Regressão Não-Linear

Vanessa Ferreira Sehaber e Vanessa Monteiro

17 de setembro de 2012

1 Crescimento populacional

O pacote `car` do *software* `R` disponibiliza um banco de dados chamado USPop referente ao censo populacional dos U.S.A., de 1790 a 2000.

```
> library(car)
> # visualizando os dados na tabela
> head(USPop)

  year population decade
1 1790   3.929214      0
2 1800   5.308483      1
3 1810   7.239881      2
4 1820   9.638453      3
5 1830  12.860702      4
6 1840  17.063353      5

> dim(USPop)

[1] 22  3
```

Para termos conhecimento da representação dos dados, vamos plotá-los a seguir:

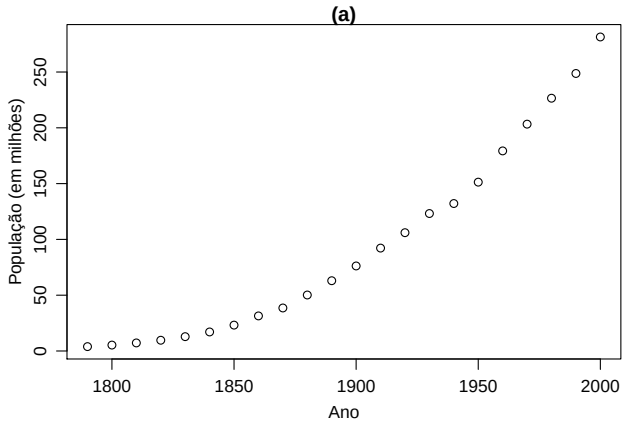

Figura 1: Crescimento populacional: dados Censo U.S.A. dos anos de 1790 a 2000.

Há o interesse em ajustar um modelo que melhor represente o comportamento do crescimento populacional dos U.S.A.

A princípio, ajustaremos um modelo linear para analisarmos o resultado que obteremos.

Por definição, **modelos lineares** são modelos com parâmetros lineares e que podem ser representados da seguinte forma:

$$Y = X\beta + \epsilon$$

- Y é o vetor resposta;
- X é a matrix das variáveis preditoras;
- β é o vetor dos coeficientes regressores;
- ϵ é o vetor de erros aleatórios.

A função média é $E(Y|X) = X\beta$, com $E(\epsilon) = 0$, a variância $V(Y|X) = \sigma^2$, os erros não são correlacionados e, ainda, $\epsilon \sim N(0, \sigma^2)$.

O método de mínimo quadrados é utilizado para estimar β e é dado por

$$S(\beta) = \sum_{i=1}^{n} (Y - \hat{Y})^2$$

sendo $\hat{Y}$ a estimação de Y.

Vamos ajustar o modelo de regressão linear:

```
> reg.lin <- lm(population ~ year, data = USPop)
```

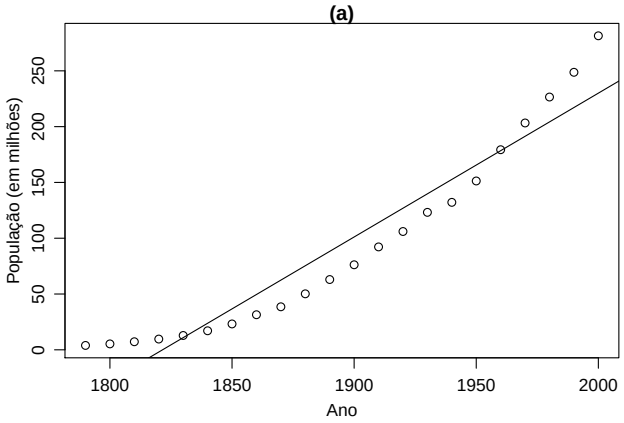

FIGURA 2: Ajuste do modelo de regressão plinear aos dados do censo populacional dos U.S.A.

Assim, podemos escrever

$$Y = -2347,713 + 1,289X + \epsilon$$

As equações sobre cálculos inferenciais de modelos lineares podem ser encontrados em [**?**] e [**?**].
Conforme `summary(reg.lin)`, podemos observar mais detalhes sobre o modelo.

```
> summary(reg.lin)

Call:
lm(formula = population ~ year, data = USPop)

Residuals:
   Min     1Q Median     3Q    Max
-25.25 -20.78 -10.11  19.58  51.42

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept) -2.348e+03  1.613e+02  -14.55 4.19e-12 ***
year         1.289e+00  8.508e-02   15.15 2.00e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 25.32 on 20 degrees of freedom
Multiple R-squared: 0.9198,        Adjusted R-squared: 0.9158
F-statistic: 229.5 on 1 and 20 DF,  p-value: 2.005e-12
```

De acordo com o teste-t, os valores dos parâmetros são significativos para o modelo. O coeficiente de determinação R^2 indica a proporção da variação explicada pela variável preditora X, $0 \leq R^2 \leq 1$. Quanto mais próximo de 1 implica que a maior variabilidade em Y é explicada pelo modelo de regressão. Neste exemplo, $R^2 = 0,9198$, ou seja, pode-se dizer que o modelo é explicativo.

```
> Anova(reg.lin)

Anova Table (Type II tests)

Response: population
          Sum Sq Df F value    Pr(>F)
year      147096  1   229.5 2.005e-12 ***
Residuals  12819 20
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

A soma dos quadrados residuais é 12819. Para ver ϵ, basta executar `residuals(reg.lin)`. A seguir, será representado o gráfico dos resíduos da regressão, conforme [**?**].

```
> # plotando a reta de regressão linear
> plot(population ~ USPop$year, data = USPop)
> PropEsp<-predict(reg.lin,newdata=list(year=seq(1790,2000,by=10)),se.fit=T)
> lines(PropEsp$fit~seq(1790,2000,by=10),lwd=2,col=6)
> upIC<-PropEsp$fit+qt(.975,summary(reg.lin)$df[2])*summary(reg.lin)$sigma
> loIC<-PropEsp$fit-qt(.975,summary(reg.lin)$df[2])*summary(reg.lin)$sigma
> lines(upIC~seq(1790,2000,by=10),col=4)
> lines(loIC~seq(1790,2000,by=10),col=4)
```

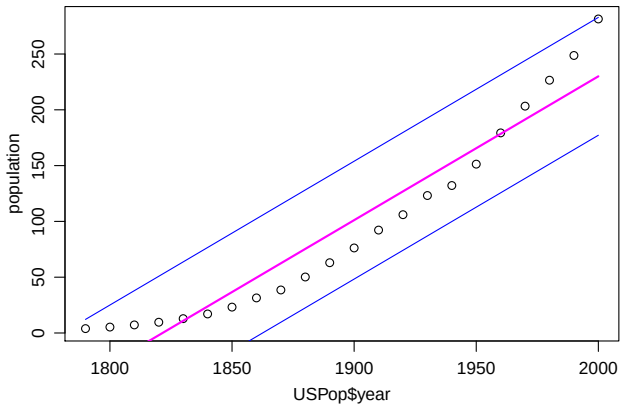

FIGURA 3: Intervalos de confiança do modelo de regressão linear aos dados do censo populacional dos U.S.A.

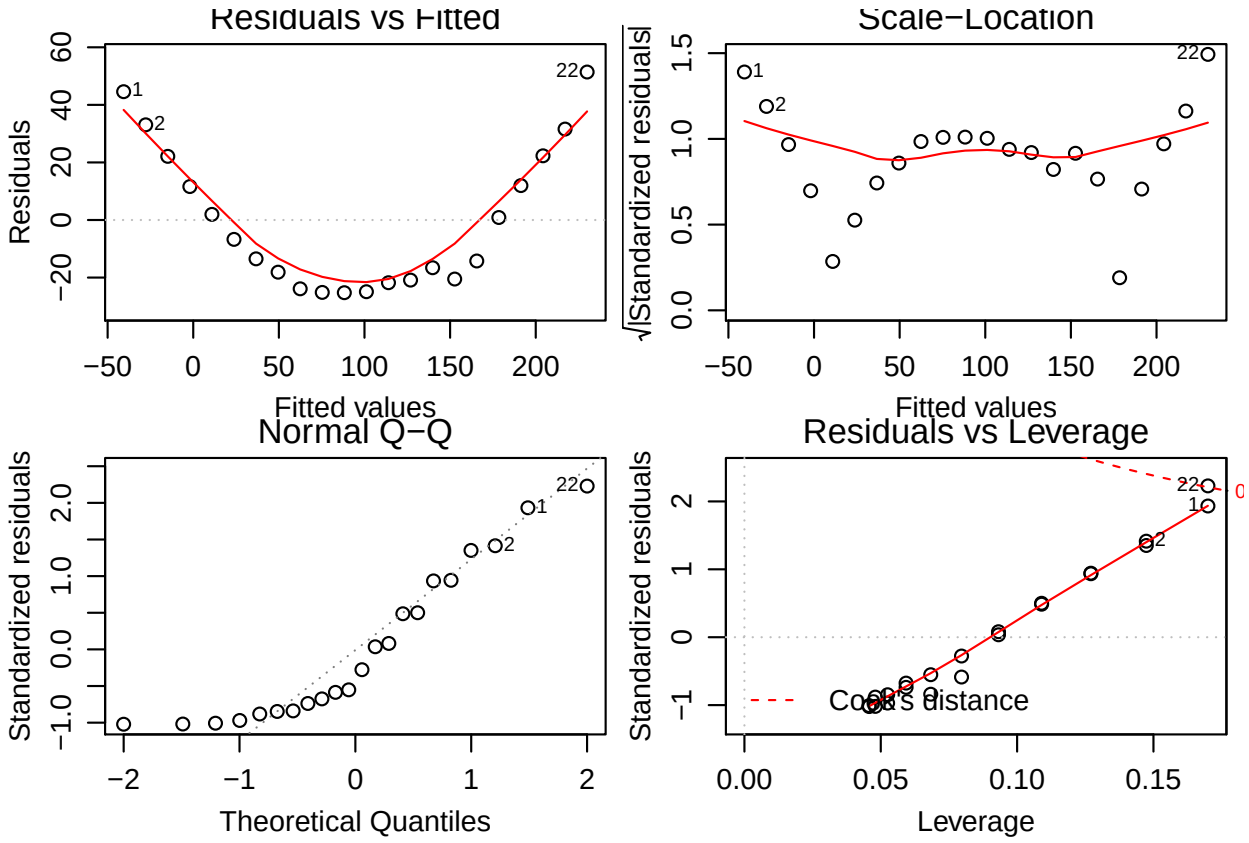

FIGURA 4: Gráficos de diagnóstico.

De acordo com [**?**], o gráfico de resíduos (Residuals vs Fitted) indica possível inadequação do modelo adotado, e as curvas sugerem que devemos procurar outras funções matemáticas que expliquem melhor o fenômeno.

Voltando ao gráfico (1), uma alternativa é ajustar uma regressão polinomial aos pontos.

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \epsilon$$

onde $E(Y|X) = \beta_0 + \beta_1 X + \beta_2 X^2$.

Regressão polinomial é utilizada em situações onde a resposta é curvilínea, até mesmo em complexas relações não-lineares, ao longo de intervalos relativamente pequenos.

```
> reg.pol <- lm(population ~ year + I(year^2), data=USPop)
> print(reg.pol)

Call:
lm(formula = population ~ year + I(year^2), data = USPop)

Coefficients:
(Intercept)         year    I(year^2)
  2.162e+04   -2.403e+01    6.681e-03

> #predict(reg.pol); residuals(reg.pol); coef(reg.pol) podem ser utilizados
```

Assim, podemos escrever o modelo como

$$Y = 21617,3916 - 24,0325X + 0,0066X^2$$

```
> summary(reg.pol)
```

```
Call:
lm(formula = population ~ year + I(year^2), data = USPop)

Residuals:
    Min      1Q  Median      3Q     Max
-7.5557 -0.4308  0.6051  1.4230  4.6486

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.162e+04  6.389e+02   33.83   <2e-16 ***
year        -2.403e+01  6.749e-01  -35.61   <2e-16 ***
I(year^2)    6.681e-03  1.780e-04   37.52   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.997 on 19 degrees of freedom
Multiple R-squared: 0.9989,        Adjusted R-squared: 0.9988
F-statistic:  8892 on 2 and 19 DF,  p-value: < 2.2e-16

> Anova(reg.pol)

Anova Table (Type II tests)

Response: population
           Sum Sq Df F value    Pr(>F)
year      11391.0  1  1268.1 < 2.2e-16 ***
I(year^2) 12648.4  1  1408.1 < 2.2e-16 ***
Residuals   170.7 19
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Comparando com os resultados do modelo anterior, R^2 está mais próximo de 1 e a soma dos quadrados residuais foi 170,7, resultando em um ajuste melhor.

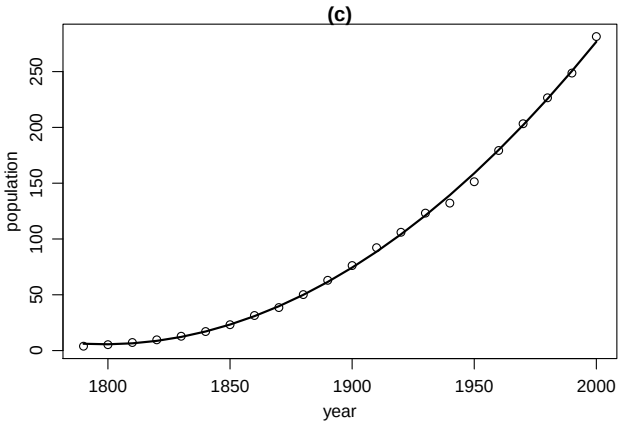

FIGURA 5: Ajuste do modelo de regressão polinomial aos dados do censo populacional dos U.S.A.

```
> # plotando o polinômio de regressão linear
> plot(population ~ year, data = USPop)
> PropEsp <-predict(reg.pol,newdata=list(year=seq(1790,2000,by=10)),se.fit=T)
> lines(PropEsp$fit~seq(1790,2000,by=10),lwd=2,col=6)
> upIC<-PropEsp$fit+qt(.975,summary(reg.pol)$df[2])*summary(reg.pol)$sigma
> loIC<-PropEsp$fit-qt(.975,summary(reg.pol)$df[2])*summary(reg.pol)$sigma
> lines(upIC~seq(1790,2000,by=10),col=4)
> lines(loIC~seq(1790,2000,by=10),col=4)
```

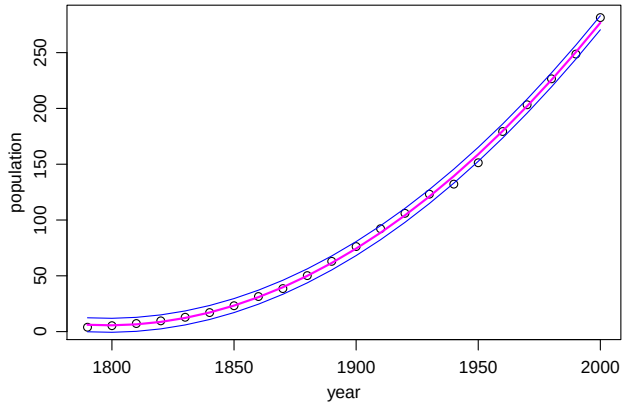

FIGURA 6: Intervalos de confiança do modelo de regressão polinomial aos dados do censo populacional dos U.S.A.

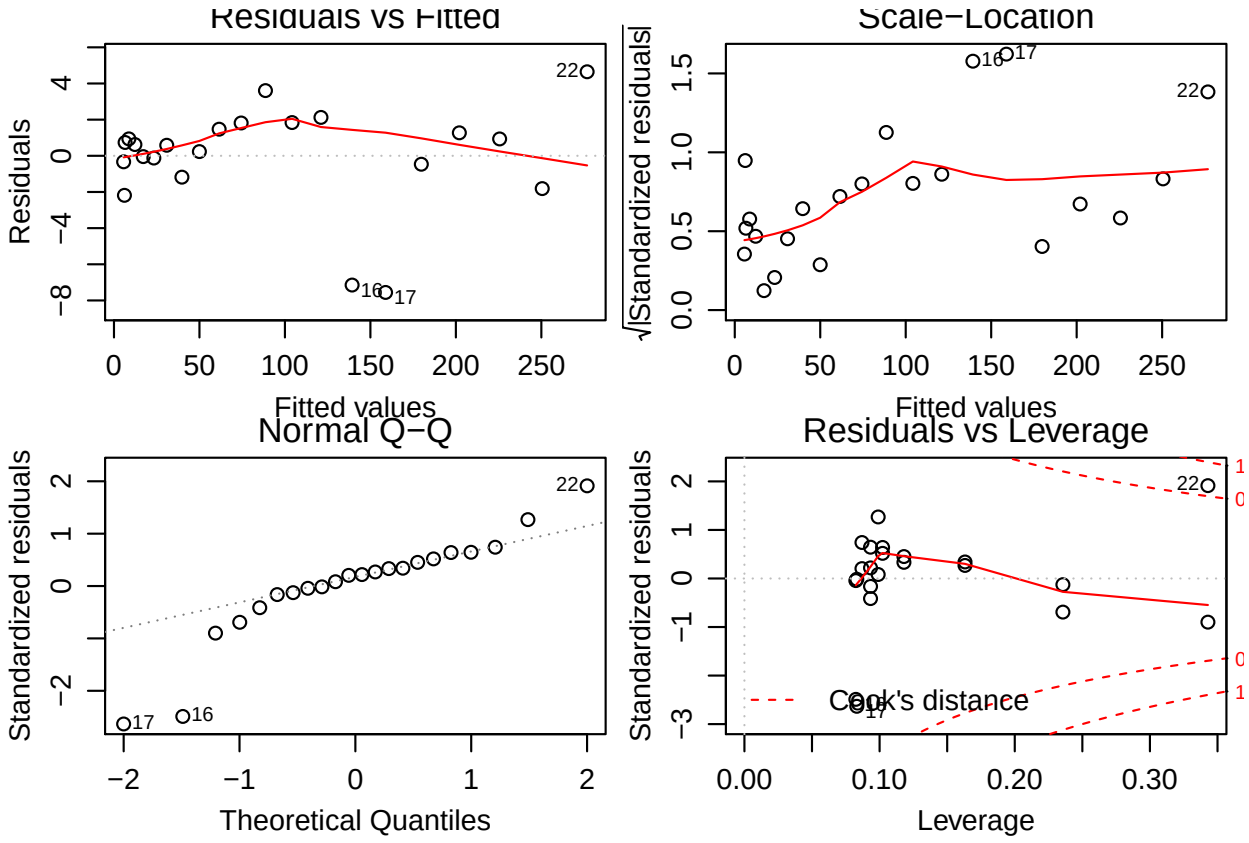

FIGURA 7: Gráficos de diagnóstico.

Em relação ao primeiro gráfico, não se observou padrão nos resíduos.
Para comparar com os modelos anteriores, vamos ajustar um modelo não-linear.
Os **modelos não-lineares** são aqueles nos quais seus parâmetros são não-lineares.
Por exemplo, [**?**] traz alguns modelos

$$E(Y|X) = e^{\theta_1+\theta_2 X} \quad (1)$$
$$E(Y|X) = \theta_1 + \theta_2 e^{-\theta_3 X} \quad (2)$$
$$E(Y|X) = (\theta_1 + \theta_2 X)^{-1} \quad (3)$$
$$E(Y|X) = (\theta_1 + \theta_2)^{-1}\left[e^{-\theta_1 X} + e^{-\theta_2 X}\right] \quad (4)$$

são todos não-lineares. No modelo (1), os parâmetros θ_1 e θ_2 são não lineares. No modelo (2), θ_1 e θ_2 são lineares enquanto que θ_3 é não-linear. Já nos modelos (3) e (4), ambos os parêmtros θ_1 e θ_2 são não-lineares.
Por definição, a relação entre Y e X é dada da seguinte forma:

$$Y = f(X;\theta) + \epsilon \quad (5)$$

A função média é $E(Y|X) = f(X,\theta)$, com $E(\epsilon) = 0$ e a variância $V(Y|X) = \sigma^2/w_i$, onde w_i é um peso utilizado para ponderar a variância.
O método de mínimo quadrados é utilizado para estimar θ e é dado por

$$S(\theta) = \sum_{i=1}^{n} w_i (Y_i - f(X_i, \hat{\theta}))^2$$

sendo $f(X, \hat{\theta})$ a estimativa de Y e $\hat{\theta}$ a estimativa de θ.
[**?**] traz algumas equações que são utilizadas como função média para o modelo de regressão não-linear:

FIGURA 8: Modelos de regressão não-linear para descrever curvas de crescimento.

Modelo	Função	Modelo	Função
A Schnute	$y_i = \frac{\beta_1}{\left(1+\beta_4 e^{(\beta_3(\beta_2 - x_i))}\right)^{\frac{1}{\beta_4}}} + e_i$	H Meloun II	$y_i = \beta_1 - e^{(-\beta_2 - \beta_3 x_i)} + e_i$
B Mitscherlich	$y_i = \beta_1\left(1 - e^{(\beta_3(\beta_2 - \beta_3 x_i))}\right) + e_i$	N Brody	$y_i = \beta_1\left(1 - \beta_2 e^{-\beta_3 x_i}\right) + e_i$
C Richards	$y_i = \frac{\beta_1}{\left(1+e^{(\beta_2 - \beta_3 x_i)}\right)^{\frac{1}{\beta_4}}} + e_i$	O von Bertalanffy	$y_i = \beta_1\left(1 - \beta_2 e^{-\beta_3 x_i}\right)^3 + e_i$
D Gompertz	$y_i = \beta_1 e^{\left(-e^{\left(\beta_2 - \beta_3 x_i\right)}\right)} + e_i$	P Michaelis-Menten	$y_i = \frac{\beta_1 x_i}{x_i + \beta_2} + e_i$
E Logístico	$y_i = \frac{\beta_1}{\left(1+e^{(\beta_2 - \beta_3 x_i)}\right)} + e_i$	Q Michaelis-Menten Modificado	$y_i = \frac{\beta_2 \beta_3^{\beta_4} + \beta_1 x_i^{\beta_4}}{\beta_3^{\beta_4} + x_i^{\beta_4}} + e_i$
F Meloun I	$y_i = \beta_1 - \beta_2 e^{(-\beta_3 x_i)} + e_i$		

Segundo [**?**], um modelo comum para crescimento populacional é o modelo de crescimento logístico, dado por:

$$Y = \frac{\theta_1}{1 + e^{-(\theta_2 + \theta_3 X)}} + \epsilon \tag{6}$$

onde Y é a resposta (o tamanho da população) e X será a preditora (ano).

Serão atribuídos valores para os parâmetros de (6) para conhecermos o seu comportamento ($\theta_1 = \theta_2 = \theta_3 = 1$).

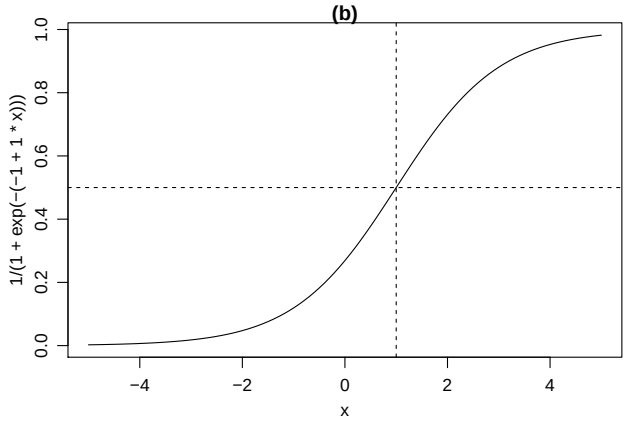

FIGURA 9: Gráfico da curva do modelo de crescimento logístico.

Alterando os valores dos parâmetros θ, os eixos poderão esticar ou encolher, e isto altera a taxa com que a curva varia desde seu menor valor 0 ao seu valor máximo (assintótico). A curva de crescimento logístico é simétrica em torno do valor X que é o ponto médio entre 0 e θ. Não é difícil mostrar que se $f\,(X = -\theta_2/\theta_3, \theta) = \theta_1/2$, e assim a curva é simétrica quando $X = -\theta_2/\theta_3$. O parâmetro θ_3 controla quão rapidamente a curva se desenvolve do menor valor assintótico 0 ao maior valor assintótico θ_1, e é, portanto, um parâmetro de taxa de crescimento.

Não é evidente que a curva da forma na figura 9 pode coincidir com os dados mostrados na figura 1, mas a parte da curva, delimitada para $X = -3$ e $X = 2$, pode ser capaz de se ajustar aos dados razoavelmente bem.

No `R` há uma função que faz a estimativa do modelo de regressão não-linear, chamada `nls`. Antes de chamar esta função, vamos mostrar seus argumentos:

```
> args(nls)

function (formula, data = parent.frame(), start, control = nls.control(),
    algorithm = c("default", "plinear", "port"), trace = FALSE,
    subset, weights, na.action, model = FALSE, lower = -Inf,
    upper = Inf, ...)
NULL
```

Vamos falar brevemente sobre cada um dos argumentos desta função:

formula o argumento `formula` é usado para chamar `nls` de uma função média. A fórmula equivalente a equação (6) é

```
population ~ theta1/(1+exp(-(-theta2 + theta3 * year)))
```

O lado esquerdo da fórmula especifica a variável resposta e seguindo por um $\sim$ que é usualmente lido como "é regressado por" ou é "modelado por". O lado direito é onde inserimos a expressão da função média.

start O argumento `start` é uma lista da função `nls` onde inserimos os valores dos parâmetros θ.

algorithm = "default" O algoritmo "`default`" na função `nls` é o algoritmo de Gauss-Newton. Outros possíveis algoritmos são "`plinear`" que refere-se ao algoritmo Golub-Pereyra para modelos parcialmente lineares e "`port`" para o caso de haver parâmetros com restrições.

Em particular, o algoritmo de Gauss-Newton estima os parâmtros para um problema de regressão não-linear por uma sequência de aproximações não lineares por meio de cálculo de Mínimos Quadrados Ponderados (ou Máxima Verossimilhança).

lower = - Inf, upper = Inf Uma das características de modelos não-lineares é que os parâmetros do modelo podem ser concentrados numa determinada região no modelo de crescimento logístico populacional, por exemplo, temos que $\theta_3 > 0$ pois a população está crescendo, e da mesma forma para $\theta_1 > 0$. Em alguns problemas, gostaríamos de ter certeza de que o algoritmo nunca irá considerar valores para θ fora do intervalo de factibilidade. Os argumentos `lower` e `upper` são vetores de limites superiores e inferiores. Se não especificado, todos os parâmetros são assumidos como irrestritos. Limites podem ser utilizados apenas com o algoritmo `port`. Os limites são ignorados com uma mensagem de `warning` se forem especificados em outros algoritmos.

control = nls.control() Esse argumento toma uma lista dos valores que modificam o critério usado no cálculo do algoritmo.

```
> args(nls.control)

function (maxiter = 50, tol = 1e-05, minFactor = 1/1024, printEval = FALSE,
    warnOnly = FALSE)
NULL
```

É utilizado para alterar o número máximo de iterações e a tolerância de convergência. Por exemplo, `control = nls.control(maxiter = 40, tol = 1e-6)`.

trace = FALSE Se `TRUE`, será imprimido os valores da soma dos quadrados residuais e os parâmetros estimados em cada iteração. O padrão é `FALSE`.

data, subset, weights, na.action Os argumentos `data, subset,na.action` especificam os dados para os quais o modelo serão ajustados e `weights` dá os pesos de w que são usados no ajuste de mínimos quadrados. Se `weights` está faltando, então todos os pesos serão iguais a 1.

1.1 Valores iniciais

Ao contrário de método de mínimos quadrados linear, a maioria dos algoritmos de mínimos quadrados não-linear requer a especificação dos valores iniciais para os parâmetros, os quais são θ_1, θ_2 e θ_3 para o modelo de crescimento logístico da equação 6. Para o modelo de crescimento logístico, podemos escrever:

$$Y \approx \frac{\theta_1}{1 + e^{-(\theta_2 + \theta_3 X)}} \tag{7}$$

$$Y/\theta_1 \approx \frac{1}{1 + e^{-(\theta_2 + \theta_3 X)}} \tag{8}$$

$$log\left[\frac{Y/\theta_1}{1 - Y/\theta_1}\right] \approx \theta_2 + \theta_3 X \tag{9}$$

Perceba que em (9) chegamos numa expressão linear com parâmetros θ_2 e θ_3. Dessa maneira, [**?**] sugere uma regressão linear para encontrarmos os valores iniciais para os parâmetros θ_2 e θ_3. A princípio, chuta-se um valor para θ_1 de modo que todos os valores da variável Y (população) atendam a transformação logit, $p/1 - p$, sendo $0 < p < 1$. Olhando para o último dado da população temos, aproximadamente, 282 milhões. Assim, uma escolha para θ_1 é 400 milhões.

```
> # Regressão linear da parte linear do modelo para estimar valores iniciais de theta 2 e theta 3
> start.values <- lm(logit(population/400) ~ year, USPop)
> print(start.values)

Call:
lm(formula = logit(population/400) ~ year, data = USPop)

Coefficients:
(Intercept)         year
  -49.24991      0.02507
```

Os valores iniciais são $\theta_1 = 400$, $\theta_2 = -49$ e $\theta_3 = 0.025$. Logo, podemos iniciar a regressão não-linear.

```
> pop.mod <- nls(population ~ theta1 /(1 + exp(-(theta2 + theta3 * year))),
+                start = list(theta1=400, theta2=-49, theta3=0.025), data = USPop,
+                trace = TRUE)

3060.786 :  400.000 -49.000   0.025
558.5357 :  426.06199142 -42.30785623   0.02142146
457.9746 :  438.41471526 -42.83690081   0.02167713
457.8071 :  440.89027810 -42.69866517   0.02160152
457.8056 :  440.81680958 -42.70804961   0.02160649
457.8056 :  440.83444805 -42.70688446   0.02160586
457.8056 :  440.83332801 -42.70697788   0.02160591

> args(nls.control)

function (maxiter = 50, tol = 1e-05, minFactor = 1/1024, printEval = FALSE,
    warnOnly = FALSE)
NULL

> print(pop.mod)

Nonlinear regression model
  model:  population ~ theta1/(1 + exp(-(theta2 + theta3 * year)))
   data:  USPop
   theta1    theta2    theta3
440.83333 -42.70698   0.02161
 residual sum-of-squares: 457.8

Number of iterations to convergence: 6
Achieved convergence tolerance: 1.481e-06

> summary(pop.mod)

Formula: population ~ theta1/(1 + exp(-(theta2 + theta3 * year)))

Parameters:
         Estimate Std. Error t value Pr(>|t|)
theta1 440.833328  35.000136   12.60 1.14e-10 ***
theta2 -42.706978   1.839138  -23.22 2.08e-15 ***
theta3   0.021606   0.001007   21.45 8.87e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 4.909 on 19 degrees of freedom

Number of iterations to convergence: 6
Achieved convergence tolerance: 1.481e-06

> residuals(pop.mod)

 [1] -3.751334 -4.184537 -4.481748 -4.817446 -4.940823 -4.818360 -3.645893 -1.385057
 [9] -1.468529  1.573398  4.202320  5.531866  7.767791  5.820992  5.299447 -5.306593
[17] -7.365064 -1.906063 -1.344812 -1.885636 -3.313655  6.519831
attr(,"label")
[1] "Residuals"

> deltaMethod(pop.mod, "-theta2/theta3")

               Estimate       SE
-theta2/theta3 1976.634 7.555785


> plot(population ~ year, USPop, xlim = c(1790,2100), ylim=c(0,450))
> with(USPop, lines(seq(1790,2100, by = 10), predict(pop.mod,
+      data.frame(year = seq(1790,2100, by = 10))), lwd = 2))
> points(2010, 307, pch = "x", cex = 1.3)
> abline(h = 0, lty = 2)
> abline(h=coef(pop.mod)[1], lty = 2)
> abline(h=0.5*coef(pop.mod)[1],lty=2)
> abline(v=-coef(pop.mod)[2]/coef(pop.mod)[3], lty = 2)
> PropEsp <-predict(pop.mod,newdata=list(year=seq(1790,2000,by=10)),se.fit=T)
> lines(PropEsp~seq(1790,2000,by=10),lwd=2,col=6)
> upIC<-PropEsp+qt(.975,summary(pop.mod)$df[2])*summary(pop.mod)$sigma
> loIC<-PropEsp-qt(.975,summary(pop.mod)$df[2])*summary(pop.mod)$sigma
> lines(upIC~seq(1790,2000,by=10),col=4)
> lines(loIC~seq(1790,2000,by=10),col=4)
```

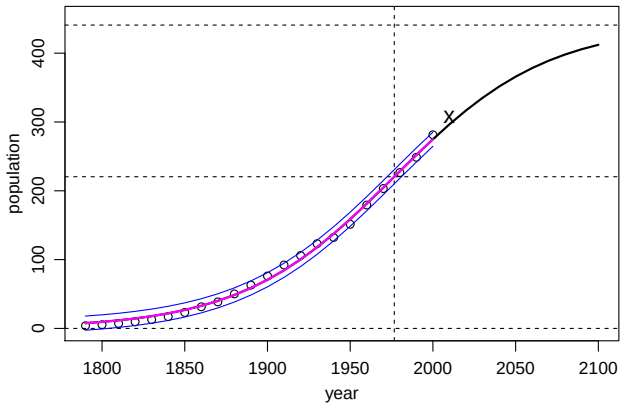

FIGURA 10: Ajuste do modelo não-linear aos dados.

FIGURA 11: Gráfico dos Resíduos da regressão não-linear dos dados.

Assim, como no gráfico de resíduos do modelo polinomial, o gráfico de resíduos da regressão não-linear não apresentou parte sistemática.

Se compararmos as somas dos erros residuais dos três modelos ajustados, verifica-se que o modelo polinomial apresentou melhor ajuste, porém seus parâmetros não fornecem explicação física sobre o crescimento populacional (crescimento da parábola para o infinito positivo). Embora a soma dos quadrados residuais do modelo não-linear ter sido um pouco maior do que o resultante do modelo polinomial, os parâmetros do modelo não-linear subsidiam um modelo físico para explicar o crescimento populacional (a taxa de crescimento é a mesma em toda a curva).

1.2 Modelos Auto-Inicializáveis (Self-Starting Models)

O procedimento anterior é uma possibilidade para dar início ao algoritmo de regressão não-linear. Outra forma de inicializar o modelo é utilizando funções do `R` que fornecem automaticamente valores iniciais.

Considerando o modelo de crescimento logístico, o software se baseia em uma equação equivalente à equação (6), veja:

$$Y = \frac{\theta_1}{1 + e^{(-\theta_2 - \theta_3 X)\frac{\theta_3}{\theta_3}}} + \epsilon \tag{10}$$

$$Y = \frac{\theta_1}{1 + e^{\left(-\frac{\theta_2 - \theta_3 X}{\theta_3}\right)\theta_3}} + \epsilon \tag{11}$$

$$Y = \frac{\theta_1}{1 + e^{\frac{-\frac{\theta_2}{\theta_3} - X}{\frac{1}{\theta_3}}}} + \epsilon \tag{12}$$

$$\tag{13}$$

chamando $\phi_1 = \theta_1$, $\phi_2 = -\frac{\theta_2}{\theta_3}$ e $\phi_3 = \frac{1}{\theta_3}$, temos:

$$Y = \frac{\phi_1}{1 + e^{-\left(\frac{X - \phi_2}{\phi_3}\right)}} + \epsilon \tag{14}$$

Na parametrização ϕ, ϕ_1 é assintótico superior, ϕ_2 é o valor de X para o qual a resposta é metade do valor assintótico, e ϕ_3 é a taxa do parâmetro. Como os parâmetros ϕ são um a um transformações não-lineares dos parâmetros θ, os dois modelos nos fornecem o mesmo ajuste para os dados.

Segue o um exemplo utilizando o *self-starting* do `R`:

```
> # Self-Starting Models
>
```

```
> pop.ss <- nls(population ~ SSlogis(year, phi1, phi2, phi3), data = USPop)
> summary(pop.ss)

Formula: population ~ SSlogis(year, phi1, phi2, phi3)

Parameters:
     Estimate Std. Error t value Pr(>|t|)
phi1  440.834     35.000   12.60 1.14e-10 ***
phi2 1976.634      7.556  261.61  < 2e-16 ***
phi3   46.284      2.157   21.45 8.87e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.909 on 19 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 3.822e-06

> deltaMethod(pop.mod, "1/theta3")

         Estimate       SE
1/theta3 46.28363 2.157445
```

1.3 Reparametrização Linear

Em alguns problemas, pode ser usual transformar linearmente as variáveis preditoras para tornar os resultados dos parâmetros mais significativos. Nos dados da população dos U.S.A, podemos considerar a mudança do preditor `year` por `decade = (year - 1790)/10`.

```
> USPop$decade <- (USPop$year - 1790)/10
> (pop.ss.rescaled <- nls(population ~ SSlogis(decade, nu1, nu2, nu3), data=USPop))

Nonlinear regression model
  model:  population ~ SSlogis(decade, nu1, nu2, nu3)
   data:  USPop
    nu1     nu2     nu3
440.834  18.663   4.628
 residual sum-of-squares: 457.8

Number of iterations to convergence: 0
Achieved convergence tolerance: 3.814e-06
```

O parâmetro assintótico estimado é o mesmo neste modelo. O tempo para "half-asymptote"é agora medido em décadas, então $18,66 \cdot 10 + 1790 = 1977$ como antes. A taxa por década é 1/10 da taxa por ano. Os resumos, como a estimativa de σ são idênticos em ambos os ajustes.

Mudanças de escalas como essa podem ser úteis para evitar problemas computacionais, ou para nos dar parâmetros que correspondem à unidades de interesse, não afetando o modelo ajustado.

1.4 Ajuste de modelos não lineares com um fator

Um problema comum com modelos não lineares é quando gostaríamos de ajustar o modelo com a mesma função média para cada um dos diversos grupos de dados. Por exemplo, os dados `CanPop` do pacote `car` tem os dados da população canadense com o mesmo formato dos dados dos U.S.A. Vamos combinar as duas tabelas em uma, e vamos plotar o gráfico para ambos os países.

```
> Data <- data.frame(rbind(data.frame(country="US",USPop[,1:2]),
+                          data.frame(country="Canada", CanPop)))
> some(Data)

    country year population
2        US 1800   5.308483
8        US 1860  31.443321
18       US 1960 179.323175
20       US 1980 226.542199
22       US 2000 281.421906
51   Canada 1891   4.833000
91   Canada 1931  10.377000
101  Canada 1941  11.507000
121  Canada 1961  17.780000
141  Canada 1981  23.774000
```

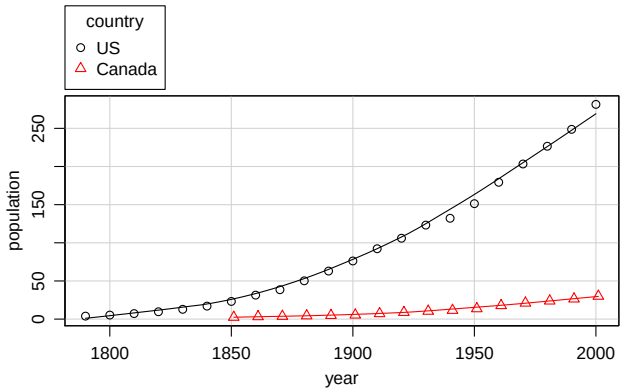

FIGURA 12: Gráfico dos Resíduos da regressão não-linear dos dados.

Vamos realizar o ajuste para ambos os dados.

```
> library(nlme)
> m.list <- nlsList(population ~ SSlogis(year, phi1, phi2, phi3)|country,
+                   pool = FALSE, data = Data)
> summary(m.list)

Call:
  Model: population ~ SSlogis(year, phi1, phi2, phi3) | country
   Data: Data

Coefficients:
   phi1
         Estimate Std. Error   t value     Pr(>|t|)
US     440.83357   35.00023 12.595163 1.13903e-10
Canada  71.44637   14.15008  5.049186 2.22768e-04
   phi2
         Estimate Std. Error  t value     Pr(>|t|)
US     1976.634   7.555803 261.6048 2.942066e-35
Canada 2015.663  16.474723 122.3488 2.730058e-21
   phi3
         Estimate Std. Error  t value     Pr(>|t|)
```

```
US     46.28366   2.157448 21.45297 8.867045e-15
Canada 47.74810   3.060072 15.60359 8.477325e-10
```

```
> (sds <- sapply(m.list, sigmaHat))
```

```
       US    Canada
4.9086692 0.5671285
```

Os parâmetros são diferentes para os dois países, onde "half-asymptote" do Canadá é atingida 40 anos depois em relação aos U.S.A. e as estimativas do desvio padrão residual são muito diferentes.

2 Músculos do coração de ratos

O objetivo de um experimento era o avaliar a influência de uma solução de cálcio na contração dos músculos do coração de ratos.

A aurícula direita de 21 corações de ratos foram isoladas. Em várias ocasiões foram estimulados eletricamente e mergulhados em várias concentrações de solução de cloreto de cálcio, e foram medidos após o encurtamento.

O conjunto de dados `muscle` em `MASS` contém informações dos dados da variáveis `Strip`, `Conc` e `Lenght`.

O modelo particular colocado pelos autores tem a forma

$$log\ Y_{ij} \quad = \quad \alpha_j + \beta\rho^{X_{ij}} + \epsilon_{ij} \tag{15}$$

onde i refere-se a concentração e j a tira do músculo. Este modelo tem 1 parâmetro não-linear e 22 parâmetros lineares. Foi tomado como estimativa inicial para $\rho = 0,1$. O primeiro passo é construir a matriz para selecionar α apropriado.

```
> data(muscle)
> head(muscle)
```

```
   Strip Conc Length
3    S01    1   15.8
4    S01    2   20.8
5    S01    3   22.6
6    S01    4   23.8
9    S02    1   20.6
10   S02    2   26.8
```

```
> dim(muscle)
```

```
[1] 60  3
```

```
> # taking advantage of linear parameters
> library(MASS)
> A <- model.matrix(~ Strip - 1, data = muscle)
> rats.nls1 <- nls(log(Length) ~ cbind(A, rho^Conc),
+                  data = muscle, start = c(rho = 0.1),
+                  algorithm = "plinear")
> (B <- coef(rats.nls1))
```

```
          rho .lin.StripS01 .lin.StripS02 .lin.StripS03 .lin.StripS04 .lin.StripS05
   0.07776401    3.08304824    3.30137838    3.44562531    2.80464433    2.60835015
.lin.StripS06 .lin.StripS07 .lin.StripS08 .lin.StripS09 .lin.StripS10 .lin.StripS11
   3.03357724    3.52301734    3.38711844    3.46709395    3.81438456    3.73878664
```

```
.lin.StripS12 .lin.StripS13 .lin.StripS14 .lin.StripS15 .lin.StripS16 .lin.StripS17
   3.51332580    3.39741114    3.47088607    3.72895847    3.31863862    3.37938672
.lin.StripS18 .lin.StripS19 .lin.StripS20 .lin.StripS21        .lin22
   2.96452195    3.58468686    3.39628029    3.36998871   -2.96015461
```

Nós podemos utilizar este vetor de coeficientes como um valor de início para o ajuste utilizando o algoritmo convencional.

```
> st <- list(alpha = B[2:22], beta = B[23], rho = B[1])
> rats.nls2 <- nls(log(Length) ~ alpha[Strip] + beta * rho ^ Conc,
+                  data = muscle, start = st)
```

Observe que se um parâmetro na regressão não-linear é indexado, tal como $alpha[Strip]$, os valores de início podem ser fornecidos como uma lista nomeada componentes separaveis.

Agora vamos mostrar os dados e predizer valores em uma janela de Trellis.

```
> attach(muscle)
> Muscle <- expand.grid(Conc = sort(unique(Conc)), Strip = levels(Strip))
> Muscle$Yhat <- predict(rats.nls2, Muscle)
> Muscle$logLength <- rep(NA, nrow(Muscle))
> ind <- match(paste(Strip, Conc), paste(Muscle$Strip, Muscle$Conc))
> Muscle$logLength[ind] <- log(Length)
> detach()
```

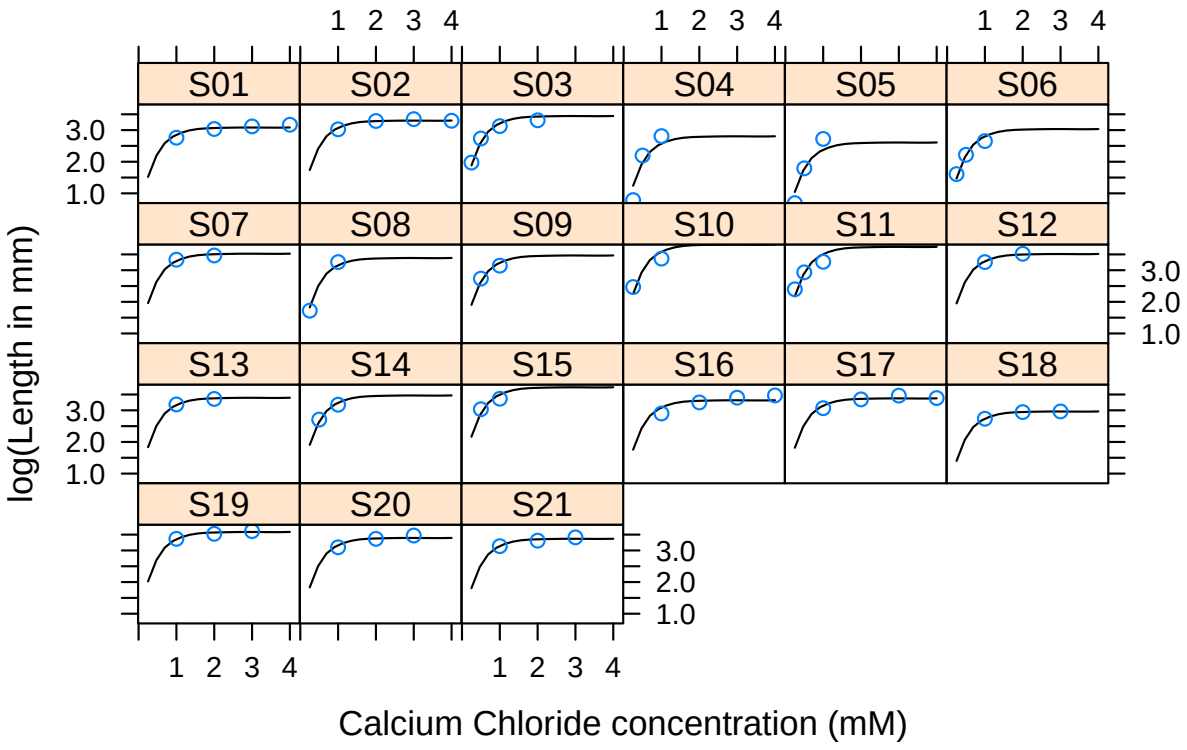

FIGURA 13: Gráfico dos Resíduos da regressão não-linear dos dados.

O modelo parece descrever a situação muito bem, mas para algumas finalidades um modelo não-linear de efeitos mistos pode ser mais apropriado.

3 Viscosímetro

Um viscosímetro Stormer mede a viscosidade de um fluido pela medida do tempo levado para que um cilindro interno no mecanismo execute um número fixo de revoluções em resposta ao acionamento de um peso. O viscosímetro é calibrado pela medição do tempo tomado com variação de pesos enquanto o mecanismo é suspenso em fluidos de viscosidades acuradas. O conjunto de dados vem de um calibrador, e considerações teóricas sugerem uma relação não-linear entre o tempo T, peso w e viscosidade v da forma

$$T = \frac{\beta_1 v}{w - \beta_2} + \epsilon$$

onde β_1 e β_2 são parâmetros desconhecidos a serem estimados. Observe que β_1 é um parâmetro linear e β_2 é não-linear.

```
> data(stormer)
> str(stormer)

'data.frame':        23 obs. of  3 variables:
 $ Viscosity: num  14.7 27.5 42 75.7 89.7 ...
 $ Wt       : int  20 20 20 20 20 20 20 50 50 50 ...
 $ Time     : num  35.6 54.3 75.6 121.2 150.8 ...

> head(stormer)

  Viscosity Wt  Time
1      14.7 20  35.6
2      27.5 20  54.3
3      42.0 20  75.6
4      75.7 20 121.2
5      89.7 20 150.8
6     146.6 20 229.0

> dim(stormer)

[1] 23  3
```

O livro S sugere que valores adequados podem ser obtidos escrevendo o modelo de regressão na forma:

$$wT = \beta_1 v + \beta_2 T + (w - \beta_2)\epsilon$$

Usando regressão linear, vamos encontrar valores iniciais para β_1 e β_2 e, em seguida realizaremos a regressão não-linear:

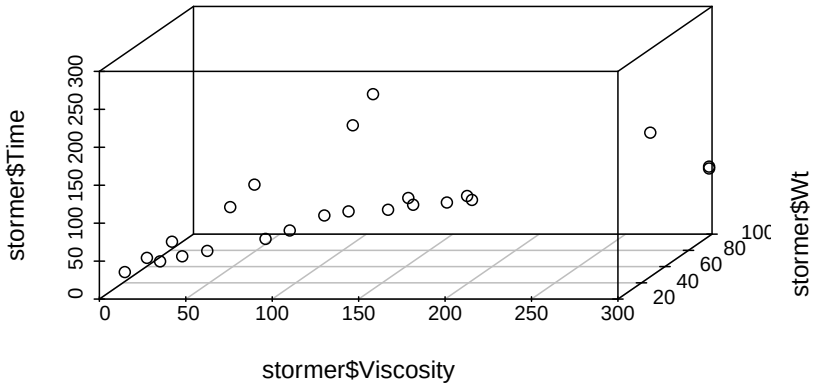

FIGURA 14: Gráfico dos dados Stormer.

```
> fm0 <- lm(Wt*Time ~ Viscosity + Time - 1, data = stormer)
> b0 <- coef(fm0); names(b0) <- c("b1", "b2"); b0

       b1        b2
28.875541  2.843728

> storm.fm <- nls(Time ~ b1 * Viscosity/(Wt-b2), data = stormer, start = b0, trace = T)

885.3645 :  28.875541  2.843728
825.1098 :  29.393464  2.233276
825.0514 :  29.401327  2.218226
825.0514 :  29.401257  2.218274
```

Desde que existam apenas 2 parâmetros, podemos exibir uma região de confiança para os parâmetros de regressão com um mapa de contorno:

```
> bc <- coef(storm.fm) # recebendo os parâmetros b1 e b2
> se <- sqrt(diag(vcov(storm.fm))) # raiz quad. dos elementos da diag. da mat. de cov.
> dv <- deviance(storm.fm) # desvio do modelo ajustado
```

Definindo $d(\beta_1, \beta_2)$ como a função soma dos quadrados:

$$d(\beta_1, \beta_2) = \sum_{i=1}^{23} \left(T_i - \frac{\beta_1 v_i}{w_i - \beta_2} \right)^2$$

Então, `dv` contém o valor mínimo, $d_0 = d(\hat{\beta}_1, \hat{\beta}_2)$, a soma dos quadrados residuais ou modelo de desvio.

Se β_1 e β_2 são os verdadeiros valores dos parâmetros, a estatística da "soma dos quadrados adicionais" é dada por

$$F(\beta_1, \beta_2) = \frac{(d(\beta_1, \beta_2) - d_0)/2}{d_0/21}$$

é aproximadamente distribuído por $F_{2,21}$. Uma aproximação do conjunto de confiança contém os valores no espaço de parâmetros para F_{β_1,β_2} em menos de 95% dos pontos da distribuição $F_{2,21}$. Vamos construir o gráfico de contorno da função F_{β_1,β_2} e marcar a região de confiança, conforme figura 17.

Uma região adequada para representar os contornos é 3 erros padrões de cada lado dos estimadores de mínimos quadrados em cada parâmetro. Uma vez que esses intervalos são iguais em suas respectivas unidades de erro padrão é útil para fazer a plotagem da região quadrática.

```
> par(pty="s")
> b1 <- bc[1] + seq(-3*se[1], 3*se[1], length=51)
> b2 <- bc[2] + seq(-3*se[2], 3*se[2], length=51)
> bv <- expand.grid(b1,b2)
```

O caminho mais simples de calcular a função soma dos quadrados é usar a função `apply`:

```
> Wt <- as.numeric(stormer$Wt)
> ssq <- function(b)  sum((stormer$Time - b[1] * stormer$Viscosity/(stormer$Wt-b[2]))^2)
> dbetas <- apply(bv, 1, ssq)
> cc <- matrix(as.vector(stormer$Time) - rep(bv[,1], rep(23,2601))*as.vector(stormer$Viscosity)/
+              (as.vector(stormer$Wt)-rep(bv[,2], rep(23,2601))), 23)
> dbetas <- matrix(drop(rep(1,23) %*% cc^2),51)
```

Calculando a estatística F:

```
> fstat <- matrix(((dbetas - dv)/2)/(dv/21), 51,51)
```

Agora, podemos produzir um mapa de contorno para estatística F, tomando cuidado para que os contornos ocorram em níveis relativamente interessantes da superfície. Note que o contorno da região de confiança é em torno de 3,5:

```
> qf(0.95, 2,21)

[1] 3.4668
```

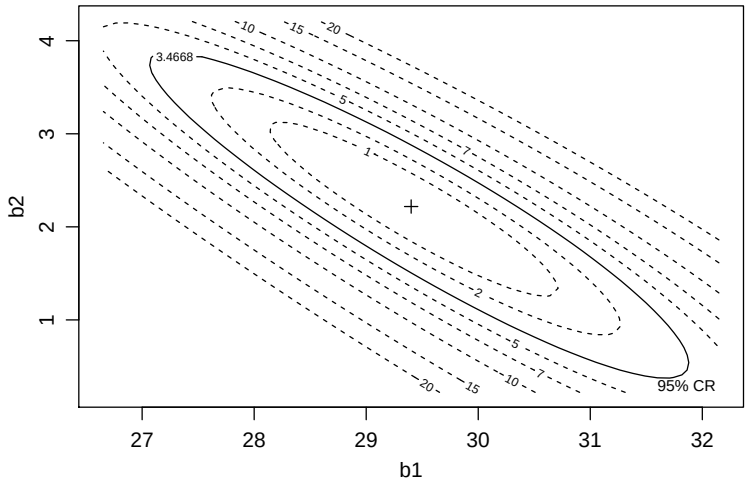

FIGURA 15: Superfície da estatística F e região de confiança dos parâmetros de regressão.

A função de máxima verossimilhança tem os mesmos contornos como os da estatística F, uma forma elíptica dos contornos é uma indicação de que a aproximação teórica se aproximou sobre uma regressão da distribuição normal, embora mais do que isso seja necessário para se confiar nesse resultado. Dada a maneira como as escalas do eixo foram escolhidas, a forma alongada do contorno mostra que as estimativas de $\hat{\beta}_1$ e $\hat{\beta}_2$ são altamente correlacionadas (negativamente).

4 Crescimento de Perus

Um experimento foi conduzido para estudar os efeitos de diferentes quantidades de metionina (A) sobre o crescimento de perus, controlando a variação do suplemento da mesma desde quantidade 0 até 0,44% da dieta total. A unidade experimental foram as penas de peruns jovens, e o tratamento foi atribuído aleatoriamente, onde foram recolhidas 10 penas que não tinham suplementação, e 5 penas que receberam cada uma 5 quantidades usadas no experimento, totalizando 35 penas. O peso médio das penas dos perus foi obtido no início e no final do experimento após 3 semanas. A variável resposta é o ganho de gramas `Gain` por peru, em cada pena. O objetivo do experimento é entender quanto é esperado de peso com a variação de metionina, ou seja, $E(Gain|A)$.

```
> library(alr3)
> some(turk0)

      A Gain
1  0.00  644
2  0.00  631
4  0.00  624
16 0.10  730
24 0.16  760
25 0.16  727
26 0.28  809
28 0.28  763
30 0.28  811
33 0.44  799

> dim(turk0)

[1] 35  2
```

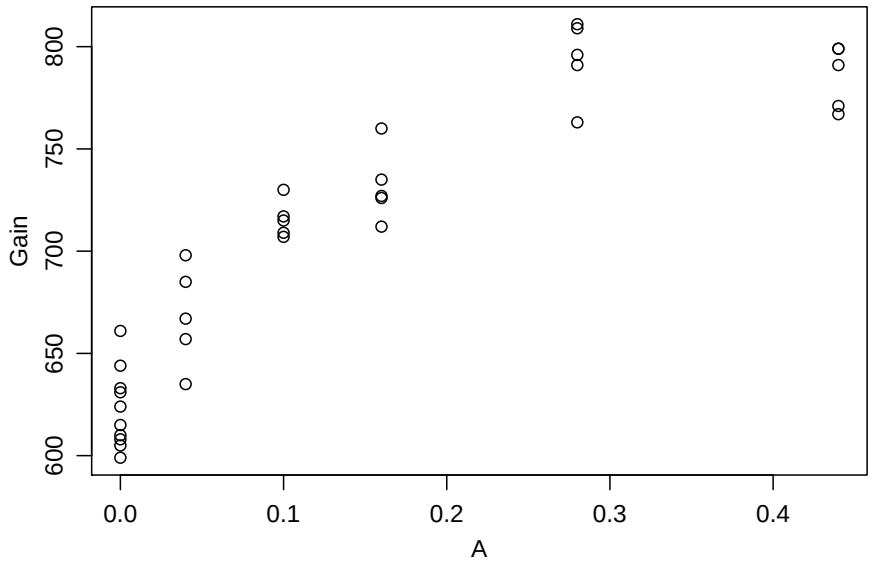

FIGURA 16: Representação gráfica dos dados de turk0.

Este gráfico mostra o crescimento do peso das penas com o aumento da quantidade de metionina na alimentação dos perus. Observa-se que existem variações pena-a-pena, refletida pela variabilidade entre a repetição das observações com as mesmas quantidades de metionina (`A`). A função média certamente não será será uma reta, uma vez que a diferença nas médias quando $\texttt{A} < 0.3$ é muito pequena em relação a diferença

nas médias quando A > 0.2. Um polinômio de grau 2 ou 3 poderia ser utilizado para ajustar aos 6 pontos médios de A, não saindo do intervalo de A e os parâmetros poderiam ter pouca explicação física. Assim, um ajuste não linear é preferível para este problema.

Para o crescimento como uma função de um aminoácido, a função média é

$$E(Gain|A) = \theta_1 + \theta_2 \left(1 - e^{-\theta_3 A}\right) \tag{16}$$

Para estimar os parâmetros na equação 16, precisamos dos valores iniciais para θ. Observa-se que o valor interceptor é próximo de 620 e o valor assintótico é em torno de 850. Assim, pode-se considerar como valores iniciais $\theta_1 = 620$ e $\theta_2 = 850 - 620 = 230$. Para encontrar θ_3, podemos tentar a linearização da equação 16.

$$\frac{(\theta_1 + \theta_2) - y_i}{\theta_2} = e^{-\theta_3 A} \tag{17}$$

$$-log\left(\frac{(\theta_1 + \theta_2) - y_i}{\theta_2}\right) = \theta_3 A \tag{18}$$

```
> turk.par <- lm( -log((620 + 230)/ 230 - turk0$Gain/230) ~ turk0$A - 1 )

> turk.nls <- nls(Gain ~ (theta1 + theta2*(1-exp(-theta3*A))),
+                 data = turk0, start = list(theta1=620, theta2=180,
+                                            theta3=3.714), trace = TRUE)

44216.45 :  620.000 180.000   3.714
35609 :  622.978007 131.003167   7.741243
12476.3 :  623.071100 178.201417   6.824178
12367.73 :  622.91140 177.99674   7.14681
12367.42 :  622.962130 178.263470   7.120128
12367.42 :  622.957684 178.250788   7.122396
12367.42 :  622.958061 178.251928   7.122206

> print(turk.nls)

Nonlinear regression model
  model:  Gain ~ (theta1 + theta2 * (1 - exp(-theta3 * A)))
   data:  turk0
 theta1  theta2  theta3
622.958 178.252   7.122
 residual sum-of-squares: 12367

Number of iterations to convergence: 6
Achieved convergence tolerance: 2.354e-06

> summary(turk.nls)

Formula: Gain ~ (theta1 + theta2 * (1 - exp(-theta3 * A)))

Parameters:
       Estimate Std. Error t value Pr(>|t|)
theta1  622.958      5.901  105.57  < 2e-16 ***
theta2  178.252     11.636   15.32 2.74e-16 ***
theta3    7.122      1.205    5.91 1.41e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 19.66 on 32 degrees of freedom

Number of iterations to convergence: 6
Achieved convergence tolerance: 2.354e-06
```

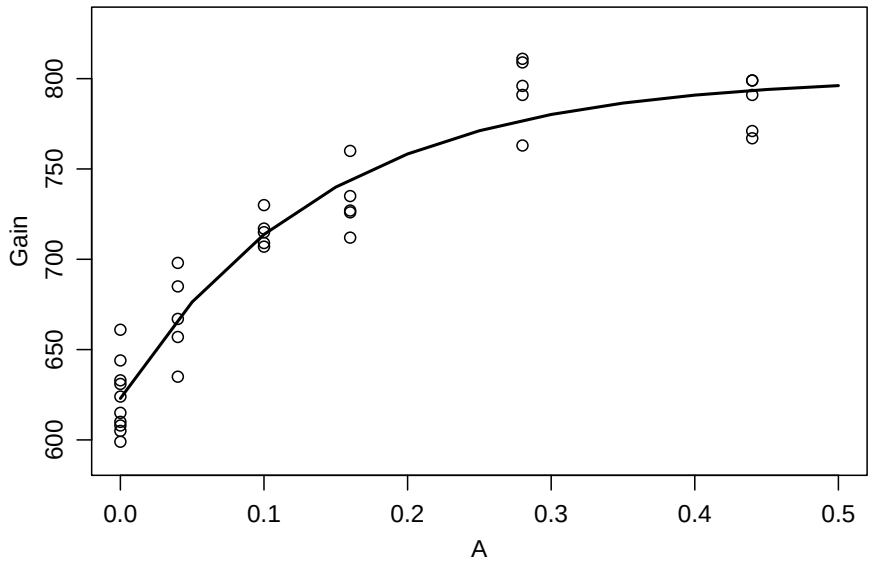

FIGURA 17: Ajuste da regressão não linear aos dados turk0.

Referências

[1] J. A. Achcar and J. Mazuchelli. Algumas considerações em regressão não-linear., 2002. [Onlineand acesso em 28-Agosto-2012].

[2] W. O. Bussab and P. A. Moretin. *Estatística básica.* Saraiva, São Paulo, 5.ed. edition, 2002.

[3] P. L. S. Carneiro, C. H. M. Malhado, J. A. Muniz, F. F. e Silva, and F. G. da Silveira. Análise de agrupamento na seleção de modelos de regressão não-lineares para curvas de crescimento de ovinos cruzados., 2011. [Onlineand acesso em 28-Agosto-2012].

[4] N. R. Draper and H. Smith. *Applied Regression Analysis.* Wiley-Interscience, New York, 3.ed. edition, 1998.

[5] E. Esteves. Apresentação do r com um exemplo de análise de regressão não-linear. [Onlineand acesso em 28-Agosto-2012].

[6] J. Fox and S. Weisberg. *An R Companion to Applied Regression.* SAGE, Los Angeles, 2.ed. edition, 2011.

[7] D. C. Montgomery, E. A. Peck, and G. G. Vining. *Introduction to Linear Regression Analyses.* Wiley-Interscience, New York, 4.ed. edition, 2006.