

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

José Sérgio Casé de Oliveira

Detectando Má Especificação 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
14 de fevereiro de 2013

Catálogo na fonte
Bibliotecária Monick Raquel Silvestre da Silva, CRB4-1217

Oliveira, José Sérgio Casé de
Detectando má especificação em regressão beta / José
Sérgio Casé de Oliveira. - Recife: O Autor, 2013.
xi, 90 f.: il., fig., tab.

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

Inclui bibliografia e apêndice.

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

310

CDD (23. ed.)

MEI2013 – 029

Universidade Federal de Pernambuco
Pós-Graduação em Estatística

Recife, 14 de fevereiro de 2013.

Nós recomendamos que a dissertação de mestrado de autoria de

José Sérgio Casé de Oliveira

Intitulada

“Detectando Má Especificação 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

Getúlio José Amorim do Amaral

UFPE

Aluísio de Souza Pinheiro

UNICAMP

Este documento será anexado à versão final da tese.

A meus pais, Clemirso e Helena.

Agradecimentos

Aos meus pais, por todo apoio e incentivo durante todos esses anos;

Ao meu orientador, Francisco Cribari Neto, pela paciência, dedicação e atenção em me atender sempre que necessitei;

Aos demais professores do Departamento de Estatística da UFPE, em especial aos professores André Toom, Leandro Rêgo, Audrey Cysneiros e Gauss Cordeiro, por todo inestimável conhecimento que passaram no decorrer do curso;

A Valéria Bittencourt, pelos conselhos, ajuda, atenção, competência, enfim, por ser essa pessoa maravilhosa e iluminada;

A minha tia Maria Cleonice, por me dar um lar em Recife;

Aos amigos, e em especial a minha grande amiga e namorada Maria Glaubervânia, pelo apoio e por compreenderem minha ausência em alguns momentos dessa caminhada;

Aos colegas de curso, pelas brincadeiras, risadas e troca de ideias. Em especial, a Marcos Pereira e Jéssica Rivas, pelas incontáveis horas de estudo conjunto e por todo incentivo;

Aos participantes da banca examinadora, pelas correções e sugestões;

A CAPES, pelo apoio financeiro.

Mathematics, rightly viewed, possesses not only truth, but supreme beauty—a beauty cold and austere, like that of sculpture, without appeal to any part of our weaker nature, without the gorgeous trappings of painting or music, yet sublimely pure, and capable of a stern perfection such as only the greatest art can show. The true spirit of delight, the exaltation, the sense of being more than Man, which is the touchstone of the highest excellence, is to be found in mathematics . . .

—BERTRAND RUSSELL (Mysticism and Logic: And Other Essays, 1919)

Resumo

Esta dissertação tem por objetivo avaliar o poder do teste de má especificação proposto por Cribari-Neto & Lima (2007) em vários cenários que configuram má especificação do modelo de regressão beta, em particular: função de ligação incorreta, presença de *outlier* na amostra, omissão de variável regressora importante, estimação com dispersão constante quando o modelo verdadeiro possui dispersão variável (e vice-versa) e má especificação da distribuição da variável resposta. O desempenho do teste foi avaliado em modelos de regressão beta com dispersão fixa e variável.

Adicionalmente, introduzimos um outro teste de má especificação, o qual também teve seu poder avaliado em diversos cenários de má especificação. O poder do teste proposto foi comparado ao do teste proposto por Cribari-Neto & Lima (2007).

Os desempenhos dos testes em amostras finitas foram avaliados numericamente por meio de simulações de Monte Carlo. Por fim, apresentamos algumas aplicações com dados reais.

Palavras-chave: Especificação incorreta; regressão beta; simulação de Monte Carlo; teste de hipóteses; teste da razão de verossimilhanças; teste escore; teste RESET; teste de Wald.

Abstract

The chief goal of this thesis is twofold. First, we propose a model misspecification test for beta regression. Our teste is a variant of that proposed by Cribari-Neto (2007). Like this test, our test is RESET-like. Second, we use Monte Carlo simulations to evaluate the size and power of both tests in finite samples. In particular, we carry out simulations under different sources of model misspecification in fixed and variable dispersion models. Empirical applications that use real (not simulated) data are also presented and discussed.

Keywords: Beta regression; Hypothesis test; Likelihood ratio test; Model misspecification; Monte Carlo simulation; RESET test; Score test; Wald test.

Lista de Figuras

2.1	Densidades beta para diferentes valores de p e q .	4
3.1	Funções de log-verossimilhança (θ escalar): teste da razão de verossimilhanças.	13
3.2	Funções de log-verossimilhança (θ escalar): teste Wald.	13
3.3	Funções de log-verossimilhança (θ escalar): teste escore.	14

Lista de Tabelas

4.1	Tamanho 1 (%).	17
4.2	Tamanho 2 (%).	18
4.3	Tamanho 3 (%).	19
4.4	Poder (%), y segue distribuição BEOI com parâmetro de mistura igual 5%.	20
4.5	Poder (%), y segue distribuição BEOI com parâmetro de mistura igual 10%.	21
4.6	Poder (%), y segue distribuição BEOI com parâmetro de mistura igual 15%.	21
4.7	Poder (%), omissão de variável regressora importante.	22
4.8	Poder (%), variável resposta com outlier.	23
4.9	Poder (%), erro no submodelo da precisão.	24
4.10	Poder (%), gerando y com 5% de poluição log-log complementar.	26
4.11	Poder (%), gerando y com 10% de poluição log-log complementar.	27
4.12	Poder (%), gerando y com 15% de poluição log-log complementar.	28
4.13	Poder (%), gerando y com 20% de poluição log-log complementar.	29
4.14	Poder (%), gerando y com 25% de poluição log-log complementar.	30
4.15	Tamanho 4 (%).	32
4.16	Tamanho 5 (%).	32
4.17	Poder (%), erro no submodelo da precisão - RESET 2.	33
4.18	Poder (%), y segue distribuição BEOI de parâmetro de mistura igual 5% - RESET 2.	34
4.19	Poder (%), y segue distribuição BEOI de parâmetro de mistura igual 10% - RESET 2.	35
4.20	Poder (%), y segue distribuição BEOI de parâmetro de mistura igual 15% - RESET 2.	35
4.21	Poder (%), omissão de variável importante - RESET 2.	36
4.22	Poder (%), gerando y com 5% de poluição log-log complementar - RESET 2.	37
4.23	Poder (%), gerando y com 10% de poluição log-log complementar - RESET 2.	37
4.24	Poder (%), gerando y com 15% de poluição log-log complementar - RESET 2.	38
4.25	Poder (%), gerando y com 20% de poluição log-log complementar - RESET 2.	38
4.26	Poder (%), gerando y com 25% de poluição log-log complementar - RESET 2.	39

Sumário

1	Introdução	1
1.1	Introdução	1
1.2	Organização da Dissertação	1
1.3	Suporte computacional	2
2	Modelo de Regressão Beta	3
2.1	Introdução	3
2.2	Distribuição Beta	3
2.3	Regressão Beta	4
2.4	Distribuição Beta Inflacionada	8
3	Teste de Má Especificação do Modelo de Regressão Beta	10
3.1	Introdução	10
3.2	Teste RESET	10
3.3	Testes Baseados na Função de Verossimilhança	11
4	Avaliação Numérica	15
4.1	Introdução	15
4.2	Tamanho - Estendendo o submodelo da média	16
4.3	Poder - Estendendo o submodelo da média	19
4.3.1	Poder em identificar má especificação da distribuição da variável resposta	19
4.3.2	Poder em identificar omissão de variável importante	22
4.3.3	A influência de outliers	23
4.3.4	Poder em identificar erro no submodelo da precisão	24
4.3.5	Poder em identificar má especificação da função de ligação para o sub-modelo da média	25
4.4	Tamanho - Estendendo os submodelos da média e dispersão	31
4.5	Poder - Estendendo os submodelos da média e dispersão	33
4.5.1	Poder em identificar erro no submodelo da precisão - RESET 2	33
4.5.2	Poder em identificar má especificação da distribuição da variável resposta - RESET 2	34
4.5.3	Poder em identificar omissão de variável importante - RESET 2	36
4.5.4	Poder em identificar má especificação da função de ligação para o sub-modelo da média - RESET 2	36

5	Aplicação	40
5.1	Modelos de Dispersão Constante	40
5.2	Modelos de Dispersão Variável	41
6	Considerações Finais	43
	Referências Bibliográficas	45
A	Otimização Não-Linear	47
B	Programa - Simulação de Tamanho	49
C	Programa - Simulação de Poder	68

CAPÍTULO 1

Introdução

1.1 Introdução

Os modelos de regressão mais utilizados na prática podem não ser apropriados quando a variável resposta assume valores no intervalo $(0, 1)$, como é o caso de taxas, índices e proporções, por exemplo. Nesse contexto, Ferrari & Cribari-Neto (2004) propôs um modelo de grande utilidade, o qual admite que a variável resposta segue lei beta. Para especificação do modelo os autores fazem uso de uma reparametrização da densidade beta (como será melhor explanado no próximo capítulo). Uma extensão desse modelo foi proposta por Simas et al. (2010), onde se utiliza além de um submodelo para a média, um submodelo para a dispersão.

Após a estimação do modelo de interesse, é de fundamental importância verificar se a forma funcional do modelo está corretamente especificada. Lima (2007) propôs um teste de má especificação para modelos de regressão beta, baseado no teste RESET (*regression specification error test*), proposto por Ramsey (1969) para modelos lineares de regressão. A variável de teste foi definida após um estudo de simulação considerando o poder do teste em dois diferentes cenários, a saber: função de ligação incorreta e não linearidade no submodelo da média. Vale salientar que esse estudo foi realizado levando em consideração apenas modelos de regressão beta com dispersão fixa.

Esta dissertação dá sequência ao trabalho iniciado por Lima (2007), tendo como objetivo avaliar o poder do teste de má especificação proposto em diversos cenários que configuram má especificação do modelo postulado, como função de ligação incorreta, presença de *outlier* na amostra, omissão de variável importante, estimação com dispersão constante quando o modelo verdadeiro possui dispersão variável (e vice-versa) e o caso em que há má especificação da distribuição da variável resposta. Adicionalmente, é proposta também uma outra forma para o teste de má especificação (que será detalhada no Capítulo 3), onde se expande tanto o submodelo da média como o submodelo da dispersão. Esta alternativa foi pensada para modelos de regressão beta com dispersão variável, e alguns estudos de simulação foram realizados, considerando os cenários já descritos anteriormente, para avaliar o poder do teste. Para o estudo desta dissertação foram considerados modelos de regressão beta com dispersão fixa e também com dispersão variável.

1.2 Organização da Dissertação

Esta dissertação é composta por seis capítulos, tendo início com esta introdução e uma breve análise sobre os *softwares* utilizados para a elaboração do trabalho e o algoritmo utilizado nas

otimizações da avaliação numérica. O segundo capítulo apresenta uma breve introdução sobre a distribuição beta, a classe de modelos de regressão beta e a distribuição beta inflacionada.

O terceiro capítulo introduz os testes assintóticos auxiliares utilizados no teste de má especificação proposto por Cribari-Neto & Lima (2007) baseado no teste RESET, a saber: teste de razão de verossimilhanças, Wald e escore de Rao. Estas estatísticas de teste são apresentados sob uma interpretação geométrica proposta por Buse (1982).

O quarto capítulo traz os resultados acerca do poder do teste de má especificação em contextos em que o modelo é especificado de forma incorreta, tanto para modelos de regressão beta com dispersão constante quanto para modelos com dispersão variável. Este estudo é feito por meio de simulação de Monte Carlo.

O quinto capítulo apresenta algumas aplicações do teste de má especificação utilizando os dados de destilação de gasolina (Prater, 1956) e os dados referentes à habilidade de leitura em crianças disléxicas e não disléxicas (Smithson & Verkuiler, 2006). Por fim, o sexto capítulo reúne as principais conclusões deste estudo.

1.3 Suporte computacional

Para a escrita desta dissertação foi utilizado o sistema de tipografia $\text{\LaTeX}$, que é um conjunto de macros para $\text{\TeX}$ criado na década de 80 por Leslie Lamport. $\text{\TeX}$ é um programa de edição de textos criado na década de 70 na Stanford University por Donald Knuth. Para simplificar a produção e o gerenciamento do trabalho utilizamos a IDE (Ambiente de Documentação Integrado) TeXnicCenter que possui licença livre e que se integra bem com a distribuição MiKTeX. Por sua vez o MiKTeX fornece as ferramentas necessárias para a utilização da linguagem de programação tipográfica científica $\text{\LaTeX}$. Para tipografia desta dissertação, utilizou-se a classe UFPETHesis versão 0.9.2 criada por Paulo G. S. da Fonseca e está disponível em <http://www.cin.ufpe.br/~paguso/ufpethesis/>.

Os programas para simulações foram feitos na linguagem de programação Ox (versão 6.20) desenvolvida por Jurgen Doornik. Ox é uma linguagem de programação matricial orientada a objetos com uma extensa biblioteca de funções matemáticas e estatísticas. Foi utilizada a versão console, que possui licença livre para uso acadêmico e está disponível em <http://www.doornik.com/>. A escrita do programa foi feita no OxEEdit (versão 6.20), que é um editor de texto voltado para a programação em Ox.

Modelo de Regressão Beta

2.1 Introdução

É de grande utilidade para as mais diversas áreas de conhecimento a modelagem de dados que assumem valores no intervalo $(0, 1)$, como, por exemplo, taxas e proporções. Muitos investigadores recorrem a transformações para que os mesmos possam ser mapeados nos reais. Porém, esse procedimento possui a limitação de perda de interpretabilidade dos parâmetros da regressão em termos da média da variável resposta original. Nesse contexto, Ferrari & Cribari-Neto (2004) propôs o modelo de regressão em que a variável resposta segue distribuição beta, que foi denominado de regressão beta. No decorrer deste capítulo o modelo de regressão beta será detalhado após uma breve explanação acerca da distribuição beta. Finalizando, será introduzida a distribuição beta inflacionada, que tem grande utilidade para modelagem de dados nos intervalos $[0, 1)$, $(0, 1]$ e $[0, 1]$.

2.2 Distribuição Beta

Seja Y uma variável aleatória que segue lei beta, ou seja $Y \sim B(p, q)$. Sua função de densidade tem a seguinte forma:

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

em que $0 < y < 1$, $p > 0$, $q > 0$ e $\Gamma(\cdot)$ é a função gama:

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt. \quad (2.2)$$

Sua função de distribuição é dada por

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

o valor esperado de Y é

$$\mathbb{E}(Y) = \frac{p}{p+q} \quad (2.3)$$

e sua variância é

$$\text{Var}(Y) = \frac{pq}{(p+q)^2(p+q+1)}. \quad (2.4)$$

A densidade beta pode assumir algumas formas de acordo com os valores assumidos por seus parâmetros p e q . Essas formas podem ser observadas na figura a seguir, onde vale destacar alguns padrões de comportamento. Por exemplo, quando $p = q = 1$ obtém-se a distribuição uniforme padrão, se $p > q$, existe assimetria à esquerda e, de forma similar, no caso em que $q > p$ existe assimetria à direita. Por fim, quando $p = q$ a função de densidade é simétrica.

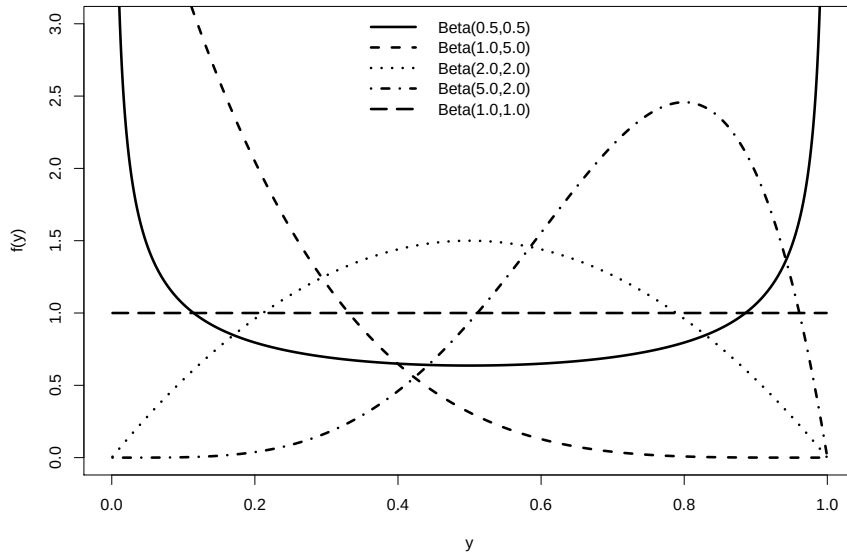


Figura 2.1 Densidades beta para diferentes valores de p e q .

2.3 Regressão Beta

O modelo de regressão beta foi introduzido por Ferrari & Cribari-Neto (2004) utilizando uma reparametrização para a função de densidade beta, a fim de se obter uma estrutura de regressão capaz de modelar a média da resposta juntamente com um parâmetro de precisão. Fazendo $p = \mu\phi$ e $q = (1 - \mu)\phi$, temos que $\mu = p/(p + q)$ e $\phi = p + q$. Substituindo p e q em (2.3) e (2.4), respectivamente, tem-se que

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

e

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

em que

$$V(\mu) = \mu(1 - \mu).$$

Dessa forma, μ pode ser interpretado como a média de Y e ϕ como um parâmetro de precisão, já que para um dado μ , à medida que ϕ aumenta, a variância de Y diminui.

Assim, tem-se a seguinte reparametrização de (2.1),

$$f(y; \mu, \phi) = \frac{\Gamma(\phi)}{\Gamma(\mu\phi)\Gamma((1-\mu)\phi)} y^{\mu\phi-1} (1-y)^{(1-\mu)\phi-1}, \quad (2.5)$$

em que y e $\mu \in (0, 1)$ e $\phi > 0$.

Sejam $y_1, \dots, y_n$ variáveis aleatórias independentes tais que $y_t \sim B(\mu_t, \phi), t = 1, \dots, n$. O modelo de regressão beta é definido igualando-se uma função da média μ ao preditor linear η_t :

$$g(\mu_t) = \sum_{i=1}^k x_{ti} \beta_i = x_t^\top \beta = \eta_t, \quad (2.6)$$

em que $\beta = (\beta_1, \dots, \beta_k)^\top$ é o vetor de parâmetros de regressão desconhecidos, $\beta \in \mathbb{R}^k$, e $x_t^\top = (x_{t1}, \dots, x_{tk})$ são as observações de k variáveis explicativas ($k < n$), as quais possuem valores conhecidos e fixos. A função $g(\cdot)$ é conhecida como função de ligação. Tal função é estritamente monótona, duas vezes diferenciável e mapeia $(0, 1)$ em $\mathbb{R}$. Requeremos que a função de ligação seja duas vezes diferenciável para que possamos obter a matriz de informação de Fisher. Existem diversas possibilidades para $g(\cdot)$, como, por exemplo,

- função logit:

$$g(\mu) = \log \left(\frac{\mu}{1-\mu} \right);$$

- função probit:

$$g(\mu) = \Phi^{-1}(\mu),$$

em que Φ é a função de distribuição acumulada de uma variável aleatória com distribuição normal padrão;

- função log-log:

$$g(\mu) = -\log[-\log(\mu)];$$

- função complementar log-log:

$$g(\mu) = -\log[-\log(1-\mu)];$$

- função Cauchy:

$$g(\mu) = \tan[\pi(\mu - 0.5)].$$

A partir de (2.6) tem-se que $\mu_t = g^{-1}(\eta_t)$ e assim

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

Note que isso implica que a variância de y_t não é constante ao longo das observações, ou seja, o modelo de regressão beta é heteroscedástico por natureza.

Simas et al. (2010) propôs uma extensão ao modelo desenvolvido por Ferrari & Cribari-Neto (2004) motivado pelo fato de que em determinadas situações o parâmetro de dispersão pode não ser constante para todas as observações, de forma que se faz necessário modelá-lo também. Assim como na modelagem da média, admite-se que $y_1, \dots, y_n$ sejam variáveis aleatórias independentes tais que $y_t \sim B(\mu_t, \phi_t), t = 1, \dots, n$, e

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

em que $\gamma = (\gamma_1, \dots, \gamma_q)$ é um vetor de coeficientes de regressão, $k + q < n$ e $\gamma \in \mathbb{R}^q$. Aqui, $z_t^\top = z_{t1}, \dots, z_{tq}$ são observações de q variáveis explicativas, as quais possuem valores conhecidos e fixos. Aqui, $h(\cdot)$ é uma função de ligação tal que $h(\cdot) : (0, \infty) \mapsto \mathbb{R}$, sendo estritamente monótona e duas vezes diferenciável. Requeremos que a função de ligação seja duas vezes diferenciável para que possamos obter a matriz de informação de Fisher. As principais funções de ligação utilizadas nesse caso são:

- logaritmo:

$$h(\phi_t) = \log(\phi_t); \text{ e}$$

- raiz quadrada:

$$h(\phi_t) = \sqrt{\phi_t}.$$

Os parâmetros do modelo de regressão beta podem ser estimados por máxima verossimilhança. A função de log-verossimilhança para uma amostra composta por n observações independentes tem a seguinte forma:

$$l(\beta, \gamma) = \sum_{t=1}^n l_t(\mu_t, \phi_t),$$

em que

$$l_t(\mu_t, \phi_t) = \log(\Gamma(\phi_t)) - \log(\Gamma(\mu_t \phi_t)) - \log(\Gamma((1 - \mu_t) \phi_t)) + (\mu_t - 1) \log(y_t) \\ + \{(1 - \mu_t) \phi_t - 1\} \log(1 - y_t).$$

Vale salientar que $\mu_t = g^{-1}(\eta_t)$ é uma função de β . Por não possuírem forma fechada, os estimadores de β e γ são obtidos por meio de maximização numérica através de algum algoritmo de otimização não-linear, como por exemplo o algoritmo de Newton ou um algoritmo quasi-Newton.

Sejam $y_t^* = \log(y_t / (1 - y_t))$ e $\mu_t^* = \psi(\mu_t \phi_t) - \psi((1 - \mu_t) \phi_t)$, onde ψ é a função digama dada pela primeira derivada do logaritmo natural de (2.2). Diferenciando a função de log-verossimilhança com respeito aos parâmetros desconhecidos, obtém-se a função escore:

$$U = (U_\beta(\beta, \gamma), U_\gamma(\beta, \gamma))^\top.$$

A função escore de β tem a seguinte forma:

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

em que X é uma matriz $n \times k$, $x_t^\top$ é a t -ésima linha da matriz X , $\Phi = \text{diag}\{\phi_1, \dots, \phi_n\}$, $T = \text{diag}\{1/g'(\mu_1), \dots, 1/g'(\mu_n)\}$, $y^* = (y_1^*, \dots, y_n^*)^\top$ e $\mu^* = (\mu_1^*, \dots, \mu_n^*)^\top$. A função escore de γ , por sua vez é dada por

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

em que Z é uma matriz $n \times q$, $z_t^\top$ é a t -ésima linha da matriz Z , $H = \text{diag}\{1/h'(\phi_1), \dots, 1/h'(\phi_n)\}$ e $a = (a_1, \dots, a_n)^\top$ onde $a_t = \mu_t(y_t^* - \mu_t^*) + \log(1 - y_t) - \psi((1 - \mu_t)\phi_t) + \psi(\phi_t)$. Vale salientar que para o modelo com dispersão fixa, $U_\gamma(\beta, \gamma)$ é um escalar.

A matriz de informação observada, obtida calculando-se a segunda derivada da função de log-verossimilhança em relação aos parâmetros desconhecidos, tem a seguinte forma

$$J = J(\beta, \gamma) = \begin{pmatrix} J_{\beta\beta} & J_{\beta\gamma} \\ J_{\gamma\beta} & J_{\gamma\gamma} \end{pmatrix},$$

em que

$$J_{\beta\beta} = X^\top \Phi Q X,$$

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

$$J_{\gamma\gamma} = Z^\top V Z,$$

em que $Q = \text{diag}\{q_1, \dots, q_n\}$, sendo

$$q_t = \left\{ \phi_t [\psi'(\mu_t \phi_t) + \psi'((1 - \mu_t)\phi_t)] + (y_t^* - \mu_t^*) \frac{g''(\mu_t)}{g'(\mu_t)} \right\} \frac{1}{[g'(\mu_t)]^2}.$$

Aqui, $\psi'(\cdot)$ é a função trigama dada pela segunda derivada do logaritmo natural de (2.2), $F = \text{diag}\{f_1, \dots, f_n\}$, com $f_t = c_t - (y_t^* - \mu_t^*)$ e

$$c_t = \phi_t [\psi'(\mu_t \phi_t) \mu_t - \psi'((1 - \mu_t)\phi_t)(1 - \mu_t)].$$

No caso do modelo com dispersão fixa,

$$c_t = \phi [\psi'(\mu_t \phi) \mu_t - \psi'((1 - \mu_t)\phi)(1 - \mu_t)].$$

Adicionalmente, $V = \text{diag}\{v_1, \dots, v_n\}$ onde, $v_t = d_t + a_t h''(\phi_t)/[h'(\phi_t)]^3$ e

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

Sob dispersão fixa,

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

Por fim, obtém-se a matriz de informação de Fisher para (β, γ) aplicando a esperança em J , sendo dada por

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

em que

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

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

e

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

de forma que

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

ou, alternativamente para modelo com dispersão constante, o parâmetro ϕ_t é substituído por ϕ e tem-se

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

Vale salientar que os parâmetros β e γ não são ortogonais, diferentemente do que ocorre na classe de modelos lineares generalizados (McCullagh & Nelder, 1989).

Estando satisfeitas as condições de regularidade, os estimadores de máxima verossimilhança têm distribuição assintótica conjunta dada por

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

As condições de regularidade para este resultado podem ser encontrados em Serfling (1980, Seção 4.2.2), onde também pode ser encontrada uma prova do resultado.

2.4 Distribuição Beta Inflacionada

Ao se estudar variáveis que se apresentam em forma de frações ou taxas, muitas vezes o conjunto de dados pode conter 1 ou 0. Nesse caso, a regressão beta não pode ser usada, pois ela requer que a resposta assuma valores em $(0, 1)$. Tendo esse problema em vista, Ospina e Ferrari (2010) propuseram três distribuições que cáptam a probabilidade massa para os casos onde o intervalo dos dados contém 0, 1 ou ambos os valores.

Para dados pertencentes aos intervalos $[0, 1)$ ou $(0, 1]$ utiliza-se uma mistura da distribuição beta e uma distribuição degenerada que atribui probabilidade para 0 ou 1, dependendo do caso de interesse. Já para variáveis aleatórias que pertencem ao intervalo $[0, 1]$, foi proposta uma mistura de distribuição Bernoulli com distribuição beta. Estes modelos são casos particulares da classe de modelos inflacionados, o que justifica o nome distribuição beta inflacionada. A seguir fornecemos uma explanação acerca dos dois primeiros modelos anteriormente citados tendo em vista que um deles é utilizado em um dos experimentos da seção Avaliação Numérica.

Admitindo-se uma distribuição degenerada em determinado valor conhecido a , qual pode assumir o valor 0 ou 1, de acordo com o caso de interesse. A mistura dessa distribuição degenerada com a distribuição beta resulta na distribuição beta inflacionada-um ou distribuição beta inflacionada-zero, dependendo do valor de a . A função de distribuição acumulada é dada por

$$BI_a(y; \alpha, \mu, \phi) = \alpha \mathbf{I}_{[a,1]}(y) + (1 - \alpha)F(y; \mu, \phi),$$

onde α é o parâmetro de mistura, assumindo valores maiores que zero e menores que um, e também a probabilidade massa de a , $\mathbf{I}_A$ é uma função indicadora qual assume 1 se $y \in A$ e 0 caso contrario e $F(y; \mu, \phi)$ é a função de distribuição acumulada da distribuição beta com parâmetros μ e ϕ .

$$bi_a(y; \alpha, \mu, \phi) = \begin{cases} \alpha, & \text{se } y = a, \\ (1 - \alpha)f(y; \mu, \phi), & \text{se } y \in (0, 1), \end{cases} \quad (2.7)$$

onde $f(y; \mu, \phi)$ é a densidade da distribuição beta descrita em (2.5).

Ospina e Ferrari (2010) denomina (2.7) como *zero-inflated beta distribution* (BEZI), denotado por $y \sim \text{BEZI}(\alpha, \mu, \phi)$, no caso em que a assume o valor 0. Quando $a = 1$, (2.7) é denominado *one-inflated beta distribution* (BEOI) e escreve-se $y \sim \text{BEOI}(\alpha, \mu, \phi)$. A esperança e a variância de y têm as seguintes formas:

$$\mathbb{E}(y) = \alpha a + (1 - \alpha)\mu,$$

$$\text{Var}(y) = (1 - \alpha) \frac{V(\mu)}{\phi + 1} + \alpha(1 - \alpha)(a - \mu)^2.$$

Note que a $\mathbb{E}(y)$ é a média ponderada da esperança da distribuição beta (2.5) e a esperança da distribuição degenerada.

Teste de Má Especificação do Modelo de Regressão Beta

3.1 Introdução

Testes de hipóteses são de fundamental importância para se ter segurança acerca dos resultados obtidos por meio de estimação pontual. Através da formulação de hipóteses pode-se verificar se os dados amostrais dão suporte para se rejeitar ou não algumas suspeitas sobre determinados parâmetros. As estatísticas de teste mais utilizadas em geral são as estatísticas da razão de verossimilhanças, Wald e Escore de Rao, as quais sob a hipótese nula e atendidas certas condições de regularidade têm a mesma distribuição assintótica, a saber: χ^2 . No contexto de modelos de regressão beta, Cribari-Neto & Lima (2007) desenvolveu um teste para detectar má especificação do modelo postulado a partir do teste RESET proposto por Ramsey em 1969.

No decorrer deste capítulo, primeiramente será introduzido o teste RESET e sua versão para modelos de regressão beta, e posteriormente serão introduzidos os testes secundários (razão de verossimilhanças, Wald e escore) necessários para a identificação ou não de má especificação.

3.2 Teste RESET

O teste RESET (*“Regression Equation Specification Error Test”*) foi proposto por Ramsey (1969) no contexto de modelos de regressão linear. RESET é um teste geral para identificar dois tipos de erros de especificação do modelo, a saber: variável omitida e forma funcional incorreta. O teste se baseia na comparação das distribuições dos resíduos sob a hipótese nula de que o modelo está corretamente especificado e sob a hipótese alternativa de que o modelo está mal especificado de alguma forma. Para detalhes sobre o teste RESET, ver Davidson & MacKinnon (1993, Seção 6.5).

Baseados no teste RESET de Ramsey, Cribari-Neto & Lima (2007) desenvolveu um teste de erro de especificação para modelos de regressão beta com dispersão fixa. Para uma breve explanação acerca da metodologia do teste, considere o seguinte modelo de regressão beta (em forma matricial)

$$g(\mu) = X\beta, \quad (3.1)$$

onde $g(\cdot)$ é uma função de ligação, μ é o vetor da média da variável resposta, X é uma matriz $n \times k$ e β é um vetor de parâmetros desconhecidos. Considere também o seguinte submodelo ampliado:

$$g(\mu) = X\beta + Z\theta, \quad (3.2)$$

onde Z é uma matriz $n \times s$ de variáveis de teste e θ é um vetor $s \times 1$ de parâmetros.

O procedimento consiste em testar a hipótese nula $H_0 : \theta = 0$, de que o modelo está corretamente especificado, contra a hipótese alternativa, $H_1 : \theta \neq 0$, de que existe má especificação do modelo, onde 0 denota um vetor $s \times 1$ de zeros. Para testar a exclusão de θ aplica-se o teste da razão de verossimilhanças, escore ou Wald, os quais serão detalhados adiante. Cribari-Neto & Lima (2007) realizou simulação de Monte Carlo para avaliar o poder do teste sob dois erros de especificação do modelo, a saber: função de ligação incorreta e não linearidade do submodelo da média, sugere utilizar como variável de teste o preditor linear (do submodelo da média) ajustado elevado ao quadrado ($\hat{\eta}^2$), ou seja, $Z = \hat{\eta}^2$. Note que nesse caso o parâmetro θ é escalar.

Em sua dissertação de mestrado, Lima (2007) utilizou esta forma do teste RESET para investigar o poder do teste em modelos de regressão beta com dispersão fixa em dois cenários de má especificação. Nesta dissertação é utilizada esta forma do teste RESET, também para investigar poder do teste, no entanto, considerando tanto o modelo de regressão beta com dispersão fixa quando o modelo de regressão beta com dispersão variável e levando-se em consideração cinco cenários de má especificação. E, adicionalmente, utiliza-se uma nova proposta para o teste RESET, que pode ser utilizada em modelos de regressão beta com dispersão variável. Considere um modelo de regressão beta onde o submodelo da média é dado por (3.1) e o submodelo da dispersão, em forma matricial, é

$$h(\phi) = W\gamma, \quad (3.3)$$

onde $h(\cdot)$ é uma função de ligação, ϕ é um vetor $n \times 1$, W é uma matriz $n \times k$ igual a X e γ é um vetor de parâmetros $k \times 1$.

A metodologia do teste consiste em adicionar o preditor linear ajustado de (3.1) elevado ao quadrado ($\hat{\eta}^2$) ao próprio submodelo (3.1), como no teste proposto por Cribari-Neto & Lima (2007) e adicionar também o preditor linear ajustado de (3.3) elevado ao quadrado ($\hat{\delta}^2$) ao próprio submodelo de (3.3), de forma que o modelo estendido é dado por (3.2) e

$$h(\phi) = W\gamma + Y\sigma,$$

onde Y é o vetor ($\hat{\delta}^2$) $s \times 1$ e σ é o parâmetro associado. Adicionadas as variáveis de teste, deve-se testar a hipótese nula de que o modelo está corretamente especificado, $H_0 : \theta = \sigma = 0$, contra a hipótese alternativa que o modelo está erroneamente especificado, $H_1 : \theta$ e/ou $\sigma \neq 0$.

Diferentemente de modelos normais de regressão linear, onde a exclusão de uma variável pode ser testada de forma exata, para a classe de regressão beta deve-se utilizar testes assintóticos.

3.3 Testes Baseados na Função de Verossimilhança

Os testes baseados na função de verossimilhança são amplamente empregados na avaliação de modelos estatísticos, já que a função de verossimilhança, sob o paradigma paramétrico, contém toda informação necessária para inferências sobre determinados parâmetros de interesse.

A função de verossimilhança “informa a ordem natural de preferência entre diversas possibilidades de θ (...) Generalizando, entre os possíveis candidatos para estimar o parâmetro verdadeiro (...) a partir dos mesmos dados y , o vetor de parâmetros mais plausíveis é aquele de maior verossimilhança” (Cordeiro, 1999, p. 5).

Considere um espaço paramétrico denotado por Θ e suponha que o interesse reside no teste da hipótese $H_0 : \theta = \theta^{(0)}$ contra $H_1 : \theta \neq \theta^{(0)}$, onde $\theta^{(0)}$ pertence ao espaço paramétrico restrito denotado por Θ_0 , que por sua vez está contido em Θ . A razão de verossimilhanças é

$$\frac{L(\hat{\theta}; y)}{L(\theta^{(0)}; y)}.$$

Ela quantifica evidências estatísticas que levam à rejeição de H_0 . A estatística da razão de verossimilhanças pode ser escrita como

$$RV = 2[l(\hat{\theta}) - l(\theta^{(0)})],$$

onde $l(\theta)$ é o logaritmo da função de verossimilhança relativo ao vetor de parâmetros θ e $\hat{\theta}$ é o estimador de máxima verossimilhança de θ . Sob a hipótese nula e atendidas algumas condições de regularidade, RV possui distribuição assintótica χ_r^2 , onde r é o número de restrições a serem testadas. Além do teste da razão de verossimilhanças, outros dois testes são também largamente utilizados na avaliação de modelos estatísticos: o teste Wald, o qual leva o nome de seu criador, e o teste escore, proposto por C. R. Rao em 1947. Suas estatísticas são dadas, respectivamente, por

$$W = (\hat{\theta} - \theta^{(0)})^\top K(\hat{\theta})(\hat{\theta} - \theta^{(0)})$$

e

$$S = U^\top(\theta^{(0)})K^{-1}(\theta^{(0)})U(\theta^{(0)}).$$

Assim como o teste de razão de verossimilhanças, sob a hipótese nula e atendidas certas condições de regularidade, tanto W quanto S tem distribuição assintótica χ_r^2 , onde r é o número de restrições em teste. Assim as estatísticas da razão de verossimilhanças, Wald e escore são assintoticamente equivalentes até a primeira ordem. As condições de regularidade para a convergência das distribuições nulas das três estatísticas de teste para a distribuição χ_r^2 podem ser encontradas em Serfling (1980, Seção 4.4), onde também pode ser encontrada uma prova do resultado.

Buse (1982) apresenta uma interpretação geométrica destas estatísticas considerando o caso particular onde θ é escalar. Esta interpretação é bastante interessante para uma melhor compreensão dos testes e pode ser visualizada na Figura 3.1. Note que $RV/2$ depende de $C(\hat{\theta})$, que é a curvatura do logaritmo da função de verossimilhança e é definida como o valor absoluto de $d^2 \log L / d\theta^2$ avaliado em $\theta = \hat{\theta}$, dependendo também da diferença $\hat{\theta} - \theta^{(0)}$, de forma que, tomando essa diferença como constante, $RV/2$ cresce à medida que a curvatura aumenta.

Diferentemente do teste de razão de verossimilhanças, o teste Wald leva em consideração apenas a distância entre $\hat{\theta}$ e $\theta^{(0)}$. A Figura 3.2 mostra que diferentes conjuntos de dados podem conduzir aos mesmos valores de $(\hat{\theta} - \theta^{(0)})^2$. Como algum dos conjuntos de dados se apresenta menos favorável à hipótese nula, por conta da diferença de curvatura de sua função de log-verossimilhança, faz-se necessária uma ponderação. O usual é que essa ponderação seja feita

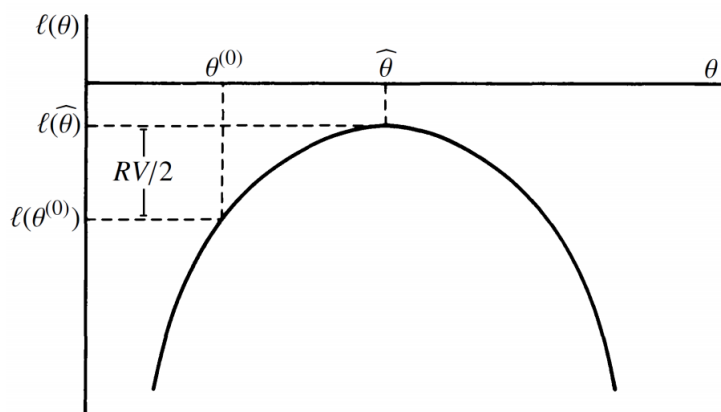


Figura 3.1 Funções de log-verossimilhança (θ escalar): teste da razão de verossimilhanças.

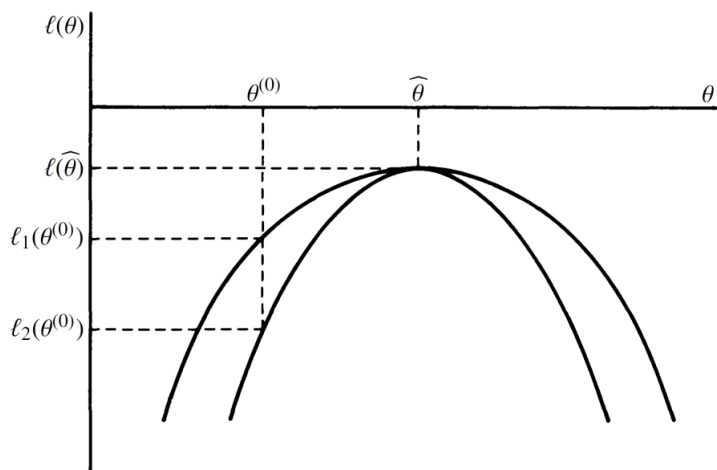


Figura 3.2 Funções de log-verossimilhança (θ escalar): teste Wald.

em termos da curvatura média $K(\hat{\theta})$, onde $K(\theta)$ é a matriz de informação esperada dada por $-\mathbb{E}(d^2 \log L / d\theta^2)$.

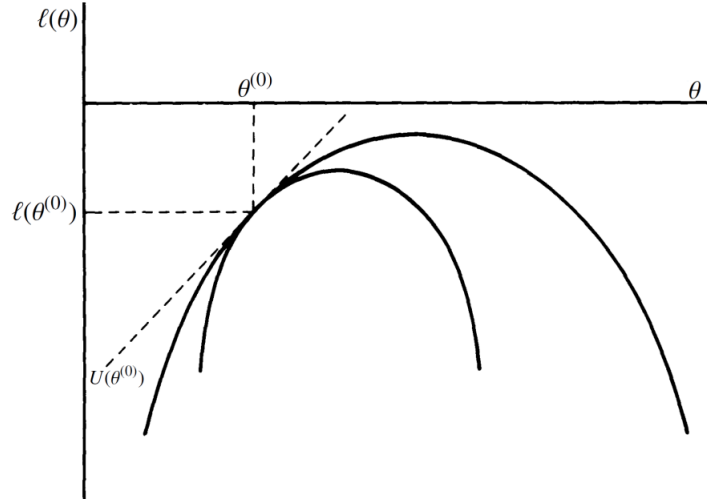


Figura 3.3 Funções de log-verossimilhança (θ escalar): teste escore.

Diferentemente do teste da razão de verossimilhanças, o qual leva em consideração as hipóteses nula e alternativa, e do teste Wald, que leva em consideração apenas a hipótese alternativa, o teste escore leva em consideração apenas a hipótese nula. Note que sempre que a hipótese nula for verdadeira, a diferença entre as estimativas restrita e irrestrita será bastante pequena e se pode utilizar $U(\theta^{(0)})$ elevado ao quadrado como uma medida de distanciamento entre $\hat{\theta}$ e $\theta^{(0)}$, em que $U(\theta)$ é a função escore relativa ao parâmetro θ e pode ser obtida através de $d \log L / d\theta$. Assim como foi descrito na construção da estatística Wald, diferentes conjuntos de dados podem gerar o mesmo valor para função escore, no entanto, como pode ser observado na Figura 3.3, a log-verossimilhança restrita de um deles será mais próxima do máximo, assim, faz-se necessário um termo de ponderação, que pode ser a curvatura $C(\hat{\theta})$. Diferentemente do que ocorre no teste de razão de verossimilhanças, à medida em que a curvatura aumenta a distância $\hat{\theta} - \theta^{(0)}$ diminui. Assim, utiliza-se o inverso da curvatura e a estatística escore pode ser escrita da seguinte forma

$$S = (U(\theta^{(0)}))^2 C^{-1}(\theta^{(0)}).$$

No entanto, na prática se substitui $C^{-1}(\theta^{(0)})$ por $K^{-1}(\theta^{(0)})$.

De forma geral, tem-se que a estatística de razão de verossimilhanças leva em consideração a diferença vertical de $\hat{\theta}$ e $\theta^{(0)}$, a estatística Wald fundamenta-se na diferença horizontal e a estatística escore baseia-se na inclinação da função de log-verossimilhança avaliada em $\theta^{(0)}$.

Expressões fechadas para estas estatísticas podem ser obtidas em Ferrari & Cribari-Neto (2004) para modelos de regressão beta com dispersão fixa, e em Simas et al. (2010) para modelos de regressão beta com dispersão variável.

CAPÍTULO 4

Avaliação Numérica

4.1 Introdução

Neste trabalho avaliou-se os desempenhos, em amostras de tamanho finito, de testes de especificação correta do modelo de regressão beta realizados com base nos testes da razão de verossimilhanças (RV), Wald e escore. Utilizamos dois modelos de regressão beta, um com dispersão fixa e o outro com dispersão variável. A especificação da média dos dois modelos é

$$\log\left(\frac{\mu_t}{1-\mu_t}\right) = \sum_{i=1}^3 \beta_i x_{ti},$$

$t = 1, \dots, n$. A especificação do parâmetro de precisão (para o modelo com dispersão variável) é

$$\log(\phi_t) = \sum_{i=1}^3 \gamma_i z_{ti},$$

$t = 1, \dots, n$.

Foram utilizados diversos tamanhos de amostra, de 20 até 3000, de acordo com o estudo de interesse. Para compor estas amostras é gerada, inicialmente e de forma aleatória, uma amostra de 20 observações dos regressores, a qual é replicada diversas vezes a fim de se conseguir compor os demais tamanhos amostrais. Isso é feito com o intuito de manter constante o grau de heterogeneidade da dispersão, medido por $\lambda = \max(\phi_t)/\min(\phi_t)$. Em todas as simulações com dispersão variável, $\lambda \approx 80$. Em todos os casos, usamos $x_{ti} = z_{ti}$, $\forall i$ e $\forall t$. De forma geral, usamos $\beta_1 = -1.8$, $\beta_2 = 1.4$, $\beta_3 = 1.6$ para o submodelo da média, tanto nos modelos com dispersão constante quanto nos modelos com dispersão variável. Para os modelos com dispersão fixa, usamos $\phi = 120$ e, sob dispersão variável, usamos $\gamma_1 = 1.5$, $\gamma_2 = 3.0$, $\gamma_3 = 2.5$. Nos casos em que foram utilizados outros valores para os parâmetros, estes serão especificados posteriormente.

As estimativas de máxima verossimilhança são obtidas por meio da maximização numérica da função de log-verossimilhança utilizando o método de otimização não-linear BFGS com primeiras derivadas analíticas. Os valores iniciais dos parâmetros no processo de otimização são escolhidos conforme sugere Cribari-Neto & Zeileis (2010). O valor inicial de β no processo iterativo é obtido regressando $g(y_t)$ sobre os regressores do submodelo da média. Já o valor inicial de ϕ no processo iterativo de maximização com dispersão fixa é

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

em que $\check{\mu}_t = g^{-1}(x_t^\top (X^\top X)^{-1} X^\top z)$, $z = (g(y_1), \dots, g(y_n))^\top$, $\check{\sigma}_t^2 = \check{\epsilon}^\top \check{\epsilon} / \{(n-k)[g'(\check{\mu}_t)]^2\}$ e $\check{\epsilon} = z - X(X^\top X)^{-1} X^\top z$. Aqui, $x_t^\top$ denota a t -ésima linha da matriz X . Por fim, os valores iniciais dos parâmetros do submodelo da dispersão no modelo com dispersão variável são obtidos da seguinte forma: para γ_1 , utiliza-se o mesmo valor inicial de ϕ no modelo com dispersão constante, aplicando-se a ele a função de ligação, e os demais parâmetros são inicializados em zero.

Nos experimentos, foram utilizados os níveis nominais $\alpha = 1\%$, 5% e 10% para cada um dos testes. Todos os resultados de simulação são baseados em 10 mil réplicas de Monte Carlo (o que proporciona uma precisão da ordem de uma casa decimal), com exceção das simulações feitas com $n = 3000$; neste caso, utilizou-se 500 réplicas de Monte Carlo devido ao fato de tais simulações serem muito intensivas computacionalmente. A geração dos valores da variável resposta foi feita a partir da distribuição beta de parâmetros μ_t e ϕ , para o modelo com dispersão fixa, e μ_t e ϕ_t para o modelo com dispersão variável.

Na Seção 4.2, são apresentados os resultados das simulações de tamanho para o teste proposto por Cribari-Neto & Lima (2007). Na Seção 4.3, são apresentados os procedimentos dos experimentos para as simulações de poder, bem como os respectivos resultados, utilizando o teste de má especificação proposto por Cribari-Neto & Lima (2007), o qual consiste em estender o submodelo da média. Posteriormente, na Seção 4.4 são apresentados os resultados das simulações de tamanho para o novo teste de má especificação, proposto nesta dissertação, que consiste em expandir simultaneamente os submodelos da média e da dispersão. Na Seção 4.5, são apresentados os resultados dos mesmos experimentos descritos na Seção 4.3, no entanto, utilizando outra forma para o teste de má especificação, a qual está detalhada no Capítulo 3. Como esta forma alternativa consiste em estender os submodelos da média e da dispersão, em todos os experimentos utilizamos apenas modelos de regressão beta com dispersão variável. Os resultados obtidos com essa nova forma de teste de má especificação são comparados com os obtidos com o teste proposto por Cribari-Neto & Lima (2007). Em todas as simulações de tamanho do teste utilizamos valores críticos assintóticos; já nas simulações de poder, foram utilizados valores críticos exatos, obtidos a partir das simulações de tamanho do teste, a fim de que se possa comparar os poderes de testes de mesmo tamanho.

4.2 Tamanho - Estendendo o submodelo da média

Nas simulações de tamanho, a variável resposta é gerada e modelada sem erro de especificação do modelo. Após a estimação dos parâmetros aplica-se o teste de má especificação proposto por Cribari-Neto & Lima (2007), utilizando-se valores críticos assintóticos. Os valores da variável resposta foram gerados de acordo com o seguinte submodelo da média:

$$g(\mu_t) = -1.8 + 1.4x_{t2} + 1.6x_{t3},$$

em que $g(\cdot)$ é a função de ligação logit e $\phi = 120$ para o modelo de regressão beta com dispersão fixa. Para o modelo com dispersão variável, utilizou-se o mesmo submodelo da média e

$$\log(\phi_t) = 1.5 + 3z_{t2} + 2.5z_{t3},$$

em que $x_{ti} = z_{ti}$ para $i = 1, 2$ e $\forall t$.

Esta simulação de tamanho será denominada ‘Tamanho 1’. Adiante serão apresentas simulações de poder, as quais utilizam valores críticos estimados exatos obtidos a partir desta simulação de tamanho.

Tabela 4.1 Tamanho 1 (%).

Teste	disp. fixa			n	Teste	disp. variável		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	2.31	8.04	14.75	20	RV	4.51	12.28	19.44
Wald	3.44	9.58	16.09		Wald	10.57	19.32	26.65
Escore	1.33	6.58	13.15		Escore	1.10	6.25	11.77
RV	1.40	6.42	12.15	40	RV	1.91	7.67	14.15
Wald	1.82	7.12	12.79		Wald	3.29	10.08	16.25
Escore	1.05	5.61	11.48		Escore	1.06	5.71	11.56
RV	1.27	5.82	11.15	60	RV	1.46	6.48	12.35
Wald	1.52	6.25	11.67		Wald	2.07	7.66	13.96
Escore	0.97	5.36	10.68		Escore	0.95	5.28	10.85
RV	1.42	5.83	11.03	80	RV	1.34	6.35	11.80
Wald	1.62	6.21	11.33		Wald	1.74	7.22	12.65
Score	1.21	5.50	10.68		Escore	0.98	5.49	10.81
RV	1.24	5.45	10.46	100	RV	1.39	6.04	11.36
Wald	1.32	5.68	10.64		Wald	1.76	6.68	12.11
Escore	1.11	5.25	10.34		Escore	1.10	5.35	10.72

As taxas de rejeição nulas estão apresentadas na Tabela 4.1. Note que à medida em que n aumenta as taxas de rejeição convergem para os níveis nominais, como esperado. Note ainda que essa convergência se dá mais rapidamente sob dispersão fixa.

Na próxima seção investigaremos até que ponto os desempenhos (em termos de poder) dos três testes são afetados pela presença de um *outlier* (observação atípica) nos dados e nas simulações de poder serão utilizados os valores críticos estimados exatos obtidos a partir desta simulação de tamanho. Aqui a variável resposta foi gerada utilizando o seguinte submodelo para a média:

$$g(\mu_t) = -2.5 + 0.8x_{t2} + 1.5x_{t3},$$

onde $g(\cdot)$ é a função de ligação logit e $\phi = 120$ para o modelo de regressão beta com dispersão fixa. Para o modelo com dispersão variável, utilizou-se o mesmo submodelo para a média juntamente com o seguinte submodelo para precisão:

$$\log(\phi_t) = 1.5 + 3z_{t2} + 2.5z_{t3},$$

onde $x_{ti} = z_{ti}$ para $i = 1, 2$ e $\forall t$. Esta simulação de tamanho será denominada ‘Tamanho 2’.

As taxas de rejeição dessa simulação de tamanho estão apresentadas Tabela 4.2. Note que, assim como no experimento anterior, as taxas de rejeição do teste de má especificação

Tabela 4.2 Tamanho 2 (%).

Teste	disp. fixa			n	Teste	disp. variável		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	2.09	8.08	14.69	20	RV	4.00	11.87	18.81
Wald	3.20	9.61	16.03		Wald	9.22	18.22	25.27
Escore	1.02	6.59	13.12		Escore	1.07	5.94	12.25
RV	1.47	6.61	12.27	40	RV	1.64	6.90	12.79
Wald	1.92	7.21	13.01		Wald	2.69	8.79	14.94
Escore	1.14	6.09	11.58		Escore	0.94	5.18	10.50
RV	1.20	5.74	11.29	60	RV	1.30	6.28	11.93
Wald	1.49	6.12	11.56		Wald	1.94	7.32	13.20
Escore	1.04	5.38	10.88		Escore	0.87	5.11	10.61
RV	1.36	5.49	10.58	80	RV	1.40	6.12	11.20
Wald	1.58	5.81	10.78		Wald	1.68	6.83	12.15
Escore	1.19	5.29	10.29		Escore	1.09	5.29	10.38
RV	1.26	5.77	11.21	100	RV	1.26	5.66	10.95
Wald	1.45	6.03	11.32		Wald	1.42	6.27	11.69
Escore	1.09	5.52	10.94		Escore	0.97	4.98	10.20

convergem para os níveis nominais à medida em que n aumenta. No entanto, aqui as taxas de rejeição convergem praticamente na mesma velocidade quando o teste é utilizado em modelos com dispersão fixa ou variável.

Por fim, será realizada uma simulação de tamanho adicional, a partir da qual serão obtidos os valores críticos estimados para uma das simulações de poder da próxima seção. Aqui, a variável repostada foi gerada utilizando o seguinte submodelo para média:

$$g(\mu_t) = -1 + 0.5x_{t2} + 0.3x_{t3},$$

onde $g(\cdot)$ é a função de ligação logit e $\phi = 120$ para o modelo de regressão beta com dispersão fixa. Para o modelos com dispersão variável, utilizou-se o mesmo submodelo para a média descrito anteriormente e o submodelo da dispersão é

$$\log(\phi_t) = 1.5 + 3z_{t2} + 2.5z_{t3},$$

onde $x_{ti} = z_{ti}$ para $i = 1, 2$ e $\forall t$. Esta simulação de tamanho será denominada ‘Tamanho 3’.

A Tabela 4.3 apresenta as taxas de rejeição do teste de má especificação (sob a hipótese nula). Note que, como esperado, as taxas de rejeição do teste de má especificação convergem para os níveis nominais à medida em que n cresce.

Em uma análise global, note que em todas as simulações de tamanho o teste de má especificação tipicamente apresentou taxas de rejeição da hipótese nula mais próximas aos níveis nominais quando se utiliza o teste escore. Note também que o teste de má especificação apresenta o pior desempenho global quando se utiliza o teste Wald.

Tabela 4.3 Tamanho 3 (%).

Teste	disp. fixa			n	Teste	disp. variável		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	2.14	8.23	14.90	20	RV	5.05	13.51	21.80
Wald	3.27	9.71	16.08		Wald	12.30	22.41	30.14
Escore	1.11	6.87	13.54		Escore	1.21	6.20	13.09
RV	1.41	6.26	12.21	40	RV	2.13	7.66	13.76
Wald	1.89	6.90	12.82		Wald	3.29	9.91	16.39
Escore	0.98	5.53	11.43		Escore	1.04	5.41	11.17
RV	1.47	5.93	11.21	60	RV	1.48	6.65	12.57
Wald	1.71	6.31	11.68		Wald	2.15	7.90	14.10
Escore	1.16	5.61	10.84		Escore	0.98	5.40	10.93
RV	1.36	5.82	11.06	80	RV	1.27	5.91	11.35
Wald	1.43	6.05	11.27		Wald	1.76	6.81	12.56
Escore	1.20	5.60	10.77		Escore	0.95	5.11	10.31
RV	1.43	5.69	10.80	100	RV	1.16	5.80	11.41
Wald	1.55	6.00	10.99		Wald	1.48	6.47	12.11
Escore	1.26	5.47	10.48		Escore	1.00	5.18	10.52

4.3 Poder - Estendendo o submodelo da média

4.3.1 Poder em identificar má especificação da distribuição da variável resposta

Para este experimento a variável resposta foi modelada utilizando regressão beta com dispersões fixa e variável. No entanto, em cada réplica de Monte Carlo, a variável resposta foi gerada seguindo distribuição beta inflacionada-um (BEOI) com parâmetro de mistura fixo, proposta por Ospina e Ferrari (2010). Como essa distribuição admite a existência de valores que não possuem suporte nos modelos de regressão beta, utilizou-se a transformação proposta por Smithson e Verkuilen (2006) a fim de garantir que $y \in (0, 1)$. A transformação consiste em criar uma nova variável resposta, denotada por y' , a qual é obtida da seguinte forma:

$$y' = \frac{y(n-1) + 0.5}{n}.$$

Essa é a variável resposta usada na estimação dos parâmetros.

Em todas as simulações feitas para este experimento foram utilizados valores críticos estimados exatos obtidos a partir da simulação Tamanho 1.

Nas Tabelas 4.4, 4.5 e 4.6 estão apresentadas as taxas de rejeição dos testes para os modelos com dispersão fixa e variável, utilizando parâmetro de mistura igual a 5%, 10% e 15%, respectivamente. Como se pode observar na Tabela 4.4, o teste de má especificação apresenta baixo poder, tanto no modelo com dispersão fixa quanto no modelo com dispersão variável. Como

Tabela 4.4 Poder (%), y segue distribuição BEOI com parâmetro de mistura igual 5%.

Teste	disp. fixa			n	Teste	disp. variável		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	3.42	7.30	9.55	20	RV	2.35	7.59	13.91
Wald	2.60	6.21	9.65		Wald	2.38	7.71	13.67
Escore	3.59	6.28	9.62		Escore	2.19	7.28	13.54
RV	3.83	6.73	9.95	40	RV	3.22	9.46	15.78
Wald	3.77	6.70	9.92		Wald	2.81	9.54	15.43
Escore	3.58	6.60	9.75		Escore	3.40	9.68	15.59
RV	3.18	6.66	10.84	60	RV	5.50	13.62	20.40
Wald	3.25	6.70	10.76		Wald	5.12	13.79	20.48
Escore	2.95	6.54	10.66		Escore	5.72	13.60	19.91
RV	2.24	6.18	11.00	80	RV	6.05	14.57	21.72
Wald	2.28	6.09	11.26		Wald	5.57	14.57	22.13
Escore	2.21	6.20	10.85		Escore	6.52	14.37	21.40
RV	2.34	6.62	13.06	100	RV	6.62	15.54	22.67
Wald	2.46	6.58	13.22		Wald	6.43	15.81	23.09
Escore	2.29	6.52	12.81		Escore	6.78	15.25	22.04

era de se esperar, à medida em que o tamanho da amostra aumenta as taxas de rejeição também se elevam. Vale ressaltar ainda que o teste de má especificação apresenta melhor desempenho quando aplicado ao modelo de regressão beta com dispersão variável do que quando aplicado ao modelo de regressão beta com dispersão fixa.

A Tabela 4.5 apresenta as taxas de rejeição dos testes para o erro de especificação em que o parâmetro de mistura é 10%. Assim como no caso anterior, os testes apresentam baixo poder na identificação de má especificação da distribuição de y . No entanto, o poder do teste de má especificação teve uma discreta melhora considerando o modelo com dispersão fixa e também uma leve piora considerando o modelo com dispersão variável.

Por fim, a Tabela 4.6 apresenta os resultados obtidos das simulações utilizando parâmetro de mistura igual a 15%. Note que, levando em consideração o cenário em que o teste de má especificação é aplicado em modelo de regressão beta com dispersão variável, para n igual a 100, o teste apresentou pior desempenho, no que tange à poder em identificação de erro de especificação, no cenário em que se utiliza parâmetro de mistura igual a 15%.

Em uma análise geral dos diferentes casos tratados (parâmetro de mistura igual a 5%, 10% e 15%), é notório que à medida em que n aumenta, o poder do teste de má especificação também se eleva, como é de se esperar. Note que, para n suficientemente grande (100) e analisando os cenários onde se utilizou parâmetros de mistura igual a 5%, 10% e 15%, o poder do teste de má especificação tende a diminuir à medida em que o parâmetro de mistura aumenta.

Tabela 4.5 Poder (%), y segue distribuição BEOI com parâmetro de mistura igual 10%.

Teste	disp. fixa			n	Teste	disp. variável		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	2.81	6.82	9.30	20	RV	2.90	8.09	13.84
Wald	2.26	5.58	9.46		Wald	2.62	7.75	13.18
Escore	2.87	5.74	9.19		Escore	2.94	8.00	14.10
RV	2.25	5.83	10.18	40	RV	3.43	9.62	15.26
Wald	2.30	5.82	10.26		Wald	2.83	9.42	14.87
Escore	2.25	5.66	9.84		Escore	3.87	9.80	15.39
RV	1.74	5.77	11.20	60	RV	4.32	11.63	18.07
Wald	1.75	5.80	11.24		Wald	3.87	11.25	17.96
Escore	1.71	5.63	11.13		Escore	4.63	11.57	17.60
RV	1.34	5.76	12.14	80	RV	4.41	11.50	17.79
Wald	1.38	5.79	12.55		Wald	4.12	11.13	17.74
Escore	1.33	5.61	11.80		Escore	4.79	11.28	17.58
RV	1.61	7.13	14.09	100	RV	4.27	11.37	18.09
Wald	1.58	7.27	14.42		Wald	4.02	11.32	17.88
Escore	1.61	6.88	13.60		Escore	4.28	11.45	17.92

Tabela 4.6 Poder (%), y segue distribuição BEOI com parâmetro de mistura igual 15%.

Teste	disp. fixa			n	Teste	disp. variável		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	2.11	6.08	8.57	20	RV	2.91	7.53	12.91
Wald	1.97	5.01	8.67		Wald	2.51	6.77	11.55
Escore	2.12	4.92	8.27		Escore	3.00	7.77	13.64
RV	1.15	4.96	9.97	40	RV	2.76	7.89	13.65
Wald	1.13	4.94	10.24		Wald	2.24	7.62	12.85
Escore	1.30	4.91	9.37		Escore	3.04	8.30	13.73
RV	1.30	5.48	11.79	60	RV	3.26	9.11	15.16
Wald	1.21	5.45	11.83		Wald	2.79	8.71	14.71
Escore	1.26	5.25	11.14		Escore	3.32	9.34	15.14
RV	1.08	6.19	12.53	80	RV	2.89	8.32	14.34
Wald	1.10	6.20	12.82		Wald	2.43	8.00	14.09
Escore	1.08	5.95	12.13		Escore	3.23	8.46	14.18
RV	1.45	7.09	13.49	100	RV	2.70	9.00	14.86
Wald	1.44	7.09	13.66		Wald	2.41	8.67	14.50
Escore	1.54	6.80	13.19		Escore	2.94	9.01	14.84

4.3.2 Poder em identificar omissão de variável importante

Para este experimento foram considerados dois modelos de regressão beta, um com dispersão fixa e outro com dispersão variável. Considere os seguintes submodelos para média:

$$g(\mu_t^*) = -1 + 0.5x_{t2} + 0.3x_{t3} + 2.5x_{t4},$$

$$g(\mu_t) = -1 + 0.5x_{t2} + 0.3x_{t3},$$

onde $g(\cdot)$ é uma função de ligação. A variável resposta foi gerada de distribuição beta com parâmetros μ_t^* e $\phi = 120$ (para o modelo com dispersão fixa) e μ_t^* e ϕ_t (para o modelo com dispersão variável), onde

$$h(\phi_t) = 1.5 + 3z_{t2} + 2.5z_{t3},$$

em que $x_{ti} = z_{ti}$ para $i = 2, 3$ e $\forall t$. $h(\cdot)$ é uma função de ligação.

Os modelos estimados utilizam os parâmetros μ_t e ϕ (dispersão fixa) e μ_t e ϕ_t (dispersão variável), respectivamente. Dessa forma, a má especificação se configura pelo fato de os modelos estimados não levarem em consideração a influência da variável x_3 nos submodelos da média. Os valores máximo e mínimo de μ_t^*/μ_t são 2.4847 e 1.1545, respectivamente.

Tabela 4.7 Poder (%), omissão de variável regressora importante.

Teste	disp. fixa			n	Teste	disp. variável		
	α	α	α			α	α	α
	1%	5%	10%			1%	5%	10%
RV	0.00	0.60	4.47	20	RV	0.14	2.11	6.83
Wald	0.00	0.68	5.02		Wald	0.41	2.12	5.91
Escore	0.01	0.38	3.59		Escore	0.37	2.68	6.20
RV	1.18	17.08	36.41	40	RV	2.62	14.84	27.24
Wald	1.38	18.68	38.27		Wald	2.86	18.13	32.04
Escore	0.75	15.06	34.04		Escore	2.05	10.17	19.56
RV	8.57	42.31	59.92	60	RV	8.64	28.03	41.66
Wald	10.83	44.66	61.20		Wald	13.26	33.84	46.34
Escore	6.60	39.43	58.37		Escore	5.22	19.10	33.46
RV	25.99	60.48	72.98	80	RV	16.90	40.37	53.72
Wald	28.43	62.04	73.97		Wald	24.30	45.99	57.66
Escore	23.01	58.59	72.05		Escore	9.87	31.62	47.37
RV	42.52	73.27	82.41	100	RV	24.18	49.02	59.87
Wald	45.76	74.64	83.16		Wald	33.65	54.11	63.13
Escore	38.86	71.88	81.70		Escore	14.16	40.28	54.57

Note na Tabela 4.7 que para n pequeno (20) o teste de má especificação apresenta maior poder quando utilizado no modelo com dispersão variável. No entanto, à medida em que n cresce o teste de má especificação apresenta maior poder quando utilizado no modelo com dispersão fixa.

4.3.3 A influência de outliers

Neste experimento, com o intuito de avaliar o desempenho do teste sob a influência de *outlier*, os coeficientes foram definidos a fim de se limitar o intervalo de μ_t em $(0, 0.5)$, dessa forma, após a escolha dos coeficientes, temos que μ_t assume valores em $[0.127555, 0.469582]$.

Para introduzir um *outlier* na amostra, em cada réplica de Monte Carlo, o maior valor da variável resposta ($y_{\max}$) foi substituído pelo mínimo entre seu dobro e 0.99, ou seja

$$y_{\text{outlier}} = \min(2 \times y_{\max}, 0.99),$$

com intuito de garantir que nenhuma das observações tenha valor igual ou superior a 1. Estes procedimentos foram adotados tanto no modelo com dispersão fixa quanto no modelo com dispersão variável. Os coeficientes dos submodelos da média, tanto no modelo com dispersão fixa como no modelo com dispersão variável, foram fixados em $\beta_1 = -2.3$, $\beta_2 = 0.8$, $\beta_3 = 1.5$. Para o modelo com dispersão fixa usamos $\phi = 120$, e para o submodelo da dispersão; no modelo com dispersão variável, usamos $\gamma_1 = 1.5$, $\gamma_2 = 3.0$, $\gamma_3 = 2.5$.

Tabela 4.8 Poder (%), variável resposta com outlier.

Teste	disp. fixa			n	Teste	disp. variável		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	87.93	89.40	89.50	20	RV	15.64	31.81	41.44
Wald	85.93	89.27	89.48		Wald	16.27	38.39	49.21
Escore	88.76	89.46	89.50		Escore	11.06	24.15	31.66
RV	85.75	93.69	95.16	40	RV	13.88	28.28	37.67
Wald	84.40	93.50	95.05		Wald	12.81	29.42	39.58
Escore	87.01	94.09	95.18		Escore	12.39	26.68	35.85
RV	85.74	94.72	96.24	60	RV	12.16	25.34	34.19
Wald	84.96	94.57	96.19		Wald	12.80	26.43	35.36
Escore	86.88	94.91	96.26		Escore	11.09	24.03	33.02
RV	86.75	95.39	96.83	80	RV	10.02	22.67	32.21
Wald	85.85	95.29	96.82		Wald	10.57	23.57	33.21
Escore	87.17	95.54	96.88		Escore	9.20	21.60	31.33
RV	86.69	95.20	96.83	100	RV	9.36	21.58	30.26
Wald	85.70	95.13	96.81		Wald	9.51	22.21	30.92
Escore	87.43	95.34	96.87		Escore	8.79	20.93	29.60

A Tabela 4.8 apresenta as taxas de rejeição do teste de má especificação sob a influência de *outlier* para o modelo com dispersão fixa e para o modelo com dispersão variável. No caso de estimação com dispersão fixa o teste possui bom poder em todos os níveis nominais, diferentemente do que ocorre no caso em que a estimação é feita considerando o modelo com dispersão variável, onde o poder ao invés de crescer à medida em que n aumenta, está diminuindo.

4.3.4 Poder em identificar erro no submodelo da precisão

Neste experimento a variável resposta é gerada a partir de uma distribuição beta de parâmetros μ_t e ϕ_t , de forma que o parâmetro de precisão não é constante e tem a seguinte forma:

$$\log(\phi_t) = 1.5 + 3z_{t2} + 2.5z_{t3}.$$

O modelo estimado, contudo, é o modelo de regressão beta com dispersão fixa. Note que isso configura um erro de especificação do modelo e este erro de especificação será chamado ‘Erro 1’.

Adicionalmente, utiliza-se também o caso inverso, ou seja: a variável resposta é gerada de distribuição beta com parâmetros μ_t e ϕ , onde $\phi = 120$, porém, o modelo estimado é o de regressão beta com dispersão variável. Assim como no caso anterior, aqui também há um erro de especificação do modelo e este erro é denotado por ‘Erro 2’.

Tabela 4.9 Poder (%), erro no submodelo da precisão.

Teste	Erro 1			n	Teste	Erro 2		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	5.62	21.00	26.90	20	RV	0.86	4.54	9.52
Wald	4.72	16.36	26.06		Wald	1.00	5.54	10.71
Escore	15.67	25.67	34.08		Escore	62.78	69.87	74.02
RV	7.97	19.64	28.32	40	RV	1.15	4.82	9.73
Wald	7.14	18.61	27.67		Wald	1.30	5.49	10.38
Escore	14.96	26.86	33.80		Escore	56.94	65.49	70.42
RV	8.68	20.68	29.18	60	RV	1.09	4.96	9.69
Wald	8.01	19.81	28.59		Wald	1.22	5.41	9.98
Escore	15.24	27.12	35.60		Escore	57.15	65.43	70.12
RV	8.97	20.97	29.58	80	RV	1.10	4.96	9.90
Wald	8.25	20.44	29.22		Wald	1.07	5.18	10.01
Escore	14.11	26.87	35.55		Escore	57.87	65.97	70.85
RV	9.53	21.98	31.47	100	RV	0.98	4.87	9.49
Wald	8.80	21.16	31.23		Wald	0.98	5.01	9.67
Escore	15.04	27.25	35.09		Escore	58.34	67.38	71.78

Note que para o cenário de má especificação onde o modelo de regressão beta é escolhido de forma incorreta, como se pode observar na Tabela 4.9, o teste escore de má especificação apresenta poder bem maior sob o Erro 2 do que sob o Erro 1, chegando o poder a ser mais que o dobro, com nível nominal de 10% e $n = 20$. No entanto, levando-se em consideração os testes da razão de verossimilhanças e Wald, o teste de má especificação apresenta mais poder sob o Erro 1 do que sob o Erro 2, inclusive chegando o poder sob o Erro 1 a ser mais que o triplo do poder sob o Erro 2, como se pode observar na Tabela 4.9 utilizando o teste da razão de verossimilhanças, com nível nominal de 10% e n igual a 100.

Note ainda que, assim como no Erro 2, o teste de má especificação apresenta maior poder sob o Erro 1 quando se utiliza o teste escore, tendo desempenho bastante superior se comparado ao poder do teste de má especificação utilizando o teste Wald ou o teste da razão de verossimilhanças, principalmente para os níveis nominais de 1% e 5%.

4.3.5 Poder em identificar má especificação da função de ligação para o submodelo da média

Este experimento consiste em estimar o poder do teste de má especificação proposto por Cribari-Neto & Lima (2007) no que tange à identificação de erro na função de ligação utilizada para modelar a média.

Lima (2007) fez este experimento em sua dissertação de mestrado levando em consideração apenas modelos de regressão beta com dispersão fixa. No entanto, Lima (2007) considerou o caso em que a função de ligação correta para o submodelo da média seria, por exemplo, logit, no entanto, ao se realizar a estimação, a função de ligação escolhida para o submodelo da média não era logit, configurando assim, um “erro completo” na escolha da função de ligação.

Aqui será considerado um “erro parcial” na escolha da função de ligação, onde a função de ligação correta para modelar a média, tanto no modelo com dispersão fixa quanto no modelo com dispersão variável, é a função logit, ou seja

$$\log\left(\frac{\mu_t}{1-\mu_t}\right) = \eta_t,$$

onde η_t é o preditor linear. No entanto, a variável resposta é gerada a partir de uma mistura em diversas proporções da função de ligação logit e log-log complementar, esta segunda função de ligação sendo dada por

$$\log[-\log(\mu_t)] = \eta_t.$$

As amostras da variável resposta são geradas com 95%, 90%, 80% ou 75% de sua composição através da função de ligação logit e os restantes 5%, 10%, 15%, 20% e 25% de sua composição por meio da função de ligação log-log complementar, respectivamente.

As Tabelas 4.10, 4.11, 4.12, 4.13 e 4.14 apresentam o poder do teste de má especificação para os graus de “poluição” de 5%, 10%, 15%, 20% e 25%, respectivamente.

Note que há uma clara tendência de aumento do poder do teste de má especificação à medida em que o grau de poluição se eleva, tanto quando se utiliza modelo com dispersão fixa quanto quando se utiliza modelo com dispersão variável. Note também que o teste de má especificação apresenta quase sempre baixo poder em identificar má especificação da função de ligação utilizada para modelar a média. No entanto, como esperado, o poder aumenta à medida em que o número de observações se eleva.

Vale ressaltar que em todos os cenários o teste de má especificação apresentou maior poder em modelos de regressão beta com dispersão variável do que em modelos com dispersão fixa. Por fim, note que o desempenho do teste de má especificação quando se utilizam os testes da razão de verossimilhanças, Wald ou escore é bastante parecido em todos os cenários. O teste de má especificação que utiliza o teste Wald apresenta poder ligeiramente menor do que os testes da razão de verossimilhanças e escore.

Tabela 4.10 Poder (%), gerando y com 5% de poluição log-log complementar.

Teste	disp. fixa			n	Teste	disp. variável		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	5.11	10.12	12.80	20	RV	2.91	8.87	15.20
Wald	4.86	8.62	12.66		Wald	2.74	8.76	14.88
Escore	4.61	8.87	12.80		Escore	2.79	8.78	14.98
RV	5.46	13.25	18.60	40	RV	5.21	14.45	22.23
Wald	5.64	13.34	18.66		Wald	4.54	14.19	21.96
Escore	4.43	12.62	18.43		Escore	5.67	14.77	22.26
RV	5.88	14.99	22.46	60	RV	9.57	21.93	30.10
Wald	6.19	15.38	22.38		Wald	8.92	21.99	30.19
Escore	5.09	14.30	22.04		Escore	9.67	22.10	30.07
RV	4.68	15.11	23.28	80	RV	11.55	26.08	35.91
Wald	5.15	15.37	23.48		Wald	10.83	26.13	36.26
Escore	4.20	14.22	22.85		Escore	12.40	26.04	36.08
RV	5.58	15.98	24.68	100	RV	14.38	30.32	40.95
Wald	5.84	16.17	25.02		Wald	14.14	30.28	41.17
Escore	5.13	15.43	24.16		Escore	14.75	30.42	40.68
RV	6.56	16.80	25.07	160	RV	23.16	40.66	50.56
Wald	6.74	17.02	25.48		Wald	23.21	40.57	50.65
Escore	6.22	16.64	25.25		Escore	23.77	40.57	50.61
RV	6.73	16.79	24.94	200	RV	25.90	43.68	52.74
Wald	6.97	16.88	25.05		Wald	25.91	44.06	52.86
Escore	6.39	16.49	24.81		Escore	26.15	43.32	52.60
RV	7.55	18.04	26.67	300	RV	33.58	45.91	52.90
Wald	7.72	18.16	26.73		Wald	33.92	46.22	53.07
Escore	7.24	17.87	26.65		Escore	33.49	45.61	52.84
RV	8.51	18.92	27.11	500	RV	33.10	44.31	51.71
Wald	8.61	19.03	27.15		Wald	33.09	44.41	51.73
Escore	8.51	18.78	27.02		Escore	33.04	44.31	51.70
RV	16.00	29.80	36.40	3000	RV	60.40	73.80	79.00
Wald	16.00	34.20	36.40		Wald	60.80	73.60	79.00
Escore	16.20	30.00	36.40		Escore	60.20	74.00	79.00

Tabela 4.11 Poder (%), gerando y com 10% de poluição log-log complementar.

Teste	disp. fixa			n	Teste	disp. variável		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	6.73	12.51	14.87	20	RV	5.29	14.12	22.24
Wald	6.15	10.85	14.62		Wald	4.29	13.08	20.71
Escore	6.26	11.31	15.06		Escore	5.19	13.82	22.17
RV	6.00	16.76	23.55	40	RV	10.94	25.83	36.04
Wald	6.09	16.70	23.46		Wald	9.09	25.30	35.79
Escore	5.31	16.27	23.29		Escore	11.80	26.20	36.17
RV	6.84	18.18	27.17	60	RV	20.15	37.99	48.10
Wald	6.87	18.32	27.04		Wald	19.27	38.22	48.50
Escore	6.32	17.85	26.97		Escore	20.06	37.77	47.63
RV	6.87	18.49	27.78	80	RV	25.10	43.23	52.82
Wald	7.07	18.41	27.78		Wald	24.06	43.23	53.14
Escore	6.48	18.05	27.78		Escore	25.79	42.87	52.63
RV	8.26	20.29	29.70	100	RV	28.90	47.21	55.89
Wald	8.27	20.09	29.90		Wald	29.04	47.52	56.05
Escore	8.05	20.19	29.52		Escore	28.96	46.80	55.49
RV	9.79	21.34	30.08	160	RV	35.30	47.23	53.94
Wald	9.67	21.45	30.43		Wald	35.63	47.22	54.14
Escore	9.48	21.39	30.31		Escore	35.07	46.76	53.81
RV	9.63	20.36	29.46	200	RV	34.45	46.56	53.15
Wald	9.63	20.37	29.40		Wald	34.87	46.91	53.22
Escore	9.57	20.46	29.61		Escore	34.46	46.20	53.11
RV	10.85	22.19	30.82	300	RV	33.80	44.62	51.51
Wald	10.98	22.14	30.83		Wald	34.07	44.79	51.72
Escore	10.46	22.17	30.88		Escore	33.62	44.55	51.48
RV	12.36	23.89	32.24	500	RV	38.06	51.48	59.52
Wald	12.30	23.92	32.10		Wald	38.09	51.54	59.54
Escore	12.35	23.69	32.24		Escore	38.06	51.36	59.58
RV	33.20	46.60	56.20	3000	RV	88.60	94.80	96.40
Wald	33.20	54.20	56.20		Wald	88.60	94.80	96.40
Escore	33.20	46.60	56.20		Escore	88.60	94.80	96.40

Tabela 4.12 Poder (%), gerando y com 15% de poluição log-log complementar.

Teste	disp. fixa			n	Teste	disp. variável		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	7.62	14.59	17.34	20	RV	7.39	20.14	29.68
Wald	6.81	12.34	16.78		Wald	5.68	17.39	27.51
Escore	7.18	13.25	17.84		Escore	7.48	19.83	29.68
RV	5.86	17.38	25.25	40	RV	17.83	37.37	47.76
Wald	5.65	16.78	24.89		Wald	15.39	36.77	47.42
Escore	5.62	17.02	25.26		Escore	18.80	37.44	47.54
RV	7.60	18.99	27.58	60	RV	29.27	47.41	56.20
Wald	7.23	18.60	26.97		Wald	28.46	47.65	56.28
Escore	7.49	18.91	27.60		Escore	28.95	47.01	55.74
RV	7.43	19.60	28.91	80	RV	33.42	48.34	56.22
Wald	7.39	19.16	28.69		Wald	33.01	48.67	56.50
Escore	7.29	19.61	29.15		Escore	33.56	47.84	55.61
RV	8.67	21.48	30.40	100	RV	34.53	47.75	54.05
Wald	8.45	20.94	30.06		Wald	34.92	47.99	54.33
Escore	8.77	21.53	30.53		Escore	34.35	46.98	53.57
RV	10.18	22.45	30.85	160	RV	32.32	42.06	49.46
Wald	9.97	22.26	31.08		Wald	32.45	42.05	49.55
Escore	10.24	22.51	31.30		Escore	32.15	41.84	49.35
RV	11.30	23.03	30.88	200	RV	30.78	43.34	50.84
Wald	11.22	22.82	30.79		Wald	30.90	43.42	50.66
Escore	11.27	23.04	30.97		Escore	30.80	43.02	50.91
RV	13.01	25.06	33.68	300	RV	33.55	46.51	54.23
Wald	12.98	25.13	33.79		Wald	33.69	46.58	54.26
Escore	12.88	25.14	33.77		Escore	33.50	46.49	54.24
RV	15.72	27.92	36.12	500	RV	43.29	56.86	64.37
Wald	15.61	27.93	35.98		Wald	43.24	56.94	64.43
Escore	15.76	27.72	36.08		Escore	43.27	56.61	64.34
RV	45.60	60.60	67.20	3000	RV	96.40	98.00	98.60
Wald	45.40	65.60	67.20		Wald	96.40	98.00	98.60
Escore	46.00	60.80	67.00		Escore	96.40	98.00	98.60

Tabela 4.13 Poder (%), gerando y com 20% de poluição log-log complementar.

Teste	disp. fixa			n	Teste	disp. variável		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	8.02	16.29	19.54	20	RV	9.47	26.12	37.58
Wald	6.88	13.34	18.54		Wald	6.41	22.12	34.81
Escore	7.69	15.06	20.15		Escore	9.89	25.91	37.39
RV	6.13	17.88	25.87	40	RV	24.33	43.58	53.75
Wald	5.76	17.01	25.32		Wald	21.69	42.83	53.39
Escore	6.29	17.90	26.00		Escore	25.07	43.31	53.51
RV	7.84	19.13	27.06	60	RV	34.23	48.67	55.46
Wald	7.43	18.43	26.35		Wald	33.87	49.16	55.78
Escore	7.97	19.32	27.55		Escore	33.26	47.81	54.86
RV	7.44	18.61	26.77	80	RV	35.17	45.82	52.35
Wald	7.24	17.97	26.26		Wald	35.09	46.30	52.47
Escore	7.48	18.82	27.10		Escore	34.94	45.17	51.86
RV	8.54	19.74	28.16	100	RV	32.92	42.51	48.61
Wald	8.06	19.23	27.90		Wald	33.06	42.56	48.55
Escore	8.84	19.93	28.17		Escore	32.38	42.10	48.31
RV	10.27	21.85	30.03	160	RV	28.02	39.59	47.59
Wald	9.86	21.53	30.14		Wald	27.83	39.15	47.41
Escore	10.37	21.96	30.43		Escore	28.11	39.55	47.54
RV	10.62	21.90	30.82	200	RV	29.34	43.39	51.06
Wald	10.37	21.70	30.59		Wald	29.16	43.45	50.96
Escore	10.75	22.07	30.90		Escore	29.55	43.11	51.00
RV	14.32	26.83	35.35	300	RV	34.99	48.48	55.71
Wald	14.19	26.57	35.18		Wald	35.10	48.60	55.85
Escore	14.19	27.00	35.41		Escore	34.93	48.44	55.79
RV	18.24	31.63	40.38	500	RV	46.26	59.48	66.86
Wald	17.92	31.59	40.30		Wald	46.12	59.52	66.88
Escore	18.32	31.48	40.43		Escore	46.15	59.37	66.82
RV	59.40	69.00	78.60	3000	RV	97.80	98.60	99.60
Wald	59.20	75.80	78.60		Wald	97.80	98.60	99.60
Escore	59.40	69.00	78.60		Escore	97.80	98.60	99.60

Tabela 4.14 Poder (%), gerando y com 25% de poluição log-log complementar.

Teste	disp. fixa			n	Teste	disp. variável		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	8.04	17.59	21.33	20	RV	11.28	32.25	45.91
Wald	6.52	14.21	19.99		Wald	6.66	26.84	42.22
Escore	8.20	16.41	22.34		Escore	12.59	31.83	45.34
RV	7.51	18.33	26.01	40	RV	29.95	47.66	56.42
Wald	6.66	17.40	25.15		Wald	27.32	47.72	56.62
Escore	7.90	18.70	26.52		Escore	29.70	47.19	55.85
RV	8.45	18.84	26.50	60	RV	34.47	44.98	49.88
Wald	7.65	18.10	25.82		Wald	34.57	45.39	50.14
Escore	8.89	19.26	26.98		Escore	33.52	44.16	49.31
RV	7.94	18.30	25.47	80	RV	31.23	39.80	45.47
Wald	7.49	17.70	25.02		Wald	31.20	39.76	45.43
Escore	8.13	18.54	25.88		Escore	30.89	39.31	45.24
RV	8.03	18.17	25.73	100	RV	27.71	36.89	43.05
Wald	7.62	17.39	25.58		Wald	27.71	36.65	42.73
Escore	8.54	18.38	25.84		Escore	27.34	36.70	42.94
RV	9.00	19.62	27.66	160	RV	25.30	37.65	45.44
Wald	8.56	19.23	27.76		Wald	24.85	37.18	45.16
Escore	9.22	19.93	28.05		Escore	25.54	37.72	45.64
RV	9.55	19.87	28.17	200	RV	28.13	41.42	49.37
Wald	9.27	19.66	28.00		Wald	27.82	41.37	49.21
Escore	9.69	19.95	28.38		Escore	28.32	41.32	49.45
RV	12.87	24.83	33.61	300	RV	35.32	48.05	55.21
Wald	12.51	24.67	33.43		Wald	35.21	48.03	55.31
Escore	12.96	24.95	33.80		Escore	35.36	48.07	55.33
RV	18.66	32.47	41.44	500	RV	46.76	60.26	67.52
Wald	18.25	32.39	41.10		Wald	46.60	60.30	67.52
Escore	18.81	32.34	41.51		Escore	46.66	60.16	67.52
RV	70.60	81.80	87.40	3000	RV	97.80	99.40	99.80
Wald	70.40	86.80	87.40		Wald	97.80	99.40	99.80
Escore	70.60	81.80	87.40		Escore	97.80	99.40	99.80

4.4 Tamanho - Estendendo os submodelos da média e dispersão

Nesta seção, assim como na Seção 4.2, a variável resposta é gerada e modelada sem erro de especificação. No entanto, aqui se aplica o novo teste de má especificação detalhado no Capítulo 3 a fim de se analisar seu comportamento em amostras de tamanho finito. Este novo teste de má especificação consiste em estender simultaneamente os submodelos da média e da dispersão.

Todas as simulações desta seção foram realizadas utilizando modelos de regressão beta com dispersão variável. Para simplificar a discussão, a partir de agora o teste proposto por Cribari-Neto & Lima (2007) será denotado ‘RESET 1’ e o novo teste proposto nesta dissertação será chamado ‘RESET 2’.

Nesta seção, além de apresentar os resultados das simulações de tamanho do teste RESET 2, será apresentado também um breve comparativo entre as simulações de tamanho dos testes RESET 1 e RESET 2. Para estas simulações de tamanho serão utilizados dois modelos, e a partir destas simulações serão obtidos os valores críticos estimados exatos usados nas simulações de poder da próxima seção.

Na primeira simulação de tamanho, os valores da variável resposta foram gerados de acordo com o submodelo da média

$$g(\mu_t) = -1.8 + 1.4x_{t2} + 1.6x_{t3},$$

onde $g(\cdot)$ é a função de ligação logit, o submodelo da precisão sendo

$$\log(\phi_t) = 1.5 + 3z_{t2} + 2.5z_{t3},$$

onde $x_{ti} = z_{ti}$ para $i = 1, 2$ e $\forall t$. Esta simulação de tamanho será denominada ‘Tamanho 4’. O que diferencia esta simulação de tamanho da simulação Tamanho 1 é o teste de má especificação utilizado.

A Tabela 4.15 apresenta as taxas de rejeição do teste RESET 2, sob a hipótese nula, e também as taxas de rejeição do teste RESET 1. Note que, de forma geral, o teste RESET 1 apresenta taxas de rejeição mais próximas aos valores nominais do que o teste RESET2, nas três estatísticas de teste consideradas.

Na segunda simulação de tamanho a variável resposta foi gerada utilizando o seguinte submodelo para média:

$$g(\mu_t) = -1 + 0.5x_{t2} + 0.3x_{t3},$$

onde $g(\cdot)$ é a função de ligação logit. O submodelo da dispersão é

$$\log(\phi_t) = 1.5 + 3z_{t2} + 2.5z_{t3},$$

onde $x_{ti} = z_{ti}$ para $i = 1, 2$ e $\forall t$. Esta simulação de tamanho será denominada ‘Tamanho 5’.

Assim como na simulação de tamanho anterior, o que diferencia esta da apresentada na Seção 4.2 é o fato de se estar utilizando o teste RESET 2 ao invés do teste RESET 1.

Note, na Tabela 4.16, que mais uma vez o teste RESET 1 apresentou melhor desempenho do que o teste RESET 2 no que diz respeito a tamanho, apresentando para todos os testes (razão de verossimilhanças, Wald e escore) taxas de rejeição mais próximas aos valores nominais.

Observando as Tabelas 4.15 e 4.16, é fácil ver que, para os modelos especificados, o teste RESET 1 apresenta desempenho superior ao teste RESET 2 em termos de tamanho.

Tabela 4.15 Tamanho 4 (%).

Teste	RESET 2			n	Teste	RESET 1		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	10.93	26.21	36.37	20	RV	4.51	12.28	19.44
Wald	48.29	59.27	65.66		Wald	10.57	19.32	26.65
Escore	0.51	2.85	5.99		Escore	1.10	6.25	11.77
RV	4.18	12.48	20.43	40	RV	1.91	7.67	14.15
Wald	14.58	25.04	33.07		Wald	3.29	10.08	16.25
Escore	0.93	3.79	7.92		Escore	1.06	5.71	11.56
RV	2.48	8.57	15.05	60	RV	1.46	6.48	12.35
Wald	7.47	15.9	23.43		Wald	2.07	7.66	13.96
Escore	0.87	3.76	7.87		Escore	0.95	5.28	10.85
RV	1.91	7.71	13.82	80	RV	1.34	6.35	11.80
Wald	5.03	12.73	19.42		Wald	1.74	7.22	12.65
Escore	0.94	4.41	8.76		Escore	0.98	5.49	10.81
RV	1.81	7.73	13.68	100	RV	1.39	6.04	11.36
Wald	4.28	11.39	17.63		Wald	1.76	6.68	12.11
Escore	0.98	4.87	9.56		Escore	1.10	5.35	10.72

Tabela 4.16 Tamanho 5 (%).

Teste	RESET 2			n	Teste	RESET 1		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	13.01	29.37	39.86	20	RV	5.05	13.51	21.80
Wald	52.07	63.14	69.58		Wald	12.3	22.41	30.14
Escore	0.39	2.50	5.840		Escore	1.21	6.20	13.09
RV	4.48	13.07	20.90	40	RV	2.13	7.66	13.76
Wald	15.50	26.68	34.22		Wald	3.29	9.91	16.39
Escore	0.97	3.65	7.95		Escore	1.04	5.41	11.17
RV	2.48	9.08	15.98	60	RV	1.48	6.65	12.57
Wald	7.82	16.73	24.06		Wald	2.15	7.90	14.10
Escore	0.77	4.20	8.83		Escore	0.98	5.40	10.93
RV	1.85	8.05	14.21	80	RV	1.27	5.91	11.35
Wald	5.39	13.13	19.90		Wald	1.76	6.81	12.56
Escore	0.92	4.17	8.62		Escore	0.95	5.11	10.31
RV	1.68	6.95	13.13	100	RV	1.16	5.80	11.41
Wald	3.97	11.37	17.58		Wald	1.48	6.47	12.11
Escore	0.85	3.95	8.67		Escore	1.00	5.18	10.52

4.5 Poder - Estendendo os submodelos da média e dispersão

Todos os experimentos que serão apresentados a seguir utilizaram procedimentos similares aos descritos na seção anterior 4.3, a única diferença reside no fato de que agora utilizamos uma forma alternativa do teste de má especificação proposto por Cribari-Neto & Lima (2007) (RESET 1). Essa nova metodologia consiste em estender simultaneamente os submodelos da média e da dispersão. O desempenho desse novo teste (RESET 2) será comparado ao desempenho do teste proposto por Cribari-Neto & Lima (2007).

Em todas as simulações apresentadas nesta seção foram utilizados valores críticos estimados exatos obtidos a partir das simulações de tamanho apresentadas na seção anterior.

4.5.1 Poder em identificar erro no submodelo da precisão - RESET 2

Neste experimento, diferentemente do experimento da Seção 4.3.4, estimamos apenas o modelo de regressão beta com dispersão variável. A variável resposta foi gerada de distribuição beta de parâmetros μ_t e ϕ , em que $\phi = 120$. No entanto, a variável resposta é modelada utilizando regressão beta com dispersão variável, configurando erro de especificação do modelo.

Tabela 4.17 Poder (%), erro no submodelo da precisão - RESET 2.

Teste	RESET 2			n	Teste	RESET 1		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	1.13	5.57	10.81	20	RV	0.86	4.54	9.52
Wald	0.86	4.23	8.98		Wald	1.00	5.54	10.71
Escore	99.98	100.00	100.00		Escore	62.78	69.87	74.02
RV	1.23	5.09	10.28	40	RV	1.15	4.82	9.73
Wald	1.09	4.80	9.40		Wald	1.30	5.49	10.38
Escore	99.98	99.98	99.99		Escore	56.94	65.49	70.42
RV	1.08	5.35	10.70	60	RV	1.09	4.96	9.69
Wald	1.05	5.13	10.48		Wald	1.22	5.41	9.98
Escore	99.97	99.98	99.98		Escore	57.15	65.43	70.12
RV	1.17	5.55	10.07	80	RV	1.10	4.96	9.90
Wald	1.01	5.19	10.14		Wald	1.07	5.18	10.01
Escore	99.98	99.99	99.99		Escore	57.87	65.97	70.85
RV	0.85	4.95	9.69	100	RV	0.98	4.87	9.49
Wald	0.73	4.86	9.57		Wald	0.98	5.01	9.67
Escore	99.99	99.99	99.99		Escore	58.34	67.38	71.78

Neste caso, tanto o teste RESET 1 como o teste RESET 2 apresentaram alto poder quando se utiliza o teste escore, como se pode observar na Tabela 4.17. No entanto vale ressaltar que o teste RESET 2 apresenta desempenho superior ao teste RESET 1, apresentando sempre poder igual ou muito próximo a 1 quando se utiliza o teste escore.

4.5.2 Poder em identificar má especificação da distribuição da variável resposta - RESET 2

Os procedimentos deste experimento são similares aos utilizados no experimento da Seção 4.3.1. No entanto, aqui se será utilizado apenas o modelo de regressão beta com dispersão variável.

Tabela 4.18 Poder (%), y segue distribuição BEOI de parâmetro de mistura igual 5% - RESET 2.

Teste	RESET 2			n	Teste	RESET 1		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	8.97	17.63	24.41	20	RV	2.35	7.59	13.91
Wald	2.21	8.67	16.59		Wald	2.38	7.71	13.67
Escore	6.45	16.88	24.65		Escore	2.19	7.28	13.54
RV	28.13	39.26	46.55	40	RV	3.22	9.46	15.78
Wald	19.51	34.97	44.75		Wald	2.81	9.54	15.43
Escore	18.22	35.02	43.57		Escore	3.40	9.68	15.59
RV	41.03	53.16	60.15	60	RV	5.50	13.62	20.40
Wald	37.47	52.44	61.13		Wald	5.12	13.79	20.48
Escore	29.96	46.59	55.04		Escore	5.72	13.60	19.91
RV	47.74	59.15	65.37	80	RV	6.05	14.57	21.72
Wald	46.42	60.38	67.48		Wald	5.57	14.57	22.13
Escore	35.88	51.21	59.10		Escore	6.52	14.37	21.40
RV	51.32	62.37	68.15	100	RV	6.62	15.54	22.67
Wald	51.97	64.66	71.04		Wald	6.43	15.81	23.09
Escore	41.02	54.88	62.30		Escore	6.78	15.25	22.04

Note que o poder do teste RESET 2 cresce mais rapidamente à medida em que n aumenta do que o poder do teste RESET 1 quando o parâmetro de mistura é igual a 5% (Tabela 4.18). Nos outros dois cenários, onde o parâmetro de mistura é igual a 10% (Tabela 4.19) e 15% (Tabela 4.20), tanto o teste RESET 1 como o teste RESET 2 demonstram ter baixo poder em identificar esse tipo de má especificação.

Note ainda que mais uma vez o teste RESET 2 se mostrou superior em termos de poder se comparado ao teste RESET 1, chegando seu poder a ser mais que três vezes maior, como se pode observar na Tabela 4.19 para n igual a 80, com nível nominal de 10% e utilizando o teste Wald.

Tabela 4.19 Poder (%), y segue distribuição BEOI de parâmetro de mistura igual 10% - RESET 2.

Teste	RESET 2			n	Teste	RESET 1		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	13.2	23.89	30.78	20	RV	2.90	8.09	13.84
Wald	3.44	12.11	20.79		Wald	2.62	7.75	13.18
Escore	8.12	21.54	30.94		Escore	2.94	8.00	14.10
RV	34.24	46.16	53.45	40	RV	3.43	9.62	15.26
Wald	23.60	41.73	51.40		Wald	2.83	9.42	14.87
Escore	19.74	39.84	48.21		Escore	3.87	9.80	15.39
RV	39.61	51.44	58.73	60	RV	4.32	11.63	18.07
Wald	37.17	51.36	59.84		Wald	3.87	11.25	17.96
Escore	26.91	43.32	51.83		Escore	4.63	11.57	17.60
RV	40.42	50.96	57.50	80	RV	4.41	11.50	17.79
Wald	39.27	52.80	59.38		Wald	4.12	11.13	17.74
Escore	27.25	42.68	51.01		Escore	4.79	11.28	17.58
RV	36.52	47.47	54.45	100	RV	4.27	11.37	18.09
Wald	37.82	49.82	56.98		Wald	4.02	11.32	17.88
Escore	25.36	39.88	48.06		Escore	4.28	11.45	17.92

Tabela 4.20 Poder (%), y segue distribuição BEOI de parâmetro de mistura igual 15% - RESET 2.

Teste	RESET 2			n	Teste	RESET 1		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	14.42	25.60	33.10	20	RV	2.91	7.53	12.91
Wald	3.80	13.75	22.68		Wald	2.51	6.77	11.55
Escore	9.07	23.64	33.01		Escore	3.00	7.77	13.64
RV	29.81	41.68	48.19	40	RV	2.76	7.89	13.65
Wald	20.77	37.46	45.86		Wald	2.24	7.62	12.85
Escore	15.53	33.27	42.49		Escore	3.04	8.30	13.73
RV	28.91	39.29	45.94	60	RV	3.26	9.11	15.16
Wald	27.69	39.53	47.06		Wald	2.79	8.71	14.71
Escore	16.65	31.74	40.00		Escore	3.32	9.34	15.14
RV	24.81	35.47	42.39	80	RV	2.89	8.32	14.34
Wald	25.30	37.61	44.68		Wald	2.43	8.00	14.09
Escore	14.54	27.52	36.28		Escore	3.23	8.46	14.18
RV	21.17	31.19	37.75	100	RV	2.70	9.00	14.86
Wald	22.72	33.52	40.49		Wald	2.41	8.67	14.50
Escore	12.40	23.54	31.96		Escore	2.94	9.01	14.84

4.5.3 Poder em identificar omissão de variável importante - RESET 2

Tabela 4.21 Poder (%), omissão de variável importante - RESET 2.

Teste	RESET 2			n	Teste	RESET 1		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	5.54	15.09	25.14	20	RV	0.14	2.11	6.83
Wald	2.22	10.30	20.26		Wald	0.41	2.12	5.91
Escore	0.63	7.47	16.82		Escore	0.37	2.68	6.20
RV	21.08	51.31	66.95	40	RV	2.62	14.84	27.24
Wald	13.76	46.37	64.31		Wald	2.86	18.13	32.04
Escore	3.66	33.20	52.55		Escore	2.05	10.17	19.56
RV	55.96	79.22	86.92	60	RV	8.64	28.03	41.66
Wald	50.08	79.22	87.73		Wald	13.26	33.84	46.34
Escore	26.62	61.60	74.61		Escore	5.22	19.10	33.46
RV	77.05	88.93	92.42	80	RV	16.90	40.37	53.72
Wald	74.55	90.58	94.23		Wald	24.30	45.99	57.66
Escore	51.26	78.79	85.64		Escore	9.87	31.62	47.37
RV	86.97	93.49	95.50	100	RV	24.18	49.02	59.87
Wald	88.24	95.13	96.60		Wald	33.65	54.11	63.13
Escore	71.53	86.72	90.71		Escore	14.16	40.28	54.57

Como se pode observar na Tabela 4.21, o teste RESET 2, mais uma vez, se mostrou superior ao teste RESET 1 em termos de poder em identificar má especificação do modelo. Note que os poderes de ambos os testes crescem à medida em que n aumenta. Note também que o poder do teste RESET 2 crescem mais rapidamente que o poder do teste RESET 1.

Por fim, note que o teste RESET 2 se mostra mais poderoso do que o teste RESET 1. Quando o número de observações é pequeno, como por exemplo $n = 20$, o poder do teste RESET 2 chega a ser quase quatro vezes maior do que o poder do teste RESET 1 para o nível nominal de 10% utilizando o teste Wald, como se pode observar Tabela 4.21.

4.5.4 Poder em identificar má especificação da função de ligação para o submodelo da média - RESET 2

Como se pode observar nas Tabelas 4.22, 4.23, 4.24, 4.25 e 4.26, neste experimento, o teste RESET 2 apresentou melhor desempenho, em termos de poder do que o teste RESET 1 em todos os níveis de poluição (5%, 10%, 15%, 20% e 25%). Note ainda que à medida em que n cresce o poder do teste RESET 2 cresce mais rapidamente que o poder do teste RESET 1. Por fim, note que para $n = 100$, com níveis de poluição iguais a 20% (Tabela 4.25) e 25% (Tabela 4.26), o poder do teste RESET 2 chega a ser mais do que o dobro do poder do teste RESET 1.

Tabela 4.22 Poder (%), gerando y com 5% de poluição log-log complementar - RESET 2.

Teste	RESET 2			n	Teste	RESET 1		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	3.93	11.35	18.57	20	RV	2.91	8.87	15.20
Wald	1.54	5.95	12.06		Wald	2.74	8.76	14.88
Escore	4.98	12.51	19.09		Escore	2.79	8.78	14.98
RV	11.55	21.11	29.58	40	RV	5.21	14.45	22.23
Wald	4.74	14.85	24.07		Wald	4.54	14.19	21.96
Escore	11.58	23.46	31.69		Escore	5.67	14.77	22.26
RV	18.19	32.65	41.37	60	RV	9.57	21.93	30.10
Wald	10.96	25.82	38.22		Wald	8.92	21.99	30.19
Escore	19.98	33.27	41.87		Escore	9.67	22.10	30.07
RV	25.47	40.34	49.67	80	RV	11.55	26.08	35.91
Wald	18.33	36.57	48.10		Wald	10.83	26.13	36.26
Escore	25.76	39.73	49.11		Escore	12.40	26.04	36.08
RV	31.73	47.17	56.38	100	RV	14.38	30.32	40.95
Wald	25.21	44.34	55.23		Wald	14.14	30.28	41.17
Escore	32.68	46.54	55.83		Escore	14.75	30.42	40.68

Tabela 4.23 Poder (%), gerando y com 10% de poluição log-log complementar - RESET 2.

Teste	RESET 2			n	Teste	RESET 1		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	6.52	16.07	24.89	20	RV	5.29	14.12	22.24
Wald	1.79	6.67	13.47		Wald	4.29	13.08	20.71
Escore	9.26	20.24	28.63		Escore	5.19	13.82	22.17
RV	18.55	32.46	43.11	40	RV	10.94	25.83	36.04
Wald	6.85	20.27	32.12		Wald	9.09	25.30	35.79
Escore	21.27	38.23	47.90		Escore	11.80	26.20	36.17
RV	32.08	50.49	61.22	60	RV	20.15	37.99	48.10
Wald	18.88	40.61	55.95		Wald	19.27	38.22	48.50
Escore	35.77	54.08	63.13		Escore	20.06	37.77	47.63
RV	45.36	63.47	72.53	80	RV	25.10	43.23	52.82
Wald	32.15	58.20	70.06		Wald	24.06	43.23	53.14
Escore	46.43	64.48	72.89		Escore	25.79	42.87	52.63
RV	56.56	73.78	81.34	100	RV	28.90	47.21	55.89
Wald	46.02	69.73	79.74		Wald	29.04	47.52	56.05
Escore	58.61	74.17	81.27		Escore	28.96	46.80	55.49

Tabela 4.24 Poder (%), gerando y com 15% de poluição log-log complementar - RESET 2.

Teste	RESET 2			n	Teste	RESET 1		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	8.120	20.23	30.04	20	RV	7.39	20.14	29.68
Wald	2.23	7.27	14.53		Wald	5.68	17.39	27.51
Escore	13.24	27.30	37.00		Escore	7.48	19.83	29.68
RV	25.32	43.10	54.87	40	RV	17.83	37.37	47.76
Wald	7.86	23.80	38.93		Wald	15.39	36.77	47.42
Escore	31.88	52.76	61.93		Escore	18.80	37.44	47.54
RV	46.00	66.33	76.25	60	RV	29.27	47.41	56.20
Wald	25.74	54.21	70.54		Wald	28.46	47.65	56.28
Escore	52.16	70.32	78.10		Escore	28.95	47.01	55.74
RV	64.16	80.53	86.89	80	RV	33.42	48.34	56.22
Wald	46.61	75.77	85.12		Wald	33.01	48.67	56.50
Escore	66.12	81.35	86.84		Escore	33.56	47.84	55.61
RV	75.83	88.72	92.90	100	RV	34.53	47.75	54.05
Wald	65.35	86.46	92.14		Wald	34.92	47.99	54.33
Escore	78.00	88.81	92.74		Escore	34.35	46.98	53.57

Tabela 4.25 Poder (%), gerando y com 20% de poluição log-log complementar - RESET 2.

Teste	RESET 2			n	Teste	RESET 1		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	8.68	22.80	34.29	20	RV	9.47	26.12	37.58
Wald	2.47	7.70	15.12		Wald	6.41	22.12	34.81
Escore	16.03	34.76	45.24		Escore	9.89	25.91	37.39
RV	32.59	53.00	65.10	40	RV	24.33	43.58	53.75
Wald	7.67	27.59	45.83		Wald	21.69	42.83	53.39
Escore	43.06	64.64	72.96		Escore	25.07	43.31	53.51
RV	59.06	78.77	86.18	60	RV	34.23	48.67	55.46
Wald	31.75	66.45	81.63		Wald	33.87	49.16	55.78
Escore	66.38	82.65	87.99		Escore	33.26	47.81	54.86
RV	78.63	90.45	94.31	80	RV	35.17	45.82	52.35
Wald	59.94	87.40	93.23		Wald	35.09	46.30	52.47
Escore	80.26	90.77	94.27		Escore	34.94	45.17	51.86
RV	87.81	95.41	97.26	100	RV	32.92	42.51	48.61
Wald	79.73	94.45	97.23		Wald	33.06	42.56	48.55
Escore	89.19	95.40	97.11		Escore	32.38	42.10	48.31

Tabela 4.26 Poder (%), gerando y com 25% de poluição log-log complementar - RESET 2.

Teste	RESET 2			n	Teste	RESET 1		
	α					α		
	1%	5%	10%			1%	5%	10%
RV	9.100	24.45	36.96	20	RV	11.28	32.25	45.91
Wald	2.530	7.92	14.70		Wald	6.66	26.84	42.22
Escore	19.43	41.56	52.77		Escore	12.59	31.83	45.34
RV	39.69	61.29	74.04	40	RV	29.95	47.66	56.42
Wald	7.280	30.18	52.79		Wald	27.32	47.72	56.62
Escore	53.07	74.39	81.55		Escore	29.70	47.19	55.85
RV	70.54	86.55	92.29	60	RV	34.47	44.98	49.88
Wald	36.29	76.01	89.09		Wald	34.57	45.39	50.14
Escore	77.32	89.68	93.31		Escore	33.52	44.16	49.31
RV	87.45	95.26	97.55	80	RV	31.23	39.80	45.47
Wald	70.47	93.39	97.13		Wald	31.20	39.76	45.43
Escore	88.74	95.64	97.47		Escore	30.89	39.31	45.24
RV	94.31	98.18	98.98	100	RV	27.71	36.89	43.05
Wald	88.31	97.75	98.97		Wald	27.71	36.65	42.73
Escore	94.99	98.15	98.98		Escore	27.34	36.70	42.94

CAPÍTULO 5

Aplicação

Este capítulo apresenta aplicações a dados reais do teste RESET adaptado a modelos de regressão beta, tanto com dispersão constante quanto com dispersão fixa. Para estas aplicações foram utilizados dois conjuntos de dados bastante conhecidos: os dados de rendimento de gasolina de Prater (1956), composto por 32 observações, e os dados referentes a habilidade de leitura entre crianças disléxicas e não disléxicas de Smithson & Verkuilen (2006), composto por 44 observações. Estes conjuntos de dados estão disponíveis no pacote *betareg* do *software* R¹. A seguir serão apresentadas as aplicações com modelos de regressão beta com dispersão constante e, em seguida, as aplicações com modelos de regressão beta com dispersão variável.

5.1 Modelos de Dispersão Constante

Para as aplicações com modelos de dispersão constante foram utilizados os dados de Prater (1956), onde a variável dependente é **yield**, que é a proporção de óleo cru convertido em gasolina após a destilação e fracionamento. As variáveis explicativas são **temp**, que é a temperatura (em graus Fahrenheit) necessária para vaporizar toda a gasolina e **bach**, que é um fator que indica dez lotes, singulares, de condições dos experimentos. Como a variável **batch** é categórica, ela foi dividida em nove variáveis *dummy*, sendo utilizada a última categoria como referência.

A primeira aplicação consiste em testar se o seguinte modelo está corretamente especificado:

$$\log \left(\frac{\mu_t}{1 - \mu_t} \right) = \beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} + \beta_3 x_{t3} + \beta_4 x_{t4} + \beta_5 x_{t5} + \\ + \beta_6 x_{t6} + \beta_7 x_{t7} + \beta_8 x_{t8} + \beta_9 x_{t9} + \beta_{10} x_{t10},$$

$t = 1, \dots, 32$. Note que, a função de ligação utilizada no submodelo da média é logit. Este modelo já foi trabalhado por Ferrari & Cribari-Neto (2004) e Cribari-Neto & Zeileis (2010), e será chamado ‘Modelo 1’.

Para o teste de má especificação, foram utilizados os testes de razão de verossimilhanças, Wald e escore. Para formar o modelo estendido adicionou-se ao submodelo da média a variável $\hat{\eta}^2$, de forma que o teste consiste em verificar se o coeficiente desta variável possui significância estatística, ou seja, $H_0 : \beta_{11} = 0$ vs. $H_1 : \beta_{11} \neq 0$.

O teste rejeitou a hipótese nula, indicando assim que o Modelo 1 está incorretamente especificado. Os p -valores dos testes da razão de verossimilhanças, Wald e escore, foram $2.2053e - 006$, $2.3810e - 008$ e $5.1252e - 005$, respectivamente.

¹Para maiores informações, ver Cribari-Neto & Zeileis (2010).

Utilizando o mesmo conjunto de dados e as mesmas variáveis dependente e explicativas formulou-se o ‘Modelo 2’, já trabalhado por Cribari-Neto & Zeileis (2010), o qual difere do Modelo 1 apenas na função de ligação adotada para o submodelo da média, que agora será log-log. O novo modelo é dado por

$$-\log[-\log(\mu_t)] = \beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} + \beta_3 x_{t3} + \beta_4 x_{t4} + \beta_5 x_{t5} + \\ + \beta_6 x_{t6} + \beta_7 x_{t7} + \beta_8 x_{t8} + \beta_9 x_{t9} + \beta_{10} x_{t10}.$$

onde $t = 1, \dots, 32$.

Foram adotados os mesmos procedimentos de testes utilizados no Modelo 1, no entanto, desta vez os testes apontaram que não há problemas de má especificação no Modelo 2, tendo os testes de razão de verossimilhanças, Wald e escore apresentado os seguintes p -valores: 0.19664, 0.19093 e 0.20202, respectivamente. Assim, pode-se concluir que a função de ligação mais indicada para este conjunto de dados é a log-log.

5.2 Modelos de Dispersão Variável

Para as aplicações utilizando modelos de regressão beta com dispersão variável se foram utilizados os dados de Smithson & Verkuilen (2006), referentes à habilidade de leitura de crianças disléxicas e não disléxicas da Austrália. A variável independente (y) é o índice de precisão na leitura e as variáveis explicativas são: x_1 , que é uma variável dicotômica responsável por indicar se a criança possui dislexia ou não, assumindo valores 1 e -1 , respectivamente, o escore padronizado de QI não verbal, x_2 , e, por fim, a interação entre as duas covariáveis.

Esta aplicação consiste em testar se o modelo a seguir, denominado ‘Modelo 3’, está corretamente especificado:

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

onde $t = 1, \dots, 44$. Este modelo já foi trabalhado por Smithson & Verkuilen (2006) e Espinheira et al. (2008a).

Assim como nos modelos com dispersão constante, utilizou-se os testes de razão de verossimilhanças, Wald e escore para testar se há má especificação e foi adicionada ao submodelo da média a variável $\hat{\eta}^2$, a fim de se testar se o coeficiente associado a esta variável possui significância estatística, ou seja, para se testar $H_0 : \beta_4 = 0$ vs. $H_1 : \beta_4 \neq 0$.

O teste apontou que não se pode rejeitar a hipótese nula, de forma que o Modelo 3 estaria corretamente especificado. Os testes de razão de verossimilhanças, Wald e escore, apresentaram p -valores respectivamente iguais a 0.23748, 0.17587 e 0.29625.

O ‘Modelo 4’, já trabalhado por Cribari-Neto & Queiroz (2012), descrito a seguir, difere do Modelo 3 apenas no submodelo da dispersão, apresentando uma covariável a mais, x_2^2 , ou seja,

$$\log\left(\frac{\mu_t}{1 - \mu_t}\right) = \beta_0 + \beta_1 x_{t1} + \beta_2 x_{t2} + \beta_3 x_{t1} x_{t2},$$

$$\log(\phi_t) = \gamma_0 + \gamma_1 x_{t1} + \gamma_2 x_{t2} + \gamma_3 x_{t2}^2,$$

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

Assim como no Modelo 3, os testes de razão de verossimilhanças, Wald e escore para o Modelo 4 não rejeitaram a hipótese nula de que o modelo estaria corretamente especificado, apresentando p -valores iguais a 0.57670, 0.45820 e 0.68367, respectivamente.

Os Modelos 3 e 4 foram utilizados em outra forma de teste, na qual além de se adicionar $\hat{\eta}^2$ ao submodelo da média no modelo estendido, adiciona-se também $\hat{\delta}^2$ ao submodelo da precisão. Posteriormente, testa-se a exclusão conjunta de $\hat{\eta}^2$ e $\hat{\delta}^2$, como foi discutido no capítulo anterior. No que tange ao Modelo 3, as conclusões inferenciais foram semelhantes às anteriores, ou seja, não rejeitaram a hipótese de que o modelo está corretamente especificado. O p -valor dos testes da razão de verossimilhanças, Wald e escore foram 0.49771, 0.40007 e 0.57955, respectivamente. No entanto, a conclusão anterior não se manteve para o Modelo 4. Os p -valores dos testes de razão de verossimilhanças, Wald e escore são iguais a 0.040122, 0.0063946 e 0.19249, respectivamente. Note que para este novo teste apenas o teste escore não rejeita a hipótese nula para níveis nominais de 5% e 10%, assim, de acordo com os testes de razão de verossimilhanças e Wald, o Modelo 4 estaria mal especificado.

Considerações Finais

O presente trabalho tem por objetivo avaliar o poder do teste de má especificação proposto por Cribari-Neto & Lima (2007) em diversos cenários que configuram má especificação. A saber: função de ligação incorreta, presença de *outlier* na amostra, omissão de variável importante, estimação com dispersão constante quando o modelo verdadeiro possui dispersão variável (e vice-versa) e o caso em que há má especificação da distribuição da variável resposta. Adicionalmente, foi introduzido o teste RESET 2, o qual teve seu poder avaliado em diferentes cenários, sempre comparando-se seu desempenho ao do teste RESET 1.

Para o cenário em que há má especificação da distribuição da variável resposta, de forma geral o teste RESET 1 não apresentou alto poder, mas vale ressaltar que o teste apresenta maior poder quando se utilizam modelos de regressão beta com dispersão variável.

Levando em consideração o cenário em que a má especificação se configura com a omissão de variável importante o poder do teste RESET 1 para $n = 20$ é baixo, no entanto, à medida em que n aumenta o poder cresce rapidamente, passando a ter desempenho satisfatório. Neste cenário o teste RESET 1 apresenta maior poder, quando são usados modelos de regressão beta com dispersão fixa.

No cenário de má especificação em que há *outlier* na amostra o teste RESET 1 apresentou alto poder quando foram utilizados modelos de regressão beta com dispersão fixa, no entanto, não apresentou poder satisfatório com modelos de regressão beta com dispersão variável.

Para o cenário de erro no submodelo da precisão o teste RESET 1 apresentou alto poder, sob o Erro 2, quando o teste score é utilizado. Utilizando os demais testes (razão de verossimilhanças e Wald), o teste RESET 1 apresenta baixo poder tanto sob o Erro 1 como sob o Erro 2. Vale destacar que quando utilizados os testes da razão de verossimilhanças e Wald, o teste RESET 1 apresenta maior poder sob o Erro 1.

O último cenário analisado foi aquele em que a má especificação se dá através da escolha incorreta da função de ligação. Neste caso o teste apresentou baixo poder quando o grau de “poluição” é pequeno, no entanto, à medida em que n cresce o poder tende a 1. Aqui, o teste apresentou maior poder quando se utilizam modelos de regressão beta com dispersão variável.

Levando-se em consideração tamanho, as simulações realizadas indicaram que o teste RESET 1 tem melhor desempenho que o teste RESET 2. Já no comparativo entre os poderes dos testes RESET 1 e RESET 2, em todos os cenários analisados o teste RESET 2 apresentou poder superior ao teste RESET 1.

Os resultados apresentados na presente dissertação indicam que o melhor teste a ser usado é aquele que chamamos de RESET 2, pois ele é tipicamente mais poderoso do que o teste RESET 1. Notamos, contudo, que o teste RESET 2 geralmente apresenta maiores distorções de tamanho. Assim, quando o tamanho da amostra é pequeno ele deve ser usado, idealmente,

em conjunto com esquema *bootstrap*. É recomendável ainda que a inferência seja realizada via teste escore.

Referências Bibliográficas

- [1] Buse, A. The Cyclical Behaviour of the Size Distribution of Income in Canada: 1947 – 78. *Canadian Journal of Economics* 15 (1982), 189 – 204.
- [2] Cordeiro, G. M. *Introdução à Teoria Assintótica*. Rio de Janeiro: Instituto de Matemática Pura e Aplicada, 1999.
- [3] Cribari-Neto, F.; Lima, L. B. A misspecification test for beta regressions. Texto para discussão, 2007.
- [4] Cribari-Neto, F.; Queiroz, M. P. F. On testing inference in beta regressions. *Journal of Statistical Computation and Simulation* 2012, 1 – 18.
- [5] Cribari-Neto, F.; Zeileis, A. Beta regression in R. *Journal of Statistical Software* 34 (2) (2010).
- [6] Davidson, R.; MacKinnon, J. G. *Estimation and Inference in Econometrics*. Oxford University Press, 1993.
- [7] Doornik, J. A. *Ox: Object-Oriented Matrix Language*, 4th ed. London: Timberlake Consultants Press and Oxford: [//www.doornik.com](http://www.doornik.com), 2009.
- [8] Ferrari, S. L. P.; Cribari-Neto, F. Beta regression for modeling rates and proportions. *Journal of Applied Statistics*, 31(2004), 799 – 815.
- [9] Frery, A. C.; Cribari-Neto, F. Elementos de Estatística Computacional Usando Plataformas de Software Livre/Gratuito. *Publicações Matemáticas, Instituto de Matemática Pura e Aplicada - IMPA*, 2005.
- [10] Jones, M. C. Kumaraswamy’s distribution: A beta-type distribution with some tractability advantages. *Statistical Methodology* 6 (2009) 70 – 81.
- [11] Lima, L. B. *Um Teste de Especificação Correta em Modelos de Regressão Beta*. UFPE: Dissertação (2007).
- [12] McCullagh, P.; Nelder, J. A. *Generalized Linear Models*. 2nd edition. London: Chapman & Hall, 1989.
- [13] Ospina, R.; Ferrari, S. L. P. A general class of zero-or-one inflated beta regression models. *Computational Statistics and Data Analysis* 56 (2012), 1609 – 1623.

- [14] Ospina, R.; Ferrari, S. L. P. Inflated beta distributions. *Statistical Papers* 51 (2010), 111 – 126.
- [15] Prater, N. H.,(1965).“Estimate Gasoline Yields from Crudes”. *Petroleum Refiner*, **35**(5), 236 – 238.
- [16] Queiroz, M. P. F. *Teste de Hipóteses em Regressão Beta Baseados em Verossimilhança Perfilada Ajustada e em Bootstrap*. UFPE: Dissertação (2011).
- [17] Ramsey, J. B. Test for specification erros in classical linear least square regression analysis. *Journal of the Royal Statistical Society B* 31 (1969), 350 – 371.
- [18] Serfling, R. J. *Approximation Theorems of Mathematical Statistics*. John Wiley & Sons, 1980.
- [19] Simas, A. B.; Barreto-Souza, W.; Rocha, A. V. Improved estimators for a general class of beta regression models. *Computational Statistics e Data Analysis*, 54(2010), 348 – 366.
- [20] Smithson, M.; Verkuilen, J. *A better lemon squeezer? Maximum-likelihood regression with beta-distributed dependent variable*, *Psychol. Methods* 11 (2006), pp. 54 – 71.

Otimização Não-Linear

Em estatística a otimização não-linear é utilizada, entre outras aplicações, para a obtenção de estimativas de máxima verossimilhança, pois em muitos casos de interesse o estimador de máxima verossimilhança não possui forma fechada, sendo portanto necessária a resolução de sistemas não-lineares por meio de otimização numérica. Em geral a otimização de funções não-lineares pode ser algo não trivial. Tomemos como exemplo a seguinte equação:

$$\Lambda(\theta) = 10 + \beta' \theta + \frac{1}{2} \theta' \Gamma \theta,$$

onde β é um vetor de dimensão $p \times 1$ e Γ é uma matriz positiva-definida¹.

Uma possível forma de resolver esse problema é por meio da utilização de algoritmos iterativos, que geralmente possuem a seguinte forma:

$$\theta_{t+1} = \theta_t + \lambda_t \Delta_t,$$

onde se inicia em θ_0 , e a cada θ_t deve-se definir o “tamanho do passo” (λ_t) e a “direção do passo” (Δ_t). Idealmente, deve-se a cada iteração definir um tamanho ótimo para o tamanho do passo por meio de uma otimização secundária chamada “procura em linha”. No entanto, essa otimização pode ser muito custosa computacionalmente. Essa otimização secundária pode ser apresentada da seguinte forma:

$$\frac{\partial \Lambda(\theta_t + \lambda_t \Delta_t)}{\partial \lambda_t} = f(\theta_t + \lambda_t \Delta_t)' \Delta_t = 0.$$

Para decidir qual será a direção do passo, em geral se utilizam métodos gradiente, que possuem a seguinte forma:

$$\Delta_t = M_t f_t,$$

onde M_t é uma matriz positiva-definida e f_t é o gradiente de Λ . Existem diversos métodos de otimização (Newton-Raphson e BHHH, por exemplo), a principal diferença entre eles reside na forma como é escolhida M_t .

Nos métodos quasi-Newton gera-se uma sequência de matrizes:

$$M_{t+1} = M_t + E_t,$$

¹Uma matriz é dita ser positiva-definida se $\forall x \neq 0, x' W x > 0$, onde W é uma matriz $p \times p$ e x é um vetor $p \times 1$.

onde E_t é uma matriz positiva-definida. Assim, se M_0 for positiva-definida, então toda a sequência de matrizes também o será. Usualmente, toma-se $M_0 = I$. A partir de um número grande de iterações é possível conseguir uma boa aproximação para $-H^{-1}$, de forma que

$$\lim_{t \rightarrow \infty} M_t = -H^{-1}.$$

Desenvolvido por Broyden, Fletcher, Goldfarb, Shanno, o método BFGS é considerado o método quasi-Newton mais robusto e é o mais utilizado na prática.

Aqui,

$$M_{t+1} = M_t + \frac{\delta_t \delta_t'}{\delta_t' v_t} + \frac{M_t v_t v_t' M_t}{v_t' M_t \delta_t} - v_t' M_t v_t \left(\frac{\delta_t}{\delta_t' v_t} - \frac{M_t v_t}{v_t' M_t v_t} \right) \left(\frac{\delta_t}{\delta_t' v_t} - \frac{M_t v_t}{v_t' M_t v_t} \right)'.$$

onde $\delta_t = \theta_{t+1} - \theta_t$ e $v_t = f(\theta_{t+1}) - f(\theta_t)$.

Uma vantagem dos métodos quasi-Newton reside no fato de que não requerem o cálculo de segundas derivadas e nem a inversão da matriz hessiana, que podem ser muito custosos computacionalmente.

APÊNDICE B

Programa - Simulação de Tamanho

Abaixo se encontra um dos programas utilizados nas simulações de tamanho desta dissertação. Este programa foi utilizado no experimento onde há omissão de variável regressora importante. O programa foi escrito na linguagem Ox (Doornik, 2009).

```
/* _____  
  
PROGRAMA: Tamanho_variavel-importante.ox  
  
USO: Realizar simulação Monte Carlo para avaliar numericamente  
o desempenho, no que diz respeito a tamanho, do teste de má  
especificação proposto por Cribari-Neto & Lima (2007).  
  
AUTOR: José Sérgio Casé de Oliveira  
  
OBS.: Este programa foi desenvolvido a partir dos programas  
apresentados nas dissertações de mestrado de Lima (2007) e  
Queiroz (2011).  
  
DATA: 10/12/2012.  
  
_____*/  
  
/* Arquivos de cabeçalho */  
#include<oxstd.h>  
#include<oxprob.h>  
#import<maximize>  
  
/* Variaveis globais */  
static decl yfuncF;  
static decl yfuncV;  
static decl yfuncR;  
static decl yfuncRv;  
static decl s_mX;
```

```

static decl s_mXr;
static decl s_mXrv;
static decl s_mZ;

const decl nobs = 20; /* Tamanho Amostral, usar 20,40,60,80,100 */
const decl nrep = 10000; // # de replicas de monte carlo
const decl k = 3; // # de betas
const decl m = 3; // # de gamas
const decl r = 4; // # de betas usando s_mXr

/* Funcao de log-verossimilhanca - disp. fixa */
floglikFixed(const vPF, const adFunc, const avScore, const amHess)
{
decl k = rows(vPF) - 1;
decl eta = s_mX*vPF[0:(k-1)];
decl mu = exp(eta) ./ (1.0+exp(eta));
decl phi = vPF[k];
decl ynew = log( yfuncF ./ (1.0-yfuncF) );
decl munew = polygamma(mu*phi, 0) - polygamma((1.0-mu)*phi, 0);
decl T = diag( mu .* (1 - mu) );

adFunc[0] = double( sumc( loggamma(phi) - loggamma(mu.*phi)-
loggamma((1-mu).*phi) +(mu.*phi-1) .* log(yfuncF) + ((1-mu).*phi-1)
.* log(1-yfuncF)));

if (avScore)
{
(avScore[0])[0:(k-1)] = phi*s_mX'*T*(ynew-munew);
(avScore[0])[k] = double(sumc( polygamma(phi, 0) - mu .*
polygamma(mu*phi, 0) - (1.0-mu) .* polygamma((1.0-mu)*phi, 0)
+ mu .* log(yfuncF) + (1.0-mu) .* log(1.0-yfuncF) ));
}
if( isnan(adFunc[0]) || isdotinf(adFunc[0]) )
return 0;
else
return 1;
}

/* Função de log-verossimilhança - disp. variavel */
floglikVar(const vPV, const adFunc, const avScore, const amHess)
{
decl k = rows(vPV) - 3;
decl m = rows(vPV) - 3;
decl eta = s_mX*vPV[0:(k-1)];
decl mu = exp(eta)./(1 + exp(eta));

```

```

decl delta = s_mZ*vPV[k:(k+m-1)];
decl phi = exp(delta);
decl ystar = log(yfuncV./(1 - yfuncV));
decl mustar = polygamma(mu.*phi,0)-polygamma((1-mu).*phi,0);
decl ydag = log(1 - yfuncV);
decl mudag = polygamma((1 - mu) .* phi, 0) - polygamma(phi, 0) ;
decl T = diag( mu .* (1 - mu) );
decl F = diag(phi);
decl H = diag(phi);
decl M = diag(mu);

adFunc[0] = double(sumc( loggamma(phi) - loggamma(mu .* phi) -
loggamma((1 - mu) .* phi) + (mu .* phi - 1) .* log(yfuncV) +
((1 - mu).*phi - 1) .* log(1 - yfuncV)));

if(avScore)
{
(avScore[0])[0:(k-1)] = s_mX'*T*F*(ystar-mustar);
(avScore[0])[k:(k+m-1)] = s_mZ'*H*(M*(ystar-mustar)+(ydag-mudag));
}
if( isnan(adFunc[0]) || isdotinf(adFunc[0]) )
return 0;
else
return 1;
}

/* Função de log-verossimilhança para teste RESET - disp. fixa */
floglikR(const vP, const adFunc, const avScore, const amHess)
{
decl r = rows(vP) - 1;
decl eta = s_mXr*vP[0:(r-1)];
decl mu = exp(eta)./(1.0+exp(eta));
decl phi = vP[r];
decl ynew = log( yfuncR./(1.0-yfuncR) );
decl munew = polygamma(mu*phi, 0) - polygamma((1.0-mu)*phi, 0);
decl T = diag(exp(eta)./(1.0+exp(eta)).^2);

adFunc[0] = double(sumc(loggamma(phi) - loggamma(mu.*phi)-
loggamma((1-mu).*phi) + (mu.*phi-1) .* log(yfuncR) +
((1-mu).*phi-1) .* log(1-yfuncR)));

if (avScore)
{
(avScore[0])[0:(r-1)] = phi*s_mXr'*T*(ynew-munew);
(avScore[0])[r] = double(sumc( polygamma(phi, 0) - mu .*
polygamma(mu*phi, 0) - (1.0-mu) .* polygamma( (1.0-mu)*phi, 0) +

```

```

mu .* log(yfuncR) + (1.0-mu) .* log(1.0-yfuncR) ));
}
if( isnan(adFunc[0]) || isdotinf(adFunc[0]) )
return 0;
else
return 1;
}

/* Função de log-verossimilhança para teste
   RESET - disp. variavel */
floglikRv(const vtheta, const adFunc, const avScore, const amHess)
{
decl r = rows(vtheta) - 3;
decl m = rows(vtheta) - 4;
decl eta = s_mXrv*vtheta[0:(r-1)];
decl mu = exp(eta)./(1 + exp(eta));
decl delta = s_mZ*vtheta[r:(r+m-1)];
decl phi = exp(delta);
decl ystar = log(yfuncRv./(1 - yfuncRv));
decl mustar = polygamma(mu.*phi,0)-polygamma((1-mu).*phi,0);
decl ydag = log(1 - yfuncRv);
decl mudag = polygamma((1 - mu) .* phi, 0) - polygamma(phi, 0) ;
decl T = diag(exp(eta)./(1 + exp(eta)).^2);
decl F = diag(phi);
decl H = diag(phi);
decl M = diag(mu);

adFunc[0] = double(sumc( loggamma(phi) - loggamma(mu .* phi) -
loggamma((1 - mu) .* phi) +(mu .* phi - 1) .* log(yfuncRv) +
((1 - mu).*phi - 1) .* log(1 - yfuncRv)));

if(avScore)
{
(avScore[0])[0:(r-1)] = s_mXrv'*T*F*(ystar-mustar);
(avScore[0])[r:(r+m-1)] = s_mZ'*H*(M*(ystar-mustar)+(ydag-mudag));
}
if( isnan(adFunc[0]) || isdotinf(adFunc[0]) )
return 0;
else
return 1;
}

main()
{

```



```

decl ir1,ir2,vp1,vp2, dfunc1,dfunc2,dExecTime,falhaF,
falhaV,mle1,mle2,est1,est2,j,beta0,gama0,emvF,emvV,
gama1,gama2,beta1,beta2,phir,theta1,theta2,etar,mur,
prF,a,qrF,prV,qrV,ynewF,ynewV,betaolsF,expfit,betaolsV,
deltar,q, phirV,olsfittednew,olsererrorvar,olsfitted,
thetainiF,mleRv,thetainiV,gamaols,txR1,mleR,falhaR,
vpR,irR,dfuncR,emvR,estR,statR,testeR1,testeR5,
testeR10,txR5,txR10,falhaRv,dfuncRv,vpRv,irRv,emvRv,
estRv,statRv,testeRv1,testeRv5,txRv1,txRv5,txRv10,
phi_iniF,phi_iniV,olsfittednewV, olsererrorvarV,expfitV,
testeRv10,olsfittedV,betaolsRF,olsfittednewRf,
olsererrorvarRf,expfitRf,olsfittedRf,betaolsRV,
olsfittednewRv,expfitRv,olsfittedRv,phi_iniRF,phi_iniRV,
olsererrorvarRv;

```

```

decl statWF,testeWf1,testeWf5,testeWf10,txWf1,txWf5,
txWf10,thetahatWf,etahat,muhat,phihat,That,psihat,
psi2hat,What,Dhat,vchat,tempinv,ghat,K1hat,K2hat,
fisherinv,statWV,statSf,statSv;

```

```

decl thetahatSf,etatil,mutil,phitil,psitil1,psitil2,
Ttil,Wtil,vctil,Dtil,tempinvtil,gtil,K1til,K2til,
fisherinvtil,mustartil,escoretil;

```

```

decl vtheta1,etatilV,mutilV,TtilV,deltatilV,phitilV,
FtilV,HtilV,Mtil,mustartilV,escorebetatilV,VdagtilV,
CovtilV,VstartilV,KbetabetatilV,KbetagamatilV,
KgamabetatilV,KgamagamatilV,fishertilV,fisherinvtilV;

```

```

decl etahatv,muhatv,deltahatv,phihatv,Tv,Hv,Mv,Fv,
Vstar,Vdag,Cov,Kbetabeta,Kbetagama,Kgamagama,Kgamabeta,
fisher2,fisherinv2;

```

```

decl testeWv1,testeWv5,testeWv10,txWv1,txWv5,txWv10,
testeSf1,testeSf5,testeSf10,txSf1,txSf5,txSf10,testeSv1,
testeSv5,testeSv10,txSv1,txSv5,
txSv10,ops,ops2;

```

```

/* Inicio da contagem do tempo */
dExecTime = timer();
ranseed(1750); // semente fixada, gerador default

```

```

/* Valores verdadeiros dos parametros */
beta0 = -1.0;

```

```

beta1 = 0.5;
beta2 = 0.3;
gama0 = 1.5;
gama1 = 3.0;
gama2 = 2.5;
phir = 120;

/* Vetor de parametros */
theta1 = beta0|beta1|beta2|phir;
theta2 = beta0|beta1|beta2|gama0|gama1|gama2;

/* Algumas inicializacoes */
mle1 = zeros(rows(theta1),nrep);
mle2 = zeros(rows(theta2),nrep);
mleR = zeros(5,nrep);
mleRv = zeros(7,nrep);
emvF = zeros(1,nrep);
emvV = zeros(1,nrep);
emvR = zeros(1,nrep);
emvRv = zeros(1,nrep);
statWF = zeros(1,nrep);
statWV = zeros(1,nrep);
statSf = zeros(1,nrep);
statSv = zeros(1,nrep);

//MaxControl(-1,1,1);

a = 1~(ranu(20,k-1));

if(nobs == 20){s_mX = a;}
    if(nobs == 40){s_mX = a|a;}
if(nobs == 60){s_mX = a|a|a;}
if(nobs == 80){s_mX = a|a|a|a;}
if(nobs == 100){s_mX = a|a|a|a|a;}

etar = s_mX * theta1[0:k-1];
mur = exp(etar)./(1.0 + exp(etar));

s_mZ = s_mX;
deltar = s_mZ * theta2[(k):k+m-1];
phirV = exp(deltar);
q = columns(s_mZ);

/* Calculando lambda */

```

```

decl th, phi_t, lambda, lambda_2, var_y, y_dataf, y_datav;
th = gama0|gama1|gama2;
phi_t = exp(s_mZ*th);
lambda = maxc(phi_t) / minc(phi_t);

var_y = (mur .* (1 - mur)) ./ (1 + phirV);
lambda_2 = maxc(var_y) / minc(var_y);

/* Contadores de falhas */
falhaF = 0;      // fixo
falhaV = 0;      // var
falhaR = 0;      // RESET disp. fixa
falhaRv = 0;     // RESET disp. variavel

for(j=0; j <nrep; j++)          //Loop de Monte Carlo//
{

decl nobsmenosk = nobs - k;
decl nobsmenosr = nobs - r;

    // Grando yfunc para estimacao com disp. fixa
    prF = (mur .* phir);
    qrF = ((1 - mur) .* phir);
    yfuncF = ranbeta(nobs,1,prF,qrF); //linhas, colunas , p e q

    ynewF = log(yfuncF ./ (1.0 - yfuncF));

    /* Calculo do vetor inicial para Maximizacao da
       log-verossimilhanca - disp. fixa*/
    if(k > 1)
    {
        ols2c(ynewF, s_mX, &betaolsF);
    }
    else if(k == 1)
    {
        betaolsF = meanc(ynewF);
    }

    olsfittednew = s_mX*betaolsF;
    olserrorvar = sumsqrc(ynewF - olsfittednew)./(nobsmenosk);
    expfit = exp(olsfittednew);
    olsfitted = expfit ./ (1.0 + expfit);

    phi_iniF = (meanc(1 ./ (olserrorvar'*(olsfitted .*
    (1.0 - olsfitted)))) - 1.0);

```

```

vp1 = betaolsF|phi_iniF;

/* Maximizacao da log-verossimilhanca - disp. fixa */
irl = MaxBFGS(floglikFixed, &vp1, &dfunc1, 0, 0);

if( irl != MAX_CONV && irl != MAX_WEAK_CONV )
{
falhaF++;
j--;
continue;
}
else
{
mle1[][j] = vp1;
emvF[][j] = dfunc1;
}

ops = s_mX*vp1[0:2][];
s_mXr = s_mX~((ops).^2);

// Grando yfunc para estimacao com disp. variável
prV = (mur .* phirV);
qrV = ((1 - mur) .* phirV);
yfuncV = ranbeta(nobs,1,prV,qrV); //linhas, colunas , p e q

ynewV = log(yfuncV ./ (1.0 - yfuncV));

/* Calculo do vetor inicial para Maximizacao da
log-verossimilhanca - disp. variavel*/
if(k > 1)
{
ols2c(ynewV, s_mX, &betaolsV);
}
else if(k == 1)
{
betaolsV = meanc(ynewV);
}

olsfittednewV = s_mX*betaolsV;
olsererrorvarV = sumsqrc(ynewV - olsfittednewV)./(nobsmenosk);
expfitV = exp(olsfittednewV);
olsfittedV = expfitV ./ (1.0 + expfitV);

```

```

phi_iniV = log((meanc(1 ./ (olerrorvarV'*(olsfittedV .*
(1.0 - olsfittedV)))) - 1.0))|0|0;

vp2 = betaolsV|phi_iniV;

/* Maximizacao da log-verossimilhanca - disp. var */
ir2 = MaxBFGS(floglikVar, &vp2, &dfunc2, 0, 0);

if( ir2 != MAX_CONV && ir2 != MAX_WEAK_CONV )
{
falhaV++;
j--;
continue;
}
else
{
mle2[][j] = vp2;
emvV[][j] = dfunc2;
}

ops2 = s_mX*vp2[0:2][];
s_mXrv = s_mX~((ops2).^2);

/* Calculo do vetor inicial para Maximizacao da log-verossimilhanca
teste RESET - disp. fixa*/
if(r > 1)
{
ols2c(ynewF, s_mXr, &betaolsRF);
}
else if(r == 1)
{
betaolsRF = meanc(ynewF);
}

olsfittednewRf = (s_mXr)*betaolsRF;
olerrorvarRf = sumsqr(ynewF - olsfittednewRf)./(nobsmenosr);
expfitRf = exp(olsfittednewRf);
olsfittedRf = expfitRf ./ (1.0 + expfitRf);

phi_iniRF = (meanc(1 ./ (olerrorvarRf'*(olsfittedRf .*
(1.0 - olsfittedRf)))) - 1.0);

```

```

/* Calculo do vetor inicial para Maximizacao da log-verossimilhanca
    teste RESET - disp. variavel*/
if(r > 1)
{
    ols2c(ynewV, s_mXrv, &betaolsRV);
}
else if(r == 1)
{
    betaolsRV = meanc(ynewV);
}

olsfittednewRv = (s_mXrv)*betaolsRV;
    olserrorvarRv = sumsqrc(ynewV - olsfittednewRv)./(nobsmenosr);
expfitRv = exp(olsfittednewRv);
    olsfittedRv = expfitRv ./ (1.0 + expfitRv);

phi_iniRV = log((meanc(1 ./ (olserrorvarRv'*(olsfittedRv .*
(1.0 - olsfittedRv)))) - 1.0))|0|0;

yfuncR = yfuncF;

yfuncRv = yfuncV;

/* Chute inicial dos parametros */
vpR = betaolsRF|phi_iniRF;
vpRv = betaolsRV|phi_iniRV;

/* Armazenando o chute inicial */
thetainiF = vp1;
thetainiV = vp2;

/***** TESTE ESCORE DISP. FIXA *****/

thetahatSf = vp1[0:k-1]|0|vp1[k];

etatil = s_mXr * (thetahatSf[0:r-1]);
mutil = exp(etatil) ./ (1+exp(etatil));
phitil = thetahatSf[k+1];
psitil1 = polygamma(mutil*phitil, 1);
psitil2 = polygamma((1.0-mutil)*phitil, 1);
Ttil = diag( exp(etatil) ./ (1.0+exp(etatil)) .^2 );

```

```

Wtil = diag(phitil*(psitil1+psitil2)) * Ttil.^2;
vctil = phitil*(psitil1.*mutil-psitil2.*(1-mutil));
Dtil = diag(psitil1.*(mutil.^2)+psitil2.*(1.0-mutil).^2
-polygamma(phitil,1));
tempinvtil = invrtsym(s_mXr'*Wtil*s_mXr);
gtil = trace(Dtil)-(1.0/phitil)*vctil'*Ttil'*s_mXr*
tempinvtil*s_mXr'*Ttil*vctil;
K1til = tempinvtil*(gtil*unit(r)+(1/phitil)*s_mXr'*
Ttil*vctil*vctil'*Ttil'*s_mXr*tempinvtil);
K2til = -tempinvtil*s_mXr'*Ttil*vctil;
fisherinvtil = (1/(phitil*gtil))*((K1til~K2til)|(-vctil'*
Ttil'*s_mXr*tempinvtil~phitil));

mustartil = polygamma(mutil*phitil, 0) - polygamma((1.0-
mutil)*phitil, 0);
escoretil = phitil*(s_mXr[][r-1:]')*Ttil*(ynewF - mustartil);

statSf[][j] = (escoretil)'*(fisherinvtil[r-1][r-1])*escoretil;

/*****/

/***** TESTE ESCORE DISP. VAR *****/

vthetal = vp2[0:r-2]|0|vp2[r-1:r+1];

etatilV = s_mXrv*(vthetal[0:r-1-1]|zeros(1,1));
mutilV = exp(etatilV) ./ (1.0+exp(etatilV));
TtilV = diag( mutilV .* (1.0-mutilV) );
deltatilV = s_mZ*(vthetal[r:r+m-1]);
phitilV = exp(deltatilV);
FtilV = diag(phitilV);
HtilV = diag(phitilV);
Mtil = diag(mutilV);
mustartilV = polygamma(mutilV .* phitilV, 0) -
polygamma((1.0 - mutilV) .* phitilV, 0);
escorebetatilV = (s_mXrv'*FtilV*TtilV)*(ynewV-mustartilV);
VdagtilV = diag(polygamma((1.0 - mutilV) .* phitilV, 1)-
polygamma(phitilV, 1));
CovtilV = diag(-polygamma((1.0 - mutilV) .* phitilV, 1));
VstartilV = diag(polygamma(mutilV .* phitilV, 1) +
polygamma((1.0 - mutilV) .* phitilV, 1));
KbetabetatilV = s_mXrv'*FtilV*TtilV*VstartilV*TtilV*
FtilV*s_mXrv;
KbetagamatilV = s_mXrv'*FtilV*(Mtil*VstartilV+CovtilV)*
TtilV*HtilV*s_mZ;

```

```

KgamabetatilV = KbetagamatilV';
KgamagamatilV = s_mZ'*HtilV*(Mtil*VstartilV*Mtil+
(Mtil+Mtil)*CovtilV+VdagtilV)*HtilV*s_mZ;
fishertilV = (KbetabetatilV~KbetagamatilV)|
(KgamabetatilV~KgamagamatilV);
fisherinvtilV = invert(fishertilV);

statSv[][j] = (escorebetatilV[r-1:][])'*
(fisherinvtilV[r-1:r-1][r-1:r-1])*(escorebetatilV[r-1:][]);

/*****

/* Maximizacao da log-verossimilhanca teste
      RESET - disp. fixa */

irR = MaxBFGS(floglikR, &vpR, &dfuncR, 0, 0);

if( irR != MAX_CONV && irR != MAX_WEAK_CONV )
{
falhaR++;
j--;
continue;
}
else
{
mleR[][j] = vpR;
emvR[][j] = dfuncR;
}

/***** TESTE WALD DISP. FIXA *****/

etahat = s_mXr * vpR[0:r-1];
muhat = exp(etahat) ./ (1 + exp(etahat));
phihat = vpR[r];
That = diag( exp(etahat) ./ (1.0+exp(etahat)) .^2 );
psihat = polygamma(muhat*phihat, 1);
psi2hat = polygamma((1.0-muhat)*phihat, 1);
What = diag(phihat*(psihat+psi2hat)) * That .^2;
Dhat = diag(psihat.*(muhat.^2)+psi2hat.*(1.0-muhat).^2-
polygamma(phihat,1));
vchat = phihat*(psihat.*muhat-psi2hat.*(1-muhat));
tempinv = invertsym(s_mXr'*What*s_mXr);
ghat = trace(Dhat)-(1.0/phihat)*vchat'*That'*s_mXr*

```



```

tempinv*s_mXr'*That*vchat;

K1hat = tempinv*(ghat*unit(r)+(1/phihat)*s_mXr'*That*vchat*
vchat'*That'*s_mXr*tempinv);
K2hat = -tempinv*s_mXr'*That*vchat;
fisherinv = (1/(phihat*ghat))*((K1hat~K2hat)|(-vchat'*
That'*s_mXr*tempinv~phihat));

statWF[][j] = vpR[r-1]'.*(fisherinv[r-1][r-1])^(-1)*vpR[r-1];

/*****

/* Maximizacao da log-verossimilhanca teste
      RESET - disp. var */

irRv = MaxBFGS(floglikRv, &vpRv, &dfuncRv, 0, 0);

      if( irRv != MAX_CONV && irRv != MAX_WEAK_CONV )
      {
falhaRv++;
j--;
continue;
}
else
{
      mleRv[][j] = vpRv;
      emvRv[][j] = dfuncRv;
}

/***** TESTE WALD DISP. VAR *****/

etahatv = s_mXrv * vpRv[0:r-1];
muhatv = exp(etahatv) ./ (1.0 + exp(etahatv));
deltahatv = s_mZ * vpRv[r:r+m-1];
phihatv = exp(deltahatv);
Tv = diag(muhatv .* (1.0 - muhatv));
Hv = diag(phihatv);
Mv = diag(muhatv);
Fv = diag(phihatv);
Vstar = diag(polygamma(muhatv .* phihatv ,1) +
polygamma((1.0 - muhatv) .* phihatv,1));
Vdag = diag(polygamma((1.0 - muhatv) .* phihatv,1) -
polygamma(phihatv,1));
Cov = diag(-polygamma((1.0 - muhatv) .* phihatv,1));

```

```

Kbetabeta = s_mXrv'*Fv*Tv*Vstar*Tv*Fv*s_mXrv;
Kbetagama = s_mXrv'*Fv*(Mv*Vstar+Cov)*Tv*Hv*s_mZ;
Kgamagama = s_mZ'*Hv*(Mv*Vstar*Mv+(Mv+Mv)*Cov+Vdag)*Hv*s_mZ;
Kgamabeta = Kbetagama';
fisher2 = (Kbetabeta~Kbetagama)|(Kgamabeta~Kgamagama);
fisherinv2 = invert(fisher2);

statWV[][j] = vpRv[r-1:r-1]'*
(fisherinv2[r-1:r-1][r-1:r-1])^(-1)*vpRv[r-1];

/*****/

} //fim do MC

est1 = meanr(mle1);
est2 = meanr(mle2);
estR = meanr(mleR);
estRv = meanr(mleRv);

// TESTE DA RAZAO DE VEROSSIMILHANÇA - caso disp. fixa
statR = -2.*(emvF - emvR);

testeR1 = (statR[0][] .> 6.63);
testeR5 = (statR[0][] .> 3.84); //quiquadrado c\ 1 grau de liberdade
testeR10 = (statR[0][] .> 2.71);

txR1 = (sumr(testeR1)/nrep)*100;
txR5 = (sumr(testeR5)/nrep)*100;
txR10 = (sumr(testeR10)/nrep)*100;

// TESTE DA RAZAO DE VEROSSIMILHANÇA - caso disp. variavel
statRv = -2.*(emvV - emvRv);

testeRv1 = (statRv[0][] .> 6.63);
testeRv5 = (statRv[0][] .> 3.84); //quiquadrado c\ 1 grau de liberdade
testeRv10 = (statRv[0][] .> 2.71);

txRv1 = (sumr(testeRv1)/nrep)*100;
txRv5 = (sumr(testeRv5)/nrep)*100;
txRv10 = (sumr(testeRv10)/nrep)*100;

testeWf1 = statWF[0][] .> 6.63;
testeWf5 = statWF[0][] .> 3.84; //quiquadrado c\ 1 grau de liberdade

```

```

testeWf10 = statWF[0][] .> 2.71;

txWf1 = (sumr(testeWf1)/nrep)*100;
txWf5 = (sumr(testeWf5)/nrep)*100;
txWf10 = (sumr(testeWf10)/nrep)*100;

testeWv1 = statWV[0][] .> 6.63;
testeWv5 = statWV[0][] .> 3.84; //quiquadrado c\ 1 grau de liberdade
testeWv10 = statWV[0][] .> 2.71;

txWv1 = (sumr(testeWv1)/nrep)*100;
txWv5 = (sumr(testeWv5)/nrep)*100;
txWv10 = (sumr(testeWv10)/nrep)*100;

testeSf1 = statSf[0][] .> 6.63;
testeSf5 = statSf[0][] .> 3.84; //quiquadrado c\ 1 grau de liberdade
testeSf10 = statSf[0][] .> 2.71;

txSf1 = (sumr(testeSf1)/nrep)*100;
txSf5 = (sumr(testeSf5)/nrep)*100;
txSf10 = (sumr(testeSf10)/nrep)*100;

testeSv1 = statSv[0][] .> 6.63;
testeSv5 = statSv[0][] .> 3.84; //quiquadrado c\ 1 grau de liberdade
testeSv10 = statSv[0][] .> 2.71;

txSv1 = (sumr(testeSv1)/nrep)*100;
txSv5 = (sumr(testeSv5)/nrep)*100;
txSv10 = (sumr(testeSv10)/nrep)*100;

/* Impressao dos resultados na tela */
println("\n PRINCIPAIS RESULTADOS\n");
println("OX PROGRAM:                ", oxfilename(0));
println("OX VERSION:                  ", oxversion());
println("NUM OBSERVACOES:               ", nob);
println("NUM REPLICAS:                  ", nrep);
println("Grau de Heterogeneidade:       ", double(lambda));
println("Lambda 2:                      ", double(lambda_2));
println("Intervalo de mu:               ", "[", double(minc(mur)), ", ",
    " ", double(maxc(mur)), "]" );

println("\n");
println("Numero de falhas de convergencia");
println("Disp. Fixa:                    ", falhaF);

```

```

println("Disp. Var:          ", falhaV);
println("R Disp. Fixa:       ", falhaR);
println("R Disp. Var:        ", falhaRv);

println("\n _____");
println("\n * DISPERSAO FIXA *");
println(" _____");

println("\n");

println("%r", {"beta0", "beta1", "beta2", "phir"},
        "%c", {"Real", "Estimativa", "Vies " , " Vies relativo"},
        theta1~est1~(est1-theta1)~((est1-theta1)./theta1)*100);

println("\n");

println("MOMENTOS",
        "%r", {"beta0", "beta1", "beta2", "phir"},
        "%c", {"Media", "Variancia", "Assimetria", "Curtose"},
        moments(mle1', 4)' [0:3] [1:4]);

println("\n");

println("%r", {"beta0", "beta1", "beta2", "phir"},
        "%c", {"Maximo", "Minimo"},
        maxc(mle1') ' ~minc(mle1') ');

////////////////////////////////////

println("\n _____");
println("\n * DISPERSAO VARIABEL *");
println(" _____");

println("\n");

println("%r", {"beta0", "beta1", "beta2", "gama0", "gama1", "gama2"},
        "%c", {"Real", "Estimativa", "Vies " , " Vies relativo"},
        theta2~est2~(theta2-est2)~((est2-theta2)./theta2)*100);

println("\n");

println("MOMENTOS",
        "%r", {"beta0", "beta1", "beta2", "gama0", "gama1", "gama2"},
        "%c", {"Media", "Variancia", "Assimetria", "Curtose"},
        moments(mle2', 4)' [0:5] [1:4]);

```

```

println("\n");

println("%r", {"beta0","beta1","beta2","gama0","gama1","gama2"},
        "%c", {"Maximo","Minimo"},
maxc(mle2') '~minc(mle2') ');

println("\n");

////////////////////////////////////

println("\n _____");
println("\n * TESTE RESET DISP. FIXA *");
println(" _____");

println("\n");

println("TAXAS DE REJEICAO",
        "%r", {"RV","Wald","Score"},
        "%c", {"alpha = 1%","alpha = 5%","alpha = 10%"},
        txR1~txR5~txR10|txWf1~txWf5~txWf10|txSf1~txSf5~txSf10);

println("\n");

println("MOMENTOS",
        "%r", {"RV","Wald","Escore"},
"%c", {"Media","Variancia","Assimetria","Curtose"},
moments(statR',4)'[0][1:4]|moments(statWF',4)'[0][1:4]|
moments(statSf',4)'[0][1:4]);

println("\n");

println("QUANTIS",
        "%r", {"RV","Wald","Escore"},
        "%c", {"0.99","0.95","0.90"},
        quantiler(statR, <0.99,0.95,0.90>)|quantiler(statWF,
        <0.99,0.95,0.90>)|quantiler(statSf, <0.99,0.95,0.90>));
savemat("quantis_F",quantiler(statR, <0.99,0.95,0.90>)|quantiler
(statWF, <0.99,0.95,0.90>)|quantiler(statSf, <0.99,0.95,0.90>));
println("\n");

println("%r", {"RV","Wald","Escore"},
        "%c", {"Maximo","Minimo"},
maxc(statR') '~minc(statR') '~maxc(statWF') '~
minc(statWF') '~maxc(statSf') '~minc(statSf') ');

```

```

////////////////////////////////////
println("\n _____");
println("\n * TESTE RESET DISP. VARIABEL *");
println(" _____");

println("\n");

println("TAXAS DE REJEICAO",
        "%r", {"RV", "Wald", "Score"},
        "%c", {"alpha = 1%", "alpha = 5%", "alpha = 10%"},
        txRv1~txRv5~txRv10|txWv1~txWv5~txWv10|txSv1~txSv5~txSv10);

println("\n");

println("MOMENTOS",
        "%r", {"RV", "Wald", "Escore"},
        "%c", {"Media", "Variancia", "Assimetria", "Curtose"},
        moments(statRv', 4)' [0] [1:4] | moments(statWV', 4)' [0] [1:4] |
        moments(statSv', 4)' [0] [1:4]);

println("\n");

println("QUANTIS",
        "%r", {"RV", "Wald", "Escore"},
        "%c", {"0.99", "0.95", "0.90"},
        quantiler(statRv, <0.99, 0.95, 0.90>) | quantiler(statWV,
        <0.99, 0.95, 0.90>) | quantiler(statSv, <0.99, 0.95, 0.90>));
savemat("quantis_V", quantiler(statRv, <0.99, 0.95, 0.90>) | quantiler
(statWV, <0.99, 0.95, 0.90>) | quantiler(statSv, <0.99, 0.95, 0.90>));

println("%r", {"RV", "Wald", "Escore"},
        "%c", {"Maximo", "Minimo"},
        maxc(statRv') '~ minc(statRv') '~ | maxc(statWV') '~
        minc(statWV') '~ | maxc(statSv') '~ minc(statSv') '~);

/* Impressao do tempo */
print( "\n DATA: ", date());
print( "\n HORA: ", time());
print( "\n TEMPO TOTAL DE EXECUCAO: ", timespan(dExecTime),
        " segundos");
print( "\n ");

```

}

/*----- Fim do Programa -----*/

APÊNDICE C

Programa - Simulação de Poder

Este apêndice contém um dos programas utilizados nas simulações de poder desta dissertação. Este programa foi utilizado no experimento onde há omissão de variável regressora importante. O programa foi escrito em linguagem Ox (Doornik, 2009).

```
/* _____
```

PROGRAMA: Poder_variavel-importante.ox

USO: Realizar simulação Monte Carlo para avaliar numericamente o desempenho, no que diz respeito a poder, do teste de má especificação proposto por Cribari-Neto & Lima (2007).

AUTOR: José Sérgio Casé de Oliveira

OBS.: Este programa foi desenvolvido a partir dos programas apresentados nas dissertações de mestrado de Lima (2007) e Queiroz (2011).

DATA: 10/12/2012.

```
_____ */
```

```
/* Arquivos de cabeçalho */  
#include<oxstd.h>  
#include<oxprob.h>  
#import<maximize>
```

```
/* Variaveis globais */  
static decl yfuncF;  
static decl yfuncV;  
static decl yfuncR;  
static decl yfuncRv;  
static decl s_mX;  
static decl s_mXr;
```



```

static decl s_mXrv;
static decl s_mZ;

const decl nobs = 20; /* Tamanho Amostral, usar 20,40,60,80,100 */
const decl nrep = 10000; // # de replicas de monte carlo
const decl k = 3;      // # de betas
const decl m = 3;      // # de gamas
const decl r = 4;      // # de betas usando s_mXr

/* Funcao de log-verossimilhanca - disp. fixa */
floglikFixed(const vPF, const adFunc, const avScore, const amHess)
{
decl eta = s_mX*vPF[0:(k-1)];
decl mu = exp(eta) ./ (1.0+exp(eta));
decl phi = vPF[k];
decl ynew = log( yfuncF ./ (1.0-yfuncF) );
decl munew = polygamma(mu*phi, 0) - polygamma((1.0-mu)*phi, 0);
decl T = diag( mu .* (1 - mu) );

adFunc[0] = double( sumc( loggamma(phi) - loggamma(mu.*phi)-
loggamma((1-mu).*phi) +
(mu.*phi-1) .* log(yfuncF) + ((1-mu).*phi-1) .* log(1-yfuncF)));

if (avScore)
{
(avScore[0])[0:(k-1)] = phi*s_mX'*T*(ynew-munew);
(avScore[0])[k] = double(sumc( polygamma(phi, 0) - mu .*
polygamma(mu*phi, 0) -
(1.0-mu) .* polygamma( (1.0-mu)*phi, 0) + mu .* log(yfuncF) +
(1.0-mu) .* log(1.0-yfuncF) ));
}
if( isnan(adFunc[0]) || isdotinf(adFunc[0]) )
return 0;
else
return 1;
}

/* Função de log-verossimilhança - disp. variavel */
floglikVar(const vPV, const adFunc, const avScore, const amHess)
{
decl eta = s_mX*vPV[0:(k-1)];
decl mu = exp(eta)./(1 + exp(eta));
decl delta = s_mZ*vPV[k:(k+m-1)];
decl phi = exp(delta);
decl ystar = log(yfuncV./(1 - yfuncV));

```

```

decl mustar = polygamma(mu.*phi,0)-polygamma((1-mu).*phi,0);
decl ydag = log(1 - yfuncV);
decl mudag = polygamma((1 - mu) .* phi, 0) - polygamma(phi, 0) ;
decl T = diag( mu .* (1 - mu) );
decl F = diag(phi);
decl H = diag(phi);
decl M = diag(mu);

adFunc[0] = double(sumc( loggamma(phi) - loggamma(mu .* phi) -
loggamma((1 - mu) .* phi) +
(mu .* phi - 1) .* log(yfuncV) + ((1 - mu).*phi - 1) .*
log(1 - yfuncV)));

if(avScore)
{
(avScore[0])[0:(k-1)] = s_mX'*T*F*(ystar-mustar);
(avScore[0])[k:(k+m-1)] = s_mZ'*H*(M*(ystar-mustar)+(ydag-mudag));
}
if( isnan(adFunc[0]) || isdotinf(adFunc[0]) )
return 0;
else
return 1;
}

/* Função de log-verossimilhança para teste RESET - disp. fixa */
floglikR(const vP, const adFunc, const avScore, const amHess)
{
decl eta = s_mXr*vP[0:(r-1)];
decl mu = exp(eta)./(1.0+exp(eta));
decl phi = vP[r];
decl ynew = log( yfuncR./(1.0-yfuncR) );
decl munew = polygamma(mu*phi, 0) - polygamma((1.0-mu)*phi, 0);
decl T = diag(exp(eta)./(1.0+exp(eta)).^2);

adFunc[0] = double(sumc(loggamma(phi) - loggamma(mu.*phi)-
loggamma((1-mu).*phi) +
(mu.*phi-1) .* log(yfuncR) + ((1-mu).*phi-1) .* log(1-yfuncR)));

if (avScore)
{
(avScore[0])[0:(r-1)] = phi*s_mXr'*T*(ynew-munew);
(avScore[0])[r] = double(sumc( polygamma(phi, 0) - mu .*
polygamma(mu*phi, 0) -
(1.0-mu) .* polygamma( (1.0-mu)*phi, 0) + mu .* log(yfuncR)
+ (1.0-mu) .* log(1.0-yfuncR) ));
}
}

```

```

if( isnan(adFunc[0]) || isdotinf(adFunc[0]) )
return 0;
else
return 1;
}

/* Função de log-verossimilhança para teste RESET - disp. variavel */
floglikRv(const vtheta, const adFunc, const avScore, const amHess)
{
decl eta = s_mXrv*vtheta[0:(r-1)];
decl mu = exp(eta)./(1 + exp(eta));
decl delta = s_mZ*vtheta[r:(r+m-1)];
decl phi = exp(delta);
decl ystar = log(yfuncRv./(1 - yfuncRv));
decl mustar = polygamma(mu.*phi,0)-polygamma((1-mu).*phi,0);
decl ydag = log(1 - yfuncRv);
decl mudag = polygamma((1 - mu) .* phi, 0) - polygamma(phi, 0) ;
decl T = diag(exp(eta)./(1 + exp(eta)).^2);
decl F = diag(phi);
decl H = diag(phi);
decl M = diag(mu);

adFunc[0] = double(sumc( loggamma(phi) - loggamma(mu .* phi) -
loggamma((1 - mu) .* phi) +
(mu .* phi - 1) .* log(yfuncRv) + ((1 - mu).*phi - 1) .*
log(1 - yfuncRv)));

if(avScore)
{
(avScore[0])[0:(r-1)] = s_mXrv'*T*F*(ystar-mustar);
(avScore[0])[r:(r+m-1)] = s_mZ'*H*(M*(ystar-mustar)+(ydag-mudag));
}
if( isnan(adFunc[0]) || isdotinf(adFunc[0]) )
return 0;
else
return 1;
}

main()
{

decl irl1,irl2,vp1,vp2, dfunc1,dfunc2,dExecTime,falhaF,falhaV,mle1,
mle2,est1,est2,j,beta0,gama0,emvF,emvV,gama1,gama2,beta1,beta2,
phir,theta1,theta2,etar,mur,prF,a,qrF,prV,qrV,ynewF,ynewV,
betaolsF,expfit,betaolsV,deltar,q, phirV,olsfittednew,olsererrorvar,

```

```

olsfitted,thetainiF,mleRv,thetainiV,gamaols,txR1,mleR,falhaR,vpR,
irR,dfuncR,emvR,estR,statR,testeR1,testeR5,testeR10,txR5,txR10,
falhaRv,dfuncRv,vpRv,irRv,emvRv,estRv,statRv,testeRv1,testeRv5,
txRv1,txRv5,txRv10,phi_iniF,phi_iniV,olsfittednewV, olsererrorvarV,
expfitV,testeRv10,olsfittedV,betaolsRF,olsfittednewRf,
olsererrorvarRf,expfitRf,olsfittedRf,betaolsRV,olsfittednewRv,
expfitRv,olsfittedRv,phi_iniRF,phi_iniRV,olsererrorvarRv;

```

```

decl statWF,testeWf1,testeWf5,testeWf10,txWf1,txWf5,txWf10,
thetahatWf,etahat,muhat,phihat,That,psihat,psi2hat,What,Dhat,
vchat,tempinv,ghat,K1hat,K2hat,fisherinv,statWV,statSf,statSv;

```

```

decl thetahatSf,etatil,mutil,phitil,psitil1,psitil2,Ttil,
Wtil,vctil,Dtil,tempinvtil,gtil,K1til,K2til,fisherinvtil,
mustartil,escorettil;

```

```

decl vtheta1,etatilV,mutilV,TtilV,deltatilV,phitilV,FtilV,
HtilV,Mtil,mustartilV,escorebetatilV,VdagtilV,CovtilV,
VstartilV,KbetabetatilV,KbetagamatilV,KgamabetatilV,
KgamagamatilV,fishertilV,fisherinvtilV;

```

```

decl etahatv,muhatv,deltahatv,phihatv,Tv,Hv,Mv,Fv,Vstar,
Vdag,Cov,Kbetabeta,Kbetagama,Kgamagama,Kgamabeta,fisher2,
fisherinv2;

```

```

decl testeWv1,testeWv5,testeWv10,txWv1,txWv5,txWv10,testeSf1,
testeSf5,testeSf10,txSf1,txSf5,txSf10,testeSv1,testeSv5,
testeSv10,txSv1,txSv5,txSv10,ops,ops2,beta3;

```

```

/* Inicio da contagem do tempo */
dExecTime = timer();
ranseed(1750); // semente fixada, gerador default

```

```

/* Valores verdadeiros dos parametros */
beta0 = -1.0;
beta1 = 0.5;
beta2 = 0.3;
beta3 = 3.5;
gama0 = 1.5;
gama1 = 3.0;
gama2 = 2.5;
phir = 120;

```

```

/* Vetor de parametros */
theta1 = beta0|beta1|beta2|phir;

```

```
theta2 = beta0|beta1|beta2|gama0|gama1|gama2;
```

```
/* Algumas inicializacoes */
```

```
mle1 = zeros(rows(theta1),nrep);
```

```
mle2 = zeros(rows(theta2),nrep);
```

```
mleR = zeros(5,nrep);
```

```
mleRv = zeros(7,nrep);
```

```
emvF = zeros(1,nrep);
```

```
emvV = zeros(1,nrep);
```

```
emvR = zeros(1,nrep);
```

```
emvRv = zeros(1,nrep);
```

```
statWF = zeros(1,nrep);
```

```
statWV = zeros(1,nrep);
```

```
statSf = zeros(1,nrep);
```

```
statSv = zeros(1,nrep);
```

```
//MaxControl(-1,1,1);
```

```
a = 1~(ranu(20,k-1));
```

```
if(nobs == 20){s_mX = a;}
  if(nobs == 40){s_mX = a|a;}
if(nobs == 60){s_mX = a|a|a;}
if(nobs == 80){s_mX = a|a|a|a;}
if(nobs == 100){s_mX = a|a|a|a|a;}
```

```
decl a2 = ranu(20,1);
```

```
decl tal;
```

```
if(nobs == 20){tal = a2;}
  if(nobs == 40){tal = a2|a2;}
if(nobs == 60){tal = a2|a2|a2;}
if(nobs == 80){tal = a2|a2|a2|a2;}
if(nobs == 100){tal = a2|a2|a2|a2|a2;}
```

```
etar = (s_mX~tal) * (theta1[0:k-1]|beta3);
```

```
mur = exp(etar)./(1.0 + exp(etar));
```

```
decl etar2 = s_mX * theta1[0:k-1];
```

```
decl mur2 = exp(etar2)./(1.0 + exp(etar2));
```

```
s_mZ = s_mX;
```

```

    deltar = s_mZ * theta2[k:k+m-1];
    phirV = exp(deltar);
    q = columns(s_mZ);

/* Calculando lambda */
    decl th, phi_t, lambda, lambda_2, var_y, y_dataf, y_datav;
    th = gama0|gama1|gama2;
    phi_t = exp(s_mZ*th);
    lambda = maxc(phi_t) / minc(phi_t);

    var_y = (mur .* (1 - mur)) ./ (1 + phirV);
    lambda_2 = maxc(var_y) / minc(var_y);

/* Contadores de falhas */
    falhaF = 0;      // fixo
    falhaV = 0;      // var
    falhaR = 0;      // RESET disp. fixa
    falhaRv = 0;     // RESET disp. variavel

for(j=0; j <nrep; j++)          //Loop de Monte Carlo//
{

    decl nobsmenosk = nobs - k;
    decl nobsmenosr = nobs - r;

    // Grando yfunc para estimacao com disp. fixa
    prF = (mur .* phir);
    qrF = ((1 - mur) .* phir);
    yfuncF = ranbeta(nobs,1,prF,qrF); // linhas, colunas , p e q

    ynewF = log(yfuncF ./ (1.0 - yfuncF));

/* Calculo do vetor inicial para Maximizacao da
log-verossimilhanca - disp. fixa*/
    if(k > 1)
    {
        ols2c(ynewF, s_mX, &betaolsF);
    }
    else if(k == 1)
    {
        betaolsF = meanc(ynewF);
    }

    olsfittednew = s_mX*betaolsF;
    olserrorvar = sumsqrc(ynewF - olsfittednew)./(nobsmenosk);

```

```

expfit = exp(olsfittednew);
    olsfitted = expfit ./ (1.0 + expfit);

phi_iniF = (meanc(1 ./ (olsererrorvar'*(olsfitted .*
(1.0 - olsfitted)))) - 1.0);

vp1 = betaolsF|phi_iniF;

/* Maximizacao da log-verossimilhanca - disp. fixa */
irl = MaxBFGS(floglikFixed, &vp1, &dfunc1, 0, 0);

if( irl != MAX_CONV && irl != MAX_WEAK_CONV )
{
    falhaF++;
    j--;
    continue;
}
else
{
    mle1[][j] = vp1;
    emvF[][j] = dfunc1;
}

ops = s_mX*vp1[0:2][];
s_mXr = s_mX~((ops).^2);

// Grando yfunc para estimacao com disp. variável
prV = (mur .* phirV);
qrV = ((1 - mur) .* phirV);
yfuncV = ranbeta(nobs,1,prV,qrV); //linhas, colunas , p e q

ynewV = log(yfuncV ./ (1.0 - yfuncV));

/* Calculo do vetor inicial para Maximizacao da
log-verossimilhanca - disp. variavel*/
if(k > 1)
{
    ols2c(ynewV, s_mX, &betaolsV);
}
else if(k == 1)
{
    betaolsV = meanc(ynewV);
}

```

```

olsfittednewV = s_mX*betaolsV;
    olsererrorvarV = sumsqrc(ynewV - olsfittednewV)./(nobsmenosk);
expfitV = exp(olsfittednewV);
    olsfittedV = expfitV ./ (1.0 + expfitV);

phi_iniV = log((meanc(1 ./ (olsererrorvarV'*(olsfittedV .*
(1.0 - olsfittedV)))) - 1.0))|0|0;

vp2 = betaolsV|phi_iniV;

/* Maximizacao da log-verossimilhanca - disp. var */
ir2 = MaxBFGS(floglikVar, &vp2, &dfunc2, 0, 0);

if( ir2 != MAX_CONV && ir2 != MAX_WEAK_CONV )
{
    falhaV++;
    j--;
    continue;
}
else
{
    mle2[][j] = vp2;
    emvV[][j] = dfunc2;
}

ops2 = s_mX*vp2[0:2][];
s_mXrv = s_mX~((ops2).^2);

/* Calculo do vetor inicial para Maximizacao da log-verossimilhanca
    teste RESET - disp. fixa*/
if(r > 1)
{
    ols2c(ynewF, s_mXr, &betaolsRF);
}
else if(r == 1)
{
    betaolsRF = meanc(ynewF);
}

olsfittednewRf = (s_mXr)*betaolsRF;
    olsererrorvarRf = sumsqrc(ynewF - olsfittednewRf)./(nobsmenosr);
expfitRf = exp(olsfittednewRf);
    olsfittedRf = expfitRf ./ (1.0 + expfitRf);

```



```

phi_iniRF = (meanc(1 ./ (olsererrorvarRf'*(olsfittedRf .*
(1.0 - olsfittedRf)))) - 1.0);

/* Calculo do vetor inicial para Maximizacao da log-verossimilhanca
    teste RESET - disp. variavel*/
if(r > 1)
{
    ols2c(ynewV, s_mXrv, &betaolsRV);
}
else if(r == 1)
{
    betaolsRV = meanc(ynewV);
}

olsfittednewRv = (s_mXrv)*betaolsRV;
olsererrorvarRv = sumsqr(ynewV - olsfittednewRv)./(nobsmenosr);
expfitRv = exp(olsfittednewRv);
olsfittedRv = expfitRv ./ (1.0 + expfitRv);

phi_iniRV = log((meanc(1 ./ (olsererrorvarRv'*(olsfittedRv .*
(1.0 - olsfittedRv)))) - 1.0))|0|0;

yfuncR = yfuncF;

yfuncRv = yfuncV;

/* Chute inicial dos parametros */

vpR = betaolsRF|phi_iniRF;
vpRv = betaolsRV|phi_iniRV;

/* Armazenando o chute inicial */
thetainiF = vp1;
thetainiV = vp2;

```

```

/***** TESTE ESCORE DISP. FIXA *****/

thetahatSf = vp1[0:k-1]|0|vp1[k];

etatil = s_mXr * (thetahatSf[0:r-1]);
mutil = exp(etatil) ./ (1+exp(etatil));
phitil = thetahatSf[k+1];
psitil1 = polygamma(mutil*phitil, 1);
psitil2 = polygamma((1.0-mutil)*phitil, 1);
Ttil = diag( exp(etatil) ./ (1.0+exp(etatil)) .^2 );
Wtil = diag(phitil*(psitil1+psitil2)) * Ttil .^2;
vctil = phitil*(psitil1.*mutil-psitil2.*(1-mutil));
Dtil = diag(psitil1.*(mutil.^2)+psitil2.*(1.0-mutil).^2-
polygamma(phitil,1));
tempinvtil = invtsym(s_mXr'*Wtil*s_mXr);
gtil = trace(Dtil)-(1.0/phitil)*vctil'*Ttil'*s_mXr*tempinvtil*
s_mXr'*Ttil*vctil;
K1til = tempinvtil*(gtil*unit(r)+(1/phitil)*s_mXr'*Ttil*vctil*
vctil'*Ttil'*s_mXr*tempinvtil);
K2til = -tempinvtil*s_mXr'*Ttil*vctil;
fisherinvtil = (1/(phitil*gtil))*((K1til~K2til)|(-vctil'*Ttil'*
s_mXr*tempinvtil~phitil));

mustartil = polygamma(mutil*phitil, 0) - polygamma((1.0-mutil)*
phitil, 0);
escoretil = phitil*(s_mXr[][r-1:]')*Ttil*(ynewF - mustartil);

statSf[][j] = (escoretil)'*(fisherinvtil[r-1][r-1])*escoretil;

/*****

/***** TESTE ESCORE DISP. VAR *****/

vthetal = vp2[0:r-2]|0|vp2[r-1:r+1];

etatilV = s_mXrv*(vthetal[0:r-1-1]|zeros(1,1));
mutilV = exp(etatilV) ./ (1.0+exp(etatilV));
TtilV = diag( mutilV .* (1.0-mutilV) );
deltatilV = s_mZ*(vthetal[r:r+m-1]);
phitilV = exp(deltatilV);
FtilV = diag(phitilV);
HtilV = diag(phitilV);
Mtil = diag(mutilV);
mustartilV = polygamma(mutilV .* phitilV, 0) -

```

```

polygamma((1.0 - mutilV) .* phitilV, 0);
escorebetatilV = (s_mXrv'*FtilV*TtilV)*(ynewV-mustartilV);
VdagtilV = diag(polygamma((1.0 - mutilV) .* phitilV, 1)-
polygamma(phitilV, 1));
CovtilV = diag(-polygamma((1.0 - mutilV) .* phitilV, 1));
VstartilV = diag(polygamma(mutilV .* phitilV, 1) +
polygamma((1.0 - mutilV) .* phitilV, 1));
KbetabetatilV = s_mXrv'*FtilV*TtilV*VstartilV*TtilV*
FtilV*s_mXrv;
KbetagamatilV = s_mXrv'*FtilV*(Mtil*VstartilV+CovtilV)*
TtilV*HtilV*s_mZ;
KgamabetatilV = KbetagamatilV';
KgamagamatilV = s_mZ'*HtilV*(Mtil*VstartilV*Mtil+(Mtil+Mtil)*
CovtilV+VdagtilV)*HtilV*s_mZ;
fishertilV = (KbetabetatilV~KbetagamatilV)|
(KgamabetatilV~KgamagamatilV);
fisherinvtilV = invert(fishertilV);

statSv[][j] = (escorebetatilV[r-1:][])'*
(fisherinvtilV[r-1:r-1][r-1:r-1])*(escorebetatilV[r-1:][]);

/*****

/* Maximizacao da log-verossimilhanca teste
      RESET - disp. fixa */
irR = MaxBFGS(floglikR, &vpR, &dfuncR, 0, 0);

if( irR != MAX_CONV && irR != MAX_WEAK_CONV )
{
falhaR++;
j--;
continue;
}
else
{
mleR[][j] = vpR;
emvR[][j] = dfuncR;
}

/***** TESTE WALD DISP. FIXA *****/

etahat = s_mXr * vpR[0:r-1];
muhat = exp(etahat) ./ (1 + exp(etahat));

```

```

phi_hat = vpR[r];
That = diag( exp(etahat) ./ (1.0+exp(etahat)) .^2 );
psi_hat = polygamma(mu_hat*phi_hat, 1);
psi2_hat = polygamma((1.0-mu_hat)*phi_hat, 1);
What = diag(phi_hat*(psi_hat+psi2_hat)) * That .^2;
Dhat = diag(psi_hat.*(mu_hat.^2)+psi2_hat.*(1.0-mu_hat).^2-
polygamma(phi_hat,1));
vchat = phi_hat*(psi_hat.*mu_hat-psi2_hat.*(1-mu_hat));
tempinv = inv(s_mXr'*What*s_mXr);
ghat = trace(Dhat)-(1.0/phi_hat)*vchat'*That'*s_mXr*tempinv*
s_mXr'*That*vchat;

K1_hat = tempinv*(ghat*unit(r)+(1/phi_hat)*s_mXr'*That*vchat*
vchat'*That'*s_mXr*tempinv);
K2_hat = -tempinv*s_mXr'*That*vchat;
fisherinv = (1/(phi_hat*ghat))*((K1_hat~K2_hat)|(-vchat'*That'*
s_mXr*tempinv~phi_hat));

statWF[][j] = vpR[r-1]'.*(fisherinv[r-1][r-1])^(-1)*vpR[r-1];

/*****

/* Maximizacao da log-verossimilhanca teste
      RESET - disp. var */
irRv = MaxBFGS(floglikRv, &vpRv, &dfuncRv, 0, 0);

    if( irRv != MAX_CONV && irRv != MAX_WEAK_CONV )
    {
        falhaRv++;
        j--;
        continue;
    }
    else
    {
        mleRv[][j] = vpRv;
        emvRv[][j] = dfuncRv;
    }

/***** TESTE WALD DISP. VAR *****/

etahatv = s_mXrv * vpRv[0:r-1];
muhatv = exp(etahatv) ./ (1.0 + exp(etahatv));
deltahatv = s_mZ * vpRv[r:r+m-1];
phi_hatv = exp(deltahatv);

```

```

Tv = diag(muhatv .* (1.0 - muhatv));
Hv = diag(phiahv);
Mv = diag(muhatv);
Fv = diag(phiahv);
Vstar = diag(polygamma(muhatv .* phiahv ,1) +
polygamma((1.0 - muhatv) .* phiahv,1));
Vdag = diag(polygamma((1.0 - muhatv) .* phiahv,1) -
polygamma(phiahv,1));
Cov = diag(-polygamma((1.0 - muhatv) .* phiahv,1));
Kbetabeta = s_mXrv'*Fv*Tv*Vstar*Tv*Fv*s_mXrv;
Kbetagama = s_mXrv'*Fv*(Mv*Vstar+Cov)*Tv*Hv*s_mZ;
Kgamagama = s_mZ'*Hv*(Mv*Vstar*Mv+(Mv+Mv)*Cov+Vdag)*Hv*s_mZ;
Kgamabeta = Kbetagama';
fisher2 = (Kbetabeta~Kbetagama)|(Kgamabeta~Kgamagama);
fisherinv2 = invert(fisher2);

statWV[][j] = vpRv[r-1:r-1]'*(fisherinv2[r-1:r-1]
[r-1:r-1])^(-1)*vpRv[r-1];

/*****/

} //fim do MC

est1 = meanr(mle1);
est2 = meanr(mle2);
estR = meanr(mleR);
estRv = meanr(mleRv);

//TESTE DA RAZAO DE VEROSSIMILHANÇA - caso disp. fixa
statR = -2.*(emvF - emvR);

if(nobs == 20){testeR1 = (statR[0][] .>
loadmat("quantis_F20")[0][0]);}
if(nobs == 40){testeR1 = (statR[0][] .>
loadmat("quantis_F40")[0][0]);}
if(nobs == 60){testeR1 = (statR[0][] .>
loadmat("quantis_F60")[0][0]);}
if(nobs == 80){testeR1 = (statR[0][] .>
loadmat("quantis_F80")[0][0]);}
if(nobs == 100){testeR1 = (statR[0][] .>
loadmat("quantis_F100")[0][0]);}

if(nobs == 20){testeR5 = (statR[0][] .>

```

```

loadmat("quantis_F20")[0][1]);}
if(nobs == 40){testeR5 = (statR[0][] .>
loadmat("quantis_F40")[0][1]);}
if(nobs == 60){testeR5 = (statR[0][] .>
loadmat("quantis_F60")[0][1]);}
if(nobs == 80){testeR5 = (statR[0][] .>
loadmat("quantis_F80")[0][1]);}
if(nobs == 100){testeR5 = (statR[0][] .>
loadmat("quantis_F100")[0][1]);}

if(nobs == 20){testeR10 = (statR[0][] .>
loadmat("quantis_F20")[0][2]);}
if(nobs == 40){testeR10 = (statR[0][] .>
loadmat("quantis_F40")[0][2]);}
if(nobs == 60){testeR10 = (statR[0][] .>
loadmat("quantis_F60")[0][2]);}
if(nobs == 80){testeR10 = (statR[0][] .>
loadmat("quantis_F80")[0][2]);}
if(nobs == 100){testeR10 = (statR[0][] .>
loadmat("quantis_F100")[0][2]);}

txR1 = (sumr(testeR1)/nrep)*100;
txR5 = (sumr(testeR5)/nrep)*100;
txR10 = (sumr(testeR10)/nrep)*100;

//TESTE DA RAZAO DE VEROSSIMILHANÇA - caso disp. variavel
statRv = -2.*(emvV - emvRv);

if(nobs == 20){testeRv1 = (statRv[0][] .>
loadmat("quantis_V20")[0][0]);}
if(nobs == 40){testeRv1 = (statRv[0][] .>
loadmat("quantis_V40")[0][0]);}
if(nobs == 60){testeRv1 = (statRv[0][] .>
loadmat("quantis_V60")[0][0]);}
if(nobs == 80){testeRv1 = (statRv[0][] .>
loadmat("quantis_V80")[0][0]);}
if(nobs == 100){testeRv1 = (statRv[0][] .>
loadmat("quantis_V100")[0][0]);}

if(nobs == 20){testeRv5 = (statRv[0][] .>
loadmat("quantis_V20")[0][1]);}
if(nobs == 40){testeRv5 = (statRv[0][] .>
loadmat("quantis_V40")[0][1]);}
if(nobs == 60){testeRv5 = (statRv[0][] .>
loadmat("quantis_V60")[0][1]);}

```

```

if(nobs == 80){testeRv5 = (statRv[0][] .>
loadmat("quantis_V80")[0][1]);}
if(nobs == 100){testeRv5 = (statRv[0][] .>
loadmat("quantis_V100")[0][1]);}

if(nobs == 20){testeRv10 = (statRv[0][] .>
loadmat("quantis_V20")[0][2]);}
if(nobs == 40){testeRv10 = (statRv[0][] .>
loadmat("quantis_V40")[0][2]);}
if(nobs == 60){testeRv10 = (statRv[0][] .>
loadmat("quantis_V60")[0][2]);}
if(nobs == 80){testeRv10 = (statRv[0][] .>
loadmat("quantis_V80")[0][2]);}
if(nobs == 100){testeRv10 = (statRv[0][] .>
loadmat("quantis_V100")[0][2]);}

txRv1 = (sumr(testeRv1)/nrep)*100;
txRv5 = (sumr(testeRv5)/nrep)*100;
txRv10 = (sumr(testeRv10)/nrep)*100;

if(nobs == 20){testeWf1 = statWF[0][] .>
loadmat("quantis_F20")[1][0];}
if(nobs == 40){testeWf1 = statWF[0][] .>
loadmat("quantis_F40")[1][0];}
if(nobs == 60){testeWf1 = statWF[0][] .>
loadmat("quantis_F60")[1][0];}
if(nobs == 80){testeWf1 = statWF[0][] .>
loadmat("quantis_F80")[1][0];}
if(nobs == 100){testeWf1 = statWF[0][] .>
loadmat("quantis_F100")[1][0];}

if(nobs == 20){testeWf5 = statWF[0][] .>
loadmat("quantis_F20")[1][1];}
if(nobs == 40){testeWf5 = statWF[0][] .>
loadmat("quantis_F40")[1][1];}
if(nobs == 60){testeWf5 = statWF[0][] .>
loadmat("quantis_F60")[1][1];}
if(nobs == 80){testeWf5 = statWF[0][] .>
loadmat("quantis_F80")[1][1];}
if(nobs == 100){testeWf5 = statWF[0][] .>
loadmat("quantis_F100")[1][1];}

if(nobs == 20){testeWf10 = statWF[0][] .>
loadmat("quantis_F20")[1][2];}
if(nobs == 40){testeWf10 = statWF[0][] .>
loadmat("quantis_F40")[1][2];}

```

```

if(nobs == 60){testeWf10 = statWF[0][] .>
loadmat("quantis_F60")[1][2];}
if(nobs == 80){testeWf10 = statWF[0][] .>
loadmat("quantis_F80")[1][2];}
if(nobs == 100){testeWf10 = statWF[0][] .>
loadmat("quantis_F100")[1][2];}

```

```

txWf1 = (sumr(testeWf1)/nrep)*100;
txWf5 = (sumr(testeWf5)/nrep)*100;
txWf10 = (sumr(testeWf10)/nrep)*100;

```

```

if(nobs == 20){testeWv1 = statWV[0][] .>
loadmat("quantis_V20")[1][0];}
if(nobs == 40){testeWv1 = statWV[0][] .>
loadmat("quantis_V40")[1][0];}
if(nobs == 60){testeWv1 = statWV[0][] .>
loadmat("quantis_V60")[1][0];}
if(nobs == 80){testeWv1 = statWV[0][] .>
loadmat("quantis_V80")[1][0];}
if(nobs == 100){testeWv1 = statWV[0][] .>
loadmat("quantis_V100")[1][0];}

```

```

if(nobs == 20){testeWv5 = statWV[0][] .>
loadmat("quantis_V20")[1][1];}
if(nobs == 40){testeWv5 = statWV[0][] .>
loadmat("quantis_V40")[1][1];}
if(nobs == 60){testeWv5 = statWV[0][] .>
loadmat("quantis_V60")[1][1];}
if(nobs == 80){testeWv5 = statWV[0][] .>
loadmat("quantis_V80")[1][1];}
if(nobs == 100){testeWv5 = statWV[0][] .>
loadmat("quantis_V100")[1][1];}

```

```

if(nobs == 20){testeWv10 = statWV[0][] .>
loadmat("quantis_V20")[1][2];}
if(nobs == 40){testeWv10 = statWV[0][] .>
loadmat("quantis_V40")[1][2];}
if(nobs == 60){testeWv10 = statWV[0][] .>
loadmat("quantis_V60")[1][2];}
if(nobs == 80){testeWv10 = statWV[0][] .>
loadmat("quantis_V80")[1][2];}
if(nobs == 100){testeWv10 = statWV[0][] .>
loadmat("quantis_V100")[1][2];}

```

```

txWv1 = (sumr(testeWv1)/nrep)*100;

```



```

txWv5 = (sumr(testeWv5)/nrep)*100;
txWv10 = (sumr(testeWv10)/nrep)*100;

if(nobs == 20){testeSf1 = statSf[0][] .>
loadmat("quantis_F20")[2][0];}
if(nobs == 40){testeSf1 = statSf[0][] .>
loadmat("quantis_F40")[2][0];}
if(nobs == 60){testeSf1 = statSf[0][] .>
loadmat("quantis_F60")[2][0];}
if(nobs == 80){testeSf1 = statSf[0][] .>
loadmat("quantis_F80")[2][0];}
if(nobs == 100){testeSf1 = statSf[0][] .>
loadmat("quantis_F100")[2][0];}

if(nobs == 20){testeSf5 = statSf[0][] .>
loadmat("quantis_F20")[2][1];}
if(nobs == 40){testeSf5 = statSf[0][] .>
loadmat("quantis_F40")[2][1];}
if(nobs == 60){testeSf5 = statSf[0][] .>
loadmat("quantis_F60")[2][1];}
if(nobs == 80){testeSf5 = statSf[0][] .>
loadmat("quantis_F80")[2][1];}
if(nobs == 100){testeSf5 = statSf[0][] .>
loadmat("quantis_F100")[2][1];}

if(nobs == 20){testeSf10 = statSf[0][] .>
loadmat("quantis_F20")[2][2];}
if(nobs == 40){testeSf10 = statSf[0][] .>
loadmat("quantis_F40")[2][2];}
if(nobs == 60){testeSf10 = statSf[0][] .>
loadmat("quantis_F60")[2][2];}
if(nobs == 80){testeSf10 = statSf[0][] .>
loadmat("quantis_F80")[2][2];}
if(nobs == 100){testeSf10 = statSf[0][] .>
loadmat("quantis_F100")[2][2];}

txSf1 = (sumr(testeSf1)/nrep)*100;
txSf5 = (sumr(testeSf5)/nrep)*100;
txSf10 = (sumr(testeSf10)/nrep)*100;

if(nobs == 20){testeSv1 = statSv[0][] .>
loadmat("quantis_V20")[2][0];}
if(nobs == 40){testeSv1 = statSv[0][] .>
loadmat("quantis_V40")[2][0];}
if(nobs == 60){testeSv1 = statSv[0][] .>
loadmat("quantis_V60")[2][0];}

```

```

if(nobs == 80){testeSv1 = statSv[0][] .>
loadmat("quantis_V80")[2][0];}
if(nobs == 100){testeSv1 = statSv[0][] .>
loadmat("quantis_V100")[2][0];}

if(nobs == 20){testeSv5 = statSv[0][] .>
loadmat("quantis_V20")[2][1];}
if(nobs == 40){testeSv5 = statSv[0][] .>
loadmat("quantis_V40")[2][1];}
if(nobs == 60){testeSv5 = statSv[0][] .>
loadmat("quantis_V60")[2][1];}
if(nobs == 80){testeSv5 = statSv[0][] .>
loadmat("quantis_V80")[2][1];}
if(nobs == 100){testeSv5 = statSv[0][] .>
loadmat("quantis_V100")[2][1];}

if(nobs == 20){testeSv10 = statSv[0][] .>
loadmat("quantis_V20")[2][2];}
if(nobs == 40){testeSv10 = statSv[0][] .>
loadmat("quantis_V40")[2][2];}
if(nobs == 60){testeSv10 = statSv[0][] .>
loadmat("quantis_V60")[2][2];}
if(nobs == 80){testeSv10 = statSv[0][] .>
loadmat("quantis_V80")[2][2];}
if(nobs == 100){testeSv10 = statSv[0][] .>
loadmat("quantis_V100")[2][2];}

txSv1 = (sumr(testeSv1)/nrep)*100;
txSv5 = (sumr(testeSv5)/nrep)*100;
txSv10 = (sumr(testeSv10)/nrep)*100;

/* Impressao dos resultados na tela */
println("\n PRINCIPAIS RESULTADOS\n");
println("OX PROGRAM:                ", oxfilename());
println("OX VERSION:                  ", oxversion());
println("NUM OBSERVACOES:                ", nobs);
println("NUM REPLICAS:                   ", nrep);
println("Grau de Heterogeneidade:        ", double(lambda));
println("Lambda 2:                       ", double(lambda_2));
println("Intervalo de mu:                 "\, "[", double(minc(mur))
, " , ", double(maxc(mur)), "]" );
println("Maximo e minimo da razao de mu's ", println("\n",
maxc(mur ./ mur2)~minc(mur ./ mur2)));

println("\n");

```

```

println("Numero de falhas de convergencia");
println("Disp. Fixa:      ", falhaF);
println("Disp. Var:      ", falhaV);
println("R Disp. Fixa:    ", falhaR);
println("R Disp. Var:     ", falhaRv);

println("\n _____");
println("\n * DISPERSAO FIXA *");
println(" _____");

println("\n");

println("%r", {"beta0","beta1","beta2","phir"},
        "%c", {"Real","Estimativa","Vies " , " Vies relativo"},
        theta1~est1~(est1-theta1)~((est1-theta1)./theta1)*100);

println("\n");

println("MOMENTOS",
        "%r", {"beta0","beta1","beta2","phir"},
        "%c", {"Media","Variancia","Assimetria","Curtose"},
        moments(mle1',4)'[0:3][1:4]);

println("\n");

println("%r", {"beta0","beta1","beta2","phir"},
        "%c", {"Maximo","Minimo"},
        maxc(mle1')'~minc(mle1')');

////////////////////////////////////

println("\n _____");
println("\n * DISPERSAO VARIABEL *");
println(" _____");

println("\n");

println("%r", {"beta0","beta1","beta2","gama0","gama1","gama2"},
        "%c", {"Real","Estimativa","Vies " , " Vies relativo"},
        theta2~est2~(theta2-est2)~((est2-theta2)./theta2)*100);

println("\n");

println("MOMENTOS",

```

```

"%r", {"beta0", "beta1", "beta2", "gama0", "gama1", "gama2"},
"%c", {"Media", "Variancia", "Assimetria", "Curtose"},
moments(mle2', 4)' [0:5] [1:4]);

println("\n");

println("%r", {"beta0", "beta1", "beta2", "gama0", "gama1", "gama2"},
"%c", {"Maximo", "Minimo"},
maxc(mle2') '~minc(mle2') ');

println("\n");

////////////////////////////////////

println("\n _____");
println("\n * TESTE RESET DISP. FIXA *");
println(" _____");

println("\n");

println("TAXAS DE REJEICAO",
"%r", {"RV", "Wald", "Score"},
"%c", {"alpha = 1%", "alpha = 5%", "alpha = 10%"},
txR1~txR5~txR10|txWf1~txWf5~txWf10|txSf1~txSf5~txSf10);

println("\n");

println("MOMENTOS",
"%r", {"RV", "Wald", "Escore"},
"%c", {"Media", "Variancia", "Assimetria", "Curtose"},
moments(statR', 4)' [0] [1:4] | moments(statWF', 4)' [0] [1:4] |
moments(statSf', 4)' [0] [1:4]);

println("\n");

println("QUANTIS",
"%r", {"RV", "Wald", "Escore"},
"%c", {"0.99", "0.95", "0.90"},
quantiler(statR, <0.99, 0.95, 0.90>) | quantiler(statWF,
<0.99, 0.95, 0.90>) | quantiler(statSf, <0.99, 0.95, 0.90>));

println("\n");

println("%r", {"RV", "Wald", "Escore"},
"%c", {"Maximo", "Minimo"},

```

```

maxc(statR') '~minc(statR') '|maxc(statWF') '~minc(statWF')
'|maxc(statSf') '~minc(statSf') ');

////////////////////////////////////

println("\n _____");
println("\n * TESTE RESET DISP. VARIABEL *");
println(" _____");

println("\n");

println("TAXAS DE REJEICAO",
        "%r", {"RV", "Wald", "Score"},
        "%c", {"alpha = 1%", "alpha = 5%", "alpha = 10%"},
        txRv1~txRv5~txRv10|txWv1~txWv5~txWv10|txSv1~txSv5~txSv10);

println("\n");

println("MOMENTOS",
        "%r", {"RV", "Wald", "Escore"},
        "%c", {"Media", "Variancia", "Assimetria", "Curtose"},
        moments(statRv', 4)' [0] [1:4] | moments(statWV', 4)' [0] [1:4] |
        moments(statSv', 4)' [0] [1:4]);

println("\n");

println("QUANTIS",
        "%r", {"RV", "Wald", "Escore"},
        "%c", {"0.99", "0.95", "0.90"},
        quantiler(statRv, <0.99, 0.95, 0.90>) | quantiler(statWV,
        <0.99, 0.95, 0.90>) | quantiler(statSv, <0.99, 0.95, 0.90>));

println("%r", {"RV", "Wald", "Escore"},
        "%c", {"Maximo", "Minimo"},
        maxc(statRv') '~minc(statRv') '|maxc(statWV') '~minc(statWV')
        '|maxc(statSv') '~minc(statSv') ');

/* Impressao do tempo */
print( "\n DATA: ", date());
print( "\n HORA: ", time());
print( "\n TEMPO TOTAL DE EXECUCAO: ", timespan(dExecTime),
        " segundos");
print( "\n" );

```

```
}
```

```
/*----- Fim do Programa -----*/
```