

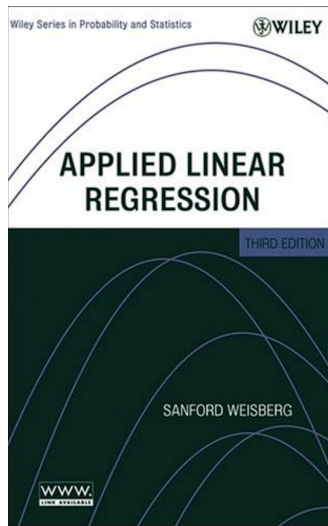
# Modelos Não Lineares no R

Filipe R. F. Teixeira

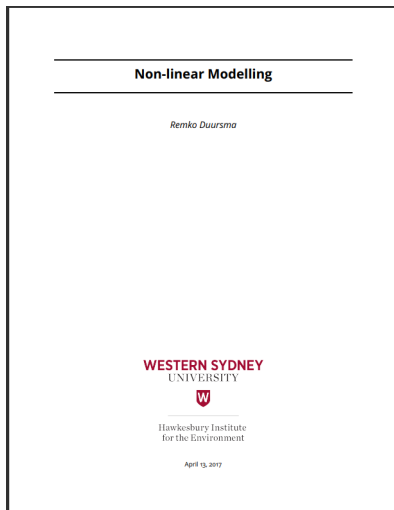
ConectaGem

25 de abril de 2023

# Sugestões de materiais



# Sugestões de materiais



## Sugestões de materiais

58° RBRAS e 15° SEAGRO

CURSO

## Modelos de regressão não linear

22 à 26 de julho de 2013

Walmes Marques Zeviani  
Paulo Justiniano Ribeiro Júnior  
Wagner Hugo Bonat

Laboratório de Estatística e Geoinformação  
Departamento de Estatística  
Universidade Federal do Paraná

Published February 19, 2014

Symposium: Statistical Concepts

## Nonlinear Regression Models and Applications in Agricultural Research

Sotirios V. Archontoulis and Fernando E. Miguez\*

### ABSTRACT

Nonlinear regression models are important tools because many crop and soil processes are better represented by nonlinear than linear models. Fitting nonlinear models is not a single-step procedure but an involved process that requires careful examination of each individual step. Depending on the objective and the application domain, different priorities are set when fitting nonlinear models; these include obtaining acceptable parameter estimates and a good model fit while meeting standard assumptions of statistical models. We propose steps in fitting nonlinear models as described by a flow diagram and discuss each step separately providing examples and updates on procedures used. The following steps are considered: (i) choose candidate models, (ii) set starting values, (iii) fit models, (iv) check convergence and parameter estimates, (v) find the “best” model among competing models, (vi) check model assumptions (residual analysis), and (vii) calculate statistical descriptors and confidence intervals. The associated feedback mechanisms are also addressed (i.e., model variance homogeneity). In particular, we emphasize the first step (choose candidate models) by providing an extensive library of nonlinear functions (77 equations with the associated parameter meanings) and examples of typical applications in agriculture. We hope that this contribution will clarify some of the difficulties and confusion with the task of using nonlinear models.

# Alguns pré-requisitos

- Conhecimento básico do software R;
- Pacotes *tidyverse*, *nlme* e *Metrics* instalados;
- Conhecimento básico de *ggplot* e *dplyr*;
- Conhecimento básico de Modelos Lineares (ML).

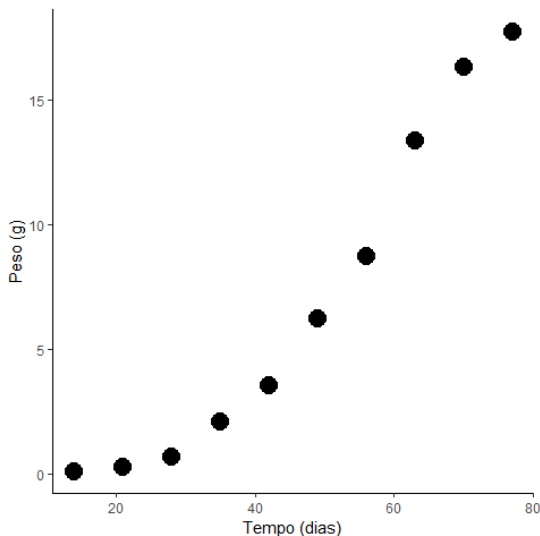
- 1 Introdução / Aspectos teóricos
  - Exemplos iniciais, vantagens e desvantagens
  - Prática 1: Análise gráfica
- 2 Estimação de Parâmetros
  - Método de Estimação (MMQ)
  - Prática 2: A função *nls*
  - Prática 3: Estabelecendo chutes iniciais
  - Prática 4: Ajustando novos modelos
  - Qualidade do ajuste
  - Prática 5: Comparando modelos
  - Prática 6: Ajustando vários elementos simultaneamente
- 3 Estudos de caso: outras aplicações no R

## Exemplo 1: Peso de folha de soja (*Soybean*)

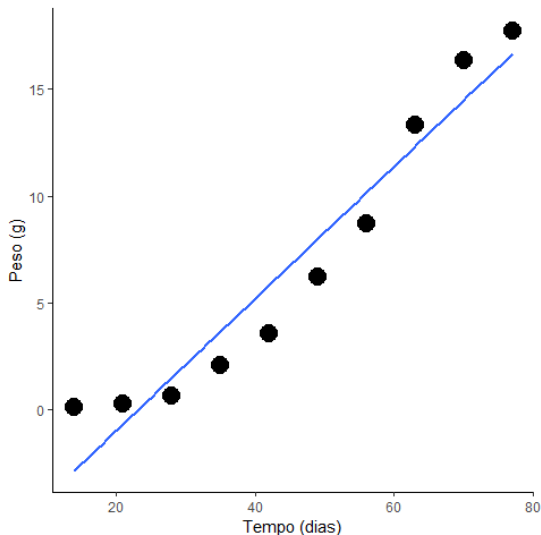
X: Tempo (dias)	Y: Peso (g)
14	0,106
21	0,261
28	0,666
35	2,110
42	3,560
49	6,230
56	8,710
63	13,350
70	16,342
77	17,751



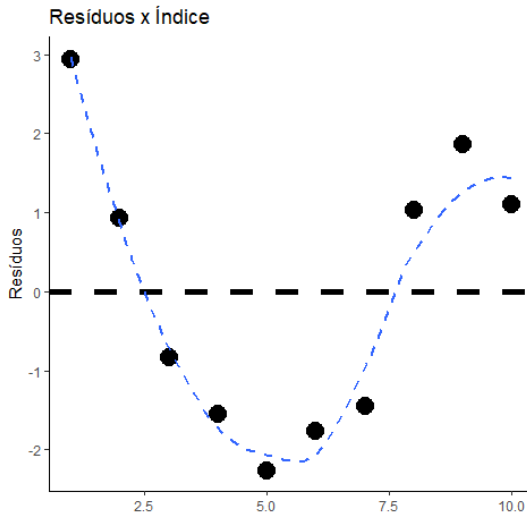
# Exemplo 1: Peso de folha de soja (*Soybean*)



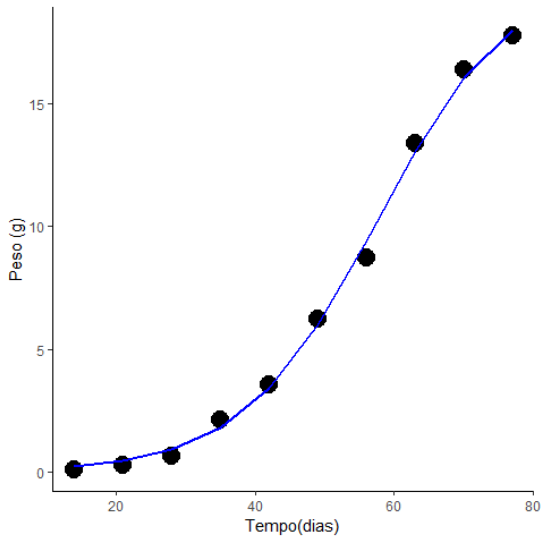
# Aplicando a Regressão Linear Simples (RLS)



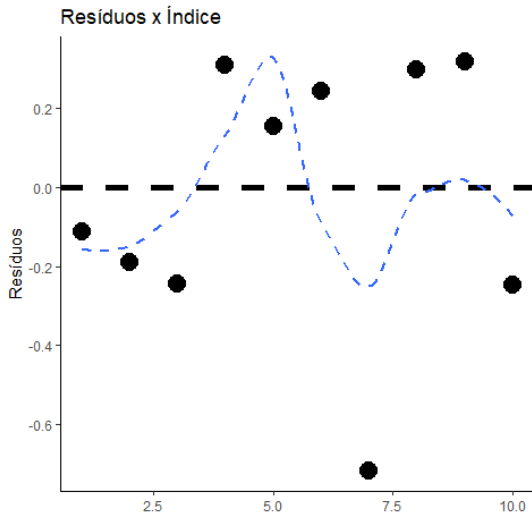
# Resíduos (RLS)



# Mesmo exemplo (MNL)



# Resíduos (MNL)



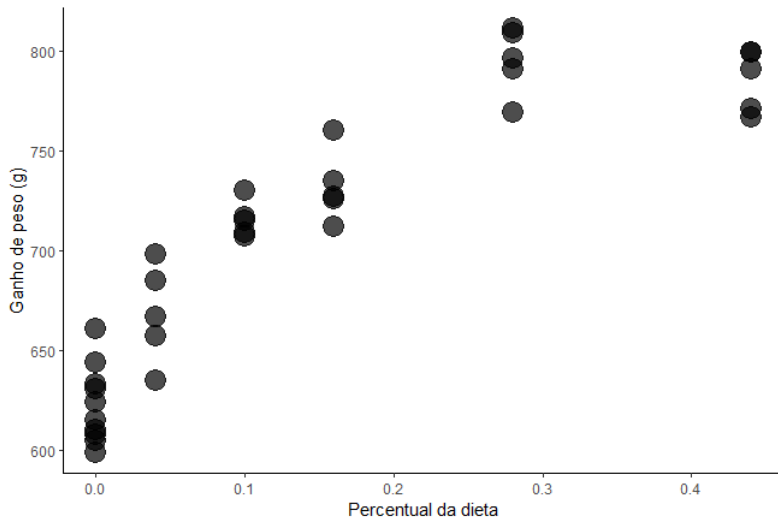
# EQM e EAM (RLS x MNL)

Modelo	EQM	EAM
RLS	2,87	1,57
MNL	0,11	0,28
Proporção	<b>27,10</b>	<b>5,53</b>

## Exemplo 2: Aminoácidos x Ganho de peso

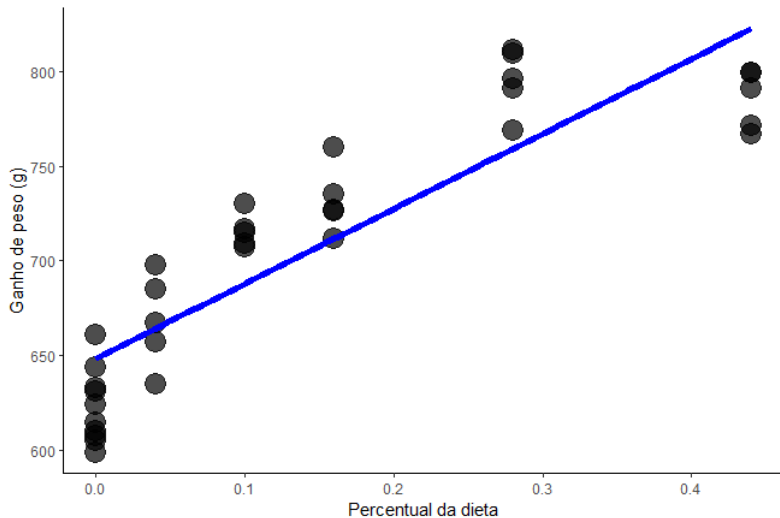
X: Percentual	Y: Ganho (g)
0.00	644, 631, 661, 624, 633, 610, 615, 605, 608, 599
0.04	698, 667, 657, 685, 635
0.10	730, 715, 717, 709, 707
0.16	735, 712, 726, 760, 727
0.28	809, 796, 763, 791, 811
0.44	767, 771, 799, 799, 791

## Exemplo 2: Aminoácidos x Ganho de peso

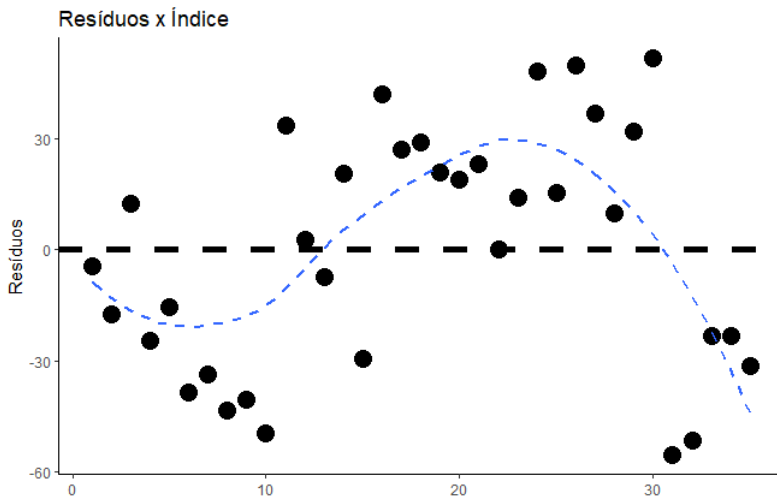




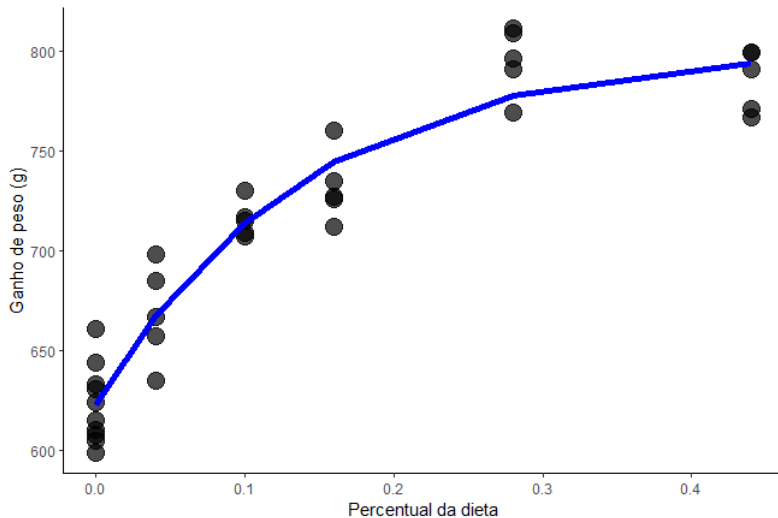
# Aplicando a Regressão Linear Simples (RLS)



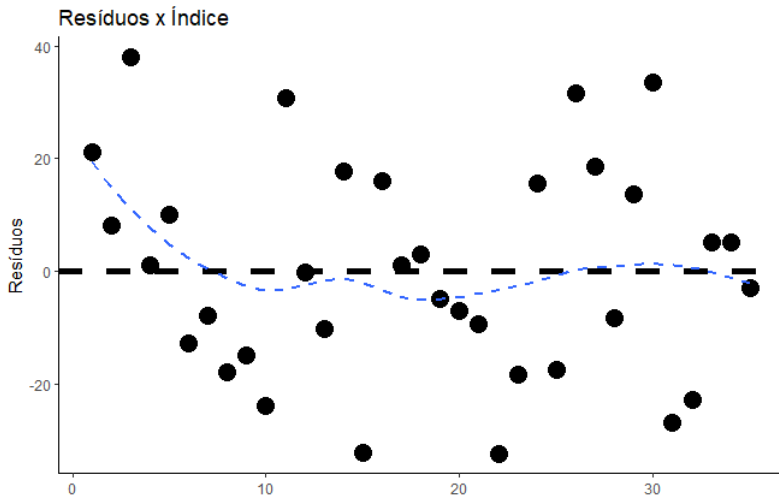
# Resíduos (RLS)



# Mesmo exemplo (MNL)



# Resíduos (MNL)



# EQM e EAM (RLS x MNL)

Modelo	EQM	EAM
RLS	1007,44	27,95
MNL	349,54	15,47
Proporção	<b>2,88</b>	<b>1,81</b>

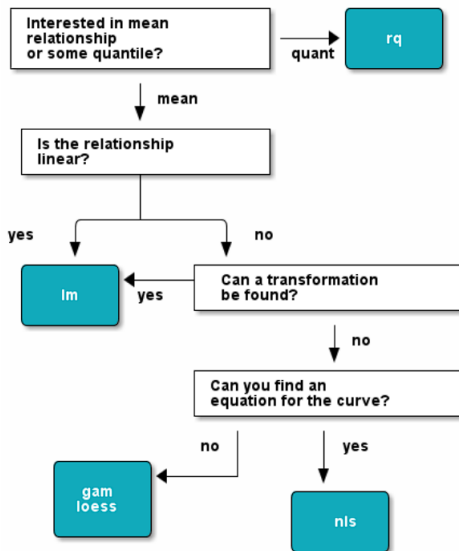
# Desvantagens da RLS (nestes exemplos)

- Identificação de autocorrelação residual;
- Quebra do pressuposto de independência entre as observações (mesmo indivíduo sendo observado em diferentes situações de **tempo**).
- Valores elevados de EQM e EAM (distância do valor observado).
- Possível solução:

# Desvantagens da RLS (nestes exemplos)

- Identificação de autocorrelação residual;
- Quebra do pressuposto de independência entre as observações (mesmo indivíduo sendo observado em diferentes situações de **tempo**).
- Valores elevados de EQM e EAM (distância do valor observado).
- Possível solução: **Modelos Não-Lineares**.

# Alternativas da Análise de Regressão





# Objetivos do curso

Ao final do curso, os alunos inscritos deverão estar aptos a:

# Objetivos do curso

Ao final do curso, os alunos inscritos deverão estar aptos a:

- Realizar a análise exploratória dos dados disponíveis;

# Objetivos do curso

Ao final do curso, os alunos inscritos deverão estar aptos a:

- Realizar a análise exploratória dos dados disponíveis;
- Ajustar modelos não-lineares no software R;

# Objetivos do curso

Ao final do curso, os alunos inscritos deverão estar aptos a:

- Realizar a análise exploratória dos dados disponíveis;
- Ajustar modelos não-lineares no software R;
- Extrair as principais informações do modelo;

# Objetivos do curso

Ao final do curso, os alunos inscritos deverão estar aptos a:

- Realizar a análise exploratória dos dados disponíveis;
- Ajustar modelos não-lineares no software R;
- Extrair as principais informações do modelo;
- Realizar a comparação entre equações, identificando o(s) melhor(es) modelo(s);

# Objetivos do curso

Ao final do curso, os alunos inscritos deverão estar aptos a:

- Realizar a análise exploratória dos dados disponíveis;
- Ajustar modelos não-lineares no software R;
- Extrair as principais informações do modelo;
- Realizar a comparação entre equações, identificando o(s) melhor(es) modelo(s);
- Resumir os principais resultados encontrados da análise;

# Objetivos do curso

Ao final do curso, os alunos inscritos deverão estar aptos a:

- Realizar a análise exploratória dos dados disponíveis;
- Ajustar modelos não-lineares no software R;
- Extrair as principais informações do modelo;
- Realizar a comparação entre equações, identificando o(s) melhor(es) modelo(s);
- Resumir os principais resultados encontrados da análise;
- **OBS:** modelos com foco em **curvas de crescimento e lactação**.

# Diferenças (ML e MNL)

- De maneira generalizada, temos que os modelos de regressão são representados pela notação:

$$y_i = f(x_i, \boldsymbol{\theta}) + e_i$$

e o objetivo é minimizar a soma de quadrados dos erros, ou seja, minimizar:

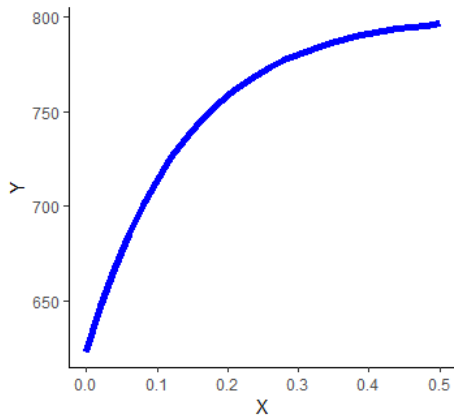
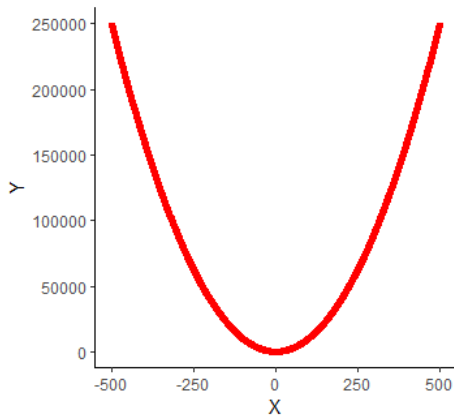
$$SQE(\boldsymbol{\theta}) = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n [y_i - f(x_i, \boldsymbol{\theta})]^2.$$



# Diferenças entre ML e MNL

- Quando  $f(x_i; \theta)$  representa uma relação **linear** entre  $X$  e  $Y$ , existe uma fórmula fechada para estimação do vetor de parâmetros  $\theta$ .
- No entanto, quando essa relação é **não-linear**, não existe uma equação que nos retorne diretamente as estimativas de  $\theta$ .
- Logo, são necessários métodos numéricos/iterativos para estimação de parâmetros.

# Como diferenciar ML e MNL?



# Diferenças entre ML e MNL

## Regressão Linear Múltipla (3 parâmetros)

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon_i$$

- $\frac{\partial f(x)}{\partial \beta_0} = 1$
- $\frac{\partial f(x)}{\partial \beta_1} = x_1$
- $\frac{\partial f(x)}{\partial \beta_2} = x_2$

# Diferenças entre ML e MNL

## Modelo Não-Linear

$$y = \theta_1 [1 - e^{-\theta_2(x-\theta_3)}]$$

- $\frac{\partial f(x)}{\partial \theta_1} = 1 - \exp \{-\theta_2(x - \theta_3)\}$
- $\frac{\partial f(x)}{\partial \theta_2} = -\theta_1(\theta_3 - x)\exp \{\theta_2(x - \theta_3)\}$
- $\frac{\partial f(x)}{\partial \theta_3} = -\theta_1\theta_2\exp \{\theta_2(x - \theta_3)\}$

# Diferenças entre ML e MNL

## Modelo Não-Linear

$$y = \theta_1 [1 - e^{-\theta_2(x-\theta_3)}]$$

- $\frac{\partial f(x)}{\partial \theta_1} = 1 - \exp \{-\theta_2(x - \theta_3)\}$
- $\frac{\partial f(x)}{\partial \theta_2} = -\theta_1(\theta_3 - x)\exp \{\theta_2(x - \theta_3)\}$
- $\frac{\partial f(x)}{\partial \theta_3} = -\theta_1\theta_2\exp \{\theta_2(x - \theta_3)\}$

⇒ **Requer uso de procedimentos computacionais.**

# Vantagens

- Ser possível quantificar relações não-lineares entre variáveis;

# Vantagens

- Ser possível quantificar relações não-lineares entre variáveis;
- Os parâmetros de  $f(x_i; \theta)$  usualmente possuem interpretação prática e/ou biológica;

# Vantagens

- Ser possível quantificar relações não-lineares entre variáveis;
- Os parâmetros de  $f(x_i; \theta)$  usualmente possuem interpretação prática e/ou biológica;
- Modelos parcimoniosos;



# Vantagens

- Ser possível quantificar relações não-lineares entre variáveis;
- Os parâmetros de  $f(x_i; \theta)$  usualmente possuem interpretação prática e/ou biológica;
- Modelos parcimoniosos;
- Permitem a extrapolação de valores.

# Desvantagens

- Requer conhecimento prévio do fenômeno em estudo e da função matemática em estudo;
- Dificuldades de convergência em algumas situações;
- Tempo computacional para o caso de grandes amostras.

# Quais modelos não-lineares utilizar?

- **Crescimento:** Gompertz, Logístico, Richards, von Bertalanffy, Weibull etc. (sigmoides);
- **Lactação:** Cobby e Le Du, Nelder, Dhanoa, Wood etc.

# Prática 1: Análise exploratória (*Orange*)

Orange {datasets}

R Documentation

## Growth of Orange Trees

### Description

The *Orange* data frame has 35 rows and 3 columns of records of the growth of orange trees.

### Usage

*Orange*

### Format

An object of class `c("nfnGroupedData", "nfGroupedData", "groupedData", "data.frame")` containing the following columns:

#### Tree

an ordered factor indicating the tree on which the measurement is made. The ordering is according to increasing maximum diameter.

#### age

a numeric vector giving the age of the tree (days since 1968/12/31)

#### circumference

a numeric vector of trunk circumferences (mm). This is probably "circumference at breast height", a standard measurement in forestry.

# Prática 1: Análise exploratória (*Orange*)

- 1 Carregar o data frame *Orange*;
- 2 Filtrar apenas a primeira e segunda árvores;
- 3 Gerar figuras;
- 4 Criar figura para todas as árvores simultaneamente;
- 5 Importância da classe *groupedData*.

# Prática 1: Análise exploratória (*Orange*)

```
##### Prática 1: Visualização de dados #####

#### 1.1. Carregando base de dados ####

head(Orange)
View(Orange)
?Orange

...

#### 1.7. Transformando data.frame em groupedData ####

head(Orange_df)

# obj_agrupado = groupedData(y ~ x, data = data.frame)
Orange_gd = groupedData(circumference ~ age | Tree,
                        data = Orange_df)

plot(Orange_df)
plot(Orange_gd)
```

# Método dos Mínimos Quadrados (MMQ) - ML

- De maneira generalizada, temos que os modelos de regressão são representados pela notação:

$$y_i = f(x_i, \boldsymbol{\theta}) + e_i$$

e o objetivo é minimizar a soma de quadrados dos erros, ou seja, minimizar:

$$SQE(\boldsymbol{\theta}) = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n [y_i - f(x_i, \boldsymbol{\theta})]^2.$$

# Método dos Mínimos Quadrados (MMQ) - ML

- De maneira generalizada, temos que os modelos de regressão são representados pela notação:

$$y_i = f(x_i, \theta) + e_i$$

e o objetivo é minimizar a soma de quadrados dos erros, ou seja, minimizar:

$$SQE(\theta) = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n [y_i - f(x_i, \theta)]^2.$$

- Se  $f(x_i, \theta)$  é linear nos parâmetros, considerando a forma matricial, então:

$$\hat{\theta} = (X'X)^{-1}X'y$$



# Método dos Mínimos Quadrados (MMQ) - MNL

- De maneira generalizada, temos que os modelos de regressão são representados pela notação:

$$y_i = f(x_i, \boldsymbol{\theta}) + e_i$$

e o objetivo é minimizar a soma de quadrados dos erros, ou seja, minimizar:

$$SQE(\boldsymbol{\theta}) = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n [y_i - f(x_i, \boldsymbol{\theta})]^2.$$

# Método dos Mínimos Quadrados (MMQ) - MNL

- De maneira generalizada, temos que os modelos de regressão são representados pela notação:

$$y_i = f(x_i, \theta) + e_i$$

e o objetivo é minimizar a soma de quadrados dos erros, ou seja, minimizar:

$$SQE(\theta) = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n [y_i - f(x_i, \theta)]^2.$$

- Isso não é possível de ser feito diretamente.

# Método dos Mínimos Quadrados (MMQ) - MNL

- De maneira generalizada, temos que os modelos de regressão são representados pela notação:

$$y_i = f(x_i, \theta) + e_i$$

e o objetivo é minimizar a soma de quadrados dos erros, ou seja, minimizar:

$$SQE(\theta) = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n [y_i - f(x_i, \theta)]^2.$$

- Isso não é possível de ser feito diretamente.
- SOLUÇÃO** → Métodos Computacionais.

# Algoritmo de Gauss-Newton

## Principais objetivos

- A partir de um chute inicial para o vetor de parâmetros de interesse, buscar valores cada vez mais próximos de  $\hat{\theta}$ .
- A cada passo do algoritmo, espera-se que  $SQE(\theta)$  diminua até que haja **convergência**.

# Algoritmo de Gauss-Newton

Neste método, o primeiro passo é reescrever a função esperança  $f(x_i, \theta)$  através de uma aproximação por Séries de Taylor em volta do chute inicial  $\hat{\theta}^*$  (WEISBERG, 2005):

$$f(x_i, \theta) \approx f(x_i, \theta^*) + \mathbf{u}_i(\theta^*)'(\theta - \theta^*)$$

Em que:

- $\theta$  é o valor original do vetor de parâmetros;
- $\theta^*$  é o vetor de chutes iniciais para o vetor de parâmetros;
- $\mathbf{u}_i(\theta^*)'$  é o vetor de escores, que contém as derivadas em relação a cada parâmetro considerando os chutes iniciais  $\partial f(x_i, \theta) / \partial \theta_j$ .

# Algoritmo de Gauss-Newton

Assim, a representação da Soma de Quadrados dos Erros pode ser reescrita da seguinte forma:

$$\begin{aligned}SQE(\boldsymbol{\theta}) &= \sum_{i=1}^n e_i^2 = \sum_{i=1}^n [y_i - f(x_i, \boldsymbol{\theta})]^2 \\&= \sum_{i=1}^n [y_i - f(x_i, \boldsymbol{\theta}^*) + \mathbf{u}_i(\boldsymbol{\theta}^*)'(\boldsymbol{\theta} - \boldsymbol{\theta}^*)]^2 \\&= \sum_{i=1}^n [\hat{e}_i^* - \mathbf{u}_i(\boldsymbol{\theta}^*)'(\boldsymbol{\theta} - \boldsymbol{\theta}^*)]^2\end{aligned}$$

# Algoritmo de Gauss-Newton

Considerando a forma matricial e isolando o termo do valor atualizado de  $\theta$  ( $\hat{\theta}$ ), podemos calcular o valor atualizado como segue:

$$\hat{\theta} = \theta^* + \left[ U(\theta^*)' U(\theta^*) \right]^{-1} U(\theta^*)' \hat{e}^*$$

Em que  $U(\theta^*)$  é a matriz de derivadas em relação à atual estimativa dos parâmetros ( $n \times p$ ) e  $e^* = y - f(x_i, \theta^*)$ .

# Passo a passo (resumo)

Portanto, o algoritmo de Gauss-Newton consiste em:

- 1 Escolher um vetor de chutes iniciais  $\boldsymbol{\theta}^{(0)}$  e computar  $SQE(\boldsymbol{\theta}^{(0)})$ ;
- 2 Definir o contador de iterações para  $j = 0$ ;
- 3 Computar  $\mathbf{U}(\boldsymbol{\theta})^{(j)}$  e  $\hat{\boldsymbol{\epsilon}}^{(j)}$ , atualizando assim o valor estimado para  $\boldsymbol{\theta}^{(j+1)}$ , e calcular  $SQE(\boldsymbol{\theta}^{(1)})$ ;
- 4 Para o algoritmo se  $SQE(\boldsymbol{\theta}^{(j)}) - SQE(\boldsymbol{\theta}^{(j+1)})$  for suficientemente pequeno (convergência). Caso contrário, definir o contador para  $j = j + 1$ .

OBS:



# Passo a passo (resumo)

Portanto, o algoritmo de Gauss-Newton consiste em:

- 1 Escolher um vetor de chutes iniciais  $\theta^{(0)}$  e computar  $SQE(\theta^{(0)})$ ;
- 2 Definir o contador de iterações para  $j = 0$ ;
- 3 Computar  $U(\theta)^{(j)}$  e  $\hat{e}^{(j)}$ , atualizando assim o valor estimado para  $\theta^{(j+1)}$ , e calcular  $SQE(\theta^{(1)})$ ;
- 4 Para o algoritmo se  $SQE(\theta^{(j)}) - SQE(\theta^{(j+1)})$  for suficientemente pequeno (convergência). Caso contrário, definir o contador para  $j = j + 1$ .

**OBS:** Se  $j$  for muito grande, o algoritmo para e consideramos que não houve convergência.

# Algoritmo de Gauss-Newton (saída do R)

```
> m1 = nls(weight ~ SSlogis(Time, Asym, xmid, scal),  
+          data = s1, trace = TRUE)  
12.99792      (4.41e+00): par = (51.48132 9.95398 17.51117)  
1.359296      (1.98e-01): par = (58.43835 10.5113 21.20206)  
1.066214      (1.07e-01): par = (57.1321 9.498378 20.15343)  
1.056704      (1.18e-03): par = (57.42053 9.617132 20.35248)  
1.056646      (4.44e-04): par = (57.40052 9.603727 20.33702)  
1.056645      (3.61e-05): par = (57.40261 9.60485 20.33855)  
1.056645      (3.46e-06): par = (57.40242 9.604743 20.33841)  
1.056645      (6.21e-06): par = (20.33841 57.40242 9.604743)
```

# Procedimento geral (MNL)

- 1 Estudar o fenômeno + análise exploratória dos dados;
- 2 Selecionar uma ou várias equações que possam descrever este fenômeno;
- 3 Ajustar o(s) modelo(s) proposto(s);
- 4 Avaliar/comparar modelos;
- 5 Extrair informações (valores ajustados, formato da curva, parâmetros estimados etc.).

# O modelo Logístico

- O modelo Logístico, usualmente utilizado para representar **crescimento**, é denotado por:

$$y = \frac{\beta_1}{1 + \exp[-\beta_3(t - \beta_2)]},$$

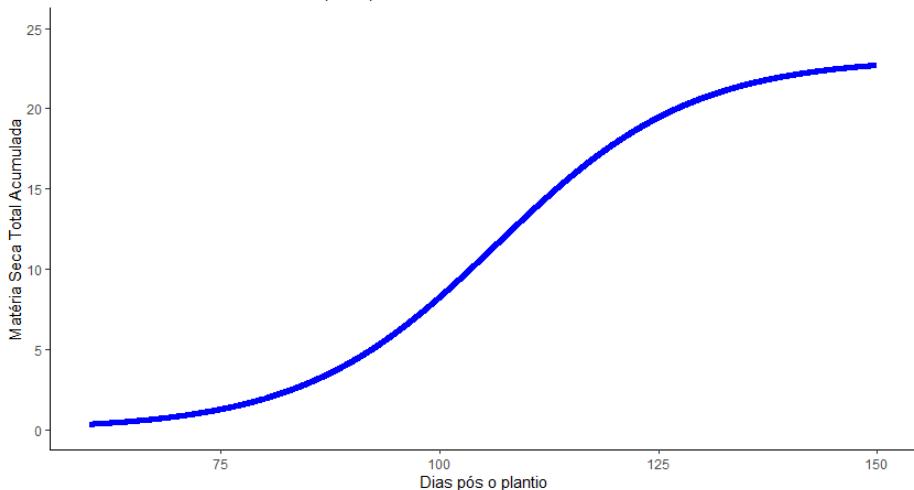
Em que:

- $y$  é a variável resposta (peso, comprimento etc);
- $\beta_1$  é o peso/comprimento/etc assintótico (*Asym*);
- $\beta_2$  é o ponto (no eixo  $X$ ) que a curva atinge o ponto de inflexão (*xmid*);
- $\beta_3$  é o parâmetro de escala (*scal*).

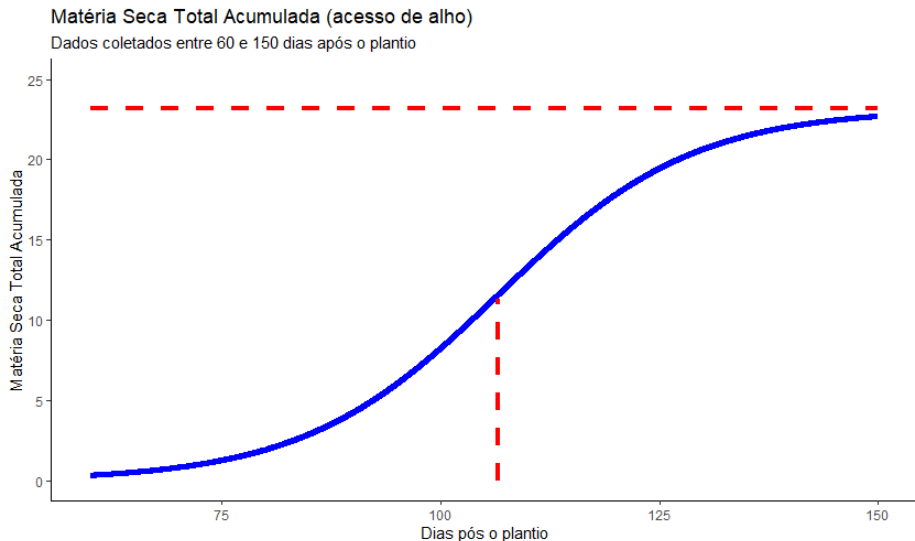
## Modelo Logístico (curva)

Matéria Seca Total Acumulada (acesso de alho)

Dados coletados entre 60 e 150 dias após o plantio



## Modelo Logístico (curva)



# A função *nls*

stats (version 3.6.2)

## **nls: Nonlinear Least Squares**

### **Description**

Determine the nonlinear (weighted) least-squares estimates of the parameters of a nonlinear model.

### **Usage**

```
nls(formula, data, start, control, algorithm,  
     trace, subset, weights, na.action, model,  
     lower, upper, ...)
```

## Prática 2.1: Usando *Self-starting function*

```
#### 2.1. Modelo Logístico (Self-Starting) ####
```

```
##### Prática 2: Ajustando Modelos não lineares no R #####
```

```
df_p1
```

```
m1_p1 = nls(circumference ~ SSlogis(age, Asym, xmid, scal),  
            data = df_p1)
```

```
summary(m1_p1)
```

```
# Dica: verificar ?selfStart
```

```
?selfStart
```



## Prática 2.2: Criando a função para o modelo Logístico

```
#### 2.2. Modelo Logístico (função própria) ####

logistic = function(age, Asym, xmid, scal){
  Asym / (1 + exp(- scal * (age - xmid)))
}

m2_p1 = nls(circumference ~ logistic(age, Asym, xmid, scal),
            start = list(Asym = 150, xmid = 600, scal = .001),
            data = df_p1)

m2_p1

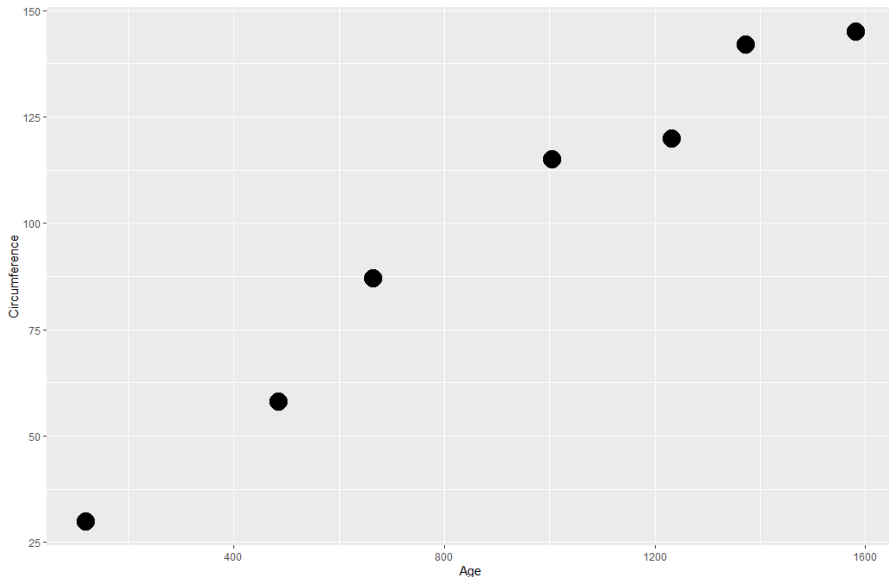
...

#### 2.4.4. Predizendo novos valores ####

dados = data.frame(age = 1200)
predict(m1_p1, dados)

dados2 = data.frame(age = c(400, 587, 1471))
predict(m1_p1, dados2)
```

# Chutes iniciais



# Requisitos para bons chutes iniciais

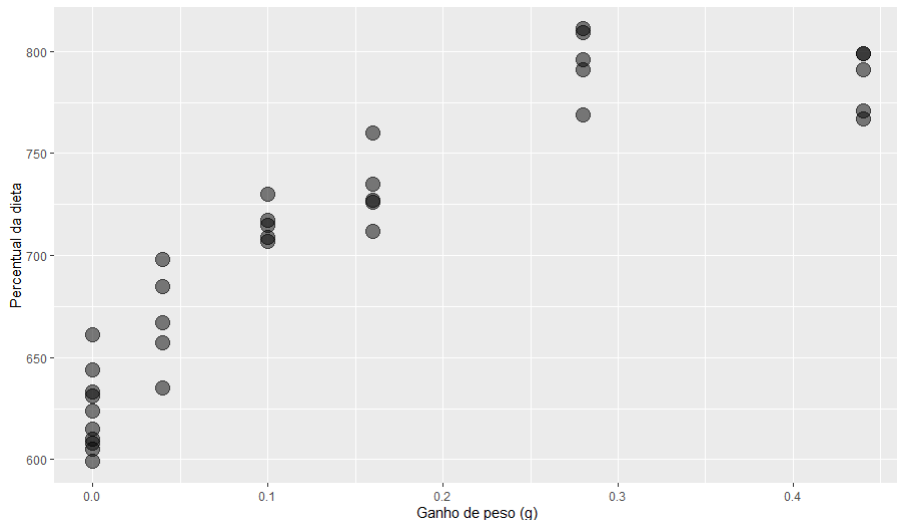
- Conhecimento prévio sobre o fenômeno descrito;
- Conhecimento da equação e dos seus parâmetros;
- Análise gráfica;
- Leitura de artigos com dados semelhantes;
- Artifícios matemáticos.

## Prática 3: Artifícios matemáticos para chutes iniciais

**Exemplo (WEISBERG, 2005):** Um experimento foi conduzido para estudar os efeitos no crescimento de determinada espécie de ave de diferentes quantidades A de metionina (aminoácido), variando de um controle sem suplementação a 0,44% da dieta total.

Quantidade (X)	Ganho (Y)
0.00	644, 631, 661, 624, 633, 610, 615, 605, 608, 599
0.04	698, 667, 657, 685, 635
0.10	730, 715, 717, 709, 707
0.16	735, 712, 726, 760, 727
0.28	809, 796, 763, 791, 811
0.44	767, 771, 799, 799, 791

# Prática 3: Artifícios matemáticos para chutes iniciais



# Prática 3: Artifícios matemáticos para chutes iniciais

Função escolhida para descrever o ganho de peso:

$$y = \beta_1 + \beta_2 [1 - e^{(-\beta_3 * A)}]$$

Em que:  $\beta_1$  é o intercepto (eixo  $Y$ ),  $\beta_2$  é a diferença entre o ganho assintótico e o intercepto e  $\beta_3$  é um parâmetro de escala.

# Prática 3: Artifícios matemáticos para chutes iniciais

Função escolhida para descrever o ganho de peso:

$$y = \beta_1 + \beta_2 [1 - e^{(-\beta_3 * A)}]$$

Em que:  $\beta_1$  é o intercepto (eixo  $Y$ ),  $\beta_2$  é a diferença entre o ganho assintótico e o intercepto e  $\beta_3$  é um parâmetro de escala.

Para quais parâmetros podemos atribuir chutes através do gráfico?

## Prática 3: Artifícios matemáticos para chutes iniciais

$$y = \beta_1 + \beta_2 [1 - e^{(-\beta_3 * A)}]$$



## Prática 3: Artifícios matemáticos para chutes iniciais

$$y = \beta_1 + \beta_2 [1 - e^{(-\beta_3 * A)}]$$

**Verificar procedimento no R**

# Prática 3: Artifícios matemáticos para chutes iniciais

```
#### Prática 3: Chutes iniciais (2) ####

#### 3.1. Ganho de peso de aves ####

#### 3.1.1 Base de dados ####

Amount = c(rep(0, 10), rep(c(0.04, .1, .16, .28, .44), each = 5))
Gain = c(644, 631, 661, 624, 633, 610, 615, 605, 608, 599,
        698, 667, 657, 685, 635, 730, 715, 717, 709, 707,
        735, 712, 726, 760, 727, 809, 796, 769, 791, 811,
        767, 771, 799, 799, 791)

...

#### 3.1.5. Gráfico de valores ajustados ####

aj = fitted(mod3)
turkey %>% mutate(aj = aj) %>%
  ggplot() +
  geom_point(aes(x = Amount, y = Gain), size = 4) +
  geom_line(aes(x = Amount, y = aj), linewidth = 2, color = 'blue')
```

# Criando outras equações

O modelo Logístico não é o único capaz de descrever curvas de crescimento. Abaixo, seguem outras duas equações:

- Gompertz:

$$y = \beta_1 e^{-e^{[-\beta_3(t-\beta_2)]}}$$

- von Bertalanffy:

$$y = \beta_1 [1 - \beta_2 e^{(-\beta_3 t)}]$$

Em que:  $\beta_1$  é o peso assintótico;  $\beta_2$  é o tempo até o ponto de inflexão e  $\beta_3$  é o parâmetro de escala (mesmos do modelo logístico).

# Criando outras equações

O modelo Logístico não é o único capaz de descrever curvas de crescimento. Abaixo, seguem outras duas equações:

- Gompertz:

$$y = \beta_1 e^{-e^{[-\beta_3(t-\beta_2)]}}$$

- von Bertalanffy:

$$y = \beta_1 [1 - \beta_2 e^{(-\beta_3 t)}]$$

Em que:  $\beta_1$  é o peso assintótico;  $\beta_2$  é o tempo até o ponto de inflexão e  $\beta_3$  é o parâmetro de escala (mesmos do modelo logístico).

⇒ **Criar no R uma função para cada modelo.**

# Prática 4: Ajustando novos modelos

```
##### Prática 4: Ajustando novos modelos #####

##### 4.1. Gompertz #####

gompertz = function(dap, phi1, phi2, phi3){
  phi1 * exp(- exp(- phi3 * (dap - phi2)))
}

...

##### 4.2. von Bertalanffy #####

vonb = function(dap, phi1, phi2, phi3){
  phi1 * ((1 - phi2 * exp((- phi3) * dap)) ^ 3)
}

m3_p1 = nls(circumference ~ vonb(age, phi1, phi2, phi3),
            start = list(phi1 = 150, phi2 = 800, phi3 = .001),
            data = df_p1)

m3_p1
summary(m3_p1)
```

# Medidas de qualidade de ajuste

## Importância

- Identificar o melhor entre vários modelos não-lineares (ex: Logístico x Gompertz).
- Encontrar o modelo que estime valores mais próximos dos dados reais.

# Medidas de qualidade de ajuste

## Importância

- Identificar o melhor entre vários modelos não-lineares (ex: Logístico x Gompertz).
- Encontrar o modelo que estime valores mais próximos dos dados reais.

→ **Algumas medidas importantes:** AIC, BIC, EQM e EAM.

# Medidas baseadas na verossimilhança

## Descrição

São medidas que levam em consideração o valor estimado por máxima verossimilhança e número de variáveis do modelo.

- Critério de Akaike (AIC):

$$AIC = -2.\ln(L) + 2.p$$

- Critério de Schartz (BIC):

$$BIC = -2.L + p.\log(N)$$

Em que:  $L$  é o logaritmo da razão de verossimilhança,  $p$  é o número de parâmetros do modelo e  $N$  é o tamanho da amostra.



# Medidas baseadas na precisão das estimativas

## Descrição

São medidas que levam em consideração a proximidade entre valores real e estimado.

- Erro Quadrático Médio (EQM):

$$EQM = \sum_{i=1}^N \frac{(y_i - \hat{y}_i)^2}{N}$$

- Erro Absoluto Médio (EAM):

$$EAM = \sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{N}$$

Em que:  $y_i$  é o  $i$ -ésimo valor observado,  $\hat{y}_i$  é o  $i$ -ésimo valor esperado e  $N$  é o tamanho amostral.

# Interpretação das medidas

Para as medidas citadas, **menores valores apontam para o melhor modelo.**

# Procedimento para comparação de modelos

- Calcular AIC, BIC, EAM, EAM e outras medidas necessárias de todos os modelos.

⇒ **Comparar modelos de Gompertz, von Bertalanffy e Logístico**

# Prática 5: Comparando modelos

```
#### Prática 5: Comparando os três modelos ####
```

```
#### 5.1. AIC ####
```

```
AIC(m1_p1) # Gompertz
```

```
AIC(m2_p1) # Logístico
```

```
AIC(m3_p1) # von Bertalanffy
```

```
...
```

```
#### 5.4. EAM (MAE) ####
```

```
mae(df_p1$circumference, fitted(m1_p1)) # Gompertz
```

```
mae(df_p1$circumference, fitted(m2_p1)) # Logístico
```

```
mae(df_p1$circumference, fitted(m3_p1)) # von Bertalanffy
```

```
# Pacote Metrics: https://cran.r-project.org/web/packages/Metrics/Metrics.pdf
```

