UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE CIÊNCIAS EXATAS E DA NATUREZA
DEPARTAMENTO DE ESTATÍSTICA
CURSO DE BACHARELADO EM ESTATÍSTICA
TRABALHO DE CONCLUSÃO DE CURSO II

# REGRESSÃO WAVELET APLICADA AO MODELO DE REGRESSÃO NORMAL NÃO LINEAR

Adenice Gomes de Oliveira Ferreira

João Pessoa, Junho de 2017

UNIVERSIDADE FEDERAL DA PARAÍBA

ADENICE GOMES DE OLIVEIRA FERREIRA

# REGRESSÃO WAVELET APLICADA AO MODELO DE REGRESSÃO NORMAL NÃO LINEAR

Orientador: Profº Dr. Eufrásio de Andrade Lima Neto.

Trabalho de Conclusão de Curso apresentado à banca examinadora, para avaliação na disciplina TCC II, do curso de Bacharelado em Estatística da Universidade Federal da Paraíba, como requisito parcial para obtenção do Grau de Bacharel.

João Pessoa
Junho de 2017

Catalogação na publicação
Biblioteca Setorial do CCEN/UFPB
Josélia M.O. Silva – CRB-15/113

F383r Ferreira, Adenice Gomes de Oliveira.
Regressão Wavelet aplicada ao modelo de regressão normal não linear / Adenice Gomes de Oliveira Ferreira. – João Pessoa, 2017.
54 p. : il. color.

Monografia (Bacharelado em Estatística) – Universidade Federal da Paraíba.
Orientador(a): Profº. Drº. Eufrásio de Andrade Lima Neto.

1. Análise de regressão. 2. Regressão Wavelet. 3. Análise de correlação. 4. Função não linear. I. Título.

UFPB/BS-CCEN CDU 519.233.5(043.2)

Ata da Sessão Pública de Defesa de Monografia de **Adenice Gomes de Oliveira Ferreira**, realizada em 05 de junho de 2017 no Departamento de Estatística da Universidade Federal da Paraíba - UFPB.

---

Aos cinco dias do mês de junho de dois mil e dezessete, às dez horas, na Sala 20 do Centro de Ciências Exatas e da Natureza da Universidade Federal da Paraíba, reuniram-se os membros da Banca Examinadora constituída para avaliar o Trabalho de Conclusão Curso intitulado "REGRESSÃO WAVELET APLICADA AO MODELO DE REGRESSÃO NÃO LINEAR" de autoria de ADENICE GOMES DE OLIVEIRA FERREIRA. A Banca Examinadora foi composta pelos professores: Prof. Dr. Eufrásio de Andrade Lima Neto (DE-UFPB, orientador), Prof. Dr. Marcelo Rodrigo Portela Ferreira (DE-UFPB, examinador) e Profa. Dra. Maria Lídia Coco Terra (DE-UFPB, examinadora). Dando início aos trabalhos, o presidente da banca cumprimentou os presentes, comunicou aos mesmos a finalidade da reunião e passou à palavra à discente para que se fizesse, oralmente, a exposição de seu trabalho de monografia. Concluída a apresentação, a discente foi arguido pela Banca Examinadora que sugeriu algumas alterações até o dia 12 de junho de 2017, de acordo com a Resolução No. 02/2014 do Colegiado do Curso de Bacharelado em Estatística da UFPB. Uma vez entregue a versão final do Trabalho de Conclusão de Curso à Coordenação do Bacharelado em Estatística, com as alterações solicitadas pela Banca Examinadora dentro do prazo estabelecido, a discente será aprovado com nota (9,5) NOVE e MEIO, que é a média aritmética das notas atribuídas pelos membros da Banca Examinadora.

João Pessoa, 05 de junho de 2017.

Prof. Eufrásio de Andrade Lima Neto (Orientador)

Prof. Marcelo Rodrigo Portela Ferreira (Examinador)

Profa. Maria Lídia Coco Terra (Examinadora)

Adenice Gomes de Oliveira Ferreira (Discente)

*À família que tanto amo e aos amigos que tanto prezo.*

# Agradecimentos

Agradeço primeiramente a Deus, pelo seu imenso amor e misericórdia, pelo seu carinho e cuidado comigo e minha família, pela força que nos tem dado para transpor os obstáculos e continuar crescendo em graça e sabedoria.

Agradeço aos meus pais por sempre me apoiarem e orientarem em todas as decisões importantes que tomei. Vocês são meu tesouro e exemplos vivos de amor, união e, sobretudo, ética. Tenho orgulho de tê-los como meus principais professores, pois "amar a Deus sobre todas as coisas e ao próximo como a si mesmo" foi uma lição que aprendi primeiro com vocês.

Agradeço ao meu irmão, João Marcos, pelas conversas e desabafos, pela cumplicidade. Te admiro muito pela sua dedicação, disciplina, perseverança, e pelo seu coração enorme que acolhe a todos, sem distinção. Você é um exemplo a ser seguido.

Agradeço ao meu doce Diogo, que me acompanha desde antes do início dessa caminhada. Na universidade, estivemos juntos em trabalhos, nas tensões pré prova e seminários, nas pesquisas e consultorias. Na vida, estivemos juntos em várias situações, das tristes até as mais felizes. Você é o presente que Deus me deu, e poder crescer contigo é um privilegio para mim.

Agradeço aos amados amigos e familiares que de forma direta ou indireta contribuíram com o aprendizado nesta etapa da minha vida. Aos meus amigos de turma - Anny, André, Zé e Lukas - desejo que nossa amizade siga para toda vida e que possamos continuar nos encontrando, colocando a conversa em dia e rindo dos momentos que vivemos. À Clarissa e Danilo: os admiro muito por sua evidente determinação. Ela os têm colocado no caminho do sucesso e a Empresa Junior foi só o primeiro passo.

Agradeço aos professores do Departamento de Estatística da UFPB que desde o início nos dão suporte em tudo. Queria deixar meus agradecimentos especiais à profª Ana Flávia (a senhora transborda amor e alegria, sua história de vida é exemplo de coragem e fé), profª Tatiene, profª Tarciana, profº Luiz, profº João Agnaldo e profº Hemílio, que acompanharam de perto nossa trajetória no curso.

Agradeço de coração ao profº Eufrasio, que talvez nem imagine o quanto me ajudou, principalmente neste percurso final. Muito obrigada por me orientar, por me corrigir quando necessário, por me incentivar e dar forças para continuar. Estou sempre aprendendo algo com o senhor. Sobretudo, muito obrigada pela humanidade com a qual tem me tratado. Passamos por muita coisa juntos desde que me aceitou como orientanda no PIBIC, e o senhor sempre demonstrou respeito e apoio nos momentos difíceis que enfrentei juntamente com minha família. Somos muito gratos por tudo.

Agradeço ao profº Aluísio Pinheiro, pela sua colaboração intelectual ao trabalho.

Por fim, meus sinceros agradecimentos ao profº Marcelo, profª Maria Lídia e profª Izabel pela solicitude em participar da banca que contribuiu no enriquecimento deste trabalho.

*"Mas o fruto do Espírito é amor, alegria, paz, paciência, amabilidade, bondade, fidelidade, mansidão e domínio próprio. Contra essas coisas não há lei".*
*(Bíblia Sagrada, Gálatas 5:22,23)*

# Resumo

Os métodos de regressão não linear representam uma relevante área de pesquisa devido a sua aplicabilidade em diversas áreas de conhecimento. Contudo, a escolha de uma função não linear, que melhor defina os dados não é uma tarefa fácil, quando a relação matemática entre a variável resposta e as independentes é desconhecida. Considerando a problemática da identificação de uma função não linear apropriada para ajuste de um modelo de regressão, o objetivo deste trabalho foi avaliar o desempenho do Modelo de Regressão Wavelet (MW) na detecção de funções não lineares. Desta forma, selecionou-se quatro funções não lineares, tomadas como verdadeiras, dentre um total de 25 funções disponíveis. Em seguida, foi realizado um estudo de simulação Monte Carlo, com 1000 réplicas, em 36 cenários distintos, variando as funções não lineares verdadeiras, o tamanho amostral e a intensidade da relação entre $X$ e $Y$, avaliando-se a taxa de classificação correta do MW. Evidenciamos, nos cenários em que o tamanho amostral é maior e/ou a relação entre $X$ e $Y$ é mais forte, que o MW mostrou-se eficiente na detecção da função não linear geradora dos dados amostrais. Este resultado é análogo para os cenários com relação moderada. Contudo, nos cenários em que a relação não linear era leve, o MW apresentou seu pior desempenho em amostras menores ($n = 128$), melhorando a taxa de acerto em amostras maiores. No geral, tem-se evidências de que o Modelo de Regressão Wavelet obteve ótimo desempenho na detecção de funções não lineares, principalmente quando considerada a medida comparativa Raiz da Média do Quadrado dos Erros (RMQE). Portanto, com base nos resultados obtidos, consideramos a utilização do MW como uma ferramenta importante para identificar a verdadeira função não linear, quando a mesma não é conhecida.

**Palavras-chave:** Regressão, Wavelets, Não Linear.

# Abstract

Nonlinear regression methods represent a relevant research field due your practical applicability in other areas of knowledge. However, defining a nonlinear function, which best defines the data, is not an easy task when the mathematical relationship between the response variable and the independent variables is unknown. Regarding the taste of how to find the best nonlinear function for a regression model, the aim of this work is to evaluate the performance of the Wavelet Regression Model (WM) in the detection of a true nonlinear function. Thus, four nonlinear functions, were selected from a total of 25 used in this work. Then, a Monte Carlo simulation study was performed, taking into account 1000 replicates, 36 different scenarios, four true nonlinear functions, three sample size and degrees of relationship between $X$ and $Y$. We evaluate the true classification rate of the WM.We identify, in the scenarios where the sample size is larger and/or the relation between $X$ and $Y$ is stronger, that the WM was efficient to detect the true nonlinear function. This result is analogous when the relationship between $X$ e $Y$ is moderate. However, the WM presented a bad performance for small samples ($n = 128$), increasing the true classification rate when the sample size increase. In general, there is evidence that the Wavelet Regression Model presented a good performance to detected the true nonlinear functions, specially for the Root Mean Square Error (RMSE) measure. Therefore, based on these results we conclude that the WM is an important tool to identify the true nonlinear regression function when it is unknown.



**Key Words:** Regression, Wavelets, Nonlinear.

# Sumário

# Lista de Tabelas

# Lista de Figuras

# Capítulo 1

# Introdução

Dentro das técnicas estatísticas, a regressão linear se destaca por sua aplicabilidade a problemas reais em diversas áreas. Entretanto, há casos em que um pesquisador dispõe de uma expressão matemática conhecida que relaciona a variável resposta às preditoras de forma não linear em seus parâmetros. Nesses casos, as técnicas de regressão linear não são suficientes, o que acarreta uma considerável complexidade nas estimativas e respectivas inferências (BATES e WATTS, 2007).

A regressão não linear representa um importante tópico dentro da Estatística, bem como em muitas ciências aplicadas, variando da biologia à engenharia, medicina, farmacologia, entre outras. Em seus livros, Ritz e Streibig (2008) e Ryan (2009) trazem vários exemplos destas aplicações práticas.

Geralmente, para o ajuste de um modelo de regressão normal não linear, é necessário o conhecimento da relação entre a variável resposta e as explicativas, entretanto, essa realidade nem sempre é possível. O pesquisador pode não ter ideia de qual é o verdadeiro modelo, mas simplesmente acredita que o mesmo é não linear. Técnicas gráficas são comumente utilizadas na tentativa de ajuste, mas não são as únicas, e nem sempre a mais eficazes (RYAN, 2009).

Esta problemática serve de motivação para o surgimento de novas técnicas diagnósticas, e conhecendo as propriedades das funções Wavelets, é possível considerá-las para tal propósito, haja vista sua vasta aplicabilidade em diversas áreas dentro da própria Estatística.

As Wavelets foram desenvolvidas na análise funcional como bases para o espaço de funções quadraticamente integráveis ($L_2(R)$), bem como alguns de seus subespaços. Estas classes de funções contêm um grande número de elementos diversos, o que as tornam adequadas para aplicações teóricas e numéricas amplas. Por exemplo, elas formam bases incondicionais para algumas classes funcionais grandes, o que leva a estimadores e testes ótimos(VIDAKOVIC, 1999). Além disso, o modelo de regressão Wavelet (MW) é de natureza não-paramétrica, ou seja, não é necessário que os dados utilizados no ajuste estejam distribuídos de acordo com uma distribuição específica (KOVAC e SILVERMAN, 2000).

Considerando a problemática de identificação de uma função não linear apropriada para ajuste de um modelo de regressão, o foco deste trabalho foi avaliar o desempenho

do Modelo de Regressão Wavelet utilizado na detecção da função não linear adequada.

Este Trabalho de Conclusão de Curso está organizado da seguinte forma: O Capítulo 2 traz uma revisão sobre o modelo de regressão não linear e Wavelets. No Capítulo 3, consideramos a metodologia utilizada para aplicação da regressão Wavelet para identificação da melhor função matemática para um modelo de regressão não linear, bem como os parâmetros utilizados no estudo de simulação. No Capítulo 4, analisamos os resultados obtidos no estudo experimental, e no Capítulo 5 apresentamos nossas considerações e conclusões. O código R está disponível como Material Suplementar.

# Capítulo 2

# Referencial Teórico

## 2.1 Modelo Normal Não Linear (MNL)

O modelo de regressão linear é comumente usado para prever valores de uma variável dependente quantitativa ($Y$), como função de valores de variáveis independentes ($X's$) (DRAPER e SMITH, 1981; MONTGOMERY e PECK, 1982). Para adequar este modelo aos dados, é necessário estimar um vetor de parâmetros $\beta$ através do vetor de dados $Y$ e da matriz modelo $X$, presumindo-se $p$ colunas linearmente independentes entre si e $n \gg p$ linhas, onde "$n$" representa o tamanho amostral. A estimação usando o método dos mínimos quadrados é frequentemente usada no caso linear e não requer pressupostos probabilísticos para a variável $Y$. O modelo linear, no entanto, não é adequado em todas as situações. Em alguns casos, a variável resposta $Y$ e as variáveis preditoras $X_j$, $j = 1, 2, ..., p$ necessitam de uma abordagem diferente, que não a regressão linear.

A regressão não linear é utilizada, na estatística, em dados que apresentam relação por meio de combinação não linear dos parâmetros do modelo, que dependem de uma ou mais variáveis independentes (RYAN, 2009).

O modelo de regressão normal não linear simples pode ser definido por:

$$y_i = f(x_i, \beta) + \varepsilon_i, \quad i = 1, 2, ..., n, \tag{2.1}$$

onde $y_i$ corresponde ao $i$-ésimo valor da variável resposta $Y$, $f$ é uma função não linear em relação aos parâmetros, e diferenciável, $x_i$ é o $i$-ésimo valor da variável independente $X$, $\beta$ é o vetor de parâmetros desconhecidos, que será estimado, e $\varepsilon_i$, o $i$-ésimo valor do vetor de erros não observados, numa amostra de tamanho igual a $n$.

Assume-se que os erros são independentes e identicamente distribuídos, seguindo distribuição normal de média $\mu_\varepsilon = 0$ e variância $\sigma^2_\varepsilon$ conhecida. De acordo com a equação (2.1), é possível afirmar que o modelo de regressão normal linear simples é um caso particular de modelo normal não linear, onde a função $f(x_i, \beta)$ é direta e $y_i$ é, consequentemente, dado por $y_i = \beta_0 + \beta_1 x_i + \varepsilon$ (GALLANT, 1982).

Os modelos não-lineares são úteis em alguns campos como ecologia, agricultura, biologia, entre outros. Uma função bastante utilizada e conhecida na bioquímica, no estudo da cinética enzimática, é a equação de Michaelis e Menten (1913),

$$v = \frac{V_{max}[S]}{K_m + S}, \tag{2.2}$$

na qual $V_{max}[S]$ é a taxa mínima obtida pelo sistema na concentração de substrato saturante, $K_m$ é a constante de Michaelis e $S$, o substrato em questão (RITZ e STREIBIG, 2008).

As equações normais para esses tipos de modelos são não lineares e, em geral, o uso de procedimentos iterativos é necessário para ajustar uma regressão desta natureza (BATES e WATTS, 2007; SEBER e WILD, 2003). Desta maneira, valores hipotéticos para o vetor de parâmetros $\beta$, chamados "chutes iniciais" são necessários, gerando assim, a "pseudo equação de predição", que será utilizada como base para o cálculo do erro, iniciando o processo de iteratividade. O processamento termina quando atinge o número máximo de iterações estipulado, ou no momento em que o algoritmo alcança o critério que estabelece convergência.

A essência da otimização utilizando o processo iterativo, está em minimizar a soma dos quadrados dos resíduos, denotada por

$$SQE_{MNL}(\beta) = \sum_{i=1}^{n} (y_i - f(x_i, \beta))^2. \tag{2.3}$$

Vários métodos de otimização podem ser usados para obter as estimativas de parâmetros que minimiza (2.3) (RYAN, 2009). O método de linearização é uma alternativa para estimativa do vetor $\beta$, entretanto, em alguns casos, este método pode não ser adequado devido ao custo computacional ou mesmo por não haver convergências. Sendo assim, outras alternativas são disponibilizadas na literatura, como por exemplo, a técnica de Gradiente Conjugado, Método de Levenberg-Marquardt ou BFGS (algorítmo de Broyden–Fletcher–Goldfarb–Shanno), que usa valores de funções e gradientes para criar uma imagem da superfície a ser otimizada.

### 2.1.1 Método do Gradiente Conjugado

O método do Gradiente Conjugado (GC) é um dos métodos mais populares para resolver problemas de otimização não linear não-limitados. A fórmula do GC, inicialmente desenvolvida por Hestenes e Stiefel (1952), é considerada um método eficiente de otimização. Além disso, o coeficiente Hestenes-Stiefel (HS) está relacionado com a condição de conjugação, independentemente do método da linha de pesquisa utilizada.

O método GC disponível no software estatístico $R$, através da função "`optim`", é o referido por Fletcher e Reeves (1964). O método do Gradiente Conjugado geralmente é mais frágil que o método BFGS (Broyden–Fletcher–Goldfarb–Shanno, também conhecido como algoritmo métrico variável), descrito por Nocedal e Stephen (2006), mas como eles não armazenam a matriz Hessiana da segunda derivada, esta proveniente da função não linear, podem ser bem sucedidos em problemas de otimização muito maiores.

Em qualquer aplicação prática, o tempo gasto na avaliação da função e do gradiente nos vários pontos necessários pode abarcar praticamente todo o tempo do processo de minimização. Logo, limita-se o número de tais avaliações tanto quanto possível. Geralmente utiliza-se 100 como o valor máximo de iterações, dependendo do método de minimização utilizado.

Um determinante no tempo de minimização é o ponto de partida ou "chute inicial". Para funções quadráticas qualquer escolha de ponto de partida é satisfatório, já para funções gerais, o melhor que pode se esperar é que o processo de minimização leve, tão rapidamente quanto possível, para o fundo de qualquer "vale" em que se inicia. Fletcher e Reeves (1964) relatam que em algumas aplicações é possível detectar quando a convergência para um mínimo indesejado ocorreu e há casos em que é desejável extrair os mesmos.

### 2.1.2 Escolha da Função Não Linear

Além do método de otimização adequado, um ponto crucial é a escolha da função não linear $f$, nos casos em que a mesma não é conhecida. Desta forma, o critério de informação Akaike (AIC), definido por Akaike (1974), pode ser usado para comparar dois ou mais modelos ajustados aos mesmos dados. Um procedimento de validação cruzada também pode ser usado para ajudar na escolha entre duas ou mais funções não-lineares. Alguns autores têm considerado esta abordagem em problemas de regressão nos últimos anos (COLBY e BAIR, 2013; SHAO, 1993).

De acordo com Ryan (2009) e Ruckstuhl (2010), alguns testes, também realizados na análise de regressão linear, podem ser usados, utilizando algumas adaptações para o MNL. Dentre eles temos a bondade do ajuste, por meio do teste $F$, análise gráfica dos resíduos padronizados, testes diagnósticos de multicolinearidade, identificação de observações influentes nos dados, e até mesmo o teste $T$ adaptado para as estimativas dos parâmetros do MNL.

## 2.2 Modelo Wavelet

A teoria de Wavelets iniciou-se por volta de 1900, mas o trabalho que unifica todos os variados conceitos por trás dessa teoria, e a torna em uma ferramenta viável para análise de dados foi apresentado por Mallat (1989), sendo conhecida por Análise de Multi-Resolução.

A Análise de Multi-Resolução (AMR) no espaço de funções quadraticamente integráveis ($L_2(R)$) representa uma sequencia de subespaços fechados, $\{V_j\}_{j\in Z}$ com quatro propriedades básicas:

i- **Hierarquia**

$$V_j \subset V_{j+1} \subset L_2(R), \forall j \in Z \tag{2.4}$$

ii- **União densa e interseção trivial**

$$\bigcup_{j\in Z} V_j = L_2(R) \text{ e } \bigcap_{j\in Z} V_j = 0 \tag{2.5}$$

iii- **Similaridade própria**

$$m(2^j t) \in V_j \Leftrightarrow m(t) \in V_0 \quad \forall j \in Z \tag{2.6}$$

iv- **Base natural** $\exists \phi \in V_0$ no qual $T^k \phi(t) = \phi(t-k) \forall k \in Z$ extensível a $V_0$, isto é

$$V_0 = \{m \in L_2(R) | f(t) = \sum_{k \in Z} c_k \phi(t-k)\}, \tag{2.7}$$

para uma sequencia apropriada $c_k k \in Z$, e $\phi(\cdot - k), k \in Z$ é chamada base ortonormal de $V_0$.

$\phi(\cdot)$ é chamada função escala. Isto gera outra base por meio de translação e dilatação: $\phi_j(t) = 2^{j/2}\phi(2^j t - k)$ $j \in Z$ $k \in Z$. O sistema ortogonal $\phi_{j,k}(\cdot)$ se estende a $V_j$ para cada $j$, isto é,

$$V_j = \{m \in L_2(R) | f(t) = \sum_{k \in Z} \alpha_{j,k} \phi_{j,k}(t-k)\}, \forall j \in Z \tag{2.8}$$

para alguma sequência $\alpha_{j,k}$, $k \in Z$, onde $\phi_{j,k}(\cdot), k \in Z$ é base ortonormal para $V_j$ e $\alpha_{j,k} \leq m, \phi_{j,k} > L_2$. Qualquer $m(\cdot)$ em $L_2(R)$ pode ser escrito como

$$m(t) = \lim_{j \to \infty} \sum_{k \in Z} \alpha_{j,k} \phi_{j,k}(t) = \lim_{j \to \infty} P_j m(t), \tag{2.9}$$

no qual $P_j m(t)$ é a projeção ortonormal de $m$ em $V_j$. É fácil de visualizar que $\lim_{j \to -\infty} P_j m(t) = 0$ e $(\phi_{j,b}, \phi_{j,a}) L_2 = \int_{-\infty}^{+\infty} \phi_{j,b}(t) \phi_{j,a}(t) dt = \delta_b^a$, onde $\delta_b^a = 0$ se $a \neq b$, e $\delta_b^a = 1$ se $a = b$. A razão para ampla aplicabilidade das Wavelets deve-se aos filtros associados que apresentam propriedades numéricas interessantes, tal que

$$\phi(t) = \sum_{k \in Z} h_k \phi_{1,k}(t) = \sum_{k \in Z} h_k \sqrt{2} \phi(2t - k), \tag{2.10}$$

onde $h_k = \sqrt{2} \int_R \phi(2t-k) dt$, $k \in Z$ é conhecida como filtro da função escala.

A Análise de Multi-Resolução (AMR) de $L_2(R)$ é chamada $r$-regular, $r \in N$, se a função escala $\phi(\cdot)$, definida por (2.7), é tal que:

$$\phi^{(k)}(t)| \leq \frac{C_m}{(1+|t|)^m}, \forall k \leq r \quad \forall k \in N \quad \forall m \in N. \tag{2.11}$$

Outro filtro $g_k$ é definido a partir de $h_k$, através da relação de quadratura espelhada (RQE): $g_n = (-1)^n H_{1-n}$. É possível escrever $g_k = \sqrt{2} \int_R \psi(t) \phi(2t-k) dt \quad \forall k \in Z$ e $\{\psi_{j,k}(t) = 2^{\frac{1}{2}} \psi(2^j t - k), j \in Z, k \in Z\}$ extensível a $L_2(R)$. Tem-se que $W_j = \{m \in L_2(R) / |m(t) \stackrel{L_2}{=} \sum_{k \in Z} \beta_{j,k} \psi_{j,k}(t)\}$. Então, $V_{j+1} = V_j \oplus W_j, \forall j \in Z$, em que $\oplus$ é a soma direta entre vetores, e

$$L_2(R) = \bigoplus_{i \in Z} W_j. \tag{2.12}$$

Consequentemente, qualquer função $m \in L_2(R)$ pode ser escrita na perspectiva de $L_2$ como:

$$m(t) = \sum_{j \in Z} \sum_{k \in Z} \beta_{j,k} \psi_{j,k}(t) = \sum_{k \in Z} \alpha_{j_0,k} \phi_{j_0,k}(t) + \sum_{j \geq j_0 Z} \sum_{k \in Z} \beta_{j,k} \psi_{j,k}(t) \tag{2.13}$$

para um $j_0$ arbitrário.

A escolha da base Wavelet depende em vários aspectos. A regularização da Wavelet é muito importante em problemas de otimização estatística, e pode ser avaliada pela quantidade de momentos nulos

$$\mathcal{M}_k = \int_R t^k \psi(t) dt. \tag{2.14}$$

Entretanto $\phi$ e $\psi$ possuem $N$ momentos nulos se, e somente se,

$$\sum_{n \in Z} n^k g_n = \sum_{n \in Z} n^k (-1)^n h_n = 0, \quad \text{para} \quad k = 0, 1, ..., N-1. \tag{2.15}$$

Em geral, os filtros possuem um número infinito de termos não-nulos. Duas classes especiais são os N-regular e as wavelets de suporte compacto. Em ambos os casos, o número de termos não-nulos é igual a $2N$ (DAUBECHIES, 1992). Uma família de funções wavelets de suporte compacto é a família *Daubechies*, e um caso particular é a base Haar, também conhecida como a primeira Wavelet, sendo definida por $\phi(t) = 1_{[0,1]}(t)$ e $\psi(t) = 1_{[0,1/2]}(t) - 1_{[1/2,1]}(t)$, ou por $h_0 = h_1 = \sqrt{2}/2$ e $g_0 = \sqrt{2}/2$, $g_1 = -\sqrt{2}/2$.

As *Daubechies* são indexadas pelo número de momentos nulos $N$ ($Daubechies(N)$), com base $[0, 2N-1]$ e filtros associados de tamanho $2N$. Por exemplo, para $Daubechies(2)$,

$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}, h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}, h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}, h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}}. \tag{2.16}$$

As *Daubechies* não possuem formas fechadas, exceto as da base Haar. Por esta razão, utiliza-se o algoritmo de cascata de Daubechies-Lagaria, que permite o cálculo de qualquer $\phi(t)$ para $t \in R$, com qualquer precisão pré-determinada. Considere $\phi(\cdot)$ a função de escala para a base da $Daubechies(N)$ e $\{h_k\}_{k \in R}$ seu filtro associado. Para qualquer $t \in (0,1)$ e $\{d_1, d_2, ...\}$ a representação diádica de $t$, definido por $t = \sum_{j=1}^{\infty} d_j 2^{-j}$, nós definimos as matrizes $T_0$ e $T_1$ como:

$$T_0 = (\sqrt{2} h_{2i-j-1})_{1 \leq i,j \leq 2N-1} T_1 = (\sqrt{2} h_{2i-j})_{1 \leq i,j \leq 2N-1}. \tag{2.17}$$

Então, $\lim_{n \to \infty} T_{d_1} ... T_{d_n}$

$$= \begin{bmatrix} \phi(t) & \phi(t) & \dots & \phi(t) \\ \phi(t+1) & \phi(t+1) & \dots & \phi(t+1) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(t+2N-2) & \phi(t+2N-2) & \dots & \phi(t+2N-2) \end{bmatrix} \tag{2.18}$$

A classe das funções quadráticas integráveis é, em geral, grande e diversa para ser de interesse na prática. Todavia há espaços menores que são suficientemente grandes para serem úteis em um bom número de problemas, mas ainda possuem condições de regularidade que são relevantes. Dois desses subespaços são os espaços de Hölder e Besov, denotados por $\mathcal{H}_\alpha(R)$ e $\mathcal{B}^{\mathbf{s}}_{p,q}$.

É importante mencionar que algumas funções $m$ pertencem a $\mathcal{H}_\alpha(R)$ (ou $\mathcal{B}^{\mathbf{s}}_{p,q}$) se, e somente se, seus coeficientes Wavelet seguem uma certa lei de decadência. A base da Wavelet é então chamada de base incondicional para $\mathcal{H}_\alpha(R)$ (ou $\mathcal{B}^{\mathbf{s}}_{p,q}$). Em aplicações, esta propriedade resulta na análise dos coeficientes estimados ordenadamente, para avaliar o grau de regularidade dos dados. Isto leva à redução de coeficientes empíricos, além da otimização de estimativas baseadas em Wavelet e procedimentos de teste em sentido minimax (MORETTIN, PINHEIRO e VIDAKOVIC, 2016; VIDAKOVIC, 1999).

Finalmente, um modelo de regressão não-paramétrico baseado em Wavelet foi proposto por Kovac e Silverman (2000). É possível aplicar a regressão Wavelet a conjuntos de dados que não são igualmente espaçados entre si. Para tando, inicialmente definimos uma grade de pontos como $\tilde{t}_k = (K + 1/2)2^{-J}$, onde $k \in \{0, .., 2^J - 1\}$. Os valores da variável resposta baseados na grade de pontos são então calculados como $\tilde{y}_k$, a partir de uma transformação linear dos $y$'s originais. Usamos simplesmente como $\tilde{y}_k$ as observações que estão em $[k2^{-J}, (k + 1)2^{-J}]$. Sempre que nenhuma observação puder ser encontrada no intervalo formado pela grade de pontos, tomamos a observação mais próxima, à esquerda do mesmo. Deste modo, transformamos dados não equidistantes em dados igualmente espaçados e, além disso, isto é feito de modo a produzir um tamanho de amostra que seja uma potência de 2. Assim, as técnicas da transformação discreta da Wavelet podem ser empregadas. Um limiar (*thresholding*) é aplicado nos coeficientes estimados, e escrevemos o estimador do modelo de regressão Wavelet como

$$\hat{m}(x) = \sum_{k=0}^{2^{j_0}} \hat{c}_{j_0 k}\psi_{j_0 k}(x) + \sum_{j \geq j_0}^{J_{max}-1} \sum_{k=0}^{2^j} \hat{d}^{trh}_{jk}\psi_{jk}(x), \tag{2.19}$$

onde $\hat{c}_{j_0 k}$ são as aproximações estimadas dos coeficientes e $\hat{d}^{trh}_{jk}$ são os coeficientes que receberam a aplicação do limiar (*threshold*), para a $j$-ésima escala (KOVAC e SILVERMAN, 2000).

# Capítulo 3

# Metodologia

Para avaliar o desempenho do Modelo de Regressão Wavelet (MW) na identificação de funções não lineares, foram consideradas algumas funções não lineares conhecidas. Neste trabalho levamos em conta um total de 25 funções não lineares, denotadas no Apêndice A.

Em seguida, selecionamos 4 funções não lineares para gerarmos os dados simulados. Foram elas:

$$Y_1 = f_1(x, \beta) = \frac{\beta_1}{\beta_2 + e^{\beta_3 x}}; \tag{3.1}$$

$$Y_2 = f_2(x, \beta) = \beta_1 + e^{-\beta_2 x}; \tag{3.2}$$

$$Y_3 = f_3(x, \beta) = \frac{\beta_2 x}{\beta_1 + x}; \tag{3.3}$$

$$Y_4 = f_4(x, \beta) = \beta_1 \cos(2x) + \beta_2 \sin(x). \tag{3.4}$$

A relação matemática (3.1) representa uma função logística com aplicabilidade em problemas relatados na Medicina e áreas da Saúde. A expressão (3.2) é uma função exponencial que tem aplicações em problemas na área industrial, como por exemplo, os que envolvem confiabilidade. Por sua vez, a relação matemática (3.3) tem vasta aplicação na área química e a função (3.4) foi escolhida por se mostrar graficamente diferenciada das outras 24 funções não lineares (MICHAELIS e MENTEN, 1913).

## 3.1 Geração dos Dados Sintéticos

Para a construção da variável $Y$ a partir das expressões (3.1) a (3.4), consideramos os seguintes parâmetros: o vetor $x$ foi gerado a partir de uma distribuição uniforme $X \sim U(a, b)$, delimitada pelo valor mínimo ($a$) e máximo ($b$). A intensidade da relação entre $X$ e $Y$ foi determinada por meio da variância do vetor de resíduos ($\varepsilon$) que, por definição, seguem distribuição normal com média zero e variância conhecida, $\varepsilon \sim N(0, \sigma_\varepsilon^2)$.

Quanto menor a variância, mais forte a relação entre $X$ e $Y$. Alterando a variância para um valor mais elevado, tem-se a relação moderada. E por fim, a variância alta acarreta em uma relação aqui classificada como leve. Estes valores foram determinados de modo empírico, visualizando-se os gráficos gerados para cada função e para cada variância definidas nos erros. Definimos como relação forte, os cenários cujos gráficos de dispersão entre $X$ e $Y$ apresentaram observações justapostas e precisa delineação da curva. Para definir os cenários de intensidade de relação leve, tivemos o cuidado em não aumentar a variância dos erros ao ponto de que, graficamente, não pudéssemos visualizar que havia relação não linear entre as variáveis. Os cenários de relação moderada foram definidos por meio da visualização de dispersão intermediária das observações geradas, quando comparado com os gráficos de relação forte e leve. A Tabela 3.1 traz os valores utilizados para os parâmetros $\beta$ e as informações acima mencionadas.

Tabela 3.1: Parâmetros utilizados para gerar os valores de $x$ e $y$ em simulação.

| Funções | $X \sim U(a,b)$ | | Relação entre $X$ e $Y$ | | | vetor $\beta$ | | |
|---|---|---|---|---|---|---|---|---|
| | $a$ | $b$ | Forte | Moderada | Leve | $\beta_1$ | $\beta_2$ | $\beta_3$ |
| $f.1$ | $-6.00$ | 6.00 | $\varepsilon \sim N(0, 0.01)$ | $\varepsilon \sim N(0, 0.1)$ | $\varepsilon \sim N(0, 0.2)$ | 2.00 | 3.00 | 1.000 |
| $f.2$ | 1.00 | 4.00 | $\varepsilon \sim N(0, 0.005)$ | $\varepsilon \sim N(0, 0.03)$ | $\varepsilon \sim N(0, 0.06)$ | 0.25 | 1.00 | – |
| $f.3$ | 5.00 | 210.00 | $\varepsilon \sim N(0, 1)$ | $\varepsilon \sim N(0, 5)$ | $\varepsilon \sim N(0, 10)$ | 20.00 | 120.00 | – |
| $f.4$ | 0.00 | 4.00 | $\varepsilon \sim N(0, 0.1)$ | $\varepsilon \sim N(0, 1)$ | $\varepsilon \sim N(0, 2)$ | 4.00 | 1.00 | – |

## 3.2 Ajuste do Modelo de Regressão Wavelet

Após gerar $X$ e $Y$, baseados nos parâmetros descritos na Tabela 3.1, procedeu-se o tratamento destes dados antes do ajuste do Modelo de Regressão Wavelet (MW).

A regressão Wavelet exige que os dados encontrem-se equidistantes. Um método para assim torná-los é descrito por Kovac (1997) e Kovac e Silverman (2000). Desta forma, foi necessário transformar os valores da variável independente, por meio da seguinte transformação

$$x^* = \frac{x - \min(x)}{(\max(x) - \min(x))}, \tag{3.5}$$

e utilizar o algoritmo de Kovac-Silverman, o qual toma um par de dados $(x, y)$, com $x$ arbitrário no intervalo $(0, 1)$, e interpola linearmente $(x, y)$ para uma grade diádica igualmente espaçada.

A próxima etapa compreende no ajuste do MW aos dados interpolados. Isto é feito através da função "`irregwd`", do pacote "`wavethresh`", disponível em $R$. No entanto, estes dados serão correlacionados devido à interpolação, e isso precisa ser levado em conta. Depois de tomar o valor estimado pelo MW e antes do *thresholding*, percebe-se que cada coeficiente tem sua própria variância. O algoritmo de Kovac-Silverman calcula essa variância eficientemente usando o conhecimento do esquema de interpolação.

Quando aplicamos a função "`threshold.irregwd`", a mesma faz uso das informações sobre a variância de cada coeficiente para avaliar se os mesmos apresentam significância

estatística, descartando do MW os coeficientes não significativos. Nesta função, utilizamos *threshold* do tipo forte, que é mais rígido na modificação dos coeficientes, e a função `madmad`, que é mais robusta e retorna o quadrado da função mediana de desvio absoluto. Isto é necessário nos cálculos intrínsecos do *threshold*. O argumento para política utilizada foi o Universal, que é implementado quando se deseja que sua estimativa de variância de limiar use apenas os melhores coeficientes de escala.

Para comparar a estimativa do MW ($\hat{y}^{MW}$) com as funções não lineares, após aplicar o limiar, é necessário reorganizar os valores da variável resposta, que são retornados em ordem diferente da variável originalmente observada. Fizemos isto por meio do comando $\hat{y}^{MW} = \hat{y}^{MW}[rank(x)]$.

Agora, otimizamos as funções não lineares utilizando o método de Gradiente Conjugado, por meio da função "`optim`", disponível no `R`. Para isto, usamos como base os valores de $x$ e $y$ gerados com as funções verdadeiras. O objetivo é calcular a estimativa do vetor de parâmetros $\beta$ para cada função não linear, e assim estimar $y$, comparando a estimativa da variável resposta obtida pelo MW com as estimativas obtidas para cada função não linear considerada.

## 3.3 Medidas Comparativas

O desempenho do MW foi avaliado por meio do cálculo da Raiz da Média do Quadrado dos Erros (RMQE) e pelo Erro Absoluto Mediano (EAM) definidos pelas expressões (3.6) e (3.7), respectivamente:

$$RMQE_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i^{MW} - \hat{y}_{ij}^{MNL})^2}, \quad j = 1, 2, ..., 25; \tag{3.6}$$

$$EAM_j = \text{mediana}(|\hat{y}_i^{MW} - \hat{y}_{ij}^{MNL}|), \quad j = 1, 2, ..., 25. \tag{3.7}$$

Note que as medidas de performance são baseadas na diferença entre os valores estimados pelo MW ($\hat{y}_i^{MW}$, para $i = 1, 2..., n$) e os respectivos valores estimados pelo $j$-ésimo modelo de regressão paramétrico ($\hat{y}_{ij}^{MNL}$, para $i = 1, 2..., n$ e $j = 1, 2, ..., 25$). Desta forma, para cada base de dados simulada, será ajustada ao MW e cada um dos 25 modelos não lineares. Para cada modelo não linear, será calculada a respectiva $RMSE_j$ e $EAM_j$, que compara os valores preditos do MNL com os valores preditos pelo MW.

Em nosso critério de avaliação, o MW terá um bom desempenho caso o menor valor de $RMSE_j$ e/ou $EAM_j$ corresponda à função não linear verdadeira. Isto significa que o MW estará se aproximando mais da verdadeira função e, consequentemente, é possível identificá-la por meio dele. Ao final, apresentaremos a porcentagem de vezes em que os menores valores das medidas comparativas ocorreram para a verdadeira função não linear.

## 3.4 Simulação

Além de considerar quatro funções não lineares ($f_1$, $f_2$, $f_3$ e $f_4$) e a intensidade da relação entre $X$ e $Y$ (forte, moderada e leve), decidimos utilizar três tamanhos amostrais a fim de avaliar o desempenho do MW em quantidades amostrais diferentes. Considerando que, por motivos computacionais inerentes ao MW, é recomendado que o tamanho amostral seja um número que pode ser descrito como uma potência de 2, escolhemos $n_1 = 2^7 = 128$, $n_2 = 2^8 = 256$ e $n_3 = 2^9 = 512$. Logo foram realizadas simulações de Monte Carlo em 36 cenários distintos.

O estudo por meio da técnica de Monte Carlo consiste em simular cenários, que se aproximam de situações reais, inúmeras vezes. Na Estatística, esta técnica é utilizada por meio de um número grande de amostragens aleatórias sucessivas, realizadas em dados de mesma natureza, a fim de avaliar, com maior precisão, o comportamento do objeto de estudo, ao longo das simulações (ECKHARDT, 1987).

Adotamos $m = 1000$ para o número de réplicas Monte Carlo, aplicadas em cada um dos 36 cenários aqui descritos. No Apêndice B, encontra-se o script, utilizado no programa $R$.

# Capítulo 4

# Resultados e Discussões

As figuras a seguir permitem a visualização dos cenários com relação forte, moderada e leve, entre $X$ e $Y$, além dos distintos tamanhos amostrais. Também é possível observar os valores estimados pelo MW, e compará-los empiricamente com os dados de origem.

Quando consideramos a expressão $f_1$, percebe-se, pela Figura 4.1, que os valores estimados pelo MW se aproximam dos dados verdadeiros, conseguindo captar as variações da curva ao longo dos eixos nos gráficos. Contudo, em menor tamanho amostral, $n = 128$, e relação leve entre $(x, y)$, o MW teve dificuldade em delinear uma curva mais nítida, apresentando descontinuidade nos valores estimados, quando nas extremidades do eixo $x$.

A Tabela 4.1 apresenta os resultados obtidos em simulação para a estrutura de regressão $f_1$. Considerando uma relação não linear forte e moderada, o MW identificou a função como correta em todos os cenários analisados, independente do tamanho amostral. Entretanto, quando consideramos uma relação leve entre $X$ e $Y$ e um tamanho amostral $n = 128$, o MW identificou a função correta apenas 7% das vezes, de acordo com a RMQE e nenhuma vez, segundo o EAM. Todavia, a partir de $n = 256$ o MW passa a identificar $f_1$ como função verdadeira em 100% das vezes para a RMQE e em 96.4% das vezes para o EAM.

Figura 4.1: Relação empírica entre as variáveis $X$ e $Y$ baseado na função não linear $f_1$ e valores estimados pelo MW, segundo tamanho amostral e intensidade da relação não linear.

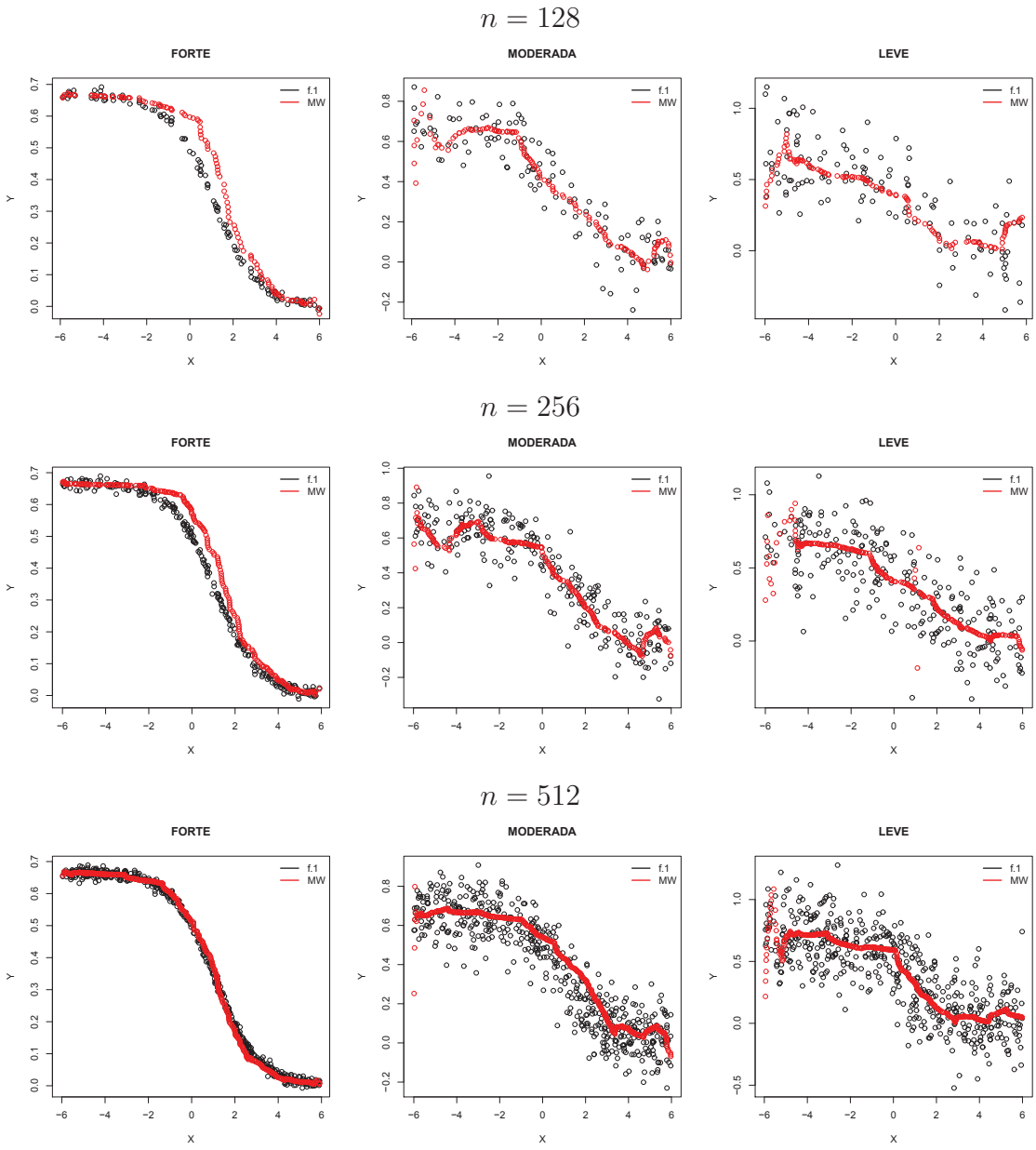
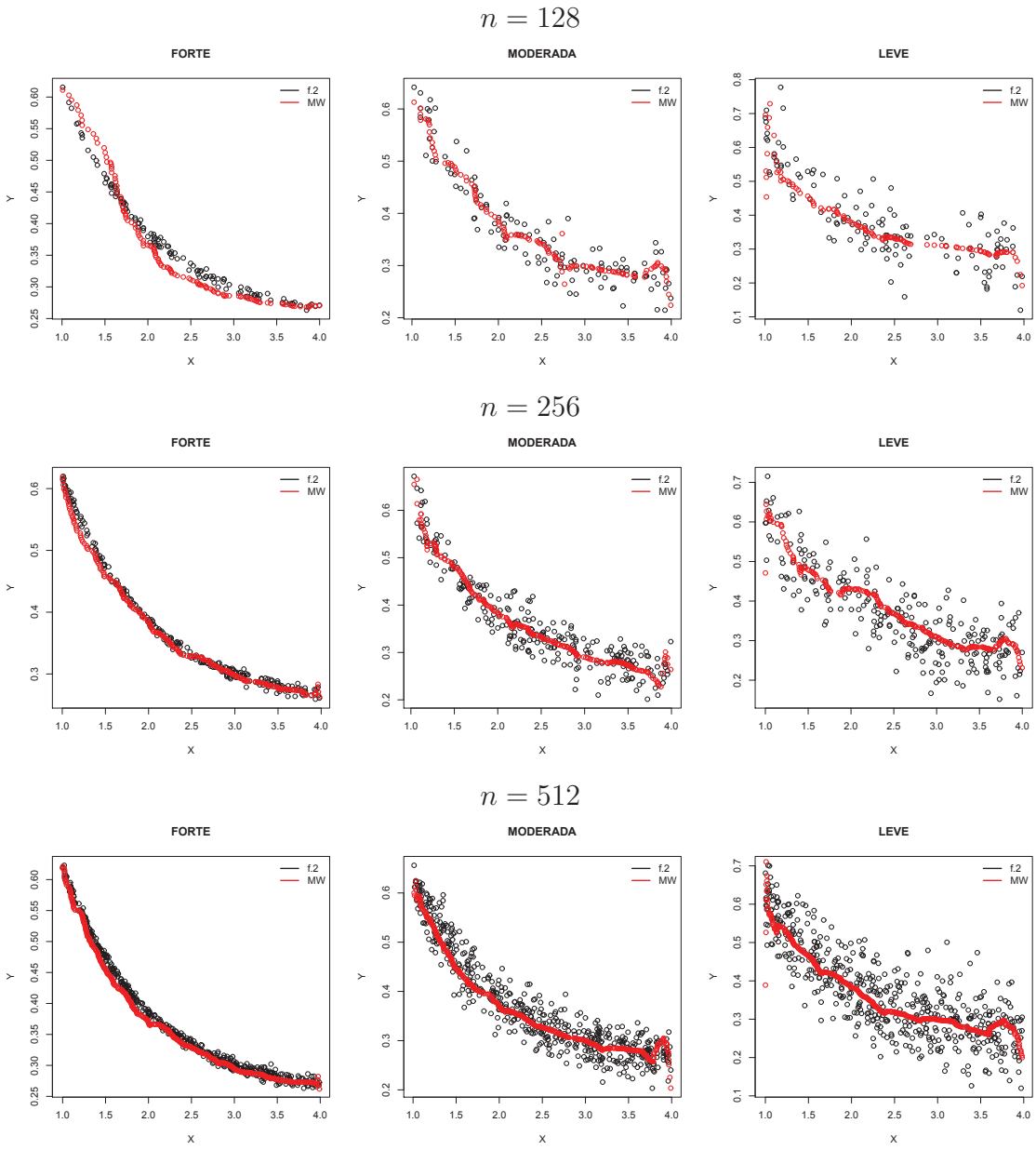



Tabela 4.1: Percentual de acerto do MW segundo a medida comparativa e tamanho da amostra. Função não linear $f_1$.

| | Forte | | Moderada | | Leve | |
|---|---|---|---|---|---|---|
| $n$ | RMQE | EAM | RMQE | EAM | RMQE | EAM |
| 128 | 100.0 | 100.0 | 100.0 | 100.0 | 7.0 | 0.0 |
| 256 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 96.4 |
| 512 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |

Nos cenários onde adotamos a função $f_2$ como verdadeira, verifica-se, pela Figura 4.2, que o MW cresce na acurácia das estimativas, proporcionalmente ao aumento no tamanho amostral e na intensidade da relação entre $X$ e $Y$. Entretanto, quando o tamanho da amostra é pequeno e/ou a relação é leve, o MW começa a apresentar valores estimados com "quebras" na continuidade da curva, o que pode interferir negativamente no desempenho de identificação da função verdadeira através do MW.

Figura 4.2: Relação empírica entre as variáveis $X$ e $Y$ baseado na função não linear $f_2$ e valores estimados pelo MW, segundo tamanho amostral e intensidade da relação não linear.




As representações gráficas presentes na Figura 4.2 corroboram com os resultados expostos na Tabela 4.2, que diz respeito às porcentagens de acerto do MW na identificação da função $f_2$. Assim como na identificação da função $f_1$, os resultados também foram satisfatórios nos cenários de relação forte, onde o MW conseguiu identificar a função verdadeira em todas as simulações, considerando a RMQE.

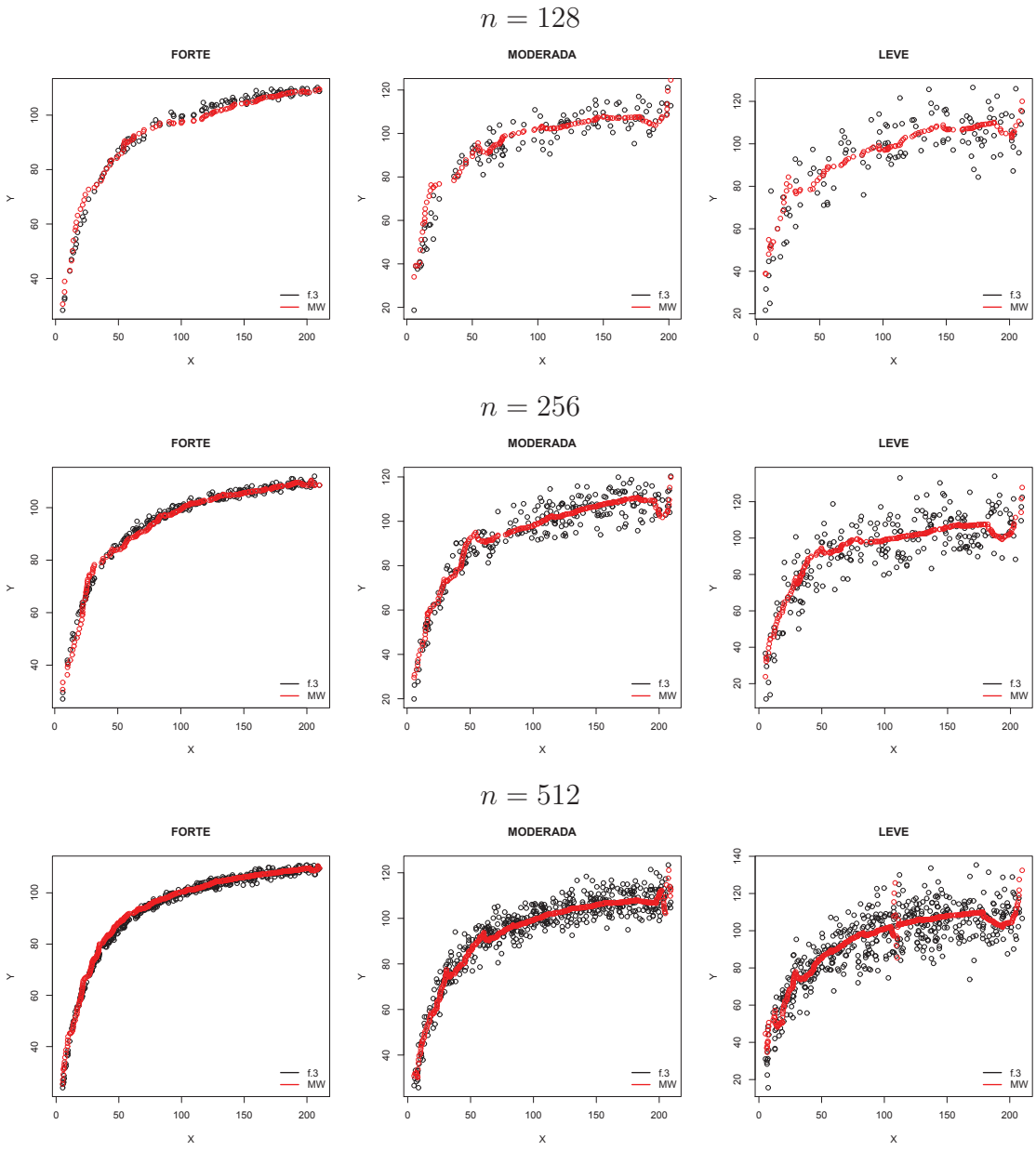
Nas situações de relação moderada entre $X$ e $Y$, houve um erro de classificação de 31.4% considerando o EAM para o cenário de menor tamanho amostral ($n = 128$). Contudo, mesmo tomando um cenário com relação leve entre $X$ e $Y$, a porcentagem de classificação correta através do MW foi maior ou igual a 99.6%, de acordo com a RMQE,o que demonstra que essa medida apresentou uma melhor performance quando comprada ao EAM.

Tabela 4.2: Percentual de acerto do MW segundo a medida comparativa e tamanho da amostra. Função não linear $f_2$.

| | Forte | | Moderada | | Leve | |
|---|---|---|---|---|---|---|
| $n$ | RMQE | EAM | RMQE | EAM | RMQE | EAM |
| 128 | 100.0 | 100.0 | 99.9 | 99.6 | 99.7 | 68.6 |
| 256 | 100.0 | 99.8 | 100.0 | 99.8 | 99.7 | 99.8 |
| 512 | 100.0 | 100.0 | 100.0 | 99.8 | 99.6 | 99.9 |

As representações gráficas da função não linear $f_3$ encontram-se na Figura 4.3, bem como a disposição dos respectivos valores estimados pelo MW. Nota-se que no cenário de relação forte, mesmo para o menor tamanho amostral, o MW consegue captar pequenos desvios na curva original e estimar valores semelhantes. Todavia, percebe-se que nos cenários em que a intensidade da relação entre $X$ e $Y$ diminui, o MW delineia um pequeno decaimento quando a variável independente $X$ apresenta valores dentro do intervalo $[175; 210]$, e este decaimento torna-se mais evidente quando o tamanho amostral aumenta.

Figura 4.3: Relação empírica entre as variáveis $X$ e $Y$ baseado na função não linear $f_3$ e valores estimados pelo MW, segundo tamanho amostral e intensidade da relação não linear.




É possível evidenciar pela Tabela 4.3 que, dentre os cenários envolvendo a função $f_3$ como verdadeira, o MW obteve 100% de acertos na identificação da mesma, nas situações de relação forte, independente do tamanho amostral. Por sua vez, quando a relação é

moderada, este resultado se repete apenas quando consideramos a RQME, pois conforme o EAM, houveram 5 erros de identificação no cenário de tamanho amostral $n = 128$ e 22 erros, quando $n = 512$.

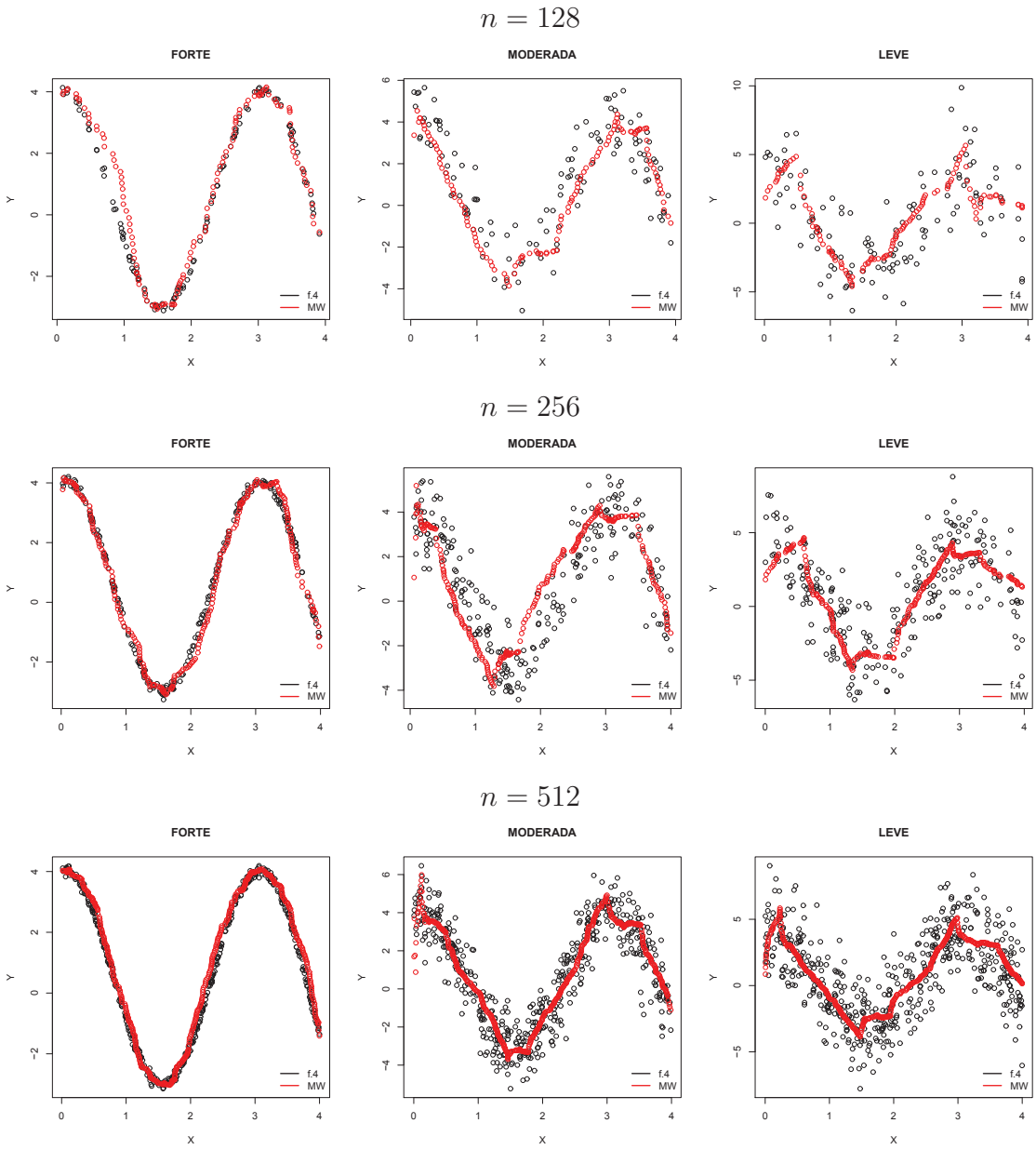
Para os cenários em que a relação entre $X$ e $Y$ foi considerada leve, o MW também obteve desempenho de 100% de acertos, de acordo com a RMQE, nos tamanhos amostrais $n = 128$ e $n = 256$. Todavia, registramos um resultado atípico, quando $n = 512$. Observamos que o MW não identificou a função verdadeira nenhuma vez, conforme a RMQE. Entretanto, quando tomamos EAM como medida comparativa, o MW obteve 99.3% de sucesso na identificação da função correta.

Tabela 4.3: Percentual de acerto do MW segundo a medida comparativa e tamanho da amostra. Função não linear $f_3$.

| | Forte | | Moderada | | Leve | |
|---|---|---|---|---|---|---|
| $n$ | RMQE | EAM | RMQE | EAM | RMQE | EAM |
| 128 | 100.0 | 100.0 | 100.0 | 99.5 | 100.0 | 79.4 |
| 256 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| 512 | 100.0 | 100.0 | 100.0 | 97.8 | 0.0 | 99.3 |

Finalmente, utilizamos a relação $f_4$ com o propósito de avaliar o desempenho do MW na identificação de uma expressão matemática completamente diferenciada das demais, devido as relações trigonométricas presentes nesta função. Realizando a análise gráfica, por meio da Figura 4.4, visualiza-se a sinuosidade ds dados gerados pela expressão $f_4$. Também é possível evidenciar que, quanto maior o tamanho amostral e/ou mais forte a relação entre $X$ e $Y$, o valores estimados pelo MW se delineiam próximo à curva dos dados simulados.

Figura 4.4: Relação empírica entre as variáveis $X$ e $Y$ baseado na função não linear $f_4$ e valores estimados pelo MW, segundo tamanho amostral e intensidade da relação não linear.




Em contrapartida, evidencia-se que em cenários com relações mais fracas e tamanhos amostrais menores, os valores estimados pelo MW não traçam uma curva sinuosa de maneira contínua, apresentando picos ou decaimentos acentuados, ou ainda, se desviando levemente dos dados simulados. É possível verificar isto nos cenários de intensidade de

relação moderada ou leve, com tamanhos amostrais $n = 256$ e/ou $n = 128$.

A Tabela 4.4 apresenta os resultados de desempenho do MW na identificação da função não linear $f_4$. Em virtude da função $f_4$ apresentar um comportamento bem diferente das demais funções consideradas, por conta das relações trigonométricas, a porcentagem de acertos foi de 100% em quase todos os cenários construídos, principalmente quando consideramos tamanhos amostrais maiores e/ou uma relação mais forte. Apesar de termos visualizado, na Figura 4.4, leves desvios nas estimativas do MW para o cenário com relação moderada e amostra de tamanho $n = 128$ e $n = 256$, o MW conseguiu identificar $f_4$ como função originadora dos dados amostrais. Contudo, houveram exceções, como no caso em que $n = 128$ e a relação entre $X$ e $Y$ é fraca. Neste cenário o MW não identificou a função correta em nenhuma das simulações, tanto para a RMQE quanto para o EAM. A outra exceção, refere-se ao resultado discrepante de 0.0% de acerto, no cenário de relação forte e $n = 512$, quando considerado o EAM.

Tabela 4.4: Percentual de acerto do MW segundo a medida comparativa e tamanho da amostra. Função não linear $f_4$.

| | Forte | | Moderada | | Leve | |
|---|---|---|---|---|---|---|
| $n$ | RMQE | EAM | RMQE | EAM | RMQE | EAM |
| 128 | 100.0 | 100.0 | 100.0 | 100.0 | 0.0 | 0.0 |
| 256 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| 512 | 100.0 | 0.0 | 100.0 | 100.0 | 100.0 | 100.0 |

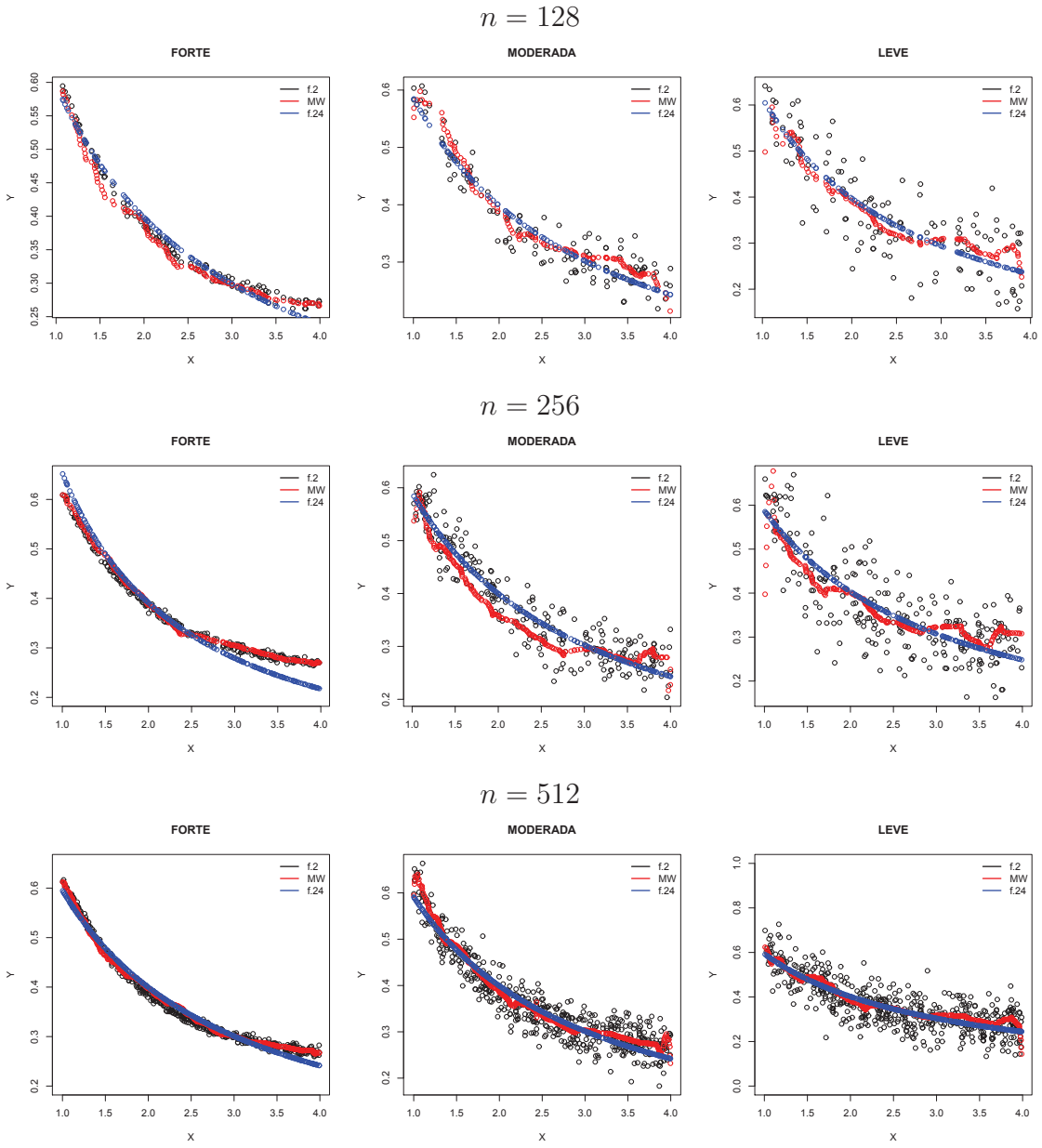
Por último, ao compararmos as medidas de performance utilizadas a Raiz da Média do Quadrado dos Erros (RMQE) apresentou um melhor percentual de classificações corretas quando comparada ao Erro Absoluto Mediano (EAM), principalmente para tamanhos amostrais menores e/ou quando a relação entre $X$ e $Y$ era leve.

## 4.1 Performance do MW em Funções Não Lineares Semelhantes

Aqui pretendemos ilustrar situações em que duas ou mais funções não lineares são semelhantes em suas respostas estimadas, pois esta situação pode acarretar pelo MW, dificuldade em identificar a correta estrutura de regressão que gerou os dados amostrais.

Neste estudo, temos a expressão $f_{24}$ e a relação matemática $f_2$, que apresentam valores semelhantes nos cenários em que a função $f_2$ foi tomada como verdadeira. Este fato pode ser visualizado pela Figura 4.5, que mostra o comportamento da $f_{24}$ ao longo dos cenários testados, bem como dos valores estimados pelo MW.

Figura 4.5: Relação empírica entre as variáveis $X$ e $Y$ baseado na função não linear $f_2$ e valores estimados pelo MW e $f_{24}$, segundo tamanho amostral e intensidade da relação não linear.



Nos cenários em que a relação de dependência entre $X$ e $Y$ foi tomada como forte, os valores estimados pelo MW, quando comparados aos da função $f_{24}$, demonstraram-se adequar melhor à curvatura gerada pelos dados da expressão $f_2$, independentemente do tamanho amostral. Entretanto, em cenários com tamanhos amostrais de $n \leq 256$ e relação de moderada a leve, os valores estimados por $f_{24}$ aparentam maior continuidade no decurso do eixo $X$, enquanto os originados pelo MW mostram-se levemente inconstantes ao longo da curva.

Entretanto, de acordo com os dados apresentado na Tabela 4.5 é possível verificar que a menor porcentagem de acerto do MW foi no cenário em que a relação entre $X$ e $Y$ é leve, e quando considerado o EAM (91.8%). Ainda assim, mesmo que a função $f_{24}$ tenha apresentado uma predição, visualmente, semelhante à $f_2$, o MW identificou a função verdadeira de maneira satisfatória nos demais cenários considerados.

Tabela 4.5: Percentual de acerto do MW segundo a medida comparativa e tamanho da amostra. Tomando $f_2$ como função verdadeira e comparando com os valores estimados pela função $f_{24}$.

| | Forte | | Moderada | | Leve | |
|---|---|---|---|---|---|---|
| $n$ | RMQE | EAM | RMQE | EAM | RMQE | EAM |
| 128 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 91.8 |
| 256 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| 512 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |

Considerando que $f_{24}$ interferiu, mesmo que minimamente, na detecção da função geradora dos dados amostrais, este caso nos abriu visão para outros projetos, como por exemplo: tentar expandir e modificar o script (Apendice B), na finalidade de que a saída do algoritmo sejam as funções testadas e suas respectivas probabilidades de ser a função verdadeira. Desta forma, o pesquisador teria como selecionar as funções de maiores probabilidades, e realizar as técnicas diagnósticas usuais para verificar o melhor ajuste.

## 4.2 Aplicação em Dados Reais

## 4.3 Aplicação em Base de Dados Reais

Considerando a importância da aplicabilidade em problemas reais, selecionamos um banco de dados disponível na literatura para testar o MW na identificação da função adequada. Utilizamos os dados apresentados por Dudzinski e Mykytowycz (1961) que, a posteriori, foram estudados por Ratkowsky (1983), com base em um modelo de regressão normal não linear.

O banco de dados *"coelhos europeus"*, disposto na Tabela 4.6, é composto por duas variáveis (explicativa e resposta), onde cada uma contém 71 observações ($n = 71$). A variável resposta ($Y$) corresponde ao peso das lentes (em mg) dos olhos de coelhos europeus (*Oryctolagus Cuniculus*) e a variável explicativa ($X$) refere-se à idade (em dias) dos coelhos.

Tabela 4.6: Valores referentes às variáveis $X$ e $Y$ do banco de dados *"coelhos europeus"*.

| $x$ | $y$ | $x$ | $y$ | $x$ | $y$ | $x$ | $y$ | $x$ | $y$ | $x$ | $y$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 21.66 | 61 | 73.09 | 98 | 104.30 | 218 | 174.18 | 285 | 189.66 | 513 | 203.30 |
| 15 | 22.75 | 64 | 79.09 | 125 | 134.90 | 218 | 173.03 | 300 | 186.09 | 535 | 209.70 |
| 15 | 22.30 | 65 | 79.51 | 142 | 130.68 | 219 | 173.54 | 301 | 186.70 | 554 | 233.90 |
| 18 | 31.25 | 65 | 65.31 | 142 | 140.58 | 224 | 178.86 | 305 | 186.80 | 591 | 234.70 |
| 28 | 44.79 | 72 | 71.90 | 147 | 155.30 | 225 | 177.68 | 312 | 195.10 | 648 | 244.30 |
| 29 | 40.55 | 75 | 86.10 | 147 | 152.20 | 227 | 173.73 | 317 | 216.41 | 660 | 231.00 |
| 37 | 50.25 | 75 | 94.60 | 150 | 144.50 | 232 | 159.98 | 338 | 203.23 | 705 | 242.40 |
| 37 | 46.88 | 82 | 92.50 | 159 | 142.15 | 232 | 161.29 | 347 | 188.38 | 723 | 230.77 |
| 44 | 52.03 | 85 | 105.00 | 165 | 139.81 | 237 | 187.07 | 354 | 189.70 | 756 | 242.57 |
| 50 | 63.47 | 91 | 101.70 | 183 | 153.22 | 246 | 176.13 | 357 | 195.31 | 768 | 232.12 |
| 50 | 61.13 | 91 | 102.90 | 192 | 145.72 | 258 | 183.40 | 375 | 202.63 | 860 | 264.70 |
| 60 | 81.00 | 97 | 110.00 | 195 | 161.10 | 276 | 186.26 | 394 | 224.82 | | |

Fonte: DUDZINSKI e MYKYTOWYCZ, 1961.

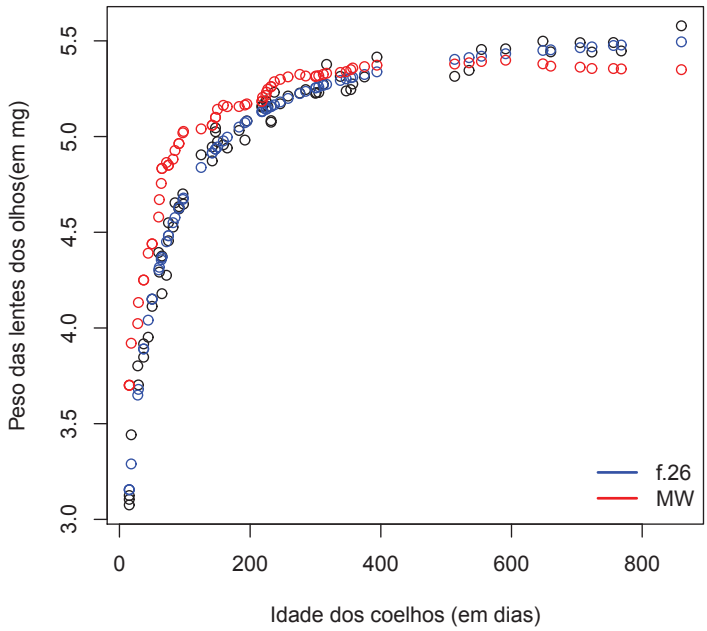
Para testar a identificação da função não linear adequada a estes dados, tomamos como verdadeiro, o modelo sugerido por Galea et al. (2005):

$$log(y_i) = \beta_1 - \frac{\beta_2}{x_i + \beta_3} + \varepsilon_i. \tag{4.1}$$

Incluímos esta função na lista de funções para teste de identificação pelo MW, nomeando-a como $f_{26}$. Calculamos também o tempo de processamento do algoritmo de identificação, a fim de registrá-lo para comparações em trabalhos futuros.

Realizando uma análise gráfica por meio da figura 4.6, percebe-se que a função $f_{26}$ realmente se aproxima dos valores da base de dados reais. É possível visualizar também que o MW consegue captar a curva dos dados e delineia curva semelhante à função tomada como verdadeira.

Figura 4.6: Relação empírica entre as variáveis $X$ e $Y$ da base de dados *"coelhos europeus"* e dos valores estimados pelo MW e pela função $f_{26}$.




Quando realizamos a comparação das funções por meio das medidas RMQE e EAM, utilizando o MW como ferramenta de identificação, tem-se que a $f_{26}$ apresentou a menor RMQE (0.25) e o menor EAM (0.12), como apresentado na Tabela 4.7. Isto significa que o MW identificou a função adequada aos dados, seja por RMQE ou EAM. O processamento do algoritmo durou 5.37 segundos.

Tabela 4.7: Resultados das Medidas Comparativas. Função $f_{26}$.

| função | RMQE | EAM | função | RMQE | EAM |
|---|---|---|---|---|---|
| $f_1$ | 0.48 | 0.44 | $f_{14}$ | 4.03 | 4.17 |
| $f_2$ | 0.48 | 0.44 | $f_{15}$ | 4.89 | 5.17 |
| $f_3$ | 1.16 | 0.76 | $f_{16}$ | 7.40 | 2.74 |
| $f_4$ | 5.01 | 5.09 | $f_{17}$ | 0.39 | 0.37 |
| $f_5$ | 5.03 | 5.17 | $f_{18}$ | 3.04 | 3.25 |
| $f_6$ | 2.84 | 2.62 | $f_{19}$ | 5.03 | 5.17 |
| $f_7$ | 2.81 | 2.66 | $f_{20}$ | 0.76 | 0.57 |
| $f_8$ | 4.03 | 4.17 | $f_{21}$ | 5.00 | 5.16 |
| $f_9$ | 4.03 | 4.17 | $f_{22}$ | 5.03 | 5.17 |
| $f_{10}$ | 4.03 | 4.17 | $f_{23}$ | 5.03 | 5.17 |
| $f_{11}$ | 4.03 | 4.17 | $f_{24}$ | 0.32 | 0.22 |
| $f_{12}$ | 4.03 | 4.17 | $f_{25}$ | 5.03 | 5.17 |
| $f_{13}$ | 4.03 | 4.17 | $f_{26}$ | **0.25** | **0.12** |

É importante ressaltar que o tamanho amostral da base de dados reais que utilizamos é menor que os tomados na parte das simulações, nas quais os menores deles foram iguais a 128.

# Capítulo 5

# Considerações Finais

A regressão não linear representa uma importante técnica Estatística, principalmente por sua frequente aplicabilidade prática em outras áreas de conhecimento. Entretanto, a escolha de uma função não linear que melhor represente os dados não é uma tarefa fácil quando a verdadeira relação matemática entre a variável resposta e as independentes é desconhecida.

Baseado nesta problemática, avaliamos o desempenho de um modelo de regressão wavelet (MW) de modo a identificar a função não linear mais adequada para representar uma relação linear entre $Y$ e $X$.

Um estudo de simulação foi considerado levando em conta diferentes funções não lineares, tamanhos de amostra, bem como o grau de associação entre as variáveis $Y$ e $X$, num total de 36 cenários distintos. Para cada cenário foram simuladas 1000 réplicas de Monte Carlo, a fim de avaliar a taxa de acerto do MW na identificação da função verdadeira, considerando duas medidas de performance.

Verificamos que, quanto maior o tamanho da amostra e/ou quando o grau da relação não linear entre $Y$ e $Y$ é forte, o MW mostrou-se eficiente na detecção da função não linear geradora dos dados amostrais. Este resultado é análogo para os cenários de relação moderada. Entretanto, quando o grau da relação não linear é apenas leve, o MW apresentou seu pior desempenho, principalmente em pequenas amostras ($n = 128$).

Com relação às medidas comparativas, o Erro Absoluto Mediano (EAM) não apresentou as melhores taxa de classificação, quando confrontado com a Raiz Média do Quadrado dos Erros (RMQE).

Ao comparar funções não lineares similares, o MW apresentou uma performance satisfatória de modo a sinalizar corretamente a verdadeira função geradora dos dados, mesmo para pequenos tamanhos de amostra.

Quando aplicado a uma base de dados real o MW identificou a função adequada definida pela literatura, tanto pelo EAM quanto pela RMQE.

No geral, tem-se evidências de que o Modelo de Regressão Wavelet obteve ótimo desempenho na detecção de funções não lineares. Portanto, com base nos resultados aqui

apresentados consideramos esta abordagem uma ferramenta importante para identificar a verdadeira função não linear, quando a mesma não é conhecida.

# Capítulo 6

# Referências Bibliográficas


BATES, D.M; WATTS, D.G.; **Nonlinear Regression Analysis and Its Applications.** Wiley series probability and Statistics, New York, 2007.

COLBY E.; BAIR E.; **Cross-validation for nonlinear mixed effects models.** J Pharmacokinet Pharmacodyn 40:243–252, 2013.

DAUBECHIES, I.; **Ten Lectures on Wavelets.** SIAM, CBMS-NSF Conference Series, 1992.

DRAPER N.R.; SMITH H.; **Applied regression analysis**. Wiley, New York, 1981.

DUDZINSKI, M.; MYKYTOWYCZ, R. **The eye lens as an indicator of age in the wild rabbit in australia.** Wildlife Research, CSIRO, v. 6, n. 2, p. 156–159, 1961.

ECKHARDT, R.; **Stan Ulam, John Von Neumann,and the Monte Carlo Method.** Los Alamos Science Special Issue, 131-136, 1987.

FLETCHER, R.; REEVES, C.M.; **Function minimizaztion by conjugate gradients.** 1964.

GALEA, M.; PAULA, G. A.; CYSNEIROS, F. J. A. **On diagnostics in symmetrical nonlinear models.** Statistics & Probability Letters, v. 73, n. 4, p. 459 – 467, 2005.

GALLANT, A.R.; **Nonlinear Statistical Models.** North Carolina, USA, a John Wiley & sons, inc. publication, 1982.

HESTENES M.R.; STIEFEL E.; **Methods of Conjugate Gradients for Solving Linear Systems**.Journal of Research of the National Bureau of Standards Vol. 49, No.6 pg 409-436, December 1952.

KOVAC, A.; R: **Irregular wavelet transform (decomposition)**. Code Copyright Arne Kovac 1997. Disponível em: https://artax.karlin.mff.cuni.cz/r-help/library/wavethresh/html/irregwd.html. Acesso em 2017.

KOVAC, A.; SILVERMAN, B.W.; **Extending the Scope of Wavelet Regression Methods by Coefficient-dependent Thresholding.** Journal of the American Statistical Associa-tion, 95, 172–183, 2000.

MALLAT, S.; **A theory for multi-resolution signal decomposition: the wavelet representation.** IEEE Transactions on Pattern Analysis and Machine Intelligence, 11 (7), 674–693, 1989.

MICHAELI L.; MENTEN M.; **Die kinetik der invertinwirkung**. Biochen Zeitung 49:333–369, 1913.

MONTGOMERY D.C.; PECK E.A.; **Introduction to linear regression analysis**. Wiley, New York, 1982.

MORETTIN, P.A.; PINHEIRO; A. and VIDAKOVIC, B.; **Wavelets in Functional Data Analysis.** Springer, New York, in press, 2016.

NOCEDAL, J.; STEPHEN W. **Numerical optimization.** Springer Science & Business Media, 2006.

RATKOWSKY, D. A. **Nonlinear regression modeling.** Dekker, New York., 1983.

RITZ, C., STREIBIG, J.C.; **Nonlinear Regression with R.** Springer, NY USA, 2008.

RUCKSTUHL, A.; **Introduction to Nonlinear Regression.** ZHAW -Zürcher Hochschule für Angewandte Wissenschaften, Outubro de 2010.

RYAN, T.P.; **Modern Regression methods** - 2nd ed. Acworth, Georgia, a John Wiley & sons, inc. publication, 489-511, 2009.

RYAN, T.P.; **Solutions Manual to Accompany. Modern Regression methods** - 2nd ed. Acworth, Georgia, a John Wiley & sons, inc. publication, 111-117, 2009.

SEBER G.A.F.; WILD C.J.; **Nonlinear regression**. Wiley, New York, 2003.

SHAO J.; **Linear model selection by cross-validation.** J Am Stat Assoc 88:486–495, 1993.

VIDAKOVIC, B.; **Statistical Modeling by Wavelets.** John Wiley & Sons, 1999.

# Apêndice A - Funções Não Lineares

$$f_1(x,\beta) = \frac{\beta_1}{\beta_2 + e^{\beta_3 x}}$$

$$f_2(x,\beta) = \beta_1 + e^{-\beta_2 x} = \beta_1 - \frac{1}{e_2^{\beta} x}$$

$$f_3(x,\beta) = \frac{\beta_2 x}{\beta_1 + x}$$

$$f_4(x,\beta) = \beta_1 \cos(2x) + \beta_2 \sin(x)$$

$$f_5(x,\beta) = \beta_1 - \frac{\beta_2}{\beta_3 + x}$$

$$f_6(x,\beta) = \beta_1 e^{\frac{-x}{\beta_2}}$$

$$f_7(x,\beta) = \frac{\beta_1}{1 + \frac{x}{\beta_2}}$$

$$f_8(x,\beta) = \beta_3 + \frac{1 - \beta_3}{1 + e^{-\beta_1 - \beta_2 x}}$$

$$f_9(x,\beta) = \frac{1}{1 + e^{-\beta_1 - \beta_2 x}}$$

$$f_{10}(x,\beta) = 1 - e^{-\beta_1 x}$$

$$f_{11}(x,\beta) = \beta_1 + (1 - \beta_1)(1 - e^{-\beta_2 x - \beta_3 (x^2)})$$

$$f_{12}(x,\beta) = 1 - e^{-\beta_1 x - \beta_2 (x^2)}$$

$$f_{13}(x,\beta) = 1 - e^{-\beta_1 x - \beta_2 (x^2) - \beta_3 (x^3) - \beta_4 (x^4)}$$

$$f_{14}(x,\beta) = \beta_1 e^{\beta_2 x}$$

$$f_{15}(x,\beta) = \beta_1 (1 - e^{-\beta_2 x})$$

$$f_{16}(x,\beta) = \beta_1 (\beta_2 - e^{-\beta_3 x})$$

$$f_{17}(x,\beta) = \frac{\beta_1 x}{\beta_2 + x}$$

$$f_{18}(x,\beta) = \beta_1 e^{-0.5 \frac{x - \beta_2}{\beta_3}}$$

$$f_{19}(x,\beta) = \frac{\beta_1 + \beta_2 x}{1 + \beta_3 x + \beta_4 x^2}$$

$$f_{20}(x,\beta) = \beta_1 \cos(x + \beta_4) + \beta_2 \cos(2x + \beta_4) + \\ + \beta_3 \cos(3x + \beta_4)$$

$$f_{21}(x,\beta) = \beta_1 (x - \beta_2)^{\beta_3}$$

$$f_{22}(x,\beta) = \beta_1 e^{-e^{\beta_2 - \beta_3 x}}$$

$$f_{23}(x,\beta) = \frac{\beta_1}{1 + e^{\beta_2 - \beta_3 x}}$$

$$f_{24}(x,\beta) = \frac{1}{\beta_1 + \beta_2 x}$$

$$f_{25}(x,\beta) = \frac{1}{\beta_1 + \beta_2 x + \beta_3 x^2}$$

# Apêndice B - Script para uso em software $R$

```
rm(list = ls())
library(wavethresh)
library(xtable)
######################## FUNCOES NAO-LINEARES ##########################

fo.1 <- function(beta,Y,X) {
a = beta[1]
b = beta[2]
c = beta[3]
f.obj.m = sum((Y - (a/(b+exp(c*X))))^2)
f.obj.m
}

fo.2 <- function(beta,Y,X) {
m = beta[1]
k = beta[2]
f.obj.r = sum((Y - (m + exp(-k*X)))^2)
f.obj.r
}

fo.3 <- function(beta,Y,X) {
a = beta[1]
b = beta[2]
f.obj.m = sum((Y- ((b*X)/(a+X)))^2)
f.obj.m
}

fo.4 <- function(beta,Y,X) { #Wavy
a = beta[1]
b = beta[2]
f.obj.r = sum((Y - (a*cos(2*X) + b*sin(X)))^2)
f.obj.r
}

fo.5 <- function(beta,Y,X) {
m = beta[1]
k = beta[2]
```

```
v = beta[3]
f.obj.r = sum((Y - ((m - (k/(v+X)))))^2)
f.obj.r
}

fo.6 <- function(beta,Y,X) {#ExpDecline
q = beta[1]
a = beta[2]
f.obj.r = sum((Y - (q*exp(-X/a)))^2)
f.obj.r
}

fo.7 <- function(beta,Y,X) {#HarmDecline
q = beta[1]
a = beta[2]
f.obj.r = sum((Y - (q/(1+(X/a))))^2)
f.obj.r
}

fo.8 <- function(beta,Y,X) {#DR-Logistic2
a = beta[1]
b = beta[2]
g = beta[3]
f.obj.r = sum((Y - (g + ((1-g)/(1 + exp(-a-b*X)))))^2)
f.obj.r
}

fo.9 <- function(beta,Y,X) {#DR-Logistic2Zero
a = beta[1]
b = beta[2]
f.obj.r = sum((Y - (((1)/(1 + exp(-a-b*X)))))^2)
f.obj.r
}

fo.10 <- function(beta,Y,X) {#MultiStageZero1
b = beta[1]
f.obj.r = sum((Y - ((1-exp(-b*X))))^2)
f.obj.r
}

fo.11 <- function(beta,Y,X) {#MultiStage2
a = beta[1]
b1 = beta[2]
b2 = beta[3]
f.obj.r = sum((Y - (a + (1-a)*(1-exp(-b1*X - b2*(X^2)))))^2)
f.obj.r
}
```

```
fo.12 <- function(beta,Y,X) {#MultiStage2Zero
b1 = beta[1]
b2 = beta[2]
f.obj.r = sum((Y - ((1-exp(-b1*X - b2*(X^2)))))^2)
f.obj.r
}

fo.13 <- function(beta,Y,X) {#MultiStageZero4
b1 = beta[1]
b2 = beta[2]
b3 = beta[3]
b4 = beta[4]
f.obj.r = sum((Y - ((1-exp(-b1*X - b2*(X^2) - b3*(X^3) - b4*(X^4) ))))^2)
f.obj.r
}

fo.14 <- function(beta,Y,X) {#Exponential
a = beta[1]
b = beta[2]
f.obj.r = sum((Y - (a*exp(b*X)))^2)
f.obj.r
}

fo.15 <- function(beta,Y,X) {#ExpAssoc2
a = beta[1]
b = beta[2]
f.obj.r = sum((Y - (a*(1 - exp(-b*X))))^2)
f.obj.r
}

fo.16 <- function(beta,Y,X) {#ExpAssoc3
a = beta[1]
b = beta[2]
c = beta[3]
f.obj.r = sum((Y - (a*(b - exp(-c*X))))^2)
f.obj.r
}

fo.17 <- function(beta,Y,X) {#SaturGrowth
a = beta[1]
b = beta[2]
f.obj.r = sum((Y - (a*X/(b + X)))^2)
f.obj.r
}

fo.18 <- function(beta,Y,X) {#GaussModel
a = beta[1]
```

```
b = beta[2]
c = beta[3]
f.obj.r = sum((Y - (a*exp(-0.5*((X-b)/c)^2)))^2)
f.obj.r
}

fo.19 <- function(beta,Y,X) {#RationalModel
a = beta[1]
b = beta[2]
c = beta[3]
d = beta[4]
f.obj.r = sum((Y - ((a + b*X)/(1 + c*X + d*X^2)))^2)
f.obj.r
}

fo.20 <- function(beta,Y,X) {#TruncFourier
a = beta[1]
b = beta[2]
c = beta[3]
d = beta[4]
f.obj.r = sum((Y - (a*cos(X+d) + b*cos(2*X+d) + c*cos(3*X+d)))^2)
f.obj.r
}

fo.21 <- function(beta,Y,X) {#ShiftPower
a = beta[1]
b = beta[2]
c = beta[3]
f.obj.r = sum((Y - (a*(X - b)^(c)))^2)
f.obj.r
}

fo.22 <- function(beta,Y,X) {#Gompertz
a = beta[1]
b = beta[2]
c = beta[3]
f.obj.r = sum((Y - (a*exp(-exp(b-c*X))))^2)
f.obj.r
}

fo.23 <- function(beta,Y,X) {#Ratkowsky
a = beta[1]
b = beta[2]
c = beta[3]
f.obj.r = sum((Y - (a/(1 + exp(b - c*X))))^2)
f.obj.r
}
```

```
fo.24 <- function(beta,Y,X) {#Reciprocal
a = beta[1]
b = beta[2]
f.obj.r = sum((Y - (1/(a + b*X)))^2)
f.obj.r
}

fo.25 <- function(beta,Y,X) {#ReciprocalQuadrad
a = beta[1]
b = beta[2]
c = beta[3]
f.obj.r = sum((Y - (1/(a + b*X + c*(X^2))))^2)
f.obj.r
}


medianaErro=function(McycleWR.rx,Yest){ median(abs(McycleWR.rx-Yest)) }
RErroQM=function(McycleWR.rx,Yest){ sqrt(mean((McycleWR.rx-Yest)^2)) }

#################################################################
#=========================SIMULACAO=============================#
#################################################################

m=1000  #numero de simulacoes (Monte Carlo)

N=c(128,256,512) #tamanhos de amostras
namostra=length(N) #quantidade dos tamanhos de amostra testados
nfuncao=4 #numero de funcoes verdadeiras
fteste=25 #numero de funcoes a serem testadas

RMSE=matrix(0,nfuncao,namostra)
MAE=matrix(0,nfuncao,namostra)

for(amostra in 1:namostra){

n=N[amostra]
Erro.Mediana=array(0,c(m,fteste,nfuncao))
Erro.RQM=array(0,c(m,fteste,nfuncao))
Comp.EM=array(0,c(m,fteste,nfuncao))
Comp.RMSE=array(0,c(m,fteste,nfuncao))

###################### MODELO VERDADEIRO ########################

for(funcao in 1:nfuncao){

if(funcao == 1){
variancia=0.01 # Forte:0.01  ; Moderada:0.1  ; Leve:0.2
Erro=matrix(rnorm(N[amostra],0,variancia),nrow=N[amostra],ncol=1)
```

```
num_var=1
X_inf = -6; X_sup = 6
X=matrix(runif((N[amostra]*num_var),X_inf,X_sup),nrow=N[amostra],
ncol=num_var)
a = 2; b = 3; c = 1
Y =a/(b+exp(c*X)) + Erro
Y_trein=Y
X_trein=X
#plot(X,Y)
}

if(funcao == 2){
variancia=0.06 # Forte:0.005  ; Moderada:0.03  ; Leve:0.06
Erro=matrix(rnorm(N[amostra],0,variancia),nrow=N[amostra],ncol=1)
num_var=1
X_inf = 1; X_sup = 4
X=matrix(runif((N[amostra]*num_var),X_inf,X_sup),nrow=N[amostra],
ncol=num_var)
mf=0.25 ;k=1
Y= mf + exp(-k*X) + Erro
Y_trein=Y
X_trein=X
#plot(X,Y)
}

if(funcao == 3){
variancia=1 # Forte:1  ; Moderada:5  ; Leve:10
# inserir 3 ni?veis de variancia
Erro=matrix(rnorm(N[amostra],0,variancia),nrow=N[amostra],ncol=1)
a=20; b=120
num_var=1
X_inf = 5; X_sup = 210
X=matrix(runif((N[amostra]*num_var),X_inf,X_sup),nrow=N[amostra],
ncol=num_var)
Y=(b*X)/(a+X) + Erro
Y_trein=Y
X_trein=X
#plot(X,Y)
}

if(funcao==4){
variancia=0.1 # Forte:0.1  ; Moderada:1.0  ; Leve:2.0
Erro=matrix(rnorm(N[amostra],0,variancia),nrow=N[amostra],ncol=1)
num_var=1
X_inf = 0; X_sup = 4
X=matrix(runif((N[amostra]*num_var),X_inf,X_sup),nrow=N[amostra],
ncol=num_var)
a=4; b=1
```

```
Y=a*cos(2*X) + b*sin(X) + Erro
Y_trein=Y
X_trein=X
#plot(X,Y)
}

l=0
for(mc in 1:m){
l=l+1
print(l)
###################### AJUSTES DOS MODELOS ########################

Zbeta1 = c(runif(1,1.5,2.5),runif(1,2.5,3.5),runif(1,0.5,1.5))
mod1 = optim(Zbeta1,fo.1,Y=Y,X=X, method="CG")
Beta1 = c(c(mod1$par))
Yest.1 <- (Beta1[1]/(Beta1[2] + exp(Beta1[3]*X)))
#########################################################################
Zbeta2 = c(runif(1,0.0,0.5),runif(1,0.5,1.5))
mod2 = optim(Zbeta2,fo.2,Y=Y,X=X, method="CG")
Beta2 = c(c(mod2$par))
Yest.2 <- (Beta2[1] + exp(-Beta2[2]*X))
#########################################################################
Zbeta3 = c(runif(1,19.8,20.2),runif(1,119.5,120.5))
mod3 = optim(Zbeta3,fo.3,Y=Y,X=X, method="CG")
Beta3 = c(c(mod3$par))
Yest.3 <- (Beta3[2]*X)/(X+Beta3[1])
#########################################################################
Zbeta4 = c(runif(1,3,5),runif(1,0.5,1.5)) #Wavy
mod4 = optim(Zbeta4,fo.4,Y=Y,X=X, method="CG")
Beta4 = c(c(mod4$par))
Yest.4 <- (Beta4[1]*cos(2*X) + Beta4[2]*sin(X))
#########################################################################
Zbeta5 = c(runif(1,18.8,19.2),runif(1,499,501),runif(1,-16.2,-15.8))
mod5 = optim(Zbeta5,fo.5,Y=Y,X=X, method="CG")
Beta5 = c(c(mod5$par))
Yest.5 <-(Beta5[1] - (Beta5[2]/(Beta5[3] + X)))
#########################################################################
Zbeta6 = c(runif(1,20,21),runif(1,-10.5,-9.5)) #ExpDecline
mod6 = optim(Zbeta6,fo.6,Y=Y,X=X, method="CG")
Beta6 = c(c(mod6$par))
Yest.6 <- (Beta6[1]*exp(-X/Beta6[2]))
#########################################################################
Zbeta7 = c(runif(1,-0.2,0.2),runif(1,1.3,1.7)) #HarmDecline
mod7 = optim(Zbeta7,fo.7,Y=Y,X=X, method="CG")
Beta7 = c(c(mod7$par))
Yest.7 <- (Beta7[1]/(1+(X/Beta7[2])))
#########################################################################
Zbeta8=c(runif(1,0.0,0.5),runif(1,0.9,1.1),runif(1,0.9,1.1))#DR-Logistic
```

```
mod8 = optim(Zbeta8,fo.8,Y=Y,X=X, method="CG")
Beta8 = c(c(mod8$par))
Yest.8 <- (Beta8[3] + ((1-Beta8[3])/(1 + exp(-Beta8[1]
-Beta8[2]*X_trein))))
#########################################################################
Zbeta9 = c(runif(1,0.9,1.1),runif(1,0.9,1.1)) #DR-LogisticZero
mod9 = optim(Zbeta9,fo.9,Y=Y_trein,X=X_trein, method="CG")
Beta9 = c(c(mod9$par))
Yest.9 <- ((1/(1 + exp(-Beta9[1]-Beta9[2]*X_trein))))
#########################################################################
Zbeta10 = c(runif(1,0.9,1.1)) #DR-Multi1Zero
mod10 = optim(Zbeta10,fo.10,Y=Y_trein,X=X_trein, method="CG")
Beta10 = c(c(mod10$par))
Yest.10 <- ((1-exp(-Beta10[1]*X_trein)))
#########################################################################
Zbeta11=c(runif(1,0.0,0.5),runif(1,0.9,1.1),runif(1,0.9,1.1))#DR-Multi2
mod11 = optim(Zbeta11,fo.11,Y=Y_trein,X=X_trein, method="CG")
Beta11 = c(c(mod11$par))
Yest.11 <- (Beta11[1] + (1-Beta11[1])*(1-exp(-Beta11[2]*X_trein
- Beta11[3]*(X_trein^2))))
#########################################################################
Zbeta12 = c(runif(1,0.9,1.1),runif(1,0.9,1.1)) #DR-Multi2Zero
mod12 = optim(Zbeta12,fo.12,Y=Y_trein,X=X_trein, method="CG")
Beta12 = c(c(mod12$par))
Yest.12 <- ((1-exp(-Beta12[1]*X_trein- Beta12[2]*(X_trein^2))))
#########################################################################
Zbeta13 = c(runif(1,0.9,1.1),runif(1,0.9,1.1),runif(1,0.9,1.1),
runif(1,0.9,1.1))
mod13=optim(Zbeta13,fo.13,Y=Y_trein,X=X_trein, method="CG")#Multi4Zero
Beta13 = c(c(mod13$par))
Yest.13 <- ((1-exp(-Beta13[1]*X_trein- Beta13[2]*(X_trein^2)
 - Beta13[3]*(X_trein^3) - Beta13[4]*(X_trein^4))))
#########################################################################
Zbeta14 = c(runif(1,20.7,21.3),runif(1,0.0,0.01)) #Exponential
mod14 = optim(Zbeta14,fo.14,Y=Y_trein,X=X_trein, method="CG")
Beta14 = c(c(mod14$par))
Yest.14 <- (Beta14[1]*exp(Beta14[2]*X_trein))
#########################################################################
Zbeta15 = c(runif(1,373,374),runif(1,0.0,0.01)) #ExpoAssoc2
mod15 = optim(Zbeta15,fo.15,Y=Y_trein,X=X_trein, method="CG")
Beta15 = c(c(mod15$par))
Yest.15 <- (Beta15[1]*(1 - exp(-Beta15[2]*X_trein)))
#########################################################################
Zbeta16=c(runif(1,69,70),runif(1,0.9,1.1),runif(1,0.15,0.25))#ExpoAssoc3
mod16 = optim(Zbeta16,fo.16,Y=Y_trein,X=X_trein, method="CG")
Beta16 = c(c(mod16$par))
Yest.16 <- (Beta16[1]*(Beta16[2] - exp(-Beta16[3]*X_trein)))
#########################################################################
```

```
Zbeta17 = c(runif(1,-278,-277),runif(1,-854,-853)) #SaturationGrowth
mod17 = optim(Zbeta17,fo.17,Y=Y_trein,X=X_trein, method="CG")
Beta17 = c(c(mod17$par))
Yest.17 <- (Beta17[1]*X_trein/(Beta17[2] + X_trein))
#########################################################################
Zbeta18=c(runif(1,0.0,0.5),runif(1,0.5,1.5),runif(1,0.5,1.5))#GaussMOdel
mod18 = optim(Zbeta18,fo.18,Y=Y_trein,X=X_trein, method="CG")
Beta18 = c(c(mod18$par))
Yest.18 <- (Beta18[1]*exp(-0.5*((X_trein-Beta18[2])/Beta18[3])^2))
#########################################################################
Zbeta19 = c(runif(1,32,33),runif(1,-0.2,-0.1),runif(1,-0.01,0.0),
runif(1,0.0,0.0001))
mod19 = optim(Zbeta19,fo.19,Y=Y_trein,X=X_trein, method="CG")
#RationalModel
Beta19 = c(c(mod19$par))
Yest.19 <- ((Beta19[1] + Beta19[2]*X_trein)/(1 + Beta19[3]*X_trein
+ Beta19[4]*X_trein^2))
#########################################################################
Zbeta20 = c(runif(1,6,7),runif(1,70,71),runif(1,13,14),runif(1,1,2))
mod20=optim(Zbeta20,fo.20,Y=Y_trein,X=X_trein, method="CG")#TruncFourier
Beta20 = c(c(mod20$par))
Yest.20 <- (Beta20[1]*cos(X_trein+Beta20[4]) +
Beta20[2]*cos(2*X_trein+Beta20[4]) + Beta20[3]*cos(3*X_trein+Beta20[4]))
#########################################################################
Zbeta21 = c(runif(1,0.0,0.1),runif(1,-13,-12),runif(1,1,2)) #ShiftPower
mod21 = optim(Zbeta21,fo.21,Y=Y_trein,X=X_trein, method="CG")
Beta21 = c(c(mod21$par))
Yest.21 <- (Beta21[1]*(X_trein- Beta21[2])^(Beta21[3]))
#########################################################################
Zbeta22 = c(runif(1,263,264),runif(1,1,2),runif(1,0.0,0.01))  #Gompertz
mod22 = optim(Zbeta22,fo.22,Y=Y_trein,X=X_trein, method="CG")
Beta22 = c(c(mod22$par))
Yest.22 <- (Beta22[1]*exp(-exp(Beta22[2]-Beta22[3]*X_trein)))
#########################################################################
Zbeta23 = c(runif(1,243,244),runif(1,2,3),runif(1,0.0,0.1)) #Ratkowsky
mod23 = optim(Zbeta23,fo.23,Y=Y_trein,X=X_trein, method="CG")
Beta23 = c(c(mod23$par))
Yest.23 <- (Beta23[1]/(1 + exp(Beta23[2] - Beta23[3]*X_trein)))
#########################################################################
Zbeta24 = c(runif(1,0.0,0.1),runif(1,-0.001,0.0)) #Reciprocal
mod24 = optim(Zbeta24,fo.24,Y=Y_trein,X=X_trein, method="CG")
Beta24 = c(c(mod24$par))
Yest.24 <- (1/(Beta24[1] + Beta24[2]*X_trein))
#########################################################################
Zbeta25 = c(runif(1,-0.25,-0.2),runif(1,0.0,0.01),runif(1,-0.001,0.0))
#ReciprocalQuaradr
mod25 = optim(Zbeta25,fo.25,Y=Y_trein,X=X_trein, method="CG")
Beta25 = c(c(mod25$par))
```

```
Yest.25 <- (1/(Beta25[1] + Beta25[2]*X_trein + Beta25[3]*(X_trein^2)))
#########################################################################


###################### AJUSTE DA WAVELET ########################

# Passos para o Ajuste da Regressão Wavelet
# Re-escalando X
x01 <- (X - min(X))/(max(X) - min(X))
McycleGrid <- makegrid(t=x01, y=Y) # Grid - equidistantes
#TimeGrid<-McycleGrid$gridt*(max(X)-min(X))+min(X)

#Ajustando o Modelo Wavelet
McycleIRRWD <- irregwd(McycleGrid)

#Aplicando o limiar (threshold)
McycleT <- threshold(McycleIRRWD, policy="universal", type="hard",
 dev=madmad)
McycleWR <- wr(McycleT) # Valores estimados pelo MW
rx = rank(X)

McycleWR.rx = McycleWR[rx] # Organizando os valores ajustados
do MW na ordem de Y

Erro.Mediana[mc,,funcao]=c(medianaErro(McycleWR.rx,Yest.1),
medianaErro(McycleWR.rx,Yest.2),medianaErro(McycleWR.rx,Yest.3),
medianaErro(McycleWR.rx,Yest.4),medianaErro(McycleWR.rx,Yest.5),
medianaErro(McycleWR.rx,Yest.6),medianaErro(McycleWR.rx,Yest.7),
medianaErro(McycleWR.rx,Yest.8),medianaErro(McycleWR.rx,Yest.9),
medianaErro(McycleWR.rx,Yest.10),medianaErro(McycleWR.rx,Yest.11),
medianaErro(McycleWR.rx,Yest.12),medianaErro(McycleWR.rx,Yest.13),
medianaErro(McycleWR.rx,Yest.14),medianaErro(McycleWR.rx,Yest.15),
medianaErro(McycleWR.rx,Yest.16),medianaErro(McycleWR.rx,Yest.17),
medianaErro(McycleWR.rx,Yest.18),medianaErro(McycleWR.rx,Yest.19),
medianaErro(McycleWR.rx,Yest.20),medianaErro(McycleWR.rx,Yest.21),
medianaErro(McycleWR.rx,Yest.22),medianaErro(McycleWR.rx,Yest.23),
medianaErro(McycleWR.rx,Yest.24),medianaErro(McycleWR.rx,Yest.25))

Erro.RQM[mc,,funcao]=c(RErroQM(McycleWR.rx,Yest.1),RErroQM(McycleWR.rx,
Yest.2),RErroQM(McycleWR.rx,Yest.3),RErroQM(McycleWR.rx,Yest.4),
RErroQM(McycleWR.rx,Yest.5),RErroQM(McycleWR.rx,Yest.6),
RErroQM(McycleWR.rx,Yest.7),RErroQM(McycleWR.rx,Yest.8),
RErroQM(McycleWR.rx,Yest.9),RErroQM(McycleWR.rx,Yest.10),
RErroQM(McycleWR.rx,Yest.11),RErroQM(McycleWR.rx,Yest.12),
RErroQM(McycleWR.rx,Yest.13),RErroQM(McycleWR.rx,Yest.14),
RErroQM(McycleWR.rx,Yest.15),RErroQM(McycleWR.rx,Yest.16),
RErroQM(McycleWR.rx,Yest.17),RErroQM(McycleWR.rx,Yest.18),
RErroQM(McycleWR.rx,Yest.19),RErroQM(McycleWR.rx,Yest.20),
```

```
RErroQM(McycleWR.rx,Yest.21),RErroQM(McycleWR.rx,Yest.22),
RErroQM(McycleWR.rx,Yest.23),RErroQM(McycleWR.rx,Yest.24),
RErroQM(McycleWR.rx,Yest.25))

}

minimos.EM = apply(Erro.Mediana[,,funcao],1,min)
minimos.RMSE = apply(Erro.RQM[,,funcao],1,min)

for(mc in 1:m){
Comp.EM[mc,,funcao]=(minimos.EM[mc]==Erro.Mediana[mc,,funcao])
Comp.RMSE[mc,,funcao]=(minimos.RMSE[mc]==Erro.RQM[mc,,funcao])
}

RMSE[funcao,amostra]=sum(Comp.RMSE[,funcao,funcao])
MAE[funcao,amostra]=sum(Comp.EM[,funcao,funcao])

}
}

RMSEp.MAEp=cbind((RMSE/m)*100,(MAE/m)*100)
rownames(RMSEp.MAEp)=1:nfuncao
colnames(RMSEp.MAEp)=c("RMSE/n=128","RMSE/n=256","RMSE/n=512",
"MAE/n=128","MAE/n=256","MAE/n=512")
xtable(RMSEp.MAEp,digits=2)

#FIGURAS

N=c(128,256,512) #tamanhos de amostras

###################### FUNCAO 1 ########################

amostra=1 # 1 , 2 ou 3
n=N[amostra]
variancia=0.01 # Forte:0.01  ; Moderada:0.1  ; Leve:0.2
Erro=matrix(rnorm(N[amostra],0,variancia),nrow=N[amostra],ncol=1)
num_var=1
X_inf = -6; X_sup = 6
X=matrix(runif((N[amostra]*num_var),X_inf,X_sup),nrow=N[amostra],
ncol=num_var)
a = 2; b = 3; c = 1
Y =a/(b+exp(c*X)) + Erro
Y_trein=Y
X_trein=X
x01 <- (X - min(X))/(max(X) - min(X))
McycleGrid <- makegrid(t=x01, y=Y)
McycleIRRWD <- irregwd(McycleGrid)
McycleT <- threshold(McycleIRRWD, policy="universal", type="hard",
```

```
 dev=madmad)
McycleWR <- wr(McycleT)
rx = rank(X)
McycleWR.rx = McycleWR[rx]
plot(X,Y,main="FORTE")  #FORTE , MODERADA ou LEVE
lines(X,McycleWR.rx,col="red",type="p")
legend("topright",legend=c("f.1","MW"),col=c("black","red"),lwd=2,bty="n")


###################### FUNCAO 2 ########################

amostra=1 # 1 , 2 ou 3
n=N[amostra]
variancia=0.005 # Forte:0.005  ; Moderada:0.03  ; Leve:0.06
Erro=matrix(rnorm(N[amostra],0,variancia),nrow=N[amostra],ncol=1)
num_var=1
X_inf = 1; X_sup = 4
X=matrix(runif((N[amostra]*num_var),X_inf,X_sup),nrow=N[amostra],
ncol=num_var)
mf=0.25 ;k=1
Y= mf + exp(-k*X) + Erro
Y_trein=Y
X_trein=X
x01 <- (X - min(X))/(max(X) - min(X))
McycleGrid <- makegrid(t=x01, y=Y)
McycleIRRWD <- irregwd(McycleGrid)
McycleT <- threshold(McycleIRRWD, policy="universal", type="hard",
 dev=madmad)
McycleWR <- wr(McycleT)
rx = rank(X)
McycleWR.rx = McycleWR[rx]
Zbeta24 = c(runif(1,0.0,0.1),runif(1,-0.001,0.0)) #Reciprocal
mod24 = optim(Zbeta24,fo.24,Y=Y_trein,X=X_trein, method="CG")
Beta24 = c(c(mod24$par))
Yest.24 <- (1/(Beta24[1] + Beta24[2]*X_trein))

plot(X,Y,main="FORTE")  #FORTE , MODERADA ou LEVE
lines(X,McycleWR.rx,col="red",type="p")
#lines(X,Yest.24,col="blue",type="p")
#legend("topright", legend=c("f.2","MW","f.24"),
 col=c("black","red","blue"), lwd=2, bty="n")
legend("topright",legend=c("f.2","MW"),col=c("black","red"),lwd=2,bty="n")

###################### FUNCAO 3 ########################

amostra=1 # 1 , 2 ou 3
n=N[amostra]
variancia=1 # Forte:1  ; Moderada:5  ; Leve:10
```

```
# inserir 3 ni?veis de variancia
Erro=matrix(rnorm(N[amostra],0,variancia),nrow=N[amostra],ncol=1)
a=20; b=120
num_var=1
X_inf = 5; X_sup = 210
X=matrix(runif((N[amostra]*num_var),X_inf,X_sup),nrow=N[amostra],
ncol=num_var)
Y=(b*X)/(a+X) + Erro
Y_trein=Y
X_trein=X
x01 <- (X - min(X))/(max(X) - min(X))
McycleGrid <- makegrid(t=x01, y=Y)
McycleIRRWD <- irregwd(McycleGrid)
McycleT <- threshold(McycleIRRWD, policy="universal", type="hard",
 dev=madmad)
McycleWR <- wr(McycleT)
rx = rank(X)
McycleWR.rx = McycleWR[rx]
plot(X,Y,main="FORTE") #FORTE , MODERADA ou LEVE
lines(X,McycleWR.rx,col="red",type="p")
legend("topright",legend=c("f.3","MW"),col=c("black","red"),lwd=2,bty="n")


###################### FUNCAO 4 ########################

amostra=1 # 1 , 2 ou 3
n=N[amostra]
variancia=0.1 # Forte:0.1  ; Moderada:1.0  ; Leve:2.0
Erro=matrix(rnorm(N[amostra],0,variancia),nrow=N[amostra],ncol=1)
num_var=1
X_inf = 0; X_sup = 4
X=matrix(runif((N[amostra]*num_var),X_inf,X_sup),nrow=N[amostra],
ncol=num_var)
a=4; b=1
Y=a*cos(2*X) + b*sin(X) + Erro
Y_trein=Y
X_trein=X
x01 <- (X - min(X))/(max(X) - min(X))
McycleGrid <- makegrid(t=x01, y=Y)
McycleIRRWD <- irregwd(McycleGrid)
McycleT <- threshold(McycleIRRWD, policy="universal", type="hard", dev=madmad)
McycleWR <- wr(McycleT)
rx = rank(X)
McycleWR.rx = McycleWR[rx]
plot(X,Y,main="FORTE") #FORTE , MODERADA ou LEVE
lines(X,McycleWR.rx,col="red",type="p")
legend("topright",legend=c("f.4","MW"),col=c("black","red"),lwd=2,bty="n")
```

```
### COMANDOS ADICIONAIS PARA APLICAÇÃO NA BASE DE DADOS "coelhos europeus" ###

fo.26 <- function(beta,Y,X) {
m = beta[1]
k = beta[2]
v = beta[3]
f.obj.r = sum((Y - ((m - (k/(v+X)))))^2)
f.obj.r
}

coelhos=read.table("lentes.txt",header=T,dec=",",sep="\t")

#Variaveis temporarias Treinamento
Y=Y_trein <- log(coelhos[,2])
X=X_trein <- coelhos[,1]

Zbeta26 = c(5,130,37)
mod26 = optim(Zbeta26,fo.26,Y=Y,X=X, method="CG")
Beta26 = c(c(mod26$par))
Yest.26 <-(Beta26[1] - (Beta26[2]/(Beta26[3] + X)))

plot(Y~X,xlab="Idade dos coelhos (em dias)",ylab="Peso das lentes dos olhos
(em mg)")
lines(Yest.26~X,type="p",col="blue")
lines(McycleWR.rx~X,type="p",col="red")
legend("bottomright",legend=c("f.26","MW"),col=c("blue","red"),lwd=2,bty="n")
```