

- [Tutorial](#)
- [Exercícios](#)
- [Apostila](#)

7. Modelos Lineares

São chamados modelos lineares aqueles que apresentam uma relação entre variáveis que seja **linear nos parâmetros**. Essa linearidade implica que **matematicamente** a variação de cada um dos parâmetros é independente dos demais parâmetros do modelo.

Em termos gerais, podemos reconhecer dois grandes grupos clássicos de modelos lineares:

- **Modelos de Regressão.**
- **Modelos de Análise de Variância.**

Nesse tópico utilizaremos os arquivos de dados:

- [Levantamento em caixetais: caixeta.csv](#) (apagar extensão .pdf)
- [Dados de biomassa de árvores: esaligna.csv](#) (apagar extensão .pdf)
- [Inventário em florestas plantadas: egrandis.csv](#) (apagar extensão .pdf)
- [Experimento sobre o crescimento de mudas em viveiro: altura-mudas.csv](#) (apagar extensão .pdf)

Regressão Linear

Os modelos lineares de regressão são utilizados para modelar a relação entre variáveis quantitativas:

- **Variável Resposta** (y): variável quantitativa (também chamada de variável dependente).
- **Variáveis Predictoras** (x): variáveis quantitativas (também chamadas de variáveis independentes).

A função 'lm'

A função utilizada para construir modelos lineares de regressão é a função 'lm' que tem os seguintes argumentos principais:

```
lm( formula, data, weights, subset, na.action )
```

- 'formula' - é uma **fórmula estatística** que indica o modelo a ser ajustado. Possui a mesma forma básica que foi vista na funções gráficas.
- 'data' - o conjunto de dados (`data.frame`).
- 'weights' - são os *pesos* para regressão ponderada.
- 'subset' - um vetor com as condições que definem um **sub-conjunto** dos dados.
- 'na.action' - função que especifica o que fazer no caso de observações perdidas (NA). O valor default é 'na.omit' que elimina as linhas (observações) que possuem observações perdidas nas variáveis definidas na fórmula.

Vejamos um exemplo simples:

```
> egr = read.csv("egrandis.csv",header=T)
> egr[1,]
  especie rot   regioao inv faz proj talhao parcela arv fuste cap ht hdom
1 E.grandis 1 Botucatu 1995 36 33      1      1 1      1 57 27  NA
  idade carac      dap
1 7.49315      N 18.14366
>
> hipsol = lm( ht ~ dap, data=egr )
> class(hipsol)
[1] "lm"
>
```

Uma Palavra sobre o Argumento 'formula'

O argumento fórmula nos modelos lineares é bastante diverso das fórmulas matemáticas usuais. Nesse argumento, sinais de mais e de menos, símbolos como circunflexo (^) e asterisco (*) têm significado bastante diferente dos significados usuais matemáticos.

Apresentaremos agora alguns aspectos básicos do argumento:

- ' $y \sim x$ ' indica: *modele y como **função estatística** de x*;
- ' $y \sim x1 + x2$ ' indica: *modele y como **função estatística** das variáveis x1 e x2 (efeito aditivo dos modelos lineares)*;

Se quisermos utilizar os símbolos matemáticos no sentido matemático usual **dentro** de uma fórmula estatística, temos que utilizar a função ' $I()$ ':

- ' $y \sim I(x1^2 * x2^3)$ ' indica: *modele y como função estatística **da variável** ($x1^2 * x2^3$)*;
- ' $y \sim I(x1 / x2)$ ' indica: *modele y como função estatística **da variável** ($x1/x2$)*;

No caso de utilizarmos **funções matemáticas** específicas a função ' $I()$ ' torna-se desnecessária:

- ' $\log(y) \sim \log(x)$ ' indica: *modele o **log(y)** com função estatística da variável **log(x)***;
- ' $\log(y) \sim \log(x1^2 * x2)$ ' indica: *modele o **log(y)** com função estatística da variável **log(x1^2 * x2)***;

Mais detalhes sobre o argumento 'formula' serão apresentados mais adiante.

Exercícios

Exercício: Relação Diâmetro-Altura em Florestas Plantadas

Ajuste um modelo de regressão linear simples da altura (ht) em função do DAP (dap) das árvores de floresta plantada ([Conjunto de Dados: Inventário em Floresta Plantada](#)) para cada uma das rotações (rot).

Exercício: Equação de Biomassa para Árvores de *Eucalyptus saligna*

Utilizando o conjunto de dados de *E. saligna* ([Conjunto de Dados: Biomassa de Árvores de Eucalyptus saligna](#)), construa um modelo de regressão da biomassa do tronco das árvores (tronco) em função do diâmetro (dap) e altura (ht), utilizando dois modelos: $b_i = \beta_0 + \beta_1 (d_i^2, h_i) + \epsilon_i$

e

$$\ln(b_i) = \beta_0 + \beta_1 \ln(d_i) + \beta_2 \ln(h_i) + \epsilon_i$$

onde:

- b_i é biomassa do tronco;
- d_i é o DAP da árvore;
- h_i é a altura total da árvore;

Funções que Atuam sobre Objetos 'lm'

O objeto produzido pela função 'lm' tem classe 'lm' (*linear model*), ou seja é um modelo linear. Como modelo linear, esse objeto receberá tratamento particular se utilizarmos algumas funções básicas sobre ele.

- **summary:** a função 'summary' apresenta um resumo do modelo linear com:
 1. estatísticas descritivas dos resíduos;
 2. teste *t* dos coeficientes de regressão;
 3. erro padrão da estimativa;
 4. coeficiente de determinação e coef. de det. ajustado;
 5. teste *F* geral do modelo.

```
> summary( hipsol )
```

Call:

```
lm(formula = ht ~ dap, data = egr)
```

Residuals:

Min	1Q	Median	3Q	Max
-12.9306	-2.1109	-0.5408	1.6642	20.9390

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	0.79604	0.12120	6.568	5.63e-11	***
dap	1.27232	0.01006	126.459	< 2e-16	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.093 on 4800 degrees of freedom

Multiple R-Squared: 0.7691, Adjusted R-squared: 0.7691

F-statistic: 1.599e+04 on 1 and 4800 DF, p-value: < 2.2e-16

- **anova:** a função 'anova' apresenta a Tabela de análise de variância, tendo as variáveis

preditoras como fatores:

```
> anova( hipsol )
Analysis of Variance Table

Response: ht
      Df Sum Sq Mean Sq F value    Pr(>F)
dap      1 153020  153020    15992 < 2.2e-16 ***
Residuals 4800  45929      10
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- **plot:** a função 'plot' apresenta uma série de gráficos para análise do modelo. Ela possui o argumento 'which' que define quais dos seis gráficos pré-definidos se deseja ver:

which	O que a função faz	Verificação do modelo
which=1	gráfico de dispersão resíduo versus valor ajustado	Linearidade na relação x-y
which=2	gráfico quantil-quantil normal dos resíduos	Normalidade
which=3	gráfico de dispersão da raiz quadrada do valor absoluto do resíduo padronizado versus valor ajustado	Homogeneidade de variâncias
which=4	distância de Cook por observação	Observações influentes
which=5	gráfico de dispersão do resíduo padronizado versus <i>leverage</i> (medida de influência) , com a distância de Cook	Observações influentes
which=6	gráfico de dispersão da distância de Cook versus <i>leverage</i> (medida de influência)	Observações influentes

O valor default do argumento which é 'which = c(1:3, 5)'.

```
> plot( hipsol )
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
>
```

- **coef:** a função 'coef' retorna os coeficientes de regressão do modelo linear:

```
> coef( hipsol )
(Intercept)      dap
  0.7960402    1.2723242
>
```

* **residuals:** a função 'residuals' (também pode ser evocada por 'resid') retorna os resíduos do modelo linear. * **fitted:** a função 'fitted' (também pode ser evocada por 'fitted.values') retorna os *valores ajustados* do modelo linear.

```
> plot( resid( hipsol ) ~ fitted( hipsol ) )
```

* **predict:** a função 'predict' retorna os valores *preditos* para novas observações:

```
> predict( hipsol, data.frame( dap=c(10,50,100) ) )
```

```

      1      2      3
13.51928 64.41225 128.02846
>
> predict( hipsol, data.frame( dap=range(egr$dap) ) )
      1      2
6.060955 38.055431
>

```

Exercícios

Exercício: Analisando os Modelos de Regressão

Utilizando as funções apresentadas acima analise os modelos de regressão construídos nos exercícios anteriores com relação a:

1. adequação das pressuposições dos modelos lineares;
2. significância das estimativas dos coeficientes de regressão;
3. qualidade dos modelos para uso em predições.

Exercício: Realizando Predições

Considere as árvores da tabela abaixo:

Árvore	DAP	Altura
1	10 cm	12 m
2	20 cm	25 m
3	30 cm	40 m

Faça a predição da biomassa do tronco das árvores com base nos dois modelos de equação de biomassa ajustados.

Regressão Ponderada

A regressão ponderada é utilizada para corrigir o problema de heterogeneidade de variâncias.

Consideremos o exercício do modelo de equação de biomassa do tronco em função do diâmetro e altura. O modelo original apresenta claramente problemas com a pressuposição de homogeneidade de variâncias:

```

> esa = read.csv("esaligna.csv",header=T)
> plot( lm( tronco ~ I(dap^2 * ht), data=esa ) , which=c(1,3) )
Hit <Return> to see next plot:
Hit <Return> to see next plot:
>

```

Se o modelo for ponderado por uma potência do inverso da variável preditora ($1/(dap^2 * ht)$), talvez se torne um modelo com variância homogênea.

```

> plot( lm( tronco ~ I(dap^2*ht), data=esa, weights=1/(dap^2*ht)^0.5 ),

```

```

which=3 )
> plot( lm( tronco ~ I(dap^2*ht), data=esa, weights=1/(dap^2*ht)^0 ),
which=3 )
> plot( lm( tronco ~ I(dap^2*ht), data=esa, weights=1/(dap^2*ht)^1 ),
which=3 )
> plot( lm( tronco ~ I(dap^2*ht), data=esa, weights=1/(dap^2*ht)^2 ),
which=3 )
> plot( lm( tronco ~ I(dap^2*ht), data=esa, weights=1/(dap^2*ht)^3 ),
which=3 )
>

```

Qual dos valores de potência (0.5; 0; 1; 2; 3) lhe parece mais adequado?

Exercícios

Exercício: Modelos de Biomassa para *Eucalyptus saligna*

Utilizando o mesmo modelo do exemplo acima, ajuste modelos de equação de biomassa para

1. **biomassa total** (total) e
2. **biomassa de ramos** (sobra),

de modo que os modelos não possuam problema de heterogeneidade de variância.

Variável Factor como Variável Indicadora (dummy)

Uma forma de incluirmos variáveis categóricas em modelos de regressão é através do uso de **variáveis indicadoras**. Para isso, as variáveis categóricas no R devem ser vistas como uma variável 'factor'.

A variável 'factor' indica uma variável que possui **níveis** (levels) sendo, portanto, uma variável categórica típica dos modelos estatísticos.

Esse tipo de variável existe no R para tornar mais fácil a modelagem estatística. Assim, quando o R lê um conjunto de dados e encontra uma variável **alfa-numérica**, ele automaticamente assume que se trata de uma variável 'factor'.

Vejamos como exemplo os dados de inventário floresta em floresta plantada:

```

> egr = read.csv("egrandis.csv", header=T)
>
> names(egr)
[1] "especie" "rot"      "regiao"  "inv"     "faz"     "proj"    "talhao"
[8] "parcela" "arv"     "fuste"   "cap"     "ht"      "hdom"    "idade"
[15] "carac"   "dap"
>

```

```
> class( egr$regiao )
[1] "factor"
> class( egr$especie )
[1] "factor"
> class( egr$rot )
[1] "integer"
> class( egr$faz )
[1] "integer"
>
```

Note que as variáveis 'regiao' e 'especie' foram assumidas como 'factor'.

Note também que as variáveis 'rot' (rotação) e 'faz' (fazenda) embora também sejam variáveis categóricas, elas foram codificadas por números inteiros e, conseqüentemente, o R assumiu tratar-se de variáveis quantitativas.

Nos **modelos lineares de regressão**, as variáveis 'factor' podem ser assumidas automaticamente como variáveis indicadoras (variáveis dummy).

```
> hipso2 = lm( ht ~ dap + regiao, data=egr )
> summary( hipso2 )
```

Call:
lm(formula = ht ~ dap + regiao, data = egr)

Residuals:

Min	1Q	Median	3Q	Max
-10.6196	-1.6235	-0.3575	1.2476	19.5109

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	5.079650	0.145714	34.86	<2e-16 ***
dap	1.116119	0.009403	118.70	<2e-16 ***
regiaoBotucatu	-3.627353	0.100221	-36.19	<2e-16 ***
regiaoItatinga	-3.827592	0.100715	-38.00	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.647 on 4798 degrees of freedom
Multiple R-Squared: 0.831, Adjusted R-squared: 0.8309
F-statistic: 7866 on 3 and 4798 DF, p-value: < 2.2e-16

Nesse caso ('ht ~ dap + regiao') a variável região entrou alterando o **intercepto** da regressão entre altura (ht) e diâmetro (dap).

Note que além do coeficiente de inclinação para variável 'dap' aparecem coeficientes de regressão associados às variáveis 'regiaoBotucatu' e 'regiaoItatinga'. O que significa isso?

O modelo ajustado pela fórmula 'ht ~ dap + regiao' é:

$$\hat{h}_i = \beta_0 + \beta_1 d_i + \beta_2 I_{\{\text{Botucatu}\}} + \beta_3 I_{\{\text{Itatinga}\}} + \varepsilon_i$$

onde:

- h_i é a altura das árvores;
- d_i é o DAP das árvores;
- I_{Botucatu} é a variável indicadora para região Botucatu, isto é, ela tem valor **1** se a região for Botucatu e valor **0** (zero) se a região não for Botucatu;
- I_{Itatinga} é a variável indicadora para região Itatinga.

A variável região tem três níveis (levels)

```
> levels(egr$regiao)
[1] "Bofete" "Botucatu" "Itatinga"
>
```

O R cria automaticamente 2 variáveis indicadoras, uma para Botucatu e outra para Itatinga, pois a região de Bofete (primeiro nível do fator) é assumida como o *default*.

Como se utiliza o modelo para predição?

- Para Bofete a predição é: $\widehat{h}_i = \beta_0 + \beta_2 d_i$
- Para Botucatu a predição é: $\widehat{h}_i = (\beta_0 + \beta_3) + \beta_2 d_i$
- Para Itatinga a predição é: $\widehat{h}_i = (\beta_0 + \beta_4) + \beta_2 d_i$

É possível ajustar um **modelo de interação completo** do diâmetro com a variável região, alterando o *intercepto* e a *inclinação* do modelo em cada regiões:

```
>
> hipso3 = lm( ht ~ dap * regiao, data=egr )
> summary( hipso3 )

Call:
lm(formula = ht ~ dap * regiao, data = egr)

Residuals:
    Min       1Q   Median       3Q      Max
-12.8439  -1.5492  -0.2357   1.2582   19.3736

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    7.99813    0.20736  38.570 < 2e-16 ***
dap             0.90370    0.01436  62.935 < 2e-16 ***
regiaoBotucatu  -9.03699    0.25969 -34.799 < 2e-16 ***
regiaoItatinga  -5.74018    0.29200 -19.658 < 2e-16 ***
dap:regiaoBotucatu  0.45060    0.01981  22.750 < 2e-16 ***
dap:regiaoItatinga  0.10746    0.02503   4.293 1.80e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.508 on 4796 degrees of freedom
Multiple R-Squared: 0.8484, Adjusted R-squared: 0.8483
F-statistic: 5369 on 5 and 4796 DF, p-value: < 2.2e-16
```


Note que se quisermos usar uma variável como indicadora, mas ela foi codificada como variável numérica, teremos que **forçar** sua transformação em variável 'factor':

```
coef( lm( ht ~ dap * rot, data = egr ) )
(Intercept)          dap          rot      dap:rot
 3.790676759  1.206524873 -1.556650365  0.004740839
>
>
> coef( lm( ht ~ dap * as.factor(rot) , data = egr ) )
(Intercept)          dap      as.factor(rot)2
dap:as.factor(rot)2
 2.234026394      1.211265712      -1.556650365
0.004740839
```

Exercícios

Exercício: Altura das Árvores Dominantes em Floresta Plantada

Considere o seguinte modelo (*modelo de Schumacher*):

$$\ln(y_i) = \beta_0 + \beta_1 (1/x_i) + \epsilon_i$$

Ajuste esse modelo de regressão à altura das árvores dominantes (hdom) em função da idade (idade) das árvores da floresta plantada ([Conjunto de Dados: Inventário em Floresta Plantada](#)).

Compare um modelo geral (ajustado a todos os dados) com um modelo ajustado por região.

Compare os resíduos do modelo geral com os resíduos do modelo por região, analisando a **distribuição do resíduo por região**.

Exercício: Relação Altura-Diâmetro de Árvores de Caixeta

Ajuste modelos de regressão linear da altura ('h') em função do diâmetro ('dap') **somente para árvores de caixaeta** (*Tabebuia cassinoides*) nos diferentes caixetais ('local').

Considere nessa tarefa os seguintes modelos:

- **Modelo A:** $h_i = \beta_0 + \beta_1 d_i + \epsilon_i$
- **Modelo B:** $\ln(h_i) = \beta_0 + \beta_1 \ln(d_i) + \epsilon_i$
- **Modelo C:** $h_i = \beta_0 + \beta_1 d_i + \beta_2 d_i^2 + \epsilon_i$

Análise de Variância

Os objetivos dos modelos lineares de análise de variância são bem diferentes dos modelos lineares de regressão. Nos modelos de regressão a questão central é estimar parâmetros, seja para explicar relações, seja para fazer previsões.

Nos modelos de análise de variância a questão é comparar a importância de **fatores** sobre o

comportamento da variável resposta.

Experimento em Blocos Casualizados

Tomemos como exemplo os dados do experimento de mudas no viveiro ([Conjunto de Dados: Experimento de Mudas](#)). Nesse experimento temos os seguintes fatores:

- Espécie ('especie'),
- Bloco ('bloco'), e
- Substrato ('substrato').

O experimento deseja saber se existe diferença entre os substratos no crescimento em altura ('altura') das mudas. O delineamento foi em blocos casualizados ('bloco').

Para visualizar esse experimento podemos ler os dados do arquivo [altura-mudas.csv](#) ([apagar extensão .pdf](#)), através da função 'plot':

```
>
> mudas = read.csv("dados/altura-mudas.csv", header=T)
> summary(mudas)
      especie      bloco      substrato      altura
paineira:60  Min.    :1.0  Min.    : 1.0  Min.    : 15.40
tamboril:60  1st Qu.:2.0  1st Qu.: 3.0  1st Qu.: 32.21
              Median :3.5  Median : 5.5  Median : 46.00
              Mean   :3.5  Mean   : 5.5  Mean   : 49.20
              3rd Qu.:5.0  3rd Qu.: 8.0  3rd Qu.: 65.00
              Max.   :6.0  Max.   :10.0  Max.   :105.12

>
>
> plot( altura ~ bloco + substrato , data=mudas ,
subset=especie=="paineira")
Hit <Return> to see next plot:
Hit <Return> to see next plot:
>
> plot( altura ~ bloco + substrato , data=mudas ,
subset=especie=="tamboril")
Hit <Return> to see next plot:
Hit <Return> to see next plot:
>
```

Para ajustar um modelo linear num experimento, podemos utilizar a função 'lm' como no caso da regressão linear:

```
> muda.pai = lm( altura ~ as.factor(bloco) + as.factor(substrato),
data=mudas, subset= especie=="paineira" )
> class(muda.pai)
[1] "lm"
>
> muda.tam = lm( altura ~ as.factor(bloco) + as.factor(substrato),
data=mudas, subset= especie=="tamboril" )
```

```
> class(muda.tam)
[1] "lm"
>
```

Para analisar as pressuposições do modelo utilizamos a função 'plot', da mesma forma que se faz na regressão linear.

Sendo um experimento, o interesse principal é verificar a importância dos **fatores**: os tratamentos ('substrato') e os blocos ('bloco'). Para isso utilizamos a função 'anova':

```
> anova( muda.pai )
Analysis of Variance Table

Response: altura

          Df Sum Sq Mean Sq F value    Pr(>F)
as.factor(bloco)      5  3822.1    764.4   13.664 4.016e-08 ***
as.factor(substrato)  9 27204.4   3022.7   54.030 < 2.2e-16 ***
Residuals            45  2517.5     55.9
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> anova( muda.tam )
Analysis of Variance Table

Response: altura

          Df Sum Sq Mean Sq F value    Pr(>F)
as.factor(bloco)      5  2582.6    516.5   5.1933 0.0007594 ***
as.factor(substrato)  9  9181.3   1020.1  10.2570 2.177e-08 ***
Residuals            45 4475.6     99.5
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

Exercícios

Exercício: Altura dos Caixetais

Verifique se existe diferenças estatisticamente significativas na **altura média** dos caixetais ([caixeta.csv](#) ([apagar extensão .pdf](#))).

Será que os caixetais diferem em termos de **altura máxima** ou **área basal**?

Outros Delineamentos Experimentais

Considere o experimento de mudas de espécies arbóreas. Se ao invés de trabalhar com a **espécie** em duas análises separadas, houvesse interesse em fazer uma análise conjunta das duas espécies, verificando a interação entre espécie e substrato.

Nesse caso, o experimento se torna um **experimento fatorial 2 x 10**:

```
>
> muda.sp = lm( altura ~ as.factor(bloco) + especie * as.factor(substrato),
data=mudas )
> anova(muda.sp)
Analysis of Variance Table

Response: altura

              Df Sum Sq Mean Sq F value    Pr(>F)
as.factor(bloco)    5   3956      791  7.9598 2.673e-06 ***
especie              1   3183      3183 32.0265 1.606e-07 ***
as.factor(substrato)  9  32910      3657 36.7909 < 2.2e-16 ***
especie:as.factor(substrato)  9   3476      386  3.8853 0.0003163 ***
Residuals          95   9442        99
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

No que a fórmula para o experimento é apresentada de modo diferente:

```
altura ~ as.factor(bloco) + especie * as.factor(substrato)
```

O elemento **especie * as.factor(substrato)** inclui todos os efeitos ligados a interação entre espécie e substrato, e que aparecem na tabela de análise de variância:

- **especie** (com 1 grau de liberdade) se refere ao **efeito principal** da espécie;
- **as.factor(substrato)** (com 9 graus de liberdade) se refere ao **efeito principal** do substrato;
- **especie:as.factor(substrato)** (com 9 graus de liberdade) se refere à **interação** espécie x substrato.

O elemento chave nessa fórmula é o asterisco (*) que representa todos os efeitos ligados a interação entre dois fatores. Como foi dito, na fórmula estatística os sinais convencionais de operações matemáticas tem outro significado. A tabela abaixo detalha os símbolos utilizados para definir diferentes delineamentos experimentais.

Símbolos utilizados nas Fórmulas Estatísticas para definir diferentes Delineamentos Experimentais

Expressão	Significado
$Y \sim X$	Modele Y como função estatística de X
$A + B$	inclui ambos os fatores A e B
$A - B$	inclui todos os efeitos em A, exceto os que estão em B
$A * B$	$A + B + A:B$
A / B	$A + B \%in\% (A)$ modelos hierárquicos
$A:B$	efeito da interação entre os fatores A e B
$B \%in\% A$	efeitos de B dentro dos níveis de A
A^m	todos os termos de A cruzados até à ordem m

Aliadas a esses símbolos, o R possui uma série de funções que permitem a análise de virtualmente qualquer delineamento experimental. Esse tópico requer, no entanto, um curso específico de análise

experimental utilizando o R, e vai muito além do objetivo desse curso introdutório.

Exercícios

Exercício: Fatores que Influência a Altura em Florestas Plantadas I

Utilizando os dados de árvores de floresta plantada ([Conjunto de Dados: Inventário em Floresta Plantada](#)), tome a altura média das árvores dominantes (média de 'hdom' por 'parcela') como variável resposta e verifique a influência dos fatores: região ('regiao') e rotação ('rot').

Para discussão: a relação entre esses fatores deve ser de **interação** ou **hierárquica**?

Exercício: Fatores que Influência a Altura em Florestas Plantadas II

Utilizando os dados de árvores de floresta plantada ([Conjunto de Dados: Inventário em Floresta Plantada](#)), tome a altura média das árvores dominantes (média de 'hdom' por 'parcela') como variável resposta e verifique a influência dos fatores: região ('regiao') e projeto ('proj').

Para discussão: a relação entre esses fatores deve ser de **interação** ou **hierárquica**?

Explorando a Interação entre Fatores

Freqüentemente, a interpretação da **interação** entre dois ou mais fatores é espinhosa e chegar a conclusões baseado apenas na tabela de análise de variância pode gerar equívocos. Uma análise gráfica de interações é sempre instrutiva.

Existe no R a função 'interaction.plot' que permite construir gráficos de interação entre fatores que facilitam a interpretação dos resultados estatístico. Seus argumentos principais são:

```
function (x.factor, trace.factor, response, fun = mean)
```

- **x.factor** é o fator que ficará nas abscissas (eixo-x);
- **trace.factor** é o fator que será usado para distinguir diferentes linhas no gráfico;
- **response** é a variável resposta que será grafada nas ordenadas (eixo-y);
- **fun** é a função da estatística a ser utilizada.

Vejamos a interação entre espécies e substrato no experimento do crescimento de mudas de espécies arbóreas:

```
> interaction.plot( mudas$substrato, mudas$especie, mudas$altura ,  
col=c("red", "blue"))
```

Exercícios

Exercício: Fatores que Influência a Altura em Florestas Plantadas III

Tomando a altura média das árvores dominantes (média de 'hdom' por 'parcela') como variável resposta (dados de árvores de floresta plantada — [Conjunto de Dados: Inventário em Floresta Plantada](#)), verifique **graficamente** a interação entre região ('regiao') e rotação ('rot').

Exercício: Variabilidade de Caixetais

Considere os dados do levantamento em três caixetais ([Conjunto de Dados: Levantamento em Caixetais](#)).

Pergunta-se: em qual dos caixetais ('local'), o diâmetro das árvores ('dap') é mais variável entre as parcelas ('parcela') quando se considera:

- diâmetro **médio**;
- diâmetro **mediano**;
- diâmetro **máximo**?

Responda com base numa análise gráfica.

Comparação de Modelos

Função "anova": mais do que ANOVA

Um aspecto essencial à construção de modelos lineares é a comparação entre modelos.

A função **"anova"**, apesar do nome, é uma função muito utilizada para comparar modelos lineares que pertençam a uma certa *hierarquia* de modelos.

Vejamos o exemplo de construção de uma equação de biomassa com o seguinte forma:

$$b_i = \beta_0 + \beta_1 d_i + \beta_2 d_i^2 + \beta_3 h_i + \beta_4 (d_i, h_i) + \beta_5 (d_i^2, h_i) + \beta_6 (d_i, h_i^2) + \epsilon_i$$

Embora esse seja um modelo muito problemático, ele serve para ilustrar o problema de seleção de modelos. Vejamos o que acontece utilizando os dados de árvores de *E. saligna* ([Conjunto de Dados: Biomassa de Árvores de Eucalyptus saligna](#)):

```
> biom = lm( total ~ dap + I(dap^2) + ht + I(dap * ht) + I(dap^2 * ht) +
I(dap * ht^2), data=esa )
> summary(biom)
```

Call:

```
lm(formula = total ~ dap + I(dap^2) + ht + I(dap * ht) + I(dap^2 *
ht) + I(dap * ht^2), data = esa)
```

Residuals:

Min	1Q	Median	3Q	Max
-----	----	--------	----	-----

```
-38.670 -7.688 -1.022 8.561 45.191
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-51.60332	65.40689	-0.789	0.437
dap	7.68140	11.24405	0.683	0.500
I(dap^2)	0.35420	0.53985	0.656	0.517
ht	7.48430	5.55867	1.346	0.189
I(dap * ht)	-1.42975	0.82821	-1.726	0.095
I(dap^2 * ht)	0.03763	0.03461	1.087	0.286
I(dap * ht^2)	0.01105	0.01593	0.694	0.493

```
---
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 15.13 on 29 degrees of freedom

Multiple R-Squared: 0.9728, Adjusted R-squared: 0.9672

F-statistic: 172.9 on 6 and 29 DF, p-value: < 2.2e-16

```
>
```

Veja que nenhuma das variáveis preditoras se mostrou significativa (nível de probabilidade de 5%) para estimar a biomassa total das árvores. Mas esse resultado é razoável?

Vejamos o que a função “**anova**” nos mostra:

```
> anova(biom)
```

Analysis of Variance Table

Response: total

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
dap	1	218121	218121	952.7442	< 2.2e-16 ***
I(dap^2)	1	17791	17791	77.7108	1.063e-09 ***
ht	1	352	352	1.5375	0.22493
I(dap * ht)	1	312	312	1.3648	0.25222
I(dap^2 * ht)	1	816	816	3.5633	0.06911 .
I(dap * ht^2)	1	110	110	0.4811	0.49343
Residuals	29	6639	229		

```
---
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```
>
```

Nesse caso parece que o diâmetro e o diâmetro ao quadrado são significativos, mas os demais termos não. Por que os testes geram resultados diferentes?

As funções “**summary**” e “**anova**” realizam testes de forma distinta:

- O teste *t* da função **summary** testa cada variável preditora assumindo que **todas as demais variáveis já estão presentes no modelo**.
- O teste *F* da função **anova** testa as variáveis preditoras na sequência apresentada no modelo, assumindo que as **variáveis anteriores** já estavam no modelo.

Desta forma, a função **anova** realiza teste de um modelo contra outro numa sequência definida. O mesmo resultado se obtém partindo o modelo original numa série de modelos:

Modelo 0: $b_i = \beta_0 + \varepsilon_i$

Modelo 1: $b_i = \beta_0 + \beta_1 d_i + \varepsilon_i$

Modelo 2: $b_i = \beta_0 + \beta_1 d_i + \beta_2 d_i^2 + \varepsilon_i$

Modelo 3: $b_i = \beta_0 + \beta_1 d_i + \beta_2 d_i^2 + \beta_3 h_i + \varepsilon_i$

Modelo 4: $b_i = \beta_0 + \beta_1 d_i + \beta_2 d_i^2 + \beta_3 h_i + \beta_4 (d_i \backslash, h_i) + \varepsilon_i$

Modelo 5: $b_i = \beta_0 + \beta_1 d_i + \beta_2 d_i^2 + \beta_3 h_i + \beta_4 (d_i \backslash, h_i) + \beta_5 (d_i^2 \backslash, h_i) + \varepsilon_i$

Modelo 6: $b_i = \beta_0 + \beta_1 d_i + \beta_2 d_i^2 + \beta_3 h_i + \beta_4 (d_i \backslash, h_i) + \beta_5 (d_i^2 \backslash, h_i) + \beta_6 (d_i \backslash, h_i^2) + \varepsilon_i$

Podemos ajustar esses modelos e utilizar a função **anova** para testá-los numa sequência:

```
> m0 = lm( total ~ 1 , data=esa )
> m1 = lm( total ~ dap , data=esa )
> m2 = lm( total ~ dap + I(dap^2) , data=esa )
> m3 = lm( total ~ dap + I(dap^2) + ht , data=esa )
> m4 = lm( total ~ dap + I(dap^2) + ht + I(dap * ht), data=esa )
> m5 = lm( total ~ dap + I(dap^2) + ht + I(dap * ht) + I(dap^2 * ht),
data=esa )
> m6 = lm( total ~ dap + I(dap^2) + ht + I(dap * ht) + I(dap^2 * ht) + I(dap
* ht^2), data=esa )
>
>
```

```
> anova(m0, m1, m2, m3, m4, m5, m6)
Analysis of Variance Table
```

```
Model 1: total ~ 1
Model 2: total ~ dap
Model 3: total ~ dap + I(dap^2)
Model 4: total ~ dap + I(dap^2) + ht
Model 5: total ~ dap + I(dap^2) + ht + I(dap * ht)
Model 6: total ~ dap + I(dap^2) + ht + I(dap * ht) + I(dap^2 * ht)
Model 7: total ~ dap + I(dap^2) + ht + I(dap * ht) + I(dap^2 * ht) + I(dap *
ht^2)
```

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	35	244142				
2	34	26021	1	218121	952.7442	< 2.2e-16 ***
3	33	8230	1	17791	77.7108	1.063e-09 ***


```

4      32    7878    1      352    1.5375    0.22493
5      31    7565    1      312    1.3648    0.25222
6      30    6749    1      816    3.5633    0.06911
7      29    6639    1      110    0.4811    0.49343
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>

```

É importante lembrar que a função **anova** realiza o teste **na ordem que os modelos são apresentados**, e que isso pode ter forte influência nos resultados obtidos.

```

> anova( m0, lm(total ~ I(dap^2*ht), data=esa), lm( total ~ I(dap^2*ht) +
dap, data=esa) )
Analysis of Variance Table

Model 1: total ~ 1
Model 2: total ~ I(dap^2 * ht)
Model 3: total ~ I(dap^2 * ht) + dap
  Res.Df    RSS Df Sum of Sq    F    Pr(>F)
1      35 244142
2      34  22160  1    221982 473.534 < 2.2e-16 ***
3      33  15470  1      6690  14.272 0.0006292 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>

```

Exercícios

Exercício: Biomassa do Tronco de Árvores de *E. saligna*

Com base nos modelos apresentados acima, construa vários modelos para biomassa do tronco ('tronco') de *E. saligna* ([Conjunto de Dados: Biomassa de Árvores de Eucalyptus saligna](#)).

Exercício: Modelo Polinomial

Construa um modelo polinomial (até quarto grau) da altura ('ht') em função do diâmetro ('dap') de árvores em caixetais ([Conjunto de Dados: Levantamento em Caixetais](#)). Verifique os termos significativos.

Algumas Funções para Comparação de Modelos

Existem várias outras funções para auxiliar na construção e comparação de modelos.

As funções 'add1' e 'drop1' permitem adicionar ou retirar **um-a-um** os termos dos modelos lineares:

```

> add1( object = m1, scope = . ~ dap + ht + I(dap*ht) + I(dap^2*ht) ,
test="F" )
Single term additions

```

```
Model:
total ~ dap
      Df Sum of Sq    RSS    AIC F value    Pr(>F)
<none>                26020.8    241.0
ht          1         1.0 26019.7    243.0   0.0013   0.97162
I(dap * ht)  1      2507.0 23513.8    239.3   3.5184   0.06956 .
I(dap^2 * ht) 1     10551.1 15469.6    224.3  22.5077 3.912e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

```
> drop1( object = m6, scope = . ~ ., test="F" )
Single term deletions
```

```
Model:
total ~ dap + I(dap^2) + ht + I(dap * ht) + I(dap^2 * ht) + I(dap *
      ht^2)
      Df Sum of Sq    RSS    AIC F value    Pr(>F)
<none>                6639.3    201.8
dap          1      106.8 6746.1    200.4   0.4667 0.49993
I(dap^2)      1       98.6 6737.8    200.4   0.4305 0.51693
ht           1      415.0 7054.3    202.0   1.8128 0.18860
I(dap * ht)   1      682.3 7321.5    203.3   2.9801 0.09493 .
I(dap^2 * ht) 1      270.7 6910.0    201.3   1.1824 0.28582
I(dap * ht^2) 1      110.2 6749.4    200.4   0.4811 0.49343
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

```
> drop1( object = m6, scope = . ~ dap + ht, test="F" )
Single term deletions
```

```
Model:
total ~ dap + I(dap^2) + ht + I(dap * ht) + I(dap^2 * ht) + I(dap *
      ht^2)
      Df Sum of Sq    RSS    AIC F value    Pr(>F)
<none>                6639.3    201.8
dap          1      106.8 6746.1    200.4   0.4667 0.4999
ht           1      415.0 7054.3    202.0   1.8128 0.1886
>
```

Nessas duas funções, o ponto (“.”) na fórmula significa todos os termos do modelo. Ou seja, para um dado modelo, o 'scope' igual a “. ~ .” significa todos os termos da fórmula do modelo.

Outras funções úteis para construção e comparação de modelos são:

- “**step**” que realiza *regressão stepwise*; e
- “**AIC**” que calcula o *Akaike Information Criterion*.

```
> aic.tab = AIC( m0, m1, m2, m3, m4, m5, m6 )
```

```
> aic.tab$AIC.d = abs( c(0, diff(aic.tab$AIC)) )
> aic.tab
      df      AIC      AIC.d
m0  2 423.7551  0.0000000
m1  3 345.1563 78.5987781
m2  4 305.7148 39.4414506
m3  5 306.1411  0.4262982
m4  6 306.6842  0.5430302
m5  7 304.5765  2.1077173
m6  8 305.9841  1.4076291
>
```

Exercícios

Exercício: Biomassa do Tronco de Árvores de *E. saligna* II

Utilizando os modelos para biomassa do tronco ('tronco') de *E. saligna* construídos no exercício acima, utilize as funções **drop1** e **add1** para analisar a importância dos termos individuais do modelo.

Exercício: Relação Altura-Diâmetro em Florestas Plantadas

Utilize a função **AIC** para analisar a importância das variáveis indicadoras nos modelos de relação altura-diâmetro ajustados para florestas de *E. grandis*.

From:

<http://ecor.ib.usp.br/> - ecoR

Permanent link:

http://ecor.ib.usp.br/doku.php?id=03_apostila:06-modelos

Last update: 2020/08/20 19:10

