bem como não rejeitar múltiplos modelos. Embora tais possibilidades não sejam naturais e sejam até mesmo indesejáveis, segundo Davidson & MacKinnon (2009) elas podem ser bastante informativas. Segundo os autores, se ambos modelos são rejeitados, então é necessário encontrar outro modelo que apresente melhor ajuste; caso nenhum deles seja rejeitado, os dados são compatíveis com ambas hipóteses. Nestas situações, uma alternativa é utilizar o teste  $MJ$ , descrito a seguir.

### 3.4 Teste $MJ$

O teste  $J$  possibilita concluir que mais de um modelo está corretamente especificado. Além disso, a necessidade de usar testes sequenciais ao se realizar o teste  $J$  ocorre porque este teste não possui uma hipótese nula natural. Hagemann (2012) propôs então o teste  $MJ$  a partir de uma modificação no teste de Davidson & MacKinnon (1981). O novo teste parte da ideia de que se, dentre  $M$  modelos testados, o modelo  $m^*$  é o modelo correto, então a estatística  $J$  correspondente tem uma distribuição assintótica, enquanto as estatísticas dos outros modelos divergem; se, entretanto, o modelo correto não está sendo testado, todas as estatísticas divergem. Logo, apenas o modelo com menor valor da estatística  $J$  pode ser o modelo correto e podemos rejeitar a hipótese de que o modelo correto está especificado se a menor das estatísticas  $J$  é grande. Desta forma, o teste  $MJ$  é útil para testar a presença do modelo correto quando se deseja avaliar vários modelos concorrentes, evitando assim testes sequenciais (Hagemann 2012).

Para construir a estatística de teste para as hipóteses em (3.2), dados  $m \in \mathcal{M} = \{1, \dots, M\}$  modelos, sejam

$$\omega_m = n^{-1/2} \left( y^\top P_l M_m y \right)_{l \in \mathcal{M} \setminus \{m\}}$$

e

$$\hat{\Sigma}_m = n^{-1} \left( y^\top P_l M_m \hat{\Omega}_m M_m P_{l'} y \right)_{l, l' \in \mathcal{M} \setminus \{m\}},$$

em que  $P_m = X_m (X_m^\top X_m)^{-1} X_m^\top$ ,  $M_m = I_n - P_m$  e  $\hat{\Omega}_m = \text{diag}(\hat{u}_{1,m}^2, \dots, \hat{u}_{n,m}^2)$ , com  $\hat{u}_{i,m}^2 = y_i - x_{i,m}^\top \hat{\theta}_m$ , é uma matriz diagonal de resíduos quadrados. Hagemann (2012) define a estatística  $J^2$  para o modelo  $m$  como sendo

$$J_m^2 = \omega_m^\top \hat{\Sigma}_m^{-1} \omega_m, \quad (3.9)$$

ou seja, o quadrado da estatística  $J$  definida por Davidson & MacKinnon (1981).

### 3.5 ADAPTAÇÃO PARA REGRESSÃO BETA

Sob as condições de regularidade dadas em Hagemann (2012), se o modelo correto  $m^*$  está sendo testado, a estatística  $J_{m^*}^2$  tem distribuição assintótica  $\chi^2$  com  $M - 1$  graus de liberdade. Além disso, para cada modelo  $m \in \mathcal{M} \setminus \{m^*\}$  e para todo  $B \in \mathbb{R}$ ,  $\lim_{n \rightarrow \infty} \mathbb{P}(J_m \geq B) = 0$ . Logo, o modelo com menor estatística  $J$  é o único candidato possível a ser o modelo correto. A aplicação do teste  $MJ$  utiliza o seguinte procedimento:

1. Para cada  $m \in \mathcal{M}$ , estimar o Modelo (3.6), calcular a estatística  $J_m^2$  definida em (3.9) e obter  $MJ = \min\{J_1^2, \dots, J_M^2\}$ .
2. Rejeite a hipótese  $H_0 : m^* \in \mathcal{M}$  (o modelo correto está especificado) em favor de  $H_1 : m^* \notin \mathcal{M}$  (o modelo correto não está especificado) se  $MJ > \chi_{1-\alpha, M-1}^2$ , em que  $\chi_{1-\alpha, M-1}^2$  é o quantil  $1 - \alpha$  da distribuição  $\chi_{M-1}^2$ .

O teste  $MJ$  é um teste de união-interseção (Berger 1982): a hipótese nula é a interseção de diferentes hipóteses nulas (um dentre os modelos em  $\mathcal{M}$  é o modelo correto) e a região de rejeição é a interseção das regiões de rejeição das respectivas hipóteses nulas (a hipótese nula é falsa se todas as hipóteses forem falsas).

O teste  $MJ$  escolhe o modelo  $\hat{m}$  para ser testado e, caso haja evidências de que  $m^* \in \mathcal{M}$ , então  $\hat{m}$  é o candidato natural a  $m^*$ . Segundo (Hagemann 2012), o teste  $MJ$  encontra de forma consistente o modelo correto entre os modelos testados, pois

$$\lim_{n \rightarrow \infty} \mathbb{P}(\hat{m} = m^*) = 1$$

e

$$\lim_{n \rightarrow \infty} \mathbb{P}(\hat{m} = m^* | MJ \leq \chi_{1-\alpha, M-1}^2) = 1.$$

### 3.5 Adaptação para regressão beta

O teste  $J$  se utiliza de um modelo ampliado criado a partir da adição de uma variável artificial, cuja exclusão é testada utilizando o teste  $t$  para hipóteses encaixadas ( $H_0 : \lambda_l = 0$  contra  $H_1 : \lambda_l \neq 0$ ). Baseado neste procedimento, no teste  $MJ$  comparamos o valor da estatística de teste a valores críticos obtidos da distribuição  $\chi^2$ . Entretanto, estes testes na forma que foram apresentados nas Seções 3.3 e 3.4 devem ser adaptados pois não são adequados aos modelos de regressão beta. Para testar a exclusão de variáveis em modelos de regressão beta, Ferrari & Cribari-Neto (2004) apresentaram os testes da razão de verossimilhanças, escore e Wald, que são amplamente utilizados por suas propriedades de otimalidade.

A classe de modelos de regressão beta se utiliza de uma modelagem para a média e outra para o parâmetro de precisão. Nestas circunstâncias, modelos não-encaixados podem diferir na especificação dos regressores do submodelo da média, da precisão ou de ambos. O mesmo ocorre para funções de ligação. Suponha que há interesse em testar dois modelos de regressão beta não-encaixados com diferença nos regressores da média cujas hipóteses testadas são

$$H_1 : \begin{aligned} g(\mu) &= \eta_1 = X_1 \beta_1 \\ h(\phi) &= \vartheta = Z \gamma \end{aligned} \quad \text{e} \quad H_2 : \begin{aligned} g(\mu) &= \eta_2 = X_2 \beta_2 \\ h(\phi) &= \vartheta = Z \gamma, \end{aligned} \quad (3.10)$$

em que  $\mu$  e  $\phi$  são vetores de médias e precisões, respectivamente;  $g : (0, 1) \mapsto \mathbb{R}$  e  $h : (0, +\infty) \mapsto \mathbb{R}$  são funções de ligação;  $\eta_1$ ,  $\eta_2$  e  $\vartheta$  são preditores lineares;  $X_1$ ,  $X_2$  e  $Z$  são matrizes de regressores e  $\beta_1$ ,  $\beta_2$  e  $\gamma$  são vetores de parâmetros. Para proceder o teste  $J$  para testar  $H_1$ , estimam-se os parâmetros do modelo  $H_2$  por máxima verossimilhança e em seguida inclui-se  $\hat{\eta}_2$  em  $H_1$  como regressor de  $g(\mu)$ . Então a estatística  $J$ , para testar a exclusão de  $\hat{\eta}_2$  é definida como

$$J_1 = 2 \left\{ \ell(\hat{\beta}, \hat{\gamma}) - \ell(\tilde{\beta}, \tilde{\gamma}) \right\}, \quad (3.11)$$

em que  $\ell(\beta, \gamma)$  é o logaritmo da função de verossimilhança definido em (2.8) e  $(\hat{\beta}, \hat{\gamma})$  e  $(\tilde{\beta}, \tilde{\gamma})$  são os estimadores de máxima verossimilhança de  $(\beta, \gamma)$  para o modelo ampliado e o modelo dado em  $H_1$ , respectivamente. Como a estatística da razão de verossimilhanças tem distribuição nula assintótica  $\chi^2$  com número de graus de liberdade correspondente ao número de restrições em teste, rejeita-se  $H_1$  ao nível de significância  $\alpha$  se  $J_1 > \chi_{1-\alpha, 1}^2$ . As estatísticas score e Wald também podem ser utilizadas como estatística  $J$ . Para testar  $H_2$ , acrescenta-se  $\hat{\eta}_1$  como regressor de  $g(\mu)$  em  $H_2$  e compara-se o valor da estatística  $J_2$  ao valor crítico  $\chi_{1-\alpha, 1}^2$ .

Para  $M$  modelos, a adaptação do teste proposto por Hagemann (2012) consiste em obter  $MJ = \min\{J_1, \dots, J_M\}$ , em que  $J_m$ ,  $m = 1, \dots, M$ , é a estatística da razão de verossimilhanças, e comparar seu valor ao ponto crítico da distribuição  $\chi^2$  com  $M - 1$  graus de liberdade. Assim, para testar dois modelos, calcula-se a estatística  $J$  dada em (3.11) correspondente a cada um deles e obtém-se  $MJ = \min\{J_1, J_2\}$ . Em seguida, compara-se  $MJ$  com  $\chi_{1-\alpha, 1}^2$ . Caso  $MJ < \chi_{1-\alpha, 1}^2$ , o modelo correto é um dos modelos testados. Se  $MJ > \chi_{1-\alpha, 1}^2$ , nenhum dos modelos especificados é o modelo correto. Adicionalmente, o teste pode ser utilizado para selecionar o modelo corretamente especificado, de modo a ser considerado o modelo correto aquele que corresponde à menor das estatísticas. Esta estratégia fornece o modelo corretamente especificado à medida em que o tamanho da amostra aumenta (Hagemann 2012). Para os demais casos, o procedimento se dá de forma semelhante.

### 3.6 Esquema bootstrap nos teste $J$ e $MJ$

Em pequenas amostras, a aproximação da distribuição nula exata da estatística  $J$  pela distribuição normal padrão pode ser pobre. O teste  $MJ$  tende a apresentar melhoras na distorção de tamanho presente no teste  $J$ , entretanto, a qualidade da inferência produzida pelos dois testes pode ser melhorada com base em valores críticos obtidos a partir de um esquema de reamostragem *bootstrap*, como sugerido por Fan & Li (1995), Godfrey (1998) e Hagemann (2012). Estudos de simulação de Queiroz (2011) mostram que os testes da razão de verossimilhanças, score e Wald, que podem ser utilizados na adaptação dos testes  $J$  e  $MJ$  para regressão beta, também apresentam melhor desempenho com um esquema *bootstrap*.

Seja  $y = (y_1, \dots, y_n)^\top$  uma amostra aleatória de uma população. Esta amostra pode ser entendida como realizações de variáveis aleatórias independentes e identicamente distribuídas,  $Y_1, \dots, Y_n$ , com função de distribuição  $F$  e densidade  $f$ . Ao serem realizadas inferências sobre uma determinada característica da população, é utilizada uma estatística  $T(Y, F)$ , cuja distribuição amostral pode ser aproximada por várias técnicas, inclusive *bootstrap*.

Introduzido por Efron (1979), o método *bootstrap* busca estimar  $F$  a partir de um grande

número de pseudo-amostras,  $y^* = (y_1^*, \dots, y_n^*)^\top$  que são baseadas na amostra original. A partir da função de distribuição estimada,  $\hat{F}$ , estima-se a distribuição amostral de  $T(Y^*, \hat{F})$ , possibilitando, portanto, a realização de inferências sobre  $T(Y, F)$ .

O método *bootstrap* pode ser classificado em paramétrico e não-paramétrico. No esquema *bootstrap* paramétrico, as amostras artificiais são geradas a partir da função de distribuição  $F$  tendo como parâmetros as estimativas obtidas com base na amostra original, ou seja, como  $F_\theta$  é desconhecido, utiliza-se  $\hat{F} = F_{\hat{\theta}}$  na geração dos dados. Classifica-se como *bootstrap* não-paramétrico o esquema em que não se tem informação acerca da função de distribuição  $F$  e as amostras artificiais são obtidas por sorteio com reposição da amostra original, sendo pois geradas de  $\hat{F}$ , a função de distribuição empírica.

O método *bootstrap* é um procedimento simples e automático que permite obter medidas de interesse sem a necessidade de cálculos analíticos complexos, embora seja computacionalmente custoso. Essa técnica pode ser aplicada a problemas estatísticos como correção de viés, testes de hipóteses, construção de intervalos de confiança e obtenção de erros-padrão. No caso de testes de hipóteses, o método *bootstrap* geralmente produz um refinamento assintótico quando a estatística de teste é assintoticamente pivotal (Beran 1988). Desta forma, se comparado aos testes assintóticos, um esquema de reamostragem *bootstrap* possibilita a redução nas distorções de tamanho em pequenas amostras.

Seja  $y = (y_1, \dots, y_n)^\top$  um vetor de variáveis aleatórias independentes tal que  $y_t \sim \mathfrak{B}(\mu_t, \phi_t)$ ,  $t = 1, \dots, n$ , sejam  $X_1, X_2$  e  $Z$  matrizes de regressores e sejam  $\eta_1, \eta_2$  e  $\vartheta$  preditores lineares dos submodelos da média e da precisão, respectivamente. O procedimento *bootstrap* paramétrico para testar a hipótese  $H_1$  dada em (3.10) utilizando a estatística  $J$  adaptada consiste dos seguintes passos:

1. Estimar o modelo  $H_2$ , incluir  $\hat{\eta}_2$  no modelo  $H_1$  e estimar o modelo artificial ampliado;
2. Calcular a estatística  $J$  descrita na Seção 3.5;
3. Gerar uma amostra *bootstrap*  $y^*$ , sob  $H_1$ , com  $y_t^* \stackrel{ind}{\sim} \mathfrak{B}(\hat{\mu}_{1t}, \hat{\phi}_t)$ , em que  $\hat{\mu}_{1t} = g^{-1}(x_{1t} \hat{\beta}_1)$  e  $\hat{\phi}_t = h^{-1}(z_t \hat{\gamma}_t)$ . Aqui,  $\hat{\beta}_1$  e  $\hat{\gamma}_t$  são as estimativas de máxima verossimilhança para os parâmetros do modelo  $H_1$ ;
4. Estimar o modelo artificial ampliado usando  $y^*$  como variável resposta e calcular a estatística  $J^*$ ;
5. Executar  $B$  vezes os passos 3 e 4;
6. Obter o quantil  $\varphi_{1-\alpha}$  da distribuição empírica das  $B$  realizações da estatística  $J^*$  obtidas nos passos 3 a 5;
7. Rejeitar  $H_1$  se  $|J| > \varphi_{1-\alpha}$ .

A regra de decisão também pode ser expressa em função do  $p$ -valor aproximado por *bootstrap*, dado por

$$p^* = \frac{\#\{|J_b^*| > |J|\}}{B}, \quad b = \{1, \dots, B\},$$

### 3.6 ESQUEMA BOOTSTRAP NOS TESTE $J$ E $MJ$

em que  $\#$  denota a cardinalidade de um conjunto e  $J_b^*$  representa a estatística  $J^*$  obtida da  $b$ -ésima amostra *bootstrap*. A hipótese  $H_1$  é então rejeitada se  $p^*$  for menor que o tamanho nominal selecionado. Para testar a hipótese  $H_2$  e para  $M > 2$  modelos, o procedimento é análogo.

O esquema *bootstrap* paramétrico para o teste  $MJ$  adaptado consiste nos seguintes passos:

1. Calcular a estatística  $MJ$  de acordo com a Seção 3.5;
2. Gerar uma amostra *bootstrap*  $y^*$ , sob  $H_0$ , com  $y_t^* \stackrel{ind}{\sim} \mathfrak{B}(\hat{\mu}_{mt}, \hat{\phi}_{mt})$ , em que  $\hat{\mu}_{mt} = g^{-1}(x_{mt} \hat{\beta}_m)$  e  $\hat{\phi}_{mt} = h^{-1}(z_{mt} \hat{\gamma}_m)$ .  $\hat{\beta}_m$  e  $\hat{\gamma}_m$  são as estimativas de máxima verossimilhança para os parâmetros do modelo  $m$  correspondente a  $MJ = \min\{J_1^2, \dots, J_M^2\}$ ;
3. Calcular a estatística  $MJ$  *bootstrap*,  $MJ^*$ ;
4. Executar  $B$  vezes os passos 2 e 3;
5. Obter o quantil  $\varphi_{1-\alpha}$  da distribuição empírica das  $B$  realizações da estatística  $MJ^*$  obtida nos passos 2 a 4;
6. Rejeitar a hipótese nula de presença do modelo correto se  $MJ > \varphi_{1-\alpha}$ , em que  $\varphi_{1-\alpha}$  é o quantil  $1 - \alpha$  da distribuição empírica.

Para um número grande de réplicas, o método *bootstrap* aproxima os quantis da distribuição empírica com boa precisão. Entretanto, dada a diversidade de procedimentos *bootstrap* que podem ser realizados, é possível que um esquema apresente bom desempenho em um teste e o mesmo não ocorra em outras situações.

# Avaliações Numéricas

## 4.1 Introdução

Os desempenhos em amostras finitas dos testes  $J$  e  $MJ$ , abordados no Capítulo 3, foram avaliados por meio de simulações de Monte Carlo. Por simplicidade, foram considerados  $M = 2$  modelos concorrentes e avaliados cenários em que os modelos apresentavam divergência nos regressores e nas funções de ligação.

Os valores das variáveis explicativas foram selecionados aleatoriamente da distribuição uniforme padrão,  $\mathcal{U}(0, 1)$ , e mantidos fixos. Os parâmetros foram estimados a partir da maximização numérica do logaritmo da função de verossimilhança aplicando o método de otimização não-linear BFGS com primeiras derivadas analíticas.

Foram consideradas amostras de tamanho  $n = 20, 30$  e  $40$ , sendo geradas 10 observações das variáveis explicativas e replicadas o número de vezes necessário para se obter os tamanhos amostrais desejados, isto é, as observações foram replicadas duas, três e quatro vezes. Este procedimento foi realizado a fim de se manter constante o grau de heterogeneidade da amostra (medido por  $\lambda = \max \phi_t / \min \phi_t$ ) à medida em que aumenta o número de observações. Adicionalmente, os tamanhos amostrais definidos foram utilizados para avaliar os desempenhos dos testes em amostras pequenas, uma vez que as aproximações assintóticas são tipicamente boas em grandes tamanhos amostrais.

Nas simulações foram realizadas 10000 réplicas de Monte Carlo e 1000 réplicas de *bootstrap*, de modo que cada experimento numérico envolveu dez milhões de réplicas. Em cada réplica de Monte Carlo foi gerada uma amostra aleatória da variável resposta  $y = (y_1, \dots, y_n)^\top$ , com  $y_t \sim \mathcal{B}(\mu_t, \phi_t)$ . Para cada réplica de *bootstrap*, por sua vez, foi gerada uma amostra aleatória da variável resposta  $y^* = (y_1^*, \dots, y_n^*)^\top$  como descrito na Seção 3.6. O gerador de números aleatórios uniformes utilizado foi o *multiply-with-carry*, desenvolvido por George Marsaglia (Marsaglia 1997), de período  $2^{60}$ . Os testes  $J$  e  $MJ$  foram realizados a partir das correspondentes estatísticas de teste da razão de verossimilhanças e foram considerados três níveis nominais:  $\alpha = 1\%, 5\%$  e  $10\%$ .

A seguir são apresentados e discutidos os resultados das simulações de tamanho dos testes nos diferentes cenários abordados. Para o teste  $MJ$  é apresentada ainda a proporção na qual a estatística de teste corresponde ao modelo correto quando a hipótese nula (o modelo corretamente especificado está entre os modelos testados) não foi rejeitada. Na Seção 4.2 são apresentados os resultados para modelos com diferença nos regressores e na Seção 4.3 são discutidos os resultados obtidos para modelos que diferem nas funções de ligação. Os programas na linguagem Ox (Doornik 2009) utilizados nas simulações são apresentados no Apêndice.

Nos casos em que os modelos divergiam unicamente no submodelo da média ou apenas

no submodelo da precisão, as estatísticas de teste foram comparadas com valores críticos da distribuição qui-quadrado com um grau de liberdade. Isto ocorre pois é testada a exclusão apenas do preditor linear do submodelo em que há divergência do modelo concorrente. Já nos cenários em que os modelos apresentavam diferença no submodelo da média e da precisão, as estatísticas de teste foram comparadas com valores críticos da distribuição qui-quadrado com dois graus de liberdade. Isto se deu em decorrência da necessidade de ser testada a exclusão de uma variável do submodelo da média e outra do submodelo da precisão.

## 4.2 Especificação incorreta de regressores

Para obter as taxas de rejeição nula dos testes  $J$  e  $MJ$  nos modelos com diferença nos regressores, foram consideradas a função de ligação logito e a função logarítmica para os submodelos da média e da precisão, respectivamente. Nos três cenários avaliados a seguir – em que há modelos com erro de especificação nos regressores da média, da precisão e de ambos simultaneamente – foram considerados, além do intercepto, dois regressores no submodelo da média, e um regressor no submodelo da precisão. A distinção entre modelos concorrentes ocorre em um dos regressores, como pode ser observado a seguir.

### 4.2.1 Regressores da média

No caso em que os modelos concorrentes apresentam divergência nos regressores do submodelo da média, a especificação dos mesmos foi definida da forma:

$$H_1 : \begin{cases} \log\left(\frac{\mu_t}{1-\mu_t}\right) = \beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} \\ \log(\phi_t) = \gamma_0 + \gamma_1 x_{t1} \end{cases} \quad \text{e} \quad H_2 : \begin{cases} \log\left(\frac{\mu_t}{1-\mu_t}\right) = \beta_0 + \beta_1 x_{t1} + \beta_3 x_{t3} \\ \log(\phi_t) = \gamma_0 + \gamma_1 x_{t1}, \end{cases}$$

com  $t = 1, \dots, n$ . Como se pode observar, nos dois modelos a especificação do parâmetro de precisão é a mesma. O modelo  $H_1$  tem como regressores do submodelo da média as variáveis  $x_1$  e  $x_2$ , enquanto  $H_2$  considera as covariáveis  $x_1$  e  $x_3$ . Considerando-se o modelo  $H_1$  correto, isto é, o modelo correspondente ao processo gerador dos dados, os valores dos parâmetros são  $\beta_0 = 1.0$ ,  $\beta_1 = 2.2$ ,  $\beta_3 = -3.7$ ,  $\gamma_0 = 1.5$  e  $\gamma_1 = 2.3$ . De acordo com os valores verdadeiros dos parâmetros,  $\mu$  assume valores no intervalo  $(0.08, 0.87)$ . O grau de heterogeneidade da precisão,  $\lambda = \max(\phi_t)/\min(\phi_t)$ , é aproximadamente 7.84 para todos os tamanhos amostrais, uma vez que foi replicada uma amostra contendo 10 observações. A avaliação dos modelos se deu a partir do teste da exclusão do preditor linear do submodelo da média do modelo concorrente.

De acordo com a Tabela 4.1, pode-se observar que tamanhos amostrais maiores estão associados a taxas de rejeição nula mais próximas dos níveis nominais. Adicionalmente, as versões *bootstrap* apresentaram melhor desempenho que os respectivos testes assintóticos, que tenderam a sobre-rejeitar a hipótese nula em todos os tamanhos de amostra considerados. As distorções nas taxas de rejeição nula dos testes são ainda maiores nos testes assintóticos em amostras de tamanho 20, cujas taxas de rejeição nula foram cerca de três vezes maior que o desejado para  $\alpha = 1\%$  e mais que o dobro quando  $\alpha = 5\%$ . Vale salientar ainda que as taxas de rejeição nula dos testes  $J$  e  $MJ$  coincidiram em todos os tamanhos amostrais considerados.

## 4.2 ESPECIFICAÇÃO INCORRETA DE REGRESSORES

Isto ocorre devido à coincidência entre as estatísticas de ambos os testes, ou seja, em todas as réplicas de Monte Carlo a menor das estatísticas ( $MJ = \min\{J_1^2, J_2^2\}$ ) correspondeu à estatística associada ao modelo  $H_1$ .

**Tabela 4.1** Taxas de rejeição nula (%) no modelo com erro no regressor da média.

$n$	$\alpha = 1\%$			$\alpha = 5\%$			$\alpha = 10\%$		
	20	30	40	20	30	40	20	30	40
$J$	3.09	1.89	1.74	10.07	7.70	7.20	16.41	13.57	12.86
$J_{boot}$	1.02	1.07	1.18	5.08	5.02	5.36	10.11	9.91	10.20
$MJ$	3.09	1.89	1.74	10.07	7.70	7.20	16.41	13.57	12.86
$MJ_{boot}$	1.02	1.07	1.18	5.08	5.02	5.36	10.11	9.91	10.12

### 4.2.2 Regressores da precisão

As taxas de rejeição nula dos testes para modelos não encaixados que diferem nos regressores da precisão foram obtidas considerando-se os seguintes modelos:

$$H_1 : \log\left(\frac{\mu_t}{1-\mu_t}\right) = \beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} \quad \text{e} \quad H_2 : \log\left(\frac{\mu_t}{1-\mu_t}\right) = \beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2}$$

$$\log(\phi_t) = \gamma_0 + \gamma_1 x_{t1} \quad \log(\phi_t) = \gamma_0 + \gamma_2 x_{t2},$$

com  $t = 1, \dots, n$ . Em  $H_1$  o submodelo da precisão contém o regressor  $x_1$  e em  $H_2$ , o regressor  $x_2$ . A especificação do submodelo da média é a mesma em  $H_2$  e  $H_1$ , que foi considerado o modelo correto. Os valores dos parâmetros do modelo foram os mesmos do cenário anterior, de forma que o intervalo de variação de  $\mu$  e o grau de heterogeneidade da precisão se mantiveram os mesmos.

Ambos os testes tenderam a sobre-rejeitar a hipótese nula (Tabela 4.2), no entanto o teste  $MJ$  apresentou uma flutuação mais próxima aos valores de referência relativamente ao teste  $J$ . Por outro lado, as taxas de rejeição nula do teste  $J$  tenderam a se aproximar dos valores desejados à medida que o tamanho da amostra foi aumentado, sendo que o aumento do número de observações de 20 para 30 conduziu a uma maior diferença nas taxas de rejeição nula do teste  $J$  do que entre amostras de tamanho 30 e 40. De modo geral, a utilização do esquema *bootstrap* conduziu a melhorias nas taxas de rejeição nula dos testes.

### 4.2.3 Regressores da média e da precisão

No cenário em que os modelos concorrentes apresentam divergência nos regressores da média e da precisão conjuntamente, as taxas de rejeição nula foram obtidas considerando-se os modelos

$$H_1 : \log\left(\frac{\mu_t}{1-\mu_t}\right) = \beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} \quad \text{e} \quad H_2 : \log\left(\frac{\mu_t}{1-\mu_t}\right) = \beta_0 + \beta_1 x_{t1} + \beta_3 x_{t3}$$

$$\log(\phi_t) = \gamma_0 + \gamma_2 x_{t2} \quad \log(\phi_t) = \gamma_0 + \gamma_3 x_{t3},$$

com  $t = 1, \dots, n$ . No modelo  $H_1$  a covariada  $x_2$  é utilizada no submodelo da média e da precisão, enquanto em  $H_2$  é utilizado o regressor  $x_3$ . Uma vez que o modelo  $H_1$  foi utilizado no processo

## 4.2 ESPECIFICAÇÃO INCORRETA DE REGRESSORES

**Tabela 4.2** Taxas de rejeição nula (%) nos modelos com erro no regressor da precisão.

$n$	$\alpha = 1\%$			$\alpha = 5\%$			$\alpha = 10\%$		
	20	30	40	20	30	40	20	30	40
$J$	3.81	2.27	2.03	11.43	8.03	7.88	18.65	14.13	13.80
$J_{boot}$	1.26	1.06	1.21	5.95	5.43	5.93	11.36	10.34	11.14
$MJ$	1.61	1.29	1.41	6.60	5.86	6.27	12.22	11.93	12.09
$MJ_{boot}$	0.85	0.88	1.24	5.07	5.69	6.26	10.48	11.29	11.16

gerador dos dados, foram utilizados os mesmos valores dos parâmetros empregados no caso de modelos com divergência nos regressores da média. O intervalo de variação de  $\mu$  e o grau de heterogeneidade da precisão também se mantiveram os mesmos.

Nos cenários anteriores os modelos foram avaliados com base na exclusão de um regressor artificial no submodelo da média ou da precisão. Aqui, como os modelos apresentavam distinção nos dois submodelos, as taxas de rejeição nula foram computadas considerando-se testes da exclusão dos preditores lineares da média e da precisão conjuntamente. Desta forma, como era testada a exclusão de dois regressores (um do submodelo da média e outro do submodelo da precisão), as estatísticas de teste foram comparadas com valores críticos obtidos da distribuição qui-quadrado com dois graus de liberdade.

Como se pode observar na Tabela 4.3, as taxas de rejeição nula dos testes assintóticos foram mais altas que as obtidas por um esquema *bootstrap*, que estiveram mais próximas dos valores desejados. As taxas de rejeição dos testes  $J$  e  $MJ$  coincidiram em todos os tamanhos de amostra, visto que o mínimo entre as estatísticas de teste dos modelos avaliados correspondeu ao modelo  $H_1$  em todas as réplicas de Monte Carlo. Para  $\alpha = 1\%$ , as taxas de rejeição da hipótese nula dos testes assintóticos tenderam a se aproximar dos valores de referência quando o tamanho da amostra aumentou. Por outro lado, para  $\alpha = 5\%$  e  $\alpha = 10\%$  o mesmo não foi observado.

**Tabela 4.3** Taxas de rejeição nula (%) no modelo com erro no regressor da média e da precisão.

$n$	$\alpha = 1\%$			$\alpha = 5\%$			$\alpha = 10\%$		
	20	30	40	20	30	40	20	30	40
$J$	5.09	2.60	2.14	9.83	14.38	8.69	17.03	22.89	15.58
$J_{boot}$	0.96	1.01	0.97	4.91	4.78	5.16	9.94	9.53	10.60
$MJ$	5.09	2.60	2.14	9.83	14.38	8.69	17.03	22.89	15.58
$MJ_{boot}$	0.96	1.01	0.97	4.91	4.78	5.16	9.94	9.53	10.60

### 4.3 Especificação incorreta da função de ligação

Na avaliação dos testes para modelos com diferença nas funções de ligação foram considerados modelos com dois regressores no submodelo da média e um regressor no submodelo da precisão, não havendo diferença entre os regressores dos modelos concorrentes.

#### 4.3.1 Função de ligação da média

Para o cenário com modelos com diferença na função de ligação do submodelo da média, foi considerado um modelo com função de ligação log-log e outro como função de ligação logito no submodelo da média:

$$H_1 : \begin{aligned} -\log\{-\log(\mu_t)\} &= \beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} \\ \log(\phi_t) &= \gamma_0 + \gamma_1 x_{t1} \end{aligned} \quad \text{e} \quad H_2 : \begin{aligned} \log\left(\frac{\mu_t}{1-\mu_t}\right) &= \beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} \\ \log(\phi_t) &= \gamma_0 + \gamma_1 x_{t1}, \end{aligned}$$

com  $t = 1, \dots, n$ . Sendo  $H_1$  considerado o modelo correto, os valores dos parâmetros são  $\beta_0 = 1.0$ ,  $\beta_1 = 1.6$ ,  $\beta_3 = -2.1$ ,  $\gamma_0 = 1.2$  e  $\gamma_1 = 1.7$ . Com base nesses valores, o intervalo de variação de  $\mu$  é  $(0.25, 0.90)$  e o grau de heterogeneidade é aproximadamente 4.58 para os tamanhos amostrais considerados.

Como se pode observar na Tabela 4.4, as taxas de rejeição nula dos testes assintóticos tendem aos valores nominais de quando o tamanho da amostra aumenta. As taxas de rejeição do teste  $MJ$  estão mais próximas dos valores desejados do que as do teste  $J$ , sendo que em alguns casos os testes subrejeitaram a hipótese nula. As versões *bootstrap* apresentaram melhor desempenho do que as correspondentes versões assintóticas.

**Tabela 4.4** Taxas de rejeição nula (%) no modelo com erro na função de ligação da média.

$n$	$\alpha = 1\%$			$\alpha = 5\%$			$\alpha = 10\%$		
	20	30	40	20	30	40	20	30	40
$J$	2.91	2.08	1.64	9.85	7.79	6.75	16.62	14.00	12.23
$J_{boot}$	0.89	1.09	1.07	4.92	5.40	4.94	10.10	10.38	9.84
$MJ$	1.64	1.27	0.95	6.46	4.83	4.15	11.36	8.93	7.60
$MJ_{boot}$	0.91	1.09	1.22	5.49	5.27	5.19	10.55	10.49	9.87

#### 4.3.2 Função de ligação da precisão

Neste cenário foram considerados modelos com função de ligação logito no submodelo da média. Um dos modelos utiliza a função de ligação raiz quadrada no submodelo da precisão e o outro, a função logarítmica. Os modelos foram especificados da seguinte forma:

$$H_1 : \begin{aligned} \log\left(\frac{\mu_t}{1-\mu_t}\right) &= \beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} \\ \sqrt{\phi_t} &= \gamma_0 + \gamma_1 x_{t1} \end{aligned} \quad \text{e} \quad H_2 : \begin{aligned} \log\left(\frac{\mu_t}{1-\mu_t}\right) &= \beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} \\ \log(\phi_t) &= \gamma_0 + \gamma_1 x_{t1}, \end{aligned}$$

#### 4.3 ESPECIFICAÇÃO INCORRETA DA FUNÇÃO DE LIGAÇÃO

com  $t = 1, \dots, n$ . Aqui,  $H_1$  foi o modelo correto e os valores dos parâmetros foram iguais aos do caso em que os modelos divergiam nos regressores da média. De forma semelhante, o intervalo de variação de  $\mu$  e o grau de heterogeneidade da precisão foram os mesmos.

As taxas de rejeição da hipótese nula dos testes assintóticos (Tabela 4.5) tendem aos valores desejados quando o tamanho da amostra aumenta. Novamente, as taxas de rejeição nula do teste  $MJ$  estão mais próximas dos níveis nominais do que as do teste  $J$ . Vale salientar ainda que para  $n = 40$  as taxas do teste  $MJ$  assintótico estão próximas dos valores desejados, de forma que a correção nas distorções por meio de um esquema *bootstrap* foi menor do que o verificado com tamanhos de amostra menores.

**Tabela 4.5** Taxas de rejeição nula (%) no modelo com erro na função de ligação da precisão.

$n$	$\alpha = 1\%$			$\alpha = 5\%$			$\alpha = 10\%$		
	20	30	40	20	30	40	20	30	40
$J$	3.53	2.39	1.82	10.84	8.42	6.88	18.10	14.51	12.73
$J_{boot}$	0.91	1.13	1.08	5.20	5.32	4.66	10.10	10.03	9.42
$MJ$	2.44	1.64	1.27	8.65	6.29	4.71	14.61	11.23	9.23
$MJ_{boot}$	0.95	1.13	1.08	5.49	5.36	4.47	10.65	10.13	9.30

#### 4.3.3 Função de ligação da média e da precisão

As taxas de erro dos testes para os modelos com diferença nas funções de ligação da média e da precisão simultaneamente foram obtidas considerando o modelo com função de ligação log-log no submodelo da média e função de ligação raiz quadrada no submodelo da precisão. O modelo concorrente possui funções de ligação logito e logarítmica. Os modelos foram especificados da seguinte forma:

$$H_1 : \begin{aligned} -\log\{-\log(\mu_t)\} &= \beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} \\ \sqrt{\phi_t} &= \gamma_0 + \gamma_1 x_{t1} \end{aligned} \quad \text{e} \quad H_2 : \begin{aligned} \log\left(\frac{\mu_t}{1-\mu_t}\right) &= \beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} \\ \log(\phi_t) &= \gamma_0 + \gamma_1 x_{t1}, \end{aligned}$$

com  $t = 1, \dots, n$ . O modelo  $H_1$  é o correto e os valores dos parâmetros são os mesmos do cenário em que os modelos divergem na função de ligação da média.

De acordo com os resultados reportados na Tabela 4.6, os testes assintóticos tendem a apresentar taxas de rejeição nula mais próximas dos valores desejados à medida que o tamanho da amostra aumenta. Para  $n = 20$  e  $\alpha = 1\%$ , a taxa de rejeição nula do teste  $J$  é aproximadamente cinco vezes maior do que o valor de referência. No geral, o teste  $MJ$  apresenta taxas de rejeição da hipótese nula mais próximas dos valores nominais e a utilização de reamostragem *bootstrap* reduz distorções de tamanho.

#### 4.3.4 Seleção do modelo correto pelo teste $MJ$

A estatística  $MJ$  é o mínimo entre as estatísticas  $J$ . É importante ressaltar que tal estatística pode ser utilizada para selecionar o modelo correto. Desta forma, foi calculada a proporção com que

### 4.3 ESPECIFICAÇÃO INCORRETA DA FUNÇÃO DE LIGAÇÃO

**Tabela 4.6** Taxas de rejeição nula (%) no modelo com erro na função de ligação da média e da precisão.

$n$	$\alpha = 1\%$			$\alpha = 5\%$			$\alpha = 10\%$		
	20	30	40	20	30	40	20	30	40
$J$	4.85	2.41	1.80	14.08	8.63	7.60	22.03	15.53	14.70
$J_{boot}$	1.31	1.14	1.70	5.66	4.52	4.30	10.68	9.49	9.60
$MJ$	3.12	1.55	1.30	9.87	5.79	5.20	16.28	10.84	9.90
$MJ_{boot}$	1.39	1.14	1.30	5.53	4.73	5.00	11.00	9.70	10.70

a seleção de modelos via  $MJ$  foi correta. Os resultados estão apresentados na Tabela 4.7, que apresenta os percentuais de vezes em o modelo correto foi selecionado quando a hipótese nula (presença do modelo corretamente especificado dentre os modelos em teste) não foi rejeitada.

Nos casos em que foram avaliados os modelos com diferença nos regressores da média, o mecanismo de seleção de modelos baseado na estatística do teste  $MJ$  forneceu o modelo correto em todas as réplicas de Monte Carlo. O mesmo resultado foi observado nos casos em que os modelos concorrentes apresentavam diferença nos regressores da média e da precisão conjuntamente. Quando os modelos diferiam nos regressores da precisão, o percentual de seleção do modelo correto cresceu progressivamente à medida que o tamanho da amostra foi aumentando, estando por volta de 79.3% para  $n = 20$  e em torno de 94.4% quando  $n = 40$ . Como afirma Bayer (2011, p. 39), “a seleção do submodelo da média [no tocante à escolha de regressores] depende fracamente da modelagem correta da precisão”, enquanto “a seleção correta do submodelo da precisão é a parte mais crítica na seleção de modelo de regressão beta com dispersão variável.” Isto é evidenciado pela exatidão da seleção de modelos via estatística  $MJ$  quando a hipótese nula não foi rejeitada nos casos em que os modelo apresentavam diferenças, quer seja nos regressores da média ou nos regressores do submodelo da média e da precisão conjuntamente.

No tocante aos modelos com funções de ligação distintas, os percentuais de seleção dos modelos corretos estiveram abaixo daqueles verificados para modelos com diferença nos regressores. Para a ligação da média, o percentual em que o modelo correto foi selecionado esteve em torno de 57.3% para  $n = 20$  e cerca de 60.3% quando  $n = 40$ , o que evidencia um crescimento suave no percentual de seleção do modelo corretamente especificado à medida que o tamanho da amostra aumenta. Quando a diferença entre os modelos residia na função de ligação do submodelo da precisão, as taxas de seleção do modelo correto apresentaram uma flutuação por volta de 51.9% para  $n = 20$  e 50.3% para  $n = 40$ , não apresentado crescimento claro, como no caso anteriormente avaliado. Para modelos com diferença na função de ligação da precisão, tais taxas estiveram levemente acima de 50%, apresentando apenas uma flutuação próxima desse percentual. Vale salientar ainda que, de modo geral, os percentuais de seleção do modelo correto nos testes *bootstrap* foram levemente maiores que nos testes assintóticos. Nesses casos pode-se utilizar conjuntamente os testes  $J$  e  $MJ$  da seguinte forma: utilizar o teste  $MJ$  para verificar a presença do modelo correto entre os modelos considerados e, caso essa hipótese não seja rejeitada, utilizar o teste  $J$  para identificar o modelo correto.

### 4.3 ESPECIFICAÇÃO INCORRETA DA FUNÇÃO DE LIGAÇÃO

**Tabela 4.7** Percentuais em que o modelo correto foi selecionado via estatística  $MJ$  ( $\hat{m} = m^*$ ) quando a hipótese nula não foi rejeitada.

Erro	$\alpha$	$n = 20$		$n = 30$		$n = 40$	
		assint.	<i>boot.</i>	assint.	<i>boot.</i>	assint.	<i>boot.</i>
Regressores da média	1%	100.00	100.00	100.00	100.00	100.00	100.00
	5%	100.00	100.00	100.00	100.00	100.00	100.00
	10%	100.00	100.00	100.00	100.00	100.00	100.00
Regressores da precisão	1%	78.85	78.80	88.59	88.61	93.83	93.88
	5%	79.10	79.52	88.94	89.37	94.09	94.47
	10%	79.46	80.30	89.22	90.06	94.57	95.29
Regressores da média e da precisão	1%	100.00	100.00	100.00	100.00	100.00	100.00
	5%	100.00	100.00	100.00	100.00	100.00	100.00
	10%	100.00	100.00	100.00	100.00	100.00	100.00
Função de ligação da média	1%	57.66	57.71	59.34	59.37	60.94	60.88
	5%	57.29	57.42	58.72	58.67	60.28	60.09
	10%	56.74	56.84	58.23	58.02	59.84	59.58
Função de ligação da precisão	1%	52.12	52.26	50.11	50.18	50.40	50.36
	5%	51.72	52.11	51.08	50.02	50.40	50.41
	10%	51.28	51.79	50.12	50.09	50.23	50.18
Função de ligação da média e da precisão	1%	53.82	53.86	55.57	55.59	56.13	56.13
	5%	53.28	53.70	55.24	55.36	56.01	56.00
	10%	53.02	53.38	55.06	55.18	55.61	55.43

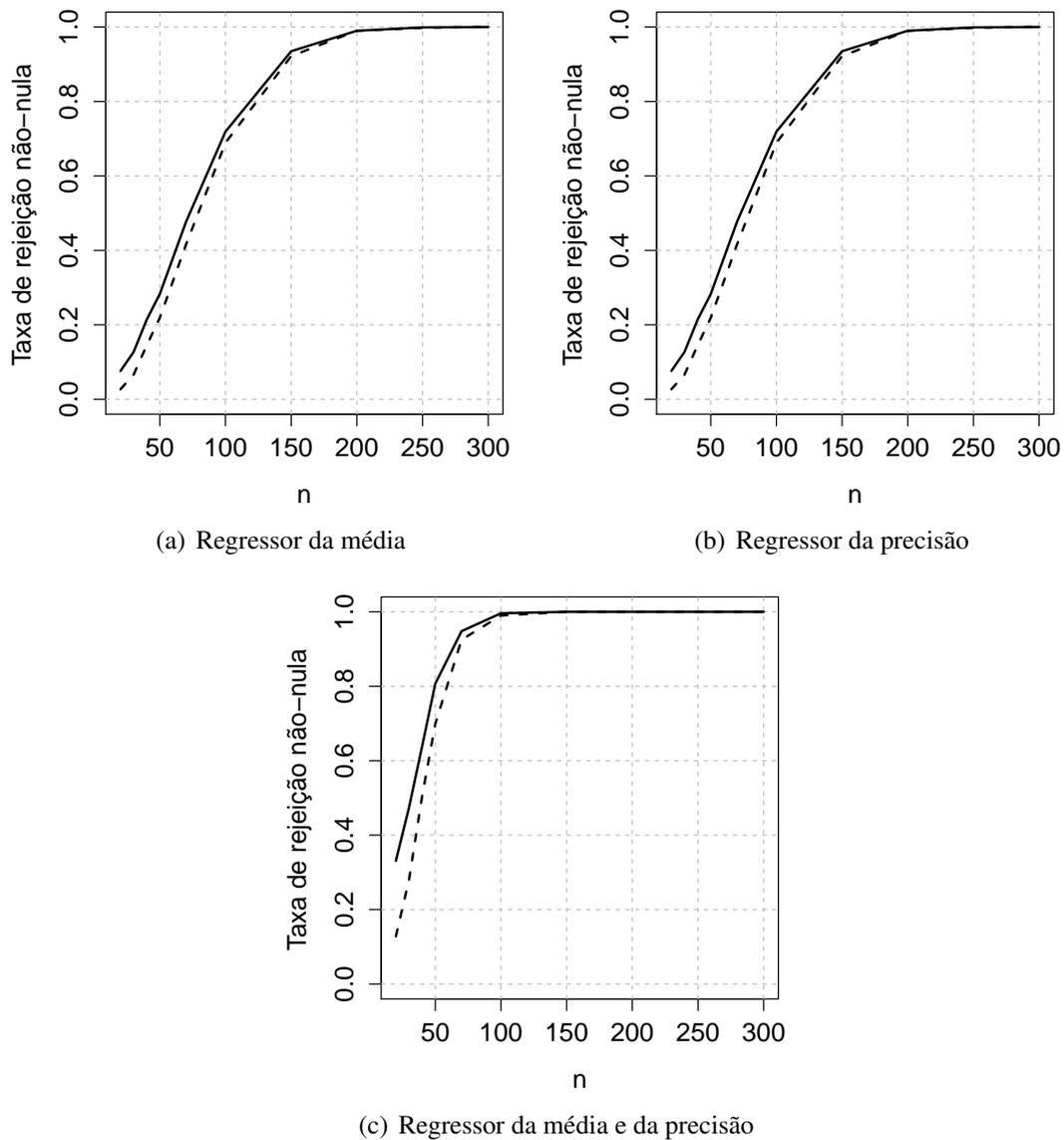
#### 4.3.5 Poderes dos testes

Os desempenhos dos testes  $J$  e  $MJ$  ao rejeitar a hipótese nula na ausência do modelo corretamente especificado também foram avaliados. Nesta etapa foram estimados os modelos descritos anteriormente, sendo que no processo de geração dos dados foi utilizado um modelo distinto. No caso de modelos com diferença nos regressores, o modelo correto era semelhante aos modelos  $H_1$  e  $H_2$  apresentados nas Seções 4.2.1, 4.2.2 e 4.2.3, com exceção de um dos regressores que não pertencia a qualquer um dos modelos estimados. No caso de modelos com diferença na função de ligação, foi utilizada a função de ligação log-log complementar no submodelo da média, já no caso de funções de ligação distintas para a precisão, foi utilizada a ligação identidade. Quando os modelos divergiam na função de ligação da média e da precisão, foram utilizadas as funções log-log complementar e identidade no processo de geração dos dados.

Nas Figuras 4.1, 4.2, 4.3 e 4.4 são apresentadas respectivamente as taxas de rejeição da hipótese nula dos testes  $J$  e  $MJ$  quando o modelo correto não estava presente para  $\alpha = 5\%$ . Claramente as taxas de rejeição não-nula crescem consoante o tamanho da amostra aumenta. Para o teste  $J$  as taxas de rejeição não-nula sinalizam um poder levemente menor para os testes *bootstrap*. Por outro lado, de modo geral o uso de *bootstrap* elevou as taxas de rejeição da hipótese nula no teste  $MJ$ , principalmente nos modelos com diferença na função de ligação da

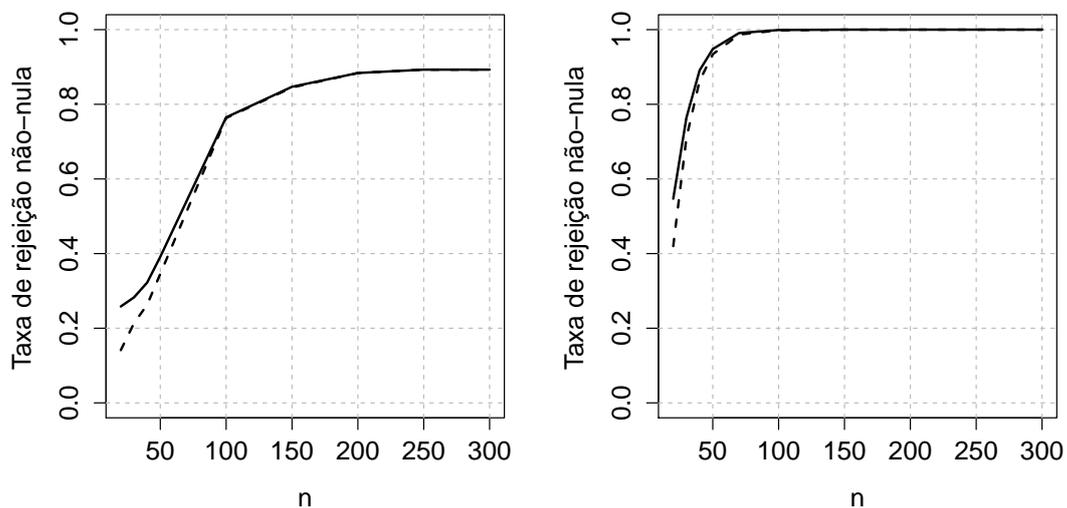
### 4.3 ESPECIFICAÇÃO INCORRETA DA FUNÇÃO DE LIGAÇÃO

média. As taxas de rejeição de ambos os testes estão próximas para os modelos com diferença nos regressores da média e nos regressores da precisão. No caso de modelos com diferença na função de ligação da média o teste  $J$  tem maior poder que o teste  $MJ$  em amostras pequenas, ocorrendo o inverso nos tamanhos de amostra maiores. Nos demais casos as taxas de rejeição não-nula do teste  $J$  são maiores.



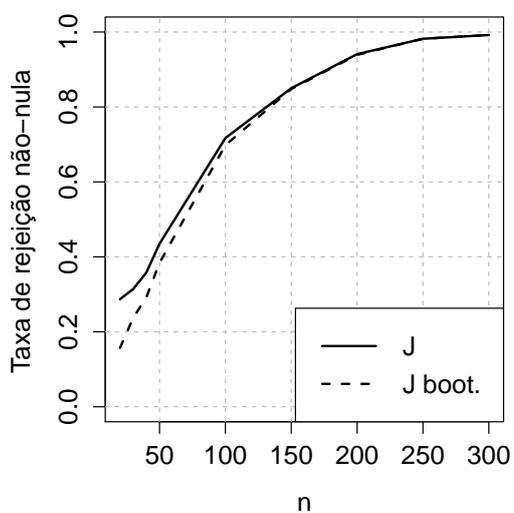
**Figura 4.1** Taxas de rejeição não-nula do teste  $J$  e do teste  $J$  bootstrap ao nível  $\alpha = 5\%$  para modelos com diferença nos regressores.

### 4.3 ESPECIFICAÇÃO INCORRETA DA FUNÇÃO DE LIGAÇÃO



(a) Ligação da média

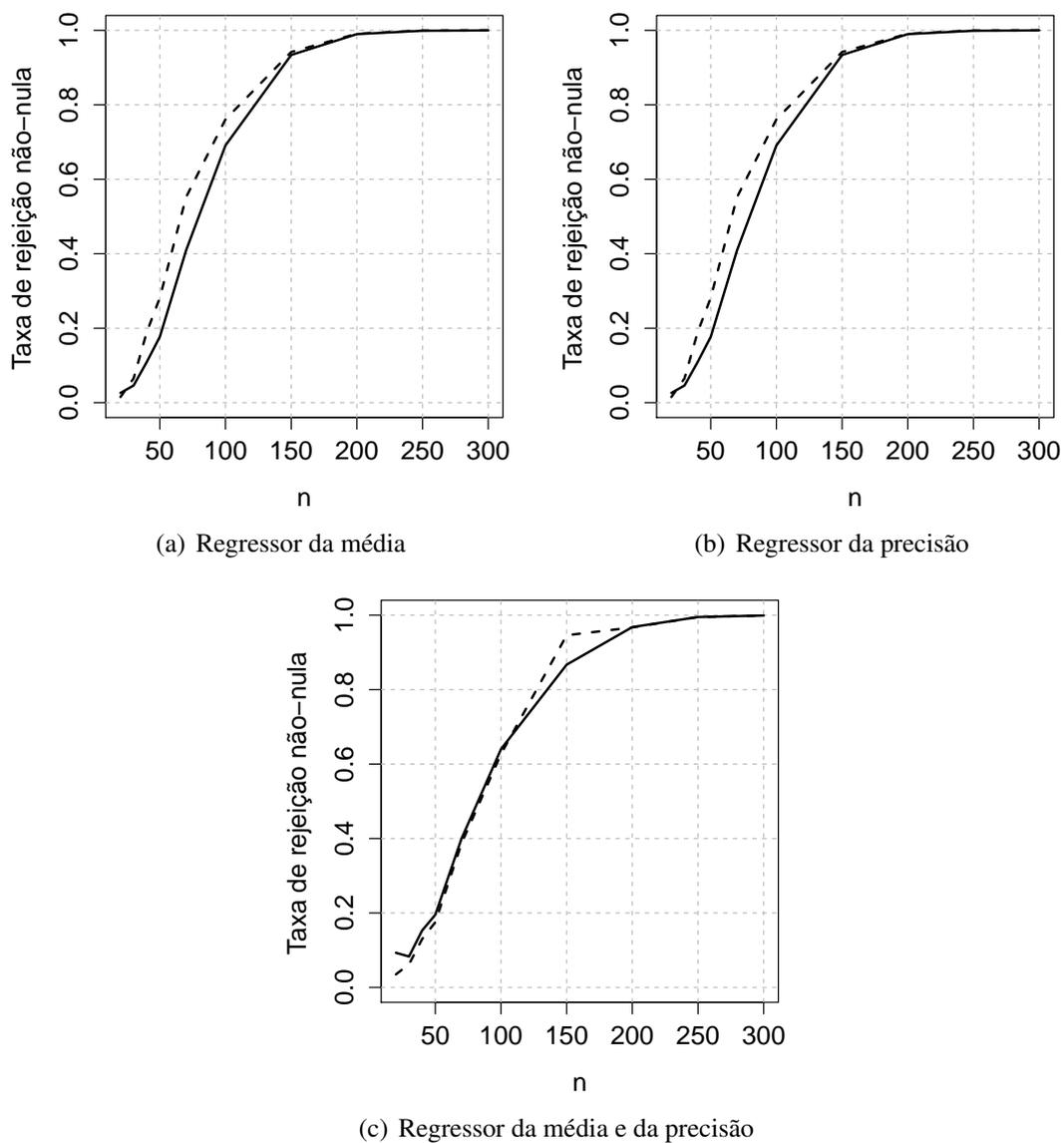
(b) Ligação da precisão



(c) Ligação da média e da precisão

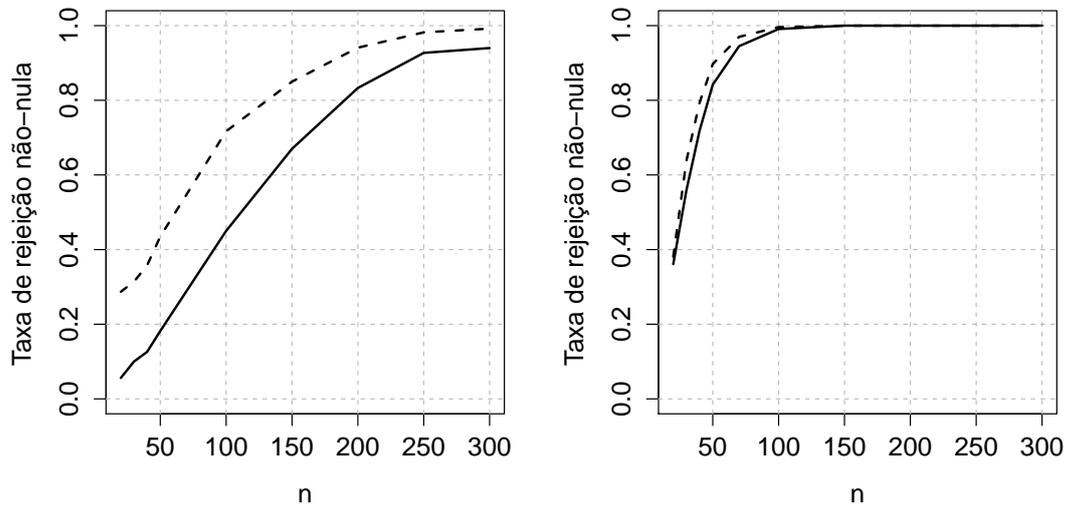
**Figura 4.2** Taxas de rejeição não-nula do teste  $J$  e do teste  $J bootstrap$  ao nível  $\alpha = 5\%$  para modelos com diferença nas funções de ligação.

### 4.3 ESPECIFICAÇÃO INCORRETA DA FUNÇÃO DE LIGAÇÃO



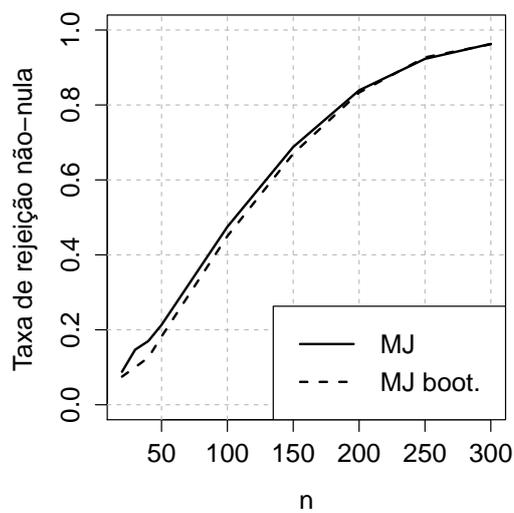
**Figura 4.3** Taxas de rejeição não-nula do teste *MJ* e do teste *MJ bootstrap* ao nível  $\alpha = 5\%$  para modelos com diferença nos regressores.

### 4.3 ESPECIFICAÇÃO INCORRETA DA FUNÇÃO DE LIGAÇÃO



(a) Ligação da média

(b) Ligação da precisão



(c) Ligação da média e da precisão

**Figura 4.4** Taxas de rejeição não-nula do teste *MJ* e do teste *MJ bootstrap* ao nível  $\alpha = 5\%$  para modelos com diferença nas funções de ligação.

## CAPÍTULO 5

# Aplicações

Para ilustrar o emprego dos métodos abordados no Capítulo 3, são apresentadas a seguir duas aplicações com dados reais. No primeiro caso são avaliados modelos de regressão beta que diferem na função de ligação da média e no segundo são analisados modelos com diferença nos regressores do submodelo da precisão.

### 5.1 Dados de Produção de Gasolina

Nesta aplicação foi utilizado o conjunto de dados apresentado por Prater (1956), cuja variável de interesse é a proporção de petróleo convertido em gasolina após destilação e fracionamento. Duas variáveis explicativas foram utilizadas: `temp`, a temperatura em graus Fahrenheit na qual toda gasolina foi vaporizada, e `batch`, um fator indicando dez condições experimentais de destilação. O banco de dados consiste de 32 observações. Na especificação do modelo foram consideradas nove variáveis *dummy* correspondentes aos nove primeiros níveis da variável `batch` ( $x_1, \dots, x_9$ ) e a variável `temp` ( $x_{10}$ ).

Cribari-Neto & Zeileis (2010) utilizaram modelos de regressão beta na modelagem dos dados e verificaram que, ao empregar a função de ligação logito no submodelo da média, a inclusão da variável `temp` no submodelo da precisão resultou em melhoras significativas no modelo. Por outro lado, ao utilizar a função de ligação log-log, deveria ser considerado o modelo com dispersão constante.

O primeiro modelo considerado por Cribari-Neto & Zeileis (2010), aqui denominado modelo 1, é dado por

$$\begin{aligned} \log\left(\frac{\mu_t}{1-\mu_t}\right) &= \beta_0 + \sum_{j=1}^{10} \beta_j x_j \\ \log(\phi_t) &= \gamma_0 + \gamma_1 x_{t10}, \end{aligned}$$

com  $t = 1, \dots, 32$ . O modelo 2, isto é, o modelo com ligação log-log, é especificado da seguinte forma:

$$\begin{aligned} -\log\{-\log(\mu_t)\} &= \beta_0 + \sum_{j=1}^{10} \beta_j x_j \\ \phi_t &= \phi, \end{aligned}$$

em que  $\phi$  é uma constante positiva.

Os modelos 1 e 2 são considerados não-encaixados por utilizarem funções de ligação diferentes na modelagem da média. As estimativas dos parâmetros e os respectivos erros-padrão estão apresentados na Tabela 5.1 juntamente com as estatísticas de teste e  $p$ -valores do teste  $z$ . Para os níveis de significância usuais, as variáveis explicativas incluídas nos modelos foram significativas.

## 5.1 DADOS DE PRODUÇÃO DE GASOLINA

**Tabela 5.1** Estimativas dos parâmetros dos modelos 1 e 2 utilizados na modelagem de dados de produção de gasolina.

Modelo	Parâmetro	Estimativa	Erro-padrão	$z$	$p$ -valor
1	$\beta_0$	-5.9232	0.1835	-32.275	$< 2 \times 10^{-16}$
	$\beta_1$	1.6020	0.0639	25.087	$< 2 \times 10^{-16}$
	$\beta_2$	1.2973	0.0991	13.090	$< 2 \times 10^{-16}$
	$\beta_3$	1.5653	0.0997	15.694	$< 2 \times 10^{-16}$
	$\beta_4$	1.0301	0.0633	16.276	$< 2 \times 10^{-16}$
	$\beta_5$	1.1542	0.0656	17.582	$< 2 \times 10^{-16}$
	$\beta_6$	1.0194	0.0664	15.364	$< 2 \times 10^{-16}$
	$\beta_7$	0.6223	0.0656	9.481	$< 2 \times 10^{-16}$
	$\beta_8$	0.5646	0.0602	9.381	$< 2 \times 10^{-16}$
	$\beta_9$	0.3594	0.0671	5.354	$< 8.63 \times 10^{-8}$
	$\beta_{10}$	0.0104	0.0004	23.751	$< 2 \times 10^{-16}$
	$\gamma_0$	1.3641	1.2258	1.113	$2.26 \times 10^{-1}$
	$\gamma_1$	0.0146	0.0036	4.027	$5.65 \times 10^{-5}$
2	$\beta_0$	-2.7938	0.0579	-48.237	$< 2 \times 10^{-16}$
	$\beta_1$	0.9039	0.0347	26.019	$< 2 \times 10^{-16}$
	$\beta_2$	0.6554	0.0377	17.399	$< 2 \times 10^{-16}$
	$\beta_3$	0.7684	0.0378	20.304	$< 2 \times 10^{-16}$
	$\beta_4$	0.5376	0.0335	16.046	$< 2 \times 10^{-16}$
	$\beta_5$	0.5517	0.0347	15.909	$< 2 \times 10^{-16}$
	$\beta_6$	0.5198	0.0351	14.795	$< 2 \times 10^{-16}$
	$\beta_7$	0.2922	0.0341	8.576	$< 2 \times 10^{-16}$
	$\beta_8$	0.2505	0.0350	7.161	$< 2 \times 10^{-16}$
	$\beta_9$	0.1871	0.0383	4.885	$< 2 \times 10^{-16}$
	$\beta_{10}$	0.0054	0.0001	40.008	$< 2 \times 10^{-16}$
	$\phi$	906.7	226.6	4.001	$6.32 \times 10^{-5}$

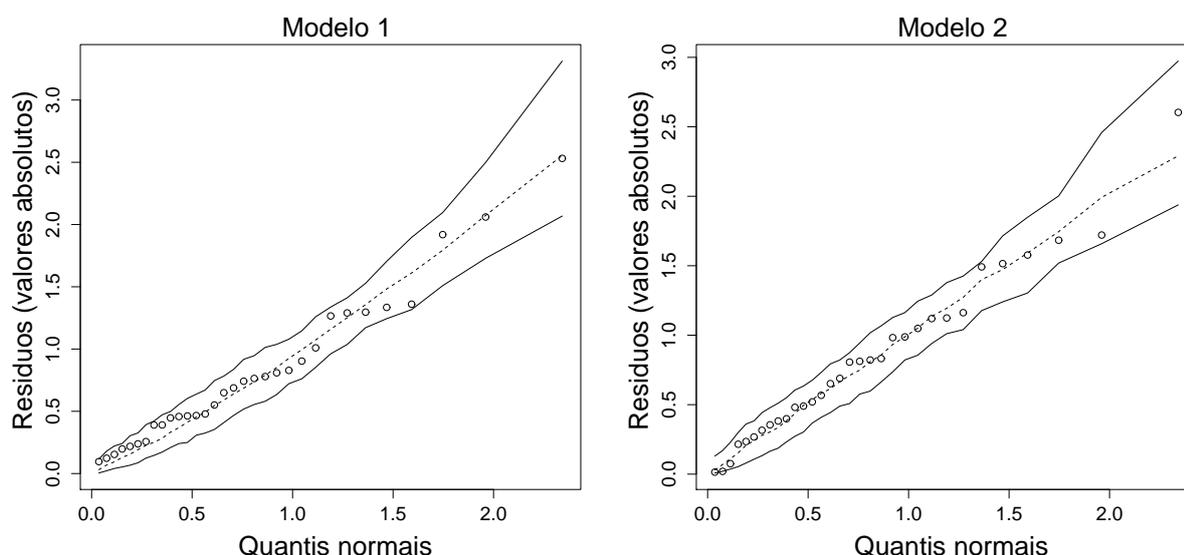
Para avaliar se os modelos apresentavam algum erro de especificação, foi utilizada a abordagem proposta por Cribari-Neto & Lima (2007) que considera uma sequência de testes diagnósticos inspirados no teste RESET (*regression specification error test*), introduzido por Ramsey (1969) para modelos lineares. Nesta abordagem, conclui-se que um modelo está mal especificado se ao acrescentarmos uma potência de seu preditor linear ou de seus valores ajustados como um regressor auxiliar, há melhorias no ajuste. Assim, para o modelo 1 foi adicionado o quadrado do preditor linear  $\hat{\eta}_1$  como regressor auxiliar no submodelo da média e testada a exclusão por meio do teste da razão de verossimilhanças. Foi verificado que a inclusão de  $\hat{\eta}_1^2$  forneceu melhorias no modelo 1 ( $\chi_1^2 = 14.486$ ,  $p$ -valor = 0.0001), o que indica que o modelo apresenta erro de especificação que pode ser decorrente de omissão de covariáveis ou forma funcional incorreta. Para o modelo 2, a inclusão do quadrado do preditor linear  $\hat{\eta}_2$  não apresentou melhorias no ajuste ( $\chi_1^2 = 1.6671$ ,  $p$ -valor = 0.1966), o que indica que o modelo não apresenta erro de especificação.

## 5.1 DADOS DE PRODUÇÃO DE GASOLINA

A qualidade do ajuste de cada modelo também foi avaliada através do pseudo- $R^2$ , do Critério de Informação de Akaike (AIC) e do Critério de Informação Bayesiano (BIC). Aqui, quanto maior o pseudo- $R^2$  melhor o ajuste do modelo, e quanto menores os valores de AIC e BIC, maiores os indicativos de que o modelo correspondente é adequado. Os três critérios favorecem o modelo 2 em detrimento do modelo 1, como pode ser verificado na Tabela 5.2. Com base nos gráficos de resíduos deviance com envelopes simulados (Figura 5.1), os dois modelos apresentam bom ajuste.

**Tabela 5.2** Medidas de avaliação dos modelos 1 e 2.

Modelo	pseudo- $R^2$	AIC	BIC
1	0.9519	-147.9541	-128.8996
2	0.9852	-168.3101	-150.7213



**Figura 5.1** Gráficos dos resíduos deviance com envelopes simulados para os modelos 1 e 2.

Como os modelos 1 e 2 são não-encaixados, os testes  $J$  e  $MJ$  são adequados para identificar a presença do modelo correto e selecionar aquele que melhor representa a relação entre a variável resposta e as covariáveis. Ambos os testes foram aplicados usando as estatísticas da razão de verossimilhanças e de Wald, bem como suas versões *bootstrap*. A aplicação do teste  $J$  se deu em duas etapas: para testar o modelo 1 foi avaliada a significância da inclusão do preditor linear do modelo 2 como regressor – na tabela: “mod.1 vs mod.2” –; de modo semelhante, o modelo 2 foi testado por meio da avaliação da inclusão do preditor linear do modelo 1 – “mod.2 vs mod.1”. No esquema *bootstrap* foram utilizadas 1000 réplicas de acordo com os procedimentos descritos na Seção 3.6. Na Tabela 5.3 são apresentados os  $p$ -valores dos testes. Para os níveis de significância usuais, o teste  $J$  rejeitou o modelo 1 e não rejeitou o

## 5.2 HABILIDADE DE LEITURA EM CRIANÇAS DISLÉXICAS E NÃO-DISLÉXICAS

modelo 2. O teste  $MJ$  não rejeitou a hipótese nula de que um dos modelos estava corretamente especificado; além disso, a estatística do teste  $MJ$  correspondeu ao modelo 2, o que indica que o modelo com ligação log-log é o modelo mais adequado. Portanto, conclui-se que o modelo de regressão beta mais adequado para modelagem da proporção média de petróleo convertido em gasolina após destilação e fracionamento é a aquele com ligação log-log no submodelo da média e com dispersão constante. Além disso, vale salientar que neste caso o teste  $J$  selecionou apenas o modelo  $H_2$  como correto, uma vez que, dentre os resultados possíveis, ambos modelos poderiam estar corretos segundo o teste. Os resultados dos dois testes confirmam então as indicações do teste RESET, do pseudo- $R^2$  e dos critérios AIC e BIC.

**Tabela 5.3**  $p$ -valores do teste  $J$  usando as estatísticas da razão de verossimilhança ( $RV$ ) e de Wald e suas versões *bootstrap*.

Estatística utilizada	Teste $J$		Teste $MJ$
	mod.1 vs mod.2	mod.2 vs mod.1	
$RV$	$1.828 \times 10^{-5}$	0.4869	0.4869
$RV_{boot}$	0.0040	0.6094	0.5554
Wald	$2.657 \times 10^{-8}$	0.4863	0.4863
Wald <sub>boot</sub>	0.0070	0.5594	0.6114

## 5.2 Habilidade de leitura em crianças disléxicas e não-disléxicas

Smithson & Verkuilen (2006) avaliaram um conjunto de dados relativo à habilidade de leitura de crianças disléxicas e não-disléxicas utilizando uma amostra contendo 44 observações. A variável de interesse é a pontuação em um teste de habilidade de leitura e as variáveis explicativas são o *status* de dislexia ( $x_1$ ) e o escore padronizado de QI não-verbal ( $x_2$ ). A variável explicativa  $x_1$  assume valor 1 se a criança é disléxica e  $-1$  caso contrário. A amostra contém 44 crianças, sendo 19 disléxicas e 25 não-disléxicas. Ao analisarem os dados, Cribari-Neto & Queiroz (2012) e Grün et al. (2012) obtiveram modelos distintos, com diferença em um dos regressores do submodelo da precisão. O modelo obtido por Cribari-Neto & Queiroz (2012), aqui denominado modelo A, é dado por

$$\begin{aligned} \log\left(\frac{\mu_t}{1-\mu_t}\right) &= \beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} + \beta_3 x_{t1} x_{t2} \\ \log(\phi_t) &= \gamma_0 + \gamma_1 x_{t1} + \gamma_2 x_{t2} + \gamma_3 (x_{t2})^2, \end{aligned}$$

enquanto Grün et al. (2012) consideraram o modelo especificado da seguinte forma (modelo B):

$$\begin{aligned} \log\left(\frac{\mu_t}{1-\mu_t}\right) &= \beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} + \beta_3 x_{t1} x_{t2} \\ \log(\phi_t) &= \gamma_0 + \gamma_1 x_{t1} + \gamma_2 x_{t2} + \gamma_3 x_{t1} x_{t2}, \end{aligned}$$

com  $t = 1, \dots, 44$ .

No modelo A é utilizado o quadrado da covariada  $x_2$  na especificação do submodelo da precisão, enquanto no modelo B é utilizado  $x_{t1} x_{t2}$ . Na Tabela 5.4 são apresentadas as estimativas

## 5.2 HABILIDADE DE LEITURA EM CRIANÇAS DISLÉXICAS E NÃO-DISLÉXICAS

dos parâmetros dos modelos, os respectivos desvios-padrão, as estatísticas e  $p$ -valores do teste  $z$ . Ao nível de 5% de significância, todas as variáveis explicativas incluídas na modelagem da média e da precisão nos dois modelos são significativas.

**Tabela 5.4** Estimativas dos parâmetros dos modelos  $A$  e  $B$  utilizados na modelagem de dados de habilidade de leitura em crianças disléxicas e não-disléxicas.

Modelo	Parâmetro	Estimativa	Erro-padrão	$z$	$p$ -valor
$A$	$\beta_0$	1.0402	0.1467	7.092	$1.32 \times 10^{-12}$
	$\beta_1$	-0.6764	0.1467	-4.611	$4.01 \times 10^{-06}$
	$\beta_2$	0.7683	0.0880	8.734	$2.00 \times 10^{-16}$
	$\beta_3$	-0.8375	0.0880	-9.519	$2.00 \times 10^{-16}$
	$\gamma_0$	2.5759	0.2973	8.664	$< 2.00 \times 10^{-16}$
	$\gamma_1$	1.7704	0.2541	6.968	$3.21 \times 10^{-12}$
	$\gamma_2$	1.9178	0.2491	7.698	$1.39 \times 10^{-14}$
	$\gamma_3$	1.2181	0.2057	5.923	$3.17 \times 10^{-09}$
$B$	$\beta_0$	1.0187	0.1452	7.014	$2.31 \times 10^{-12}$
	$\beta_1$	-0.6378	0.1452	-4.392	$1.13 \times 10^{-05}$
	$\beta_2$	0.6897	0.1268	5.440	$5.32 \times 10^{-08}$
	$\beta_3$	-0.7760	0.1268	-6.121	$9.33 \times 10^{-10}$
	$\gamma_0$	3.0401	0.2577	11.796	$2.00 \times 10^{-16}$
	$\gamma_1$	1.7675	0.2577	6.858	$6.98 \times 10^{-12}$
	$\gamma_2$	1.4369	0.2574	5.583	$2.37 \times 10^{-08}$
	$\gamma_3$	-0.6109	0.2574	-2.373	$1.76 \times 10^{-02}$

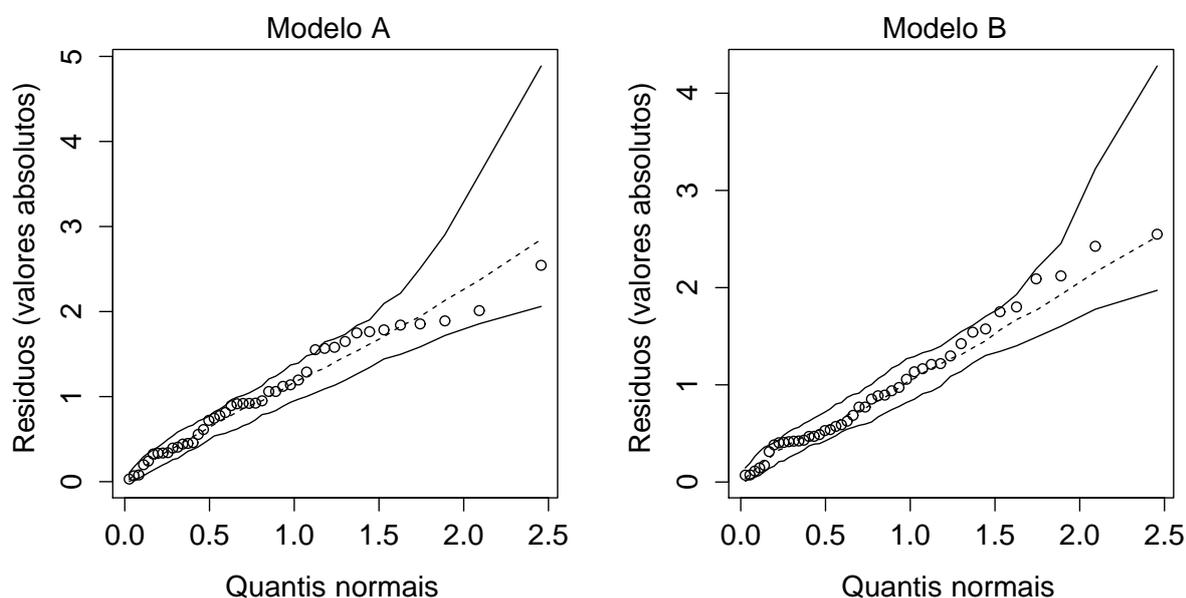
Para verificar a existência de algum erro de especificação nos modelos foi utilizado o teste RESET, sendo adicionado ao submodelo da média o quadrado do preditor linear estimado. A exclusão de  $\hat{\eta}_A^2$  no modelo  $A$  foi testada utilizando o teste da razão de verossimilhanças e conclui-se que tal variável não apresenta melhorias no ajuste ( $\chi_{0,95,1}^2 = 0.312$ ,  $p$ -valor = 0.577). De forma semelhante, a inclusão de  $\hat{\eta}_B^2$  no modelo  $B$  não melhorou o ajuste ( $\chi_{0,95,1}^2 = 0.519$ ,  $p$ -valor = 0.471). Assim, de acordo com o teste RESET não é necessária a inclusão de outros regressores nos modelos.

Para avaliar a qualidade do ajuste também foram calculados o pseudo- $R^2$ , AIC e BIC, apresentados na Tabela 5.5. O pseudo- $R^2$  do modelo  $B$  é ligeiramente maior, entretanto os critérios de informação de Akaike e Bayesiano favorecem o modelo  $A$ . De acordo com os gráficos de resíduos dos dois modelos (Figura 5.2), ambos os modelos apresentam bom ajuste.

**Tabela 5.5** Medidas de avaliação dos modelos  $A$  e  $B$  utilizados na modelagem de dados de habilidade de leitura em crianças disléxicas e não-disléxicas.

Modelo	pseudo- $R^2$	AIC	BIC
$A$	0.5145	-131.3091	-117.0356
$B$	0.5175	-117.4682	-103.1947

## 5.2 HABILIDADE DE LEITURA EM CRIANÇAS DISLÉXICAS E NÃO-DISLÉXICAS



**Figura 5.2** Gráficos dos resíduos deviance com envelopes simulados para os modelos A e B utilizados na modelagem de dados de habilidade de leitura em crianças disléxicas e não-disléxicas.

**Tabela 5.6**  $p$ -valores dos testes  $J$  e  $MJ$  usando as estatísticas da razão de verossimilhança ( $RV$ ) e de Wald e suas versões *bootstrap* para os modelos com dados de habilidade de leitura em crianças disléxicas e não-disléxicas.

Estatística utilizada	Teste $J$		Teste $MJ$
	mod.A vs mod.B	mod.B vs mod.A	
$RV$	0.2089	$5.86 \times 10^{-7}$	0.2089
$RV_{boot}$	0.3397	0.0030	0.2448
Wald	0.1182	$< 2.2 \times 10^{-16}$	0.1182
$Wald_{boot}$	0.3646	0.0080	0.3347

Os testes  $J$  e  $MJ$  foram utilizados para identificar qual dos modelos está corretamente especificado. Os testes foram aplicados utilizando as estatísticas de Wald e da razão de verossimilhanças, juntamente com as respectivas versões *bootstrap* – utilizando 1000 réplicas. Como se pode observar na Tabela 5.6, o teste  $J$  sinaliza a não rejeição do modelo A e a rejeição do modelo B. Isto é, com base no teste  $J$  é possível concluir que o modelo A está corretamente especificado, enquanto o modelo B não está. O teste  $MJ$  indica que um dos modelos testados é o modelo mais adequado. Adicionalmente, a menor estatística  $J$  corresponde ao modelo A e podemos utilizar este resultado como indicativo de que este é o modelo corretamente especificado. Portanto, conclui-se que o modelo apresentado por Cribari-Neto & Queiroz (2012) é o modelo mais adequado para modelar os dados de habilidade de leitura em crianças disléxicas e não-disléxicas.

## Considerações Finais

Os modelos de regressão beta permitem modelar variáveis resposta restritas ao intervalo unitário padrão, incorporando de forma natural a heteroscedasticidade e a assimetria inerentes a esses tipos de dados. A partir da parametrização da função de densidade beta proposta por Ferrari & Cribari-Neto (2004), a média da resposta é modelada diretamente e envolve um parâmetro de precisão, tornando-se um importante instrumento de modelagem devido a sua flexibilidade e possibilidade de aplicação.

Uma vez estimado o modelo, usualmente são realizados testes para determinar se o modelo em análise é compatível com os dados observados. Quando um dos modelos em análise não pode ser obtido como caso particular de outro por meio de restrições sobre os parâmetros que os indexam, estes são ditos não encaixados. No caso de modelos de regressão beta, os modelos podem diferir nos regressores ou na função de ligação do submodelo da média, da precisão ou de ambos conjuntamente.

O teste  $J$  proposto por Davidson & MacKinnon (1981) é o procedimento mais utilizado quando se trata de modelos de regressão não-encaixados e se baseia em um modelo artificial construído de tal forma que cada modelo avaliado possa ser obtido deste como um caso particular. Entretanto, esse teste requer a utilização de testes sequenciais de forma a avaliar cada modelo na hipótese nula. Neste sentido, Hagemann (2012) propôs o teste  $MJ$ , que permite avaliar a presença do modelo corretamente especificado entre os modelos testados, evitando testes sequenciais e possíveis distorções de tamanho.

A presente dissertação teve por objetivo adaptar os testes  $J$  e  $MJ$  para modelos de regressão beta, uma vez que originalmente esses testes foram apresentados para modelos lineares de regressão. Foram avaliados os desempenhos dos testes e suas versões *bootstrap* em amostras finitas, comparativamente aos testes assintóticos. Foram considerados modelos de regressão beta com dispersão variável e avaliados cenários envolvendo dois modelos concorrentes.

De forma geral, tamanhos amostrais maiores estão associados a taxas de rejeição nula mais próximas dos níveis nominais. A utilização de um esquema *bootstrap* reduziu consideravelmente distorções de tamanho. Nos casos em que os modelos diferem nos regressores da média ou nos regressores da média e da precisão conjuntamente, os testes  $J$  e  $MJ$  apresentaram o mesmo desempenho no que diz respeito às taxas de rejeição nula. Nos demais cenários, o teste  $MJ$  apresentou distorções tipicamente menores.

Na presença do verdadeiro modelo, o teste  $MJ$  selecionou corretamente o modelo em todas as réplicas de Monte Carlo quando a hipótese nula não foi rejeitada e os modelos diferiam nos regressores da média ou da média e da precisão conjuntamente. Para modelos com diferença na função de ligação, o teste  $MJ$  selecionou o modelo correto em pouco mais de 50% das réplicas. Nos modelos com diferença na função de ligação da precisão, o mecanismo de seleção de

## CAPÍTULO 6 CONSIDERAÇÕES FINAIS

modelos baseado na estatística  $MJ$  apresentou os menores percentuais de seleção do modelo correto.

Quando avaliado o poder dos testes nos diferentes cenários, as taxas de rejeição não-nula de ambos os testes foram semelhantes quando havia diferença nos regressores da média e nos regressores da precisão. Nos casos em que os modelos diferiam na função de ligação da média, o teste  $MJ$  apresentou maior poder em tamanhos de amostra maiores. Nos outros cenários o teste  $J$  apresentou maior poder. Por fim, a versão *bootstrap* do teste  $MJ$  melhorou o desempenho do teste no tocante ao poder, principalmente em modelos com diferença na função de ligação da média.

Com base nas conclusões apresentadas, ambos os testes apresentaram bom desempenho no tocante às taxas de rejeição nula, sendo que as versões *bootstrap* são fortemente recomendadas em pequenos tamanhos de amostra. O teste  $MJ$  é recomendável por evitar os testes sequenciais necessários à aplicação do teste  $J$ , principalmente quando dentre os modelos avaliados há diferença nos regressores.

## Referências Bibliográficas

- Atkinson, A. (1970), 'A method for discriminating between models (with discussion)', *Journal of the Royal Statistical Society B* **32**, 323–353.
- Bayer, F. M. (2011), Modelagem e inferência em regressão beta, tese de doutorado, Universidade Federal de Pernambuco, Recife.
- Beran, R. (1988), 'Prepivoting test statistics: a bootstrap view of asymptotic refinements', *Journal of the American Statistical Association* **83**, 687–697.
- Berger, R. L. (1982), 'Multiparameter hypothesis testing and acceptance sampling', *Technometrics* **24**, 295–300.
- Bury, K. (1999), *Statistical Distributions in Engineering*, Cambridge University Press, Cambridge.
- Cox, D. (1961), 'Tests of separated families of hypothesis', *Proceedings of the Fourth Berkley Symposium on Mathematical Statistics and Probability* .
- Cox, D. (1962), 'Further results on tests of separate families of hypotheses', *Journal of the Royal Statistical Society B* **24**, 406–424.
- Cribari-Neto, F. & Lima, L. B. (2007), A misspecification test for beta regressions, Technical report.
- Cribari-Neto, F. & Queiroz, M. P. (2012), 'On testing inference in beta regressions', *Journal of Statistical Computation and Simulation* .
- Cribari-Neto, F. & Zeileis, A. (2010), 'Beta regression in R', *Journal of Statistical Software* **34**(2), 1–24.
- Dastoor, N. (1983), 'Some aspects of testing nonnested hypotheses', *Journal of Econometrics* **21**, 213–218.
- Davidson, R. & MacKinnon, J. G. (2009), *Econometric Theory and Methods*, Oxford University Press.
- Davidson, R. & MacKinnon, J. (1981), 'Several tests for model specification in the presence of alternative hypotheses', *Econometrica* **49**, 781–793.

## REFERÊNCIAS BIBLIOGRÁFICAS

- Deaton, A. S. (1982), *Evaluating the Reliability of Macroeconomic Models*, Wiley, New York, chapter Model selection procedures, or, does the consumption function exist?, pp. 43–65.
- Doornik, J. A. (2009), *Ox: an objected-oriented matrix language*, 4th edn, Timberlake Consultants Press and Oxford, London.  
**URL:** <http://www.doornik.com/>
- Efron, B. (1979), ‘Bootstrap methods: another look at the jackknife’, *Annals of Statistics* **7**, 1–26.
- Espinheira, P. L., Ferrari, S. L. P. & Cribari-Neto, F. (2008), ‘On beta regression residuals’, *Journal of Applied Statistics* **35**, 407–419.
- Fan, Y. & Li, Q. (1995), ‘Bootstrapping J-type tests for non-nested regression models’, *Economic Letters* **48**, 107–112.
- Ferrari, S. L. P. & Cribari-Neto, F. (2004), ‘Beta regression for modelling rates and proportions’, *Journal of Applied Statistics* **31**, 799–815.
- Fingleton, B. (2005), ‘Beyond neoclassical ortodoxy: a view based on the new economic geography and uk regional wave data’, *Papers in Regional Science* **84**(3), 351–375.
- Godfrey, L. (1998), ‘Tests of non-nested regression models: some results on small sample behaviour and the bootstrap’, *Journal of Econometrics* **84**, 59–74.
- Godfrey, L. G. (2011), ‘Robust non-nested testing for ordinary least squares regression when some of the regressors are lagged dependent variables’, *Oxford Bulletin of Economics and Statistics* **73**(5), 651–668.
- Gouriéroux, C., Monfort, A. & Trognon, A. (1983), ‘Testing nested or nonnested hypotheses’, *Journal of Econometrics* **21**, 83–115.
- Grün, B., Kosmidis, I. & Zeileis, A. (2012), ‘Extended beta regression in r: Shaken, stirred, mixed, and partitioned’, *Journal of Statistical Software* **48**(11), 1–25.
- Hagemann, A. (2012), ‘A simple test for regression specification with non-nested alternatives’, *Journal of Econometrics* **166**, 247–254.
- Johnson, N. L., Kotz, S. & Balakrishnan, N. (1995), *Continuous Univariate Distributions*, Vol. 2, 2nd edn, Wiley, New York.
- Kieschnick, R. & McCullough, B. D. (2003), ‘Regression analysis of variates observed on (0,1): percentages, proportions and fractions’, *Statistical Modelling* **3**, 193–213.
- Lamport, L. (1986), *LATEX: a document preparation system*, Addison-Wesley Pub. Co.
- MacKinnon, J., White, H. & Davidson, R. (1983), ‘Tests for model specifacaton in the presence of alternative hypotheses: some further results’, *Journal of Econometrics* **21**, 53–70.

## REFERÊNCIAS BIBLIOGRÁFICAS

- Marsaglia, G. (1997), 'A random number generator for C'. Discussion paper, posting on sci.stat.math usenet group.
- McAleer, M. (1995), 'The significance of testing empirical non-nested models', *Journal of Econometrics* **67**, 149–171.
- McCullagh, P. & Nelder, J. A. (1989), *Generalized Linear Models*, Chapman and Hall, London.
- Michelis, L. (1999), 'The distribution of the J and cox non-nested tests in regression models with weakly correlated regressors', *Journal of Econometrics* **93**, 369–401.
- Mittelbach, F., Goossens, M., Braams, J. & Rowley, C. (2004), *The LaTeX Companion*, Addison-Wesley series on tools and techniques for computer typesetting, Addison-Wesley.
- Mizon, G. E. & Richard, J. F. (1986), 'The encompassing principle and its applications to non-nested hypotheses', *Econometrica* **54**, 657–678.
- Nocedal, J. & Wright, S. J. (1999), *Numerical Optimization*, Springer-Verlag, New York.
- Ospina, R., Cribari-Neto, F. & Vasconcellos, K. L. P. (2006), 'Improved point and interval estimation for a beta regression model', *Computational Statistics & Data Analysis* **51**, 960–981.
- Ospina, R. & Ferrari, S. L. P. (2010), 'Inflated beta distributions', *Statistical Papers* **51**, 111–216.
- Ospina, R. & Ferrari, S. L. P. (2012), 'A general class of zero-or-one in ated regression models', *Computational Statistics & Data Analysis* **56**, 1609–1623.
- Pesaran, M. H. & Weeks, M. (2001), *A Companion to Theoretical Econometrics*, Blackwell, Malden, chapter Nonnested hypothesis testing: an overview, pp. 279–309.
- Prater, N. H. (1956), 'Estimate gasoline yields from crudes', *Petroleum Refiner* **35**(5), 236–238.
- Queiroz, M. P. F. (2011), Testes de hipóteses em regressão beta baseados em verossimilhança perfilada ajustada e em bootstrap, Master's thesis, Universidade Federal de Pernambuco.
- Ramalho, E. A., Ramalho, J. J. S. & Murteira, J. M. R. (2011), 'Alternativa estimating and testing empirical strategies for fractional regression models', *Journal of Economic Surveys* **25**(1), 19–68.
- Ramsey, J. B. (1969), 'Tests for specification error in classical linear least squares regression analysis', *Journal of the Royal Statistical Society B* **31**, 350–371.
- Rao, C. R. (1973), *Linear Statistical Inference and Its Applications*, Wiley, New York.
- Rocha, A. V. & Simas, A. B. (2011), 'Influence diagnostics in a general class of beta regression models', *Test* **20**, 95–119.

## REFERÊNCIAS BIBLIOGRÁFICAS

- Sapra, S. K. (2008), 'Robust nonnested hypothesis testing', *Applied Economics Letters* **15**(1), 1–4.
- Simas, A. B., Barreto-Souza, W. & Rocha, A. V. (2010), 'Improved estimators for a general class of beta regression models', *Computational Statistics and Data Analysis* **54**, 348–366.
- Smithson, M. & Verkuilen, J. (2006), 'A better lemon squeezer? maximum-likelihood regression with beta-distributed dependent variables', *Psychological Methods* **11**(7), 54–71.
- Vasconcellos, K. L. P. & Cribari-Neto, F. (2005), 'Improved maximum likelihood estimation in a new class of beta regression models', *Brazilian Journal of Probability and Statistics* **19**, 13–31.
- Venerables, W. N., Smithson, D. M. & R Development Core Team (2009), *An Introduction to R*, 2nd edn, Network Theory Limited.
- Wooldridge, J. M. (1990), 'A unified approach to robust, regression-based specification tests', *Econometric Theory* **6**, 17–43.

# APÊNDICE A

## Códigos em Ox

A seguir são apresentados os códigos em Ox utilizados nas avaliações numéricas apresentadas na Seção 4.

### A.1 Modelos com diferença nos regressores da média

```
/******  
*  
*   PROGRAMA: betareg_nonnested_media_n20_4.0.ox  
*  
*   DESCRIÇÃO: realiza os testes J (com teste da razão de veros-  
*   similhanças) e MJ para modelos de regressão beta com disper-  
*   são variável  
*  
*   MODELOS: (M1) logit(mu_t) = beta0 + beta1*x1 + beta2*x2  
*             log(phi_t) = gama0 + gama1*x1  
*  
*             (M2) logit(mu_t) = beta0 + beta1*x1 + beta2*x3  
*             log(phi_t) = gama0 + gama1*x1  
*  
*   AUTOR: Sadraque Eneas de Figueiredo Lucena  
*  
*   VERSÃO: 3.0  
*  
*   DATA: 21/05/2012  
*  
*   OBS.: Programa desenvolvido com base no programa apresentado  
*   na dissertação de mestrado de Queiroz (2011).  
*  
*****/  
  
/* Arquivos cabeçalho */  
#include <oxstd.h> /* Biblioteca padrão do OX */  
#include <oxprob.h> /* Necessário para gerar ocorrências beta */  
#import <maximize> /* Necessário para otimização não-linear */  
  
/* Variáveis globais */  
static decl s_vY; /* Amostra aleatória */  
static decl s_mX1; /* Matriz X1 */  
static decl s_mX2; /* Matriz X2 */  
static decl s_mZ; /* Matriz Z */  
static decl s_vYB; /* Amostra bootstrap */  
static decl s_mCovMU; /* Matriz de covariáveis do modelo para mu */  
static decl s_mCovPHI; /* Matriz de covariáveis do modelo para phi */  
  
static decl boot = 0;  
  
/* Definição de constantes */  
const decl NOBS = 20; /* Tamanho amostral */  
const decl NREP = 10000; /* Número de réplicas de Monte Carlo */  
const decl NBOOT = 1000; /* Número de amostras bootstrap */
```

## A.1 MODELOS COM DIFERENÇA NOS REGRESSORES DA MÉDIA

```

const decl k = 3; /* Número de parâmetros no modelo para mu */
const decl m = 2; /* Número de parâmetros no modelo para phi */
const decl beta = <1.0, 2.2, -3.7>; /* Parâmetros do modelo para mu */
const decl gama = <1.5, 2.3>; /* Parâmetros do modelo para phi */

/* Algumas funções */
EstMV(const ystar, const mX, const mZ); /* EMV do modelo com regressores X e Z */

/* Função de log-verossimilhança do modelo com ligações logit(mu) e log(phi) */
floglik_logit_log(const vtheta, const adFunc, const avScore, const amHess)
{
decl y;
if(boot == 1) /* usar amostra bootstrap */
y = s_vYB;
if(boot == 0) /* usar amostra */
y = s_vY;

decl k = sizec(s_mCovMU); /* número de colunas da matriz de regressores de mu */
decl m = sizec(s_mCovPHI); /* número de colunas da matriz de regressores de phi */
decl eta = s_mCovMU * vtheta[0:(k-1)]; /* usado para obter mu */
decl mu = exp(eta) ./ (1.0 + exp(eta)); /* inversa da ligação logit */
decl delta = s_mCovPHI * vtheta[k:(k+m-1)]; /* usado para obter phi */
decl phi = exp(delta); /* inversa da ligação log */
decl i, funlog = zeros(NOBS, 1);
for(i = 0; i < NOBS; i++) /* logaritmo da densidade */
funlog[i] = log( densbeta(y[i], mu[i] * phi[i], (1 - mu[i]) * phi[i]) );

adFunc[0] = double( sumc( funlog ) ); /* log-verossimilhança */

/* Parâmetros utilizados no cálculo das primeiras derivadas */
decl ystar = log(y ./ (1.0 - y));
decl mustar = polygamma(mu.*phi,0) - polygamma((1.0-mu).*phi,0);
decl ydag = log(1.0 - y);
decl mudag = polygamma((1.0-mu).*phi,0) - polygamma(phi,0);
decl T = diag( mu .* (1-mu)); /* 1/g'(mu) */
decl F = diag(phi);
decl H = diag(phi); /* 1/h'(phi) */
decl M = diag(mu);

if(avScore){
(avScore[0])[0:(k-1)] = s_mCovMU'*T*F*(ystar-mustar);
(avScore[0])[k:(k+m-1)] = s_mCovPHI'*H*(M*(ystar-mustar)+(ydag-mudag));
}

if( isnan(adFunc[0]) || isdotinf(adFunc[0]) )
return 0;
else
return 1; /* 1 indica sucesso */
}

/* INÍCIO DO PROGRAMA */
main()
{
/* Declaração de variáveis utilizadas no programa */
/* Parâmetros do sistema */
decl dExecTime, i, j, b, falha1MC, falha2MC, falha1MCampliado, falha2MCampliado,
acertoJ, acertoJb, acertoMJ, acertoMJb, falha1BOOT, falha2BOOT, falha1BOOTampliado,
falha2BOOTampliado, acertomm, acertomm;
/* Parâmetros do modelo */
decl eta, mu, delta, phi, irl, ir2, irlampliado, ir2ampliado;
/* Parâmetros do modelo bootstrap */
decl irlb, ir2b, irlampliadob, ir2ampliadob;
/* Parâmetros dos testes */
decl estat_testeJ1, estat_testeJ2, estat_testeJ1b, estat_testeJ2b, estatMJ, estatMJb;
/* Parâmetros auxiliares */
decl chi2, ystar, ystarb;

```

## A.1 MODELOS COM DIFERENÇA NOS REGRESSORES DA MÉDIA

```

dExecTime = timer(); /* Inicia o relógio */

ranseed("GM"); /* Gerador de números aleatórios */
ranseed({6548,8984}); /* Semente */

/* Inicializações */
s_vY = s_vYB = zeros(NOBS, 1); /* Vetores de amostra e amostra bootstrap */
irl = ir2 = zeros((k+m+2), NREP); /* Estimativas de MV */
irlb = ir2b = zeros((k+m+2), NBOOT); /* Estimativas de MV bootstrap */
falha1MC = falha2MC = falha1BOOT = falha2BOOT = 0; /* Contadores de falha */
falha1MCampliado = falha2MCampliado = 0; /* Contadores de falha */
falha1BOOTampliado = falha2BOOTampliado = 0; /* Contadores de falha */
acertomm = acertomm = acertoJ = acertoJb = acertoMJ = acertoMJb = zeros(3,1); /* Contador
de acertos do teste J */
estat_testeJ1b = estat_testeJ2b = estatMJb = zeros(NBOOT, 1); /* Armazena est. bootstrap */

/* Quantis da qui-quadrado */
chi2 = quanchi(.99,1) | quanchi(.95,1) | quanchi(.90,1);

/* Obtendo matrizes de regressores */
s_mX1 = 1 ~ (ranu(10,k-1)); /* matriz do modelo 1 */
s_mX2 = s_mX1[][0:1] ~ (ranu(10,1)); /* matriz do modelo 2 */
/* Replicando para manter o grau de heterogeneidade constante */
for(i = 10; i < NOBS; i += 10){
s_mX1 = s_mX1 | s_mX1[0:9][];
s_mX2 = s_mX2 | s_mX2[0:9][];
}
s_mZ = s_mX1[][0:1];

/* Valores verdadeiros dos parâmetros */
eta = s_mX1 * beta';
mu = exp(eta) ./ (1.0 + exp(eta)); /* inversa da ligação logit */
delta = s_mZ * gama';
phi = exp(delta); /* inversa da ligação log */

decl lambda = max(phi)/min(phi);

/* Início do laço de Monte Carlo */
for(i = 0; i < NREP; i++){
for(j = 0; j < NOBS; j++){
s_vY[j] = ranbeta(1,1,mu[j]*phi[j], (1.0-mu[j])*phi[j]); /* beta(mu,phi) */

/* Obtendo y* */
ystar = log( s_vY ./ (1.0-s_vY) ); /* transformação logit */

/* Estimando o modelo m1 - regressores X1 (mu) e Z (phi) */
irl[][i] = EstMV(ystar, s_mX1, s_mZ);
if(irl[k+m+1][i] != MAX_CONV && irl[k+m+1][i] != MAX_WEAK_CONV){
falha1MC++;
i--;
continue;
}

/* Estimando o modelo m2 - regressor X2 (mu) e Z (phi) */
ir2[][i] = EstMV(ystar, s_mX2, s_mZ);
if(ir2[k+m+1][i] != MAX_CONV && ir2[k+m+1][i] != MAX_WEAK_CONV){
falha2MC++;
i--;
continue;
}

decl etab1 = s_mX1*irl[0:(k-1)][i];
decl mub1 = exp(etab1) ./ (1.0 + exp(etab1)); /* inversa da ligação logit */
/* mub1 será também utilizado para gerar amostras do modelo 1 artificial */

```

## A.1 MODELOS COM DIFERENÇA NOS REGRESSORES DA MÉDIA

```

decl etab2 = s_mX2*ir2[0:(k-1)][i];
decl mub2 = exp(etab2) ./ (1.0 + exp(etab2)); /* inversa da ligação logit */

/* Ampliando o modelo 1: incluindo eta do modelo 2 */
irlampliado = EstMV(ystar, s_mX1~mub2, s_mZ );
if(irlampliado[k+m+2] != MAX_CONV && irlampliado[k+m+2] != MAX_WEAK_CONV){
falha1MCampliado++;
i--;
continue;
}

/* Ampliando o modelo 2: incluindo eta do modelo 1 */
ir2ampliado = EstMV(ystar, s_mX2~mub1, s_mZ );
if(ir2ampliado[k+m+2] != MAX_CONV && ir2ampliado[k+m+2] != MAX_WEAK_CONV){
falha2MCampliado++;
i--;
continue;
}

/* Teste J - usando estatística RV */
estat_testeJ1 = 2*(irlampliado[k+m+1]-ir1[k+m][i]); /* modelo 1 */

/* Teste MJ */
estat_testeJ2 = 2*(ir2ampliado[k+m+1]-ir2[k+m][i]); /* modelo 2 */
estatMJ = min(estat_testeJ1, estat_testeJ2);

/* Parâmetros a serem usados no laço bootstrap */
/* Modelo 1 artificial */
decl deltab1 = s_mZ * ir1[k:(k+m-1)][i];
decl phib1 = exp(deltab1);
/* mub1 definido anteriormente */

boot = 1; /* indica que vai entrar no laço bootstrap */

/* Início do laço bootstrap */
for(b = 0; b < NBOOT; b++){

/** Modelo 1 **/
/* y* sob Ho: alpha=0 */
for(j = 0; j < NOBS; j++)
s_vYB[j] = ranbeta(1,1,mub1[j]*phib1[j], (1.0-mub1[j])*phib1[j]); /* beta(mu,phi) */

ystar = log( s_vYB ./ (1.0-s_vYB) ); /* transformação logit */

/* Estimando o modelo 1 artificial - regressores X (mu) e Z (phi) */
irlb[][b] = EstMV(ystar, s_mX1, s_mZ);
if(irlb[k+m+1][b] != MAX_CONV && irlb[k+m+1][b] != MAX_WEAK_CONV){
falha1BOOT++; /* contando falhas de convergência */
b--;
continue;
}

/* Ampliando o modelo 1 artificial: incluindo eta do modelo 2 */
irlampliadob = EstMV(ystarb, s_mX1~mub2, s_mZ);
if(irlampliadob[k+m+2] != MAX_CONV && irlampliadob[k+m+2] != MAX_WEAK_CONV){
falha1BOOTampliado++; /* contando falhas de convergência */
b--;
continue;
}

/* Estimando o modelo 2 artificial - regressores X (mu e phi) */
ir2b[][b] = EstMV(ystarb, s_mX2, s_mZ);
if(ir2b[k+m+1][b] != MAX_CONV && ir2b[k+m+1][b] != MAX_WEAK_CONV){
falha2BOOT++; /* contando falhas de convergência */
b--;
continue;
}

```

## A.1 MODELOS COM DIFERENÇA NOS REGRESSORES DA MÉDIA

```
}

/* Ampliando o modelo 2 artificial: incluindo eta do modelo 1 */
ir2ampliados = EstMV(ystarb, s_mX2~mub1, s_mZ);
if(ir2ampliados[k+m+2] != MAX_CONV && ir2ampliados[k+m+2] != MAX_WEAK_CONV){
falha2BOOTampliados++; /* contando falhas de convergência */
b--;
continue;
}

/* Estatística do teste J bootstrap - usando estatística escore */
estat_testeJlb[b] = 2*(ir1ampliados[k+m+1] - ir1b[k+m][b]); /* modelo 1 */

/* Estatística do teste MJ bootstrap */
estat_testeJ2b[b] = 2*(ir2ampliados[k+m+1] - ir2b[k+m][b]); /* modelo 2 */
estatMJb[b] = min(estat_testeJlb[b], estat_testeJ2b[b]);
} /* Fim do laço bootstrap */

boot = 0; /* Indica que saiu do laço bootstrap */

/* Contagem de acertos do teste J */
if(estat_testeJ1 < chi2[0]) /* 1% sig. */
acertoJ[0]++;
if(estat_testeJ1 < chi2[1]) /* 5% sig. */
acertoJ[1]++;
if(estat_testeJ1 < chi2[2]) /* 10% sig. */
acertoJ[2]++;

/* Contagem de acertos do teste J bootstrap */
if(estat_testeJ1 < quantilec(estat_testeJlb, .99)) /* 1% sig. */
acertoJb[0]++;
if(estat_testeJ1 < quantilec(estat_testeJlb, .95)) /* 5% sig. */
acertoJb[1]++;
if(estat_testeJ1 < quantilec(estat_testeJlb, .90)) /* 10% sig. */
acertoJb[2]++;

/* Contagem de acertos do teste MJ */
if(estatMJ < chi2[0]) /* 1% sig. */
acertoMJ[0]++;
if(estatMJ < chi2[1]) /* 5% sig. */
acertoMJ[1]++;
if(estatMJ < chi2[2]) /* 10% sig. */
acertoMJ[2]++;

/* Contagem de acertos do teste MJ bootstrap */
if(estatMJ < quantilec(estatMJb, .99)) /* 1% sig. */
acertoMJb[0]++;
if(estatMJ < quantilec(estatMJb, .95)) /* 5% sig. */
acertoMJb[1]++;
if(estatMJ < quantilec(estatMJb, .90)) /* 10% sig. */
acertoMJb[2]++;

/* Contagem de m=m* quando Ho nao foi rejeitada */
if(estatMJ < chi2[0] && estatMJ == estat_testeJ1)
acertomm[0]++;
if(estatMJ < chi2[1] && estatMJ == estat_testeJ1)
acertomm[1]++;
if(estatMJ < chi2[2] && estatMJ == estat_testeJ1)
acertomm[2]++;

/* Contagem de m=m* quando Ho nao foi rejeitada via bootstrap */
if(estatMJ < quantilec(estatMJb, .99) && estatMJ == estat_testeJ1)
acertomb[0]++;
if(estatMJ < quantilec(estatMJb, .95) && estatMJ == estat_testeJ1)
acertomb[1]++;
if(estatMJ < quantilec(estatMJb, .90) && estatMJ == estat_testeJ1)
```

## A.1 MODELOS COM DIFERENÇA NOS REGRESSORES DA MÉDIA

```

    acertommb[2]++;
} /* Fim do laço de Monte Carlo */

/* Impressão de resultados */
println( "\n\n PROGRAMA OX: ", oxfilename(0) );
println( " VERSAO OX: ", oxversion() );
println( " REGRESSAO BETA - Modelos nao Encaixados" );
println( " Erro no regressor da media." );
println( "\t\t Modelo 1: logit(mu_t) = beta0 + beta1*x1 + beta2*x2" );
println( "\t\t\t\t\t log(phi_t) = gama0 + gamal*x1" );
println( "\n\t\t\t Modelo 2: logit(mu_t) = beta0 + beta1*x1 + beta2*x3" );
println( "\t\t\t\t\t log(phi_t) = gama0 + gamal*x1" );
println( "\n Hipotese nula (Ho): Modelo 1 correto." );
println( " TAMANHO DA AMOSTRA: ", NOBS );
println( " NUMERO DE REPLICAS MC:", NREP );
println( " NUMERO DE REPLICAS BOOTSTRAP:", NBOOT );
println( "\t NUM. REPLICAS MC COM CONVERGENCIA: " );
println( "\t\t\t Modelo 1: ", NREP - falha1MC );
println( "\t\t\t Modelo 2: ", NREP - falha2MC );
println( "\t NUM. REPLICAS MC COM CONVERGENCIA - Modelo ampliado: " );
println( "\t\t\t Modelo 1: ", NREP - falha1MCampliado );
println( "\t\t\t Modelo 2: ", NREP - falha2MCampliado );
println( "\t NUM. REPLICAS MC SEM CONVERGENCIA: " );
println( "\t\t\t Modelo 1: ", falha1MC );
println( "\t\t\t Modelo 2: ", falha2MC );
println( "\t NUM. REPLICAS MC SEM CONVERGENCIA - Modelo ampliado: " );
println( "\t\t\t Modelo 1: ", falha1MCampliado );
println( "\t\t\t Modelo 2: ", falha2MCampliado );
println( "\t NUM. REPLICAS BOOTSTRAP COM CONVERGENCIA: " );
println( "\t\t\t Modelo 1: ", NBOOT*NREP - falha1BOOT );
println( "\t\t\t Modelo 2: ", NBOOT*NREP - falha2BOOT );
println( "\t NUM. REPLICAS BOOTSTRAP COM CONVERGENCIA - Modelo ampliado: " );
println( "\t\t\t Modelo 1: ", NBOOT*NREP - falha1BOOTampliado );
println( "\t\t\t Modelo 2: ", NBOOT*NREP - falha2BOOTampliado );
println( "\t NUM. REPLICAS BOOTSTRAP SEM CONVERGENCIA: " );
println( "\t\t\t Modelo 1: ", falha1BOOT );
println( "\t\t\t Modelo 2: ", falha2BOOT );
println( "\t NUM. REPLICAS BOOTSTRAP SEM CONVERGENCIA - Modelo ampliado: " );
println( "\t\t\t Modelo 1: ", falha1BOOTampliado );
println( "\t\t\t Modelo 2: ", falha2BOOTampliado );
println( " VALORES VERDADEIROS DOS PARAMETROS:" );
println( "%r", {" beta0:", " beta1:"}, beta', "%r", {" gama0:", " gamal:"}, gama' );
println( "\n ESTIMATIVAS DOS PARAMETROS: " );
println( "%r", {" beta0:", " beta1:", " gama0:", " gamal:"}, meanr(irl[0:k+m-1][]) );
println( "\n Grau de heterogeneidade da dispersao: ", lambda );
println( "\n TAXA DE FALHA (%) DO TESTE J:" );
println( "%r", {" 1%:", " 5%:", " 10%:"}, "%c", {"RV", "RV boot."},
    100*(1 - (acertoJ/NREP)) ~ 100*(1 - (acertoJb/NREP)) );
println( "\n TAXA DE FALHA (%) DO TESTE MJ:" );
println( "%r", {" 1%:", " 5%:", " 10%:"}, "%c", {"assintot.", "boot."},
    100*(1 - (acertoMJ/NREP)) ~ 100*(1 - (acertoMJb / NREP)) );
println("Proporcao de m=m* do teste MJ:", "%c", {"assintot.", "boot."},
    100*(acertomm./acertoMJ) ~ 100*(acertomm./acertoMJb));
println("TEMPO DE EXECUCAO: ", timespan(dExecTime));
}

EstMV(const ystar, const mX, const mZ){
decl betaols, etaols, muols, varols, gamaols, vtheta, dfunc, converge;
decl l = sizec(mZ);
/* Chute inicial para beta */
if(k > 1)
ols2c(ystar, mX, &betaols); /* Estimativas de MQO */
else if(k == 1)
betaols = meanc(ystar);
/* Quantidades para o chute inicial de gama */
etaols = mX * betaols;

```

## A.2 MODELOS COM DIFERENÇA NOS REGRESSORES DA DISPERSÃO

```

muols = exp(etaols) ./ (1.0 + exp(etaols)); /* inversa da ligação logit */
varols = ( (ystar-etaols)' * (ystar - etaols) .* ((muols.*(1.0-muols)).^2) ) ./
(NOBS - k);
gamaols = meanc( muols .* (1.0-muols) ./ varols ) - 1.0;
gamaols = log(gamaols) | zeros(1-1,1); /* Chute inicial para gama */

vtheta = betaols | gamaols;
MaxControl(50, -1);

s_mCovMU = mX; /* covariáveis do modelo para mu */
s_mCovPHI = mZ; /* covariáveis do modelo para phi */

/* Estimação de máxima verossimilhança */
converge = MaxBFGS(floglik_logit_log, &vtheta, &dfunc, 0, 0);

return vtheta | dfunc | converge;
}

```

## A.2 Modelos com diferença nos regressores da dispersão

```

/*****
*
* PROGRAMA: betareg_nonnested_disp_n20_3.0.ox
*
* DESCRIÇÃO: realiza os testes J (com teste da razão de veros-
* similhanças) e MJ para modelos de regressão beta com disper-
* são variável
*
* MODELOS: (M1) logit(mu_t) = beta0 + betal*x1 + beta2*x2
*           log(phi_t) = gama0 + gamal*x1
*
*           (M2) logit(mu_t) = beta0 + betal*x1 + beta2*x2
*           log(phi_t) = gama0 + gamal*x2
*
* AUTOR: Sadraque Eneas de Figueiredo Lucena
*
* VERSÃO: 3.0
*
* DATA: 21/05/2012
*
* OBS.: Programa desenvolvido com base no programa apresentado
* na dissertação de mestrado de Queiroz (2011).
*
*****/

/* Arquivos cabeçalho */
#include <oxstd.h> /* Biblioteca padrão do OX */
#include <oxprob.h> /* Necessário para gerar ocorrências beta */
#import <maximize> /* Necessário para otimização não-linear */

/* Variáveis globais */
static decl s_vY; /* Amostra aleatória */
static decl s_mX; /* Matriz X */
static decl s_mZ1; /* Matriz Z1 */
static decl s_mZ2; /* Matriz Z2 */
static decl s_vYB; /* Amostra bootstrap */
static decl s_mCovMU; /* Matriz de covariáveis do modelo para mu */
static decl s_mCovPHI; /* Matriz de covariáveis do modelo para phi */

static decl boot = 0;

/* Definição de constantes */
const decl NOBS = 20; /* Tamanho amostral */

```

## A.2 MODELOS COM DIFERENÇA NOS REGRESSORES DA DISPERSÃO

```

const decl NREP = 10000; /* Número de réplicas de Monte Carlo */
const decl NBOOT = 1000; /* Número de amostras bootstrap */
const decl k = 3; /* Número de parâmetros no modelo para mu */
const decl m = 2; /* Número de parâmetros no modelo para phi */
const decl beta = <1.0, 2.2, -3.7>; /* Parâmetros do modelo para mu */
const decl gama = <1.5, 2.3>; /* Parâmetros do modelo para phi */

/* Algumas funções */
EstMV(const ystar, const mX, const mZ); /* EMV do modelo com regressores X e Z */

/* Função de log-verossimilhança do modelo com ligações logit(mu) e log(phi) */
floglik_logit_log(const vtheta, const adFunc, const avScore, const amHess)
{
decl y;
if(boot == 1) /* usar amostra bootstrap */
y = s_vYB;
if(boot == 0) /* usar amostra */
y = s_vY;

decl k = sizec(s_mCovMU); /* número de colunas da matriz de regressores de mu */
decl m = sizec(s_mCovPHI); /* número de colunas da matriz de regressores de phi */
decl eta = s_mCovMU * vtheta[0:(k-1)]; /* usado para obter mu */
decl mu = exp(eta) ./ (1.0 + exp(eta)); /* inversa da ligação logit */
decl delta = s_mCovPHI * vtheta[k:(k+m-1)]; /* usado para obter phi */
decl phi = exp(delta); /* inversa da ligação log */
decl i, funlog = zeros(NOBS, 1);
for(i = 0; i < NOBS; i++) /* logaritmo da densidade */
funlog[i] = log( densbeta(y[i], mu[i] * phi[i], (1 - mu[i]) * phi[i]) );

adFunc[0] = double( sumc( funlog ) ); /* log-verossimilhança */

/* Parâmetros utilizados no cálculo das primeiras derivadas */
decl ystar = log(y ./ (1.0 - y));
decl mustar = polygamma(mu.*phi,0) - polygamma((1.0-mu).*phi,0);
decl ydag = log(1.0 - y);
decl mudag = polygamma((1.0-mu).*phi,0) - polygamma(phi,0);
decl T = diag( mu .* (1-mu) ); /* 1/g'(mu) */
decl F = diag(phi);
decl H = diag(phi); /* 1/h'(phi) */
decl M = diag(mu);

if(avScore){
(avScore[0])[0:(k-1)] = s_mCovMU'*T*F*(ystar-mustar);
(avScore[0])[k:(k+m-1)] = s_mCovPHI'*H*(M*(ystar-mustar)+(ydag-mudag));
}

if( isnan(adFunc[0]) || isdotinf(adFunc[0]) )
return 0;
else
return 1; /* 1 indica sucesso */
}

/* INÍCIO DO PROGRAMA */
main()
{
/* Declaração de variáveis utilizadas no programa */
/* Parâmetros do sistema */
decl dExecTime, i, j, b, falha1MC, falha2MC, falha1MCampliado, falha2MCampliado,
acertoJ, acertoJb, acertoMJ, acertoMJb, falha1BOOT, falha2BOOT, falha1BOOTampliado,
falha2BOOTampliado, acertomm, acertomm;
/* Parâmetros do modelo */
decl eta, mu, delta, phi, ir1, ir2, irlampliado, ir2ampliado;
/* Parâmetros do modelo bootstrap */
decl ir1b, ir2b, irlampliadob, ir2ampliadob;
/* Parâmetros dos testes */
decl estat_testeJ1, estat_testeJ2, estat_testeJ1b, estat_testeJ2b, estatMJ, estatMJb;

```

## A.2 MODELOS COM DIFERENÇA NOS REGRESSORES DA DISPERSÃO

```

/* Parâmetros auxiliares */
decl chi2, ystar, ystarb;

dExecTime = timer(); /* Inicia o relógio */

ranseed("GM"); /* Gerador de números aleatórios */
ranseed({6548,8984}); /* Semente */

/* Inicializações */
s_vY = s_vYB = zeros(NOBS, 1); /* Vetores de amostra e amostra bootstrap */
ir1 = ir2 = zeros((k+m+2), NREP); /* Estimativas de MV */
ir1b = ir2b = zeros((k+m+2), NBOOT); /* Estimativas de MV bootstrap */
falha1MC = falha2MC = falha1BOOT = falha2BOOT = 0; /* Contadores de falha */
falha1MCampliado = falha2MCampliado = 0; /* Contadores de falha */
falha1BOOTampliado = falha2BOOTampliado = 0; /* Contadores de falha */
acertomm = acertommb = acertoJ = acertoJb = acertoMJ = acertoMjb = zeros(3,1); /* Contador
de acertos do teste J */
estat_testeJ1b = estat_testeJ2b = estatMJb = zeros(NBOOT, 1); /* Armazena est. bootstrap */

/* Quantis da qui-quadrado */
chi2 = quanchi(.99,1) | quanchi(.95,1) | quanchi(.90,1);

/* Obtendo matrizes de regressores */
s_mX = 1 ~ (ranu(10,k-1));
/* Replicando para manter o grau de heterogeneidade constante */
for(i = 10; i < NOBS; i += 10){
s_mX = s_mX | s_mX[0:9][];
}
s_mZ1 = s_mX[][0:1];
s_mZ2 = s_mX[][0] ~ s_mX[][2];

/* Valores verdadeiros dos parâmetros */
eta = s_mX * beta';
mu = exp(eta) ./ (1.0 + exp(eta)); /* inversa da ligação logit */
delta = s_mZ1 * gama';
phi = exp(delta); /* inversa da ligação log */

decl lambda = max(phi)/min(phi);

/* Início do laço de Monte Carlo */
for(i = 0; i < NREP; i++){
for(j = 0; j < NOBS; j++){
s_vY[j] = ranbeta(1,1,mu[j]*phi[j], (1.0-mu[j])*phi[j]); /* beta(mu,phi) */

/* Obtendo y* */
ystarb = log( s_vY ./ (1.0-s_vY) ); /* transformação logit */

/* Estimando o modelo m1 - regressores X1 (mu) e Z (phi) */
ir1[][i] = EstMV(ystarb, s_mX, s_mZ1);
if(ir1[k+m+1][i] != MAX_CONV && ir1[k+m+1][i] != MAX_WEAK_CONV){
falha1MC++;
i--;
continue;
}

/* Estimando o modelo m2 - regressor X2 (mu) e Z (phi) */
ir2[][i] = EstMV(ystarb, s_mX, s_mZ2);
if(ir2[k+m+1][i] != MAX_CONV && ir2[k+m+1][i] != MAX_WEAK_CONV){
falha2MC++;
i--;
continue;
}

decl deltab1 = s_mZ1 * ir1[k:(k+m-1)][i];
decl phib1 = exp(deltab1);

```

## A.2 MODELOS COM DIFERENÇA NOS REGRESSORES DA DISPERSÃO

```

decl deltab2 = s_mZ2 * ir2[k:(k+m-1)][i];
decl phib2 = exp(deltab2);

/* Ampliando o modelo 1: incluindo eta do modelo 2 */
irlampliado = EstMV(ystar, s_mX, s_mZ1~phib2);
if(irlampliado[k+m+2] != MAX_CONV && irlampliado[k+m+2] != MAX_WEAK_CONV){
falha1MCampliado++;
i--;
continue;
}

/* Ampliando o modelo 2: incluindo eta do modelo 1 */
ir2ampliado = EstMV(ystar, s_mX, s_mZ2~phib1);
if(ir2ampliado[k+m+2] != MAX_CONV && ir2ampliado[k+m+2] != MAX_WEAK_CONV){
falha2MCampliado++;
i--;
continue;
}

/* Teste J - usando estatística RV */
estat_testeJ1 = 2*(irlampliado[k+m+1]-ir1[k+m][i]); /* modelo 1 */

/* Teste MJ */
estat_testeJ2 = 2*(ir2ampliado[k+m+1]-ir2[k+m][i]); /* modelo 2 */
estatMJ = min(estat_testeJ1, estat_testeJ2);

/* Parâmetros a serem usados no laço bootstrap */
/* Modelo 1 artificial */
decl etab1 = s_mX*ir1[0:(k-1)][i];
decl mub1 = exp(etab1) ./ (1.0 + exp(etab1)); /* inversa da ligação logit */

/* mub1 definido anteriormente */

boot = 1; /* indica que vai entrar no laço bootstrap */

/* Início do laço bootstrap */
for(b = 0; b < NBOOT; b++){

/** Modelo 1 **/
/* y* sob Ho: alpha=0 */
for(j = 0; j < NOBS; j++)
s_vYB[j] = ranbeta(1,1,mub1[j]*phib1[j], (1.0-mub1[j])*phib1[j]); /* beta(mu,phi) */

ystarb = log( s_vYB ./ (1.0-s_vYB) ); /* transformação logit */

/* Estimando o modelo 1 artificial - regressores X (mu) e Z (phi) */
irlb[][b] = EstMV(ystarb, s_mX, s_mZ1);
if(irlb[k+m+1][b] != MAX_CONV && irlb[k+m+1][b] != MAX_WEAK_CONV){
falha1BOOT++; /* contando falhas de convergência */
b--;
continue;
}

/* Ampliando o modelo 1 artificial: incluindo eta do modelo 2 */
irlampliadob = EstMV(ystarb, s_mX, s_mZ1~phib2);
if(irlampliadob[k+m+2] != MAX_CONV && irlampliadob[k+m+2] != MAX_WEAK_CONV){
falha1BOOTampliado++; /* contando falhas de convergência */
b--;
continue;
}

/* Estimando o modelo 2 artificial - regressores X (mu e phi) */
ir2b[][b] = EstMV(ystarb, s_mX, s_mZ2);
if(ir2b[k+m+1][b] != MAX_CONV && ir2b[k+m+1][b] != MAX_WEAK_CONV){
falha2BOOT++; /* contando falhas de convergência */
b--;
}

```

## A.2 MODELOS COM DIFERENÇA NOS REGRESSORES DA DISPERSÃO

```
continue;
}

/* Ampliando o modelo 2 artificial: incluindo eta do modelo 1 */
ir2ampliados = EstMV(ystarb, s_mX, s_mZ2~phib1);
if(ir2ampliados[k+m+2] != MAX_CONV && ir2ampliados[k+m+2] != MAX_WEAK_CONV){
falha2BOOTampliados++; /* contando falhas de convergência */
b--;
continue;
}

/* Estatística do teste J bootstrap - usando estatística escore */
estat_testeJ1b[b] = 2*(irlampliados[k+m+1] - irlb[k+m][b]); /* modelo 1 */

/* Estatística do teste MJ bootstrap */
estat_testeJ2b[b] = 2*(ir2ampliados[k+m+1] - ir2b[k+m][b]); /* modelo 2 */
estatMJb[b] = min(estat_testeJ1b[b], estat_testeJ2b[b]);
} /* Fim do laço bootstrap */

boot = 0; /* Indica que saiu do laço bootstrap */

/* Contagem de acertos do teste J */
if(estat_testeJ1 < chi2[0]) /* 1% sig. */
acertoJ[0]++;
if(estat_testeJ1 < chi2[1]) /* 5% sig. */
acertoJ[1]++;
if(estat_testeJ1 < chi2[2]) /* 10% sig. */
acertoJ[2]++;

/* Contagem de acertos do teste J bootstrap */
if(estat_testeJ1 < quantilec(estat_testeJ1b, .99)) /* 1% sig. */
acertoJb[0]++;
if(estat_testeJ1 < quantilec(estat_testeJ1b, .95)) /* 5% sig. */
acertoJb[1]++;
if(estat_testeJ1 < quantilec(estat_testeJ1b, .90)) /* 10% sig. */
acertoJb[2]++;

/* Contagem de acertos do teste MJ */
if(estatMJ < chi2[0]) /* 1% sig. */
acertoMJ[0]++;
if(estatMJ < chi2[1]) /* 5% sig. */
acertoMJ[1]++;
if(estatMJ < chi2[2]) /* 10% sig. */
acertoMJ[2]++;

/* Contagem de acertos do teste MJ bootstrap */
if(estatMJ < quantilec(estatMJb, .99)) /* 1% sig. */
acertoMJb[0]++;
if(estatMJ < quantilec(estatMJb, .95)) /* 5% sig. */
acertoMJb[1]++;
if(estatMJ < quantilec(estatMJb, .90)) /* 10% sig. */
acertoMJb[2]++;

/* Contagem de m=m* quando Ho nao foi rejeitada */
if(estatMJ < chi2[0] && estatMJ == estat_testeJ1)
acertomm[0]++;
if(estatMJ < chi2[1] && estatMJ == estat_testeJ1)
acertomm[1]++;
if(estatMJ < chi2[2] && estatMJ == estat_testeJ1)
acertomm[2]++;

/* Contagem de m=m* quando Ho nao foi rejeitada via bootstrap */
if(estatMJ < quantilec(estatMJb, .99) && estatMJ == estat_testeJ1)
acertommb[0]++;
if(estatMJ < quantilec(estatMJb, .95) && estatMJ == estat_testeJ1)
acertommb[1]++;
```



### A.3 MODELOS COM DIFERENÇA NOS REGRESSORES DA MÉDIA E DA DISPERSÃO

```
etaols = mX * betaols;
muols = exp(etaols) ./ (1.0 + exp(etaols)); /* inversa da ligação logit */
varols = ( (ystar-etaols)' * (ystar - etaols) .* (muols.*(1.0-muols)).^2 ) ./
(NOBS - k);
gamaols = meanc( muols .* (1.0-muols) ./ varols ) - 1.0;
gamaols = log(gamaols) | zeros(1-1,1); /* Chute inicial para gama */

vtheta = betaols | gamaols;
MaxControl(50, -1);

s_mCovMU = mX; /* covariáveis do modelo para mu */
s_mCovPHI = mZ; /* covariáveis do modelo para phi */

/* Estimação de máxima verossimilhança */
converge = MaxBFGS(floglik_logit_log, &vtheta, &dfunc, 0, 0);

return vtheta | dfunc | converge;
}
```

### A.3 Modelos com diferença nos regressores da média e da dispersão

```
/******
*
* PROGRAMA: betareg_nonnested_media_dispersao_n20_3.0.ox
*
* DESCRIÇÃO: realiza os testes J (com teste da razão de veros-
* similhanças) e MJ para modelos de regressão beta com disper-
* são variável
*
* MODELOS: (M1) logit(mu_t) = beta0 + beta1*x1 + beta2*x2
*           log(phi_t) = gama0 + gamal*x2
*
*           (M2) logit(mu_t) = beta0 + beta1*x1 + beta2*x3
*           log(phi_t) = gama0 + gamal*x3
*
* AUTOR: Sadraque Eneas de Figueiredo Lucena
*
* VERSÃO: 3.0
*
* DATA: 21/05/2012
*
* OBS.: Programa desenvolvido com base no programa apresentado
* na dissertação de mestrado de Queiroz (2011).
*
* *****/

/* Arquivos cabeçalho */
#include <oxstd.h> /* Biblioteca padrão do OX */
#include <oxprob.h> /* Necessário para gerar ocorrências beta */
#import <maximize> /* Necessário para otimização não-linear */

/* Variáveis globais */
static decl s_vY; /* Amostra aleatória */
static decl s_mX1; /* Matriz X1 */
static decl s_mX2; /* Matriz X2 */
static decl s_mZ1; /* Matriz Z1 */
static decl s_mZ2; /* Matriz Z2 */
static decl s_vYB; /* Amostra bootstrap */
static decl s_mCovMU; /* Matriz de covariáveis do modelo para mu */
static decl s_mCovPHI; /* Matriz de covariáveis do modelo para phi */

static decl boot = 0;
```

### A.3 MODELOS COM DIFERENÇA NOS REGRESSORES DA MÉDIA E DA DISPERSÃO

```

/* Definição de constantes */
const decl NOBS = 20; /* Tamanho amostral */
const decl NREP = 10000; /* Número de réplicas de Monte Carlo */
const decl NBOOT = 1000; /* Número de amostras bootstrap */
const decl k = 3; /* Número de parâmetros no modelo para mu */
const decl m = 2; /* Número de parâmetros no modelo para phi */
const decl beta = <1.0, 2.2, -3.7>; /* Parâmetros do modelo para mu */
const decl gama = <1.5, 2.3>; /* Parâmetros do modelo para phi */

/* Algumas funções */
EstMV(const ystar, const mX, const mZ); /* EMV do modelo com regressores X e Z */

/* Função de log-verossimilhança do modelo com ligações logit(mu) e log(phi) */
floglik_logit_log(const vtheta, const adFunc, const avScore, const amHess)
{
decl y;
if(boot == 1) /* usar amostra bootstrap */
y = s_vYB;
if(boot == 0) /* usar amostra */
y = s_vY;

decl k = sizec(s_mCovMU); /* número de colunas da matriz de regressores de mu */
decl m = sizec(s_mCovPHI); /* número de colunas da matriz de regressores de phi */
decl eta = s_mCovMU * vtheta[0:(k-1)]; /* usado para obter mu */
decl mu = exp(eta) ./ (1.0 + exp(eta)); /* inversa da ligação logit */
decl delta = s_mCovPHI * vtheta[k:(k+m-1)]; /* usado para obter phi */
decl phi = exp(delta); /* inversa da ligação log */
decl i, funlog = zeros(NOBS, 1);
for(i = 0; i < NOBS; i++) /* logaritmo da densidade */
funlog[i] = log( densbeta(y[i], mu[i] * phi[i], (1 - mu[i]) * phi[i]) );

adFunc[0] = double( sumc( funlog ) ); /* log-verossimilhança */

/* Parâmetros utilizados no cálculo das primeiras derivadas */
decl ystar = log(y ./ (1.0 - y));
decl mustar = polygamma(mu.*phi,0) - polygamma((1.0-mu).*phi,0);
decl ydag = log(1.0 - y);
decl mudag = polygamma((1.0-mu).*phi,0) - polygamma(phi,0);
decl T = diag( mu .* (1-mu) ); /* 1/g'(mu) */
decl F = diag(phi);
decl H = diag(phi); /* 1/h'(phi) */
decl M = diag(mu);

if(avScore){
(avScore[0])[0:(k-1)] = s_mCovMU' * T * F * (ystar - mustar);
(avScore[0])[k:(k+m-1)] = s_mCovPHI' * H * (M * (ystar - mustar) + (ydag - mudag));
}

if( isnan(adFunc[0]) || isdotinf(adFunc[0]) )
return 0;
else
return 1; /* 1 indica sucesso */
}

/* INÍCIO DO PROGRAMA */
main()
{
/* Declaração de variáveis utilizadas no programa */
/* Parâmetros do sistema */
decl dExecTime, i, j, b, falha1MC, falha2MC, falha1MCampliado, falha2MCampliado,
acertoJ, acertoJb, acertoMJ, acertoMJB, falha1BOOT, falha2BOOT, falha1BOOTampliado,
falha2BOOTampliado, acertomm, acertomm;
/* Parâmetros do modelo */
decl eta, mu, delta, phi, ir1, ir2, ir1ampliado, ir2ampliado;
/* Parâmetros do modelo bootstrap */
decl ir1b, ir2b, ir1ampliadob, ir2ampliadob;

```

### A.3 MODELOS COM DIFERENÇA NOS REGRESSORES DA MÉDIA E DA DISPERSÃO

```

/* Parâmetros dos testes */
decl estat_testeJ1, estat_testeJ2, estat_testeJ1b, estat_testeJ2b, estatMJ, estatMJb;
/* Parâmetros auxiliares */
decl chi2, ystar, ystarb;

dExecTime = timer(); /* Inicia o relógio */

ranseed("GM"); /* Gerador de números aleatórios */
ranseed({6548,8984}); /* Semente */

/* Inicializações */
s_vY = s_vYB = zeros(NOBS, 1); /* Vetores de amostra e amostra bootstrap */
irl = ir2 = zeros((k+m+2), NREP); /* Estimativas de MV */
irlb = ir2b = zeros((k+m+2), NBOOT); /* Estimativas de MV bootstrap */
falha1MC = falha2MC = falha1BOOT = falha2BOOT = 0; /* Contadores de falha */
falha1MCampulado = falha2MCampulado = 0; /* Contadores de falha */
falha1BOOTampulado = falha2BOOTampulado = 0; /* Contadores de falha */
acertomm = acertomb = acertoJ = acertoJb = acertoMJ = acertoMJb = zeros(3,1); /* Contador
de acertos do teste J */
estat_testeJ1b = estat_testeJ2b = estatMJb = zeros(NBOOT, 1); /* Armazena est. bootstrap */

/* Quantis da qui-quadrado */
chi2 = quanchi(.99,2) | quanchi(.95,2) | quanchi(.90,2);

/* Obtendo matrizes de regressores */
s_mX1 = 1 ~ (ranu(10,k-1));
s_mX2 = s_mX1[][0:1] ~ (ranu(10,1));
/* Replicando para manter o grau de heterogeneidade constante */
for(i = 10; i < NOBS; i += 10){
s_mX1 = s_mX1 | s_mX1[0:9][];
s_mX2 = s_mX2 | s_mX2[0:9][];
}
s_mZ1 = s_mX1[][0:1];
s_mZ2 = s_mX2[][0] ~ s_mX2[][2];

/* Valores verdadeiros dos parâmetros */
eta = s_mX1 * beta';
mu = exp(eta) ./ (1.0 + exp(eta)); /* inversa da ligação logit */
delta = s_mZ1 * gama';
phi = exp(delta); /* inversa da ligação log */

decl lambda = max(phi)/min(phi);

/* Início do laço de Monte Carlo */
for(i = 0; i < NREP; i++){
for(j = 0; j < NOBS; j++){
s_vY[j] = ranbeta(1,1,mu[j]*phi[j], (1.0-mu[j])*phi[j]); /* beta(mu,phi) */

/* Obtendo y* */
ystarb = log( s_vY ./ (1.0-s_vY) ); /* transformação logit */

/* Estimando o modelo m1 - regressores X1 (mu) e Z (phi) */
irl[][i] = EstMV(ystarb, s_mX1, s_mZ1);
if(irl[k+m+1][i] != MAX_CONV && ir1[k+m+1][i] != MAX_WEAK_CONV){
falha1MC++;
i--;
continue;
}

/* Estimando o modelo m2 - regressor X2 (mu) e Z (phi) */
ir2[][i] = EstMV(ystarb, s_mX2, s_mZ2);
if(ir2[k+m+1][i] != MAX_CONV && ir2[k+m+1][i] != MAX_WEAK_CONV){
falha2MC++;
i--;
continue;
}
}
}

```

### A.3 MODELOS COM DIFERENÇA NOS REGRESSORES DA MÉDIA E DA DISPERSÃO

```

decl etab1 = s_mX1*ir1[0:(k-1)][i];
decl mub1 = exp(etab1) ./ (1.0 + exp(etab1)); /* inversa da ligação logit */
decl deltab1 = s_mZ1 * ir1[k:(k+m-1)][i];
decl phib1 = exp(deltab1);

decl etab2 = s_mX2*ir2[0:(k-1)][i];
decl mub2 = exp(etab2) ./ (1.0 + exp(etab2)); /* inversa da ligação logit */
decl deltab2 = s_mZ2 * ir2[k:(k+m-1)][i];
decl phib2 = exp(deltab2);

/* Ampliando o modelo 1: incluindo eta do modelo 2 */
irlampliado = EstMV(ystar, s_mX1~mub2, s_mZ1~phib2 );
if(irlampliado[k+m+3] != MAX_CONV && irlampliado[k+m+3] != MAX_WEAK_CONV){
falha1MCampliado++;
i--;
continue;
}

/* Ampliando o modelo 2: incluindo eta do modelo 1 */
ir2ampliado = EstMV(ystar, s_mX2~mub1, s_mZ2~phib1 );
if(ir2ampliado[k+m+3] != MAX_CONV && ir2ampliado[k+m+3] != MAX_WEAK_CONV){
falha2MCampliado++;
i--;
continue;
}

/* Teste J - usando estatística RV */
estat_testeJ1 = 2*(irlampliado[k+m+2]-ir1[k+m][i]); /* modelo 1 */

/* Teste MJ */
estat_testeJ2 = 2*(ir2ampliado[k+m+2]-ir2[k+m][i]); /* modelo 2 */
estatMJ = min(estat_testeJ1, estat_testeJ2);

boot = 1; /* indica que vai entrar no laço bootstrap */

/* Início do laço bootstrap */
for(b = 0; b < NBOOT; b++){

/** Modelo 1 **/
/* y* sob Ho: alpha=0 */
for(j = 0; j < NOBS; j++)
s_vYB[j] = ranbeta(1,1,mub1[j]*phib1[j], (1.0-mub1[j])*phib1[j]); /* beta(mu,phi) */

ystar_b = log( s_vYB ./ (1.0-s_vYB) ); /* transformação logit */

/* Estimando o modelo 1 artificial - regressores X (mu) e Z (phi) */
irlb[][b] = EstMV(ystar_b, s_mX1, s_mZ1);
if(irlb[k+m+1][b] != MAX_CONV && irlb[k+m+1][b] != MAX_WEAK_CONV){
falha1BOOT++; /* contando falhas de convergência */
b--;
continue;
}

/* Ampliando o modelo 1 artificial: incluindo eta do modelo 2 */
irlampliadob = EstMV(ystar_b, s_mX1~mub2, s_mZ1~phib2);
if(irlampliadob[k+m+3] != MAX_CONV && irlampliadob[k+m+3] != MAX_WEAK_CONV){
falha1BOOTampliado++; /* contando falhas de convergência */
b--;
continue;
}

/* Estimando o modelo 2 artificial - regressores X (mu e phi) */
ir2b[][b] = EstMV(ystar_b, s_mX2, s_mZ2);
if(ir2b[k+m+1][b] != MAX_CONV && ir2b[k+m+1][b] != MAX_WEAK_CONV){
falha2BOOT++; /* contando falhas de convergência */
}

```

### A.3 MODELOS COM DIFERENÇA NOS REGRESSORES DA MÉDIA E DA DISPERSÃO

```
b--;
continue;
}

/* Ampliando o modelo 2 artificial: incluindo eta do modelo 1 */
ir2ampliadb = EstMV(ystarb, s_mX2~mub1, s_mZ2~phib1);
if(ir2ampliadb[k+m+3] != MAX_CONV && ir2ampliadb[k+m+3] != MAX_WEAK_CONV){
falha2BOOTampliado++; /* contando falhas de convergência */
b--;
continue;
}

/* Estatística do teste J bootstrap - usando estatística escore */
estat_testeJlb[b] = 2*(irl1ampliadb[k+m+2] - irlb[k+m][b]); /* modelo 1 */

/* Estatística do teste MJ bootstrap */
estat_testeJ2b[b] = 2*(ir2ampliadb[k+m+2] - ir2b[k+m][b]); /* modelo 2 */
estatMJb[b] = min(estat_testeJlb[b], estat_testeJ2b[b]);
} /* Fim do laço bootstrap */

boot = 0; /* Indica que saiu do laço bootstrap */

/* Contagem de acertos do teste J */
if(estat_testeJ1 < chi2[0]) /* 1% sig. */
acertoJ[0]++;
if(estat_testeJ1 < chi2[1]) /* 5% sig. */
acertoJ[1]++;
if(estat_testeJ1 < chi2[2]) /* 10% sig. */
acertoJ[2]++;

/* Contagem de acertos do teste J bootstrap */
if(estat_testeJ1 < quantilec(estat_testeJlb, .99)) /* 1% sig. */
acertoJb[0]++;
if(estat_testeJ1 < quantilec(estat_testeJlb, .95)) /* 5% sig. */
acertoJb[1]++;
if(estat_testeJ1 < quantilec(estat_testeJlb, .90)) /* 10% sig. */
acertoJb[2]++;

/* Contagem de acertos do teste MJ */
if(estatMJ < chi2[0]) /* 1% sig. */
acertoMJ[0]++;
if(estatMJ < chi2[1]) /* 5% sig. */
acertoMJ[1]++;
if(estatMJ < chi2[2]) /* 10% sig. */
acertoMJ[2]++;

/* Contagem de acertos do teste MJ bootstrap */
if(estatMJ < quantilec(estatMJb, .99)) /* 1% sig. */
acertoMJb[0]++;
if(estatMJ < quantilec(estatMJb, .95)) /* 5% sig. */
acertoMJb[1]++;
if(estatMJ < quantilec(estatMJb, .90)) /* 10% sig. */
acertoMJb[2]++;

/* Contagem de m=m* quando Ho nao foi rejeitada */
if(estatMJ < chi2[0] && estatMJ == estat_testeJ1)
acertomm[0]++;
if(estatMJ < chi2[1] && estatMJ == estat_testeJ1)
acertomm[1]++;
if(estatMJ < chi2[2] && estatMJ == estat_testeJ1)
acertomm[2]++;

/* Contagem de m=m* quando Ho nao foi rejeitada via bootstrap */
if(estatMJ < quantilec(estatMJb, .99) && estatMJ == estat_testeJ1)
acertomb[0]++;
if(estatMJ < quantilec(estatMJb, .95) && estatMJ == estat_testeJ1)
```

### A.3 MODELOS COM DIFERENÇA NOS REGRESSORES DA MÉDIA E DA DISPERSÃO

```

    acertommb[1]++;
if(estatMJ < quantilec(estatMJb, .90) && estatMJ == estat_testeJ1)
    acertommb[2]++;
} /* Fim do laço de Monte Carlo */

/* Impressão de resultados */
println( "\n\n PROGRAMA OX: ", oxfilename(0) );
println( " VERSAO OX: ", oxversion() );
println( " REGRESSAO BETA - Modelos nao Encaixados" );
println( " Erro no regressor da media e da dispersao." );
println( "\t\t Modelo 1: logit(mu_t) = beta0 + betal*x1 + beta2*x2" );
println( "\t\t\t\t\t log(phi_t) = gama0 + gamal*x1 + gama2*x2" );
println( "\n\t\t Modelo 2: logit(mu_t) = beta0 + betal*x1 + beta2*x3" );
println( "\t\t\t\t\t log(phi_t) = gama0 + gamal*x1 + gama2*x3" );
println( "\n Hipotese nula (Ho): Modelo 1 correto." );
println( " TAMANHO DA AMOSTRA: ", NOBS );
println( " NUMERO DE REPLICAS MC:", NREP );
println( " NUMERO DE REPLICAS BOOTSTRAP:", NBOOT );
println( "\t NUM. REPLICAS MC COM CONVERGENCIA: " );
println( "\t\t Modelo 1: ", NREP - falha1MC );
println( "\t\t Modelo 2: ", NREP - falha2MC );
println( "\t NUM. REPLICAS MC COM CONVERGENCIA - Modelo ampliado: " );
println( "\t\t Modelo 1: ", NREP - falha1MCampliado );
println( "\t\t Modelo 2: ", NREP - falha2MCampliado );
println( "\t NUM. REPLICAS MC SEM CONVERGENCIA: " );
println( "\t\t Modelo 1: ", falha1MC );
println( "\t\t Modelo 2: ", falha2MC );
println( "\t NUM. REPLICAS MC SEM CONVERGENCIA - Modelo ampliado: " );
println( "\t\t Modelo 1: ", falha1MCampliado );
println( "\t\t Modelo 2: ", falha2MCampliado );
println( "\t NUM. REPLICAS BOOTSTRAP COM CONVERGENCIA: " );
println( "\t\t Modelo 1: ", NBOOT*NREP - falha1BOOT );
println( "\t\t Modelo 2: ", NBOOT*NREP - falha2BOOT );
println( "\t NUM. REPLICAS BOOTSTRAP COM CONVERGENCIA - Modelo ampliado: " );
println( "\t\t Modelo 1: ", NBOOT*NREP - falha1BOOTampliado );
println( "\t\t Modelo 2: ", NBOOT*NREP - falha2BOOTampliado );
println( "\t NUM. REPLICAS BOOTSTRAP SEM CONVERGENCIA: " );
println( "\t\t Modelo 1: ", falha1BOOT );
println( "\t\t Modelo 2: ", falha2BOOT );
println( "\t NUM. REPLICAS BOOTSTRAP SEM CONVERGENCIA - Modelo ampliado: " );
println( "\t\t Modelo 1: ", falha1BOOTampliado );
println( "\t\t Modelo 2: ", falha2BOOTampliado );
println( " VALORES VERDADEIROS DOS PARAMETROS:" );
println( "%r", {" beta0:", " betal:"}, beta', "%r", {" gama0:", " gamal:"}, gama' );
println( "\n ESTIMATIVAS DOS PARAMETROS: " );
println( "%r", {" beta0:", " betal:", " gama0:", " gamal:"}, meanr(irl[0:k+m-1][]) );
println( "\n Grau de heterogeneidade da dispersao: ", lambda );
println( "\n TAXA DE FALHA (%) DO TESTE J:" );
println( "%r", {" 1%: ", " 5%: ", " 10%: ", "%c", {"RV", "RV boot."},
    100*(1 - (acertoJ/NREP)) ~ 100*(1 - (acertoJb/NREP)) );
println( "\n TAXA DE FALHA (%) DO TESTE MJ:" );
println( "%r", {" 1%: ", " 5%: ", " 10%: ", "%c", {"assintot.", "boot."},
    100*(1 - (acertoMJ/NREP)) ~ 100*(1 - (acertoMJb / NREP)) );
println("Proporcao de m=m* do teste MJ:", "%c", {"assintot.", "boot."},
    100*(acertomm./acertoMJ) ~ 100*(acertommb./acertoMJb));
println("TEMPO DE EXECUCAO: ", timespan(dExecTime));
}

EstMV(const ystar, const mX, const mZ){
decl betaols, etaols, muols, varols, gamaols, vtheta, dfunc, converge;
decl l = sizec(mZ);
/* Chute inicial para beta */
if(k > 1)
ols2c(ystar, mX, &betaols); /* Estimativas de MQO */
else if(k == 1)
betaols = meanc(ystar);

```

## A.4 MODELOS COM DIFERENÇA NA FUNÇÃO DE LIGAÇÃO DA MÉDIA

```
/* Quantidades para o chute inicial de gama */
etaols = mX * betaols;
muols = exp(etaols) ./ (1.0 + exp(etaols)); /* inversa da ligação logit */
varols = ( (ystar-etaols)' * (ystar - etaols) .* ((muols.*(1.0-muols)).^2) ) ./
(NOBS - k);
gamaols = meanc( muols .* (1.0-muols) ./ varols ) - 1.0;
gamaols = log(gamaols) | zeros(1-1,1); /* Chute inicial para gama */

vtheta = betaols | gamaols;
MaxControl(50, -1);

s_mCovMU = mX; /* covariáveis do modelo para mu */
s_mCovPHI = mZ; /* covariáveis do modelo para phi */

/* Estimação de máxima verossimilhança */
converge = MaxBFGS(floglik_logit_log, &vtheta, &dfunc, 0, 0);

return vtheta | dfunc | converge;
}
```

## A.4 Modelos com diferença na função de ligação da média

```
/******
*
* PROGRAMA: betareg_nonnested_linkmedia_n20_4.0.ox
*
* DESCRIÇÃO: realiza os testes J (com teste da razão de veros-
* similhanças) e MJ para modelos de regressão beta com disper-
* são variável
*
* MODELOS: (M1) log-log(mu_t) = beta0 + betal*x1 + beta2*x2
*          log(phi_t) = gama0 + gamal*x1
*
*          (M2) logit(mu_t) = beta0 + betal*x1 + beta2*x2
*          log(phi_t) = gama0 + gamal*x1
*
* AUTOR: Sadraque Eneas de Figueiredo Lucena
*
* VERSÃO: 2.0
*
* DATA: 05/06/2012
*
* OBS.: Programa desenvolvido com base no programa apresentado
* na dissertação de mestrado de Queiroz (2011).
*
* *****/

/* Arquivos cabeçalho */
#include <oxstd.h> /* Biblioteca padrão do OX */
#include <oxprob.h> /* Necessário para gerar ocorrências beta */
#import <maximize> /* Necessário para otimização não-linear */

/* Variáveis globais */
static decl s_vY; /* Amostra aleatória */
static decl s_mX; /* Matriz X */
static decl s_mZ; /* Matriz Z */
static decl s_vYB; /* Amostra bootstrap */
static decl s_mCovMU; /* Matriz de covariáveis do modelo para mu */
static decl s_mCovPHI; /* Matriz de covariáveis do modelo para phi */

static decl boot = 0;

/* Definição de constantes */
```

## A.4 MODELOS COM DIFERENÇA NA FUNÇÃO DE LIGAÇÃO DA MÉDIA

```

const decl NOBS = 20; /* Tamanho amostral */
const decl NREP = 10000; /* Número de réplicas de Monte Carlo */
const decl NBOOT = 1000; /* Número de amostras bootstrap */
const decl k = 3; /* Número de parâmetros no modelo para mu */
const decl m = 2; /* Número de parâmetros no modelo para phi */
const decl beta = <1.0, 1.6, -2.1>; /* Parâmetros do modelo para mu */
const decl gama = <1.2, 1.7>; /* Parâmetros do modelo para phi */

/* Algumas funções */
EstMV_logit(const ystar, const mX, const mZ); /* EMV do modelo com regressores X e Z */
EstMV_loglog(const ystar, const mX, const mZ); /* EMV do modelo com ligação log-log */

/* Função de log-verossimilhança do modelo com ligações logit(mu) e log(phi) */
floglik_logit_log(const vtheta, const adFunc, const avScore, const amHess)
{
  decl y;
  if(boot == 1) /* usar amostra bootstrap */
    y = s_vYB;
  if(boot == 0) /* usar amostra */
    y = s_vY;

  decl k = sizec(s_mCovMU); /* número de colunas da matriz de regressores de mu */
  decl m = sizec(s_mCovPHI); /* número de colunas da matriz de regressores de phi */
  decl eta = s_mCovMU * vtheta[0:(k-1)]; /* usado para obter mu */
  decl mu = exp(eta) ./ (1.0 + exp(eta)); /* inversa da ligação logit */
  decl delta = s_mCovPHI * vtheta[k:(k+m-1)]; /* usado para obter phi */
  decl phi = exp(delta); /* inversa da ligação log */
  decl i, funlog = zeros(NOBS, 1);
  for(i = 0; i < NOBS; i++) /* logaritmo da densidade */
    funlog[i] = log( densbeta(y[i], mu[i] * phi[i], (1 - mu[i]) * phi[i]) );

  adFunc[0] = double( sumc( funlog ) ); /* log-verossimilhança */

  /* Parâmetros utilizados no cálculo das primeiras derivadas */
  decl ystar = log(y ./ (1.0 - y));
  decl mustar = polygamma(mu.*phi,0) - polygamma((1.0-mu).*phi,0);
  decl ydag = log(1.0 - y);
  decl mudag = polygamma((1.0-mu).*phi,0) - polygamma(phi,0);
  decl T = diag( mu .* (1-mu) ); /* 1/g'(mu) */
  decl F = diag(phi);
  decl H = diag(phi); /* 1/h'(phi) */
  decl M = diag(mu);

  if(avScore){
    (avScore[0])[0:(k-1)] = s_mCovMU' * T * F * (ystar - mustar);
    (avScore[0])[k:(k+m-1)] = s_mCovPHI' * H * (M * (ystar - mustar) + (ydag - mudag));
  }

  if( isnan(adFunc[0]) || isdotinf(adFunc[0]) )
    return 0;
  else
    return 1; /* 1 indica sucesso */
}

/* Função de log-verossimilhança do modelo com ligações log-log(mu) e log(phi) */
floglik_loglog_log(const vtheta, const adFunc, const avScore, const amHess)
{
  decl y;
  if(boot == 1) /* usar amostra bootstrap */
    y = s_vYB;
  if(boot == 0) /* usar amostra */
    y = s_vY;

  decl k = sizec(s_mCovMU); /* número de colunas da matriz de regressores de mu */
  decl m = sizec(s_mCovPHI); /* número de colunas da matriz de regressores de phi */
  decl eta = s_mCovMU * vtheta[0:(k-1)]; /* usado para obter mu */

```

#### A.4 MODELOS COM DIFERENÇA NA FUNÇÃO DE LIGAÇÃO DA MÉDIA

```

decl mu = exp( -exp( -eta ) ); /* inversa da ligação log-log */
decl delta = s_mCovPHI * vtheta[k:(k+m-1)]; /* usado para obter phi */
decl phi = exp(delta); /* inversa da ligação log */
decl i, funlog = zeros(NOBS, 1);
for(i = 0; i < NOBS; i++) /* logaritmo da densidade */
funlog[i] = log( densbeta(y[i], mu[i] * phi[i], (1 - mu[i]) * phi[i]) );

adFunc[0] = double( sumc( funlog ) ); /* log-verossimilhança */

/* Parâmetros utilizados no cálculo das primeiras derivadas */
decl ystar = log(y ./ (1.0 - y));
decl mustar = polygamma(mu.*phi,0) - polygamma((1.0-mu).*phi,0);
decl ydag = log(1.0 - y);
decl mudag = polygamma((1.0-mu).*phi,0) - polygamma(phi,0);
decl T = diag( -mu .* log(mu) ); /* 1/g'(mu) */
decl F = diag(phi);
decl H = diag(phi); /* 1/h'(phi) */
decl M = diag(mu);

if(avScore){
(avScore[0])[0:(k-1)] = s_mCovMU' * T * F * (ystar - mustar);
(avScore[0])[k:(k+m-1)] = s_mCovPHI' * H * (M * (ystar - mustar) + (ydag - mudag));
}

if( isnan(adFunc[0]) || isdotinf(adFunc[0]) )
return 0;
else
return 1; /* 1 indica sucesso */
}

/* INÍCIO DO PROGRAMA */
main()
{
/* Declaração de variáveis utilizadas no programa */
/* Parâmetros do sistema */
decl dExecTime, i, j, b, falha1MC, falha2MC, falha1MCampliado, falha2MCampliado,
acertoJ, acertoJb, acertoMJ, acertoMJb, falha1BOOT, falha2BOOT, falha1BOOTampliado,
falha2BOOTampliado, acertomm, acertomm;
/* Parâmetros do modelo */
decl eta, mu, delta, phi, irl, ir2, irlampliado, ir2ampliado;
/* Parâmetros do modelo bootstrap */
decl irlb, ir2b, irlampliadob, ir2ampliadob;
/* Parâmetros dos testes */
decl estat_testeJ1, estat_testeJ2, estat_testeJ1b, estat_testeJ2b, estatMJ, estatMJb;
/* Parâmetros auxiliares */
decl chi2, ystar, ystarb;

dExecTime = timer(); /* Inicia o relógio */

ranseed("GM"); /* Gerador de números aleatórios */
ranseed({6548,8984}); /* Semente */

/* Inicializações */
s_vY = s_vYB = zeros(NOBS, 1); /* Vetores de amostra e amostra bootstrap */
irl = ir2 = zeros((k+m+2), NREP); /* Estimativas de MV */
irlb = ir2b = zeros((k+m+2), NBOOT); /* Estimativas de MV bootstrap */
falha1MC = falha2MC = falha1BOOT = falha2BOOT = 0; /* Contadores de falha */
falha1MCampliado = falha2MCampliado = 0; /* Contadores de falha */
falha1BOOTampliado = falha2BOOTampliado = 0; /* Contadores de falha */
acertomm = acertomm = acertoJ = acertoJb = acertoMJ = acertoMJb = zeros(3,1); /* Contador
de acertos do teste J */
estat_testeJ1b = estat_testeJ2b = estatMJb = zeros(NBOOT, 1); /* Armazena est. bootstrap */
estat_testeJ1 = estat_testeJ2 = estatMJ = zeros(NREP, 1); /* Armazena estatísticas */

/* Quantis da qui-quadrado */
chi2 = quanchi(.99,1) | quanchi(.95,1) | quanchi(.90,1);

```

#### A.4 MODELOS COM DIFERENÇA NA FUNÇÃO DE LIGAÇÃO DA MÉDIA

```

/* Obtendo matrizes de regressores */
s_mX = 1 ~ (ranu(10,k-1));
/* Replicando para manter o grau de heterogeneidade constante */
for(i = 10; i < NOBS; i += 10){
s_mX = s_mX | s_mX[0:9][];
}
s_mZ = s_mX[][0:1];

/* Valores verdadeiros dos parâmetros */
eta = s_mX * beta';
mu = exp( -exp( -eta ) ); /* inversa da ligação log-log */
delta = s_mZ * gama';
phi = exp(delta); /* inversa da ligação log */

decl lambda = max(phi)/min(phi);

/* Início do laço de Monte Carlo */
for(i = 0; i < NREP; i++){
for(j = 0; j < NOBS; j++){
s_vY[j] = ranbeta(1,1,mu[j]*phi[j], (1.0-mu[j])*phi[j]); /* beta(mu,phi) */

/* Obtendo y* */
ystar = -log( -log(s_vY) ); /* transformação log-log */

/* Estimando o modelo m1 - ligação log-log(mu) */
irl[][i] = EstMV_loglog(ystar, s_mX, s_mZ);
if(irl[k+m+1][i] != MAX_CONV && irl[k+m+1][i] != MAX_WEAK_CONV){
falha1MC++;
i--;
continue;
}

/* Estimando o modelo m2 - ligação logit(mu) */
ir2[][i] = EstMV_logit(ystar, s_mX, s_mZ);
if(ir2[k+m+1][i] != MAX_CONV && ir2[k+m+1][i] != MAX_WEAK_CONV){
falha2MC++;
i--;
continue;
}

decl etab1 = s_mX*irl[0:(k-1)][i];
decl mub1 = exp( -exp( -etab1 ) ); /* inversa da ligação log-log */
/* mub1 será também utilizado para gerar amostras do modelo 1 artificial */

decl etab2 = s_mX*ir2[0:(k-1)][i];
decl mub2 = exp(etab2) ./ (1.0 + exp(etab2)); /* inversa da ligação logit */

/* Ampliando o modelo 1: incluindo eta do modelo 2 */
irlampliado = EstMV_loglog(ystar, s_mX~mub2, s_mZ );
if(irlampliado[k+m+2] != MAX_CONV && irlampliado[k+m+2] != MAX_WEAK_CONV){
falha1MCampliado++;
i--;
continue;
}

/* Ampliando o modelo 2: incluindo eta do modelo 1 */
ir2ampliado = EstMV_logit(ystar, s_mX~mub1, s_mZ );
if(ir2ampliado[k+m+2] != MAX_CONV && ir2ampliado[k+m+2] != MAX_WEAK_CONV){
falha2MCampliado++;
i--;
continue;
}

/* Teste J - usando estatística RV */
estat_testeJ1[i] = 2*(irlampliado[k+m+1]-irl[k+m][i]); /* modelo 1 */

```

#### A.4 MODELOS COM DIFERENÇA NA FUNÇÃO DE LIGAÇÃO DA MÉDIA

```

/* Teste MJ */
estat_testeJ2[i] = 2*(ir2ampliado[k+m+1]-ir2[k+m][i]); /* modelo 2 */
estatMJ[i] = min(estat_testeJ1[i], estat_testeJ2[i]);

/* Parâmetros a serem usados no laço bootstrap */
/* Modelo 1 artificial */
decl deltab1 = s_mZ * ir1[k:(k+m-1)][i];
decl phib1 = exp(deltab1);
/* mub1 definido anteriormente */

boot = 1; /* indica que vai entrar no laço bootstrap */

/* Início do laço bootstrap */
for(b = 0; b < NBOOT; b++){

  /** Modelo 1 ***/
  /* y* sob Ho: alpha=0 */
  for(j = 0; j < NOBS; j++)
    s_vYB[j] = ranbeta(1,1,mub1[j]*phib1[j], (1.0-mub1[j])*phib1[j]); /* beta(mu,phi) */

  ystarb = -log( -log( s_vYB ) ); /* transformação log-log */

  /* Estimando o modelo 1 artificial - logação log-log(mu) */
  irlb[][b] = EstMV_loglog(ystarb, s_mX, s_mZ);
  if(irlb[k+m+1][b] != MAX_CONV && irlb[k+m+1][b] != MAX_WEAK_CONV){
    falha1BOOT++; /* contando falhas de convergência */
    b--;
    continue;
  }

  /* Ampliando o modelo 1 artificial: incluindo eta do modelo 2 */
  irlampliadob = EstMV_loglog(ystarb, s_mX~mub2, s_mZ);
  if(irlampliadob[k+m+2] != MAX_CONV && irlampliadob[k+m+2] != MAX_WEAK_CONV){
    falha1BOOTampliado++; /* contando falhas de convergência */
    b--;
    continue;
  }

  /* Estimando o modelo 2 artificial - ligação logit(mu) */
  ir2b[][b] = EstMV_logit(ystarb, s_mX, s_mZ);
  if(ir2b[k+m+1][b] != MAX_CONV && ir2b[k+m+1][b] != MAX_WEAK_CONV){
    falha2BOOT++; /* contando falhas de convergência */
    b--;
    continue;
  }

  /* Ampliando o modelo 2 artificial: incluindo eta do modelo 1 */
  ir2ampliadob = EstMV_logit(ystarb, s_mX~mub1, s_mZ);
  if(ir2ampliadob[k+m+2] != MAX_CONV && ir2ampliadob[k+m+2] != MAX_WEAK_CONV){
    falha2BOOTampliado++; /* contando falhas de convergência */
    b--;
    continue;
  }

  /* Estatística do teste J bootstrap - usando estatística escore */
  estat_testeJ1b[b] = 2*(irlampliadob[k+m+1] - irlb[k+m][b]); /* modelo 1 */

  /* Estatística do teste MJ bootstrap */
  estat_testeJ2b[b] = 2*(ir2ampliadob[k+m+1] - ir2b[k+m][b]); /* modelo 2 */
  estatMJb[b] = min(estat_testeJ1b[b], estat_testeJ2b[b]);
} /* Fim do laço bootstrap */

boot = 0; /* Indica que saiu do laço bootstrap */

/* Contagem de acertos do teste J */

```



## A.4 MODELOS COM DIFERENÇA NA FUNÇÃO DE LIGAÇÃO DA MÉDIA

```

println( "\t\t Modelo 2: ", falha2MCampliado );
println( "\t NUM. REPLICAS BOOTSTRAP SEM CONVERGENCIA: " );
println( "\t\t Modelo 1: ", falha1BOOT );
println( "\t\t Modelo 2: ", falha2BOOT );
println( "\t NUM. REPLICAS BOOTSTRAP SEM CONVERGENCIA - Modelo ampliado: " );
println( "\t\t Modelo 1: ", falha1BOOTampliado );
println( "\t\t Modelo 2: ", falha2BOOTampliado );
println( " VALORES VERDADEIROS DOS PARAMETROS:" );
println( "%r", {" beta0:", " betal:"}, beta', "%r", {" gama0:", " gamal:"}, gama' );
println( "\n ESTIMATIVAS DOS PARAMETROS: " );
println( "%r", {" beta0:", " betal:", " gama0:", " gamal:"}, meanr(irl[0:k+m-1][]) );
println( "\n Grau de heterogeneidade da dispersao: ", lambda );
println( "\n Momentos: ", "%c", {"Estat. J", "Estat. MJ"}, "%r",
{" n:", " media:", " erro padrao:", " assimetria:", " curtose:"},
moments(estat_testeJ1) ~ moments(estatMJ));
println( "\n Quantis: ", "%c", {"Estat. J", "Estat. MJ"}, "%r", {" 90%:", " 95%:", " 99%:"},
(quantilec(estat_testeJ1, .90) | quantilec(estat_testeJ1, .95) | quantilec(estat_testeJ1, .99))
~ (quantilec(estatMJ, .90) | quantilec(estatMJ, .95) | quantilec(estatMJ, .99)));
println( "\n TAXA DE FALHA (%) DO TESTE J:" );
println( "%r", {" 1%:", " 5%:", " 10%:"}, "%c", {"Assintot.", "boot."},
100*(1 - (acertoJ/NREP)) ~ 100*(1 - (acertoJb/NREP)) );
println( "\n TAXA DE FALHA (%) DO TESTE MJ:" );
println( "%r", {" 1%:", " 5%:", " 10%:"}, "%c", {"Assintot.", "boot."},
100*(1 - (acertoMJ/NREP)) ~ 100*(1 - (acertoMJb/NREP)) );
println( "Proporcao de m=m* do teste MJ:", "%c", {"assintot.", "boot."},
100*(acertomm./acertoMJ) ~ 100*(acertommb./acertoMJb));
println( "TEMPO DE EXECUCAO: ", timespan(dExecTime));
}

```

```

EstMV_logit(const ystar, const mX, const mZ){
decl betaols, etaols, muols, varols, gamaols, vtheta, dfunc, converge;
decl l = sizec(mZ);
/* Chute inicial para beta */
if(k > 1)
ols2c(ystar, mX, &betaols); /* Estimativas de MQO */
else if(k == 1)
betaols = meanc(ystar);
/* Quantidades para o chute inicial de gama */
etaols = mX * betaols;
muols = exp(etaols) ./ (1.0 + exp(etaols)); /* inversa da ligação logit */
varols = ( ((ystar-etaols)' * (ystar - etaols)) .* ((muols.*(1.0-muols)).^2) ) ./
(NOBS - k);
gamaols = meanc( muols .* (1.0-muols) ./ varols ) - 1.0;
gamaols = log(gamaols) | zeros(l-1,1); /* Chute inicial para gama */

vtheta = betaols | gamaols;
MaxControl(50, -1);

s_mCovMU = mX; /* covariáveis do modelo para mu */
s_mCovPHI = mZ; /* covariáveis do modelo para phi */

/* Estimação de máxima verossimilhança */
converge = MaxBFGS(floglik_logit_log, &vtheta, &dfunc, 0, 0);

return vtheta | dfunc | converge;
}

```

```

EstMV_loglog(const ystar, const mX, const mZ){
decl betaols, etaols, muols, varols, gamaols, vtheta, dfunc, converge;
decl l = sizec(mZ);
/* Chute inicial para beta */
if(k > 1)
ols2c(ystar, mX, &betaols); /* Estimativas de MQO */
else if(k == 1)
betaols = meanc(ystar);
/* Quantidades para o chute inicial de gama */

```

## A.5 MODELOS COM DIFERENÇA NA FUNÇÃO DE LIGAÇÃO DA DISPERSÃO

```
etaols = mX * betaols;
muols = exp( -exp( -etaols ) ); /* inversa da ligação log-log */
varols = ( (ystar-etaols)' * (ystar - etaols)) .* ((muols .* log(muols)).^2) ) ./
(NOBS - k); /* forma para a ligação log-log */
gamaols = meanc( muols .* (1.0-muols) ./ varols ) - 1.0;
gamaols = log(gamaols) | zeros(1-1,1); /* Chute inicial para gama */

vtheta = betaols | gamaols;
MaxControl(50, -1);

s_mCovMU = mX; /* covariáveis do modelo para mu */
s_mCovPHI = mZ; /* covariáveis do modelo para phi */

/* Estimação de máxima verossimilhança */
converge = MaxBFGS(floglik_loglog_log, &vtheta, &dfunc, 0, 0);

return vtheta | dfunc | converge;
}
```

## A.5 Modelos com diferença na função de ligação da dispersão

```
/******
*
* PROGRAMA: betareg_nonnested_linkdispersao_n20_v4.0.ox
*
* DESCRIÇÃO: realiza os testes J (com teste da razão de veros-
* similhanças) e MJ para modelos de regressão beta com disper-
* são variável
*
* MODELOS: (M1) log-log(mu_t) = beta0 + beta1*x1 + beta2*x2
*          raiz(phi_t) = gama0 + gama1*x1
*
*          (M2) logit(mu_t) = beta0 + beta1*x1 + beta2*x2
*          log(phi_t) = gama0 + gama1*x1
*
* AUTOR: Sadraque Eneas de Figueiredo Lucena
*
* VERSÃO: 4.0
*
* DATA: 02/07/2012
*
* OBS.: Programa desenvolvido com base no programa apresentado
* na dissertação de mestrado de Queiroz (2011).
*
* *****/

/* Arquivos cabeçalho */
#include <oxstd.h> /* Biblioteca padrão do OX */
#include <oxprob.h> /* Necessário para gerar ocorrências beta */
#import <maximize> /* Necessário para otimização não-linear */

/* Variáveis globais */
static decl s_vY; /* Amostra aleatória */
static decl s_mX; /* Matriz X */
static decl s_mZ; /* Matriz Z */
static decl s_vYB; /* Amostra bootstrap */
static decl s_mCovMU; /* Matriz de covariáveis do modelo para mu */
static decl s_mCovPHI; /* Matriz de covariáveis do modelo para phi */

static decl boot = 0;

/* Definição de constantes */
const decl NOBS = 20; /* Tamanho amostral */
```

## A.5 MODELOS COM DIFERENÇA NA FUNÇÃO DE LIGAÇÃO DA DISPERSÃO

```

const decl NREP = 10000; /* Número de réplicas de Monte Carlo */
const decl NBOOT = 1000; /* Número de amostras bootstrap */
const decl k = 3; /* Número de parâmetros no modelo para mu */
const decl m = 2; /* Número de parâmetros no modelo para phi */
const decl beta = <1.0, 2.2, -3.7>; /* Parâmetros do modelo para mu */
const decl gama = <1.5, 2.3>; /* Parâmetros do modelo para phi */

/* Algumas funções */
EstMV_log(const ystar, const mX, const mZ); /* EMV do modelo com regressores X e Z */
EstMV_raiz(const ystar, const mX, const mZ); /* EMV do modelo com ligação log-log */

/* Função de log-verossimilhança do modelo com ligações logit(mu) e log(phi) */
floglik_logit_log(const vtheta, const adFunc, const avScore, const amHess)
{
decl y;
if(boot == 1) /* usar amostra bootstrap */
y = s_vYB;
if(boot == 0) /* usar amostra */
y = s_vY;

decl k = sizec(s_mCovMU); /* número de colunas da matriz de regressores de mu */
decl m = sizec(s_mCovPHI); /* número de colunas da matriz de regressores de phi */
decl eta = s_mCovMU * vtheta[0:(k-1)]; /* usado para obter mu */
decl mu = exp(eta) ./ (1.0 + exp(eta)); /* inversa da ligação logit */
decl delta = s_mCovPHI * vtheta[k:(k+m-1)]; /* usado para obter phi */
decl phi = exp(delta); /* inversa da ligação log */
decl i, funlog = zeros(NOBS, 1);
for(i = 0; i < NOBS; i++) /* logaritmo da densidade */
funlog[i] = log( densbeta(y[i], mu[i] * phi[i], (1 - mu[i]) * phi[i]) );

adFunc[0] = double( sumc( funlog ) ); /* log-verossimilhança */

/* Parâmetros utilizados no cálculo das primeiras derivadas */
decl ystar = log(y ./ (1.0 - y));
decl mustar = polygamma(mu.*phi,0) - polygamma((1.0-mu).*phi,0);
decl ydag = log(1.0 - y);
decl mudag = polygamma((1.0-mu).*phi,0) - polygamma(phi,0);
decl T = diag( mu .* (1-mu) ); /* 1/g'(mu) */
decl F = diag(phi);
decl H = diag(phi); /* 1/h'(phi) */
decl M = diag(mu);

if(avScore){
(avScore[0])[0:(k-1)] = s_mCovMU'*T*F*(ystar-mustar);
(avScore[0])[k:(k+m-1)] = s_mCovPHI'*H*(M*(ystar-mustar)+(ydag-mudag));
}

if( isnan(adFunc[0]) || isdotinf(adFunc[0]) )
return 0;
else
return 1; /* 1 indica sucesso */
}

/* Função de log-verossimilhança do modelo com ligações logit(mu) e raiz(phi) */
floglik_logit_raiz(const vtheta, const adFunc, const avScore, const amHess)
{
decl y;
if(boot == 1) /* usar amostra bootstrap */
y = s_vYB;
if(boot == 0) /* usar amostra */
y = s_vY;

decl k = sizec(s_mCovMU); /* número de colunas da matriz de regressores de mu */
decl m = sizec(s_mCovPHI); /* número de colunas da matriz de regressores de phi */
decl eta = s_mCovMU * vtheta[0:(k-1)]; /* usado para obter mu */
decl mu = exp(eta) ./ (1.0 + exp(eta)); /* inversa da ligação logit */

```

## A.5 MODELOS COM DIFERENÇA NA FUNÇÃO DE LIGAÇÃO DA DISPERSÃO

```

decl delta = s_mCovPHI * vtheta[k:(k+m-1)]; /* usado para obter phi */
decl phi = (delta .^ 2); /* inversa da ligação raiz */
decl i, funlog = zeros(NOBS, 1);
for(i = 0; i < NOBS; i++) /* logaritmo da densidade */
funlog[i] = log( densbeta(y[i], mu[i] * phi[i], (1 - mu[i]) * phi[i]) );

adFunc[0] = double( sumc( funlog ) ); /* log-verossimilhança */

/* Parâmetros utilizados no cálculo das primeiras derivadas */
decl ystar = log(y ./ (1.0 - y));
decl mustar = polygamma(mu.*phi,0) - polygamma((1.0-mu).*phi,0);
decl ydag = log(1.0 - y);
decl mudag = polygamma((1.0-mu).*phi,0) - polygamma(phi,0);
decl T = diag( mu .* (1-mu) ); /* 1/g' (mu) */
decl F = diag(phi);
decl H = diag( 2 * (phi .^ .5) ); /* 1/h' (phi) */
decl M = diag(mu);

if(avScore){
(avScore[0])[0:(k-1)] = s_mCovMU' * T * F * (ystar - mustar);
(avScore[0])[k:(k+m-1)] = s_mCovPHI' * H * (M * (ystar - mustar) + (ydag - mudag));
}

if( isnan(adFunc[0]) || isdotinf(adFunc[0]) )
return 0;
else
return 1; /* 1 indica sucesso */
}

/* INÍCIO DO PROGRAMA */
main()
{
/* Declaração de variáveis utilizadas no programa */
/* Parâmetros do sistema */
decl dExecTime, i, j, b, falha1MC, falha2MC, falha1MCampliado, falha2MCampliado,
acertoJ, acertoJb, acertoMJ, acertoMJb, falha1BOOT, falha2BOOT, falha1BOOTampliado,
falha2BOOTampliado, acertomm, acertommmb;
/* Parâmetros do modelo */
decl eta, mu, delta, phi, irl, ir2, irlampliado, ir2ampliado;
/* Parâmetros do modelo bootstrap */
decl irlb, ir2b, irlampliadob, ir2ampliadob;
/* Parâmetros dos testes */
decl estat_testeJ1, estat_testeJ2, estat_testeJ1b, estat_testeJ2b, estatMJ, estatMJb;
/* Parâmetros auxiliares */
decl chi2, ystar, ystarb;

dExecTime = timer(); /* Inicia o relógio */

ranseed("GM"); /* Gerador de números aleatórios */
ranseed({6548,8984}); /* Semente */

/* Inicializações */
s_vY = s_vYB = zeros(NOBS, 1); /* Vetores de amostra e amostra bootstrap */
irl = ir2 = zeros((k+m+2), NREP); /* Estimativas de MV */
irlb = ir2b = zeros((k+m+2), NBOOT); /* Estimativas de MV bootstrap */
falha1MC = falha2MC = falha1BOOT = falha2BOOT = 0; /* Contadores de falha */
falha1MCampliado = falha2MCampliado = 0; /* Contadores de falha */
falha1BOOTampliado = falha2BOOTampliado = 0; /* Contadores de falha */
acertomm = acertommmb = acertoJ = acertoJb = acertoMJ = acertoMJb = zeros(3,1); /* Contador
de acertos do teste J */
estat_testeJ1b = estat_testeJ2b = estatMJb = zeros(NBOOT, 1); /* Armazena est. bootstrap */
estat_testeJ1 = estat_testeJ2 = estatMJ = zeros(NREP, 1); /* Armazena estatísticas */

/* Quantis da qui-quadrado */
chi2 = quanchi(.99,1) | quanchi(.95,1) | quanchi(.90,1);

```

## A.5 MODELOS COM DIFERENÇA NA FUNÇÃO DE LIGAÇÃO DA DISPERSÃO

```

/* Obtendo matrizes de regressores */
s_mX = 1 ~ (ranu(10,k-1));
/* Replicando para manter o grau de heterogeneidade constante */
for(i = 10; i < NOBS; i += 10){
s_mX = s_mX | s_mX[0:9][];
}
s_mZ = s_mX[][0:1];

/* Valores verdadeiros dos parâmetros */
eta = s_mX * beta';
mu = exp(eta) ./ (1.0 + exp(eta)); /* inversa da ligação logit */
delta = s_mZ * gama';
phi = (delta .^ 2); /* inversa da ligação raiz */

decl lambda = max(phi)/min(phi);

/* Início do laço de Monte Carlo */
for(i = 0; i < NREP; i++){
for(j = 0; j < NOBS; j++){
s_vY[j] = ranbeta(1,1,mu[j]*phi[j], (1.0-mu[j])*phi[j]); /* beta(mu,phi) */

/* Obtendo y* */
ystar = log( s_vY ./ (1.0-s_vY) ); /* transformação logit */

/* Estimando o modelo m1 - ligação raiz(phi) */
irl[][i] = EstMV_raiz(ystar, s_mX, s_mZ);
if(irl[k+m+1][i] != MAX_CONV && irl[k+m+1][i] != MAX_WEAK_CONV){
falha1MC++;
i--;
continue;
}

/* Estimando o modelo m2 - log(phi) */
ir2[][i] = EstMV_log(ystar, s_mX, s_mZ);
if(ir2[k+m+1][i] != MAX_CONV && ir2[k+m+1][i] != MAX_WEAK_CONV){
falha2MC++;
i--;
continue;
}

decl deltab1 = s_mZ * irl[k:(k+m-1)][i];
decl phib1 = (deltab1 .^ 2); /* inversa da ligação raiz */
/* phib1 será também utilizado para gerar amostras do modelo 1 artificial */

decl deltab2 = s_mZ * ir2[k:(k+m-1)][i];
decl phib2 = exp(deltab2); /* inversa da ligação log */

/* Ampliando o modelo 1: incluindo eta do modelo 2 */
irlampliado = EstMV_raiz(ystar, s_mX, s_mZ~phib2 );
if(irlampliado[k+m+2] != MAX_CONV && irlampliado[k+m+2] != MAX_WEAK_CONV){
falha1MCampliado++;
i--;
continue;
}

/* Ampliando o modelo 2: incluindo eta do modelo 1 */
ir2ampliado = EstMV_log(ystar, s_mX, s_mZ~phib1 );
if(ir2ampliado[k+m+2] != MAX_CONV && ir2ampliado[k+m+2] != MAX_WEAK_CONV){
falha2MCampliado++;
i--;
continue;
}

/* Teste J - usando estatística RV */
estat_testeJ1[i] = 2*(irlampliado[k+m+1]-irl[k+m][i]); /* modelo 1 */

```

## A.5 MODELOS COM DIFERENÇA NA FUNÇÃO DE LIGAÇÃO DA DISPERSÃO

```

/* Teste MJ */
estat_testeJ2[i] = 2*(ir2ampliado[k+m+1]-ir2[k+m][i]); /* modelo 2 */
estatMJ[i] = min(estat_testeJ1[i], estat_testeJ2[i]);

/* Parâmetros a serem usados no laço bootstrap */
/* Modelo 1 artificial */
decl etab1 = s_mX*irl[0:(k-1)][i];
decl mubl = exp(etab1) ./ (1.0 + exp(etab1)); /* inversa da ligação logit */
/* phib1 definido anteriormente */

boot = 1; /* indica que vai entrar no laço bootstrap */

/* Início do laço bootstrap */
for(b = 0; b < NBOOT; b++){

  /*** Modelo 1 ***/
  /* y* sob Ho: alpha=0 */
  for(j = 0; j < NOBS; j++)
    s_vYB[j] = ranbeta(1,1,mubl[j]*phib1[j], (1.0-mubl[j])*phib1[j]); /* beta(mu,phi) */

  ystarb = log( s_vYB ./ (1.0-s_vYB) ); /* transformação logit */

  /* Estimando o modelo 1 artificial - ligação raiz(phi) */
  irlb[][b] = EstMV_raiz(ystarb, s_mX, s_mZ);
  if(irlb[k+m+1][b] != MAX_CONV && irlb[k+m+1][b] != MAX_WEAK_CONV){
    falhaBOOT++; /* contando falhas de convergência */
    b--;
    continue;
  }

  /* Ampliando o modelo 1 artificial: incluindo eta do modelo 2 */
  irlampliadob = EstMV_raiz(ystarb, s_mX, s_mZ~phib2);
  if(irlampliadob[k+m+2] != MAX_CONV && irlampliadob[k+m+2] != MAX_WEAK_CONV){
    falhaBOOTampliado++; /* contando falhas de convergência */
    b--;
    continue;
  }

  /* Estimando o modelo 2 artificial - ligação log(phi) */
  ir2b[][b] = EstMV_log(ystarb, s_mX, s_mZ);
  if(ir2b[k+m+1][b] != MAX_CONV && ir2b[k+m+1][b] != MAX_WEAK_CONV){
    falha2BOOT++; /* contando falhas de convergência */
    b--;
    continue;
  }

  /* Ampliando o modelo 2 artificial: incluindo eta do modelo 1 */
  ir2ampliadob = EstMV_log(ystarb, s_mX, s_mZ~phib1);
  if(ir2ampliadob[k+m+2] != MAX_CONV && ir2ampliadob[k+m+2] != MAX_WEAK_CONV){
    falha2BOOTampliado++; /* contando falhas de convergência */
    b--;
    continue;
  }

  /* Estatística do teste J bootstrap - usando estatística escore */
  estat_testeJ1b[b] = 2*(irlampliadob[k+m+1] - irlb[k+m][b]); /* modelo 1 */

  /* Estatística do teste MJ bootstrap */
  estat_testeJ2b[b] = 2*(ir2ampliadob[k+m+1] - ir2b[k+m][b]); /* modelo 2 */
  estatMJb[b] = min(estat_testeJ1b[b], estat_testeJ2b[b]);
} /* Fim do laço bootstrap */

boot = 0; /* Indica que saiu do laço bootstrap */

/* Contagem de acertos do teste J */
if(estat_testeJ1[i] < chi2[0]) /* 1% sig. */

```

## A.5 MODELOS COM DIFERENÇA NA FUNÇÃO DE LIGAÇÃO DA DISPERSÃO

```

acertoJ[0]++;
if(estat_testeJ1[i] < chi2[1]) /* 5% sig. */
acertoJ[1]++;
if(estat_testeJ1[i] < chi2[2]) /* 10% sig. */
acertoJ[2]++;

/* Contagem de acertos do teste J bootstrap */
if(estat_testeJ1[i] < quantilec(estat_testeJ1b, .99)) /* 1% sig. */
acertoJb[0]++;
if(estat_testeJ1[i] < quantilec(estat_testeJ1b, .95)) /* 5% sig. */
acertoJb[1]++;
if(estat_testeJ1[i] < quantilec(estat_testeJ1b, .90)) /* 10% sig. */
acertoJb[2]++;

/* Contagem de acertos do teste MJ */
if(estatMJ[i] < chi2[0]) /* 1% sig. */
acertoMJ[0]++;
if(estatMJ[i] < chi2[1]) /* 5% sig. */
acertoMJ[1]++;
if(estatMJ[i] < chi2[2]) /* 10% sig. */
acertoMJ[2]++;

/* Contagem de acertos do teste MJ bootstrap */
if(estatMJ[i] < quantilec(estatMJb, .99)) /* 1% sig. */
acertoMJb[0]++;
if(estatMJ[i] < quantilec(estatMJb, .95)) /* 5% sig. */
acertoMJb[1]++;
if(estatMJ[i] < quantilec(estatMJb, .90)) /* 10% sig. */
acertoMJb[2]++;

/* Contagem de m=m* quando Ho nao foi rejeitada */
if(estatMJ[i] < chi2[0] && estatMJ[i] == estat_testeJ1[i])
    acertomm[0]++;
if(estatMJ[i] < chi2[1] && estatMJ[i] == estat_testeJ1[i])
    acertomm[1]++;
if(estatMJ[i] < chi2[2] && estatMJ[i] == estat_testeJ1[i])
    acertomm[2]++;

/* Contagem de m=m* quando Ho nao foi rejeitada via bootstrap */
if(estatMJ[i] < quantilec(estatMJb, .99) && estatMJ[i] == estat_testeJ1[i])
    acertommb[0]++;
if(estatMJ[i] < quantilec(estatMJb, .95) && estatMJ[i] == estat_testeJ1[i])
    acertommb[1]++;
if(estatMJ[i] < quantilec(estatMJb, .90) && estatMJ[i] == estat_testeJ1[i])
    acertommb[2]++;
} /* Fim do laço de Monte Carlo */

/* Impressão de resultados */
println( "\n\n PROGRAMA OX: ", oxfilename(0) );
println( " VERSAO OX: ", oxversion() );
println( " REGRESSAO BETA - Modelos nao Encaixados " );
println( " Erro na ligacao da dispersao." );
println( "\t\t Modelo 1: logit(mu_t) = beta0 + beta1*x1 + beta2*x2" );
println( "\t\t\t\t\t raiz(phi_t) = gama0 + gama1*x1" );
println( "\n\t\t Modelo 2: logit(mu_t) = beta0 + beta1*x1 + beta2*x2" );
println( "\t\t\t\t\t log(phi_t) = gama0 + gama1*x1" );
println( "\n Hipotese nula (Ho): Modelo 1 correto." );
println( " TAMANHO DA AMOSTRA: ", NOBS );
println( " NUMERO DE REPLICAS MC:", NREP );
println( " NUMERO DE REPLICAS BOOTSTRAP:", NBOOT );
println( "\t NUM. REPLICAS MC SEM CONVERGENCIA: " );
println( "\t\t Modelo 1: ", falha1MC );
println( "\t\t Modelo 2: ", falha2MC );
println( "\t NUM. REPLICAS MC SEM CONVERGENCIA - Modelo ampliado: " );
println( "\t\t Modelo 1: ", falha1MCampliado );
println( "\t\t Modelo 2: ", falha2MCampliado );

```

## A.5 MODELOS COM DIFERENÇA NA FUNÇÃO DE LIGAÇÃO DA DISPERSÃO

```

println( "\t NUM. REPLICAS BOOTSTRAP SEM CONVERGENCIA: " );
println( "\t\t Modelo 1: ", falha1BOOT );
println( "\t\t Modelo 2: ", falha2BOOT );
println( "\t NUM. REPLICAS BOOTSTRAP SEM CONVERGENCIA - Modelo ampliado: " );
println( "\t\t Modelo 1: ", falha1BOOTampliado );
println( "\t\t Modelo 2: ", falha2BOOTampliado );
println( " VALORES VERDADEIROS DOS PARAMETROS:" );
println( "%r", {" beta0:", " betal:"}, beta', "%r", {" gama0:", " gamal:"}, gama' );
println( "\n ESTIMATIVAS DOS PARAMETROS: " );
println( "%r", {" beta0:", " betal:", " gama0:", " gamal:"}, meanr(irl[0:k+m-1][]) );
println( "\n Grau de heterogeneidade da dispersao: ", lambda );
println("\n Momentos: ", "%c", {"Estat. J", "Estat. MJ"}, "%r",
{" n:", " media:", " erro padrao:", " assimetria:", " curtose:"},
moments(estat_testeJ1) ~ moments(estatMJ));
println("\n Quantis: ", "%c", {"Estat. J", "Estat. MJ"}, "%r", {" 90%:", " 95%:", " 99%:"},
(quantilec(estat_testeJ1, .90) | quantilec(estat_testeJ1, .95) | quantilec(estat_testeJ1, .99))
~ (quantilec(estatMJ, .90) | quantilec(estatMJ, .95) | quantilec(estatMJ, .99)));
println( "\n TAXA DE FALHA (%) DO TESTE J:" );
println( "%r", {" 1%:", " 5%:", " 10%:"}, "%c", {"Assintot.", "boot."},
100*(1 - (acertoJ/NREP)) ~ 100*(1 - (acertoJb/NREP)) );
println( "\n TAXA DE FALHA (%) DO TESTE MJ:" );
println( "%r", {" 1%:", " 5%:", " 10%:"}, "%c", {"Assintot.", "boot."},
100*(1 - (acertoMJ/NREP)) ~ 100*(1 - (acertoMJb / NREP)) );
println("Proporcao de m=m* do teste MJ:", "%c", {"assintot.", "boot."},
100*(acertomm./acertoMJ) ~ 100*(acertomb./acertoMJb));
println("TEMPO DE EXECUCAO: ", timespan(dExecTime));
}

```

```

EstMV_log(const ystar, const mX, const mZ){
decl betaols, etaols, muols, varols, gamaols, vtheta, dfunc, converge;
decl l = sizec(mZ);
/* Chute inicial para beta */
if(k > 1)
ols2c(ystar, mX, &betaols); /* Estimativas de MQO */
else if(k == 1)
betaols = meanc(ystar);
/* Quantidades para o chute inicial de gama */
etaols = mX * betaols;
muols = exp(etaols) ./ (1.0 + exp(etaols)); /* inversa da ligação logit */
varols = ( ((ystar-etaols)' * (ystar - etaols)) .* ((muols.*(1.0-muols)).^2) ) ./
(NOBS - k);
gamaols = meanc( muols .* (1.0-muols) ./ varols ) - 1.0;
gamaols = log(gamaols) | zeros(l-1,1); /* Chute inicial para gama */

vtheta = betaols | gamaols;
MaxControl(50, -1);

s_mCovMU = mX; /* covariáveis do modelo para mu */
s_mCovPHI = mZ; /* covariáveis do modelo para phi */

/* Estimação de máxima verossimilhança */
converge = MaxBFGS(floglik_logit_log, &vtheta, &dfunc, 0, 0);

return vtheta | dfunc | converge;
}

```

```

EstMV_raiz(const ystar, const mX, const mZ){
decl betaols, etaols, muols, varols, gamaols, vtheta, dfunc, converge;
decl l = sizec(mZ);
/* Chute inicial para beta */
if(k > 1)
ols2c(ystar, mX, &betaols); /* Estimativas de MQO */
else if(k == 1)
betaols = meanc(ystar);
/* Quantidades para o chute inicial de gama */
etaols = mX * betaols;

```

## A.6 MODELOS COM DIFERENÇA NAS FUNÇÕES DE LIGAÇÃO DA MÉDIA E DA DISPERSÃO

```
muols = exp(etaols) ./ (1.0 + exp(etaols)); /* inversa da ligação logit */
varols = ( (ystar-etaols)' * (ystar - etaols) .* ((muols.*(1.0-muols)).^2) ) ./
(NOBS - k);
gamaols = meanc( muols .* (1.0-muols) ./ varols ) - 1.0;
gamaols = (gamaols .^ .5) | zeros(1-1,1); /* Chute inicial para gama */

vtheta = betaols | gamaols;
MaxControl(50, -1);

s_mCovMU = mX; /* covariáveis do modelo para mu */
s_mCovPHI = mZ; /* covariáveis do modelo para phi */

/* Estimação de máxima verossimilhança */
converge = MaxBFGS(floglik_logit_raiz, &vtheta, &dfunc, 0, 0);

return vtheta | dfunc | converge;
}
```

## A.6 Modelos com diferença nas funções de ligação da média e da dispersão

```
/******
*
* PROGRAMA: betareg_nonnested_linkmedia_linkdisp_n20_4.0.ox
*
* DESCRIÇÃO: realiza os testes J (com teste da razão de veros-
* similhanças) e MJ para modelos de regressão beta com disper-
* são variável
*
* MODELOS: (M1) log-log(mu_t) = beta0 + beta1*x1 + beta2*x2
* raiz(phi_t) = gama0 + gama1*x1
*
* (M2) logit(mu_t) = beta0 + beta1*x1 + beta2*x2
* log(phi_t) = gama0 + gama1*x1
*
* AUTOR: Sadraque Eneas de Figueiredo Lucena
*
* VERSÃO: 4.0
*
* DATA: 05/06/2012
*
* OBS.: Programa desenvolvido com base no programa apresentado
* na dissertação de mestrado de Queiroz (2011).
*
* *****/

/* Arquivos cabeçalho */
#include <oxstd.h> /* Biblioteca padrão do OX */
#include <oxprob.h> /* Necessário para gerar ocorrências beta */
#import <maximize> /* Necessário para otimização não-linear */

/* Variáveis globais */
static decl s_vY; /* Amostra aleatória */
static decl s_mX; /* Matriz X */
static decl s_mZ; /* Matriz Z */
static decl s_vYB; /* Amostra bootstrap */
static decl s_mCovMU; /* Matriz de covariáveis do modelo para mu */
static decl s_mCovPHI; /* Matriz de covariáveis do modelo para phi */

static decl boot = 0;

/* Definição de constantes */
```

## A.6 MODELOS COM DIFERENÇA NAS FUNÇÕES DE LIGAÇÃO DA MÉDIA E DA DISPERSÃO

```

const decl NOBS = 20; /* Tamanho amostral */
const decl NREP = 10000; /* Número de réplicas de Monte Carlo */
const decl NBOOT = 1000; /* Número de amostras bootstrap */
const decl k = 3; /* Número de parâmetros no modelo para mu */
const decl m = 2; /* Número de parâmetros no modelo para phi */
const decl beta = <1.0, 1.6, -2.1>; /* Parâmetros do modelo para mu */
const decl gama = <1.2, 1.7>; /* Parâmetros do modelo para phi */

/* Algumas funções */
EstMV_logit_log(const ystar, const mX, const mZ); /* EMV do modelo com regressores X e Z */
EstMV_loglog_raiz(const ystar, const mX, const mZ); /* EMV do modelo com ligação log-log */

/* Função de log-verossimilhança do modelo com ligações logit(mu) e log(phi) */
floglik_logit_log(const vtheta, const adFunc, const avScore, const amHess)
{
  decl y;
  if(boot == 1) /* usar amostra bootstrap */
    y = s_vYB;
  if(boot == 0) /* usar amostra */
    y = s_vY;

  decl k = sizec(s_mCovMU); /* número de colunas da matriz de regressores de mu */
  decl m = sizec(s_mCovPHI); /* número de colunas da matriz de regressores de phi */
  decl eta = s_mCovMU * vtheta[0:(k-1)]; /* usado para obter mu */
  decl mu = exp(eta) ./ (1.0 + exp(eta)); /* inversa da ligação logit */
  decl delta = s_mCovPHI * vtheta[k:(k+m-1)]; /* usado para obter phi */
  decl phi = exp(delta); /* inversa da ligação log */
  decl i, funlog = zeros(NOBS, 1);
  for(i = 0; i < NOBS; i++) /* logaritmo da densidade */
    funlog[i] = log( densbeta(y[i], mu[i] * phi[i], (1 - mu[i]) * phi[i]) );

  adFunc[0] = double( sumc( funlog ) ); /* log-verossimilhança */

  /* Parâmetros utilizados no cálculo das primeiras derivadas */
  decl ystar = log(y ./ (1.0 - y));
  decl mustar = polygamma(mu.*phi,0) - polygamma((1.0-mu).*phi,0);
  decl ydag = log(1.0 - y);
  decl mudag = polygamma((1.0-mu).*phi,0) - polygamma(phi,0);
  decl T = diag( mu .* (1-mu) ); /* 1/g'(mu) */
  decl F = diag(phi);
  decl H = diag(phi); /* 1/h'(phi) */
  decl M = diag(mu);

  if(avScore){
    (avScore[0])[0:(k-1)] = s_mCovMU' * T * F * (ystar - mustar);
    (avScore[0])[k:(k+m-1)] = s_mCovPHI' * H * (M * (ystar - mustar) + (ydag - mudag));
  }

  if( isnan(adFunc[0]) || isdotinf(adFunc[0]) )
    return 0;
  else
    return 1; /* 1 indica sucesso */
}

/* Função de log-verossimilhança do modelo com ligações log-log(mu) e raiz(phi) */
floglik_loglog_raiz(const vtheta, const adFunc, const avScore, const amHess)
{
  decl y;
  if(boot == 1) /* usar amostra bootstrap */
    y = s_vYB;
  if(boot == 0) /* usar amostra */
    y = s_vY;

  decl k = sizec(s_mCovMU); /* número de colunas da matriz de regressores de mu */
  decl m = sizec(s_mCovPHI); /* número de colunas da matriz de regressores de phi */
  decl eta = s_mCovMU * vtheta[0:(k-1)]; /* usado para obter mu */

```

## A.6 MODELOS COM DIFERENÇA NAS FUNÇÕES DE LIGAÇÃO DA MÉDIA E DA DISPERSÃO

```

decl mu = exp( -exp( -eta ) ); /* inversa da ligação log-log */
decl delta = s_mCovPHI * vtheta[k:(k+m-1)]; /* usado para obter phi */
decl phi = (delta .^ 2); /* inversa da ligação raiz */
decl i, funlog = zeros(NOBS, 1);
for(i = 0; i < NOBS; i++) /* logaritmo da densidade */
funlog[i] = log( densbeta(y[i], mu[i] * phi[i], (1 - mu[i]) * phi[i]) );

adFunc[0] = double( sumc( funlog ) ); /* log-verossimilhança */

/* Parâmetros utilizados no cálculo das primeiras derivadas */
decl ystar = log(y ./ (1.0 - y));
decl mustar = polygamma(mu.*phi,0) - polygamma((1.0-mu).*phi,0);
decl ydag = log(1.0 - y);
decl mudag = polygamma((1.0-mu).*phi,0) - polygamma(phi,0);
decl T = diag( -mu .* log(mu) ); /* 1/g'(mu) */
decl F = diag(phi);
decl H = diag( 2 * (phi .^ .5) ); /* 1/h'(phi) */
decl M = diag(mu);

if(avScore){
(avScore[0])[0:(k-1)] = s_mCovMU' * T * F * (ystar - mustar);
(avScore[0])[k:(k+m-1)] = s_mCovPHI' * H * (M * (ystar - mustar) + (ydag - mudag));
}

if( isnan(adFunc[0]) || isdotinf(adFunc[0]) )
return 0;
else
return 1; /* 1 indica sucesso */
}

/* INÍCIO DO PROGRAMA */
main()
{
/* Declaração de variáveis utilizadas no programa */
/* Parâmetros do sistema */
decl dExecTime, i, j, b, falha1MC, falha2MC, falha1MCampliado, falha2MCampliado,
acertoJ, acertoJb, acertoMJ, acertoMJb, falha1BOOT, falha2BOOT, falha1BOOTampliado,
falha2BOOTampliado, acertomm, acertommmb;
/* Parâmetros do modelo */
decl eta, mu, delta, phi, irl, ir2, irlampliado, ir2ampliado;
/* Parâmetros do modelo bootstrap */
decl irlb, ir2b, irlampliadob, ir2ampliadob;
/* Parâmetros dos testes */
decl estat_testeJ1, estat_testeJ2, estat_testeJ1b, estat_testeJ2b, estatMJ, estatMJb;
/* Parâmetros auxiliares */
decl chi2, ystar, ystarb;

dExecTime = timer(); /* Inicia o relógio */

ranseed("GM"); /* Gerador de números aleatórios */
ranseed({6548,8984}); /* Semente */

/* Inicializações */
s_vY = s_vYB = zeros(NOBS, 1); /* Vetores de amostra e amostra bootstrap */
irl = ir2 = zeros((k+m+2), NREP); /* Estimativas de MV */
irlb = ir2b = zeros((k+m+2), NBOOT); /* Estimativas de MV bootstrap */
falha1MC = falha2MC = falha1BOOT = falha2BOOT = 0; /* Contadores de falha */
falha1MCampliado = falha2MCampliado = 0; /* Contadores de falha */
falha1BOOTampliado = falha2BOOTampliado = 0; /* Contadores de falha */
acertomm = acertommmb = acertoJ = acertoJb = acertoMJ = acertoMJb = zeros(3,1); /* Contador
de acertos do teste J */
estat_testeJ1b = estat_testeJ2b = estatMJb = zeros(NBOOT, 1); /* Armazena est. bootstrap */
estat_testeJ1 = estat_testeJ2 = estatMJ = zeros(NREP, 1); /* Armazena estatísticas */

/* Quantis da qui-quadrado */
chi2 = quanchi(.99,2) | quanchi(.95,2) | quanchi(.90,2);

```

## A.6 MODELOS COM DIFERENÇA NAS FUNÇÕES DE LIGAÇÃO DA MÉDIA E DA DISPERSÃO

```

/* Obtendo matrizes de regressores */
s_mX = 1 ~ (ranu(10,k-1));
/* Replicando para manter o grau de heterogeneidade constante */
for(i = 10; i < NOBS; i += 10){
s_mX = s_mX | s_mX[0:9][];
}
s_mZ = s_mX[][0:1];

/* Valores verdadeiros dos parâmetros */
eta = s_mX * beta';
mu = exp( -exp( -eta ) ); /* inversa da ligação log-log */
delta = s_mZ * gama';
phi = (delta .^ 2); /* inversa da ligação raiz */

decl lambda = max(phi)/min(phi);

/* Início do laço de Monte Carlo */
for(i = 0; i < NREP; i++){
for(j = 0; j < NOBS; j++){
s_vY[j] = ranbeta(1,1,mu[j]*phi[j], (1.0-mu[j])*phi[j]); /* beta(mu,phi) */

/* Obtendo y* */
ystar = -log( -log(s_vY) ); /* transformação log-log */

/* Estimando o modelo m1 - ligação log-log(mu) e raiz(phi) */
irl[][i] = EstMV_loglog_raiz(ystar, s_mX, s_mZ);
if(irl[k+m+1][i] != MAX_CONV && irl[k+m+1][i] != MAX_WEAK_CONV){
falha1MC++;
i--;
continue;
}

/* Estimando o modelo m2 - ligação logit(mu) e log(phi) */
ir2[][i] = EstMV_logit_log(ystar, s_mX, s_mZ);
if(ir2[k+m+1][i] != MAX_CONV && ir2[k+m+1][i] != MAX_WEAK_CONV){
falha2MC++;
i--;
continue;
}

decl etab1 = s_mX*irl[0:(k-1)][i];
decl mub1 = exp( -exp( -etab1 ) ); /* inversa da ligação log-log */
decl deltab1 = s_mZ * irl[k:(k+m-1)][i];
decl phib1 = (deltab1 .^ 2); /* inversa da ligação raiz */
/* mub1 e phib1 serão também utilizado para gerar amostras do modelo 1 artificial */

decl etab2 = s_mX*ir2[0:(k-1)][i];
decl mub2 = exp(etab2) ./ (1.0 + exp(etab2)); /* inversa da ligação logit */
decl deltab2 = s_mZ * ir2[k:(k+m-1)][i];
decl phib2 = exp(deltab2); /* inversa da ligação log */

/* Ampliando o modelo 1: incluindo eta do modelo 2 */
irlampliado = EstMV_loglog_raiz(ystar, s_mX~mub2, s_mZ~phib2 );
if(irlampliado[k+m+3] != MAX_CONV && irlampliado[k+m+3] != MAX_WEAK_CONV){
falha1MCampliado++;
i--;
continue;
}

/* Ampliando o modelo 2: incluindo eta do modelo 1 */
ir2ampliado = EstMV_logit_log(ystar, s_mX~mub1, s_mZ~phib1 );
if(ir2ampliado[k+m+3] != MAX_CONV && ir2ampliado[k+m+3] != MAX_WEAK_CONV){
falha2MCampliado++;
i--;
continue;
}

```

## A.6 MODELOS COM DIFERENÇA NAS FUNÇÕES DE LIGAÇÃO DA MÉDIA E DA DISPERSÃO

```

}

/* Teste J - usando estatística RV */
estat_testeJ1[i] = 2*(irlampliado[k+m+2]-irl[k+m][i]); /* modelo 1 */

/* Teste MJ */
estat_testeJ2[i] = 2*(ir2ampliado[k+m+2]-ir2[k+m][i]); /* modelo 2 */
estatMJ[i] = min(estat_testeJ1[i], estat_testeJ2[i]);

boot = 1; /* indica que vai entrar no laço bootstrap */

/* Início do laço bootstrap */
for(b = 0; b < NBOOT; b++){

  /*** Modelo 1 ***/
  /* y* sob Ho: alpha=0 */
  for(j = 0; j < NOBS; j++){
    s_vYB[j] = ranbeta(1,1,mub1[j]*phib1[j], (1.0-mub1[j])*phib1[j]); /* beta(mu,phi) */

    ystarb = -log( -log( s_vYB ) ); /* transformação log-log */

    /* Estimando o modelo 1 artificial - logações log-log(mu) e raiz(phi) */
    irlb[][b] = EstMV_loglog_raiz(ystarb, s_mX, s_mZ);
    if(irlb[k+m+1][b] != MAX_CONV && irlb[k+m+1][b] != MAX_WEAK_CONV){
      falha1BOOT++; /* contando falhas de convergência */
      b--;
      continue;
    }

    /* Ampliando o modelo 1 artificial: incluindo eta do modelo 2 */
    irlampliadob = EstMV_loglog_raiz(ystarb, s_mX~mub2, s_mZ~phib2);
    if(irlampliadob[k+m+3] != MAX_CONV && irlampliadob[k+m+3] != MAX_WEAK_CONV){
      falha1BOOTampliado++; /* contando falhas de convergência */
      b--;
      continue;
    }

    /* Estimando o modelo 2 artificial - ligações logit(mu) e log(phi) */
    ir2b[][b] = EstMV_logit_log(ystarb, s_mX, s_mZ);
    if(ir2b[k+m+1][b] != MAX_CONV && ir2b[k+m+1][b] != MAX_WEAK_CONV){
      falha2BOOT++; /* contando falhas de convergência */
      b--;
      continue;
    }

    /* Ampliando o modelo 2 artificial: incluindo eta do modelo 1 */
    ir2ampliadob = EstMV_logit_log(ystarb, s_mX~mub1, s_mZ~phib1);
    if(ir2ampliadob[k+m+3] != MAX_CONV && ir2ampliadob[k+m+3] != MAX_WEAK_CONV){
      falha2BOOTampliado++; /* contando falhas de convergência */
      b--;
      continue;
    }

    /* Estatística do teste J bootstrap - usando estatística escore */
    estat_testeJ1b[b] = 2*(irlampliadob[k+m+2] - irlb[k+m][b]); /* modelo 1 */

    /* Estatística do teste MJ bootstrap */
    estat_testeJ2b[b] = 2*(ir2ampliadob[k+m+2] - ir2b[k+m][b]); /* modelo 2 */
    estatMJb[b] = min(estat_testeJ1b[b], estat_testeJ2b[b]);
  } /* Fim do laço bootstrap */

  boot = 0; /* Indica que saiu do laço bootstrap */

  /* Contagem de acertos do teste J */
  if(estat_testeJ1[i] < chi2[0]) /* 1% sig. */
    acertoJ[0]++;
}

```

## A.6 MODELOS COM DIFERENÇA NAS FUNÇÕES DE LIGAÇÃO DA MÉDIA E DA DISPERSÃO

```

if(estat_testeJ1[i] < chi2[1]) /* 5% sig. */
acertoJ[1]++;
if(estat_testeJ1[i] < chi2[2]) /* 10% sig. */
acertoJ[2]++;

/* Contagem de acertos do teste J bootstrap */
if(estat_testeJ1[i] < quantilec(estat_testeJ1b, .99)) /* 1% sig. */
acertoJb[0]++;
if(estat_testeJ1[i] < quantilec(estat_testeJ1b, .95)) /* 5% sig. */
acertoJb[1]++;
if(estat_testeJ1[i] < quantilec(estat_testeJ1b, .90)) /* 10% sig. */
acertoJb[2]++;

/* Contagem de acertos do teste MJ */
if(estatMJ[i] < chi2[0]) /* 1% sig. */
acertoMJ[0]++;
if(estatMJ[i] < chi2[1]) /* 5% sig. */
acertoMJ[1]++;
if(estatMJ[i] < chi2[2]) /* 10% sig. */
acertoMJ[2]++;

/* Contagem de acertos do teste MJ bootstrap */
if(estatMJ[i] < quantilec(estatMJb, .99)) /* 1% sig. */
acertoMJb[0]++;
if(estatMJ[i] < quantilec(estatMJb, .95)) /* 5% sig. */
acertoMJb[1]++;
if(estatMJ[i] < quantilec(estatMJb, .90)) /* 10% sig. */
acertoMJb[2]++;

/* Contagem de m=m* quando Ho nao foi rejeitada */
if(estatMJ[i] < chi2[0] && estatMJ[i] == estat_testeJ1[i])
acertomm[0]++;
if(estatMJ[i] < chi2[1] && estatMJ[i] == estat_testeJ1[i])
acertomm[1]++;
if(estatMJ[i] < chi2[2] && estatMJ[i] == estat_testeJ1[i])
acertomm[2]++;

/* Contagem de m=m* quando Ho nao foi rejeitada via bootstrap */
if(estatMJ[i] < quantilec(estatMJb, .99) && estatMJ[i] == estat_testeJ1[i])
acertommb[0]++;
if(estatMJ[i] < quantilec(estatMJb, .95) && estatMJ[i] == estat_testeJ1[i])
acertommb[1]++;
if(estatMJ[i] < quantilec(estatMJb, .90) && estatMJ[i] == estat_testeJ1[i])
acertommb[2]++;
} /* Fim do laço de Monte Carlo */

/* Impressão de resultados */
println( "\n\n PROGRAMA OX: ", oxfilename(0) );
println( " VERSAO OX: ", oxversion() );
println( " REGRESSAO BETA - Modelos nao Encaixados" );
println( " Erro na ligacao da media e da dipersao." );
println( "\t\t Modelo 1: log-log(mu_t) = beta0 + betal*x1 + beta2*x2" );
println( "\t\t\t\t\t raiz(phi_t) = gama0 + gamal*x1" );
println( "\n\t\t Modelo 2: logit(mu_t) = beta0 + betal*x1 + beta2*x2" );
println( "\t\t\t\t\t log(phi_t) = gama0 + gamal*x" );
println( "\n Hipotese nula (Ho): Modelo 1 correto." );
println( " TAMANHO DA AMOSTRA: ", NOBS );
println( " NUMERO DE REPLICAS MC:", NREP );
println( " NUMERO DE REPLICAS BOOTSTRAP:", NBOOT );
println( "\t NUM. REPLICAS MC SEM CONVERGENCIA: " );
println( "\t\t Modelo 1: ", falha1MC );
println( "\t\t Modelo 2: ", falha2MC );
println( "\t NUM. REPLICAS MC SEM CONVERGENCIA - Modelo ampliado: " );
println( "\t\t Modelo 1: ", falha1MCampliado );
println( "\t\t Modelo 2: ", falha2MCampliado );
println( "\t NUM. REPLICAS BOOTSTRAP SEM CONVERGENCIA: " );

```

## A.6 MODELOS COM DIFERENÇA NAS FUNÇÕES DE LIGAÇÃO DA MÉDIA E DA DISPERSÃO

```

println( "\t\t Modelo 1: ", falha1BOOT );
println( "\t\t Modelo 2: ", falha2BOOT );
println( "\t NUM. REPLICAS BOOTSTRAP SEM CONVERGENCIA - Modelo ampliado: " );
println( "\t\t Modelo 1: ", falha1BOOTampliado );
println( "\t\t Modelo 2: ", falha2BOOTampliado );
println( " VALORES VERDADEIROS DOS PARAMETROS:" );
println( "%r", {" beta0:", " betal:"}, beta', "%r", {" gama0:", " gamal:"}, gama' );
println( "\n ESTIMATIVAS DOS PARAMETROS: " );
println( "%r", {" beta0:", " betal:", " gama0:", " gamal:"}, meanr(irl[0:k+m-1][]) );
println( "\n Grau de heterogeneidade da dispersao: ", lambda );
println( "\n Momentos: ", "%c", {"Estat. J", "Estat. MJ"}, "%r",
{" n:", " media:", " erro padrao:", " assimetria:", " curtose:"},
moments(estat_testeJ1) ~ moments(estatMJ));
println( "\n Quantis: ", "%c", {"Estat. J", "Estat. MJ"}, "%r", {" 90%:", " 95%:", " 99%:"},
(quantilec(estat_testeJ1, .90) | quantilec(estat_testeJ1, .95) | quantilec(estat_testeJ1, .99))
~ (quantilec(estatMJ, .90) | quantilec(estatMJ, .95) | quantilec(estatMJ, .99)));
println( "\n TAXA DE FALHA (%) DO TESTE J:" );
println( "%r", {" 1%:", " 5%:", " 10%:"}, "%c", {"Assintot.", "boot."},
100*(1 - (acertoJ/NREP)) ~ 100*(1 - (acertoJb/NREP)) );
println( "\n TAXA DE FALHA (%) DO TESTE MJ:" );
println( "%r", {" 1%:", " 5%:", " 10%:"}, "%c", {"Assintot.", "boot."},
100*(1 - (acertoMJ/NREP)) ~ 100*(1 - (acertoMJb / NREP)) );
println( "Proporcao de m=m* do teste MJ:", "%c", {"assintot.", "boot."},
100*(acertomm./acertoMJ) ~ 100*(acertommb./acertoMJb));
println( "TEMPO DE EXECUCAO: ", timespan(dExecTime));
}

EstMV_logit_log(const ystar, const mX, const mZ){
decll betaols, etaols, muols, varols, gamaols, vtheta, dfunc, converge;
decll l = sizec(mZ);
/* Chute inicial para beta */
if(k > 1)
ols2c(ystar, mX, &betaols); /* Estimativas de MQO */
else if(k == 1)
betaols = meanc(ystar);
/* Quantidades para o chute inicial de gama */
etaols = mX * betaols;
muols = exp(etaols) ./ (1.0 + exp(etaols)); /* inversa da ligacão logit */
varols = ( (ystar-etaols)' * (ystar - etaols)) .* ((muols.*(1.0-muols)).^2) ./
(NOBS - k);
gamaols = meanc( muols .* (1.0-muols) ./ varols ) - 1.0;
gamaols = log(gamaols) | zeros(l-1,1); /* Chute inicial para gama */

vtheta = betaols | gamaols;
MaxControl(50, -1);

s_mCovMU = mX; /* covariáveis do modelo para mu */
s_mCovPHI = mZ; /* covariáveis do modelo para phi */

/* Estimacão de máxima verossimilhança */
converge = MaxBFGS(floglik_logit_log, &vtheta, &dfunc, 0, 0);

return vtheta | dfunc | converge;
}

EstMV_loglog_raiz(const ystar, const mX, const mZ){
decll betaols, etaols, muols, varols, gamaols, vtheta, dfunc, converge;
decll l = sizec(mZ);
/* Chute inicial para beta */
if(k > 1)
ols2c(ystar, mX, &betaols); /* Estimativas de MQO */
else if(k == 1)
betaols = meanc(ystar);
/* Quantidades para o chute inicial de gama */
etaols = mX * betaols;
muols = exp(-exp(-etaols)); /* inversa da ligacão log-log */

```

## A.6 MODELOS COM DIFERENÇA NAS FUNÇÕES DE LIGAÇÃO DA MÉDIA E DA DISPERSÃO

```
varols = ( (ystar-etaols)' * (ystar - etaols)) .* ((muols .* log(muols)).^2) ) ./  
  (NOBS - k); /* forma para a ligação log-log */  
gamaols = meanc( muols .* (1.0-muols) ./ varols ) - 1.0;  
gamaols = (gamaols .^ .5) | zeros(1-1,1); /* Chute inicial para gama */  
  
vtheta = betaols | gamaols;  
MaxControl(50, -1);  
  
s_mCovMU = mX; /* covariáveis do modelo para mu */  
s_mCovPHI = mZ; /* covariáveis do modelo para phi */  
  
/* Estimação de máxima verossimilhança */  
converge = MaxBFGS(floglik_loglog_raiz, &vtheta, &dfunc, 0, 0);  
  
return vtheta | dfunc | converge;  
}
```

## APÊNDICE B

# Códigos em R

A seguir são apresentados os comandos em R utilizados na análise dos dados de produção de gasolina apresentados na Seção 5.1 e para os dados de habilidade de leitura em crianças disléxicas e não-disléxicas (Seção 5.2).

### B.1 Dados de Produção de Gasolina

```
# Pacotes exigidos
require(betareg)
require(lmtest)

set.seed(16769362) # semente
BootRep <- 1000 # número de réplicas de bootstrap

# Ler banco de dados
data("GasolineYield")

# Modelo com ligação logit(mu)
gy_logit <- betareg(yield ~ batch + temp, data = GasolineYield)
summary(gy_logit)

# Modelo com ligação logit(mu) dispersão variável
gy_logit2 <- betareg(yield ~ batch + temp | temp, data = GasolineYield)
summary(gy_logit2)

# Testando dispersão constante
lrtest(gy_logit, gy_logit2)
waldtest(gy_logit, gy_logit2)

# Ajustando modelo com ligação log-log(mu)
gy_loglog <- betareg(yield ~ batch + temp, data = GasolineYield, link = "loglog")
summary(gy_loglog)

# Teste RESET (usando razão de log-verossimilhança)
lrtest(gy_logit2, . ~ . + I(predict(gy_logit2, type = "link")^2))
lrtest(gy_loglog, . ~ . + I(predict(gy_loglog, type = "link")^2))

# Pseudo R^2
summary(gy_logit2)$pseudo.r.squared
summary(gy_loglog)$pseudo.r.squared

# AIC
AIC(gy_logit2, gy_loglog)

# BIC
BIC(gy_logit2, gy_loglog)

# Gráfico de envelopes simulados
plot(gy_logit2, which = 5, type = "deviance", sub.caption = " ")
```

## B.1 DADOS DE PRODUÇÃO DE GASOLINA

```
title(expression(modelo~H[1]))
dev.copy2eps(file=~ /Documentos/Mestrado/Dissertação/figuras/EnvelopeSimulado_gy_logit2.eps")

plot(gy_loglog, which = 5, type = "deviance", sub.caption = "")
title(expression(modelo~H[2]))
dev.copy2eps(file=~ /Documentos/Mestrado/Dissertação/figuras/EnvelopeSimulado_gy_loglog.eps")

#####
## Teste J
#####

# Testando modelo gy_logit2

# Usando razão de verossimilhanças
Jtest.logit2.rv <- lrtest(gy_logit2, . ~ . + I(predict(gy_loglog)))
Jtest.logit2.rv

# Usando razão de verossimilhanças bootstrap
mu.logit2 <- predict(gy_logit2, type="response")
phi.logit2 <- predict(gy_logit2, type="precision")

y.boot <- vector()
Jtest.logit2.rv.b <- vector()
for(j in 1:BootRep){
  for(i in 1:length(GasolineYield[,5]))
    y.boot[i] <- rbeta(1,mu.logit2[i]*phi.logit2,(1-mu.logit2[i])*phi.logit2)

  # Estimandos os modelos
  gy_logit2.b <- betareg(y.boot ~ batch + temp | temp, data = cbind(y.boot,GasolineYield))
  gy_loglog.b <- betareg(y.boot ~ batch + temp, data = cbind(y.boot,GasolineYield),
    link = "loglog")

  # Estatística J
  Jtest.logit2.rv.b[j] <- lrtest(gy_logit2.b, . ~ . + I(predict(gy_loglog.b)))$Chisq[2]
}

pvalue.Jtest.logit2.rv.boot <- (1+sum(iffelse(Jtest.logit2.rv.b>Jtest.logit2.rv$Chisq[2],
  1,0)))/(1+BootRep)
pvalue.Jtest.logit2.rv.boot

# Usando Wald
Jtest.logit2.wald <- waldtest(gy_logit2, . ~ . + I(predict(gy_loglog)))
Jtest.logit2.wald

# Usando Wald bootstrap
mu.logit2 <- predict(gy_logit2, type="response")
phi.logit2 <- predict(gy_logit2, type="precision")

y.boot <- vector()
Jtest.logit2.wald.b <- vector()
for(j in 1:BootRep){
  for(i in 1:length(GasolineYield[,5]))
    y.boot[i] <- rbeta(1,mu.logit2[i]*phi.logit2,(1-mu.logit2[i])*phi.logit2)

  # Estimandos os modelos
  gy_logit2.b <- betareg(y.boot ~ batch + temp | temp, data = cbind(y.boot,GasolineYield))
  gy_loglog.b <- betareg(y.boot ~ batch + temp, data = cbind(y.boot,GasolineYield),
    link = "loglog")

  # Estatística J
  Jtest.logit2.wald.b[j] <- waldtest(gy_logit2.b, . ~ . + I(predict(gy_loglog.b)))$Chisq[2]
}

pvalue.Jtest.logit2.wald.boot <- (1+sum(iffelse(Jtest.logit2.wald.b>Jtest.logit2.wald$Chisq[2],
  1,0)))/(1+BootRep)
```

## B.1 DADOS DE PRODUÇÃO DE GASOLINA

```
pvalue.Jtest.logit2.wald.boot

# Testando o modelo log-log

# Usando a razão de verossimilhanças
Jtest.loglog.rv <- lrtest(gy_loglog, . ~ . + I(predict(gy_logit2)))
Jtest.loglog.rv

# Usando a razão de verossimilhanças bootstrap
mu.loglog <- predict(gy_loglog, type="response")
phi.loglog <- predict(gy_loglog, type="precision")

y.boot <- vector()
Jtest.loglog.rv.b <- vector()
for(j in 1:BootRep){
  for(i in 1:length(mu.loglog))
    y.boot[i] <- rbeta(1,mu.loglog[i]*phi.loglog, (1-mu.loglog[i])*phi.loglog)

  # Estimandos os modelos
  gy_logit2.b <- betareg(y.boot ~ batch + temp | temp, data = cbind(y.boot,GasolineYield))
  gy_loglog.b <- betareg(y.boot ~ batch + temp, data = cbind(y.boot,GasolineYield),
    link = "loglog")

  # Estatística J
  Jtest.loglog.rv.b[j] <- lrtest(gy_loglog.b, . ~ . + I(predict(gy_logit2.b)))$Chisq[2]
}

pvalue.Jtest.loglog.rv.boot <- (1+sum(iffelse(Jtest.loglog.rv.b>Jtest.loglog.rv$Chisq[2],
  1,0)))/(1+BootRep)

pvalue.Jtest.loglog.rv.boot

# Usando Wald
Jtest.loglog.wald <- waldtest(gy_loglog, . ~ . + I(predict(gy_logit2)))
Jtest.loglog.wald

# Usando Wald bootstrap
mu.loglog <- predict(gy_loglog, type="response")
phi.loglog <- predict(gy_loglog, type="precision")

y.boot <- vector()
Jtest.loglog.wald.b <- vector()
for(j in 1:BootRep){
  for(i in 1:length(mu.loglog))
    y.boot[i] <- rbeta(1,mu.loglog[i]*phi.loglog, (1-mu.loglog[i])*phi.loglog)

  # Estimandos os modelos
  gy_logit2.b <- betareg(y.boot ~ batch + temp | temp, data = cbind(y.boot,GasolineYield))

  gy_loglog.b <- betareg(y.boot ~ batch + temp, data = cbind(y.boot,GasolineYield),
    link = "loglog")

  # Estatística J
  Jtest.loglog.wald.b[j] <- waldtest(gy_loglog.b, . ~ . + I(predict(gy_logit2.b)))$Chisq[2]
}

pvalue.Jtest.loglog.wald.boot <- (1+sum(iffelse(Jtest.loglog.wald.b>Jtest.loglog.wald$Chisq[2],
  1,0)))/(1+BootRep)

pvalue.Jtest.loglog.wald.boot

#####
## Teste MJ
#####

# Usando razão de verossimilhanças
```

## B.2 HABILIDADE DE LEITURA EM CRIANÇAS DISLÉXICAS E NÃO-DISLÉXICAS

```
MJtest.rv <- min(Jtest.logit2.rv$Chisq[2], Jtest.loglog.rv$Chisq[2])
1-pchisq(MJtest.rv, 1)

# Usando razão de verossimilhanças bootstrap
mu.loglog <- predict(gy_loglog, type="response")
phi.loglog <- predict(gy_loglog, type="precision")

y.boot <- vector()
MJtest.rv.b <- vector()
for(j in 1:BootRep){
  for(i in 1:length(mu.loglog))
    y.boot[i] <- rbeta(1,mu.loglog[i]*phi.loglog[i], (1-mu.loglog[i])*phi.loglog[i])

  # Estimandos os modelos
  gy_logit2.b <- betareg(y.boot ~ batch + temp | temp, data = cbind(y.boot,GasolineYield))
  gy_loglog.b <- betareg(y.boot ~ batch + temp, data = cbind(y.boot,GasolineYield),
    link = "loglog")

  Jtest.logit2.b <- lrtest(gy_logit2.b, . ~ . + I(predict(gy_loglog.b)))$Chisq[2]
  Jtest.loglog.b <- lrtest(gy_loglog.b, . ~ . + I(predict(gy_logit2.b)))$Chisq[2]

  # Estatística MJ
  MJtest.rv.b[j] <- min(Jtest.loglog.b,Jtest.logit2.b)
}

pvalue.MJtest.rv.boot <- (1+sum(iffelse(MJtest.rv.b>MJtest.rv,1,0)))/(1+BootRep)
pvalue.MJtest.rv.boot

# Usando Wald
MJtest.wald <- min(Jtest.logit2.wald$Chisq[2], Jtest.loglog.wald$Chisq[2])
1-pchisq(MJtest.wald, 1)

# Usando Wald bootstrap
mu.loglog <- predict(gy_loglog, type="response")
phi.loglog <- predict(gy_loglog, type="precision")

y.boot <- vector()
MJtest.wald.b <- vector()
for(j in 1:BootRep){
  for(i in 1:length(mu.loglog))
    y.boot[i] <- rbeta(1,mu.loglog[i]*phi.loglog[i], (1-mu.loglog[i])*phi.loglog[i])

  # Estimandos os modelos
  gy_logit2.b <- betareg(y.boot ~ batch + temp | temp, data = cbind(y.boot,GasolineYield))
  gy_loglog.b <- betareg(y.boot ~ batch + temp, data = cbind(y.boot,GasolineYield),
    link = "loglog")

  Jtest.logit2.b <- waldtest(gy_logit2.b, . ~ . + I(predict(gy_loglog.b)))$Chisq[2]
  Jtest.loglog.b <- waldtest(gy_loglog.b, . ~ . + I(predict(gy_logit2.b)))$Chisq[2]

  # Estatística MJ
  MJtest.wald.b[j] <- min(Jtest.loglog.b,Jtest.logit2.b)
}

pvalue.MJtest.wald.boot <- (1+sum(iffelse(MJtest.wald.b>MJtest.wald,1,0)))/(1+BootRep)
pvalue.MJtest.wald.boot
```

## B.2 Habilidade de leitura em crianças disléxicas e não-disléxicas

```
# Pacotes exigidos
require(betareg)
```

## B.2 HABILIDADE DE LEITURA EM CRIANÇAS DISLÉXICAS E NÃO-DISLÉXICAS

```
require(lmtest)

set.seed(16759362) # semente
BootRep <- 1000 # número de réplicas de bootstrap

# Ler banco de dados
data("ReadingSkills")
iq2 = ReadingSkills$iq^2

# Modelo A
modelA <- betareg(accuracy ~ dyslexia + iq + dyslexia*iq | dyslexia + iq + iq2, link="logit",
  link.phi="log",data = ReadingSkills)
summary(modelA)

# Modelo B
modelB <- betareg(accuracy ~ dyslexia + iq + dyslexia*iq | dyslexia + iq + dyslexia*iq,
  link="logit", link.phi="log",data = ReadingSkills)
summary(modelB)

# Teste RESET (usando razão de log-verossimilhança)
lrtest(modelA, . ~ . + I(predict(modelA, type = "link")^2))
lrtest(modelB, . ~ . + I(predict(modelB, type = "link")^2))

# Pseudo R^2
summary(modelA)$pseudo.r.squared
summary(modelB)$pseudo.r.squared

# AIC
AIC(modelA, modelB)

# BIC
BIC(modelA, modelB)

# Gráfico de envelopes simulados
plot(modelA, which = 5, type = "deviance", sub.caption = "")
dev.copy2eps(file=~ /Documentos/Mestrado/Dissertação/figuras/EnvelopeSimulado_m1_smithson.eps")

plot(modelB, which = 5, type = "deviance", sub.caption = "")
dev.copy2eps(file=~ /Documentos/Mestrado/Dissertação/figuras/EnvelopeSimulado_m2_smithson.eps")

#####
## Teste J
#####

# modelo 1
modelAa <- betareg(accuracy ~ dyslexia + iq + dyslexia*iq | dyslexia + iq + iq2
  + predict(modelB,type="precision"), link="logit", link.phi="log",data = ReadingSkills)
Jtest.modelA <- lrtest(modelA, modelAa)
Jtest.modelA

Jtest.modelA.wald <- waldtest(modelA, modelAa)
Jtest.modelA.wald

# modelo 2
modelBa <- betareg(accuracy ~ dyslexia + iq + dyslexia*iq | dyslexia + iq + iq*dyslexia
  + predict(modelA,type="precision"), link="logit", link.phi="log",data = ReadingSkills)
Jtest.modelB <- lrtest(modelB, modelBa)
Jtest.modelB

Jtest.modelB.wald <- waldtest(modelB, modelBa)
Jtest.modelB.wald

# MJ
MJtest <- min(Jtest.modelA$Chisq[2],Jtest.modelB$Chisq[2])
1-pchisq(MJtest,1)
```

## B.2 HABILIDADE DE LEITURA EM CRIANÇAS DISLÉXICAS E NÃO-DISLÉXICAS

```
MJtest.wald <- min(Jtest.modeloA.wald$Chisq[2],Jtest.modeloB.wald$Chisq[2])
1-pchisq(MJtest.wald,1)

##### Versão bootstrap

## Teste J bootstrap para o modelo 1
mu.modeloA <- predict(modelA, type="response")
phi.modeloA <- predict(modelA, type="precision")

y.boot <- vector()
Jtest.modeloA.b <- vector()
Jtest.modeloA.wald.b <- vector()
for(j in 1:BootRep){
  for(i in 1:length(mu.modeloA))
    y.boot[i] <- rbeta(1,mu.modeloA[i]*phi.modeloA,(1-mu.modeloA[i])*phi.modeloA)

  # Estimandos os modelos
  modeloA.b <- betareg(y.boot ~ dyslexia + iq + dyslexia*iq | dyslexia + iq + iq2,
    link.phi="log", data = cbind(y.boot,ReadingSkills))
  modeloAa.b <- betareg(y.boot ~ dyslexia + iq + dyslexia*iq | dyslexia + iq + iq2
    + predict(modelB, type="precision"), link.phi="log",
    data = cbind(y.boot,ReadingSkills,iq2))

  # Estatística J
  Jtest.modeloA.b[j] <- lrtest(modelA.b, modeloAa.b)$Chisq[2]
  Jtest.modeloA.wald.b[j] <- waldtest(modelA.b, modeloAa.b)$Chisq[2]
}

pvalue.Jtest.modeloA.boot <- (1+sum(iffelse(Jtest.modeloA.b>Jtest.modeloA$Chisq[2],1,0),na.rm=T))/(
  (1+length(Jtest.modeloA.b))

pvalue.Jtest.modeloA.boot

pvalue.Jtest.modeloA.wald.boot <- (1+sum(iffelse(Jtest.modeloA.wald.b>Jtest.modeloA.wald$Chisq[2],
  1,0),na.rm=T))/(1+length(Jtest.modeloA.wald.b))

pvalue.Jtest.modeloA.wald.boot

## Teste J bootstrap para o modelo 2
mu.modeloB <- predict(modelB, type="response")
phi.modeloB <- predict(modelB, type="precision")

y.boot <- vector()
Jtest.modeloB.b <- vector()
Jtest.modeloB.wald.b <- vector()
for(j in 1:BootRep){
  for(i in 1:length(mu.modeloB))
    y.boot[i] <- rbeta(1,mu.modeloB[i]*phi.modeloB,(1-mu.modeloB[i])*phi.modeloB)

  # Estimandos os modelos
  modeloB.b <- betareg(y.boot ~ dyslexia + iq + dyslexia*iq | dyslexia + iq + dyslexia*iq,
    link.phi="log", data = cbind(y.boot,ReadingSkills))
  modeloBa.b <- betareg(y.boot ~ dyslexia + iq + dyslexia*iq | dyslexia + iq + dyslexia*iq
    + predict(modelA, type="precision"), link.phi="log",
    data = cbind(y.boot,ReadingSkills,iq2))

  # Estatística J
  Jtest.modeloB.b[j] <- lrtest(modelB.b, modeloBa.b)$Chisq[2]
  Jtest.modeloB.wald.b[j] <- waldtest(modelB.b, modeloBa.b)$Chisq[2]
}

pvalue.Jtest.modeloB.boot <- (1+sum(iffelse(Jtest.modeloB.b>Jtest.modeloB$Chisq[2],1,0),na.rm=T))/(
  (1+length(Jtest.modeloB.b))

pvalue.Jtest.modeloB.boot

pvalue.Jtest.modeloB.wald.boot <- (1+sum(iffelse(Jtest.modeloB.wald.b>Jtest.modeloB.wald$Chisq[2],
```

## B.2 HABILIDADE DE LEITURA EM CRIANÇAS DISLÉXICAS E NÃO-DISLÉXICAS

```
1,0),na.rm=T))/(1+length(Jtest.modelB.wald.b))
pvalue.Jtest.modelB.wald.boot

# Teste MJ bootstrap
mu.modelA <- predict(modelA, type="response")
phi.modelA <- predict(modelA, type="precision")

y.boot <- vector()
MJtest.b <- vector()
MJtest.wald.b <- vector()
for(j in 1:BootRep){
  for(i in 1:length(mu.modelA))
    y.boot[i] <- rbeta(1,mu.modelA[i]*phi.modelA,(1-mu.modelA[i])*phi.modelA)

  # Estimandos os modelos
  modelA.b <- betareg(y.boot ~ dyslexia + iq + dyslexia*iq | dyslexia + iq + iq2,
    link.phi="log", data = cbind(y.boot,ReadingSkills))
  modelAa.b <- betareg(y.boot ~ dyslexia + iq + dyslexia*iq | dyslexia + iq + iq2
    + predict(modelB, type="precision"), link.phi="log",
    data = cbind(y.boot,ReadingSkills,iq2))

  modelB.b <- betareg(y.boot ~ dyslexia + iq + dyslexia*iq | dyslexia + iq + dyslexia*iq,
    link.phi="log", data = cbind(y.boot,ReadingSkills))
  modelBa.b <- betareg(y.boot ~ dyslexia + iq + dyslexia*iq | dyslexia + iq + dyslexia*iq
    + predict(modelA, type="precision"), link.phi="log",
    data = cbind(y.boot,ReadingSkills))

  # Estatística J
  Jtest.modelA.b <- lrtest(modelA.b, modelAa.b)$Chisq[2]
  Jtest.modelA.wald.b <- waldtest(modelA.b, modelAa.b)$Chisq[2]

  Jtest.modelB.b <- lrtest(modelB.b, modelBa.b)$Chisq[2]
  Jtest.modelB.wald.b <- waldtest(modelB.b, modelBa.b)$Chisq[2]

  # Estatística MJ
  MJtest.b[j] <- min(Jtest.modelA.b, Jtest.modelB.b)
  MJtest.wald.b[j] <- min(Jtest.modelA.wald.b, Jtest.modelB.wald.b)
}

pvalue.MJtest.boot <- (1+sum(iffelse(MJtest.b>MJtest,1,0),na.rm=T))/(1+length(MJtest.b))
pvalue.MJtest.boot

pvalue.MJtest.wald.boot <- (1+sum(iffelse(MJtest.wald.b>MJtest,1,0),na.rm=T))/
  (1+length(MJtest.wald.b))
pvalue.MJtest.wald.boot
```