Regressão Não Linear

Modelos de Regressão

August 17, 2024

Conceito

"Qualquer modelo que não seja linear nos parâmetros desconhecidos é um ***modelo de regressão não linear****"* (Montgomery, 2006).

Conceito

Neste caso, pode-se seguir três caminhos diferentes:

1. linearizar a relação transformando os dados;
2. ajustar modelos polinomiais ou splines complexos aos dados;
3. **ajustar funções não lineares aos dados.**

Exemplo

O modelo

$$\mathbf{y} = \theta_1 e^{\theta_2 x} + \epsilon$$

é não linear nos parâmetros desconhecidos θ_1 e θ_2. Podendo ser escrito também como

$$\mathbf{y} = f(\mathbf{x}, \boldsymbol{\theta}) + \epsilon$$

Conceito

Nos modelos de regressão não linear pelo menos uma das derivadas da função esperança $f(\mathbf{x}, \boldsymbol{\theta})$ depende de um dos parâmetros. Na regressão linear essas derivadas não são função de tais parâmetros desconhecidos.

Regressão Linear

Modelo:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \epsilon$$

Função esperança:

$$f(\mathbf{x}, \boldsymbol{\beta})$$

Derivada: $\frac{\partial f(\mathbf{x}, \boldsymbol{\beta})}{\partial \beta_j} = x_j$

Regressão Não Linear

Modelo:

$$y = \theta_1 e^{\theta_2 x} + \epsilon$$

Função esperança:

$$f(\mathbf{x}, \boldsymbol{\theta})$$

Derivadas:
$\frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \theta_1} = e^{\theta_2 x}$; $\frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \theta_2} = \theta_1 x e^{\theta_2 x}$

Vantagens

De maneira resumida, os modelos não lineares (MNL) têm as seguintes vantagens sobre os modelos lineares (ML):

1. Sua escolha têm sustentação baseada em teoria ou princípios mecanísticos (físicos, químicos ou biológicos) ou qualquer outra informação prévia;
2. Certos parâmetros são quantidade de interesse para o pesquisador providos de interpretação;
3. São parcimoniosos pois tipicamente possuem menos parâmetros;
4. Partem do conhecimento do pesquisador sobre o fenômeno alvo.

Desvantagens

1. Requerem procedimentos iterativos de estimação baseados no fornecimento de valores iniciais para os parâmetros;
2. Métodos de inferência são aproximados;
3. Exigem conhecimento do pesquisador sobre o fenômeno alvo.

Modelos de Regressão Não Linear: nota

Idealmente um modelo de regressão não linear é escolhido com base em considerações teóricas sobre o problema, o que às vezes faz com que eles sejam usados em situações muito específicas.

Nesse caso, as aplicações mais comuns estão relacionadas a **modelos de crescimento**, tipicamente na área da **biologia** onde plantas e organismo crescem com o tempo.

Estudo de caso: uma curva de degradação

Um solo foi enriquecido com o herbicida metramitron até a concentração de 100 ng/g. Foi então colocado em 24 recipientes de alumínio, dentro de uma câmara climática a 20°C. Foram selecionados 3 recipientes aleatoriamente em 8 momentos diferentes e a concentração residual de metamitron foi medida.

Dados

```
x <- "https://www.casaonofri.it/_datasets/degradation.csv"
dataset <- read.csv(x, header=T)
head(dataset, 10)
##    Time   Conc
## 1     0  96.40
## 2    10  46.30
## 3    20  21.20
## 4    30  17.89
## 5    40  10.10
## 6    50   6.90
## 7    60   3.50
## 8    70   1.90
## 9     0 102.30
## 10   10  49.20
```

Gráfico de dispersão

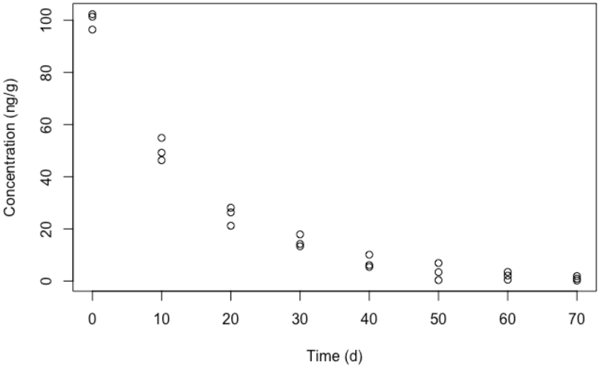


Figure: Degradação do metamitron no solo.

Formas de funções

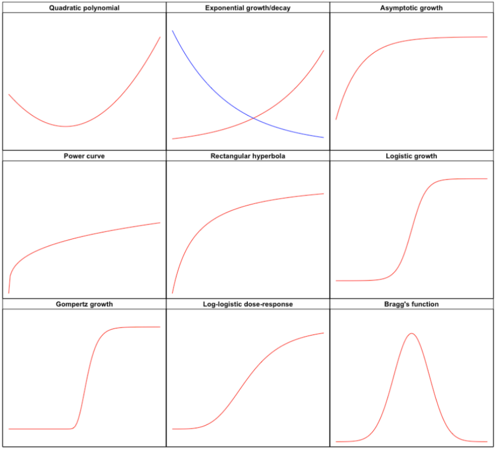


Figure: Formas das funções mais importantes.

Equações por trás das funções

Name	Equation	R function
Straight line	$Y = b_0 + b_1\, X$	`NLS.linear()`
Quadratic polynomial	$Y = b_0 + b_1\, X + b_2\, X^2$	`NLS.poly2()`
Exponential	$Y = a\, e^{kX}$	`NLS.expoGrowth()` `NLS.expoDecay()`
Asymptotic	$Y = a - (a - b)\ \exp(-cX)$	`NLS.asymReg()`
Power	$Y = a\, X^b$	`NLS.powerCurve()`
Logarithmic	$Y = a + b\ \log(X)$	`NLS.logCurve()`
Rectangular hyperbola	$Y = \frac{a\,X}{b+X}$	`SSmicmen()`
Logistic	$Y = c + \frac{d-c}{1+exp(-b(X-e))}$	`NLS.L4()` `NLS.L3()` `NLS.L2()`
Gompertz	$Y = c + (d-c)\exp\{-\exp[-b\,(X-e)]\}$	`NLS.G4()` `NLS.G3()` `NLS.G2()`
Modified Gompertz	$Y = c + (d-c)\,\{1-\exp\{-\exp[b\,(X-e)]\}\}$	`NLS.E4()` `NLS.E3()` `NLS.E2()`
Log-logistic	$Y = c + \frac{d-c}{1+\exp\{-b[\log(X)-\log(e)]\}}$	`NLS.LL4()` `NLS.LL3()` `NLS.LL2()`
Weibull I	$Y = c + (d-c)\exp\{-\exp[-b\,(\log(X)-\log(e))]\}$	`NLS.W1.4()` `NLS.W1.3()` `NLS.W1.2()`
Weibull II	$Y = c + (d-c)\,\{1-\exp\{-\exp[b\,(\log(X)-\log(e))]\}\}$	`NLS.W2.4()` `NLS.W2.3()` `NLS.W2.2()`
Bragg	$Y = c + (d-c)\ \exp\left[-b(X-e)^2\right]$	`NLS.bragg.3()` `NLS.bragg.4()`
Lorentz	$Y = c + \frac{d-c}{1+b(X-e)^2}$	`NLS.lorentz.3()` `NLS.lorentz.4()`
Beta	$Y = d\left\{\left(\frac{X-X_b}{X_o-X_b}\right)\left(\frac{X_c-X}{X_c-X_o}\right)^{\frac{X_c-X_o}{X_o-X_b}}\right\}^b$	`NLS.beta()`

Figure: Equações e funções no R.

Modelo

$$Y_i = ae^{-kX_i} + \epsilon_i$$

Onde
Y é a **concentração no tempo** X
a é a **concentração inicial de metamitrona**
k é a t**axa de degradação constante**.

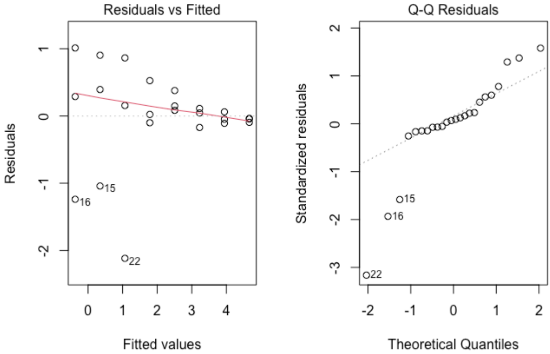
Estimação dos parâmetros: linearização

```
mod <- lm(log(Conc) ~ Time, data=dataset)
summary(mod)
## Call:
## lm(formula = log(Conc) ~ Time, data = dataset)
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.11738 -0.09583  0.05336  0.31166  1.01243

## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.662874   0.257325   18.12 1.04e-14 ***
## Time        -0.071906   0.006151  -11.69 6.56e-11 ***
## Residual standard error: 0.6905 on 22 degrees of freedom
## Multiple R-squared:  0.8613, Adjusted R-squared:  0.855
## F-statistic: 136.6 on 1 and 22 DF,  p-value: 6.564e-11
```

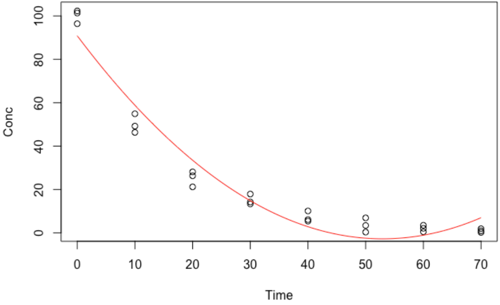
Estimação dos parâmetros: linearização

```
par(mfrow = c(1,2))
plot(mod, which = 1)
plot(mod, which = 2)
```



Estimação dos parâmetros: aproximação polinomial

```
mod2 <- lm(Conc ~ Time + I(Time^2), data=dataset)
pred <- predict(mod2, newdata = data.frame(Time = seq(0, 70
plot(Conc ~ Time, data=dataset)
lines(pred ~ seq(0, 70, by = 0.1), col = "red")
```



Estimação dos parâmetros: mínimos quadrados não lineares

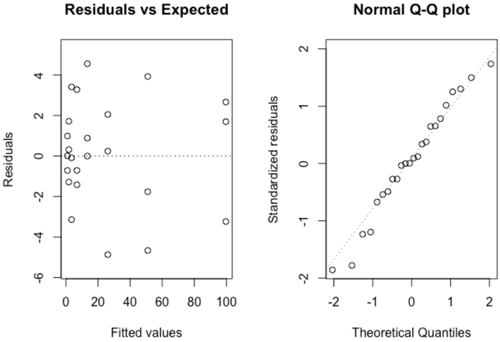
```
modNlin <- nls(Conc ~ A*exp(-k*Time),
               start=list(A=100, k=0.05),
               data = dataset)
summary(modNlin)

## Formula: Conc ~ A * exp(-k * Time)

## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## A 99.634902   1.461047   68.19   <2e-16 ***
## k  0.067039   0.001887   35.53   <2e-16 ***
##
## Residual standard error: 2.621 on 22 degrees of freedom
##
## Number of iterations to convergence: 5
## Achieved convergence tolerance: 4.33e-07
```

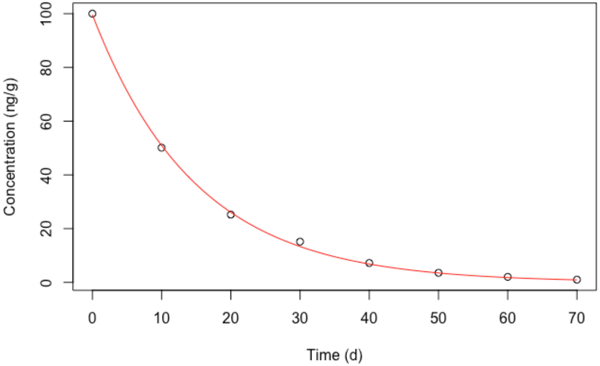
Ajuste do modelo

```r
par(mfrow=c(1,2))
plot(modNlin, which = 1)
plot(modNlin, which = 2)
```



Ajuste do modelo

```
plot(modNlin, type = "means",
        xlab = "Time(d)", ylab = "Concentration(ng/g)")
```



Medida de qualidade do ajuste

```
MSE <- summary(modNlin)$sigma ^ 2
MST <- var(dataset$Conc)
1 - MSE/MST
## [1] 0.9936359
```

Predição

```
func <- list(~A * exp(-k * time))
const <- data.frame(time = c(5, 10, 15))
gnlht(modNlin, func,  const)
##             Form          time Estimate      p-value
## 1 A * exp(-k * time)     5 71.25873  5.340741e-28
## 2 A * exp(-k * time)    10 50.96413  3.163518e-25
## 3 A * exp(-k * time)    15 36.44947  6.029672e-22
```

Referências

MONTGOMERY, Douglas C.; PECK, Elizabeth A.; VINING, G. Geoffrey. **Introduction to Linear Regression Analysis**. 4ª ed. Hoboken: John Wiley Sons, 2006.

NABERGOI, Carlos. **Curso de Modelos Não Lineares em R**. Disponível em: https://www.ime.unicamp.br/ cnaber/cursomodelosnaolinearesR.pdf. Acesso em: 21 jul. 2024.

STATFORBIOLOGY.COM. **Nonlinear Regression**. Disponível em: https://www.statforbiology.com/-statbookeng/nonlinear-regression. Acesso em: 21 jul. 2024.

Obrigada! :)

```
print("sextou")
```