UNIVERSIDADE FEDERAL DE LAVRAS
DEPARTAMENTO DE CIÊNCIAS EXATAS

Recursos Computacionais Utilizando R

Daniel Furtado Ferreira

LAVRAS
Minas Gerais - Brasil
11 de maio de 2011

Sumário

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução ao programa R

O programa R é um dos melhores *software* de análise estatística existentes na atualidade. Atualmente, somente o programa R tem competido com o SAS® nas mesmas condições. O programa R possibilita tratar arquivos de dados e realizar análises estatísticas, fornecendo relatórios nas mais variadas formas. Para utilizarmos o R, precisamos conhecer como é sua estrutura e como se dá o seu funcionamento. O ambiente de interação com o usuário do R possui uma janela, denominada de console. O console do programa R (Figura 1.1) permite ao usuário a entrada de dados e de códigos para a realização das diferentes análises que deverão ser realizadas, bem como, retorna os resultados de qualquer análise efetuada.

Todo o conteúdo da janela do console pode ser salvo, marcado e eliminado utilizando os recursos do Windows e da barra de ferramentas. Uma forma de preferida de utilizar o R, é digitar os programas e macros que serão executados em uma janela denominada *script*. Para isso o usuário escolhe a opção file (ou arquivo) e a opção new script (ou novo script). Uma janela será aberta e as macros, seqüência de comandos e códigos, serão digitadas ou transportadas de outro programa para este editor. A vantagem de se utilizar este editor *script* refere-se ao fato de podermos modificar qualquer linha, pois ao contrário do *console* em que uma linha não pode ser acessada de imediato, neste editor isso é possível. Assim, após o código ser digitado ele poderá ser executado marcando-se o texto ou as linhas de interesse e utilizando a combinação de tecla *CTRL R*. Podemos executar cada linha separadamente utilizando a mesma combinação de tecla, bastando o cursor estar posicionado sobre ela. Na Figura 1.2 apresentamos um exemplo em

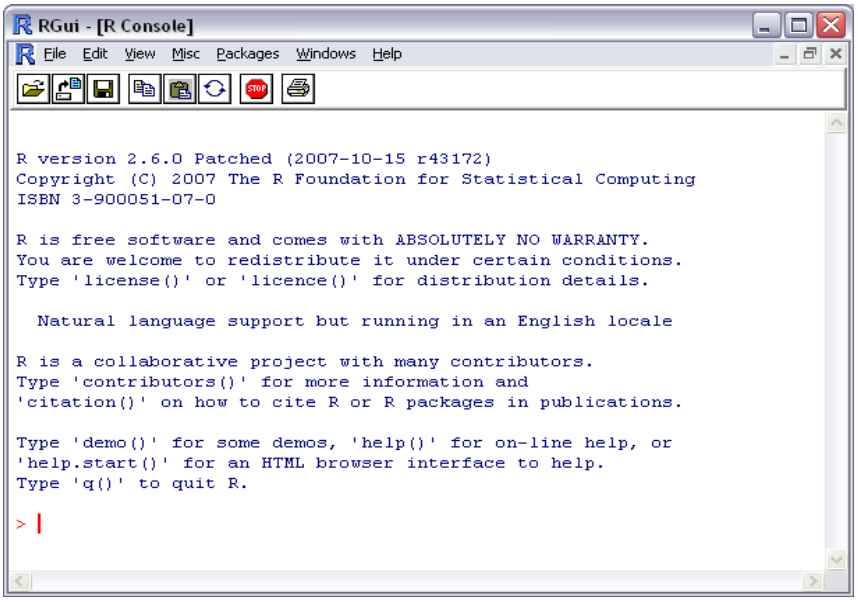


Figura 1.1. Console do programa R ilustrando seu Menu de opções e o comando de prompt aguardando uma ação do usuário.

que definimos um vetor de dados, imprimimos seu conteúdo no console e determinamos sua média. Esta será a maneira pela qual iremos utilizar o programa R neste material.

O R, infelizmente, como o próprio SAS, não é um programa com muita interatividade, a menos que interfaces gráficas sejam utilizadas. Esta é uma possibilidade que o R possui e muitas destas interfaces têm sido implementadas e distribuídas gratuitamente pela internet. O R possui uma vastíssima documentação disponibilizada na internet, no sistema de auxílio do próprio programa e nos diversos livros publicados e muitas vezes disponibilizados gratuitamente pela internet. Existem manuais on line em HTML e em PDF que podem ser consultados pela internet ou que podem ser baixados e utilizados gratuitamente. Uma das maiores vantagens do R sobre o SAS, é o fato dele ser um programa gratuito, podendo ser baixado do site http://www.r-project.org/. Outra vantagem do R são os pacotes (*library*) escritos por pesquisadores das mais diferentes áreas do conhecimento e profissionais da área de estatística. Qualquer pessoa pode contribuir para o desenvolvimento do programa R mediante criação de pa-

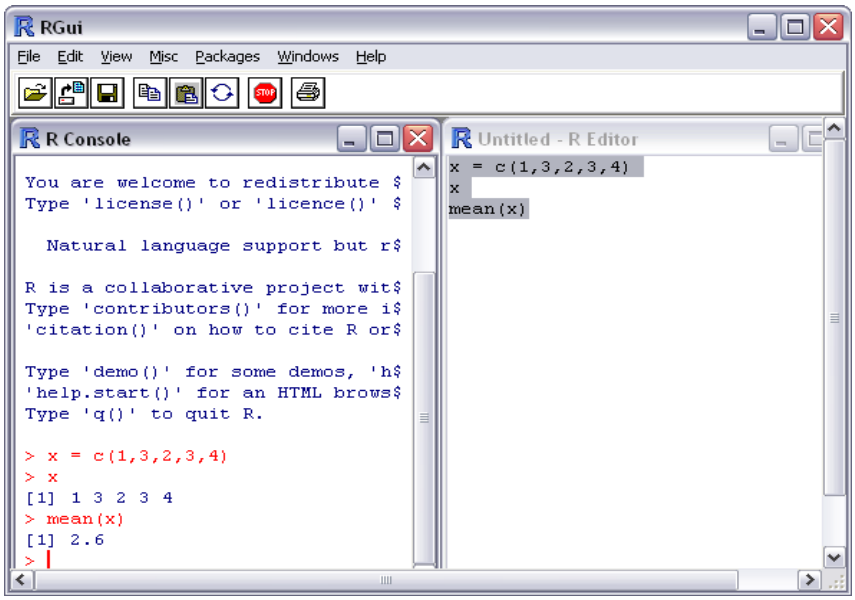


Figura 1.2. Console do programa R ilustrando a janela de *script*, em que um programa é executado e apresenta a definição de um vetor, sua impressão e o cálculo da média.

cotes que façam determinadas análises. Esses pacotes podem ser baixados utilizando o próprio sistema R, após ele ter sido instalado. Desta forma, as extensões do programa são criadas por especialistas em todo o mundo e sempre é possível encontrar um pacote de extensão de análise para uma técnica que outros programas ainda não disponibilizaram mecanismos de análises estatísticas. Se isso não for possível, podemos desenvolver nossas próprias funções e até mesmo criar um novo pacote para realizar tal tarefa. Se desejarmos, podemos disponibilizá-lo para que outros pesquisadores utilizem esta ferramenta. Não temos o objetivo de mostrar como construir um pacote de extensão de análise no R, mas iremos utilizar vários pacotes disponibilizados na literatura para resolvermos nossos problemas, se eles não forem disponibilizados nas ferramentas básicas do *kernel* do R. Esse último tipo de funções não precisa de que pacotes sejam carregados para que elas funcionem. Pesquisando as inúmeras possibilidades na internet, podemos baixar aquelas que nos interessam e ampliar a capacidade do programa. Algum risco se corre, pois mesmo sendo recomendadas pelo *R core Team*,

ou seja, pelo grupo mantenedor do R, algumas funções em determinados pacotes podem não estar otimizadas ou em algumas raras ocasiões ter fraco desempenho.

Neste material veremos apenas algumas das principais rotinas do programa R para realizarmos análises estatísticas. Por ser um programa livre, de código fonte aberto, e que recebe inúmeras contribuições originadas de todas as partes do mundo, seu crescimento em uso e em possibilidades de análises é imenso, como já dissemos. Enfatizaremos os principais recursos relacionados as análises de estatística básica, regressão e estatística experimental. Estes recursos são os mais variados e flexíveis e são abordados de maneira simples, sendo que daremos ênfase nas interpretações estatísticas dos fundamentos dos métodos e da inferência. Utilizaremos apenas exemplos acadêmicos simples, que muitas vezes foram simulados ou são dados fictícios.

O R surgiu do desenvolvimento da linguagem e do ambiente para análise de dados denominada S, originalmente desenvolvida pelos laboratórios da Bell da AT&T e agora Lucent Technologies. O S-PLUS é versão comercial implementada em linguagem S. Os sistemas R e S-PLUS podem ser instalados no windows, UNIX e no Linux. O R também funciona no MacOS, FreeBSD e outros sistemas operacionais.

1.1 Entrada de dados

O R possui inúmeros recursos de importação dos mais diferentes banco de dados e planilhas. Utilizaremos o recurso mais comum de simplesmente "colarmos" os dados em um arquivo texto (ASCII) utilizando o bloco de notas ou qualquer planilha ou editor de texto e importarmos os seu resultado para um objeto R do tipo *data frame*, usando o comando *read.table*. Este formato é mais robusto, livre de vírus, além de os arquivos resultantes ocuparem menos memória. O R diferencia comandos e nomes de variáveis e/ou objetos escritos em letras maiúsculas e minúsculas. Estão temos que tomar muito cuidado com este detalhe, que não pode ser considerado desprezível. Quando possuímos valores perdidos no nosso arquivo ou banco de dados, podemos substituir a célula do arquivo que foi perdida por um *NA* (maiúsculo). Este é o *default* do programa R, podendo ser mudado de acordo com a preferência do usuário.

O arquivo de dados pode ser lido no R de inúmeras maneiras diferentes, porém utilizaremos a forma mais simples. Temos que pensar que cada variável deve ocupar uma coluna do arquivo e cada observação ou unidade amostral uma linha. Esta é a estrutura utilizada pela maioria dos programas de análise estatística. O arquivo de dados deve ser salvo no formato texto, com colunas separadas por espaço ou por símbolo de tabulação. O arquivo texto, que denominaremos de *coelhos.txt* está apresentado a seguir. Este exemplo se refere aos dados de peso em kg de coelhos híbridos Norfolk abatidos aos 90 dias de idade. A primeira linha refere-se ao cabeçalho do arquivo, na qual colocamos os nomes das variáveis. Neste caso há apenas uma coluna e, portanto, apenas um nome de variável no cabeçalho, que é peso.

peso
2.50
2.58
2.60
2.62
2.65
2.66
2.58
2.70
2.55
2.57
2.70
2.62
2.59
2.54
2.53
2.20

Cada linha de comando do R tem algumas palavras reservadas de comandos e termina com um <;>. Apesar de termos inúmeros comandos diferentes para entrarmos com os dados no R, utilizaremos quase sempre a seguinte estrutura:

```
> # lê o arquivo e atribui ao objeto coelhos
> coelhos <- read.table("C:/daniel/Cursos/RCursoTeX/coelhos.txt",
```

```
+                header=TRUE)
> coelhos    # imprime na tela seu conteúdo

   peso
1  2.50
2  2.58
3  2.60
4  2.62
5  2.65
6  2.66
7  2.58
8  2.70
9  2.55
10 2.57
11 2.70
12 2.62
13 2.59
14 2.54
15 2.53
16 2.20
```

Podemos explicar os comandos usados neste simples programa da seguinte forma:

1. objeto coelhos: recebe o resultado da leitura do arquivo *coelhos.txt* que se encontra no diretório "*C:/daniel/Cursos/RCursoTeX/*". Neste caso, este objeto será do tipo *data frame*, pois recebe um arquivo do tipo texto.

2. <- : este comando refere-se a atribuição, ou seja, o objeto denominado coelhos recebe o resultado da função *read.table*. Este comando pode ser substituído pelo símbolo de igual, =.

3. read.table(): esta função é utilizada para a leitura de um arquivo do tipo texto externo. Ela deve receber alguns argumentos. Entre eles, devemos especificar o caminho completo do arquivo do tipo texto dado por "*C:/daniel/Cursos/RCursoTeX/coelhos.txt*". O caminho deve ser especificado utilizando a "/" ou "\\", mas nunca "\". O outro argumento, *header=TRUE*, indica ao R que o arquivo texto possui na primeira, o cabeçalho, que corresponde ao nome das variáveis. No exemplo, o cabeçalho é peso, nome da única variável do arquivo.

4. coelhos: o objeto coelhos, como qualquer outro, quando digitado em uma linha do console, sem nenhuma atribuição irá imprimir o seu conteúdo. O resultado deste objeto, foi impresso no exemplo e possui dezesseis linhas e uma coluna nomeada de peso;

5. linha de comando: os comandos do R são colocados em uma linha ou mais linhas físicas do console. Quando um comando é dividido em mais de uma linha, o R automaticamente muda o seu *prompt* que é representado por ">" para "+". Isso indica que o comando ainda não foi completamente digitado e o programa está esperando os demais códigos para completá-lo. Também podemos, em uma única linha física do console, digitar mais de um comando. Para isso devemos obrigatoriamente digitar um ponto e vírgula separando esses comandos;

6. comentários: os comentários são antecedidos pelo símbolo #. Tudo que estiver após este símbolo, em uma mesma linha não será executado pelo programa, pois será considerado um texto que o usuário criou para o seu próprio auxílio, ou para tornar o código interpretável para outras pessoas que venham a utilizá-lo.

Podemos digitar o programa em um *script*. Selecionamos os comandos e utilizamos a combinação de teclas <CTRL R> para executarmos o programa. Após isso, o R irá ler o arquivo de dados coelhos.txt e imprimir o seu conteúdo na tela do console. Podemos confirmar que o objeto coelhos é do tipo *data frame* com o seguinte comando:

```
> is.data.frame(coelhos)

[1] TRUE
```

Observamos que a resposta é <TRUE>, ou seja, o objeto é do tipo desejado. Observamos que quando não atribuímos o comando a nenhum objeto, seu resultado é impresso de imediato no console. Podemos acessar qualquer coluna do objeto coelhos adicionando a coelhos, o símbolo $, seguido do nome da coluna, ou variável. No caso teríamos coelhos$peso, conforme ilustrado no exemplo a seguir:

```
>   coelhos$peso
```

```
 [1] 2.50 2.58 2.60 2.62 2.65 2.66 2.58 2.70 2.55 2.57
[11] 2.70 2.62 2.59 2.54 2.53 2.20
```

Um segundo exemplo com mais de uma variável é apresentado na seqüência com dados de dez árvores de *Araucaria angustifolia*. A primeira variável Y é o volume em $m^3/acre$, a segunda variável X_1 é a área basal das árvores, a terceira variável X_2 é esta mesma área basal, mas tomada com referência a área basal de outra espécie (*Pinus taeda*) e a quarta variável X_3 é a altura das árvores em pés. Devemos criar um arquivo dos dados das $n = 10$ árvores para que o R possa acessá-lo, conforme o método que optamos por utilizar. Denominamos este arquivo de *arvores.txt* e o colocamos no mesmo diretório que havíamos posto o arquivo *coelhos.txt*.

Y	X1	X2	X3
65	41	79	35
78	71	48	53
82	90	80	64
86	80	81	59
87	93	61	66
90	90	70	64
93	87	96	62
96	95	84	67
104	100	78	70
113	101	96	71

O programa R para acessá-lo é dado por:

```
> # lê o arquivo e atribui ao objeto árvores
> arvores <- read.table("C:/daniel/Cursos/RCursoTeX/arvores.txt",
+               header=TRUE)
> arvores                 # imprime na tela seu conteúdo

     Y  X1 X2 X3
1   65  41 79 35
2   78  71 48 53
3   82  90 80 64
4   86  80 81 59
5   87  93 61 66
6   90  90 70 64
7   93  87 96 62
```

```
8   96  95 84 67
9  104 100 78 70
10 113 101 96 71
```

```
> is.data.frame(arvores)  # verifica se o objeto é um data frame
```

```
[1] TRUE
```

```
> arvores$Y               # imprime a variável Y
```

```
 [1]  65  78  82  86  87  90  93  96 104 113
```

Uma importante situação que acontece em exemplos reais é a ocorrência de variáveis qualitativas. Estas variáveis são identificadas por nomes alfanuméricos e o R permite sua presença. Assim, se um conjunto de dados possui 3 variáveis, sendo por exemplo blocos, tratamentos e produção e a variável tratamento possui seus níveis qualitativos (nomes), então devemos criar o arquivo de dados normalmente. As variáveis qualitativas, em geral, são fatores, ou seja variáveis que são utilizadas como classificatórias em um modelo linear, ou seja, em uma análise de variância. As variáveis quantitativas podem ser fatores ou numéricas. No primeiro caso, elas se comportam como os fatores qualitativos e no segundo, são apropriadas para para uso em modelos de regressão ou como covariáveis em modelos de análise de variância ou cofatores em modelos generalizados e em regressão logística. As variáveis qualitativas são aquelas variáveis cujos níveis são nomes e não números. Os dados a seguir a produtividade proveniente de um exemplo fictício de um experimento em blocos casualizados, com 4 blocos e 3 tratamentos (A, B, C).

bl	trat	prod
1	A	12.23
1	B	10.31
1	C	11.90
2	A	14.56
2	B	10.17
2	C	13.45
3	A	16.11
3	B	19.12
3	C	14.73
4	A	12.78
4	B	10.67
4	C	11.34

O programa R para realizarmos a leitura deste arquivo é dado por:

```
> # lê o arquivo e atribui ao objeto dadfict
> dadfict <- read.table("C:/daniel/Cursos/RCursoTeX/dadfic.txt",
+               header=TRUE)
> dadfict                 # imprime na tela seu conteúdo

   bl trat  prod
1   1    A 12.23
2   1    B 10.31
3   1    C 11.90
4   2    A 14.56
5   2    B 10.17
6   2    C 13.45
7   3    A 16.11
8   3    B 19.12
9   3    C 14.73
10  4    A 12.78
11  4    B 10.67
12  4    C 11.34

> is.data.frame(dadfict)  # verifica se o objeto é um data frame

[1] TRUE

> names(dadfict)          # imprime os nomes das colunas

[1] "bl"   "trat" "prod"
```

```
> dadfict$bl              # imprime a variável bl

 [1] 1 1 1 2 2 2 3 3 3 4 4 4

> is.factor(dadfict$bl)   # verifica se bl é um fator

[1] FALSE

> is.numeric(dadfict$bl)  # verifica se bl é numérico

[1] TRUE

> is.factor(dadfict$trat) # verifica se trat é um fator

[1] TRUE
```

O comando *names(dadfict)* imprime os nomes das colunas do objeto *dadfict*, que recebeu o arquivo *dadfic.txt*. Os comandos *is.factor(dadfict$bl)* e *is.numeric(dadfict$bl)* foram utilizados para verificar se o fator bloco, utilizado como argumento, era um fator ou era numérico, respectivamente. É óbvio que basta utilizarmos um deles para obtermos o resultado pretendido. Neste caso, verificamos que bloco é tratado como uma variável numérica e não como um fator. Assim, se formos utilizar bloco em um modelo de análise de variância, devemos especificá-lo como um fator, se quisermos obter o resultado correto desta análise. Caso isso não seja feito, o bloco será tratado como uma covariável no modelo de análise de variância, como se fosse uma variável regressora e o seu efeito possuirá apenas 1 grau de liberdade e não 3, como deve ser esse caso. Para realizarmos essa transformação, utilizamos os seguintes códigos:

```
> # transforma bloco em fator e o checa
> dadfict$bl <- as.factor(dadfict$bl) # transforma bl em um fator
> is.factor(dadfict$bl)               # verifica se bl agora é fator

[1] TRUE

> dadfict$bl                          # imprime bloco

 [1] 1 1 1 2 2 2 3 3 3 4 4 4
Levels: 1 2 3 4
```

Neste caso, bloco foi transformado em um fator com 4 níveis, conforme é mostrado na impressão do objeto. Desse ponto em diante é possível utilizar bloco como um efeito na análise de variância sem que isso cause

erros ou resultados imprevistos e indesejados. A variável tratamento é lida automaticamente como um fator. É conveniente salientarmos que arquivos texto com variáveis qualitativas devem ser cuidadosamente preparados. Isso por que o R diferencia níveis escritos em letras maiúsculas de níveis escritos em letras minúsculas. Assim, o nível a é diferente do nível A. Outro aspecto é uso de espaços nos níveis dos fatores qualitativos, que não é permitido nesse tipo de entrada de dados, embora possamos utilizar cedilhas e letras acentuadas.

1.2 Transformações de variáveis

Para obtermos novas variáveis no R a partir de um grupo de variáveis já existentes, não precisamos criá-las fisicamente no arquivo texto. Podemos fazer isso utilizando alguns comandos que irão operar nas colunas do *data frame*, utilizando para isso as funções do R. O objeto R irá conter as variáveis criadas ou transformadas. Podemos utilizar uma série de operadores, sejam eles lógicos ou não. Alguns exemplos destes operadores são: $+$: soma; $-$: subtração; log: logaritmo neperiano; $\log 2$: logaritmo na base 2; $\log 10$: logaritmo na base 10; $*$: multiplicação; $/$: divisão; e $**$ ou $\hat{}$: potenciação do tipo X^Y, que no R é obtido por $X**Y$ ou $X\hat{}Y$. Operadores lógicos como $>$, $(>=)$, $<$, $(<=)$ ou $==$ podem ser usados também. Estruturas condicionais *if* (*cond*) *else* são permitidas, entre outras.

Apresentamos na sequência exemplos que utilizam transformações de variáveis, para ilustrarmos os procedimentos. Os dados que foram considerados são os dados do objeto coelhos.

```
> # cria variáveis a partir do objeto coelhos
> coelhos$sqrtp <- coelhos$peso**0.5 # recebe raiz quadrada de peso
> coelhos$pln <- log(coelhos$peso)   # recebe ln de peso
> # classe = 1 se peso<2.55 e 2 c.c.
> coelhos$classe <- rep(1,times=length(coelhos$peso))
> coelhos$classe[coelhos$peso>=2.55]=2
> coelhos # imprime o data frame modificado

   peso    sqrtp       pln classe
1  2.50 1.581139 0.9162907      1
2  2.58 1.606238 0.9477894      2
3  2.60 1.612452 0.9555114      2
4  2.62 1.618641 0.9631743      2
```

```
5  2.65 1.627882 0.9745596      2
6  2.66 1.630951 0.9783261      2
7  2.58 1.606238 0.9477894      2
8  2.70 1.643168 0.9932518      2
9  2.55 1.596872 0.9360934      2
10 2.57 1.603122 0.9439059      2
11 2.70 1.643168 0.9932518      2
12 2.62 1.618641 0.9631743      2
13 2.59 1.609348 0.9516579      2
14 2.54 1.593738 0.9321641      1
15 2.53 1.590597 0.9282193      1
16 2.20 1.483240 0.7884574      1
```

1.3 Ordenamento de dados

Podemos utilizar a função *order* do R para ordenarmos um *data frame*, especificando as variáveis que almejamos utilizar como chaves do processo de ordenação dos valores do conjunto de dados. Podemos ordenar em ordem crescente ou decrescente. Para ordenarmos em ordem crescente, o argumento chave do objeto order deve receber a variável original. Para ordenamento decrescente, devemos utilizar a variável multiplicada por -1, como chave. Podemos utilizar várias chaves, devendo estar separadas por vírgula no argumento da função *order*. O *data frame* recebe o argumento order como primeiro argumento e o segundo argumento separado por vírgula, deve ser vazio. Isso indica para o R que todas as colunas, variáveis, devem ser ordenadas. Em um primeiro exemplo, ordenamos o *data frame coelhos* utilizando como chave primária a variável classe, considerando a ordem crescente. A segunda variável utilizada como chave secundária foi o peso. No entanto, escolhemos ordenar peso de forma decrescente em cada classe. Dai o sinal negativo antes da variável peso.

```
> # ordena coelhos por classe: crescente e peso: decrescente
> coelhos <- coelhos[ order(coelhos$classe,-coelhos$peso),]
> coelhos    # imprime o data frame modificado

   peso    sqrtp       pln classe
14 2.54 1.593738 0.9321641      1
15 2.53 1.590597 0.9282193      1
1  2.50 1.581139 0.9162907      1
16 2.20 1.483240 0.7884574      1
```

```
8  2.70 1.643168 0.9932518      2
11 2.70 1.643168 0.9932518      2
6  2.66 1.630951 0.9783261      2
5  2.65 1.627882 0.9745596      2
4  2.62 1.618641 0.9631743      2
12 2.62 1.618641 0.9631743      2
3  2.60 1.612452 0.9555114      2
13 2.59 1.609348 0.9516579      2
2  2.58 1.606238 0.9477894      2
7  2.58 1.606238 0.9477894      2
10 2.57 1.603122 0.9439059      2
9  2.55 1.596872 0.9360934      2
```

Ilustramos o uso do *order* em um segundo exemplo exemplo, em que a sala de aula da primeira turma do curso foi dividida em dois grupos de acordo com os lugares que os alunos escolhiam para se sentar. Os da bancada da direita foram denominados de grupo 1 e os da esquerda de grupo 2. Foram mensurados os pesos e altura destes alunos. Criamos o arquivo *bancada.txt* e usamos o *order* para ordenar por grupos em ordem crescente e por peso em ordem decrescente dentro de cada grupo, o *data frame* criado a partir do arquivo, que foi denominado de *bancada*.

```
> bancada <- read.table("C:/daniel/Cursos/RCursoTeX/bancada.txt",
+               header=TRUE)
> bancada     # imprime na tela seu conteúdo original

  grupo peso  alt
1     2 72.0 1.80
2     1 48.5 1.58
3     2 88.0 1.80
4     1 86.0 1.83
5     2 62.0 1.72
6     1 79.0 1.69
7     2 95.0 1.93
8     1 53.0 1.60

> # ordena bancada por grupo: crescente e por peso: decrescente
> bancada  <- bancada[ order(bancada$grupo,-bancada$peso),]
> bancada    # imprime o data frame modificado

  grupo peso  alt
4     1 86.0 1.83
```

```
6     1 79.0 1.69
8     1 53.0 1.60
2     1 48.5 1.58
7     2 95.0 1.93
3     2 88.0 1.80
1     2 72.0 1.80
5     2 62.0 1.72
```

1.4 Procedimentos para análise estatística

A estatística se preocupa fundamentalmente em entender estruturas nos dados. Estas estruturas revelam informações úteis para a compreensão dos fenômenos que estamos estudando. O ambiente R propicia um ambiente poderoso para análise de dados e obtenção de gráficos. Vamos utilizar neste material basicamente algumas funções R para realizarmos análises estatísticas. Estas funções ou procedimentos no R são métodos, que executam determinada ação e retornam resultados em *objetos*, que são variáveis, vetores ou matrizes, *data frames* ou listas. Vamos neste material apresentar a lógica de tais procedimentos, suas sintaxes e principalmente vamos enfatizar os métodos estatísticos que estão envolvidos neste procedimento. Vamos procurar também mostrar que o ambiente R é por si só um ambiente de programação poderoso. O programa R fornece ao usuário uma poderosa e flexível linguagem de programação interativa em um ambiente dinâmico. Por ser um programa com orientação a objeto, todas as suas variáveis são tratadas como objetos e podem ser de diferentes tipos básicos, como vetor, matriz, variável escalar, *data frames*, listas, entre outros. A programação é dinâmica por causa do dimensionamento das matrizes e da alocação de memória serem feitos de forma automática.

Vamos utilizar alguns procedimentos do R para efetuarmos análises de estatística básica, análises de regressão linear e regressão não-linear, análises de modelos lineares ordinários e modelos lineares mistos. Poderemos eventualmente utilizar algum outro procedimento específico para realizarmos algumas análises multivariadas.

O R é um programa que consideramos praticamente completo. Vamos neste material abordar situações específicas da estatística para fazermos uma introdução ao sistema R. Não temos de forma alguma a pretensão de que este seja um material de consulta imprescindível, mas que sirva de

um roteiro básico para aqueles que desejam ter uma noção inicial de como efetuar análises estatísticas utilizando o sistema R.

Capítulo 2

Estatística básica no R

O R possui muitos recursos para realizarmos análises estatísticas descritivas de uma amostra de tamanho n. Neste capítulo vamos abordar as principais estatísticas descritivas utilizando o pacote *fBasics* e o pacote BSDA . Vamos ilustrar a obtenção de estimativas pontuais de vários parâmetros, histogramas e estimadores de Kernel. Vamos realizar inferência sobre média de uma população e de dados emparelhados, tanto testes de hipóteses como estimação intervalar e vamos inferir sobre a distribuição de probabilidade dos dados amostrais. Para alguns parâmetros vamos utilizar a linguagem R para construirmos funções capazes de determinar os intervalos de confiança, utilizando a teoria de inferência estatística como referência. Vamos utilizar diferentes recursos dentro do contexto da estatística básica nesse capítulo.

2.1 Estatísticas descritivas

Vamos utilizar basicamente os comandos básicos disponibilizados pelo R e os pacotes *fBasics* e *BSDA* para obtermos estatísticas descritivas de uma população. Vamos supor que temos uma população com parâmetros desconhecidos e vamos considerar, inicialmente, que essa população possui uma determinada distribuição de probabilidade, que é o modelo probabilístico normal. A função densidade de probabilidade do modelo normal é dada por:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{2.1.1}$$

em que os parâmetros μ e σ^2 são a média e a variância respectivamente.

Este modelo é simétrico em relação à média. Um parâmetro usado para medir a simetria da distribuição é o coeficiente de assimetria que pode ter dois estimadores, o estimador beta e o estimador gama. No R o estimador gama de simetria é obtido e o seu valor de referência na distribuição normal é o valor 0. Este estimador (Ferreira, 2005) é dado por:

$$g_1 = \frac{m_3\sqrt{n(n-1)}}{(n-2)m_2^{3/2}}, \tag{2.1.2}$$

em que $m_r = \sum_{i=1}^{n}(X_i - \bar{X}.)^r/n$ é o estimador de centrado de momento de ordem r, sendo $r \geq 2$. O estimador beta do coeficiente de assimetria é dado por

$$\sqrt{b_1} = \frac{m_3}{m_2^{3/2}}. \tag{2.1.3}$$

O coeficiente de curtose populacional da distribuição normal tem como referência o valor zero, se for considerado o estimador gama ou o valor 3 se for considerado o estimador beta. O coeficiente de curtose mede o grau de achatamento da curva. Como o R estima somente o parâmetro gama, temos o seguinte estimador do coeficiente de curtose:

$$g_2 = \frac{(n-1)\left[(n+1)m_4 - 3(n-1)m_2^2\right]}{(n-2)(n-3)m_2^2}, \tag{2.1.4}$$

e o estimador beta, cujo parâmetro de referência assume valor 3 na normal, é dado por

$$b_2 = \frac{m_4}{m_2^2}. \tag{2.1.5}$$

Assim uma distribuição com coeficiente de assimetria igual a zero é considerada simétrica; se o coeficiente de assimetria for maior que zero, esta

distribuição será assimétrica à direita e se for menor que zero, assimétrica à esquerda. Da mesma forma uma distribuição com coeficiente de curtose igual a 0 ou 3, conforme o estimador que estiver sendo utilizado, será considerada mesocúrtica; se o coeficiente de curtose for negativo ou menor do que 3, será considerada platicúrtica e se for maior que zero ou maior do que 3, será considerada leptocúrtica.

Caracterizada a distribuição, o interesse se volta para a locação e a dispersão da distribuição da população. A média amostral é dada por:

$$\bar{X}_{\cdot} = \frac{1}{n}\sum_{i=1}^{n} X_i. \tag{2.1.6}$$

A variância amostral é dada por:

$$S^2 = \frac{1}{n-1}\left[\sum_{i=1}^{n} X_i^2 - \frac{\left(\sum_{i=1}^{n} X_i\right)^2}{n}\right]. \tag{2.1.7}$$

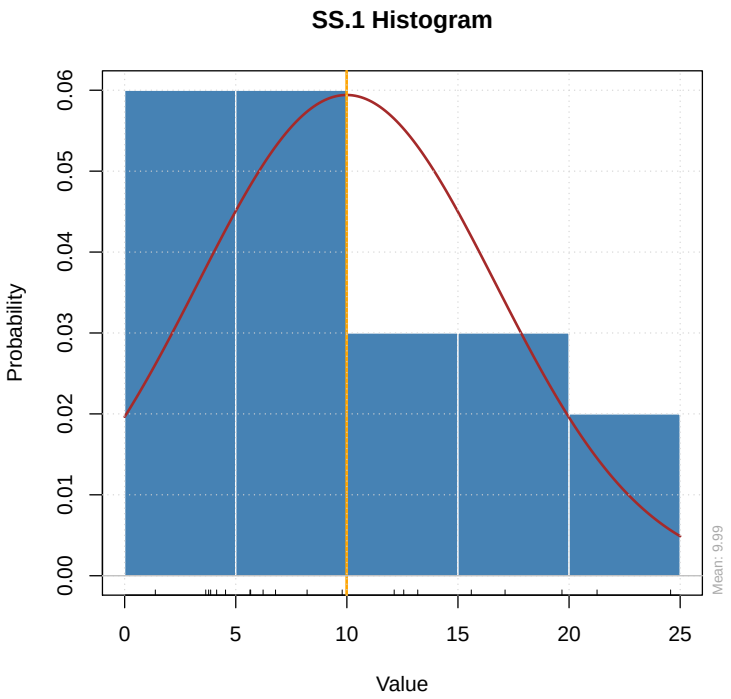
O R estima ainda várias outras estatísticas descritivas, como o desvio padrão S, o erro padrão da média $S_{\bar{X}}$, a mediana m_d, a soma dos dados, alguns percentis entre outras estimativas. Podemos utilizar o pacote *fBasics* e a função *basicStats* para esta finalidade. O *summary* é outra opção que temos para obtermos estatísticas descritivas simples, como os valores mínimo e máximo, a mediana, o primeiro e terceiro quartis e a média. Esse pacote possui a função *histPlot* para estimarmos o histograma, bem como de permitir um ajuste da distribuição normal a este histograma, indicando a média da normal ajustada (média amostral) e a mediana. Este pacote tem uma série de funções para ajustes de algumas funções densidades (*nFit*, *tFit*, *nigFit*, *ghtFit*) para que a normal, t de Student, normal invertida e t GH assimétrica. Possui funções para estimar o a densidade por meio de um estimador Kernel de densidades, sendo plotados os seus gráficos, *densityPlot*. Calcula ainda gráficos de probabilidade acumuladas e os qqplots para a distribuição normal. Na sequência apresentamos os principais funções do pacote *fBasics*, descrevendo algumas de suas opções.

Vamos ilustrar a utilização dessas funções por intermédio de um conjunto de dados de feijão, em que foram avaliadas as produtividades em g/planta de $n = 20$ plantas da geração F_2. Neste programa optamos por apresentar o histograma e as estatísticas descritivas.

```
> # lê o arquivo e atribui ao objeto feijao
> feijao  <- read.table("C:/daniel/Cursos/RCursoTeX/feijao.txt",
+               header=TRUE)
> feijao    # imprime na tela seu conteúdo

    prod
1   1.38
2   3.65
3   3.78
4   3.87
5   4.14
6   4.54
7   5.64
8   5.67
9   6.23
10  6.79
11  8.21
12  9.79
13 12.13
14 12.56
15 13.19
16 15.60
17 17.12
18 19.68
19 21.26
20 24.57

> summary(feijao$prod) # comando básico do R

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.38    4.44    7.50    9.99   13.79   24.57

> library(fBasics)     # carrega o pacote fBasics
> basicStats(feijao$prod, ci = 0.95) # Obtenção de est. Desc.

            X..feijao.prod
nobs             20.000000
NAs               0.000000
```

```
Minimum          1.380000
Maximum         24.570000
1. Quartile      4.440000
3. Quartile     13.792500
Mean             9.990000
Median           7.500000
Sum            199.800000
SE Mean          1.501456
LCL Mean         6.847416
UCL Mean        13.132584
Variance        45.087421
Stdev            6.714717
Skewness         0.670656
Kurtosis        -0.861216
```

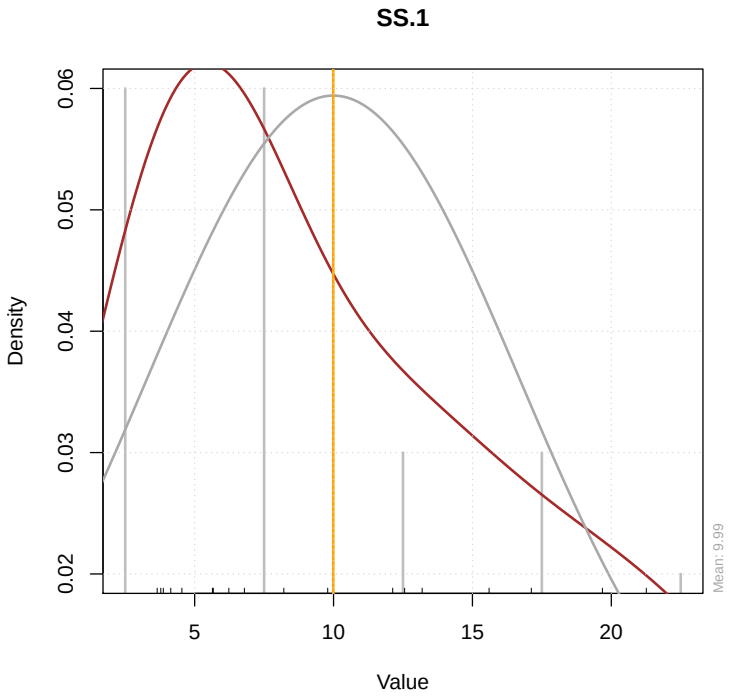
```
> histPlot(as.timeSeries(feijao$prod)) # histograma pelo pacote fBasics
```



Uma observação que devemos fazer, refere-se a forma com que o R estima os coeficientes de curtose e assimetria. Ele utiliza uma variação do estimador beta. Se aplicarmos as fórmulas (2.1.3) e (2.1.5) encontraremos resultados diferentes. Isso porque o R utiliza o estimador de momento da variância (de ordem 2) não viesado, ou seja, com divisor $n-1$ ao invés de n.

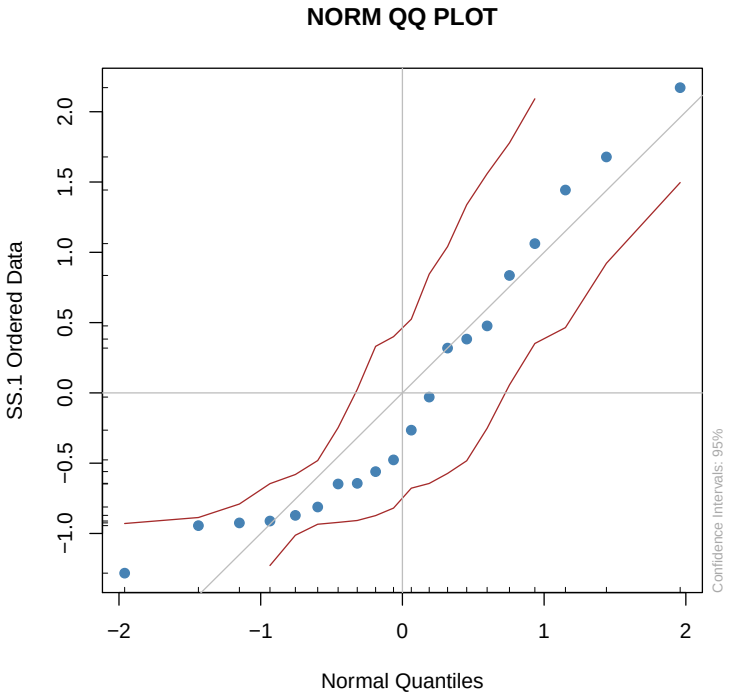
Além disso, para o coeficiente de curtose, ele considera o desvio do valor em relação ao valor da normal padrão, que é 3. No entanto, o numerador dos estimadores continuam sendo definidos em termos de seu estimador viesado, ou seja, divisor n. Assim, se compararmos com algum outro programa de análise estatística não encontraremos resultado equivalente no R. Apesar de apresentarem poucas diferenças, podemos dizer, nesse caso específico, que não concordamos com o estimador apresentado no R. A seguir apresentamos o estimador kernel da densidade, considerando o valor padrão da função para o parâmetro de suavização. Além da função *densityPlot* do pacote *fBasics*, podemos utilizar a função kernel diretamente do escopo principal do R, ou seja, sem a necessidade de utilizarmos um pacote específico.

```
> # estimador kernel da densidade
> densityPlot(as.timeSeries(feijao$prod))
```



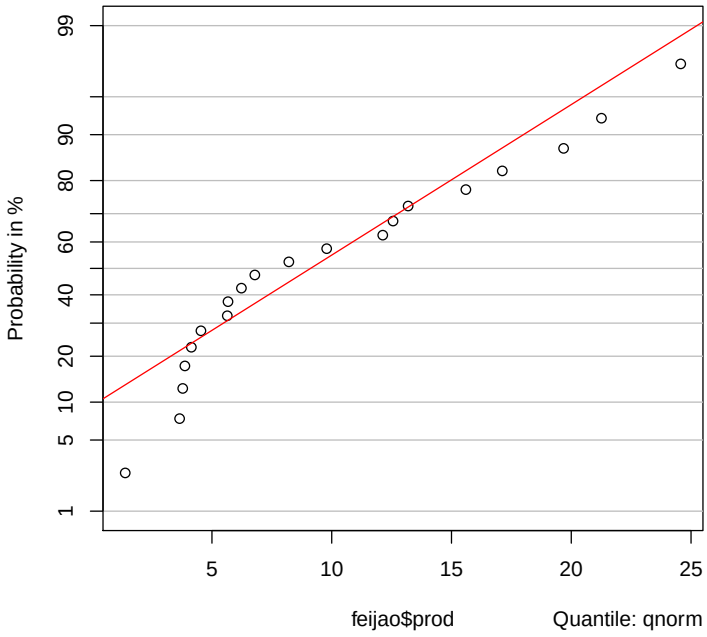
Também traçamos o gráfico dos quantis, utilizando o comando *qqnormPlot*. Automaticamente a este gráfico são adicionados as curvas dos limites de 95% de confiança.

```
> # gráfico dos quantis-quantis da normal
> qqnormPlot(feijao$prod)
```



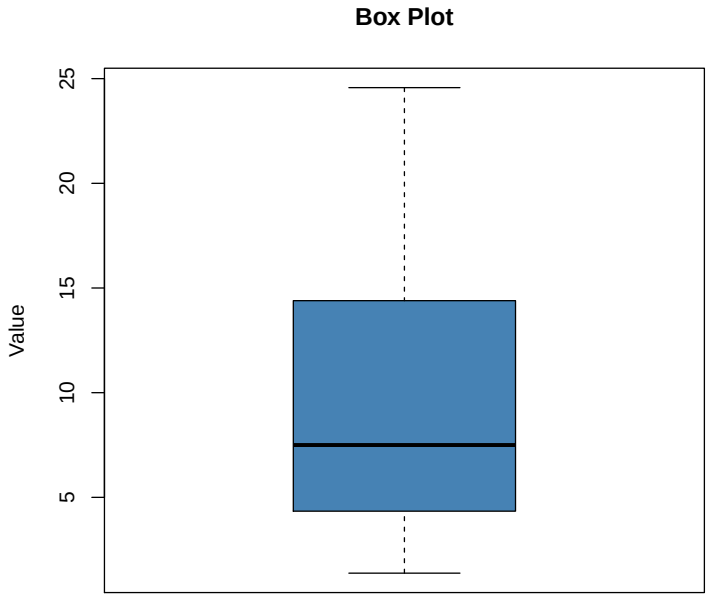
Também traçamos os gráficos das probabilidades acumuladas , utilizando os comandos *probplot* do pacote *e1071*. Devemos instalar o pacote, utilizando a opção do menu pacotes, instalar pacotes (selecionamos o site espelho) e, da lista apresentada, selecionamos o pacote pretendido. Para isso devemos estar conectados na internet. Caso isso não seja possível para a máquina que está sendo utilizada, podemos baixar os pacotes em outro local indo ao site do R/CRAN. Levamos por algum meio magnético os pacotes baixados e instalamos no R, escolhendo a opção instalar pacotes de uma pasta local compactada. O uso tanto do pacote quanto da função está apresentado a seguir. Existem vários tipos de gráficos de probabilidades. Neste foram plotadas as probabilidades acumuladas e esperadas da normal na ordenada e os quantis observados, na abscissa. Em alguns casos, escolhemos para a abscissa os valores das probabilidades acumuladas observadas, ou seja, estimativas a partir dos dados ordenados considerando as médias e variâncias amostrais como parâmetros da distribuição pretendida.

```
> # gráfico das  probab. acumuladas: probabilidades
> # na ordenada e quantis observados na abscissa
> library(e1071)  # carregando o pacote
> probplot(feijao$prod) # chamando a função almejada
```



Finalmente, vamos apresentar os *boxplots*, que são gráficos descritivos de muita utilidade, que permitem uma avaliação da forma geral da distribuição dos dados. Eles são representados por uma linha central que é a mediana e delimitado por um retângulo definidos pelos $1^{\underline{0}}$ e $3^{\underline{0}}$ quartis. Seus limites máximos são os valores mínimo e máximo, observados na amostra. Neste caso a mediana foi igual a 7,5, o primeiro quartil foi 4,4, o terceiro quartil 13,8, o valor mínimo 1,4 e o valor máximo 24,6. Devemos chamar a atenção para a forma como os *outliers* dominam esses gráficos. Isso é um aspecto positivo desse tipo de gráfico, ou seja, nos possibilita detectar os *outliers* facilmente. A função *boxplot*, possui uma série de opções que podem ser consultadas na vasta documentação do pacote *graphics* do R. Veja que estamos nos referindo a função *boxplot* e não a *boxPlot*, do pacote *fBasics*, que utilizamos a seguir. Assim, o usuário que pretende plotar gráficos deste tipo, mas com opções mais adequadas e específicas ao seu problema, deverá utilizar a função *boxplot*.

```
> # gráfico box-plot padrão utilizando fBasics
> boxPlot(feijao$prod) # chamando a função almejada
```



Ao observamos todos esses resultados, podemos verificar que embora as evidências descritivas não sejam muito fortes, não parece haver uma boa concordância da distribuição dos dados amostrais com a distribuição normal. Testes formais precisam ser feitos para que haja ou não uma confirmação dessas evidências descritivas. Um outro comentário simples que gostaríamos de fazer neste instante diz respeito à forma que devemos sumariar os resultados descritivos de posição e dispersão em um trabalho científico. Em geral, se a distribuição é simétrica utilizamos a média como medida de posição. Associada a essa medida de posição devemos apresentar uma de dispersão. Podemos escolher o desvio padrão ou o erro padrão, conforme o objetivo do trabalho. Se queremos retratar a variabilidade dos dados populacionais em relação a média desta população, devemos utilizar o desvio padrão como uma estimativa desta medida. O coeficiente de variação também pode ser utilizado se pretendemos apresentar esta variabilidade em uma escala relativa e não absoluta. Se por outro lado desejamos caracterizar a precisão com que a média populacional foi estimada, ou seja, a precisão da estimativa obtida, deveremos reportar o erro padrão da média. Os gráficos são muito informativos e esclarecedores. Uma imagem

gráfica diz mais do que muitas palavras. Sua utilização em uma análise exploratória é essencial.

A forma como essas medidas devem ser apresentadas também é alvo de muita polêmica no meio científico. Muitas críticas surgem quando apresentamos em uma tabela ou no texto, os resultados por $\bar{X}_{\cdot} \pm S$ ou por $\bar{X}_{\cdot} \pm S_{\bar{X}_{\cdot}}$. O uso do $\pm$ é muito criticado, pois gera ambiguidade dos resultados e das interpretações. Isso porque ao encontrarmos tal símbolo, podemos presumir que o resultado se trata de um intervalo de confiança, o que não é verdade. Assim, é preferível que os resultados sejam apresentados por $\bar{X}_{\cdot}(S)$ ou por $\bar{X}_{\cdot}\left(S_{\bar{X}_{\cdot}}\right)$. Em ambos os casos deve ficar claro para o leitor que se trata da estimativa da média seguida, entre parênteses, pelo desvio padrão ou pelo erro padrão. Não temos restrições ao uso particular de um desses estimadores: coeficiente de variação, desvio padrão ou erro padrão. Isto porque podemos calcular a partir de um deles os demais. Então se torna preponderante a apresentação do tamanho da amostra n utilizado no experimento ou no levantamento amostral (Ferreira, 2005).

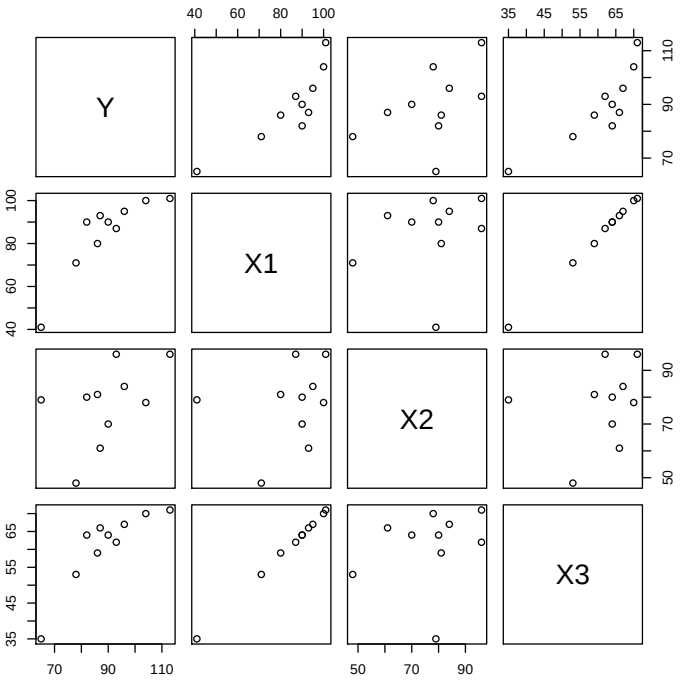
Podemos enfatizar ainda, que além das estatísticas descritivas apresentadas pela função *basicStats*, também foi obtido o intervalo de confiança de 95% para a média populacional, supondo normalidade dos dados. Definimos a confiança do intervalo com a opção $ci = 0.95$, argumento da função, aplicada ao objeto feijao\$prod (variável), *basicStats(feijao\$prod, ci = 0.95)*. Obtivemos os resultados $LCL\ \ Mean = 6{,}847416$ e $UCL\ \ Mean = 13{,}132584$, que são os limites inferior e superior de 95% de confiança. A omissão da opção $ci = 0.95$, irá resultar sempre em um intervalo de 95%. Então a sua aplicação foi realizada apenas para indicar ao leitor como proceder para alterar este coeficiente de confiança, quando ele precisar disso.

Para *data frames* mais complexos, com mais de uma coluna, o uso do *summary* é bastante interessante, bem como a aplicação do comando *plot* ao objeto como um todo. Vejamos o exemplo a seguir dessas duas funções aplicadas ao *data frame arvores*. Verificamos que o R aplica o comando para cada coluna, variável, do *data frame arvores*. Da mesma forma um gráfico cruzado das observações de todos os pares de variáveis é realizado. Isso é bastante informativo, para avaliarmos possíveis relações entre as variáveis do conjunto de dados. Nesse caso, a variável X_1 se correlaciona positiva e fortemente com a variável X_3. A variável X_2, possui correlações lineares baixas com as demais variáveis do arquivo de dados.

```
> # comandos summary e plot aplicados a um data frame mais complexo
> summary(arvores) # chamando a função summary
```

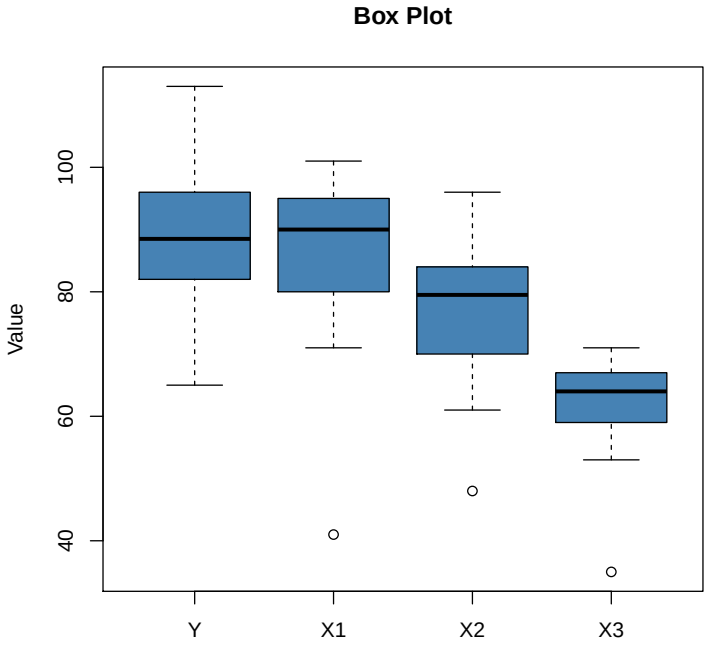
```
       Y                X1               X2
 Min.   : 65.00   Min.   : 41.00   Min.   :48.00
 1st Qu.: 83.00   1st Qu.: 81.75   1st Qu.:72.00
 Median : 88.50   Median : 90.00   Median :79.50
 Mean   : 89.40   Mean   : 84.80   Mean   :77.30
 3rd Qu.: 95.25   3rd Qu.: 94.50   3rd Qu.:83.25
 Max.   :113.00   Max.   :101.00   Max.   :96.00
       X3
 Min.   :35.00
 1st Qu.:59.75
 Median :64.00
 Mean   :61.10
 3rd Qu.:66.75
 Max.   :71.00
```

```
> plot(arvores)    # gráfico conjunto de todas variáveis
```



Podemos aplicar também a função *boxPlot* ao conjunto de dados completo, para esses casos em que temos mais de uma coluna numérica no *data frame*. Os resultados obtidos são ilustrados na sequência.

```
> # comando boxPlot aplicado a um data frame complexo
> boxPlot(arvores) # chamando a função boxPlot de fBasics
```



Para um conjunto mais completo de estatísticas descritivas aplicadas a cada coluna numérica do conjunto de dados, podemos utilizar a função *basicStats* ao *data frame*. Os resultados desse procedimento ilustrativo do comando estão apresentados na sequência.

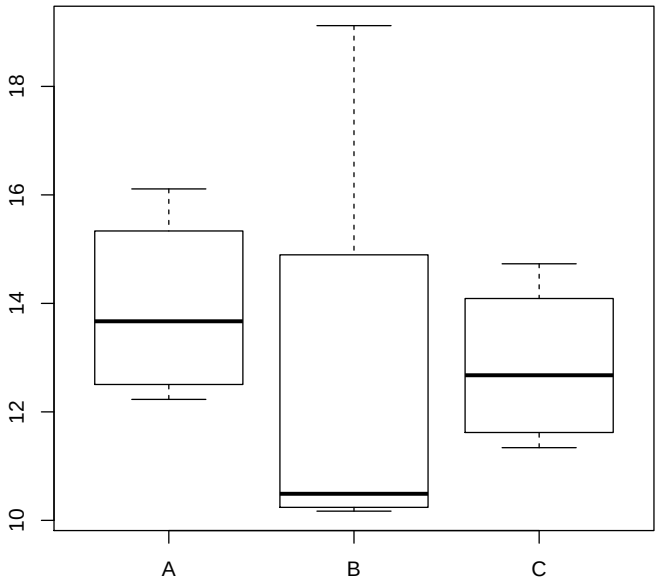
```
> # comandos basicStats aplicados a um data frame complexo
> basicStats(arvores) # chamando a função basicStats de fBasics
```

	Y	X1	X2
nobs	10.000000	10.000000	10.000000
NAs	0.000000	0.000000	0.000000
Minimum	65.000000	41.000000	48.000000
Maximum	113.000000	101.000000	96.000000
1. Quartile	83.000000	81.750000	72.000000
3. Quartile	95.250000	94.500000	83.250000
Mean	89.400000	84.800000	77.300000
Median	88.500000	90.000000	79.500000
Sum	894.000000	848.000000	773.000000
SE Mean	4.248398	5.632840	4.650090
LCL Mean	79.789455	72.057630	66.780766

```
UCL Mean     99.010545  97.542370  87.819234
Variance    180.488889 317.288889 216.233333
Stdev        13.434615  17.812605  14.704874
Skewness      0.005117  -1.364156  -0.516365
Kurtosis     -0.752301   0.830545  -0.738799
                    X3
nobs         10.000000
NAs           0.000000
Minimum      35.000000
Maximum      71.000000
1. Quartile  59.750000
3. Quartile  66.750000
Mean         61.100000
Median       64.000000
Sum         611.000000
SE Mean       3.341490
LCL Mean     53.541024
UCL Mean     68.658976
Variance    111.655556
Stdev        10.566719
Skewness     -1.379219
Kurtosis      0.899760
```

No caso de uma objeto que temos um fator e queremos obter os *boxplots* para cada nível do fator, podemos utilizar o comando R de modelagem do tipo *boxplot(Y∼fator)*. Isso faz com que o R gere os gráficos individuais para cada nível do fator, mas colocados conjuntamente em um único ambiente gráfico. Vamos ilustrar para o caso do *data frame dadfict*, apresentado no capítulo 1. O programa para isso é apresentado na sequência. Temos nesse caso o *boxplot* para cada nível de tratamento, A, B e C.Verificamos que o tratamento B, possui a menor mediana, mas possui a maior variação dos três níveis do fator. Este gráfico é interessante, de uma maneira geral, pois possibilita detectarmos algum problema de *outlier*, heterogeneidade de variância, entre outros problemas.

```
> # gráfico box-plot padrão em função dos níveis do fator trat
> boxplot(dadfict$prod~dadfict$trat) # chamando a função almejada
```



2.2 Estimação de Parâmetros

Vamos apresentar vários procedimentos para estimação dos principais parâmetros de uma população. Nesta seção vamos considerar a estimação de média, proporção, variância, desvio padrão, coeficiente de variação e diferenças de médias.

2.2.1 Estimação de Médias, Desvio Padrão e Variâncias

Vamos apresentar a função R para estimação intervalar de médias de uma população normal. Para isso vamos utilizar a função *basicStats*. Neste caso utilizamos a opção *ci=0.95* (*confidence interval*). O valor especificado é o *default* do coeficiente de confiança que é dado por $1 - \alpha$. O intervalo de confiança para a média de uma normal é dado por:

$$IC_{1-\alpha}(\mu) : \bar{X}_{\cdot} \pm t_{\alpha/2;\nu} \frac{S}{\sqrt{n}}, \tag{2.2.1}$$

em que $t_{\alpha/2;\nu}$ é o quantil superior $100\alpha/2\%$ da distribuição t de Student com $\nu = n - 1$ graus de liberdade.

O programa R, criado para realizarmos a estimação por intervalo para a média de uma população normal, considerando os dados de feijão como exemplo, está apresentado na sequência. Vamos considerar que os desvios de normalidade observados anteriormente para este conjunto de dados são atribuídos somente ao acaso. Vamos a partir deste instante fazer algumas simplificações nos programas, apresentando somente a parte contendo os comandos de interesse e omitindo a parte de entrada de dados. Só apresentaremos a parte de entrada de dados quando se tratar de conjuntos de valores que ainda não foram descritos anteriormente. O programa simplificado é:

```
> # comandos basicStats aplicados ao
> # data frame feijao para obter o IC 95%
> basicStats(feijao, ci=0.95) #chamando a função basicStats

                 prod
nobs        20.000000
NAs          0.000000
Minimum      1.380000
Maximum     24.570000
1. Quartile  4.440000
3. Quartile 13.792500
Mean         9.990000
Median       7.500000
Sum        199.800000
SE Mean      1.501456
LCL Mean     6.847416
UCL Mean    13.132584
Variance    45.087421
Stdev        6.714717
Skewness     0.670656
Kurtosis    -0.861216
```

Os valores estimados para o LI e LS foram 6,85 e 13,13, indicando que a média em g/planta da produtividade da população com 95% de confiança deve estar neste intervalo. Se por outro lado, considerarmos a distribuição dos dados de produção como não normais, então podemos pensar em outras alternativas. Uma possibilidade é realizarmos uma análise *bootstrap*.

Existe uma função R denominada *boot.ci* do pacote *boot*, que nos permite realizar tal tarefa. Podemos especificar cinco métodos diferentes de intervalos *bootstrap*, que são: aproximação normal de primeira ordem; intervalo básico de *bootstrap*; intervalo *bootstrap* baseado na *estudentização*; o intervalo percentil *bootstrap*; e o intervalo percentil com correção de viés (BCa) (Ferreira, 2005). O uso desta função não é trivial, mas tentaremos explicar de uma forma bastante simples. Ela exige que seja criada uma função dos dados, para a qual queremos obter o intervalo de confiança *bootstrap*. Essa função tem de ter como primeiro argumento os dados e um segundo argumento uma variável peso. A média é a função que iremos criar. Se os pesos de cada observação forem iguais a 1, a média será obtida multiplicando cada dado amostral pelo peso e dividindo o resultado pela soma dos pesos. Então esta será nossa função R. O segundo passo, exige que chamemos uma função denominada *boot*, que deve receber como argumentos os dados, a função criada, o número de simulações *bootstrap*, o tipo de argumento que o segundo parâmetro de nossa função significa, no caso receberá o valor "w", indicando que trata-se de pesos e, finalmente, o tipo de simulação que utilizaremos, que será a comum (*ordinary*). Por último, invocamos a função *boot.ci* com os seguintes argumentos: o objeto que recebeu o resultado da função *boot* no segundo passo, os coeficientes de confiança e os tipos de *bootstrap*. Para os tipos podemos usar "all", ou especificar cada um eles por *type = c("norm","basic", "stud", "perc", "bca")*. O programa R e os resultados de 999 repetições *bootstrap* estão apresentados a seguir.

```
> # comandos para obter IC bootstrap do
> # data frame feijao. IC de 90% e 95%
> library(boot)  # certificando que o pacote foi carregado
> media <- function(x, w) sum(x*w)/sum(w)
> feijao.boot <- boot(feijao$prod, media, R = 999,
+                     stype = "w",sim = "ordinary")
> boot.ci(feijao.boot, conf = c(0.90,0.95),
+          type = c("norm","basic","perc", "bca"))

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 999 bootstrap replicates

CALL :
boot.ci(boot.out = feijao.boot, conf = c(0.9, 0.95), type = c("norm",
    "basic", "perc", "bca"))

Intervals :
Level      Normal              Basic
```

```
90%   ( 7.595, 12.371 )   ( 7.533, 12.316 )
95%   ( 7.137, 12.829 )   ( 6.887, 12.632 )

Level     Percentile            BCa
90%   ( 7.664, 12.447 )   ( 7.787, 12.661 )
95%   ( 7.348, 13.093 )   ( 7.490, 13.314 )
Calculations and Intervals on Original Scale
```

Os resultados obtidos para 95% forma muito parecidos com o intervalo assumindo normalidade e utilizando a distribuição t de Student. O resultado mais parecido foi o do intervalo BCa, que, em geral, é considerado mais adequado. A grande vantagem deste procedimento é podermos utilizá-lo em funções complicadas dos dados, para as quais não temos a menor ideia da sua distribuição amostral. Isso pode acontecer até mesmo para situações em que os dados originais sejam normais, dependendo da função dos dados que temos interesse em estimar. Maiores detalhes podem ser obtidos na documentação da função.

Também podemos utilizar o R para obtermos intervalos de confiança para o desvio padrão e variância de uma população normal. Infelizmente, não encontramos uma função pronta para realizarmos esta tarefa, o que não significa que não tenha. Assim, iremos programar nossa própria função. Essa será uma oportunidade para aprendermos como utilizar o R como um ambiente de programação. O intervalo de confiança para a variância de uma população normal é dado por:

$$IC_{1-\alpha}(\sigma^2) : \left[\frac{(n-1)S^2}{\chi^2_{\alpha/2;\nu}}; \frac{(n-1)S^2}{\chi^2_{1-\alpha/2;\nu}} \right], \tag{2.2.2}$$

em que $\chi^2_{\alpha/2;\nu}$ e $\chi^2_{1-\alpha/2;\nu}$ são os quantis superiores $100\alpha/2\%$ e $100(1-\alpha/2)\%$ da distribuição qui-quadrado com $\nu = n-1$ graus de liberdade, respectivamente.

O intervalo de confiança para o desvio padrão populacional (σ) é obtido calculando a raiz quadrada dos limites do intervalo de confiança para variância. A função R para obtenção destes intervalos e um exemplo de sua utilização, utilizando os dados do feijão, são dados por:

```
> # função que retorna os valores IC para variâncias
> # e desvios padrões normais para nível de confiança 1-alpha
> CI_var_sd <- function(x,alpha=0.05)
```

```
+ {
+    # proteção para alpha, que assume valor default
+    # se estiver fora ]0,1[
+    if ((alpha<=0) | (alpha>=1.0)) alpha <- 0.05
+    df     <- length(x)-1
+    s2     <- var(x)
+    CI.var <- c(df*s2/qchisq(c(alpha/2, 1-alpha/2),
+                df, lower.tail=FALSE))
+    CI.sd  <- sqrt(CI.var)
+    return(list(CI.var=CI.var, CI.sd=CI.sd,
+                CL=1-alpha))
+ }
> # exemplos de uso -  dados feijao
> CI_var_sd(feijao$prod,0.05)

$CI.var
[1] 26.07611 96.18362

$CI.sd
[1] 5.106478 9.807325

$CL
[1] 0.95
```

Da mesma forma se a amostra não for normal, podemos utilizar os intervalos de confiança bootstrap para variância e para o desvios padrão populacionais. Para obtermos o programa R, utilizamos o pacote *boot* e as três funções necessárias para a obtenção do intervalo de confiança requerido. Os resultados obtidos foram apresentados a seguir, com ilustrações dos dados de feijão.

```
> # comandos para obter IC bootstrap do
> # data frame feijao. IC de 90% e 95%
> # para variância
> library(boot)  # certificando que o pacote foi carregado
> varianc <- function(x, w)
+ {
+    n    <- length(x)
+    varp <- sum(x*x*w) - sum(x*w)^2/sum(w)
+    return(varp*n/(n-1))
+ }
> varianc.boot <- boot(feijao$prod, varianc, R = 999,
+                      stype = "w",sim = "ordinary")
> boot.ci(varianc.boot, conf = c(0.90,0.95),
+          type = c("norm","basic","perc", "bca"))
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 999 bootstrap replicates

CALL :
boot.ci(boot.out = varianc.boot, conf = c(0.9, 0.95), type = c("norm",
    "basic", "perc", "bca"))

Intervals :
Level      Normal              Basic
90%   (28.81, 67.26 )   (28.56, 67.19 )
95%   (25.13, 70.94 )   (24.76, 70.17 )

Level     Percentile            BCa
90%   (22.98, 61.61 )   (30.67, 71.90 )
95%   (20.00, 65.41 )   (27.87, 78.87 )
Calculations and Intervals on Original Scale
Some BCa intervals may be unstable
```

```
> # para o desvio padrão
> sdv <- function(x, w)
+ {
+    n    <- length(x)
+    varp <- sum(x*x*w) - sum(x*w)^2/sum(w)
+    return((varp*n/(n-1))^0.5)
+ }
> sdv.boot <- boot(feijao$prod, sdv, R = 999,
+                      stype = "w",sim = "ordinary")
> boot.ci(sdv.boot, conf = c(0.90,0.95),
+          type = c("norm","basic","perc", "bca"))
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 999 bootstrap replicates

CALL :
boot.ci(boot.out = sdv.boot, conf = c(0.9, 0.95), type = c("norm",
    "basic", "perc", "bca"))

Intervals :
Level      Normal              Basic
90%   ( 5.453,  8.405 )   ( 5.565,  8.481 )
95%   ( 5.170,  8.688 )   ( 5.389,  8.880 )

Level     Percentile            BCa
90%   ( 4.948,  7.864 )   ( 5.463,  8.276 )
95%   ( 4.550,  8.040 )   ( 5.204,  8.544 )
Calculations and Intervals on Original Scale
Some BCa intervals may be unstable
```

2.2.2 Estimação de Proporções

Para estimarmos por intervalo proporções binomiais podemos utilizar a aproximação normal em grandes amostras e o intervalo de confiança exato. Estes métodos são disponibilizados no pacote *binom*, função *binom.confint*, do programa R. Dada uma amostra de tamanho n de eventos Bernoulli independentes e com probabilidade de sucesso constante p, em que exatamente y sucessos foram observados, o intervalo de confiança normal aproximado para p é dado por:

$$IC_{1-\alpha}(p) : \hat{p} \pm z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}, \tag{2.2.3}$$

em que $\hat{p} = y/n$ é estimador pontual de p e $z_{\alpha/2}$ é o quantil superior $\alpha/2$ da distribuição normal padrão.

O intervalo de confiança exato para as proporções binomiais deve ser utilizado principalmente se n for pequeno e se p se afastar muito de $1/2$. Este intervalo é baseado na relação da binomial com a beta incompleta e portanto com a distribuição F. O intervalo de confiança exato para as proporções binomiais é dado por:

$$IC_{1-\alpha}(p) : \left[\frac{1}{1+\dfrac{(n-y+1)F_{\alpha/2;2(n-y+1),2y}}{y}}; \frac{1}{1+\dfrac{n-y}{(y+1)F_{\alpha/2;2(y+1),2(n-y)}}}\right], \tag{2.2.4}$$

em que $F_{\alpha/2;\nu_1,\nu_2}$ é o quantil superior $100\alpha/2\%$ da distribuição F com ν_1 e ν_2 graus de liberdade.

O programa R, utilizando a função *binom.confint* do pacote *binom*, onde o usuário deve trocar os valores de y e de n apresentados, conforme forem os resultados de sua pesquisa. O valor de α também deve ser alterado se tivermos interesse em outro coeficiente de confiança do que aquele apresentado. Ilustramos uma situação, em que foram amostradas $n = 2401$ plantas F_2 e $y = 1061$ possuíam flores brancas e $n - y = 1340$ flores violetas. Assumindo que a herança da cor de flor em feijão é monogênica e o alelo que confere cor violeta é dominante sobre o alelo que confere cor branca, estimar a proporção de flores branca por intervalo e verificar se o valor paramétrico esperado sob a hipótese de herança monogênica com do-

minância do alelo e cor violeta sobre o alelo de flor de cor branca, que é de 0,25, é um valor plausível para ser a verdadeira proporção binomial populacional. Utilizamos a função escolhendo os métodos exato e aproximado, considerando a aproximação normal. Em grandes amostras, como é o caso, os intervalos aproximados, incluindo outras aproximações não ilustradas, e exato se aproximam.

```
> # IC de 95% para proporções, utilizando
> # o pacote binom e a função  binom.confint
> # métodos exato e aproximado
> library(binom)  # certificando que o pacote foi carregado
> n  <- 2401      # definindo o tamanho da amostra
> y  <- 1061      # definindo o número de sucessos do evento
> CL <- 0.95      # definindo o nível de confiança
> binom.confint(y, n, conf.level = CL,
+               methods = c("exact", "asymptotic"))

      method    x    n      mean     lower     upper
1 asymptotic 1061 2401 0.4418992 0.4220351 0.4617634
2      exact 1061 2401 0.4418992 0.4219080 0.4620324
```

Por meio dos resultado apresentados, podemos inferir que a cor de flor no feijão não é explicada pelo mecanismo de herança monogênica, pois o valor 0,25 não é plausível para ser o verdadeiro valor paramétrico da proporção de flores brancas em uma geração F_2. Na verdade, a herança é devida a dois genes independentes, com interação epistática, digamos A e B e que os fenótipos violeta são dados pelos genótipos A_B_ e os de cor branca por A_bb, aaB_ ou aabb. A segregação violeta:branca esperada na geração F_2 é dada por 9 : 7, logo a proporção de flores brancas é de $7/16 = 0{,}4375$, que é um valor plausível, de acordo com os intervalos de 95% de confiança obtidos, corroborando a hipótese de dois genes independentes e com interação epistática.

2.2.3 Estimação de Coeficientes de Variação

Para estimar o intervalo de confiança do coeficiente de variação populacional de uma normal, seja $\hat{\kappa} = S/\bar{X}_{\cdot}$, o estimador do coeficiente de variação. O intervalo aproximado proposto por Vangel (1996) é dado por:

$$IC_{1-\alpha}(\kappa): \begin{cases} LI = \dfrac{\hat{\kappa}}{\sqrt{\left(\dfrac{\chi^2_{\alpha/2}+2}{\nu+1}-1\right)\hat{\kappa}^2+\dfrac{\chi^2_{\alpha/2}}{\nu}}} \\ \\ LS = \dfrac{\hat{\kappa}}{\sqrt{\left(\dfrac{\chi^2_{1-\alpha/2}+2}{\nu+1}-1\right)\hat{\kappa}^2+\dfrac{\chi^2_{1-\alpha/2}}{\nu}}}, \end{cases} \tag{2.2.5}$$

em que $\chi^2_{\alpha/2}$ e $\chi^2_{1-\alpha/2}$ são os quantis superiores $100\alpha/2\%$ e $100(1-\alpha/2)\%$ da distribuição de qui-quadrado com $\nu = n-1$ graus de liberdade.

A distribuição exata do coeficiente de variação amostral é função da distribuição t não central, ou seja, a partir da distribuição amostral da seguinte quantidade, dada por

$$\frac{\sqrt{n}\bar{X}_{\cdot}}{S} \sim t_{NC}\left(n-1, \frac{\mu\sqrt{n}}{\sigma}\right), \tag{2.2.6}$$

podemos realizar inferências para $\kappa = \mu/\sigma$, em que t_{NC} é a distribuição t não-central, com graus de liberdade $n-1$ e parâmetro de não-centralidade $\mu\sqrt{n}/\sigma$.

Assim, para obtermos o intervalo de confiança para o parâmetro κ, temos que resolver a equação não linear nos parâmetros dada por

$$P\left[t_{NC}(\alpha/2) \leq \frac{\sqrt{n}\bar{X}_{\cdot}}{S} \leq t_{NC}(1-\alpha/2)\right] = 1-\alpha \tag{2.2.7}$$

em que $t_{NC}(\alpha/2)$ e $t_{NC}(1-\alpha/2)$ são quantis $\alpha/2$ e $1-\alpha/2$ da distribuição t não-central, ou seja, são os valores da inversa da função de distribuição com argumentos $\alpha/2$ e $1-\alpha/2$, respectivamente, avaliadas no ponto $\sqrt{n}\bar{X}_{\cdot}/S$.

Assim, os valores dos limites de confiança inferior e superior são dados pelas equações não-lineares, relacionadas com a função de distribuição da t

não central. Os valores de μ/σ, que são as soluções das equações apresentadas a seguir são os limites de confiança do intervalo almejado.

$$F_{NC}\left(n-1, \frac{\sqrt{n}}{\sigma/\mu}\right)\left(\frac{\sqrt{n}\bar{X}_.}{S}\right) = \alpha/2 \tag{2.2.8}$$

$$F_{NC}\left(n-1, \frac{\sqrt{n}}{\sigma/\mu}\right)\left(\frac{\sqrt{n}\bar{X}_.}{S}\right) = 1-\alpha/2 \tag{2.2.9}$$

A dificuldade é que a função inversa da distribuição t não-central deve ser obtida, que por si só já é uma tarefa complicada de ser realizada, mas com o agravante dessa função de distribuição inversa depender do parâmetro de não-centralidade, que é função, por sua vez, da quantidade desconhecida σ/μ, que é o coeficiente de variação populacional. A equação não-linear deve ser resolvida para obtermos o intervalo de confiança desejado. Isso, felizmente não precisamos resolver diretamente, pois a função *ci.cv* do pacote *MBESS* já faz isso por nós. Para a aproximação de McKay modificado por por Vangel (1996), tivemos de implementar uma função para resolver o problema. Denominamos a função de *ci.cvvangel*. O programa R, ilustrando o uso destas duas funções está apresentado a seguir. Na função *ci.cv* podemos fornecer a matriz de dados ou a média, o desvio padrão e o tamanho da amostra. Na função que implementamos, somente a segunda opção deve ser utilizada. O exemplo utiliza os dados apresentados no artigo original de Vangen (1996).

```
> ci.cvvangel <- function(mean, sd, n, alpha=0.05)
+ {
+    khat <- sd/mean
+    qui1 <- qchisq(1-alpha/2,n-1)
+    qui2 <- qchisq(alpha/2,n-1)
+    LICV <- khat/(((qui1+2)/n-1)*khat**2+qui1/(n-1))**0.5
+    LSCV <- khat/(((qui2+2)/n-1)*khat**2+qui2/(n-1))**0.5
+    return(list(CV=khat,CV.Lower=LICV, CV.Upper=LSCV,CL=1-alpha))
+ }
> # exemplo de uso
> xbar <- 194.8333
> sd <- 26.2947^0.5
> n <- 6
> alpha <- 0.05
> ci.cvvangel(xbar,sd,n,alpha)
```

```
$CV
[1] 0.02631909

$CV.Lower
[1] 0.01642533

$CV.Upper
[1] 0.06462172

$CL
[1] 0.95
```

```
> # Exemplo para usar o IC exato
> # aplicando a função ci.cv do pacote MBESS
> library(MBESS)
> ci.cv(mean=xbar, sd=sd, n=n, alpha.lower=.025,
+       alpha.upper=.025, conf.level=NULL)

$Lower.Limit.CofV
[1] 0.01676212

$Prob.Less.Lower
[1] 0.025

$Upper.Limit.CofV
[1] 0.0651081

$Prob.Greater.Upper
[1] 0.025

$C.of.V
[1] 0.02631909
```

Podemos realizar a estimação por intervalo do coeficiente de variação de uma população qualquer não-normal utilizando o método *bootstrap*. Fizemos isso, utilizando os dados de produtividade das plantas F_2 de feijão. Esse método é bastante interessante e não possui a limitação de assumir normalidade para a distribuição dos dados ou do resíduo.

```
> # comandos para obter IC bootstrap do
> # data frame feijao. IC de 90% e 95%
> # para CV
```

```
> # inicialmente é apresentado o CV exato assumindo normalidade
> ci.cv(data=feijao$prod, alpha.lower=.025,
+       alpha.upper=.025, conf.level=NULL)

$Lower.Limit.CofV
[1] 0.4715071

$Prob.Less.Lower
[1] 0.025

$Upper.Limit.CofV
[1] 1.195831

$Prob.Greater.Upper
[1] 0.025

$C.of.V
[1] 0.6721438

> # utilizando a aproximação Vangen(1996)
> xbar <- mean(feijao$prod)
> sd <- var(feijao$prod)^0.4
> n <- length(feijao$prod)
> alpha <- 0.05
> ci.cvvangel(xbar,sd,n,alpha)

$CV
[1] 0.4592579

$CV.Lower
[1] 0.3344416

$CV.Upper
[1] 0.752109

$CL
[1] 0.95

> # IC bootstrap para o CV
> CVboot <- function(x, w)
+ {
+    n    <- length(x)
+    varp <- sum(x*x*w) - sum(x*w)^2/sum(w)
```

```
+    mean <- sum(x*w)/sum(w)
+    return((varp*n/(n-1))^0.5/mean)
+ }
> CV.boot <- boot(feijao$prod, CVboot, R = 999,
+                       stype = "w",sim = "ordinary")
> boot.ci(CV.boot, conf = c(0.90,0.95),
+          type = c("norm","basic","perc", "bca"))

BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 999 bootstrap replicates

CALL :
boot.ci(boot.out = CV.boot, conf = c(0.9, 0.95), type = c("norm",
    "basic", "perc", "bca"))

Intervals :
Level      Normal              Basic
90%   ( 0.5484,  0.8303 )   ( 0.5472,  0.8280 )
95%   ( 0.5214,  0.8573 )   ( 0.5163,  0.8494 )

Level     Percentile            BCa
90%   ( 0.5163,  0.7971 )   ( 0.5548,  0.8444 )
95%   ( 0.4949,  0.8280 )   ( 0.5291,  0.8647 )
Calculations and Intervals on Original Scale
Some BCa intervals may be unstable
```

Os três intervalos apresentaram intervalos diferentes, sendo o mais amplo o IC normal aproximado, seguido do IC normal exato e do IC *bootstrap*. Os intervalos *bootstrap* são bem estreitos, mas seu desempenho em modelos normais e não normais ainda requerem avaliação. O desempenho pode ser avaliado por meio de simulação Monte Carlo, utilizando como referência a probabilidade de cobertura.

2.2.4 Diferença de Duas Médias Independentes

Esta é uma situação de muito interesse para os pesquisadores, pois é muito comum obter amostras independentes de duas populações. O objetivo é obter o intervalo de confiança para a diferença das médias $\mu_1 - \mu_2$ das duas populações. Algumas suposições são feitas para a utilização dos procedimentos estatísticos tradicionais. Inicialmente pressupomos que ambas as populações possuem distribuição normal com médias μ_1 e μ_2 e variâncias

σ_1^2 e σ_2^2, respectivamente. Ao obtermos as amostras aleatórias de tamanhos n_1 e n_2 das populações 1 e 2, respectivamente, devemos supor independência entre as observações das diferentes amostras e também das observações dentro das duas amostras. Finalmente, supomos que as variâncias das duas populações são homogêneas, ou seja, que $\sigma_1^2 = \sigma_2^2 = \sigma$.

Sejam $\bar{X}_1$ e $\bar{X}_2$ os estimadores das médias da populações 1 e 2 e S_1^2 e S_2^2 os estimadores das variâncias populacionais obtidos em amostras de tamanho n_1 e n_2, respectivamente, então duas situações distintas podem ser consideradas. A primeira quando $\sigma_1^2 = \sigma_2^2$ e a segunda quando $\sigma_1^2 \neq \sigma_2^2$. Estas duas situações estão destacadas na sequência.

a. Se $\sigma_1^2 = \sigma_2^2$: O intervalo de confiança quando as variâncias são homogêneas é dado por:

$$IC_{1-\alpha}(\mu_1 - \mu_2) : \bar{X}_1 - \bar{X}_2 \pm t_{\alpha/2;\nu}\sqrt{S_p^2\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}, \tag{2.2.10}$$

em que $t_{\alpha/2;\nu}$ é o quantil superior $\alpha/2$ da distribuição t de Student com $\nu = n_1 + n_2 - 2$ graus de liberdade e S_p^2 é a variância combinada (*pooled*) dada por:

$$S_p^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}. \tag{2.2.11}$$

b. Se $\sigma_1^2 \neq \sigma_2^2$: Neste caso a distribuição t de Student não é mais exata para obtermos o intervalo de confiança. No entanto, esta distribuição é utilizada de forma aproximada, ajustando somente os graus de liberdade. Este ajuste aos graus de liberdade é atribuído a Satterthwaite (1946). O intervalo de confiança aproximado é dado por:

$$IC_{1-\alpha}(\mu_1 - \mu_2) : \bar{X}_1 - \bar{X}_2 \pm t_{\alpha/2;\nu}\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}. \tag{2.2.12}$$

Neste caso os graus de liberdade ν para a obtenção do quantil superior da distribuição t de Student é ajustado (Satterthwaite, 1946) por:

$$\nu \cong \frac{\left(\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}\right)^2}{\frac{\left(\frac{S_1^2}{n_1}\right)^2}{n_1 - 1} + \frac{\left(\frac{S_2^2}{n_2}\right)^2}{n_2 - 1}}. \tag{2.2.13}$$

O procedimento mais apropriado para estimar duas médias populacionais por intervalo requer que tenhamos o conhecimento sobre a homogeneidade ou não das variâncias das duas populações. Como se tratam de parâmetros desconhecidos podemos inferir apenas a este respeito. Para isso podemos utilizar o teste F. Um artifício que utilizamos é considerar a variância maior no numerador da expressão, multiplicando o valor encontrado por 2. Assim, para testarmos a hipótese $H_0 : \sigma_1^2 = \sigma_2^2$ calculamos:

$$F_c = \frac{S^2_{Maior}}{S^2_{Menor}} \tag{2.2.14}$$

e o valor-p é determinado por $2 \times P(F > F_c)$. Se valor-p for menor ou igual ao valor nominal α, rejeitamos H_0.

O programa R resultante desse procedimento, considerando normalidade, é apresentado a seguir. Como o R não testa, na função *t.test* se as variâncias são homogêneas ou não, temos que utilizar uma função apropriada para isso. Inclusive o *default* é considerar as variâncias heterogêneas. Essa função para testar a homocedasticidade é *var.test*. Se a hipótese nula não for rejeitada chamamos a função *t.test* com um de seus argumentos indicadores fixado para variâncias homogêneas. Caso contrário, fixamos seu valor para variâncias heterogêneas. A função utilizada é apropriada para testar a hipótese de igualdade das médias populacionais, mas como subproduto podemos determinar os intervalos de confiança, que é o nosso interesse nesse instante. Neste exemplo, utilizamos o conjunto de dados bancada para variável peso, de duas formas diferentes de entrada. No primeiro caso, utilizamos a variável grupo com dois níveis e modelamos *peso∼grupo*. No segundo caso, entramos com duas variáveis distintas por meio da criação do vetor de pesos das duas bancadas (x e y). O R aplica o intervalo sempre para a diferença do grupo 1 em relação ao grupo 2, nessa ordem.

```
> # comandos para obter IC normais da
> # diferença de duas médias independentes.
> # Testa a homogeneidade das  variâncias e aplica
> # o método adequado
> bancada <- read.table("C:/daniel/Cursos/RCursoTeX/bancada.txt",
+                         header=TRUE)
> # aplica o teste de homogeneidade de variâncias
> vari.test <- var.test(bancada$peso~bancada$grupo)
> vari.test  # imprime o resultado

        F test to compare two variances

data:  bancada$peso by bancada$grupo
F = 1.5453, num df = 3, denom df = 3, p-value =
0.7293
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
  0.1000891 23.8580844
sample estimates:
ratio of variances
          1.545295

> alpha <- 0.05  # especifica valor da significância
> # aplica o teste adequado em função do resultado
> if (vari.test$p.value > alpha)
+    diff.test <- t.test(bancada$peso ~ bancada$grupo,
+        alternative = "two.sided", var.equal = TRUE,
+        conf.level = 1-alpha) else
+    diff.test <- t.test(bancada$peso ~ bancada$grupo,
+        alternative = "two.sided", var.equal = FALSE,
+        conf.level = 1-alpha)
> diff.test    # imprime o resultado

        Two Sample t-test

data:  bancada$peso by bancada$grupo
t = -1.0553, df = 6, p-value = 0.3319
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -41.89806  16.64806
sample estimates:
mean in group 1 mean in group 2
         66.625          79.250
```

```
> # forma alternativa
> g1 = c(48.5,86,79,53) # especifica o vetor do grupo 1
> g2 = c(72,88,62,95)   # especifica o vetor do grupo 2
> # aplica o teste de homogeneidade de variâncias
> vari.test2 <- var.test(g1,g2)
> vari.test2  # imprime o resultado

        F test to compare two variances

data:  g1 and g2
F = 1.5453, num df = 3, denom df = 3, p-value =
0.7293
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
  0.1000891 23.8580844
sample estimates:
ratio of variances
          1.545295

> alpha <- 0.05  # especifica valor da significância
> # aplica o teste adequado em função do resultado
> if (vari.test2$p.value > alpha)
+    diff.test2 <- t.test(g1,g2,
+        alternative = "two.sided", var.equal = TRUE,
+        conf.level = 1-alpha) else
+    diff.test2 <- t.test(g1,g2,
+        alternative = "two.sided", var.equal = FALSE,
+        conf.level = 1-alpha)
> diff.test2    # imprime o resultado

        Two Sample t-test

data:  g1 and g2
t = -1.0553, df = 6, p-value = 0.3319
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -41.89806  16.64806
sample estimates:
mean of x mean of y
   66.625    79.250
```

A análise dos resultados nos mostra que as duas médias populacionais não diferem entre si, pois o intervalo abrange 0, e que as variâncias são con-

sideradas homogêneas. Isso é esperado, pois não há razão para supor que a população de alunos que sentam na bancada 1 tenham peso diferente dos que sentam na bancada 2. Este exemplo, embora bastante artificial, possibilitou a exemplificação das funções R apropriadas. A diferença de médias das duas populações $\mu_1 - \mu_2$, deve ser um valor do intervalo $[-41{,}90;\ 16{,}65]$ com 95% de confiança. Convém salientarmos que os vetores $g1$ e $g2$, poderiam ser colunas de um *data frames* em vez de vetores.

2.2.5 Estimação da Diferenças de Duas Médias Em Dados Emparelhados

Em muitas ocasiões experimentais nos deparamos com a necessidade de inferir sobre o efeito de algum medicamento, fertilizante, fungicida entre outros tratamentos. Realizamos experimentos onde temos o maior grau de controle local possível, ou seja, mensuramos os indivíduos ou as unidades experimentais antes da aplicação do tratamento e após a sua aplicação. Neste experimento temos a mesma unidade experimental servindo de controle local. Isto torna este experimento mais eficiente que o experimento em que as amostras são tomadas de forma independente na população tratada e não tratada. Uma alternativa a este delineamento experimental é possível de ser obtida se utilizarmos duas parcelas experimentais locadas e submetidas sob as mesmas condições e sorteamos uma para receber o tratamento e outra para não recebê-lo.

Se X_i e Y_i são as respostas mensuradas antes e após a aplicação do tratamento, respectivamente, na $i-$ésima unidade amostral, para $i = 1, 2, \cdots, n$, então podemos gerar a variável aleatória $d_i = Y_i - X_i$. A estimação pontual do valor esperado desta variável aleatória $E(d_i) = \delta = \mu_Y - \mu_X$ pode ser feita por:

$$\bar{d} = \frac{\sum_{i=1}^{n} d_i}{n}. \tag{2.2.15}$$

O estimador da variância populacional das diferenças é dado por:

$$S_d^2 = \frac{1}{n-1}\left[\sum_{i=1}^{n} d_i^2 - \frac{\left(\sum_{i=1}^{n} d_i\right)^2}{n}\right]. \tag{2.2.16}$$

Assim, o intervalo de confiança pode ser obtido por:

$$IC_{1-\alpha}(\delta) : \bar{d} \pm t_{\alpha/2;\nu=n-1}\frac{s_d}{\sqrt{n}}. \tag{2.2.17}$$

O artifício que usaremos para obter o intervalo de confiança almejado consiste em considerar com um conjunto de dados, para o qual especificamos em cada parcela a variável X e a variável Y (antes e após). Em seguida utilizando o processo de transformação de variáveis descritos na seção 1.2 devemos gerar $D = Y - X$. Finalmente, utilizamos a função *basicStats* do pacote *fBasic* para obtermos o intervalo de confiança para a média da diferença. Podemos também utilizar a função *t.test* e indicar que os dados são emparelhados (*paired*). No programa seguinte descrevemos esse processo com a utilização das duas alternativas. Este exemplo refere-se a produção de leite média diária em kg de todos os animais de cada fazenda em uma amostra de $n = 6$ fazendas da região de Marechal Cândido Rondom antes X e após Y um plano governamental. A questão era responder se o plano foi eficiente e se sim, qual foi o aumento na produção média diária de leite dos animais em kg. Tomamos apenas uma parte dos dados ($n = 6$), da pesquisa originalmente realizada, para ilustrar de forma didática esta situação. O programa R, contendo o *data frame* lido de um arquivo texto denominado *dataleite.txt*, é dado por:

```
> # comandos para obter IC normais da
> # diferença de duas médias emparelhadas.
> leite <- read.table("C:/daniel/Cursos/RCursoTeX/dataleite.txt",
+                          header=TRUE)
> leite  # imprime o data frame

      X     Y
1 12.00 12.56
2 11.58 13.98
```

```
3 11.67 14.23
4 12.32 14.56
5 11.23 13.71
6 11.25 16.78

> #  isso permitirá que o usuário reproduza o exemplo
> alpha <- 0.05  # especifica valor da significância
> # aplica o teste adequado em função do resultado
> diff.test <- t.test(leite$Y,leite$X, paired = TRUE,
+        alternative = "two.sided",  var.equal = TRUE,
+        conf.level = 1-alpha)
> diff.test    # imprime o resultado

        Paired t-test

data:  leite$Y and leite$X
t = 4.0039, df = 5, p-value = 0.01028
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.9409085 4.3157582
sample estimates:
mean of the differences
               2.628333

> # forma alternativa
> D <- leite$Y-leite$X  # calcula as diferenças
> D # imprime as diferenças

[1] 0.56 2.40 2.56 2.24 2.48 5.53

> # Obtém o IC e outras estatísticas descritivas
> basicStats(D, ci=1-alpha) # chamando a função basicStats

                   D
nobs        6.000000
NAs         0.000000
Minimum     0.560000
Maximum     5.530000
1. Quartile 2.280000
3. Quartile 2.540000
Mean        2.628333
Median      2.440000
Sum        15.770000
SE Mean     0.656437
```

```
LCL Mean     0.940908
UCL Mean     4.315758
Variance     2.585457
Stdev        1.607936
Skewness     0.621753
Kurtosis    -0.775527
```

A diferença das médias da produtividade obtida após e antes do plano com 95% de confiança é um valor do intervalo [0,94; 4,32]. Avaliando esse resultado, podemos afirmar que o plano governamental foi eficiente em aumentar a produtividade leiteira média.

2.3 Testes de Hipóteses

Neste seção trataremos dos testes de hipóteses sobre os principais parâmetros de uma ou duas populações. Antes de apresentarmos os métodos e recursos computacionais para realizarmos os testes de hipóteses, devemos atentar para o fato de que existe uma relação estreita entre os procedimentos de estimação e decisão.

Se já temos um intervalo de confiança construído, podemos testar uma hipótese bilateral apenas verificando se este intervalo contém o valor hipotético. Caso o valor hipotético pertença ao intervalo de confiança não temos evidências significativas para rejeitar a hipótese nula. Por outro lado, se o valor hipotético não pertence ao intervalo de confiança, podemos concluir a favor da hipótese alternativa, rejeitando a hipótese nula. Assim, vamos apresentar somente os procedimentos para testarmos médias de uma população e de duas, sejam elas independentes ou emparelhadas. Testes sobre variâncias, desvios padrões ou coeficientes de variação poderão ser realizados com o uso dos intervalos de confiança apresentados anteriormente.

2.3.1 Teste Sobre Médias

Para testarmos hipóteses sobre médias normais devemos utilizar o teste t de Student. Assim, para testarmos a hipótese nula $H_0 : \mu = \mu_0$ utilizamos os seguintes procedimentos. Inicialmente, calculamos a estatística do teste por

$$t_c = \frac{\bar{X} - \mu_0}{\frac{S}{\sqrt{n}}}. \tag{2.3.1}$$

Se a hipótese alternativa for do tipo bilateral $H_1 : \mu \neq \mu_0$, calculamos o valor-p por $P(t > |t_c|)$; se a hipótese alternativa for unilateral do tipo $H_1 : \mu > \mu_0$, calculamos o valor-p por $P(t > t_c)$; e se a hipótese alternativa for unilateral do tipo $H_1 : \mu < \mu_0$, calculamos o valor-p por $P(t < t_c)$. Finalmente, confrontamos o valor-p com o valor nominal do nível de significância α. Se o valor-p for inferior ou igual a α, devemos rejeitar a hipótese nula neste nível de significância; caso contrário, não devemos rejeitar H_0.

Se a distribuição dos dados não for normal podemos utilizar dois testes não-paramétricos: o teste do sinal e o teste dos postos com sinais de Wilcoxon. Vamos descrever o teste do sinal com detalhes e realizar apenas uma breve descrição do teste de Wilcoxon.

Para aplicarmos o teste do sinal, inicialmente calculamos o número de sinais positivos e negativos para a diferença de cada observação amostral com o valor hipotético. Se $X_i - \mu_0$ representa esta diferença, então podemos definir n_+ como o número de observações para as quais $X_i > \mu_0$ (sinais positivos) e n_- com o número de observações para as quais $X_i < \mu_0$ (sinais negativos). Devemos desprezar todas as observações para as quais $X_i = \mu_0$. Assim, o número de observações efetivas amostrais é $n_e = n_+ + n_-$. Ao realizarmos este teste estamos supondo que se a hipótese nula for verdadeira, o número de sinais positivos deve ser igual ao número de sinais negativos. Aplicamos, então, um teste binomial para $p = 1/2$, em que p é a proporção de sinais positivos ou negativos. Assim, a estatística do teste sinal é dada por:

$$M_c = \frac{n_+ - n_-}{2}. \tag{2.3.2}$$

O valor-p é calculado utilizando a distribuição binomial em um teste bilateral por:

$$valor - p = P(M > |M_c|) = \left(\frac{1}{2}\right)^{(n_e-1)} \sum_{j=0}^{\min(n_+,n_-)} \binom{n_e}{j}. \qquad (2.3.3)$$

O valor-p é confrontado com o valor de α e tomamos a decisão de rejeitar ou não a hipótese nula utilizando procedimentos semelhantes ao que apresentamos anteriormente para o teste t. O R calcula como estatística, o número de sinais positivos s e não a estatística M.

A estatística do teste do sinal com postos de Wilcoxon é obtida calculando-se todos os desvios das observações em relação ao valor hipotético e tomando-se os postos dos valores destas diferenças em módulo $d_i = |X_i - \mu_0|$. Se algum valor amostral for igual a zero, devemos eliminá-lo da amostra, como fazemos no teste do sinal. Se houver empates, tomamos a média dos postos que seriam atribuídos a estas observações empatadas. Retornamos os sinais de $X_i - \mu_0$ aos postos das diferenças e somamos os valores positivos. Esta soma é representada por W^+ e é a estatística do teste. Os valores-p podem ser obtidos utilizando-se uma aproximação normal ou a distribuição nula da estatística W^+, derivada pela atribuição de sinais positivos ou negativos a cada posto amostral em todas as combinações possíveis. O teste de Wilcoxon é, em geral, mais poderoso do que o teste do sinal. Nenhum detalhe adicional será apresentado neste material.

Como vimos anteriormente, para o caso de duas médias, podemos utilizar a função *t.test* para testarmos hipóteses sobre a média de uma população. A função *t.test* é apropriada somente para o caso de dados normais. A função *SIGN.test* do pacote *BSDA* é apropriada para aplicarmos o teste do sinal. A função *wilcox.test* é apropriada para aplicar o teste do sinal com postos de Wilcoxon. Devemos optar pelo teste mais apropriado conforme for o caso. Esta escolha deve ser pautada no atendimento ou não das pressuposições básicas de cada teste. Um programa R é apresentado na sequência para testarmos a hipótese da igualdade da média do peso dos coelhos híbridos Norfolk abatidos aos 90 dias a 2,50 kg, ou seja, para testarmos $H_0 : \mu = 2{,}50$.

```
> # comandos teste de hipóteses sobre
> # média de uma população. Ex. teste de H0: mu = 2,50.
> # teste t - assumindo dados normais
> t.test(coelhos$peso,mu=2.50)
```

```
        One Sample t-test

data:  coelhos$peso
t = 2.5816, df = 15, p-value = 0.02085
alternative hypothesis: true mean is not equal to 2.5
95 percent confidence interval:
 2.512969 2.635781
sample estimates:
mean of x
 2.574375

> # teste sinal - assumindo dados não-normais
> library(BSDA)
> SIGN.test(coelhos$peso,md=2.50)

        One-sample Sign-Test

data:  coelhos$peso
s = 14, p-value = 0.0009766
alternative hypothesis: true median is not equal to 2.5
95 percent confidence interval:
 2.545173 2.634482
sample estimates:
median of x
      2.585

                  Conf.Level L.E.pt U.E.pt
Lower Achieved CI     0.9232 2.5500 2.6200
Interpolated CI       0.9500 2.5452 2.6345
Upper Achieved CI     0.9787 2.5400 2.6500

> # teste Wilcoxon - assumindo dados não-normais
> # sem correção de continuidade e aproximação normal (p-value)
> wilcox.test(coelhos$peso,mu=2.50,exact=TRUE,correct=FALSE)

        Wilcoxon signed rank test

data:  coelhos$peso
V = 105, p-value = 0.01055
alternative hypothesis: true location is not equal to 2.5
```

Os três testes, nesse exemplo, levaram a mesma conclusão considerando um valor nominal de significância de 5%, ou seja, nos três casos a hipótese

nula foi rejeitada, indicando que a verdadeira média populacional ao abate deve ser superior ao valor hipotético de 2,50 kg.

2.3.2 Teste Sobre Médias de Duas Populações Emparelhadas

Quando temos dados emparelhados, antes e após a aplicação de um tratamento podemos estar interessados em testes de hipóteses sobre o efeito deste tratamento. Podemos utilizar o mesmo procedimento descrito anteriormente para média e assim testar hipóteses sobre o efeito do tratamento. A hipótese nula de interesse é dada por $H_0 : \delta = \delta_0$. Podemos utilizar o teste t de Student, se as variáveis (X_i, Y_i) tiverem distribuição normal bivariada ou, em caso contrário, os testes não-paramétricos do sinal e do sinal com postos de Wilcoxon.

Seja $d_i = Y_i - X_i$ a diferença entre a observação da i-ésima unidade amostral após Y_i e antes X_i da aplicação do tratamento, sendo $i = 1, 2, \cdots n$. Sejam $\bar{d}$ e S_d^2 a média e a variância amostral destas n observações, então a estatística do teste da hipótese $H_0 : \delta = \delta_0$ supondo normalidade bivariada é dado por:

$$t_c = \frac{\bar{d} - \delta_0}{\frac{S_d}{\sqrt{n}}}, \tag{2.3.4}$$

que segue a distribuição t de Student com $\nu = n - 1$ graus de liberdade sob a hipótese nula.

O teste do sinal é obtido contando-se o número de vezes que $d_i > \delta_0$ e desprezando-se os casos em que $d_i = \delta_0$. As expressões (2.3.2) e (2.3.3) são usadas para testar a hipótese de interesse. O teste do sinal com postos de Wilcoxon também é obtido da mesma forma considerando tanto o posto da diferença $d_i - \delta_0$ considerada em módulo, quanto o sinal da diferença. Como se trata apenas de uma aplicação do mesmo procedimento adaptado para esta situação, não faremos nenhum comentário adicional.

A seguir detalharemos o programa R para aplicar o teste de avaliação da eficiência de um plano governamental no aumento da média dos índices zootécnicos da região de Marechal Cândido Rondom. A produção média diária de leite de $n = 6$ fazendas foi avaliadas antes (X) e após (Y) o

plano governamental. As funções que mencionamos possuem alternativas para o caso de dados emparelhados, adicionando a opção *paired* entre seus argumentos. Neste exemplo, a hipótese nula consiste na afirmativa que o plano não foi eficiente, ou seja, $H_0 : \delta = \delta_0 = 0$. Assim, ao utilizarmos as funções devemos especificar a hipótese com a opção *mu=0* ou *md=0* ou simplesmente não especificar nada, pois o valor 0 é o *default* destas funções. O programa resultante é dado por:

```
> # comandos teste de hipóteses sobre
> # média de duas populações pareadas. Ex. teste de H0: delta = 0.
> # teste t - assumindo dados normais
> t.test(leite$Y,leite$X,mu=0,paired=TRUE)

        Paired t-test

data:  leite$Y and leite$X
t = 4.0039, df = 5, p-value = 0.01028
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.9409085 4.3157582
sample estimates:
mean of the differences
               2.628333

> # teste sinal - assumindo dados não-normais
> # necessário obter D pois só se aplica a uma amostra
> D <- leite$Y - leite$X
> SIGN.test(D,md=0)

        One-sample Sign-Test

data:  D
s = 6, p-value = 0.03125
alternative hypothesis: true median is not equal to 0
95 percent confidence interval:
 0.728 5.233
sample estimates:
median of x
       2.44

                 Conf.Level L.E.pt U.E.pt
Lower Achieved CI     0.7812  2.240  2.560
```

```
Interpolated CI       0.9500  0.728  5.233
Upper Achieved CI     0.9688  0.560  5.530
```

```
> # teste Wilcoxon - assumindo dados não-normais
> # sem correção de continuidade e aproximação normal (p-value)
> wilcox.test(leite$Y,leite$X,mu=0,paired=TRUE,
+             exact=TRUE,correct=FALSE)

        Wilcoxon signed rank test

data:  leite$Y and leite$X
V = 21, p-value = 0.03125
alternative hypothesis: true location shift is not equal to 0
```

Os resultados dos três testes foram praticamente equivalentes se considerarmos o valor nominal de 5% de significância. A hipótese nula deve ser rejeitada nos três casos e o plano governamental foi eficiente para aumentar a média diária da produção leiteira na região.

2.3.3 Teste Sobre Médias de Duas Populações Independentes

Finalmente podemos testar a hipótese da igualdade de duas médias populacionais independentes. Para esse caso, podemos utilizar as funções *t.test* e *wilcox.test* do programa R, conforme vimos anteriormente. No caso de aplicarmos a função *wilcox.test*, estaremos aplicando um teste não-paramétrico equivalente ao de Mann-Whitney. Conforme já apresentamos na seção de estimação por intervalo, devemos inicialmente aplicar o teste de igualdade de variâncias e de acordo com os resultados obtidos, escolhemos entre o teste t de Student exato ou aproximado. O teste exato ocorre quando as variâncias são consideradas homogêneas; o teste é aproximado quando as variâncias são heterogêneas. Devemos neste último caso utilizar o ajuste de graus de liberdade pelo procedimento de Satterthwaite (1946).

Vamos apresentar na sequência as funções comentadas anteriormente, com o objetivo de ilustrar sua utilização. Para isso, um exemplo em dois grupos de alunos foram avaliados com relação ao peso em kg e a altura em m. Os grupos referem-se aos alunos que sentam na bancada da direita (grupo 1) e da esquerda (grupo 2) do laboratório de informática. A primeira turma desta disciplina foi amostrada para esta finalidade. Esperamos a princípio que não haja diferenças significativas entre os dois grupos, uma

vez que a distribuição é completamente aleatória nas duas bancadas da sala de aula.

Devemos fazer um conjunto de dados criando uma variável para identificarmos os grupos. Esta variável tem que ter sempre dois níveis para podermos utilizar a função *t.test*. Sejam $\bar{X}_1$ e $\bar{X}_2$ as médias das amostras aleatórias de tamanhos n_1 e n_2, respectivamente, retiradas das populações 1 e 2. Sejam S_1^2 e S_2^2 as variâncias amostrais relativas às populações 1 e 2. Pressupomos que as amostras sejam aleatórias e independentes e que a distribuição das duas populações seja normal.

Inicialmente devemos testar a hipótese sobre a igualdade das variâncias $H_0 : \sigma_1^2 = \sigma_2^2$. Assim, de acordo com este teste devemos aplicar o teste de igualdade da diferença das médias populacionais a um valor de interesse, ou seja, $H_0 : \mu_1 - \mu_2 = \delta_0$ utilizando os seguintes procedimentos:

a) Se $\sigma_1^2 = \sigma_2^2$:

Neste caso, o teste de igualdade da diferença das médias populacionais a um valor de interesse é exato e a estatística do teste, dada por

$$t_c = \frac{\bar{X}_1 - \bar{X}_2 - \delta_0}{\sqrt{S_p^2\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}} \tag{2.3.5}$$

segue a distribuição t de Student com $\nu = n_1 + n_2 - 2$ graus de liberdade. O significado de S_p^2 foi apresentado na equação (2.2.11).

b) Se $\sigma_1^2 \neq \sigma_2^2$:

Neste caso, a estatística do teste não segue de forma exata a distribuição t de Student. Então, ajustamos os graus de liberdade pelo procedimento de Satterthwaite (1946) ou ajustamos as probabilidades pelo procedimento de Cochran e Cox. A estatística do teste dada por

$$t_c = \frac{\bar{X}_1 - \bar{X}_2 - \delta_0}{\sqrt{\frac{S_1^2}{n_1} + \frac{S_2^2}{n_2}}} \tag{2.3.6}$$

segue aproximadamente a distribuição t de Student com ν graus de liberdade obtidos com o uso da expressão (2.2.13).

Para utilizarmos as funções mencionadas anteriormente, devemos especificar o valor δ_0. Isto é feito utilizando a opção $mu = \delta_0$. O programa R utilizando o exemplo dos grupos de alunos é dado por:

```
> # comandos para aplicar teste para a
> # diferença de duas médias independentes.
> # Testa a homogeneidade das  variâncias e aplica
> # o método adequado-no caso normal
> bancada <- read.table("C:/daniel/Cursos/RCursoTeX/bancada.txt",
+                       header=TRUE)
> # aplica o teste de homogeneidade de variâncias
> vari.test <- var.test(bancada$peso~bancada$grupo)
> vari.test  # imprime o resultado

        F test to compare two variances

data:  bancada$peso by bancada$grupo
F = 1.5453, num df = 3, denom df = 3, p-value =
0.7293
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
  0.1000891 23.8580844
sample estimates:
ratio of variances
          1.545295

> alpha <- 0.05  # especifica valor da significância
> # aplica o teste adequado em função do resultado
> if (vari.test$p.value > alpha)
+    diff.test <- t.test(bancada$peso ~ bancada$grupo,
+        alternative = "two.sided", var.equal = TRUE,
+        mu = 0) else
+    diff.test <- t.test(bancada$peso ~ bancada$grupo,
+        alternative = "two.sided", var.equal = FALSE,
+        mu = 0)
> diff.test    # imprime o resultado

        Two Sample t-test

data:  bancada$peso by bancada$grupo
t = -1.0553, df = 6, p-value = 0.3319
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
```

```
 -41.89806  16.64806
sample estimates:
mean in group 1 mean in group 2
         66.625          79.250

> # teste não paramétrico
> diff.wilcox <- wilcox.test(bancada$peso ~ bancada$grupo,
+        mu=0,paired=FALSE,exact=TRUE,correct=FALSE)
> diff.wilcox

        Wilcoxon rank sum test

data:  bancada$peso by bancada$grupo
W = 4, p-value = 0.3429
alternative hypothesis: true location shift is not equal to 0
```

Devemos observar que a variável grupo deve possuir dois níveis que são usados para identificar as populações. Se quisermos testar um valor diferente para cada variável, devemos fazer vários comandos repetidos, como no programa anterior, especificando um valor hipotético diferente para cada variável. Por *default* as funções utilizam o valor zero se nada for especificado. Obtivemos para ambas variáveis resultados não significativos para os testes da igualdade variâncias e de médias dos dois grupos, como era esperado.

2.3.4 Teste de Normalidade

O R nos permite realizar testes de normalidade para os dados amostrais coletados em n unidades. Anteriormente, já apresentamos alguns procedimentos gráficos para avaliarmos a normalidade. Muitos testes diferentes de normalidade podem ser aplicados no R: Kolmogorov-Smirnov, Shapiro-Wilk, Jarque–Bera e D'Agostino do pacote *fBasics* e Anderson–Darling, Cramer–von Mises, Lilliefors (Kolmogorov-Smirnov), qui-quadrado de Pearson e Shapiro–Francia do pacote *nortest*. Os comandos para aplicarmos estes testes, na ordem respectiva em que foram mencionados, são: *ksnormTest, shapiroTest, jarqueberaTest, dagoTest, adTest, cvmTest, lillieTest, pchiTest* e *sfTest*. Um dos mais poderosos e eficientes teste de normalidade é o teste de Shapiro-Wilk.

O R fornece o valor da estatística de cada teste e o valor-p associado. Se este valor-p for menor do que o valor nominal de significância α previ-

amente adotado, então devemos rejeitar a hipótese nula de normalidade; caso contrário, não haverá evidências significativas neste nível para rejeitar a hipótese de normalidade.

Devemos enfatizar que o teste de normalidade aplicado no contexto de uma amostra aleatória simples onde não há controle local e efeitos de diferentes tratamentos atuando é totalmente justificável, pois estamos diante de um modelo linear simples do tipo:

$$Y_i = \mu + \epsilon_i,$$

em que Y_i é a observação amostral da i-ésima unidade amostral, μ a média geral e ϵ_i o erro associado a i-ésima unidade amostral.

O programa apresentado a seguir ilustra alguns desses testes de normalidade:

```
> # testes de normalidade
> # Pacote fBasics
> shapiroTest(feijao$prod)

Title:
 Shapiro - Wilk Normality Test

Test Results:
  STATISTIC:
    W: 0.908
  P VALUE:
    0.05836

Description:
 Wed May 11 13:23:17 2011 by user: Daniel

> jarqueberaTest(feijao$prod)

Title:
 Jarque - Bera Normalality Test

Test Results:
  STATISTIC:
    X-squared: 2.0796
  P VALUE:
    Asymptotic p Value: 0.3535
```

```
Description:
 Wed May 11 13:23:18 2011 by user: Daniel

> dagoTest(feijao$prod)

Title:
 D'Agostino Normality Test

Test Results:
  STATISTIC:
    Chi2 | Omnibus: 2.5146
    Z3  | Skewness: 1.5524
    Z4  | Kurtosis: -0.3235
  P VALUE:
    Omnibus  Test: 0.2844
    Skewness Test: 0.1206
    Kurtosis Test: 0.7463

Description:
 Wed May 11 13:23:18 2011 by user: Daniel

> # pacote nortest
> library(nortest) # garante que seja carregado
> result <- adTest(feijao$prod)
> result

Title:
 Anderson - Darling Normality Test

Test Results:
  STATISTIC:
    A: 0.7164
  P VALUE:
    0.05172

Description:
 Wed May 11 13:23:18 2011 by user: Daniel

> cvmTest(feijao$prod)

Title:
 Cramer - von Mises Normality Test

Test Results:
```

```
  STATISTIC:
    W: 0.1208
  P VALUE:
    0.05328

Description:
 Wed May 11 13:23:18 2011 by user: Daniel
```

```
> sfTest(feijao$prod)
```

```
Title:
 Shapiro - Francia Normality Test

Test Results:
  STATISTIC:
    W: 0.9165
  P VALUE:
    0.07949

Description:
 Wed May 11 13:23:18 2011 by user: Daniel
```

Nos modelos lineares a suposição de normalidade é feita sobre os resíduos e não sobre a variável dependente. Neste modelo linear simples, ao erro de todas as observações é acrescido uma única constante e esta constante somente faz uma translação dos valores de Y, não alterando a sua distribuição. Assim, testar a normalidade de Y ou de ϵ são procedimentos equivalentes. O que muitos pesquisadores fazem muitas vezes dentro do contexto da experimentação é testar a hipótese de normalidade da variável resposta para verificar se esta pressuposição foi atendida, para validar as inferências realizadas. Isto muitas vezes é incorreto, pois se pressupõe resíduos e não variáveis respostas normais. Então, sob um modelo mais complexo, onde existe controle local, efeito de bloco (β_j) e\ou efeitos de tratamentos (τ_i), a variável resposta Y terá uma distribuição que é na verdade uma mistura de distribuições normais com diferentes médias. Observe que para o modelo linear

$$Y_{ij} = \mu + \beta_j + \tau_i + \epsilon_{ij},$$

a variável Y_{ij} tem a seguinte média: $E(Y_{ij}) = \mu + \beta_j + \tau_i$. Assim, se variarmos a unidade experimental (i, j), teremos diferentes valores médios para Y_{ij}. Como supomos independência e homocedasticidade de variâncias, a mistura de distribuições terá diferentes distribuições normais com diferentes médias, mas com a mesma variância. Então, em uma amostra de tamanho n, não podemos testar a hipótese de normalidade utilizando os valores de Y, mas devemos estimar o erro cuja média é zero e a variância é constante para realizarmos tal teste.

Capítulo 3

Regressão Linear

Os modelos de regressão linear desempenham um grande papel nas mais diferentes áreas do conhecimento. Os pesquisadores buscam sempre modelar seus dados por um modelo e, então, passam a compreender melhor o fenômeno sob estudo. Os modelos lineares são apenas uma das classes utilizadas pelos pesquisadores na compreensão dos problemas de suas pesquisas. A classificação de um modelo como linear é muitas vezes confundida com o tipo de curva matemática que aquele modelo descreve e, ainda, é mal compreendida. Assim, iniciaremos nossa discussão com a classificação de dois modelos como linear ou não-linear. O primeiro modelo é dado por $Y_i = \beta_0 + \beta_1 X_i^2 + \epsilon_i$, em que Y_i e X_i^2 são as variáveis resposta e regressoras, respectivamente; β_0 e β_1 são os seus parâmetros; e ϵ_i é o resíduo ou erro. O segundo modelo é $Y_i = \beta_0 X_i^{\beta_1} + \epsilon_i$. Ambos os modelos descrevem curvas que não são uma reta simples. Esta é uma das causas de confusões na classificação de um modelo como linear. Nestes exemplos, o primeiro modelo é linear e o segundo é não-linear.

Para esclarecermos e definirmos um modelo como linear, vamos apresentar inicialmente um conceito filosófico. Dizemos que um modelo é linear nos parâmetros em função de os parâmetros estarem na forma aditiva e com efeitos simples, sem, entretanto, nos preocuparmos com o tipo de curva que a função representa. Formalmente, podemos dizer que um modelo é linear, se as derivadas parciais de primeira ordem da variável dependente em relação a cada parâmetro não forem funções dos próprios parâmetros. Assim, as derivadas parciais do primeiro modelo são: $\partial Y_i/\partial\beta_0 = 1$ e $\partial Y_i/\partial\beta_1 = X_i^2$. Como nenhuma das derivadas parciais depende dos próprios parâmetros,

então este modelo é linear. No segundo caso, as derivadas parciais são: $\partial Y_i/\partial\beta_0 = X_i^{\beta_1}$ e $\partial Y_i/\partial\beta_1 = \beta_0 X_i^{\beta_1}\ln(X_i)$. O segundo modelo é não-linear nos parâmetros, pois as duas derivadas parciais são funções dos próprios parâmetros. Bastaria uma derivada parcial ser função dos parâmetros para classificarmos o modelo como não-linear.

Dois procedimentos, entre outros, podem ser utilizados para analisarmos os modelos lineares e não-lineares. Utilizaremos o *lm* para os modelos lineares e o *nls* para modelos não-lineares. Neste capítulo, estudaremos apenas os modelos lineares nos parâmetros. O *lm* é, entre os possíveis procedimentos de regressão do R, aquele específico para lidarmos com essa classe de modelos. Este procedimento permite entre outras as seguintes análises:

- Especificação de múltiplos modelos
- Métodos de seleção de modelos
- Diagnósticos de regressão
- Obtenção de valores preditos
- Diagnose de multicolinearidade
- Gráficos de resíduos

3.1 Método dos Quadrados Mínimos

O *lm* foi idealizado para ajustar modelos lineares e fornecer várias ferramentas de diagnóstico da qualidade de ajuste. Seja o modelo linear de regressão com $m+1$ parâmetros definido por:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_m X_{im} + \epsilon_i \qquad (3.1.1)$$

em que Y_i é a i-ésima observação da variável resposta; X_{ik} é i-ésima observação da k-ésima variável; β_k são os parâmetros do modelo; ϵ_k é o resíduo de regressão associado a i-ésima unidade amostral; $k = 0,1,2,\cdots,m$ e $i = 1,2,\cdots,n$; X_{i0} é constante com todos os valores iguais a 1; m representa o número de variáveis e n o tamanho da amostra.

O método dos quadrados mínimos é baseado na ideia de minimizar a soma de quadrados dos resíduos dos modelos lineares. Assim, se $Q = \sum_i^n \epsilon_i^2$ é a soma de quadrados de resíduos, o seu valor mínimo deve ser encontrado para obtermos uma solução de quadrados mínimos. Matricialmente temos o modelo (3.1.1) expresso da seguinte forma:

$$\underset{\sim}{Y} = X\underset{\sim}{\beta} + \underset{\sim}{\epsilon} \tag{3.1.2}$$

em que $\underset{\sim}{Y}$ é o vetor de observações de dimensões $n \times 1$; X é a matriz do modelo de dimensões $n \times (m+1)$ das derivadas parciais de Y_i em relação aos parâmetros; $\underset{\sim}{\beta}$ é o vetor de parâmetros $[(m+1) \times 1]$; e $\underset{\sim}{\epsilon}$ é o vetor de resíduos $(n \times 1)$.

Os resíduos podem ser isolados por $\underset{\sim}{\epsilon} = \underset{\sim}{Y} - X\underset{\sim}{\beta}$ e a soma de quadrados do resíduos matricialmente é expressa por:

$$Q = \underset{\sim}{\epsilon}'\underset{\sim}{\epsilon} = \left(\underset{\sim}{Y} - X\underset{\sim}{\beta}\right)' \left(\underset{\sim}{Y} - X\underset{\sim}{\beta}\right)$$

$$Q = \underset{\sim}{\epsilon}'\underset{\sim}{\epsilon} = \left(\underset{\sim}{Y}'\underset{\sim}{Y} - 2\underset{\sim}{\beta}'X'\underset{\sim}{Y} + \underset{\sim}{\beta}'X'X\underset{\sim}{\beta}\right)$$

Obtemos as derivadas de Q com relação a β e encontramos:

$$\frac{\partial Q}{\partial \underset{\sim}{\beta}} = -2X'\underset{\sim}{Y} + 2X'X\underset{\sim}{\beta}$$

Igualamos a zero e obtemos as conhecidas equações normais (EN) na sequência. Assim, temos:

$$-2X'\underset{\sim}{Y} + 2X'X\underset{\sim}{\hat{\beta}} = 0$$

$$X'X\underset{\sim}{\hat{\beta}} = X'\underset{\sim}{Y} \tag{3.1.3}$$

em que $\underset{\sim}{\hat{\beta}}$ é o estimador de mínimos quadrados do parâmetro β.

A matriz de derivadas parciais ou de modelo X, em geral, possui posto coluna completo nos modelos de regressão. Assim, a matriz $X'X$ possui inversa única e a solução do sistema é:

$$\underset{\sim}{\hat{\beta}} = (X'X)^{-1}X'\underset{\sim}{Y}. \tag{3.1.4}$$

O valor esperado de $\underset{\sim}{Y}$ é $E(\underset{\sim}{Y}) = X\underset{\sim}{\beta}$. Podemos obter os valores estimados substituindo $\underset{\sim}{\beta}$ por $\underset{\sim}{\hat{\beta}}$. Assim, os valores preditos são dados por:

$$\underset{\sim}{\hat{Y}} = X\underset{\sim}{\hat{\beta}}. \tag{3.1.5}$$

É importante obtermos as somas de quadrados do modelo e do resíduo, para aplicar uma análise de variância e realizarmos inferência a respeito do modelo ajustado. Nenhuma pressuposição foi feita até o momento sobre a distribuição dos resíduos, mas se temos a intenção de realizar inferências é necessário pressupormos normalidade e ainda distribuição idêntica e independente de todos os componentes do vetor de resíduos. Podemos estimar Q substituindo $\underset{\sim}{\beta}$ por $\underset{\sim}{\hat{\beta}}$. Obtemos após algumas simplificações:

$$\hat{Q} = \underset{\sim}{Y}'\underset{\sim}{Y} - \underset{\sim}{\hat{\beta}}'X'\underset{\sim}{Y}$$

Assim, podemos interpretar esta expressão da seguinte forma:

$$\text{SQRes} = \text{SQTotal não corrigida} - \text{SQModelo}$$

Assim, a soma de quadrados de modelo é dada por:

$$\text{SQModelo} = \underset{\sim}{\hat{\beta}}'X'\underset{\sim}{Y} \tag{3.1.6}$$

O número de graus de liberdade associado ao modelo é igual ao posto coluna da matriz X. Se esta matriz tem posto coluna completo $m+1$, concluímos que a soma de quadrados do modelo está associada a $m+1$ graus de liberdade e a soma de quadrados do resíduo a $n-m-1$ graus de liberdade. O que fazemos é definir sub-modelos a partir do modelo completo com $m+1$

parâmetros. Desta forma podemos definir dois tipos básicos de soma de quadrados: a sequencial (tipo I) e a parcial (tipo II). Na sequencial tomamos o modelo completo e o reduzimos eliminando a variável m. Obtemos a soma de quadrado do modelo completo, que representamos por $R(\beta_0,\beta_1,\cdots,\beta_m)$, e a do modelo reduzido, representada por $R(\beta_0,\beta_1,\cdots,\beta_{m-1})$. A notação R indica uma redução particular do modelo que estamos abordando. Se tomarmos a diferença da soma de quadrados dos dois modelos teremos $R(\beta_m/\beta_0,\beta_1,\cdots,\beta_{m-1}) = R(\beta_0,\cdots,\beta_m)$-$R(\beta_0,\cdots,\beta_{m-1})$. Se do modelo com $m-1$ variáveis eliminarmos a última e repetirmos este procedimento, teremos a soma de quadrado da $(m-1)$-ésima variável ajustada para todas as outras que a precedem. Se fizermos isso repetidas vezes até reduzirmos o modelo ao termo constante apenas, teremos as somas de quadrados de cada variável ajustada para todas as outras que a precedem, ignorando as variáveis que a sucedem. Esta é a soma de quadrados tipo I ou sequencial.

Para obtermos as somas de quadrados parciais ou do tipo II, devemos a partir do modelo completo formar um novo modelo eliminando uma das variáveis. A soma de quadrados do modelo reduzido é comparada com a soma de quadrado do modelo completo e a sua diferença é a soma de quadrados do tipo II. Assim, teremos o ajuste de cada variável para todas as outras do modelo. Podemos perceber que as somas de quadrados tipo I e tipo II da m-ésima variável são iguais. Via de regra as somas de quadrados tipo I e tipo II não serão iguais para as demais variáveis, a menos de ortogonalidade. Podemos resumir o dois tipos de somas de quadrados conforme esquema apresentado na Tabela 3.1.

Tabela 3.1. Tipos de somas de quadrados de um modelo de regressão contendo m variáveis.

FV	SQ Tipo I	SQ Tipo II
X_1	$R(\beta_1/\beta_0)$	$R(\beta_1/\beta_0,\beta_2,\cdots,\beta_m)$
X_2	$R(\beta_2/\beta_0,\beta_1)$	$R(\beta_2/\beta_0,\beta_1,\cdots,\beta_m)$
$\vdots$	$\vdots$	$\vdots$
X_m	$R(\beta_m/\beta_0,\beta_1,\cdots,\beta_{m-1})$	$R(\beta_m/\beta_0,\beta_1,\cdots,\beta_{m-1})$

Uma forma alternativa bastante útil para podermos obter as somas de quadrados tipo II é baseada no método da inversa de parte da inversa de Searle (1971, 1987). Por este método podemos obter as somas de quadrados tipo II de uma forma mais direta do que por redução de modelos. Vamos

apresentar o método no contexto de regressão linear na sequência. Seja a matriz $(X'X)^{-1}$ definida por:

$$(X'X)^{-1} = \begin{bmatrix} x_{00} & x_{01} & \cdots & x_{0m} \\ x_{10} & x_{11} & \cdots & x_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m0} & x_{m1} & \cdots & x_{mm} \end{bmatrix} \tag{3.1.7}$$

Assim, para obtermos a soma de quadrados do tipo II para a variável X_k podemos simplesmente calcular:

$$R(\beta_k/\beta_0,\cdots,\beta_{k-1},\beta_{k+1},\cdots,\beta_m) = \frac{\hat{\beta}_k^2}{x_{kk}} \tag{3.1.8}$$

3.2 Um Exemplo de Regressão

Vamos mostrar um exemplo de um ajuste de um modelo de regressão utilizando programação R, sem utilizar uma função especializada. O objetivo é mostrar todos os cálculos, utilizando as fórmulas anteriormente apresentadas, por meio de um programa matricial. Seja para isso um exemplo em que a variável X representa o número de horas de exposição solar de uma planta e a variável resposta Y o crescimento da planta. Os dados deste exemplo estão apresentados na Tabela 3.2.

Tabela 3.2. Crescimento de uma planta Y após ser submetida a um tempo X de exposição solar em horas.

X	Y
0,1	0,88
0,2	0,90
0,3	0,99
0,5	1,12
0,8	1,40
1,0	1,62
1,5	2,20
2,0	3,10

Vamos ajustar um modelo linear quadrático do tipo:

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \epsilon_i \tag{3.2.1}$$

em que β_0, β_1 e β_2 são os parâmetros que desejamos estimar.

Para este modelo vamos estimar os parâmetros e obter as somas de quadrados dos tipos I e II utilizando funções matriciais no R. A matriz X do modelo é dada por:

$$X = \begin{bmatrix} 1 & 0{,}1 & 0{,}01 \\ 1 & 0{,}2 & 0{,}04 \\ 1 & 0{,}3 & 0{,}09 \\ 1 & 0{,}5 & 0{,}25 \\ 1 & 0{,}8 & 0{,}64 \\ 1 & 1{,}0 & 1{,}00 \\ 1 & 1{,}5 & 2{,}25 \\ 1 & 2{,}0 & 4{,}00 \end{bmatrix}$$

O vetor de parâmetros é dado por:

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

O vetor de observações é dado por:

$$\underset{\sim}{Y} = \begin{bmatrix} 0{,}88 \\ 0{,}90 \\ 0{,}99 \\ 1{,}12 \\ 1{,}40 \\ 1{,}62 \\ 2{,}20 \\ 3{,}10 \end{bmatrix}$$

Desta forma podemos formular o programa R para ajustar este modelo e obter as somas de quadrados e testes de hipóteses relativo aos parâmetros. Vamos apenas ilustrar uma parte de todos os cálculos, pois felizmente podemos utilizar a função *lm* do R que nos fornece todas as estimativas e testes de hipóteses que desejarmos, com comando mais simples. O nosso objetivo é possibilitar ao leitor obter um maior conhecimento de todo o processo de regressão linear. O programa resultante desta análise é:

```
> # programa para ajustarmos um modelo de regressão linear
> # quadrática no R e obtermos as somas de quadrados do tipo II
> # testes de hipóteses H_0: Beta_i = 0, etc.
> x <- c(0.1,0.2,0.3,0.5,0.8,1.0,1.5,2.0)
> y <- c(0.88,0.90,0.99,1.12,1.40,1.62,2.20,3.10)
> n <- length(y)
> x1 <- matrix(1,n,1)
> X <- cbind(x1,x,x^2)
> IXLX <- solve(t(X) %*% X) # inversa de X'X
> XLY <- t(X) %*% y         # X'Y
> beta <- IXLX %*% XLY      # parâmetros
> IXLX

                    x
   0.7096108 -1.566702  0.6461362
x -1.5667022  4.832225 -2.2213309
   0.6461362 -2.221331  1.0926845

> XLY

     [,1]
  12.2100
x 13.3650
  20.2799

> beta

       [,1]
  0.8289504
x 0.4048794
  0.3607692

> # obtenção das somas de quadrados
> sqb0b1b2 <- t(beta) %*% XLY
> glm1 <- ncol(X)
```

```
> sqtotal <- sum(y*y)
> sqresm1 <- sqtotal - sqb0b1b2
> sqtotal

[1] 22.8533

> sqresm1

            [,1]
[1,] 0.004239938

> glr1 <- n - glm1
> glr1

[1] 5

> # somas de quadrados do tipo II
> sqb1 <- beta[2]^2/IXLX[2,2]
> sqb2 <- beta[3]^2/IXLX[3,3]
> sqb1

[1] 0.03392378

> sqb2

[1] 0.1191143

> # teste t: H0: bi= 0
> tcb0   <- (beta[1]-0)/(sqresm1/glr1*IXLX[1,1])^0.5
> prtcb0 <- 2*(1-pt(abs(tcb0),glr1))
>  beta[1]

[1] 0.8289504

> tcb0

         [,1]
[1,] 33.79276

> prtcb0

             [,1]
[1,] 4.266968e-07

> tcb1   <- (beta[2]-0)/(sqresm1/glr1*IXLX[2,2])^0.5
> prtcb1 <- 2*(1-pt(abs(tcb1),glr1))
> beta[2]
```

```
[1] 0.4048794

> tcb1

         [,1]
[1,] 6.324954

> prtcb1

            [,1]
[1,] 0.001456167

> tcb2   <- (beta[3]-0)/(sqresm1/glr1*IXLX[3,3])^0.5
> prtcb2 <- 2*(1-pt(abs(tcb2),glr1))
> beta[3]

[1] 0.3607692

> tcb2

         [,1]
[1,] 11.85188

> prtcb2

             [,1]
[1,] 7.530072e-05

> sqtotc <- sqtotal - sum(y)^2/n
> sqtotc

[1] 4.217787

> R2 <- 1-sqresm1/sqtotc
> R2

          [,1]
[1,] 0.9989947
```

Os principais resultados obtidos neste procedimento são apresentados na sequência. Iniciamos pelas matrizes $X'X$ e $X'\underset{\sim}{Y}$, dadas por:

$$X'X = \begin{bmatrix} 8 & 6{,}4 & 8{,}28 \\ 6{,}4 & 8{,}28 & 13{,}048 \\ 8{,}28 & 13{,}048 & 22{,}5444 \end{bmatrix}$$

e

$$X'\underset{\sim}{Y} = \begin{bmatrix} 12{,}21 \\ 13{,}365 \\ 20{,}2799 \end{bmatrix}$$

A matriz inversa $(X'X)^{-1}$ é dada por:

$$(X'X)^{-1} = \begin{bmatrix} 0{,}7096 & -1{,}5667 & 0{,}6461 \\ -1{,}5667 & 4{,}8322 & -2{,}2213 \\ 0{,}6461 & -2{,}2213 & 1{,}0927 \end{bmatrix}$$

Finalmente, o vetor $\underset{\sim}{\beta}$ é estimado por:

$$\underset{\sim}{\hat{\beta}} = \begin{bmatrix} 0{,}8289504 \\ 0{,}4048794 \\ 0{,}3607692 \end{bmatrix}$$

Portanto, o modelo de regressão ajustado é $\hat{Y}_i = 0{,}8289504 + 0{,}4048794\ X_i + 0{,}3607692 X_i^2$. O gráfico desta função quadrática está apresentado na Figura (3.1) e o programa R para gerá-lo é dado por:

```
> fx <- function(x) 0.8289504 + 0.4048794*x+0.3607692*x^2
> xx <- seq(min(x),max(x),by=0.01)
> plot(xx,fx(xx),type="l")
```

As somas de quadrados para modelo $(\beta_0, \beta_1, \beta_2)$, total não corrigido e resíduo foram iguais a 22,84906, 22,8533 e 0,0042399, respectivamente. O R^2, proporção da variação total corrigida explicada pelo modelo de regressão, é dado por: $R^2 = 1 -$ sqresíduo/sqtotal corrigida $= 99{,}90\%$. Um excelente ajuste foi encontrado, mas é necessário que se faça a análise de resíduo para termos uma confirmação disso, o que não será feito neste instante. A soma de quadrado total corrigida foi obtida por SQtotal c = sqtotal nc $- G^2/n =$ 4,2178, em que $G = \sum_{i=1}^{n} Y_i = 12{,}21$.

No passo seguinte obtivemos as somas de quadrados do tipo II para X e X^2 por $0{,}4048794^2/4{,}8322 = 0{,}03392$ e $0{,}3607692^2/1{,}0927 = 0{,}1191$,

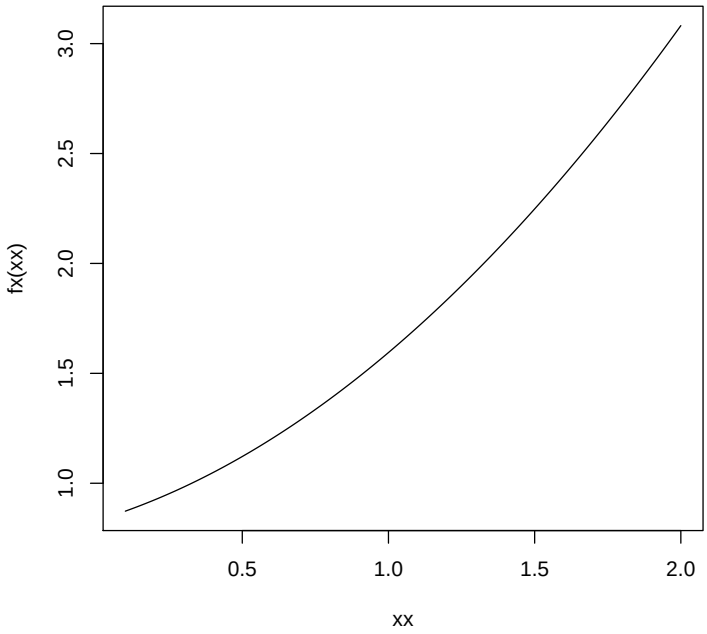


Figura 3.1. Equação quadrática resultante do ajuste de quadrados mínimos do exemplo.

respectivamente. Podemos efetuar um teste F para a hipótese $H_0 : \beta_i = 0$ se desejarmos, dividindo o quadrado médio do tipo II de cada variável pelo quadrado médio do erro e calcularmos o valor-p utilizando a distribuição F de Snedecor. O quadrado médio do tipo II para cada parâmetro é igual a soma de quadrados, pois está associado a 1 grau de liberdade. Finalmente podemos utilizar o teste t de Student para obtermos um teste de hipótese equivalente ao realizado pelo teste F, baseado em somas de quadrados parciais ou somas de quadrados do tipo II. Este teste está descrito formalmente nas equações (3.3.4) a (3.3.7). Os resultados destes testes de hipótese bilateral estão apresentados na Tabela 3.3.

Podemos fazer muitas outras análises no R executando passo a passo a análise pretendida. Isso, entretanto, não será necessário, pois o R possui alguns procedimentos apropriados para lidarmos com ajustes de modelos lineares. Entre esses procedimentos destacamos a função *lm*, para a qual, anteriormente, já apontamos suas principais características, ou seja,

Tabela 3.3. Testes de hipótese do tipo $H_0 : \beta_i = 0$, com $i = 0, 1, 2$ utilizando a distribuição t de Student com $\nu = 5$ graus de liberdade.

Parâmetro	Estimativa	t_c	$Pr(t > \|t_c\|)$
β_0	0,82895	33,793	$4{,}267 \times 10^{-7}$
β_1	0,40488	6,325	0,0014562
β_2	0,36077	11,852	0,0000753

as análises com que é capaz de lidar. Como o ambiente de programação R representa um é muito poderoso, mas requer conhecimentos especiais de estatística e de álgebra matricial, não o abordaremos mais, nesse capítulo. Faremos todas as análises de modelos lineares de regressão utilizando a função *lm*, *linear models*.

3.3 A função *lm*

Vamos apresentar a função *lm* para realizarmos o ajuste do modelo anterior e em seguida apresentaremos um exemplo de regressão múltipla, onde aparentemente ocorre um resultado paradoxal na inferência realizada. Utilizamos este exemplo para elucidar aspectos de testes de hipóteses que são muitas vezes ignorados. Inicialmente vamos apresentar os comandos necessários para ajustarmos o modelo (3.2.1). A função *lm* permite a criação de variáveis no próprio modelo por intermédio da função $I()$. O argumento dessa função deve ser uma função de algum objeto ou variável pré-existente. Nesse exemplo quadrático, podemos utilizar o termo quadrático por meio do comando *I(X**2)*. Assim,o programa simplificado para o ajuste do modelo quadrático é dado por:

```
> # programa para ajustarmos um modelo de regressão linear
> # quadrática no R e obtermos as somas de quadrados do tipo II
> # testes de hipóteses H_0: Beta_i = 0, etc. usando lm
> x <- c(0.1,0.2,0.3,0.5,0.8,1.0,1.5,2.0)
> y <- c(0.88,0.90,0.99,1.12,1.40,1.62,2.20,3.10)
> rq <- lm(y ~ x+I(x^2))
> rq

Call:
lm(formula = y ~ x + I(x^2))

Coefficients:
(Intercept)            x       I(x^2)
     0.8290       0.4049       0.3608
```

```
> summary(rq)

Call:
lm(formula = y ~ x + I(x^2))

Residuals:
        1         2         3         4         5
 0.006954 -0.024357  0.007117 -0.001582  0.016254
        6         7         8
 0.025401 -0.048000  0.018214

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.82895    0.02453  33.793 4.27e-07 ***
x            0.40488    0.06401   6.325  0.00146 **
I(x^2)       0.36077    0.03044  11.852 7.53e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02912 on 5 degrees of freedom
Multiple R-squared: 0.999,        Adjusted R-squared: 0.9986
F-statistic:  2484 on 2 and 5 DF,  p-value: 3.204e-08

> anova(rq)

Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value    Pr(>F)
x          1 4.0944  4.0944 4828.41 1.169e-08 ***
I(x^2)     1 0.1191  0.1191  140.47 7.530e-05 ***
Residuals  5 0.0042  0.0008
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> # Pacote car
> library(car) # garantindo que será carregado
> Anova(rq,type="II")

Anova Table (Type II tests)

Response: y
            Sum Sq Df F value   Pr(>F)
x         0.033924  1  40.005 0.001456 **
I(x^2)    0.119114  1 140.467 7.53e-05 ***
Residuals 0.004240  5
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

A linha de comando *rq <- lm(y ~ x+I(x**2))*, nos permite fazer o ajuste do modelo (3.2.1). A função *Anova(rq,type="II")* do pacote *car* possibilita

o cálculo das somas de quadrados do tipo II. O comando *anova(rq)*, faz o mesmo para somas de quadrados do tipo I. A função *summary* do R apresenta as estimativas dos parâmetros do modelo com seus erros padrões e testes de hipóteses associados, a análise de variância, o R^2, média geral, os resíduos e o teste F para o modelo. Esse teste F da análise de variância está relacionado a seguinte hipótese:

$$\begin{cases} H_0 : \beta_1 = \beta_2 = \beta_3 = \cdots = \beta_m = 0 \\ H_1 : \beta_i \neq 0 \quad \text{Para algum } i = 1, 2, \cdots, m \end{cases} \tag{3.3.1}$$

Neste exemplo observamos que o F observado foi igual a 2484,4 e o valor-p associado é $Pr(F > F_c) < 3{,}24 \times 10^{-8}$. Assim a hipótese nula global de que nenhuma variável explica significativamente a variação na variável resposta Y_i foi rejeitada. O R realiza o teste t para as hipóteses do tipo $H_0 : \beta_i = 0$, $i = 1, 2, \cdots, m$. Neste exemplo os valores da estatística t e as respectivas significâncias estão apresentadas na Tabela 3.3. Concluímos que ambas as variáveis têm efeito significativamente diferente de zero na variação de Y. O teste t de Student é equivalente ao teste F parcial. Embora este teste tenha sido aplicado por ser padrão no R, é conveniente utilizar para este exemplo um teste sequencial. Isto porque esta análise refere-se ao ajuste de um modelo polinomial e usualmente nestes casos utilizamos testes que envolvem somas de quadrados tipo I. Este tipo de procedimento é comumente encontrado nos livros de estatística experimental e está apresentado na saída do *anova(rq)*.

Vamos apresentar um segundo exemplo, como dissemos anteriormente, para elucidarmos alguns pontos interessantes da análise de regressão linear. Nosso exemplo, refere-se a uma amostra de $n = 10$ árvores, na qual foram mensurados o volume (Y), em $m^3.acre^{-1}$, sendo que 1 *acre* é igual a 4.064 m^2, a área basal (X_1) em dm^2, a área basal tomada em % em relação à área de outra espécie (X_2) e a altura em pés (X_3) (1 pé = 30,48 cm). Na Tabela 3.4 temos os dados amostrados na população de *Araucaria angustifolia*.

Esses dados foram arquivados em um arquivo texto de nome *arvores.txt*. Vamos inicialmente ajustar um modelo linear simples para cada variável utilizando o modelo linear dado por:

Tabela 3.4. Dados de uma amostra de $n = 10$ árvores de araucária (*Araucaria angustifolia*) mensuradas em relação ao volume Y, área basal X_1, área basal relativa X_2 e altura em pés X_3.

Y	X_1	X_2	X_3
65	41	79	35
78	71	48	53
82	90	80	64
86	80	81	59
87	93	61	66
90	90	70	64
93	87	96	62
96	95	84	67
104	100	78	70
113	101	96	71

$$Y_i = \beta_0 + \beta_1 X_{hi} + \epsilon_i, \quad \text{Para} \qquad h = 1{,}2 \text{ ou } 3, \quad i = 1, 2, \cdots, n \qquad (3.3.2)$$

O programa para realizarmos estes ajustes, para cada uma das variáveis regressoras, é dado por:

```
> # lê o arquivo e atribui ao objeto árvores
> arvores <- read.table("C:/daniel/Cursos/RCursoTeX/arvores.txt",
+                header=TRUE)
> m1 <- lm(arvores$Y ~ arvores$X1)
> summary(m1)

Call:
lm(formula = arvores$Y ~ arvores$X1)

Residuals:
     Min       1Q   Median       3Q      Max
-10.7994  -2.6942  -0.1651   3.7156  13.0095

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  33.9634    11.4989   2.954  0.01832 *
arvores$X1    0.6537     0.1330   4.916  0.00117 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 7.106 on 8 degrees of freedom
Multiple R-squared: 0.7513,        Adjusted R-squared: 0.7202
F-statistic: 24.17 on 1 and 8 DF,  p-value: 0.00117
```

```
> anova(m1)
```

```
Analysis of Variance Table

Response: arvores$Y
           Df  Sum Sq Mean Sq F value  Pr(>F)
arvores$X1  1 1220.39  1220.4  24.165 0.00117 **
Residuals   8  404.01    50.5
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> m2 <- lm(arvores$Y ~ arvores$X2)
> summary(m2)
```

```
Call:
lm(formula = arvores$Y ~ arvores$X2)

Residuals:
    Min      1Q  Median      3Q     Max
-25.149  -4.934   2.581   4.543  15.357

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  55.3278    22.2240   2.490   0.0375 *
arvores$X2    0.4408     0.2829   1.558   0.1579
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.48 on 8 degrees of freedom
Multiple R-squared: 0.2328,        Adjusted R-squared: 0.1369
F-statistic: 2.427 on 1 and 8 DF,  p-value: 0.1579
```

```
> anova(m2)
```

```
Analysis of Variance Table

Response: arvores$Y
           Df Sum Sq Mean Sq F value Pr(>F)
```

```
arvores$X2  1  378.1  378.10   2.427 0.1579
Residuals   8 1246.3  155.79
```

```
> m3 <- lm(arvores$Y ~ arvores$X3)
> summary(m3)

Call:
lm(formula = arvores$Y ~ arvores$X3)

Residuals:
     Min       1Q   Median       3Q      Max
-10.6050  -2.5658  -0.4999   3.9853  12.6587

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  21.8732    13.7590   1.590  0.15056
arvores$X3    1.1052     0.2222   4.973  0.00109 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.044 on 8 degrees of freedom
Multiple R-squared: 0.7556,        Adjusted R-squared: 0.7251
F-statistic: 24.73 on 1 and 8 DF,  p-value: 0.001088

> anova(m3)

Analysis of Variance Table

Response: arvores$Y
           Df  Sum Sq Mean Sq F value   Pr(>F)
arvores$X3  1 1227.42 1227.42  24.735 0.001088 **
Residuals   8  396.98   49.62
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Na Tabela 3.5 apresentamos os resultados mais importantes destes ajustes, que iremos mencionar futuramente. Selecionamos o F calculado e sua significância e o R^2 do modelo.

Observamos que o modelo 2 não se ajustou aos dados, embora isso fosse esperado, uma vez que a variável X_2 é resultante de uma medida relativa entre uma variável mensurada diretamente na espécie e outra medida em outra espécie. Portanto, o resultado é perfeitamente justificável, pois a

Tabela 3.5. Resultados mais importantes do ajuste dos modelos lineares simples para os dados dos volumes das $n = 10$ árvores de araucária *Araucaria angustifolia*.

Modelo	F_c	$Pr(F > F_c)$	R^2
1: $E(Y_i) = \beta_0 + \beta_1 X_{1i}$	24,17	0,0012	0,7513
2: $E(Y_i) = \beta_0 + \beta_1 X_{2i}$	2,43	0,1579	0,2328
3: $E(Y_i) = \beta_0 + \beta_1 X_{3i}$	24,73	0,0011	0,7556

covariação existente entre X_2 e Y pode ser atribuída meramente à fatores de acaso. As demais variáveis apresentam explicações significativas ($P < 0{,}05$) da variação que ocorre na variável resposta, com R^2 igual a 75,13% para X_1 e 75,56% para X_3. Agora vamos ajustar o modelo linear múltiplo dado por:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \epsilon_i \tag{3.3.3}$$

O programa R, que faz uso do *lm* para ajustar o modelo (3.3.3), é dado por:

```
> rlm <- lm(arvores$Y ~ arvores$X1+arvores$X2+arvores$X3)
> summary(rlm)

Call:
lm(formula = arvores$Y ~ arvores$X1 + arvores$X2 + arvores$X3)

Residuals:
     Min       1Q   Median       3Q      Max
-10.3705  -2.5313  -0.1433   3.7844   7.4460

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -33.8227    75.3585  -0.449    0.669
arvores$X1   -2.2267     4.0281  -0.553    0.600
arvores$X2    0.2698     0.1533   1.759    0.129
arvores$X3    4.7659     6.7865   0.702    0.509

Residual standard error: 6.543 on 6 degrees of freedom
Multiple R-squared: 0.8419,        Adjusted R-squared: 0.7628
F-statistic: 10.65 on 3 and 6 DF,  p-value: 0.008117
```

Os principais resultados obtidos do ajuste do modelo (3.3.3) são apresentados e discutidos na sequência. A princípio, vamos apresentar (Tabela 3.6) o resumo da análise de variância.

Tabela 3.6. Resumo da análise de variância do ajuste de regressão múltipla aos dados do volume das árvores de araucária.

FV	GL	QM	F_c	$Pr(F > F_c)$
Regressão	3	455,85296	10,65	0,0081
Erro	6	42,80685		
Total Corrigido	9			

Podemos concluir que pelo menos uma variável explica significativamente a variação que ocorre na variável resposta Y, ou seja, a hipótese nula (3.3.1) deve ser rejeitada se for considerado o nível nominal de 5%. Na Tabela 3.7 apresentamos os testes t de Student para a hipótese nula $H_0 : \beta_h = 0$, em que $h = 1, 2, 3$. Devemos neste instante apresentar a expressão geral para realizarmos os testes de hipóteses sobre componentes do vetor de parâmetros. A variância do estimador do vetor de parâmetros é dada por:

$$V\left(\underset{\sim}{\hat{\beta}}\right) = (X'X)^{-1}\sigma^2 \tag{3.3.4}$$

O estimador desta variância é obtido substituindo a variância paramétrica pelo estimador da variância ($S^2 = QME$). Assim, temos o estimador da variância do estimador dos parâmetros dada por:

$$\hat{V}\left(\underset{\sim}{\hat{\beta}}\right) = (X'X)^{-1}S^2 \tag{3.3.5}$$

Desta forma, o erro padrão de $\hat{\beta}_i$ é dado por:

$$S_{(\hat{\beta}_i)} = \sqrt{x_{ii}S^2} \tag{3.3.6}$$

em que x_{ii} é o elemento correspondente a i-ésima diagonal da matriz inversa $(X'X)^{-1}$.

Logo, o teste t de Student para a hipótese $H_0 : \beta_i = \delta_0$, em que δ_0 é uma constante real de interesse pode ser aplicado, pois sob H_0 a distribuição da estatística do teste dada por

$$t_c = \frac{\hat{\beta}_i - \delta_0}{S_{(\hat{\beta}_i)}} \tag{3.3.7}$$

é t de Student com $\nu = n - m - 1$ graus de liberdade.

O R testa a hipótese nula, assumindo que a constante δ_0 é igual a zero. Os resultados para este caso estão apresentados na Tabela 3.7.

Tabela 3.7. Estimativas dos parâmetros e teste t de Student para a nulidade das estimativas.

Parâmetros	Estimativas	$S_{(\hat{\beta}_i)}$	t_c	$Pr(t > \lvert t_c \rvert)$
β_0	-33,82268	75,35853	-0,45	0,6693
β_1	-2,22672	4,02805	-0,55	0,6004
β_2	0,26976	0,15332	1,76	0,1290
β_3	4,76590	6,78649	0,70	0,5088

Quando observamos os resultados dos testes de hipóteses na Tabela 3.7, verificamos que nenhuma variável explicou significativamente a variação da variável resposta Y. Este resultado é aparentemente contraditório ao resultado do teste da hipótese global do modelo de regressão, hipótese esta que foi significativamente rejeitada. Este suposto paradoxo na verdade é um problema de interpretação do que está sendo realmente testado pelos testes t individuais. O que ocorre é que o teste t é equivalente ao teste F, obtido a partir das somas de quadrados parciais ou do tipo II. Assim, o que o t realmente testa é a contribuição de uma variável, eliminando a explicação das demais variáveis no modelo. Então, se a explicação da variável para a variação de Y for expressiva, após ser eliminada a redundância da informação com as outras variáveis do modelo, a estatística do teste tenderá a pertencer a região crítica. Essa redundância é dependente da estrutura de correlação existente entre a variável que está sendo testada e as demais variáveis do modelo.

O que acontece neste exemplo é que temos uma forte estrutura de correlação entre as três variáveis do modelo e, portanto, na presença das outras, a variável que está sendo testada não contribui com uma explicação signifi-

cativa da variação total. Podemos observar as correlações entre as variáveis, independente e regressoras, utilizando o comando:

```
> # obtém as correlações do objeto árvores
> cor(arvores)

           Y        X1        X2        X3
Y  1.0000000 0.8667675 0.4824552 0.8692601
X1 0.8667675 1.0000000 0.2450168 0.9995353
X2 0.4824552 0.2450168 1.0000000 0.2429134
X3 0.8692601 0.9995353 0.2429134 1.0000000
```

Podemos perceber que duas das variáveis que apresentaram resultados não significativos para o teste t, são individualmente importantes para a variação do volume, pois apresentaram significâncias menores que 5% nos testes individuais. Portanto, não tem nada de paradoxal nos resultados encontrados. O que temos são variáveis correlacionadas que não necessitariam estar todas, ao mesmo tempo, no modelo e parte delas nem precisaria ser mensurada, onerando menos os experimentos de campo.

Um outro parâmetro que é estimado pelo *summary* do objeto *lm* é o R^2, o qual mede a proporção da variação do total dos dados que é explicada pelo modelo de regressão. Um outro importante parâmetro é o coeficiente de determinação ajustado ($R^2_{Aj.}$). Este ajuste, feito para o número de parâmetros no modelo, fornece uma medida mais adequada para comparar modelos com diferentes quantidades de parâmetros. O R^2 ajustado é dado por:

$$R^2_{Aj.} = 1 - \frac{n-i}{n-p}\left(1 - R^2\right) \tag{3.3.8}$$

em que n é o tamanho da amostra, p é o número de parâmetros (incluindo o intercepto) e i é igual a 1, se o modelo inclui o intercepto ou 0, se o modelo não inclui β_0.

A função *Anova* do pacote *car* que é interessante para calcularmos as somas de quadrados do tipos II pode ser utilizada com a opção *type="II"*. Para as somas de quadrados do tipo I, simplesmente utilizamos a função *anova*. Essas funções devem ser aplicadas ao objeto *lm*. O programa simplificado ilustrando a obtenção das somas de quadrados do tipo I e II é dado por:

```
> anova(rlm)

Analysis of Variance Table

Response: arvores$Y
           Df  Sum Sq Mean Sq F value   Pr(>F)
arvores$X1  1 1220.39 1220.39 28.5092 0.001762 **
arvores$X2  1  126.06  126.06  2.9448 0.136970
arvores$X3  1   21.11   21.11  0.4932 0.508829
Residuals   6  256.84   42.81
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> Anova(rlm,type="II")

Anova Table (Type II tests)

Response: arvores$Y
            Sum Sq Df F value Pr(>F)
arvores$X1  13.081  1  0.3056 0.6004
arvores$X2 132.513  1  3.0956 0.1290
arvores$X3  21.111  1  0.4932 0.5088
Residuals  256.841  6
```

Além das estimativas dos parâmetros podemos observar as somas de quadrados tipo I e II resultantes das funções utilizadas. Outras características que são importantes na função *lm* referem-se a possibilidade de obtermos os valores preditos de Y_i, seus intervalos de confiança para o valor médio da resposta ou seus intervalos de confiança para uma predição estocástica ou predição futura. Para apresentarmos estes conceitos, sejam Y_i a observação da variável resposta na i-ésima unidade amostral e o vetor $\underset{\sim}{z_i} = [1 \quad X_{1i} \quad X_{2i} \quad \cdots \quad X_{mi}]'$ o vetor de variáveis regressoras, incluindo a indicadora do intercepto, então o valor predito $\hat{Y}_i$ é dado por:

$$\hat{Y}_i = \underset{\sim}{z}' \underset{\sim}{\hat{\beta}} = \hat{\beta}_0 + \hat{\beta}_1 X_{1i} + \cdots + \hat{\beta}_1 X_{mi} \tag{3.3.9}$$

Este vetor $\underset{\sim}{z}_i$ não necessita necessariamente ser observado entre o conjunto de observações. O estimador do erro padrão desta predição para o intervalo da média é dado por:

$$S(\hat{Y}_i) = \sqrt{\underset{\sim}{z}'(X'X)^{-1}\underset{\sim}{z}S^2}. \tag{3.3.10}$$

O intervalo de confiança do valor médio é dado por:

$$\hat{Y}_i \pm t_{\alpha/2,\nu} S(\hat{Y}_i). \tag{3.3.11}$$

Se diferenciarmos a predição futura da predição média simplesmente utilizando a notação $\tilde{Y}_i$, mas mantivermos a mesma combinação linear determinada pelo vetor $\underset{\sim}{z}$, teremos o intervalo de confiança do valor futuro dado por:

$$\tilde{Y}_i \pm t_{\alpha/2,\nu} S(\tilde{Y}_i). \tag{3.3.12}$$

Esse intervalo distingue-se do anterior somente pelo estimador do erro padrão do valor da predição futura, o qual envolve uma variância residual a mais em relação ao erro padrão da predição do valor médio. Esse estimador do erro padrão da predição futura é dado por:

$$S(\tilde{Y}_i) = \sqrt{\left[1 + \underset{\sim}{z}'(X'X)^{-1}\underset{\sim}{z}\right] S^2}. \tag{3.3.13}$$

O programa R simplificado para ilustrarmos o uso destas opções está apresentado na sequência. Podemos especificar o valor de $1-\alpha$ com a opção *level=0.95*. Claro que se o valor de interesse for 95%, como nesse caso, essa opção não precisa ser utilizada, por se tratar do padrão da função.

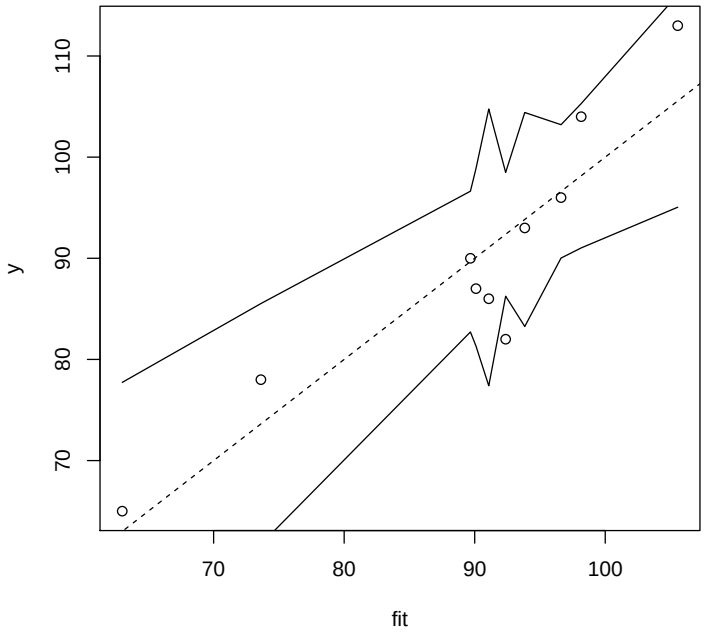
```
> f <- predict(rlm,interval = "confidence",    level = 0.95) # obs. média
> f

         fit      lwr       upr
1   62.99915 48.26922  77.72908
2   73.62104 61.71841  85.52367
3   92.37049 86.25085  98.49013
4   91.07800 77.39582 104.76018
5   90.09668 81.32836  98.86500
6   89.67289 82.71425  96.63154
7   93.83503 83.25735 104.41271
```

```
8   96.61361 90.02134 103.20588
9   98.15913 91.02563 105.29262
10 105.55398 95.04829 116.05967
```

```
> ff <- predict(rlm,interval = "prediction",    level = 0.95) # obs. futura
> ff
```

```
         fit      lwr       upr
1   62.99915 41.24434  84.75395
2   73.62104 53.67177  93.57031
3   92.37049 75.23133 109.50965
4   91.07800 70.01849 112.13751
5   90.09668 71.84334 108.35001
6   89.67289 72.21656 107.12922
7   93.83503 74.64680 113.02326
8   96.61361 79.30007 113.92715
9   98.15913 80.63236 115.68589
10 105.55398 86.40534 124.70262
```

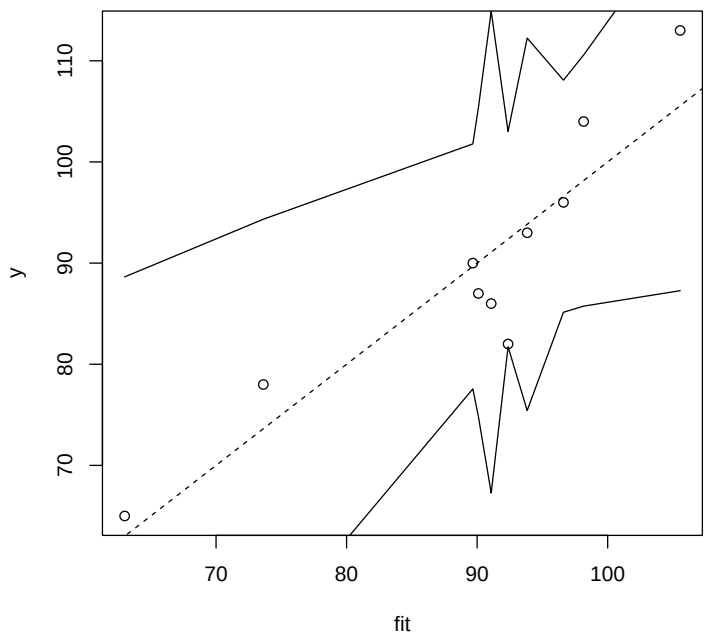
```
> # intervalo de confiança - gráfico para média
> fit <- f[,1] # valores preditos
> lower <- f[,2] # limite inferior
> upper <- f[,3] # limite superior
> y <- arvores$Y
> plot(fit, y)
> abline(0, 1, lty = 2)
> ord <- order(fit)
> lines(fit[ord], lower[ord])
> lines(fit[ord], upper[ord])
> # intervalo de confiança - gráfico para obs, futura
> fit <- ff[,1] # valores preditos
> lower <- ff[,2] # limite inferior
> upper <- ff[,3] # limite superior
> plot(fit, y)
> abline(0, 1, lty = 2)
> ord <- order(fit)
> lines(fit[ord], lower[ord],lty=3) # linhas pontilhadas
> lines(fit[ord], upper[ord],lty=3) # linhas pontilhadas
```



Alguns autores questionam o procedimento de ligar os limites dos intervalos de confiança ao longo de todas as observações para obtermos um intervalo simultâneo de todos os valores médios preditos. O correto seria substituir os quantis da distribuição t, por quantis da F. Utilizando esse procedimento, obtivemos o intervalo de confiança exato ao longo das observações médias preditas e obtivemos o gráfico correspondente. Os resultados obtidos são:

```
> cilm.adj <- function(object, alpha = 0.05, plot.it = T)
+ {
+     f <- predict(object, se.fit = T)
+     p <- length(coef(object))
+     fit <- f$fit
+     adjust <- (p * qf(1 - alpha, p,
+                length(fit) - p))^0.5 * f$se.fit
+     lower <- fit - adjust
+     upper <- fit + adjust
+     if(plot.it)
+     {
+         y <- fit + resid(object)
+         plot(fit, y)
```

```
+         abline(0, 1, lty = 2)
+         ord <- order(fit)
+         lines(fit[ord], lower[ord])
+         lines(fit[ord], upper[ord])
+         invisible(list(lower=lower, upper=upper))
+     }
+     else list(lower = lower, upper = upper)
+ }
> cilm.adj(rlm,0.05,T)
> cilm.adj(rlm,0.05,T)$lower

       1        2        3        4        5        6
37.36389 52.90628 81.72017 67.26620 74.83672 77.56241
       7        8        9       10
75.42615 85.14075 85.74434 87.27039

> cilm.adj(rlm,0.05,T)$upper

       1        2        3        4        5        6
 88.6344  94.3358 103.0208 114.8898 105.3566 101.7834
       7        8        9       10
112.2439 108.0865 110.5739 123.8376
```



Podemos utilizar ainda algumas outras opções do modelo de regressão. Particularmente interessante são os coeficientes de determinações semi-parciais dos tipos I e II. Não encontramos funções prontas para obtermos essas correlações semi-parciais quadráticas no R. Os coeficientes de determinação semi-parciais são estimados por:

$$R^2_{sp1} = \frac{R(\beta_h/\beta_0, \cdots, \beta_{h-1})}{\text{SQtotal corrigida}} \tag{3.3.14}$$

e

$$R^2_{sp2} = \frac{R(\beta_h/\beta_0, \cdots, \beta_{h-1}, \beta_{h+1}, \cdots, \beta_m)}{\text{SQtotal corrigida}} \tag{3.3.15}$$

em que R^2_{sp1} e R^2_{sp2} são os coeficientes de determinação semi-parciais dos tipos I e II, respectivamente, para a h-ésima variável.

Também são úteis os coeficientes de determinação parciais dos tipos I e II. Também não encontramos opções para estimarmos essas correlações parciais no R. Os estimadores correspondentes são dados por:

$$R^2_{p1} = \frac{R(\beta_h/\beta_0, \cdots, \beta_{h-1})}{R(\beta_h/\beta_0, \cdots, \beta_{h-1}) + SQE^*} \tag{3.3.16}$$

em que SQE^* é a soma de quadrados do erro resultante do ajuste de um modelo contendo as variáveis X_1, X_2, $\cdots$, X_h e

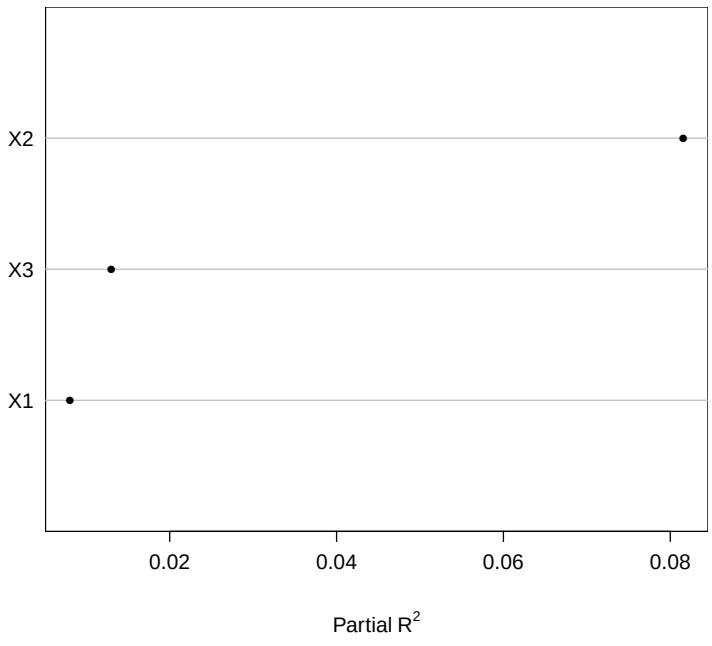
$$R^2_{p2} = \frac{R(\beta_h/\beta_0, \cdots, \beta_{h-1}, \beta_{h+1}, \cdots, \beta_m)}{R(\beta_h/\beta_0, \cdots, \beta_{h-1}, \beta_{h+1}, \cdots, \beta_m) + \text{SQE}} \tag{3.3.17}$$

em que SQE é a soma de quadrados do erro resultante do ajuste do modelo completo.

```
> library(Design) # requires Hmisc
> attach(arvores)

The following object(s) are masked from 'arvores (position 4)':

    X1, X2, X3, Y
```

```
> f <- ols(Y ~ X1 + X2 + X3) #ordinary least square
> f

Linear Regression Model

ols(formula = Y ~ X1 + X2 + X3)

         n Model L.R.       d.f.          R2      Sigma
        10      18.44          3      0.8419      6.543

Residuals:
     Min       1Q   Median       3Q      Max
-10.3705  -2.5313  -0.1433   3.7844   7.4460

Coefficients:
             Value Std. Error       t Pr(>|t|)
Intercept -33.8227    75.3585 -0.4488   0.6693
X1         -2.2267     4.0281 -0.5528   0.6004
X2          0.2698     0.1533  1.7594   0.1290
X3          4.7659     6.7865  0.7023   0.5088

Residual standard error: 6.543 on 6 degrees of freedom
Adjusted R-Squared: 0.7628

> fp <- anova(f)  # anova do objeto
> fp

                Analysis of Variance          Response: Y

 Factor     d.f. Partial SS MS        F     P
 X1         1      13.08146  13.08146  0.31 0.6004
 X2         1     132.51347 132.51347  3.10 0.1290
 X3         1      21.11118  21.11118  0.49 0.5088
 REGRESSION 3    1367.55888 455.85296 10.65 0.0081
 ERROR      6     256.84112  42.80685

> spII <- plot(fp, what='partial R2')
> spII # correlações quadráticas semi-parciais do tipo II

         X2          X3          X1
0.081576872 0.012996292 0.008053104

> fI <- anova(lm(Y ~ X1 + X2 + X3))
> total <- sum(fI$"Sum Sq")
```

```
> spI  <- fI$"Sum Sq"[1:(length(fI$"Sum Sq")-1)]/total
> spI # correlações quadráticas semi-parciais do tipo I

[1] 0.75128589 0.07760337 0.01299629

> c <- fI$"Sum Sq"
> for (i in 2:length(fI$"Sum Sq")) c[i] <- c[i-1] + c[i]
> resids <- total - c
> seq <- 1:(length(fI$"Sum Sq")-1)
> pI <- fI$"Sum Sq"[seq]/(resids[seq] + fI$"Sum Sq"[seq])
> pI # correlações quadráticas parciais do tipo I

[1] 0.75128589 0.31201836 0.07595252

> ult <- length(fI$"Sum Sq")
> pII <- fp[,2][seq]/(fp[,2][ult+1] + fp[,2][seq])
> pII # correlações quadráticas parciais do tipo II

        X1         X2         X3
0.04846376 0.34034136 0.07595252

> detach(arvores)
```



3.4 Seleção de Modelos

A seleção de modelos é bastante interessante na pesquisa científica, pois muitas vezes temos variáveis correlacionadas que não contribuem para a variação da variável resposta de forma significativa, na presença das outras. Dizemos que existe uma redundância da informação. Assim, procedimentos para selecionarmos modelos de regressão linear são importantes no sentido de evitarmos a inclusão em um modelo de variáveis que são correlacionadas com outras variáveis candidatas. Evitamos com isso mensurações desnecessárias e onerosas. A literatura especializada nos fornece diferentes métodos de seleção de modelos, que são: *forward*, *backward*, *stepwise*, *maxr*, *minr*, *rsquare*, *adjrsq* e *cp*. Cada um destes métodos tem uma característica especial. Enfocaremos nesta seção apenas os três primeiros: *forward*, *backward* e *stepwise*. convém salientar que o R possui a função *step* do pacote *base*, que nos permite aplicar estes três métodos. A diferença básica dos três métodos em relação aos mesmos métodos encontrados em outros programas de análise estatística é que o R usa o AIC (*Akaike Information Criterion*) em vez de utilizar o F parcial. A função *stepAIC* do pacote *MASS* realiza o mesmo procedimento, porém com maiores detalhes na análise.

Vamos apresentar algumas características do uso desse método implementado na função *stepAIC*. A função possui uma opção em que podemos especificar a direção da aplicação do algoritmo, que é dada por *direction = c("both", "backward", "forward")*. Assim, com a escolha de uma delas, determinamos como será o comportamento do método. Inicialmente, m variáveis regressoras candidatas são submetidas ao procedimento. Devemos escolher um modelo inicial para aplicarmos os diferentes métodos e outros dois modelos, um mínimo e outro máximo. O critério AIC é computado para esse modelo inicial. Para explicarmos o restante do método, vamos considerar que a opção de direção tenha sido a *forward* e, nesse caso, vamos escolher um modelo da variável resposta em função apenas do intercepto. Assim, a esse modelo mínimo é acrescentado uma variável de cada vez e o modelo resultante é ajustado. O critério AIC é calculado para cada um deles. Os que apresentarem valores menores do critério são colocados acima do modelo inicial e os de maiores valores de AIC, abaixo. Entre aqueles modelos em que as variáveis regressoras apresentaram AIC menor que o modelo inicial, devemos escolher aquela variável que apresentou menor va-

lor do critério *AIC*. A variável escolhida é introduzida no modelo, que passa a ser denominado de modelo atual. A esse modelo são introduzidas cada uma das outras variáveis remanescentes, formando $m-1$ modelos de duas variáveis. Estes modelos são formados pela variável escolhida no passo primeiro passo, com cada uma das variáveis candidatas a entrar. Novamente entre aquelas variáveis que apresentaram *AIC* menor que o modelo atual, escolhemos aquela que gerou o modelo de *AIC* mínimo. Se nenhuma variável apresentou modelo com *AIC* menor que o modelo atual, encerramos o processo e ficamos com um modelo com a variável que entrou no primeiro passo. Já no primeiro passo, poderíamos ter interrompido o processo também, se nenhum dos modelos com uma das m variáveis tivesse apresentado menor *AIC* em relação ao modelo inicial, só com o intercepto. Se uma das candidatas foi escolhida no segundo passo, formamos um modelo com esta variável e aquela escolhida no passo 1. As variáveis candidatas são avaliadas uma por vez na presença destas duas variáveis e todo o processo é repetido. Devemos parar quando nenhuma das candidatas conseguir compor um modelo com menor *AIC* do que o modelo atual, do passo anterior. Também paramos se não houver mais variáveis candidatas a entrar no modelo.

A opção *both* é muito parecida com a *forward*, exceto pelo fato de que em cada passo, após a entrada de uma das variáveis candidatas, devemos testar as variáveis que estavam no modelo. Se uma ou mais delas apresentarem modelos, ao serem removidas, com *AIC* menor do que o *AIC* do modelo atual, aquela que apresentar menor valor de *AIC* ao ser removida para teste, deve sair do modelo. No próximo passo testamos todas as candidatas a entrarem, quanto para saírem do modelo atual e escolhemos o modelo com *AIC* mínimo que seja ainda menor que o *AIC* do modelo atual. As variáveis remanescentes, candidatas a entrar no modelo, são colocadas um por vez no modelo final e o processo continua com entradas e saídas até não termos mais candidatas para entrarem ou as candidatas a sair não produzirem modelos com valores de *AIC* menores do que o do modelo atual. O modelo inicial para essa opção pode ser o modelo mínimo, o máximo ou qualquer modelo intermediário.

A opção *backward* deve ser iniciada com o modelo máximo, para o qual o critério *AIC* deve ser calculado. As variáveis no modelo são excluídas uma por vez e os para os modelos resultantes devemos determinar o *AIC*. Se todos os modelos resultantes apresentarem *AICs* maiores que o do modelo

atual, o processo é encerrado e o modelo final é o modelo atual. Se por outro lado, alguns ou todos os modelos resultantes tiverem *AICs* menores do que o do atual, escolhemos aquele de valor mínimo. Se for eliminada uma variável, o procedimento é repetido para as $m-1$ variáveis remanescentes no modelo. Paramos o processo se todas as variáveis de um passo resultarem em modelos com *AICs* maiores que o do modelo atual ou se modelo resultar em um modelo somente com o intercepto.

O programa R para realizarmos a escolha de modelos de regressão, para os dados das árvores, utilizando como variáveis candidatas apenas as variáveis originais e os vários métodos descritos é dado por:

```
> library(MASS) # carregando pacote MASS
> attach(arvores) # liberando nomes variáveis

The following object(s) are masked from 'arvores (position 4)':

    X1, X2, X3, Y

> modelo1 <- lm(Y~1) # modelo mínimo-só intercepto
> # aplica o forward considerando modelo inicial y~1
> # limitado ao modelo mínimo que é esse e ao modelo máximo
> # y~X1+X2+X3, trace = 1, solta os passos do procedimento
> result1 <- stepAIC(modelo1,direction="forward",
+               scope = list(upper = ~X1+X2+X3, lower = ~1),trace=1)

Start:  AIC=52.9
Y ~ 1

       Df Sum of Sq     RSS    AIC
+ X3    1    1227.4  396.98 40.813
+ X1    1    1220.4  404.01 40.989
+ X2    1     378.1 1246.30 52.253
<none>              1624.40 52.903

Step:  AIC=40.81
Y ~ X3

       Df Sum of Sq    RSS    AIC
+ X2    1   127.059 269.92 38.956
<none>              396.98 40.813
+ X1    1     7.627 389.35 42.619
```

```
Step:  AIC=38.96
Y ~ X3 + X2

       Df Sum of Sq    RSS    AIC
<none>              269.92 38.956
+ X1    1    13.082 256.84 40.459

> result1$anova

Stepwise Model Path
Analysis of Deviance Table

Initial Model:
Y ~ 1

Final Model:
Y ~ X3 + X2


  Step Df  Deviance Resid. Df Resid. Dev      AIC
1                           9  1624.4000 52.90309
2 + X3  1 1227.4180         8   396.9820 40.81306
3 + X2  1  127.0594         7   269.9226 38.95550

> summary(result1)

Call:
lm(formula = Y ~ X3 + X2)

Residuals:
    Min      1Q  Median      3Q     Max
-11.058  -2.241  -0.792   3.830   8.614

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.9524    14.6517   0.475  0.64958
X3            1.0161     0.2019   5.032  0.00151 **
X2            0.2634     0.1451   1.815  0.11235
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.21 on 7 degrees of freedom
```

```
Multiple R-squared: 0.8338,        Adjusted R-squared: 0.7864
F-statistic: 17.56 on 2 and 7 DF,  p-value: 0.00187
```

```
> # aplica stepwise considerando modelo inicial y~1
> # limitado ao modelo mínimo que é esse e ao modelo máximo
> # y~X1+X2+X3, trace = 1, solta os passos do procedimento
> result2 <- stepAIC(modelo1,direction="both",
+               scope = list(upper = ~X1+X2+X3, lower = ~1),trace=1)

Start:  AIC=52.9
Y ~ 1

       Df Sum of Sq     RSS    AIC
+ X3    1    1227.4  396.98 40.813
+ X1    1    1220.4  404.01 40.989
+ X2    1     378.1 1246.30 52.253
<none>              1624.40 52.903

Step:  AIC=40.81
Y ~ X3

       Df Sum of Sq     RSS    AIC
+ X2    1    127.06  269.92 38.956
<none>               396.98 40.813
+ X1    1      7.63  389.35 42.619
- X3    1   1227.42 1624.40 52.903

Step:  AIC=38.96
Y ~ X3 + X2

       Df Sum of Sq     RSS    AIC
<none>               269.92 38.956
+ X1    1     13.08  256.84 40.459
- X2    1    127.06  396.98 40.813
- X3    1    976.38 1246.30 52.253

> result2$anova

Stepwise Model Path
Analysis of Deviance Table

Initial Model:
Y ~ 1
```

```
Final Model:
Y ~ X3 + X2


  Step Df  Deviance Resid. Df Resid. Dev      AIC
1                           9  1624.4000 52.90309
2 + X3  1 1227.4180         8   396.9820 40.81306
3 + X2  1  127.0594         7   269.9226 38.95550
```

```
> summary(result2)
```

```
Call:
lm(formula = Y ~ X3 + X2)

Residuals:
    Min      1Q  Median      3Q     Max
-11.058  -2.241  -0.792   3.830   8.614

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.9524    14.6517   0.475  0.64958
X3            1.0161     0.2019   5.032  0.00151 **
X2            0.2634     0.1451   1.815  0.11235
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.21 on 7 degrees of freedom
Multiple R-squared: 0.8338,        Adjusted R-squared: 0.7864
F-statistic: 17.56 on 2 and 7 DF,  p-value: 0.00187
```

```
> modelo2 <- lm(Y~X1+X2+X3) # modelo máximo
> # aplica o backward considerando modelo inicial máximo
> # limitado ao modelo mínimo y~1 e a esse modelo máximo
> # trace = 1, solta os passos do procedimento
> result3 <- stepAIC(modelo2,direction="backward",
+               scope = list(upper = ~X1+X2+X3, lower = ~1),trace=1)

Start:  AIC=40.46
Y ~ X1 + X2 + X3

       Df Sum of Sq    RSS    AIC
- X1    1    13.081 269.92 38.956
```

```
- X3    1    21.111 277.95 39.249
<none>              256.84 40.459
- X2    1   132.513 389.35 42.619

Step:  AIC=38.96
Y ~ X2 + X3

       Df Sum of Sq     RSS    AIC
<none>               269.92 38.956
- X2    1    127.06  396.98 40.813
- X3    1    976.38 1246.30 52.253
```

> result3$anova

```
Stepwise Model Path
Analysis of Deviance Table

Initial Model:
Y ~ X1 + X2 + X3

Final Model:
Y ~ X2 + X3


  Step Df Deviance Resid. Df Resid. Dev      AIC
1                          6   256.8411 40.45873
2 - X1  1 13.08146         7   269.9226 38.95550
```

> summary(result3)

```
Call:
lm(formula = Y ~ X2 + X3)

Residuals:
    Min      1Q  Median      3Q     Max
-11.058  -2.241  -0.792   3.830   8.614

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.9524    14.6517   0.475  0.64958
X2            0.2634     0.1451   1.815  0.11235
X3            1.0161     0.2019   5.032  0.00151 **
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.21 on 7 degrees of freedom
Multiple R-squared: 0.8338,        Adjusted R-squared: 0.7864
F-statistic: 17.56 on 2 and 7 DF,  p-value: 0.00187
```

```
> # aplica stepwise considerando modelo inicial máximo
> # limitado ao modelo mínimo  e ao modelo máximo
> # trace = 1, solta os passos do procedimento
> result4 <- stepAIC(modelo2,direction="both",
+               scope = list(upper = ~X1+X2+X3, lower = ~1),trace=1)

Start:  AIC=40.46
Y ~ X1 + X2 + X3

       Df Sum of Sq    RSS    AIC
- X1    1    13.081 269.92 38.956
- X3    1    21.111 277.95 39.249
<none>               256.84 40.459
- X2    1   132.513 389.35 42.619

Step:  AIC=38.96
Y ~ X2 + X3

       Df Sum of Sq     RSS    AIC
<none>                269.92 38.956
+ X1    1     13.08  256.84 40.459
- X2    1    127.06  396.98 40.813
- X3    1    976.38 1246.30 52.253

> result4$anova

Stepwise Model Path
Analysis of Deviance Table

Initial Model:
Y ~ X1 + X2 + X3

Final Model:
Y ~ X2 + X3


  Step Df Deviance Resid. Df Resid. Dev      AIC
```

```
1                          6   256.8411 40.45873
2 - X1  1 13.08146         7   269.9226 38.95550
```

```
> summary(result4)
```

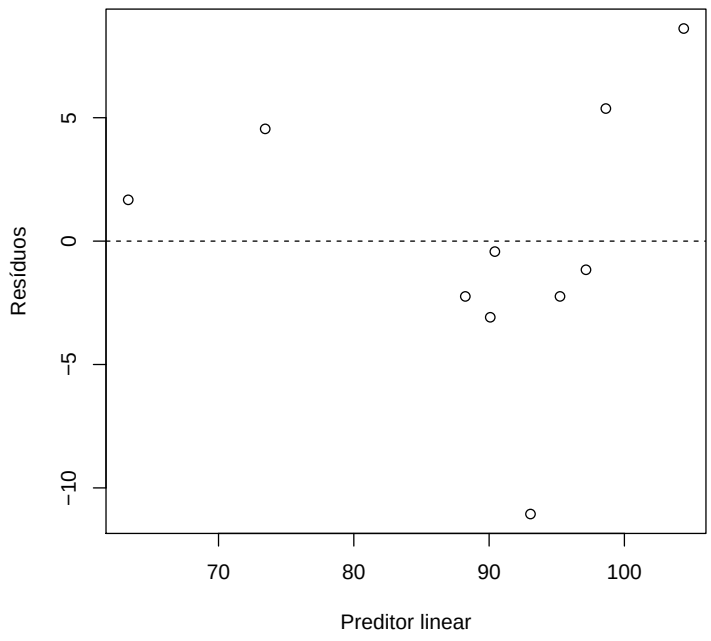
```
Call:
lm(formula = Y ~ X2 + X3)

Residuals:
    Min      1Q  Median      3Q     Max
-11.058  -2.241  -0.792   3.830   8.614

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.9524    14.6517   0.475  0.64958
X2            0.2634     0.1451   1.815  0.11235
X3            1.0161     0.2019   5.032  0.00151 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.21 on 7 degrees of freedom
Multiple R-squared: 0.8338,        Adjusted R-squared: 0.7864
F-statistic: 17.56 on 2 and 7 DF,  p-value: 0.00187
```

```
> detach(arvores) # concatena variáveis a arvores novamente
```

Nos três métodos obtivemos o mesmo modelo ajustado, da variável resposta Y em função das variáveis X_2 e X_3. Algumas vezes os procedimentos podem resultar em conclusões conflitantes quanto ao modelo e o pesquisador deve escolher o que melhor lhe convier. Esta escolha, entre outras coisas, pode ser embasada na análise de resíduos e na qualidade da predição da variável aleatória Y. Convém salientar, que se compararmos esse procedimento com os obtidos, por exemplo, pelo programa SAS, veremos que os modelos finais são bem diferentes. O SAS utiliza a soma de quadrados parciais, teste F, e baseia sua decisão em níveis de significância de permanência e de entrada, que podem ser manipulados pelos usuários. A filosofia apresentada no R, é considerada por muitos especialistas como sendo melhor. No SAS o modelo final é dado por Y em função de X_3, considerando níveis de permanência e de entrada de variáveis igual a 5%. A variável X_2, foi adicionada ao modelo no R, pois o modelo resultante apresentou menor *deviance* residual, ou seja, menor *AIC*.

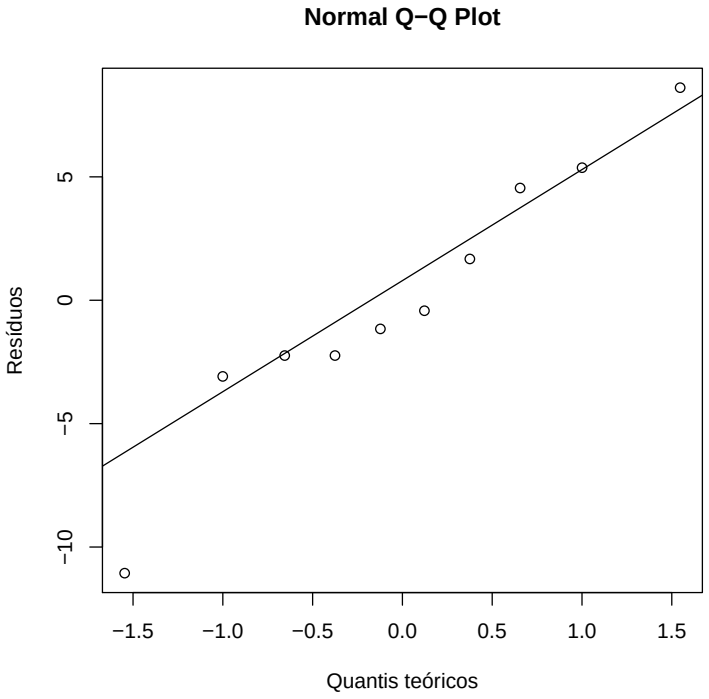
Apresentamos na sequência os gráficos de resíduos, para avaliarmos se há algum distúrbio no modelo de regressão selecionado. O gráfico dos resíduos é:

```
> # gráfico resíduo
> rs <- resid(result4)
> plot(predict(result4), rs, xlab = "Preditor linear",
+ ylab = "Resíduos")
> abline(h = 0, lty = 2)
```



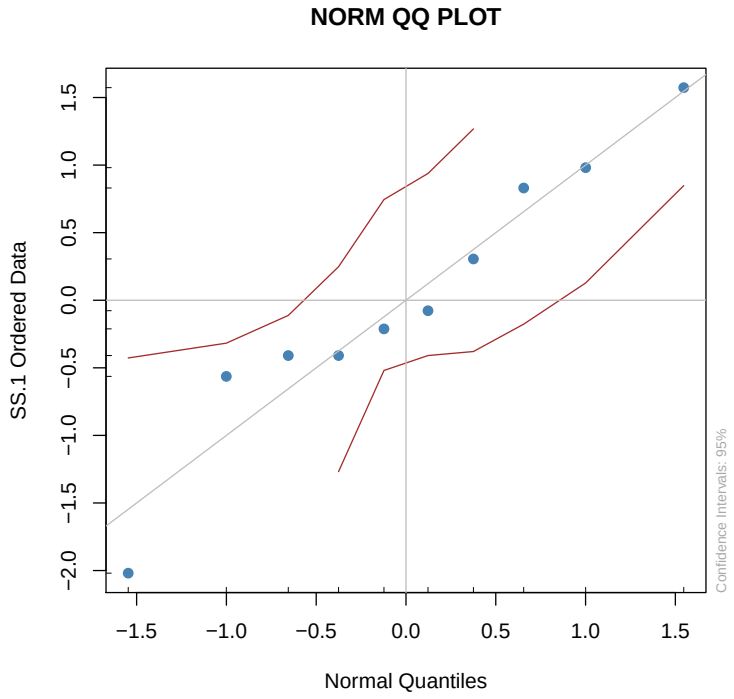
O *qq-plot* é dado por:

```
> # q-qplot
> qqnorm(rs, xlab = "Quantis teóricos", ylab = "Resíduos")
> qqline(rs)
```



Podemos utilizar ainda o pacote *fBasics*, para obtermos o *qq-plot*, juntamente com o intervalos de confiança.

```
> # q-qplot - fBasics
> library(fBasics) # garantindo que o pacote será carregado
> qqnormPlot(rs) #usando o fBasics
```



3.5 Diagnóstico em Regressão Linear

Seja o modelo de regressão linear dado por

$$\underset{\sim}{Y} = X\underset{\sim}{\beta} + \underset{\sim}{\epsilon}$$

em que $\underset{\sim}{Y}$ é o vetor de observações de dimensões $n \times 1$; X é a matriz do modelo de dimensões $n \times (m+1)$ das derivadas parciais de Y_i em relação aos parâmetros; $\underset{\sim}{\beta}$ é o vetor de parâmetros $[(m+1) \times 1]$; e $\underset{\sim}{\epsilon}$ é o vetor de resíduos $(n \times 1)$ não observáveis e com $E\left(\underset{\sim}{\epsilon}\right) = \underset{\sim}{0}$ e $V\left(\underset{\sim}{\epsilon}\right) = I\sigma^2$.

Na metodologia clássica de modelos lineares, onde se encontram os modelos de regressão linear, pressupomos que exista uma linearidade nos parâmetros do preditor e aditividade dos erros e, ainda, que os erros são independentes, têm média zero, variância constante e que sua distribuição seja normal, ou seja, $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$. Além disso outras condições são importantes, como por exemplo, supomos que algumas poucas observações não devam ter influência demasiada sobre as estimativas dos parâmetros

do modelo e de suas variâncias. Assim, diagnósticos numéricos são funções dos dados cujos valores permitem detectar respostas que são anormalmente grandes ou pequenas (outliers ou valores discrepantes) ou que estão afastadas do grupo majoritário dos dados, influenciando em demasia o ajustamento. Assim, temos interesse particular nas análises denominadas de influência, onde utilizamos um conjunto de técnicas destinadas a detecção de pontos influentes e/ou discrepantes que podem afetar o ajustamento.

Muitas causas podem ser atribuídas a alguns problemas normalmente encontrados na análise de regressão. Algumas destas possibilidades são, entre outras, devidas à medidas erradas ou erro no registro da realização da variável resposta, ou ainda, erros de transcrição; observações tomadas em condições distintas das demais; modelo mal especificado; e distribuição não normal dos resíduos, apesar de o modelo e a escala estarem corretos.

A forma utilizada normalmente para verificar a influência de uma observação é retirá-la do modelo e verificar como as estimativas dos parâmetros, predições e variâncias são afetadas. Assim, se retirarmos a i-ésima observação e reestimarmos as quantidades mais importantes do modelo, poderemos avaliar a influência da observação retirada na estimação destes parâmetros de interesse. Podemos, no entanto, evitar que todos os cálculos sejam refeitos, utilizando algumas relações e propriedades apresentadas por Velleman e Welsch, (1981). Vários métodos de avaliar a influência de observações no ajuste de um modelo de regressão linear são apresentados por Chatterjee e Hadi (1986).

3.5.1 Análise de resíduos

O preditor dos resíduos é dado por:

$$\underset{\sim}{e} = \underset{\sim}{Y} - X\underset{\sim}{\hat{\beta}} \tag{3.5.1}$$

Podemos reescrever o erro como uma combinação linear de $\underset{\sim}{Y}$ por:

$$\underset{\sim}{e} = \underset{\sim}{Y} - X(X'X)^{-1}X'\underset{\sim}{Y} = [I - X(X'X)^{-1}X']\underset{\sim}{Y}$$

A matriz $X(X'X)^{-1}X'$ é denominada projetor e representada por P, pois projeta o vetor de observações $\underset{\sim}{Y}$, n-dimensional, no sub-espaço $(m+1)$-dimensional. Aplicando esta matriz ao vetor de observações, obtemos o vetor de valores preditos $\underset{\sim}{\hat{Y}}$, ou seja, $\underset{\sim}{\hat{Y}} = P\underset{\sim}{Y}$. Na análise de regressão linear simples, a matriz P é denominada de matriz *Hat* e representada por H. Vamos representar a i-ésima observação pelo vetor composto por $[Y_i \quad \underset{\sim}{z_i}']'$, sendo que $\underset{\sim}{z_i} = [1 \quad X_{1i} \quad X_{2i} \quad \cdots \quad X_{mi}]'$ é o vetor dos elementos da i-ésima linha da matriz X do modelo. O elemento da diagonal correspondente na matriz H é denominado simplesmente por h_i. Assim,

$$\underset{\sim}{e} = (I - H)\underset{\sim}{Y} \tag{3.5.2}$$

é o preditor do vetor de erros, que é equivalente a equação (3.5.1).

A esperança de $\underset{\sim}{e}$ é dada por:

$$\begin{aligned} E\left(\underset{\sim}{e}\right) =& E\left[(I - H)\underset{\sim}{Y}\right] = (I - H)E\left(\underset{\sim}{Y}\right) \\ =& [I - X(X'X)^{-1}X']X\underset{\sim}{\beta} = X\underset{\sim}{\beta} - X(X'X)^{-1}X'X\underset{\sim}{\beta} \\ =& X\underset{\sim}{\beta} - X\underset{\sim}{\beta} = \underset{\sim}{0} \end{aligned}$$

Assim, a covariância do vetor de resíduos preditos é:

$$\begin{aligned} V\left(\underset{\sim}{e}\right) =& (I - H)V\left(\underset{\sim}{Y}\right)(I - H) = (I - H)I\sigma^2(I - H)' \\ =& (I - H)(I - H')\sigma^2 = (I - H) - (I - H)H' \\ =& (I - H - H' + HH')\sigma^2 = (I - H - H + H)\sigma^2 \\ =& (I - H)\sigma^2 \end{aligned}$$

Para a i-ésima observação temos que a variância $V(e_i)$ é dada por:

$$V(e_i) = (1 - h_i)\sigma^2 \tag{3.5.3}$$

em que e_i é o i-ésimo elemento do vetor de resíduos preditos, ou seja, é o erro predito para a i-ésima observação. Neste contexto é denominado de resíduo ordinário.

O problema básico destes resíduos é que eles não são comparáveis entre si, por possuírem variâncias distintas. Devemos buscar alguma forma de padronização para termos a mesma dispersão em todos os n resíduos preditos. Temos basicamente três formas de padronizações que podemos efetuar e que discutiremos na sequência. Podemos ter os resíduos padronizados, resíduos estudentizados internamente e resíduos estudentizados externamente, também conhecidos por resíduos de *jackknife* (Chatterjee e Hadi, 1986). Em todos os casos vamos substituir a variância σ^2 pelo seu estimador $S^2 = QME$.

A primeira opção, não computada pelo R, é obtida pela divisão dos resíduos ordinários pelo desvio padrão $S = \sqrt{QME}$. Este artifício reduz a variabilidade a uma faixa específica, mas não elimina o problema de variâncias distintas. Este resíduo padronizado é dado por:

$$z_i = \frac{e_i}{S} \tag{3.5.4}$$

Pela razão anteriormente apontada, os resíduos estudentizados foram propostos na literatura especializada. Os resíduos estudentizados internamente são obtidos por meio da razão entre o resíduo ordinário e o seu estimador do erro padrão específico, ou seja, por

$$r_i = \frac{e_i}{\sqrt{(1-h_i)S^2}} \tag{3.5.5}$$

Este tipo de resíduo é mais interessante que o anterior, devido ao fato de considerar a variância individual de cada resíduo ordinário. Entretanto, se a i-ésima observação for um *outlier* pode ocorrer que a estimativa da variância estará afetada por este valor.

A última proposta de padronização foi feita para contornar este problema e tem ainda algumas propriedades mais interessantes do que as demais formas de padronização. Esta última padronização resulta nos resíduos estudentizados externamente, também denominados de resíduos de *jackknife*. A ideia é eliminar a i-ésima observação e obtermos uma estima-

dor da variância, digamos, $S^2_{(i)}$. O subscrito i apresentado entre parênteses foi utilizado para indicar que se trata de um estimador aplicado a todos as $n-1$ observações resultante da eliminação da i-ésima observação da amostra completa. Felizmente, não precisamos reajustar o modelo eliminando a i-ésima observação para obtermos uma estimativa desta variância (Chatterjee e Hadi, 1986). Um estimador obtido a partir da análise original (Beckman e Trussell, 1974) é dado por:

$$S^2_{(i)} = \frac{(n-m-1)S^2}{n-m-2} - \frac{e_i^2}{(n-m-2)(1-h_i)} \tag{3.5.6}$$

O resíduo estudentizado externamente é definido por:

$$t_i = \frac{e_i}{\sqrt{(1-h_i)S^2_{(i)}}} \tag{3.5.7}$$

Este resíduo é denominado por *RSTUDENT* na literatura especializada de regressão. Observações que apresentarem este tipo de resíduo superior em módulo a 2, devem receber atenção especial. Existe uma preferência por este tipo de resíduo na literatura e as razões para isso podem ser apontadas (Chatterjee e Hadi, 1986) por:

- Os resíduos estudentizados externamente t_i sob a hipótese de normalidade seguem a distribuição t de Student com $\nu = n-m-2$ graus de liberdade, enquanto $r_i^2/(n-m-1)$ segue a distribuição beta;

- podemos mostrar facilmente que:

  $$t_i = r_i\sqrt{\frac{n-m-2}{n-m-1-r_i^2}}$$

  de onde se observa que t_i é uma transformação monotônica de r_i e que $t_i \to \infty$ à medida que $r_i \to (n-m-1)$. Assim, t_i reflete um resíduo fora de faixa de forma mais acentuada do que faz r_i; e

- o estimador $S^2_{(i)}$ é robusto à grandes e grosseiros erros da i-ésima observação, ou seja, se esta observação for discrepante.

É importante ressaltarmos que a detecção de valores discrepantes não deve implicar em descarte automático de observações. É possível, por exemplo, que o valor discrepante se deva a erro de transcrição, situação em que esse valor pode ser facilmente corrigido ou então pode ser um indicativo de modelo inadequado, possibilitando que modelos melhores sejam adotados e ajustados.

3.5.2 Influência no Espaço das Variáveis Preditoras

Além dos resíduos podemos verificar a influência das observações em uma série de quantidades importantes da análise de regressão. Uma interessante medida de diagnóstico é o próprio elemento h_i da matriz de projeção H. Esta estatística é denominada de influência (*leverage*). O critério utilizado é baseado em algumas propriedades (Velleman e Welsch, 1981) de h_i, dadas por: $0 \leq h_i \leq 1$ e $\sum_{i=1}^{n} h_i = (m+1)$. Assim, o valor médio da influência é $(m+1)/n$. Como $h_i = \partial\hat{Y}_i/\partial Y_i$, uma estimativa igual a zero é indicativo de que não há influência no ajuste do modelo e uma estimativa igual a 1, é indicativo que um grau de liberdade foi efetivamente atribuído ao ajuste daquela observação. O problema é determinar quais observações amostrais têm alta influência no ajuste e, portanto, receber atenção especial. Se $m > 14$ e $(n-m) > 31$ podemos utilizar o critério de que a i-ésima observação merece atenção se $h_i > 2(m+1)/n$. Se estas condições envolvendo m e n não forem verificadas, podemos utilizar $h_i > 3(m+1)/n$ como um melhor critério.

Devemos chamar a atenção de que a influência medida pelo h_i refere-se ao papel das variáveis regressoras (fatores). Assim, medimos a influência, com h_i, no espaço dos fatores e, com a análise de resíduos, no espaço da variável resposta. Assim, a influência pode ocorrer no espaço dos fatores, no espaço das respostas ou em ambos os casos.

3.5.3 Influência no Vetor de Estimativas dos Parâmetros

A ideia de medir a influência da i-ésima observação na estimativa do vetor de parâmetros pode ser desenvolvida a partir da eliminação desta observação. Após esta eliminação, estimamos novamente os parâmetros do modelo e aplicamos uma medida de distância entre as estimativas. Esta distância pode ser dada pela diferença entre as estimativas obtidas com

e sem a eliminação da i-ésima observação. Em geral é isso que fazemos, tomando-se o cuidado apenas de padronizar os resultados. Seja $\hat{\beta_{ij}}$, o estimador do j-ésimo parâmetro após eliminarmos a i-ésima observação, para $i = 1,2,\cdots,n$ e $j = 0,1,\cdots,m$. A estatística que utilizaremos para isso é conhecida por $DFBETA_{ij}$, em que DF são as iniciais de *Deviation of Fit*. Por meio dela podemos determinar a influência de cada observação na estimativa de cada parâmetro do modelo. Esta estatística é dada por:

$$DFBETA_{ij} = \frac{\hat{\beta}_j - \hat{\beta_{ij}}}{\hat{V}\left(\hat{\beta}_j\right)} \tag{3.5.8}$$

A dificuldade é obter as estimativas do vetor de parâmetros para cada um dos n casos, em que um das variáveis é eliminada. Felizmente, não precisamos estimar n vezes o vetor de parâmetros para calcularmos os $DFBETAS$. Existe uma relação interessante (Chatterjee e Hadi, 1986) para a diferença entre os vetor de estimativas com e sem a i-ésima observação que é dada por:

$$\underset{\sim}{\hat{\beta}} - \underset{\sim}{\hat{\beta_{(i)}}} = \frac{1}{1-h_i}(X'X)^{-1}\underset{\sim}{Z_i}e_i \tag{3.5.9}$$

em que $\underset{\sim}{\hat{\beta_{(i)}}}$ é o estimador do vetor de parâmetros após a eliminação da i-ésima observação.

Também sabemos que o vetor de estimadores dos parâmetros é dado por:

$$\underset{\sim}{\hat{\beta}} = (X'X)^{-1}X'\underset{\sim}{Y} = C\underset{\sim}{Y} \tag{3.5.10}$$

Assim, o $DFBETA$ não padronizado é dado por:

$$DFBETA_{ij} = c_{ji}\frac{e_i}{1-h_i} \tag{3.5.11}$$

em que c_{ji} é o elemento da j-ésima linha e i-ésima coluna da matriz $C = (X'X)^{-1}X'$.

Se a expressão (3.5.11) for dividida pelo erro padrão do vetor de parâmetros $\hat{V}\left(\hat{\beta}_j\right)$, obteremos uma expressão equivalente (3.5.8). A expressão resultante é utilizada para obtermos os $DFBETAS$, sendo dada por:

$$DFBETA_{ij} = \frac{c_{ji}t_i}{\sqrt{(1-h_i)\underset{\sim}{C_j}'\underset{\sim}{C_j}}} \tag{3.5.12}$$

em que $\underset{\sim}{C_j}$ é vetor obtido a partir da j-ésima linha da matriz C.

Estas estatísticas são muito dependentes do número de observações, sendo que tanto menor será o efeito da observação sobre os valores de $DFBETAS$, quanto maior for o número de observações. Para estabelecer um valor limite para essa estatística, podemos tomar como base o valor limite para os resíduos, que é igual a 2. Assim, teremos que observações cujos $|DFBETA_{ij}| > 2/\sqrt{n}$ devem ter atenção especial, pois o vetor de estimativas pode ter sofrido alterações significativas.

3.5.4 Influência no Vetor de Valores Preditos

O impacto da i-ésima observação no i-ésimo valor predito pode ser medido pela padronização da mudança no valor predito na presença e ausência desta observação. A estatística utilizada para fazer tal mensuração é denominada de $DFFITS$ e é dada por:

$$DFFITS_i = \frac{\left|\underset{\sim}{Y_i} - \underset{\sim}{\hat{Y}_{i(i)}}\right|}{\sqrt{(1-h_i)S^2_{(i)}}} = |t_i|\sqrt{\frac{h_i}{1-h_i}} \tag{3.5.13}$$

Podemos verificar que quanto maior a influência da i-ésima observação, mais h_i se aproxima de 1 e, consequentemente, maior será o coeficiente $|t_i|$. Como vimos anteriormente $h_i/(1-h_i)$ está relacionada a uma medida da distância entre as linhas de X. Assim, a grandeza do valor de $DFFITS$ pode ser atribuída à discrepância do valor da resposta, do conjunto de valores das variáveis preditoras ou de ambos. Um ponto geral para a determinação de observações influentes é considerado o valor 2. Um ponto de corte ajustado para determinar a influência é $2\sqrt{(m+1)/n}$.

A distância de Cook é outra estatística utilizada para medir a influência de uma observação na predição dos valores da variável resposta Y. Esta estatística pode ser vista como a distância Euclidiana entre os valores preditos com e sem a i-ésima observação. O estimador da distância de Cook é dado por:

$$D_i = \frac{1}{(m+1)} \frac{h_i}{(1-h_i)} r_i^2 \tag{3.5.14}$$

Apesar de que a distância de Cook não deva ser usada como teste de significância, sugere-se o uso dos quantis da distribuição F central com $m+1$ e $n-m-1$ graus de liberdade para servir de referência para o valor D_i. Outros autores sugerem que se $D_i > 1$, a i-ésima observação deve ser considerada influente.

A distância de Cook utiliza r_i^2, sendo que implicitamente está utilizando S^2 para padronizar a variância. Existe uma sugestão de que esta estatística possa ter melhores propriedades se for utilizado o estimador $S^2_{(i)}$ no lugar de S^2. Assim, a distância modificada de Cook utiliza esta substituição e faz um ajuste para o número de observações e toma ainda a raiz quadrada da distância transformada. A distância modificada de Cook é dada por:

$$D_i^* = |t_i| \sqrt{\frac{h_i(n-m-1)}{(1-h_i)(m+1)}} = DFFITS \sqrt{\frac{n-m-1}{m+1}} \tag{3.5.15}$$

Com essa modificação, temos que: a nova estatística enfatiza mais os pontos extremos; o gráfico de probabilidade normal pode ser utilizado para checagem; nos casos perfeitamente balanceados $[h_i = (m+1)/n]$ para qualquer i, a distância modificada tem comportamento idêntico ao $DFFITS$; a distância modificada com sinal pode ser plotada contra variáveis exploratórias do modelo.

Dado o limite máximo estabelecido para $DFFITS$, um valor da distância modificada de Cook maior que 2 pode ser considerado um indicativo de observação influente.

3.5.5 Influência na Matriz de Covariâncias

Uma medida da influência da i-ésima observação na $V\left(\underset{\sim}{\hat{\beta}}\right)$ é obtida comparando a razão de variâncias generalizadas (determinantes) da estimativa da covariância com e sem a i-ésima observação. Esta estatística é dada por:

$$COVRATIO_i = \frac{\det\left[S^2_{(i)}\left(X'_{(i)}X_{(i)}\right)^{-1}\right]}{\det\left[S^2\left(X'X\right)^{-1}\right]}$$

$$= \frac{\left(\frac{n-m-1-r_i^2}{n-m-2}\right)^{m+1}}{(1-h_i)} \qquad (3.5.16)$$

em que $X_{(i)}$ é a matriz do modelo obtida após a eliminação da i-ésima observação amostral.

Um valor não muito preciso para determinar pontos influentes é dado por $|COVRATIO_i - 1| > 3(m+1)/n$.

3.5.6 Comandos R

Felizmente todas estes métodos de diagnóstico em regressão linear podem ser obtidas utilizando duas opções simples do comandos model: *r* e *influence.measures*. Apresentamos na sequência um exemplo do programa R utilizado para obter o diagnóstico de regressão para o exemplo do volume de madeira das árvores.

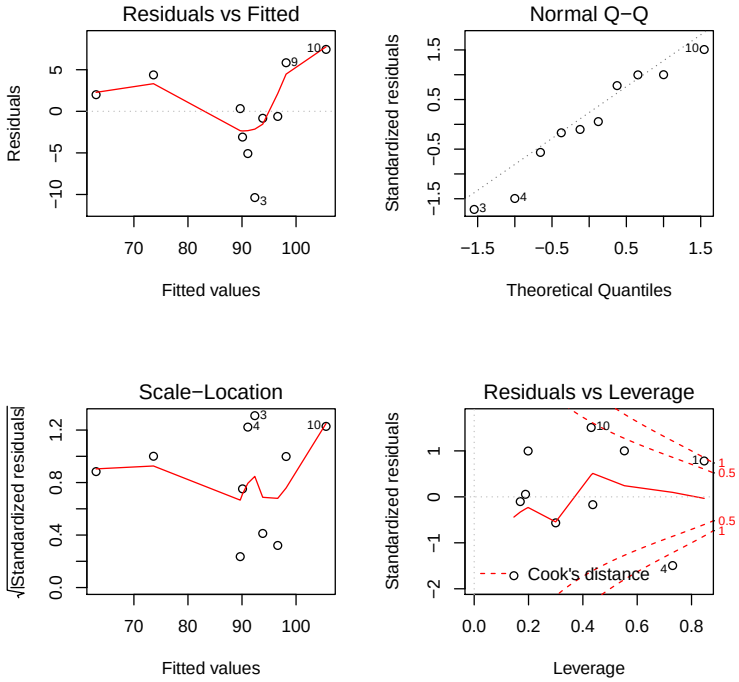
```
> # medidas de influência e análise de resíduo
> attach(arvores)

The following object(s) are masked from 'arvores (position 4)':

    X1, X2, X3, Y

> result <- lm(Y~X1+X2+X3) # objeto lm
> par(mfrow = c(2,2)) # vários gráficos
> plot(result)
> influence.measures(result) # medidas de influência
```

```
Influence measures of
         lm(formula = Y ~ X1 + X2 + X3) :

    dfb.1_   dfb.X1  dfb.X2   dfb.X3   dffit cov.r
1   0.3969  0.17372  0.4576 -0.22529  1.7660 8.804
2   0.1151 -0.07019 -0.9211  0.06560  1.1128 2.233
3  -0.4150 -0.45122 -0.0613  0.44512 -0.9074 0.164
4   2.5208  2.60230 -0.5359 -2.59173 -2.8351 1.193
5  -0.0178  0.00195  0.2653 -0.00696 -0.3474 2.379
6   0.0123  0.01169 -0.0117 -0.01144  0.0245 2.551
7  -0.0694 -0.08053 -0.0828  0.08099 -0.1369 3.610
8  -0.0116 -0.01559 -0.0101  0.01508 -0.0425 2.479
9   0.1055  0.14433 -0.0707 -0.13430  0.4960 1.253
10 -0.9286 -0.74811  0.8846  0.76341  1.5196 0.541
     cook.d   hat inf
1  0.840545 0.847   *
2  0.309472 0.553
3  0.125873 0.146
4  1.513327 0.730   *
5  0.034282 0.300
6  0.000179 0.189
7  0.005599 0.437   *
8  0.000541 0.170
9  0.061585 0.199
10 0.430108 0.431
```

```
> infl = lm.influence(result, do.coef = FALSE)
> rstudent(result, infl, res = infl$wt.res)
```

```
          1           2           3           4
 0.75186592  1.00094930 -2.19346862 -1.72242928
          5           6           7           8
-0.53075412  0.05069031 -0.15558711 -0.09403162
          9          10
 0.99663965  1.74733620
```

```
> detach(arvores)
```



3.6 Exercícios

1. Utilize os dados do exemplo da amostra de $n = 10$ árvores e ajuste o seguinte modelo:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \beta_4 X_{1i} X_{2i} + \beta_5 \frac{1}{X_{3i}} + \epsilon_i$$

2. Existe alguma variável redundante? Se houver utilize os métodos de seleção de modelos apresentados neste capítulo e determine qual é o melhor modelo.

3. Os métodos de seleção de modelo chegaram a um mesmo modelo?

4. Para o modelo final utilizar as opções apresentadas e verificar a qualidade da predição, fazer o gráfico dos valores preditos e do intervalos de confiança para média e para o valor futuro e plotar os resíduos em relação aos valores preditos na abscissa.

5. Utilize variáveis candidatas diferentes das apresentadas no exercício (1) e aplique os métodos de seleção de modelos. Você chegou a um modelo melhor do que o anteriormente obtido? Justifique devidamente suas conclusões.

6. Utilizando os dados da amostra de $n = 10$ árvores ajuste o modelo:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \beta_4 X_{1i} X_{2i} + \beta_5 \frac{1}{X_{3i}} + \epsilon_i$$

Faça a diagnose de influência e verifique se existe alguma observação influente. Justifique devidamente suas conclusões.

Capítulo 4

Regressão Não-Linear

Outro assunto extremamente importante para os pesquisadores, em geral, é o ajuste de regressões não-lineares em suas pesquisas aplicadas. Temos o objetivo de apresentar neste capítulo as principais idéias sobre os processos de estimação de parâmetros de modelos não-lineares e as funções R como, por exemplo, *nls*, para realizar essa tarefa. O que devemos considerar é que os modelos não-lineares nos parâmetros têm uma maior plasticidade e, portanto, são considerados mais apropriados para modelarem fenômenos biológicos.

Nesse capítulo vamos discutir um pouco sobre métodos de estimação de parâmetros de modelos não-lineares e sobre a sintaxe da função principal *nls*. Vamos apresentar programas de modelos de *Response Plateau* linear e não-linear. Ambos são não-lineares nos parâmetros, mas descrevem curvas lineares e quadráticas, respectivamente, além do *plateau* no ponto de junção dos segmentos, que é uma linha reta paralela à abscissa.

Os procedimentos de estimação não-linear são, em geral, iterativos. O processo deve iniciar com um valor específico arbitrário de seus parâmetros e a soma de quadrado do resíduo deve ser determinada. Então, uma nova estimativa dos parâmetros é obtida, buscando-se minimizar a soma de quadrados do resíduo. Este processo é repetido até que esse mínimo seja alcançado. Vários algoritmos e métodos existem para realizar esse processo de estimação. Faremos uma descrição detalhada desses métodos, que aceleram a convergência e são eficientes para estimarmos os parâmetros que conduzem ao mínimo global para a soma de quadrados de resíduos.

4.1 Introdução aos Modelos Não-Lineares

Um modelo é considerado não-linear nos parâmetros e essa classificação não é influenciada pela função matemática descrita (hipérbole, parábola, etc.). Como já dissemos no capítulo 3, se as derivadas parciais forem funções dos próprios parâmetros, teremos um modelo não-linear. Podemos ter múltiplos parâmetros no modelo ou apenas um e, da mesma forma, podemos ter apenas uma variável regressora ou mais de uma. Assim, $Y = \alpha\beta^Z$ é um modelo não-linear com dois parâmetros α e β e $Y = \alpha + \beta Z^2$ é um modelo linear, independentemente de a função descrever uma parábola, pois este modelo é linear em relação aos parâmetros α e β.

Os detalhes computacionais envolvidos nos procedimentos não-lineares são muito complexos. Vamos simplificar o máximo que pudermos, sem no entanto, deixarmos de ter o rigor necessário. Seja o modelo não-linear F definido de forma geral para o vetor de parâmetros $\underset{\sim}{\beta} = [\beta_1 \quad \beta_2 \quad \cdots \quad \beta_m]'$ e para o vetor de variáveis regressoras da j-ésima unidade amostral $\underset{\sim}{Z}'_j = [Z_{1j} \quad Z_{2j} \quad \cdots \quad Z_{pj}]$ por

$$Y_j = F_j\left(\underset{\sim}{\beta}, \underset{\sim}{Z}_j\right) + \epsilon_j. \tag{4.1.1}$$

Podemos expressar este modelo em notação matricial por:

$$\underset{\sim}{Y} = \underset{\sim}{F}\left(\underset{\sim}{\beta}\right) + \underset{\sim}{\epsilon}. \tag{4.1.2}$$

em que podemos expressar o vetor do modelo $\underset{\sim}{F}\left(\underset{\sim}{\beta}\right)$, simplesmente por $\underset{\sim}{F}$.

Para ficar claro a notação que estamos utilizando, consideremos o modelo $Y_j = \alpha\theta^{Z_j} + \epsilon_j$. Nesse caso temos um vetor de parâmetros dado por $\underset{\sim}{\beta}' = [\alpha \quad \theta]$ e uma única variável regressora Z. O vetor do modelo é dado por:

$$\underset{\sim}{F} = \begin{bmatrix} \alpha\theta^{Z_1} \\ \alpha\theta^{Z_2} \\ \vdots \\ \alpha\theta^{Z_n} \end{bmatrix}$$

O vetor de observações é dado por:

$$\underset{\sim}{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}$$

Finalmente, o vetor de resíduos é dado por:

$$\underset{\sim}{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

O modelo pode ser escrito por:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} \alpha\theta^{Z_1} \\ \alpha\theta^{Z_2} \\ \vdots \\ \alpha\theta^{Z_n} \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

Um dos métodos utilizados baseia-se na minimização da soma de quadrados dos resíduos $L\left(\underset{\sim}{\beta}\right) = \underset{\sim}{\epsilon}'\underset{\sim}{\epsilon}$. Substituindo $\underset{\sim}{\epsilon} = \underset{\sim}{Y} - \underset{\sim}{F}$ e derivando com respeito a $\underset{\sim}{\beta}$, obtivemos:

$$L\left(\underset{\sim}{\beta}\right) = \underset{\sim}{\epsilon}'\underset{\sim}{\epsilon} = \left(\underset{\sim}{Y} - \underset{\sim}{F}\right)'\left(\underset{\sim}{Y} - \underset{\sim}{F}\right) = \underset{\sim}{Y}'\underset{\sim}{Y} - 2\underset{\sim}{Y}'\underset{\sim}{F} + \underset{\sim}{F}'\underset{\sim}{F}$$

$$\frac{\partial L}{\partial \underset{\sim}{\beta}} = \frac{-\partial 2\underset{\sim}{Y}'\underset{\sim}{F}}{\partial \underset{\sim}{\beta}} + \frac{\partial \underset{\sim}{F}'\underset{\sim}{F}}{\partial \underset{\sim}{\beta}}$$

Mas,

$$\frac{-\partial 2\underset{\sim}{Y}'\underset{\sim}{F}}{\partial \underset{\sim}{\beta}} = \frac{-\partial 2\underset{\sim}{Y}'\underset{\sim}{F}}{\partial \underset{\sim}{F}} \times \frac{\partial \underset{\sim}{F}}{\partial \underset{\sim}{\beta}} = -2\underset{\sim}{Y}'X$$

em que $X = \partial \underset{\sim}{F}/\partial \underset{\sim}{\beta}$ é a matriz de derivadas parciais, em que cada coluna é formada pela derivada da função linear em relação aos parâmetros.

Também podemos simplificar $\partial \underset{\sim}{F}' \underset{\sim}{F}/\partial \underset{\sim}{\beta}$ por:

$$\frac{\partial \underset{\sim}{F}' \underset{\sim}{F}}{\partial \underset{\sim}{\beta}} = \frac{\partial \underset{\sim}{F}' \underset{\sim}{F}}{\partial \underset{\sim}{F}} \times \frac{\partial \underset{\sim}{F}}{\partial \underset{\sim}{\beta}} = 2\underset{\sim}{F}' X$$

Logo,

$$\frac{\partial L}{\partial \underset{\sim}{\beta}} = -2\underset{\sim}{Y}' X + 2\underset{\sim}{F}' X$$

Igualando a zero a primeira derivada, temos as equações normais para os modelos não-lineares:

$$X' \underset{\sim}{F} = X' \underset{\sim}{Y} \tag{4.1.3}$$

Como $\underset{\sim}{F}$ e X são funções de $\underset{\sim}{\beta}$, então uma forma fechada para a solução, em geral, não existe. Então devemos utilizar um processo iterativo. Para isso precisamos de um valor inicial para o vetor de parâmetros, que deve ser melhorado continuamente até que a soma de quadrados de resíduos $\underset{\sim}{\epsilon}' \underset{\sim}{\epsilon}$ seja minimizada.

Se considerarmos o modelo $Y_j = \alpha\theta^{Z_j} + \epsilon_j$, que utilizamos anteriormente para ilustrar alguns aspectos do modelo, podemos construir a matriz X das derivadas parciais facilmente. Sejam as derivadas parciais $\partial Y_j/\partial\alpha = \theta^{Z_j}$ e $\partial Y_j/\partial\theta = Z_j\alpha\theta^{(Z_j-1)}$

$$X = \begin{bmatrix} \theta^{Z_1} & Z_1\alpha\theta^{(Z_1-1)} \\ \theta^{Z_2} & Z_2\alpha\theta^{(Z_2-1)} \\ \vdots & \vdots \\ \theta^{Z_n} & Z_n\alpha\theta^{(Z_n-1)} \end{bmatrix}$$

As equações normais para este exemplo são:

$$\begin{bmatrix} \theta^{Z_1} & \dots & \theta^{Z_n} \\ Z_1\alpha\theta^{(Z_1-1)} & \dots & Z_n\alpha\theta^{(Z_n-1)} \end{bmatrix} \begin{bmatrix} \alpha\theta^{Z_1} \\ \alpha\theta^{Z_2} \\ \vdots \\ \alpha\theta^{Z_n} \end{bmatrix} =$$

$$= \begin{bmatrix} \theta^{Z_1} & \dots & \theta^{Z_n} \\ Z_1\alpha\theta^{(Z_1-1)} & \dots & Z_n\alpha\theta^{(Z_n-1)} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}$$

Devemos iniciar o processo iterativo para um determinado valor inicial $\underset{\sim}{\beta}_0$. Para o valor corrente (k-ésimo passo do processo iterativo) do vetor de parâmetros, devemos calcular a matriz X e estimar o vetor de resíduos por $\underset{\sim}{e} = \underset{\sim}{Y} - \underset{\sim}{F}\left(\underset{\sim}{\beta}_k\right)$. No ponto inicial ($k = 0$), avaliamos X e o vetor de resíduos, considerando o valor arbitrário do vetor de parâmetros especificado. Neste caso, se $SQE\left(\underset{\sim}{\beta}_k\right) = \underset{\sim}{e}'\underset{\sim}{e}$ for a soma de quadrados dos resíduos avaliada na k-ésima iteração, então X e $\underset{\sim}{Y}$ são usados para calcular um vetor $\underset{\sim}{\Delta}$ de tal forma que

$$SQE\left(\underset{\sim}{\beta}_k + \lambda\underset{\sim}{\Delta}\right) < SQE\left(\underset{\sim}{\beta}_k\right)$$

para uma constante λ qualquer.

Existem quatro métodos implementados no R. Estes quatro métodos diferem na forma como $\underset{\sim}{\Delta}$ é calculado para propiciar as trocas no vetor de parâmetros. De uma forma geral, os critérios básicos são:

$$\begin{cases} \text{Gradiente:} & \underset{\sim}{\Delta} = & X'\underset{\sim}{e} \\ \text{Gauss-Newton:} & \underset{\sim}{\Delta} = & (X'X)^{-}X'\underset{\sim}{e} \\ \text{Newton:} & \underset{\sim}{\Delta} = & G^{-}X'\underset{\sim}{e} \\ \text{Marquardt:} & \underset{\sim}{\Delta} = & [X'X + \delta\text{diag}(X'X)]^{-}X'\underset{\sim}{e} \end{cases} \tag{4.1.4}$$

em que $(X'X)^-$ é uma inversa generalizada. Pode ser uma inversa reflexiva (g_2), mas o ideal é que seja uma inversa de Moore-Penrose (g_4).

Os métodos Gauss-Newton e Marquardt realizam a regressão dos resíduos em relação as primeiras derivadas do modelo não-linear em relação aos parâmetros, até que haja a convergência. O método de Newton faz a regressão destes resíduos em relação a uma função das segundas derivadas do modelo não-linear com relação aos parâmetros (G^-).

4.1.1 Método do Gradiente

Este método é baseado no gradiente ou grau de variação de $\underset{\sim}{\epsilon}'\underset{\sim}{\epsilon}$. Seja $\underset{\sim}{\beta_k}$ a estimativa do vetor de parâmetros na k-ésima iteração do processo. Assim, este gradiente é definido por:

$$\frac{1}{2}\frac{\partial L\left(\underset{\sim}{\beta_k}\right)}{\partial \underset{\sim}{\beta_k}} = -X'\underset{\sim}{Y} + X'\underset{\sim}{F} = -X'\underset{\sim}{e}$$

pois X e $\underset{\sim}{F}$ são avaliados no ponto $\underset{\sim}{\beta_k}$.

A quantidade $-X'\underset{\sim}{e}$ é o gradiente para o qual $\underset{\sim}{\epsilon}'\underset{\sim}{\epsilon}$ cresce. Sendo assim, $\underset{\sim}{\Delta} = X'\underset{\sim}{e}$ é o grau de variação para o método de gradiente. Para utilizarmos o método do gradiente devemos inicialmente estipular um valor arbitrário para o vetor de parâmetros, digamos $\underset{\sim}{\beta_0}$. Calculamos $\underset{\sim}{e}$ e $\underset{\sim}{\Delta}$. Assim, podemos obter o valor do parâmetro no (k+1)-ésimo passo, tomando a estimativa do k-ésimo passo anterior por:

$$\underset{\sim}{\beta_{k+1}} = \underset{\sim}{\beta_k} + \lambda\underset{\sim}{\Delta} \tag{4.1.5}$$

em que o escalar λ é escolhido no k-ésimo passo para que

$$SQE\left(\underset{\sim}{\beta_k} + \lambda\underset{\sim}{\Delta}\right) < SQE\left(\underset{\sim}{\beta_k}\right). \tag{4.1.6}$$

O método do gradiente possui convergência muito lenta e, em geral, não é utilizado para estimar parâmetros dos modelos não-lineares. Quando, no

entanto, as estimativas iniciais são pobres, este método se torna particularmente útil.

4.1.2 Método de *Newton*

O método de *Newton* utiliza a segunda derivada do erro em relação aos parâmetros e obtém o vetor $\underset{\sim}{\Delta}$ por:

$$\underset{\sim}{\Delta} = G^{-} X' \underset{\sim}{e} \tag{4.1.7}$$

em que

$$G = (X'X) + \sum_{j=1}^{n} H_j \left(\underset{\sim}{\beta_k} \right) e_j \tag{4.1.8}$$

sendo que a matriz H_j, de dimensão $r \times r$, avaliada para o vetor de parâmetros $\underset{\sim}{\beta_k}$ no k-ésimo passo para a j-ésima observação amostral, é a matriz Hessiana do vetor de erros $\underset{\sim}{\epsilon}$. O elemento (ℓ, k) desta matriz, $[H_j]_{\ell k}$, é dado por:

$$[H_j]_{\ell k} = \left[\frac{\partial^2 \epsilon_j}{\partial \beta_\ell \partial \beta_k} \right]_{\ell k} \tag{4.1.9}$$

Estimado o vetor $\underset{\sim}{\Delta}$, devemos aplicar as equações (4.1.5) e (4.1.6) para obtermos uma nova equação e recalcularmos o vetor de parâmetros.

Para o exemplo anterior, considerando o modelo $Y_j = \alpha\theta^{Z_j} + \epsilon_j$, a matriz de segundas derivadas para a j-ésima observação é:

$$H_j = \begin{bmatrix} 0 & -Z_j\theta^{(Z_j-1)} \\ -Z_j\theta^{(Z_j-1)} & -Z_j(Z_j-1)\alpha\theta^{(Z_j-2)} \end{bmatrix}$$

4.1.3 Método de *Gauss-Newton*

O método de *Gauss-Newton* usa a expansão em série de *Taylor* do vetor de funções

$$\underset{\sim}{F}\left(\underset{\sim}{\beta}\right) = \underset{\sim}{F}\left(\underset{\sim}{\beta_0}\right) + X\left(\underset{\sim}{\beta} - \underset{\sim}{\beta_0}\right) + \cdots$$

em que a matriz de primeiras derivadas X é avaliada no ponto $\underset{\sim}{\beta}_0$.

Se substituirmos os dois termos desta expansão nas equações normais obtemos

$$\begin{aligned} X'\underset{\sim}{F}\left(\underset{\sim}{\beta}\right) &= X'\underset{\sim}{Y} \\ X'\left[\underset{\sim}{F}\left(\underset{\sim}{\beta}_0\right) + X\left(\underset{\sim}{\beta} - \underset{\sim}{\beta}_0\right)\right] &= X'Y \\ X'X\left(\underset{\sim}{\beta} - \underset{\sim}{\beta}_0\right) &= X'Y - X'\underset{\sim}{F}\left(\underset{\sim}{\beta}_0\right) \\ X'X\underset{\sim}{\Delta} &= X'\underset{\sim}{e} \end{aligned}$$

e portanto,

$$\underset{\sim}{\Delta} = (X'X)^{-}X'\underset{\sim}{e} \tag{4.1.10}$$

Estimado o valor de $\underset{\sim}{\Delta}$ para o vetor $\underset{\sim}{\beta}_0$, aplicam-se as equações (4.1.5) e (4.1.6) para se obter o vetor de estimativas do passo 1. O processo é repetido um determinado número de vezes até que o vetor de estimativas não se altere mais dentro de uma precisão pré-estipulada.

4.1.4 Método de *Marquardt*

O método de *Marquardt* mantém um compromisso entre o método de *Gauss-Newton* e o método do gradiente. A fórmula de atualização do vetor de parâmetros é dada por:

$$\underset{\sim}{\Delta} = \left[(X'X) + \delta \text{diag}(X'X)\right]^{-} X'\underset{\sim}{e} \tag{4.1.11}$$

Se $\delta \to 0$, há uma aproximação ao método de *Gauss-Newton* e se $\delta \to \infty$, há uma aproximação ao método do gradiente. Por padrão a *nls* começa com valor de $\delta = 10^{-7}$. Se $SQE\left(\underset{\sim}{\beta}_0 + \underset{\sim}{\Delta}\right) < SQE\left(\underset{\sim}{\beta}_0\right)$, então $\delta = \delta/10$ na próxima iteração; se por outro lado ocorrer o contrário, ou seja, se $SQE\left(\underset{\sim}{\beta}_0 + \underset{\sim}{\Delta}\right) > SQE\left(\underset{\sim}{\beta}_0\right)$, então $\delta = 10\delta$. Assim, se a soma de

quadrados do resíduo decresce a cada iteração, estaremos utilizando essencialmente o método de *Gauss-Newton*; se ocorrer o contrário o valor de δ é aumentado em cada iteração, sendo que passaremos a utilizar o método de gradiente.

4.1.5 Tamanho do passo da iteração

Devemos estipular o tamanho do passo que daremos em cada iteração. Assim, se $SQE\left(\underset{\sim}{\beta}_k + \lambda\underset{\sim}{\Delta}\right) > SQE\left(\underset{\sim}{\beta}_k\right)$, começando com $\lambda = 1$, devemos reduzir o valor pela metade em cada passo $SQE\left(\underset{\sim}{\beta}_k + 0{,}5\underset{\sim}{\Delta}\right)$, $SQE\left(\underset{\sim}{\beta}_k + 0{,}25\underset{\sim}{\Delta}\right)$, e assim por diante até que um quadrado médio do resíduo menor seja encontrado. Podemos muitas vezes encontrar dificuldades em obter avanços no processo iterativo. Quando isso acontece, o R interrompe o processo e comunica ao usuário de ocorrência de singularidade no gradiente no passo atual da iteração. A possível causa, em geral, é atribuída a valores iniciais inadequados.

4.2 A função *nls*

A função *nls* do pacote *stats* é o procedimento R apropriado para ajustarmos modelos não-lineares. O algoritmo padrão é o *Gauss-Newton*. Outra possibilidade é utilizar a opção *plinear* que indica ao R para utilizar o algoritmo de *Golub-Pereyra* de quadrados mínimos parciais ou a opção *port* para o algoritmo *nl2sol* do pacote *Port*. O usuário não precisa especificar as derivadas parciais no R. Isso é uma grande vantagem do R em relação a outros programas, pois o R possibilita o cálculo simbólico das derivadas parciais necessárias. No entanto, fornecer as derivadas parciais de primeira ordem da variável resposta em relação aos parâmetros pode reduzir o número de iterações, aumentando, portanto, a velocidade dos cálculos, aumentar a precisão numérica, e aumentar a chance de convergência. Em geral, as derivadas devem ser fornecidas sempre que possível. Fazer isso não é muito trivial no R e por isso daremos apenas um exemplo desse caso.

Vamos ilustrar nesta seção os comandos básicos para ajustarmos um modelo de regressão não-linear utilizando a função *nls*. Vamos especificar a forma de entrar com o modelo e, também, como escolher os métodos de

estimação a serem utilizados. Antes disso, devemos realizar algumas considerações a respeito de como atribuir valores iniciais para os parâmetros. Podemos utilizar, entre outras possibilidades, estimativas publicadas na literatura especializada, que utilizam modelos e conjuntos de dados similares aos de nossa pesquisa. Se o modelo pode ser linearizado, ignorando o fato de ter resíduos aditivos, podemos aplicar uma transformação para linearizá-lo e então, ajustar, o modelo linear resultante. As estimativas de quadrados mínimos ordinários, devidamente transformadas para a escala original, quando for o caso, são utilizadas como valores iniciais. Algumas vezes, antes da linearização, podemos efetuar algum tipo de reparametrização e proceder da mesma forma. Os processos iterativos possuem convergência bem mais rápida, quando os valores iniciais estão mais próximos dos valores reais.

Para apresentarmos os comandos básicos da *nls*, vamos utilizar os dados da Tabela 3.2 e o seguinte modelo não-linear nos parâmetros:

$$y_i = \alpha\beta^{x_i} + \epsilon_i. \tag{4.2.1}$$

Nesse caso temos $n = 8$ árvores e as seguintes derivadas parciais em relação aos parâmetros α e β: $\partial y_i/\partial\alpha = \beta^{x_i}$ e $\partial y_i/\partial\beta = x_i\alpha\beta^{(x_i-1)}$. Como estas derivadas parciais são funções dos parâmetros α e β, temos caracterizado um modelo não-linear nos parâmetros. Vamos atribuir valores iniciais arbitrários iguais a 0,5 e 1,8 para α e β, respectivamente. Poderíamos ter linearizado este modelo facilmente aplicando a função logaritmo, ignorando é claro o fato de o erro ser aditivo. Esse seria um artifício para obtermos valores iniciais mais acurados. O modelo linearizado é dado por $\ln(y_i) = \ln(\alpha)+\beta\ln(x_i)+\epsilon_i^*$, que poderia ser reescrito por $z_i = A+\beta w_i+\epsilon_i^*$. Nesse caso a estimativa do parâmetro A do modelo linear dever ser transformada para a escala original por $\hat{\alpha} = \exp(\hat{A})$. A estimativa de β não precisa ser modificada, pois o parâmetro β não foi alterado pela transformação efetuada. Realizar isso é deixado a cargo do leitor na forma de exercício. O programa R resultante é:

```
> # demonstrar como ajustar um modelo não-linear no R
> # utilizando a função nls
> # lê o arquivo crescpl.txt e o atribui ao objeto nls1
> nls1  <- read.table("C:/daniel/Cursos/RCursoTeX/crescpl.txt",
```

```
+               header=TRUE)
> m1.nlsfit <- nls(y~a*b^x,data=nls1,
+                  start=list(a=0.5,b=1.8), trace=T)

4.646657 :  0.5 1.8
0.1700421 :  0.808718 2.065252
0.002830186 :  0.8105998 1.9577810
0.002761698 :  0.8116737 1.9541516
0.002761698 :  0.8116704 1.9541598

> summary(m1.nlsfit)

Formula: y ~ a * b^x

Parameters:
  Estimate Std. Error t value Pr(>|t|)
a  0.81167    0.00873   92.98 1.04e-10 ***
b  1.95416    0.01371  142.52 8.05e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02145 on 6 degrees of freedom

Number of iterations to convergence: 4
Achieved convergence tolerance: 5.447e-07
```

Nesse programa a e b representam os parâmetros α e β, respectivamente; a função *nls* apresenta os seguintes argumentos, nessa ordem:

- *formula* $y \sim a*x\hat{}b$: esse é o modelo que queremos ajustar, com parâmetros a e b; a fórmula é uma expressão algébrica do lado direito da variável resposta, contendo parâmetros e variáveis regressoras. Os operadores possuem suas usuais funções aritméticas;

- *data*: especifica o *data frame* para a análise. É opcional;

- *start*: uma lista ou vetor numérico que tem a função de especificar os valores iniciais dos parâmetros. Os nomes dos componentes indicam ao R quais *variáveis* do lado direito do modelo são parâmetros e devem ter a mesma sintaxe; todas as outras variáveis são assumidas variáveis regressoras;

- *trace*: um argumento booleano que informa se os resultados de cada passo iterativo deve ou não ser impresso. Por padrão, nenhum resultado do processo iterativo é impresso;

- *algorithm*: deve receber uma *string*, que indica ao R que método deve ser usado. As *strings* são: "Gauss-Newton", "plinear" e "port". A opção *plinear* que indica ao R para utilizar o algoritmo de *Golub-Pereyra* de quadrados mínimos parciais ou a opção *port* para o algoritmo *nl2sol* do pacote *Port*. Quando omitida, a função *nls* utiliza a opção padrão, que é o algoritmo *Gauss-Newton*.

Vamos, nesse mesmo exemplo ilustrar como devemos incorporar as derivadas parciais no ajuste do modelo não-linear. Devemos criar uma função, cujo objetivo é associar ao modelo a matriz das derivadas parciais de primeira ordem, o Jacobiano, que o r denomina de gradiente. Vamos denominar essa função de *der.model*. Em seguida chamamos a função *nls*, com o argumento do lado direito da formula associado a função. O restante é bem similar. O programa resultante dessas operações está apresentado a seguir.

```
> # demonstrar como ajustar um modelo não-linear no R
> # utilizando a função nls e incorporando as derivadas parciais
> der.model <- function(a,b,x)
+ {
+    model.func <- a*b^x # função do modelo
+    J <- cbind(b^x,a*x*b^(x-1)) # Jacobiana
+    dimnames(J) <- list(NULL,c("a","b"))
+    attr(model.func, "gradient") <- J # atribui ao modelo o J
+    model.func # retorna a função + J
+ }
> m1.nlsfitder <- nls(y~der.model(a,b,x),data=nls1,
+                  start=list(a=0.5,b=1.8), trace=T)

4.646657 :  0.5 1.8
0.1700421 :  0.808718 2.065252
0.002830186 :  0.8105998 1.9577810
0.002761698 :  0.8116737 1.9541516
0.002761698 :  0.8116704 1.9541598

> summary(m1.nlsfitder)
```

```
Formula: y ~ der.model(a, b, x)

Parameters:
  Estimate Std. Error t value Pr(>|t|)
a  0.81167    0.00873   92.98 1.04e-10 ***
b  1.95416    0.01371  142.52 8.05e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02145 on 6 degrees of freedom

Number of iterations to convergence: 4
Achieved convergence tolerance: 5.505e-07
```

Neste exemplo, não houve diferença na velocidade e nem na convergência, comparando-se os ajustes sem e com a informação do gradiente ou Jacobiano, ou seja, das derivadas parciais. Existem situações, que a incorporação da informação sobre as derivadas parciais traz benefícios muito grandes a velocidade e até mesmo possibilita que a convergência seja alcançada em situações em que não se obteve ajuste do modelo, sem a informação das derivadas. a matriz de covariâncias das estimativas dos parâmetros pode ser obtida com a função *vcov* e a *deviance* com a função deviance do pacote *MASS*.

```
> # demonstrar como obter informações de um modelo não-linear no R
> deviance(m1.nlsfitder) # deviance

[1] 0.002761698

> vcov(m1.nlsfitder)  # matriz de covariâncias

              a             b
a  7.620655e-05 -0.0001087740
b -1.087740e-04  0.0001879969

> # novo ajuste com trace=FALSE
> # evitar confint de repercutir na tela grandes quantidades de valores
> # do que é chamado perfis
> m1.nlsfitder <- nls(y~der.model(a,b,x),data=nls1,start=list(a=0.5,b=1.8),
+                  control=list(maxiter = 500), trace=F)
> confint(m1.nlsfitder, level = 0.95) # Intervalo de confiança
```

```
       2.5%     97.5%
a 0.7904266 0.8331078
b 1.9210421 1.9880805

> # isso que se vê, antes dos IC's são os perfis
> fitted(m1.nlsfitder) # valores preditos

[1] 0.8679121 0.9280508 0.9923565 1.1346443 1.3872278
[6] 1.5861337 2.2172764 3.0995587
attr(,"label")
[1] "Fitted values"

> # R2
> # soma de quadrados residual
> SQE <- summary(m1.nlsfitder)$sigma^2*summary(m1.nlsfitder)$df[2]
> # soma de quadrado total corrigida
> SQT <- var(nls1$y)*(length(nls1$y)-1)
> R2 <- 1 - SQE/SQT
> R2

[1] 0.9993452
```

O R utilizou 4 iterações e apresentou uma mensagem que o ajuste do modelo atingiu convergência. O modelo ajustado foi $\hat{y}_i = 0{,}8117 \times 1{,}9542^{x_i}$. Ambos os parâmetros foram significativamente diferentes de zero, pois os intervalos assintóticos de 95% de confiança não abrangeram o valor 0, ou o teste t apresentou resultado significativo para os testes das hipóteses $H_0 : \alpha = 0$ e $H_0 : \beta = 0$. O intervalo assintótico de 95% confiança para o parâmetro α foi $[0{,}7904; 0{,}8331]$ e para o parâmetro β, $[1{,}9210; 1{,}9881]$. O R^2 do modelo pode ser estimado por $R^2 = 1 - SQRes/SQTotal$. Para este exemplo, o $R^2 = 1 - 0{,}00276/4{,}2178 = 0{,}9993$, indicando que 99,93% da variação do crescimento das plantas foi explicado pelo modelo de regressão.

Vamos ilustrar a função *nls* para o ajuste de mais um modelo aos dados da Tabela 3.2, dado por:

$$y_i = \alpha x_i^{\beta} + \epsilon_i \tag{4.2.2}$$

As derivadas parciais em relação a cada parâmetro são dadas pelas funções $\partial y_i/\partial\alpha = x_i^{\beta}$ e $\partial y_i/\partial\beta = \alpha x_i^{\beta} \ln(x_i)$. O programa correspondente a este exemplo é dado por:

```
> # demonstrar como ajustar um segundo modelo não-linear no R
> # utilizando a função nls
> m2.nlsfit <- nls(y~a*x^b,data=nls1,start=list(a=0.5,b=1.8),
+                  control=list(maxiter = 500), trace=F)
> summary(m2.nlsfit)

Formula: y ~ a * x^b

Parameters:
  Estimate Std. Error t value Pr(>|t|)
a  1.85476    0.10888  17.034 2.62e-06 ***
b  0.57496    0.09072   6.338 0.000722 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2703 on 6 degrees of freedom

Number of iterations to convergence: 8
Achieved convergence tolerance: 7.872e-06

> confint(m2.nlsfit, level = 0.95) # Intervalo de confiança

      2.5%     97.5%
a 1.582807 2.1212843
b 0.353482 0.8609511

> # isso que se vê, antes dos IC's são os perfis
> fitted(m2.nlsfit) # valores preditos

[1] 0.4935397 0.7351969 0.9282179 1.2451024 1.6314239
[6] 1.8547554 2.3417087 2.7629193
attr(,"label")
[1] "Fitted values"

> # R2
> # soma de quadrados residual
> SQE <- summary(m2.nlsfit)$sigma^2*summary(m2.nlsfit)$df[2]
> R2 <- 1 - SQE/SQT
> R2

[1] 0.8960709
```

Especificamos um número máximo de iterações igual a 500. O padrão do R, se nada for especificado, é 50. Neste caso ocorreu convergência

com apenas 8 iterações. Este comando (*maxiter=nit*) se torna útil apenas quando o valor inicial é precário, requerendo um número grande de iterações, principalmente se houver correlações elevadas entre os estimadores dos parâmetros. Neste exemplo, o modelo ajustado foi $\hat{y}_i = 1{,}8548x_i^{0{,}575}$, sendo que este ajuste foi um pouco inferior ao ajuste do modelo anterior. Isso pode ser constatado comparando o valor do coeficiente de determinação $R^2 = 89{,}61\%$ desse com o valor anteriormente obtido. Os dois modelos ajustados estão apresentados na Figura 4.1. Os pontos amostrais foram plotados juntamente com os modelos ajustados. Verificamos claramente que o primeiro modelo se ajustou melhor aos dados, conforme indicação dos valores de R^2 obtidos para cada um deles. Devemos procurar sempre, além de um bom ajuste, modelos que possam ter uma relação com o fenômeno que estamos estudando. Apesar dos bons ajustes alcançados, podemos para este exemplo escolher, do ponto de vista biológico, melhores modelos não-lineares.

```
> fx1 <- function(x) 0.8117*1.9542^x
> fx2 <- function(x) 1.8548*x^0.575
> xx <- seq(min(nls1$x),max(nls1$x),by=0.01)
> matplot(xx,cbind(fx1(xx),fx2(xx)),type="l",xlab="x",ylab="f(x)")
> points(nls1$x,nls1$y)
```

4.3 Modelos Segmentados

Entre os modelos segmentados, existe o modelo de "*response plateau*" que é muito utilizado na pesquisa em diversas áreas. Esse modelo possui dois segmentos, sendo que o primeiro descreve uma curva crescente ou decrescente até uma determinada altura da ordenada (P) que é o platô. A partir desse ponto o valor Y assume um valor constante P. O ponto correspondente ao valor P na abscissa é o ponto X_0, que também é um parâmetro a ser estimado. Vários modelos podem ser utilizados para modelar o comportamento da curva entre a origem e o ponto onde se encontra o platô. Nesta seção apresentamos o exemplo do manual do SAS (*proc iml*) com um modelo quadrático anterior ao platô, para podermos ajustá-lo no R. Na Figura 4.2 é apresentado um exemplo de um modelo de *response plateau*, destacando-se os pontos X_0 e P.

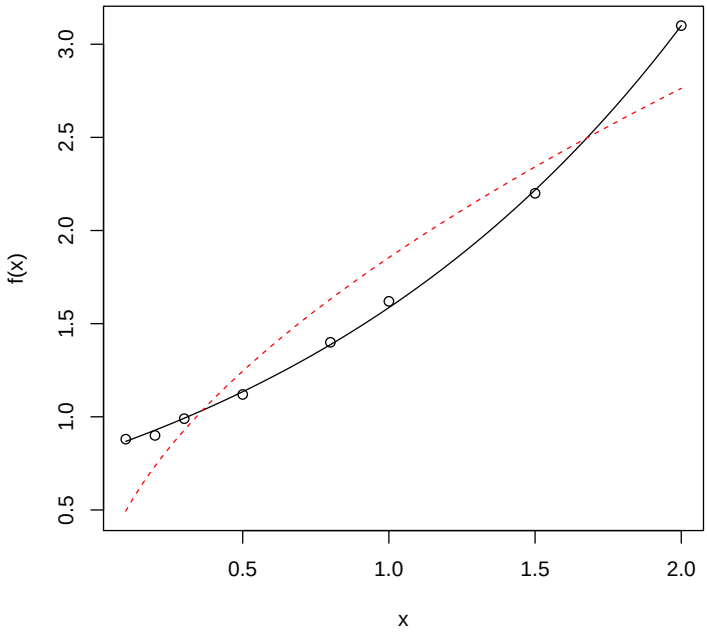


Figura 4.1. Modelos não lineares ajustados - modelo $\hat{y}_i = 1{,}8548x_i^{0{,}575}$ em vermelho e modelo $\hat{y}_i = 0{,}8117 \times 1{,}9542^{x_i}$ em preto.

Para ilustrarmos o ajuste de um modelo bi-segmentado dessa natureza é considerado o exemplo mencionado anteriormente e a função *nls*. Seja para isso o seguinte modelo quadrático de platô de resposta (*quadratic response plateau*):

$$Y_i = \begin{cases} \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \epsilon_i & \text{se } X_i \leq X_0 \\ P + \epsilon_i & \text{se } X_i > X_0. \end{cases} \tag{4.3.1}$$

Para valores de $X \leq X_0$, Y é explicado por um modelo quadrático (parábola) e para valores de $X > X_0$, a equação explicativa é constante e paralela a abscissa. O ponto X_0 é considerado desconhecido e deve ser estimado juntamente com os demais parâmetros do modelo. Este ponto representa a junção do segmento quadrático com o segmento de platô. As curvas devem ser contínuas (os dois segmentos devem se encontrar em X_0) e suavizada, ou seja, as primeiras derivadas com relação a X nos dois seg-

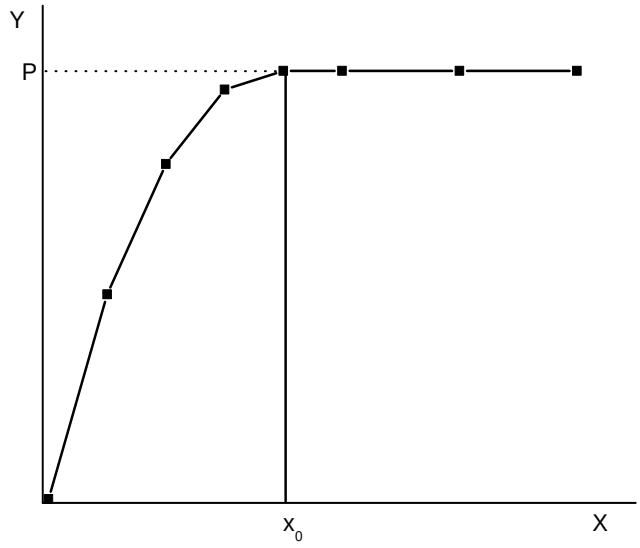


Figura 4.2. Modelo segmentado considerando um plateau no ponto $X = X_0$ com valor de $Y = P$ e um modelo crescente para $X < X_0$.

mentos devem ser a mesma no ponto X_0. Essas condições implicam em algumas conseqüências descritas a seguir.

A primeira derivada de Y em relação a X no modelo quadrático é dada por:

$$\frac{dY_i}{dX_i} = \beta_1 + 2\beta_2 X_i$$

Se igualarmos esta derivada a zero, resolvermos a equação resultante em X e substituirmos o valor de X por X_0, ponto em que a curva deve ser contínua e suavizada, obtemos:

$$X_0 = \frac{-\beta_1}{2\beta_2}.$$

Substituindo esse valor na equação (4.3.1) obtemos o máximo, que corresponde ao platô almejado. Assim, este platô é dado por:

$$Y = P = \beta_0 + \beta_1 X_0 + \beta_2 X_0^2 = \beta_0 - \frac{\beta_1^2}{2\beta_2} + \frac{\beta_1^2 \beta_2}{4\beta_2^2} = \beta_0 - \frac{\beta_1^2}{4\beta_2}.$$

Nesse caso temos apenas três parâmetros efetivos, pois tanto X_0, quanto P são determinados a partir de β_0, β_1 e β_2. Este é um modelo não-linear nos parâmetros, pois as derivadas parciais de Y são funções dos parâmetros em alguns casos, justificando o uso da *nls*. O programa final é apresentado na seqüência. Podemos destacar que o modelo é dividido em duas partes: a primeira com a parte quadrática polinomial e a segunda, com a parte do platô. Em cada ciclo do processo iterativo não imprimimos os resultados, pois escolhemos *trace=FALSE*. Com isso na obtenção dos intervalos de confiança, não teremos os perfis impressos juntamente na saída dos resultados almejados, uma vez que isso é desnecessário e ocupa muito espaço. Utilizamos o *plot* para produzir um gráfico dos valores ajustados e observados. Nesse modelo, $b0$ representa β_0, $b1$ representa β_1 e $b2$ representa β_2.

```
> # função para obter o ajuste do modelo quadrático com platô de
> # resposta - exemplo do manual do SAS
> # lê o arquivo QRP.txt e o atribui ao objeto QRP
> QRP  <- read.table("C:/daniel/Cursos/RCursoTeX/QRP.txt",
+                header=TRUE)
> QRP # imprime o data frame

    x    y
1   1 0.46
2   2 0.47
3   3 0.57
4   4 0.61
5   5 0.62
6   6 0.68
7   7 0.69
8   8 0.78
9   9 0.70
10 10 0.74
11 11 0.77
12 12 0.78
13 13 0.74
14 13 0.80
15 15 0.80
16 16 0.78
```

```
> x <- QRP$x
> y <- QRP$y
> qrp.fit <- nls(y ~ (b0 + b1*x + b2*I(x^2))*
+                  (x <= -0.5*b1/b2)
+                 +(b0 +I(-b1^2/(4*b2)))*
+                 (x > -0.5*b1/b2),
+                 data=QRP,
+                 start=list(b0=0.45, b1=0.05, b2=-0.0025),
+                 trace=F)
> qrp.coef <- coef(qrp.fit)
> X0 <- -0.5*qrp.coef[2]/qrp.coef[3]
> X0 # Ponto da abscissa referente ao platô

      b1
12.74767

> P <- qrp.coef[1] - qrp.coef[2]^2/(4*qrp.coef[3])
> P # platô

       b0
0.7774974

> summary(qrp.fit)

Formula: y ~ (b0 + b1 * x + b2 * I(x^2)) * (x <= -0.5 * b1/b2) + (b0 +
    I(-b1^2/(4 * b2))) * (x > -0.5 * b1/b2)

Parameters:
     Estimate Std. Error t value Pr(>|t|)
b0  0.3921154  0.0266741  14.700 1.77e-09 ***
b1  0.0604631  0.0084230   7.178 7.17e-06 ***
b2 -0.0023715  0.0005513  -4.302 0.000861 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.02783 on 13 degrees of freedom

Number of iterations to convergence: 6
Achieved convergence tolerance: 1.574e-06

> confint(qrp.fit, level = 0.95) # Intervalo de confiança

           2.5%        97.5%
b0  0.329434068  0.450282145
b1  0.042982216  0.083546179
b2 -0.004086228 -0.001296211

> # isso que se vê, antes dos IC's são os perfis
> fitted(qrp.fit) # valores preditos
```

```
 [1] 0.4502070 0.5035555 0.5521610 0.5960233 0.6351426
 [6] 0.6695189 0.6991520 0.7240421 0.7441891 0.7595931
[11] 0.7702539 0.7761717 0.7774974 0.7774974 0.7774974
[16] 0.7774974
attr(,"label")
[1] "Fitted values"

> # R2
> # soma de quadrados residual
> SQE <- summary(qrp.fit)$sigma^2*summary(qrp.fit)$df[2]
> SQE

[1] 0.01006599

> SQT <- var(QRP$y)*(length(QRP$y)-1)
> # R2
> R2 <- 1 - SQE/SQT
> R2

[1] 0.946155
```

O modelo ajustado foi $\hat{Y}_i = 0{,}3921 + 0{,}0605X_i - 0{,}00237X_i^2$ se $X_i < 12{,}7477$ e $\hat{Y}_i = 0{,}7775$, caso contrário. As estimativas de β_0 e β_1 foram significativamente ($P < 0{,}05$) superiores a zero e a de β_2, significativamente inferior a zero. Estes resultados foram obtidos analisando os intervalos de confiança assintóticos ou os testes t assintóticos. O R^2 do modelo foi igual a $1 - 0{,}0101/0{,}1869 = 0{,}9460$. É conveniente atentarmos para o fato de que o R utiliza um procedimento considerado mais eficiente para a obtenção dos intervalos de confiança para os parâmetros do modelo, que o método assintótico baseado na distribuição t de Student, frequentemente apresentadas nos programas de análise estatística. Esses resultados, embora mais eficientes, são muito próximos dos resultados assintóticos.

Na Figura 4.3 apresentamos o gráfico correspondente a esse modelo ajustado, juntamente com o resultado do programa R para obtê-lo. Apresentamos o programa sem os resultados separadamente da figura. O programa é dado por:

```
> f <- function(x,x0)
+ {
+     cond <- matrix(TRUE,length(x),1)
+     cond[x>x0] <- FALSE
+     y <- matrix(0,length(x),1)
+     y[cond] <- 0.3921 + 0.0605*x[cond] - 0.00237*x[cond]^2
```

```
+    y[!cond] <-  0.3921 + 0.0605*x0 - 0.00237*x0^2
+    return(y)
+ }
> x0 <- 12.7477
> p <- 0.3921 + 0.0605*x0 - 0.00237*x0^2
> x <- seq(1,16.3,by=0.1)
> y<-f(x,x0)
> plot(x,y,type="l",xlab="x",ylab="f(x)",ylim=c(0.449,0.801))
> segments(x0, 0, x0, p)
> segments(0, p, x0, p)
> points(QRP$x,QRP$y,pch=19)
```

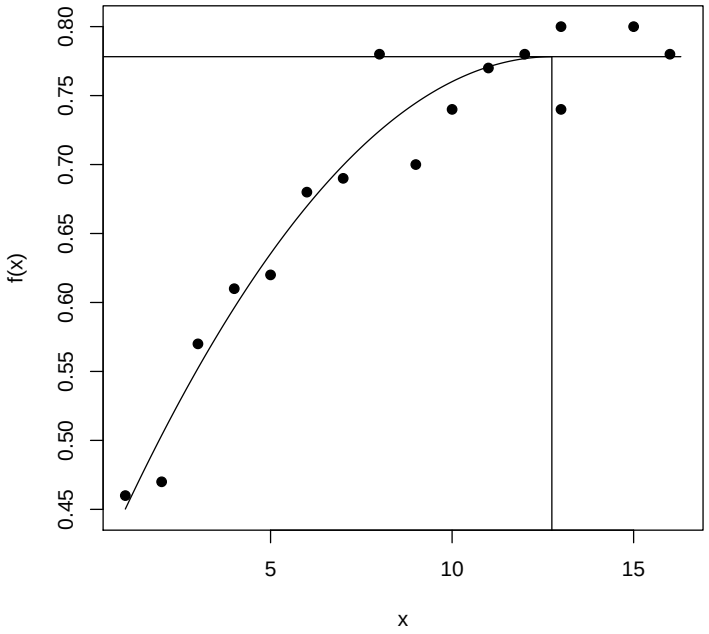


Figura 4.3. Modelo platô de resposta ajustado $\hat{Y}_i = 0{,}3921 + 0{,}0605X_i - 0{,}00237X_i^2$ se $X_i < 12{,}7477$ e $\hat{Y}_i = 0{,}7775$, caso contrário.

Outro modelo segmentado que aparece freqüentemente na literatura é o *linear response plateau* ou LRP. Este modelo possui um segmento de reta, crescente ou decrescente, antes do ponto de junção (X_0) com o platô. O modelo LRP é dado por:

$$Y_i = \begin{cases} \beta_0 + \beta_1 X_i + \epsilon_i & \text{se } X_i \leq X_0 \\ P + \epsilon_i & \text{se } X_i > X_0. \end{cases} \tag{4.3.2}$$

É comum utilizarmos uma variável binária (*Dummy*) para representarmos o modelo. Neste caso utilizaremos a variável Z_i, que receberá o valor 1 se $X_i \leq X_0$, ou 0 se $X_i > X_0$. Este modelo poderá ser reescrito por $Y_i = (\beta_0 + \beta_1 X_i)\, Z_i + P(1 - Z_i)$. Para termos continuidade em X_0, devemos igualar $\beta_0 + \beta_1 X_0 = P$, ou seja, $X_0 = (P - \beta_0)/\beta_1$.

Neste caso temos um modelo com três parâmetros (β_0, β_1 e P). Diferentemente do modelo anterior, P não pôde ser expresso em função dos demais parâmetros considerando o LRP. Apesar de as variáveis parciais não dependerem dos parâmetros, este é um modelo não-linear uma vez que a matriz Jacobiana depende de X_0 para ser construída, sendo que X_0 é função de β_0, β_1 e de P. Assim, as derivadas parciais, dadas por $\partial Y_i/\partial\beta_0 = Z_i$, $\partial Y_i/\partial\beta_1 = X_i Z_i$ e $\partial Y_i/\partial P = 1 - Z_i$, dependem dos parâmetros por meio de X_0. A cada passo do processo iterativo, o parâmetro X_0 é estimado e a matriz do modelo é composta, pois os Z_i's ficam completamente definidos.

Utilizamos duas abordagens diferentes. Na primeira construímos uma função para realizarmos o processo de estimação, considerando as características e propriedades do modelo descritos anteriormente. Para isso, como, dado um valor de X_0, as derivadas parciais não dependem dos parâmetros, utilizamos a função *lm* para estimar os parâmetros. Com as novas estimativas, re-estimamos X_0 e repetimos o processo. As iterações são repetidas até que as estimativas de um dado ponto sejam praticamente iguais, considerando um determinado critério numérico, as estimativas do passo anterior. Utilizamos por padrão 1×10^{-8}. Este tipo de procedimento foi colocado aqui, considerando o importante aspecto didático que está por trás de todo esse processo. Na segunda abordagem, utilizamos os recursos da função *nls* para estimar os parâmetros do modelo segmentado LRP. A segunda grande vantagem da primeira abordagem, refere-se ao fato de que a função *nls*, para esse modelo, é muito sensível aos valores arbitrários iniciais utilizados como argumento da função. A vantagem de utilizarmos os dois procedimentos simultaneamente refere-se ao fato de que no primeiro caso P foi considerado um parâmetro e no segundo, X_0. Assim, os intervalos de confiança para todos os parâmetros podem ser obtidos de ambos os casos.

Como a função que programamos é mais estável e menos sensível às escolhas dos valores iniciais, podemos utilizar o seu resultado final como valores iniciais da função *nls*. Para isso, devemos observar que $P = \beta_0 + \beta_1 X_0$.

O resultado final dos códigos necessários ao ajuste do LRP está apresentado na seqüência para um conjunto simulado de dados. Neste conjunto de dados os parâmetros são $\beta_0 = 2$, $\beta_1 = 2$ e $P = 10$, o que corresponde a um valor de X_0 de 4.

```
> # função para obter o ajuste do modelo linear de platô de resposta
> LRP.fit <- function(x,y,a=1,b=2,p=2,precis = 1e-8,maxit = 500)
+ {
+    montax <- function(n,X0)
+    {
+       x1 <- matrix(1,n,1)
+       x2 <- x
+       x3 <- matrix(0,n,1)
+       x1[x>X0] <- 0
+       x2[x>X0] <- 0
+       x3[x>X0] <- 1
+       X <- cbind(x1,x2,x3)
+       return(X)
+    }
+    iterage <- function(a,b,p)
+    {
+       X0 <- (p-a)/b
+       n <- length(x)
+       X <- montax(n,X0)
+       reg <- lm(y~X[,1]+X[,2]+X[,3]-1)
+       an <- reg$coefficients[1]
+       bn <- reg$coefficients[2]
+       pn <- reg$coefficients[3]
+       return(list(a=an,b=bn,p=pn))
+    }
+    cont <- 1
+    repeat
+    {
+       coefn <- iterage(a,b,p)
+       if (is.na(coefn$a)) aa <- 0 else aa <- coefn$a
+       if (is.na(coefn$b)) bb <- 0 else bb <- coefn$b
+       if (is.na(coefn$p)) pp <- 0 else pp <- coefn$p
+       diff <- max(abs(a-aa),abs(b-bb),abs(p-pp))
+       if (diff < precis)
+       {
+          if (!is.na(coefn$a)) a <- aa else a <- a - 0.5*a
```

```
+          if (!is.na(coefn$b)) b <- bb else b <- b - 0.5*b
+          if (!is.na(coefn$p)) p <- pp else p <- p + 0.5*p
+          break
+       } else
+       {
+         if (!is.na(coefn$a)) a <- aa else a <- a - 0.5*a
+         if (!is.na(coefn$b)) b <- bb else b <- b - 0.5*b
+         if (!is.na(coefn$p)) p <- pp else p <- p + 0.5*p
+         cont <- cont + 1
+       }
+       if (cont > maxit) break
+    }
+    if (cont > maxit)
+    {
+      mess <- "Convergence criterion failed"
+      X0 <- (p-a)/b
+      n <- length(x)
+      X <- montax(n,X0)
+      reg <- lm(y~X[,1]+X[,2]+X[,3]-1)
+    }  else
+    {
+       mess <- "Convergence criterion met!"
+       X0 <- (p-a)/b
+       n <- length(x)
+       X <- montax(n,X0)
+       reg <- lm(y~X[,1]+X[,2]+X[,3]-1)
+    }
+    return(list(reg=reg,X0=X0,iter=cont,message=mess))
+ }
> # lê o arquivo LRP1.txt e o atribui ao objeto LRP1
> RLP1  <- read.table("C:/daniel/Cursos/RCursoTeX/LRP1.txt",
+               header=TRUE)
> RLP1 # conjunto de dados utilizado


    x     y
1 1.0  4.10
2 2.0  5.90
3 2.5  7.10
4 3.0  7.80
5 4.0  9.90
6 5.0 10.10
7 6.0 10.20
8 7.0  9.80
9 8.0  9.78
```

```
> x <- RLP1$x
> y <- RLP1$y
> LRPR <- LRP.fit(x,y,a=10,b=200,p=20,precis = 1e-8,maxit = 500)
> LRPR

$reg

Call:
lm(formula = y ~ X[, 1] + X[, 2] + X[, 3] - 1)

Coefficients:
X[, 1]  X[, 2]  X[, 3]
 2.135   1.930   9.970


$X0
  X[, 3]
4.059585

$iter
[1] 10

$message
[1] "Convergence criterion met!"

> n <- length(y)
> summary(LRPR$reg)

Call:
lm(formula = y ~ X[, 1] + X[, 2] + X[, 3] - 1)

Residuals:
   Min     1Q Median     3Q    Max
-0.190 -0.125  0.035  0.130  0.230

Coefficients:
       Estimate Std. Error t value Pr(>|t|)
X[, 1]  2.13500    0.20989   10.17 5.26e-05 ***
X[, 2]  1.93000    0.07795   24.76 2.86e-07 ***
X[, 3]  9.97000    0.08715  114.39 3.01e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1743 on 6 degrees of freedom
Multiple R-squared: 0.9997,        Adjusted R-squared: 0.9996
F-statistic:  7224 on 3 and 6 DF,  p-value: 4.638e-11
```

```
> anava <- anova(LRPR$reg)
> anava

Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq  F value    Pr(>F)
X[, 1]     1 242.21  242.21  7971.74 1.330e-10 ***
X[, 2]     1  18.62   18.62   612.98 2.857e-07 ***
X[, 3]     1 397.60  397.60 13086.24 3.008e-11 ***
Residuals  6   0.18    0.03
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> confint(LRPR$reg) # intervalo de confiança

          2.5 %    97.5 %
X[, 1] 1.621406  2.648594
X[, 2] 1.739256  2.120744
X[, 3] 9.756742 10.183258

> R2 <- 1- anava$"Sum Sq"[4]/(var(y)*(n-1))
> # Valor correto de R2
> R2

[1] 0.9953185

> # usando a função nls - muito sensível a valores iniciais
> x <- RLP1$x
> y <- RLP1$y
> lrp.fit<- nls(y~(b0 + b1*x)*(x<=x0)
+                  +(b0+b1*x0)*(x>x0),
+                  data=RLP1,
+                  start=list(b0=2, b1=2, x0=4),
+                  trace=F)
> lrp.fit

Nonlinear regression model
  model:  y ~ (b0 + b1 * x) * (x <= x0) + (b0 + b1 * x0) * (x > x0)
   data:  RLP1
   b0    b1    x0
2.135 1.930 4.060
 residual sum-of-squares: 0.1823

Number of iterations to convergence: 2
Achieved convergence tolerance: 1.828e-08

> summary(lrp.fit)
```

```
Formula: y ~ (b0 + b1 * x) * (x <= x0) + (b0 + b1 * x0) * (x > x0)

Parameters:
   Estimate Std. Error t value Pr(>|t|)
b0  2.13500    0.20989   10.17 5.26e-05 ***
b1  1.93000    0.07795   24.76 2.86e-07 ***
x0  4.05959    0.08740   46.45 6.67e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1743 on 6 degrees of freedom

Number of iterations to convergence: 2
Achieved convergence tolerance: 1.828e-08
```

```
> confint(lrp.fit) # IC para os parâmetros

       2.5%    97.5%
b0 1.553167 2.648596
b1 1.739251 2.181292
x0 3.799484 4.290454
```

O modelo ajustado foi $\hat{Y}_i = 2{,}135 + 1{,}93X_i$ se $X_i \leq 4{,}06$ e $\hat{Y}_i = 9{,}97$, se $X_i > 4{,}06$. O coeficiente de determinação do modelo foi igual a $R^2 = 99{,}53\%$. Todos os valores paramétricos estão dentro do intervalo de confiança assintótico construído, conforme já era esperado.

```
> f.lrp <- function(x,x0)
+ {
+    cond <- matrix(TRUE,length(x),1)
+    cond[x>x0] <- FALSE
+    y <- matrix(0,length(x),1)
+    y[cond] <- 2.135+1.93*x[cond]
+    y[!cond] <-  9.97
+    return(y)
+ }
> x0 <- 4.06
> p <- 9.97
> x <- seq(1,8,by=0.1)
> y<-f.lrp(x,x0)
> plot(x,y,type="l",xlab="x",ylab="f(x)",ylim=c(4,10.21))
> segments(x0, 0, x0, p)
> segments(0, p, x0, p)
> points(RLP1$x,RLP1$y,pch=19)
```

Na Figura **??** apresentamos o modelo ajustado e os pontos observados da variável resposta, que foram simulados nesse exemplo.

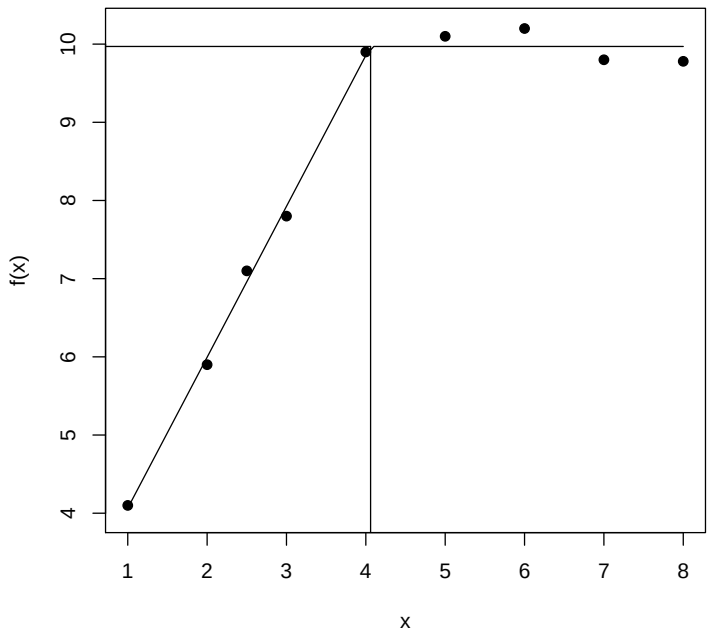


Figura 4.4. Modelo platô de resposta linear ajustado: $\hat{Y}_i = 2{,}135 + 1{,}93X_i$ se $X_i \leq 4{,}06$ e $\hat{Y}_i = 9{,}97$ se $X_i > 4{,}06$.

Apresentamos na seqüência um outro exemplo, também simulado, em que temos os parâmetros iguais a $\beta_0 = 5$, $\beta_1 = 2{,}4$, $P = 29$ e $\sigma^2 = 1$.

```
> # função para obter o ajuste do modelo linear de platô de resposta
> # mesma do exemplo anterior
>
> # lê o arquivo LRP2.txt e o atribui ao objeto LRP2
> RLP2  <- read.table("C:/daniel/Cursos/RCursoTeX/LRP2.txt",
+               header=TRUE)
> RLP2 # conjunto de dados utilizado

   x         y
1  1  8.626484
2  2  8.940873
3  3 11.909886
4  4 13.936262
5  5 17.945067
```

```
6   6 18.732450
7   7 21.847226
8   8 23.769043
9   9 27.671300
10 10 28.441954
11 11 27.811677
12 12 30.827451
13 13 28.817408
14 14 30.665168
15 15 28.813364
16 16 29.127870
17 17 28.218656
18 18 28.309338
19 19 28.651342
20 20 29.230743
```

```
> x <- RLP2$x
> y <- RLP2$y
> LRPR <- LRP.fit(x,y,a=10,b=20,p=20,precis = 1e-8,maxit = 500)
> LRPR
```

```
$reg

Call:
lm(formula = y ~ X[, 1] + X[, 2] + X[, 3] - 1)

Coefficients:
X[, 1]  X[, 2]  X[, 3]
 5.073   2.383  29.047


$X0
  X[, 3]
10.05863

$iter
[1] 6

$message
[1] "Convergence criterion met!"
```

```
> n <- length(y)
> summary(LRPR$reg)
```

```
Call:
lm(formula = y ~ X[, 1] + X[, 2] + X[, 3] - 1)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-1.2356 -0.6486 -0.2737  0.3763  1.7801

Coefficients:
       Estimate Std. Error t value Pr(>|t|)
X[, 1]   5.0731     0.6326    8.02 3.53e-07 ***
X[, 2]   2.3834     0.1019   23.38 2.30e-14 ***
X[, 3]  29.0473     0.2928   99.20  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.926 on 17 degrees of freedom
Multiple R-squared: 0.9988,        Adjusted R-squared: 0.9986
F-statistic:  4748 on 3 and 17 DF,  p-value: < 2.2e-16
```

```
> anava <- anova(LRPR$reg)
> anava
```

```
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value  Pr(>F)
X[, 1]     1 3305.9  3305.9 3855.64 < 2e-16 ***
X[, 2]     1  468.7   468.7  546.61 2.3e-14 ***
X[, 3]     1 8437.5  8437.5 9840.61 < 2e-16 ***
Residuals 17   14.6     0.9
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> confint(LRPR$reg) # intervalo de confiança
```

```
           2.5 %    97.5 %
X[, 1]  3.738538  6.407686
X[, 2]  2.168358  2.598530
X[, 3] 28.429514 29.665090
```

```
> R2 <- 1- anava$"Sum Sq"[4]/(var(y)*(n-1))
> # Valor correto de R2
> R2
```

```
[1] 0.9864221
```

```
> # usando a função nls - muito sensível a valores iniciais
> x <- RLP2$x
> y <- RLP2$y
> lrp.fit<- nls(y~(b0 + b1*x)*(x<=x0)
+                  +(b0+b1*x0)*(x>x0),
```

```
+                  data=RLP2,
+                  start=list(b0=2, b1=2, x0=4),
+                  trace=F)
> lrp.fit

Nonlinear regression model
  model:  y ~ (b0 + b1 * x) * (x <= x0) + (b0 + b1 * x0) * (x > x0)
   data:  RLP2
   b0    b1    x0
4.931 2.422 9.934
 residual sum-of-squares: 14.58

Number of iterations to convergence: 4
Achieved convergence tolerance: 5.341e-09

> summary(lrp.fit)

Formula: y ~ (b0 + b1 * x) * (x <= x0) + (b0 + b1 * x0) * (x > x0)

Parameters:
   Estimate Std. Error t value Pr(>|t|)
b0   4.9308     0.6727   7.329 1.18e-06 ***
b1   2.4222     0.1195  20.261 2.42e-13 ***
x0   9.9335     0.2980  33.332  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.926 on 17 degrees of freedom

Number of iterations to convergence: 4
Achieved convergence tolerance: 5.341e-09

> confint(lrp.fit) # IC para os parâmetros

       2.5%     97.5%
b0 3.511614  6.408951
b1 2.168505  2.674441
x0 9.353236 10.662954
```

O modelo ajustado para este exemplo foi $\hat{Y}_i = 5{,}0731 + 2{,}3834X_i$ se $X_i \leq 10{,}06$ e $\hat{Y}_i = 29{,}05$, se $X_i > 10{,}06$. Convém salientar que os métodos de ajuste, nesse caso, levaram a estimativas um pouco diferentes entre si. Escolhemos as estimativas do primeiro procedimento para apresentarmos os demais resultados. Isso ocorre em decorrência de os métodos possuírem diferentes fundamentações teóricas. O coeficiente de determinação do modelo para o primeiro ajuste foi igual a $R^2 = 98{,}64\%$. Também neste

caso, todos os valores paramétricos estão dentro do intervalo de confiança assintótico construído. Na seqüência, apresentamos os comandos R para obtermos o gráfico em questão.

```
> f.lrp <- function(x,x0)
+ {
+     cond <- matrix(TRUE,length(x),1)
+     cond[x>x0] <- FALSE
+     y <- matrix(0,length(x),1)
+     y[cond] <- 5.0731+2.3834*x[cond]
+     y[!cond] <-  29.05
+     return(y)
+ }
> x0 <- 10.06
> p <-  29.05
> x <- seq(1,20,by=0.1)
> y<-f.lrp(x,x0)
> plot(x,y,type="l",xlab="x",ylab="f(x)",ylim=c(8.6,29.3))
> segments(x0, 0, x0, p)
> segments(0, p, x0, p)
> points(RLP2$x,RLP2$y,pch=19)
```

Na Figura 4.5 apresentamos o modelo ajustado e os pontos observados. Os valores observados estão distribuídos de forma muito próxima da função ajustada, indicando que o modelo é adequado.

4.4 Exercícios

1. Utilize os dados da Tabela 3.2 e a função *nls* do R para ajustar o seguinte modelo:
$$Y_i = \frac{\alpha}{\beta_0 + \beta_i X_i} + \epsilon_i$$

2. Este modelo se ajustou melhor do que aqueles da seção 4.2? Justifique sua resposta.

3. Tente ajustar um modelo LRP aos dados da Tabela 3.2. Qual foi o modelo encontrado? Este modelo é um modelo LRP? Justifique sua resposta. Plote os dados e verifique se existe uma dispersão dos pontos que justifique a representação por meio de um modelo LRP.

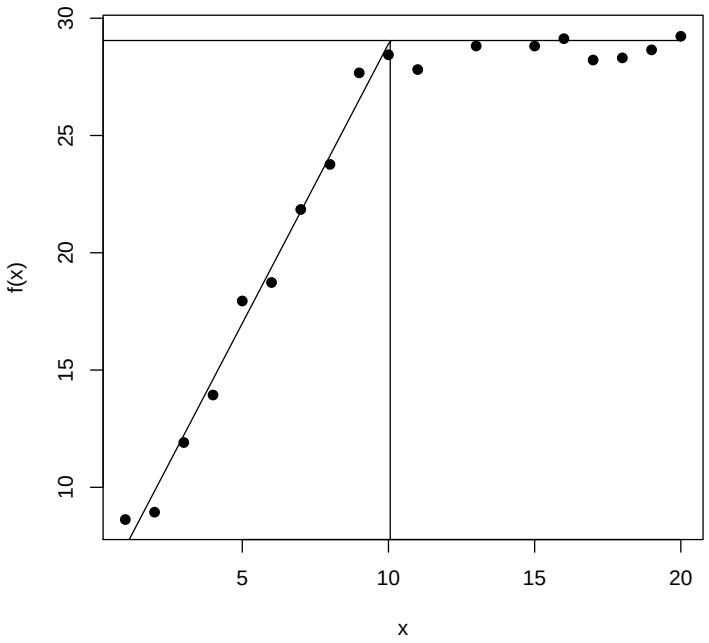


Figura 4.5. Modelo platô de resposta linear ajustado: $\hat{Y}_i = 5{,}0731 + 2{,}3834X_i$ se $X_i \leq 10{,}06$ e $\hat{Y}_i = 29{,}05$, se $X_i > 10{,}06$.

4. Utilize os resíduos gerados no exemplo apresentado em aula do ajuste do modelo LRP e realize a análise gráfica dos resíduos.

5. Busque em sua área de atuação dados que poderiam se enquadrar dentro dos modelos segmentados quadrático e linear. Descreva as situações e os possíveis benefícios de ajustar um modelo deste tipo. Se os dados estiverem disponíveis, utilize os programas apresentados em aula para ajustar o modelo de platô de resposta quadrático ou linear.

Capítulo 5

Análise de Variância para Dados Balanceados

Para realizarmos inferências sobre a hipótese de igualdade entre várias médias dos níveis de algum fator de interesse, utilizamos o teste F da análise de variância (Anava). Esta hipótese pode ser formalizada por:

$$\begin{cases} H_0 : \mu_1 = \mu_2 = \cdots = \mu_\ell = \mu \\ H_1 : \text{pelo menos uma média difere das demais} \end{cases} \tag{5.0.1}$$

em que ℓ é o número de níveis desse fator de interesse e μ_i é a média do i-ésimo nível, $i = 1, 2, \cdots, \ell$.

Um valor de F observado superior a um valor crítico da distribuição F para um nível α de significância indica que devemos rejeitar a hipótese nula H_0; caso contrário, não existirão evidências significativas para rejeitar a hipótese nula. Podemos ter mais de um fator. Nesse caso teremos uma hipótese nula para cada fator separadamente. Além disso, estes fatores podem interagir. Se houver algum tipo de interação entre eles, um teste F específico para a hipótese de haver interação irá apresentar efeito significativo da estatística. Também podemos ter efeitos hierarquizados, onde os níveis de um fator A, por exemplo, dentro de um determinado nível de outro fator, digamos B, são diferentes dos níveis de A em outro nível de B. Isto ocorre, por exemplo, quando temos diferentes procedências de eucalipto e dentro de cada procedência, temos diferentes progênies.

Nesse capítulo estaremos interessados nesses diferentes modelos estatísticos, contendo um ou mais fatores, cujos efeitos podem ser cruzados ou hierarquizados, porém em uma estrutura experimental balanceada. Entenderemos por estrutura balanceada, aquele conjunto de dados cujo número de observações em cada combinação dos níveis dos fatores é o mesmo. Cada nível de um fator, ou cada nível resultante da combinação dos níveis de dois ou mais fatores, é denominado de casela. Se houver diferenças nesse número de observações por casela, teremos dados não balanceados. A função R apropriada para lidar com essas estruturas é a *aov*. Se a estrutura é não-balanceada podemos utilizar a função *lm*.

5.1 A função aov

A função *aov* nos permite realizar análises de variância envolvendo dados balanceados. Vamos apresentar na seqüência alguns dos comandos básicos e específicos para ilustrar a sintaxe da *aov*. Podemos utilizar também a função *lm*, acompanhada da *aov* ou da *anova*. A diferença básica entre a função *aov* e a *lm* é a forma tradicional dos resultados da análise de variância que é apresentada pela *aov*, enquanto na *lm*, os resultados são apresentados em conformidade com aqueles da teoria dos modelos lineares. A função *aov* é na verdade um envelope da função *lm*, ou seja, a função em questão invoca a função *lm* em seus cálculos.

A função *aov*, que vamos utilizar preferencialmente nesse capítulo possui a seguinte sintaxe geral:

aov(formula, data = NULL, projections = FALSE, qr = TRUE, contrasts = NULL, ...)

Nesse comando, as opções são utilizadas para passar informações importantes para a função *aov*. Essas funções são: *formula* indica o modelo linear que utilizaremos, *data* indica o *data frame* que deve ser utilizado, *projections* indica por meio de uma opção *booleana* se as projeções devem ser retornadas na saída, *qr* indica também por meio de uma variável *booleana*, cujo padrão é verdadeiro, se a decomposição QR deve ser retornada e *contrasts* indica uma lista de contrastes para ser utilizada nos fatores do modelo, com exceção dos termos que são erros nesse modelo. A fórmula (modelo) pode ter mais de um erro ou múltiplas respostas. Podemos uti-

lizar as funções *print* e *summary* para reproduzir resultados dos objetos retornados pela função *aov* ou pela *lm*.

Uma importante observação refere-se ao fato de que os fatores de uma análise de variância devem ser interpretados pelo R como fatores e não como covariáveis. Não considerar isso é um erro comum no R. Se a variável não for considerada um fator, ela entrará no modelo como uma variável regressora ou covariável e com 1 grau de liberdade associado. O primeiro passo a ser dado antes de analisarmos um experimento é organizarmos o *data frame* apropriado. Podemos considerar que o *data frame* é uma matriz, cujas colunas são as variáveis e as linhas são as observações experimentais. Numa análise de variância as colunas devem ser consideradas fatores e não covariáveis, a não ser em alguns casos em que realmente alguma coluna deve ser considerada uma variável numérica. Nesses casos, estamos interessados em uma análise conhecida como análise de covariância. Essa análise pode ser considerada uma mistura de análise de variância com análise de regressão. Uma forma de explorar os dados são as funções *interaction.plot*, *plot.design* e *plot.factor*.

Vamos nesse capítulo ilustrar as principais maneiras de especificar o modelo, que é feito pela opção *formula*. Por intermédio dessa opção, temos inúmeras possibilidades diferentes para especificarmos o modelo que iremos ajustar. Devemos modelar a variável resposta em função das variáveis classificatórias, os fatores. Podemos especificá-las por meio de uma soma de termos correspondentes aos efeitos do nosso modelo. Obviamente, devemos usar os mesmos nomes especificados no cabeçalho de nosso *data frame*. Antes de utilizarmos a função *aov*, devemos certificarmos que nossas variáveis classificatórias serão interpretadas como fatores. Fazemos isso utilizando a função *as.factor*. Também podemos realizar análises de variância multivariada com a função *manova*. Podemos também ajustar modelos sem intercepto, bastando para iso utilizar o termo -1, na especificação do modelo.

Na medida que nossas necessidades forem aparecendo, iremos apresentar e ilustrar o uso de outras funções para realizarmos análises específicas e complementares à análise de variância. Uma análise complementar muito importante são os testes de comparações múltiplas.

Vários testes de comparações múltiplas são utilizados na comunidade científica: o teste de Bonferroni, o teste de Duncan, o teste de Dunnett

para comparar um tratamento controle com os demais, o teste *LSD* ou *t*, o teste *Scheffe*, o teste *SNK*, o teste *Tukey* e o teste *Waller-Duncan*.

Finalmente, podemos apresentar funções que nos possibilitarão aplicar o teste de homogeneidade de variâncias para os grupos de tratamentos, no modelo inteiramente casualizado. Os testes mais utilizados na literatura são: *Bartlett*, *Levene* utilizando os desvios da média absolutos ou quadráticos, *Brown* e *Forsythe* e de *O' Brien*. O teste de *Brown* e *Forsythe* é uma variação do teste de *Levene*, que utiliza desvios da mediana; o teste *O' Brien* é também uma variação do teste Levene . Ferreira (2005) descreve com detalhes estes testes.

Modelos com mais de um erro ou modelos , também serão considerados nesse material oportunamente. Análises conjuntas e análises individuais serão consideradas. Vamos ilustrar algumas formas que podemos utilizar para especificar o modelo de análise de variância. Suponhamos que A, B e C sejam fatores de interesse e Y a variável resposta. Quando lemos um aruivo texto, cujas colunas são numéricas, o R não considera as variáveis como fatores. Podemos ao especificar um efeito, digamos A, indicar ao R que se trata de um fator, utilizando para isso a sintaxe: $A <- as.factor(A)$. Podemos especificar diferentes modelos utilizando os seguintes comandos:

a) Exemplos de modelos com efeitos simples: *Y~A* ou *Y~A + B* ou *Y~A +B+C*.

b) Exemplos de efeitos cruzados: *Y~A+B+A*B* ou simplesmente *Y~A*B*. Nesse último caso, a $A * B$, sem os fatores principais, é uma notação geral para a estrutura de efeitos. No exemplo particular significa que o modelo ajustado é função dos efeitos principais e da interação, ou seja, é igual ao primeiro modelo desse item.

c) Exemplos de efeitos hierárquicos: *Y~B+ A:B*, indicando que temos um modelo com o fator principal B e com o fator A hierarquizado, dentro dos níveis de B. Isso significa que os níveis de A não são os mesmos, quando consideramos dois diferentes níveis de B. Um outro exemplo, onde temos os níveis de A dentro da combinação dos níveis de B e C é dado por: *Y~B+C+A:B:C*.

d) Exemplos de modelos com efeitos cruzados e hierárquicos: *Y~A +B:A +C:A+(B*C):A*.

5.2 Delineamento Inteiramente Casualizado

Os delineamentos inteiramente casualizados, com um fator, serão utilizados para ilustrarmos inicialmente os comandos básicos do *aov*. Para isso, utilizaremos os dados apresentados por Gomes (2000), onde os efeitos no ganho de peso de animais em kg de 4 rações foram comparados. Os dados estão apresentados na Tabela 5.1.

Tabela 5.1. Ganho de peso (gp), em kg, de animais que foram submetidos a uma dieta com determinadas rações. Um delineamento inteiramente casualizado com cinco repetições (animais) e 4 rações foi utilizado (Gomes, 2000).

1	2	3	4
35	40	39	27
19	35	27	12
31	46	20	13
15	41	29	28
30	33	45	30

O modelo de análise de variância adotado é dado por:

$$Y_{ij} = \mu + \tau_i + \epsilon_{ij} \tag{5.2.1}$$

em que Y_{ij} é o ganho de peso observado no j-ésimo animal para a i-ésima ração, μ é a constante geral, τ_i é o efeito da i-ésima ração e ϵ_{ij} é o efeito do erro experimental suposto normal e independentemente distribuído com média 0 e variância comum σ^2.

Podemos fazer, inicialmente, algumas análises exploratórias, plotando os gráficos de delineamento e os *box plot*. Os gráficos de delineamento são bastantes interessantes e podem ser construídos utilizando tanto a média quanto a mediana. No primeiro comando, especificamos o arquivo de dados como argumento da função *plot.design* e os *labels* dos eixos x e y. Dividimos o ambiente gráfico em uma janela com duas células dispostas horizontalmente com o comando *par(mfrow = c(1,2))*. No final do programa retornamos a janela para uma célula simples com o comando *par(mfrow = c(1,1))*. Os resultados das marcas nas linhas verticais, mostrados na figura da esquerda, representam as médias dos tratamentos 1, 2, 3 e 4. Os tra-

tamentos possuem médias, da menor para a maior, na seguinte ordem: 1, 4, 3 e 2. As médias dos tratamentos 3 e 2 são maiores que a média geral, representada pelo traço central na linha vertical. Já os tratamentos 1 e 4, possuem média menores do que a média geral, que possui valor próximo de 30. Do lado direito, apresentamos uma figura semelhante, mas cujas marcas na linha vertical representam as medianas dos tratamentos. Houve nesse exemplo, uma inversão entre os tratamentos 1 e 3, quando comparado com a ordem estabelecida pela média. O tratamento 1, apresentou mediana idêntica a mediana global. É possível, haja vista que os gráficos da mediana e da média diferiram entre si, que haja algum efeito adverso na análise que seja atribuído a possíveis casos de *outliers*. Não faremos, nenhuma eliminação de observações nesse exemplo, embora haja suspeitas da presença desse tipo de observações discrepantes. O gráfico da esquerda mostra uma distribuição assimétrica dos dados em torno da mediana global e o tratamento 3, com mediana abaixo dessa mediana global, parece ter sido o tratamento que apresenta algum tipo de *outlier*. Uma ou duas de suas observações poderiam ser consideradas suspeitas, a observação 39 e a 45. Escolhemos, ainda, uma mesma escala vertical para fins de comparação com a opção *ylim=c(20,42)*.

```
> # lê o arquivo pimen43.txt e o atribui ao objeto pimen43
> pimen43  <- read.table("C:/daniel/Cursos/RCursoTeX/pimen43.txt",
+               header=TRUE)
> pimen43 # imprime o data frame
```

```
   trat gp
1     1 35
2     1 19
3     1 31
4     1 15
5     1 30
6     2 40
7     2 35
8     2 46
9     2 41
10    2 33
11    3 39
12    3 27
13    3 20
14    3 29
```

```
15    3 45
16    4 27
17    4 12
18    4 13
19    4 28
20    4 30
```

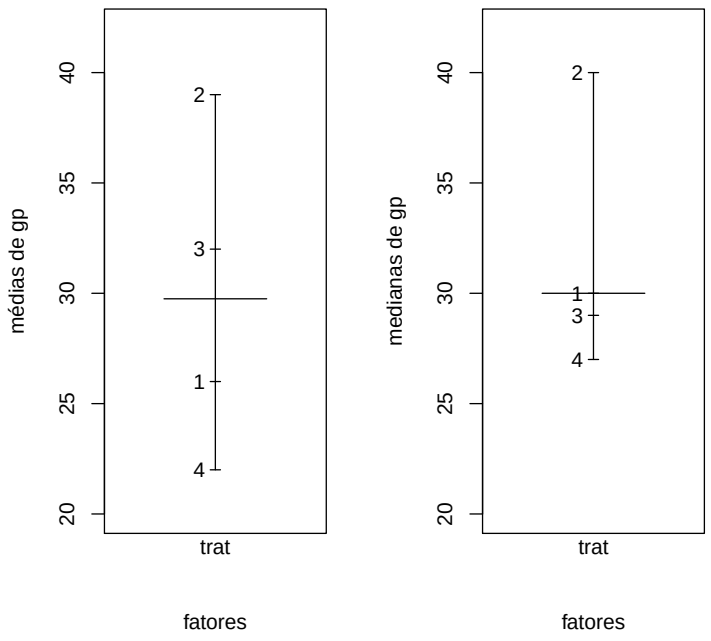
```
> is.factor(pimen43$trat) #veja que trat não é fator
```

```
[1] FALSE
```

```
> pimen43$trat <- as.factor(pimen43$trat)
> is.factor(pimen43$trat) # agora é fator
```

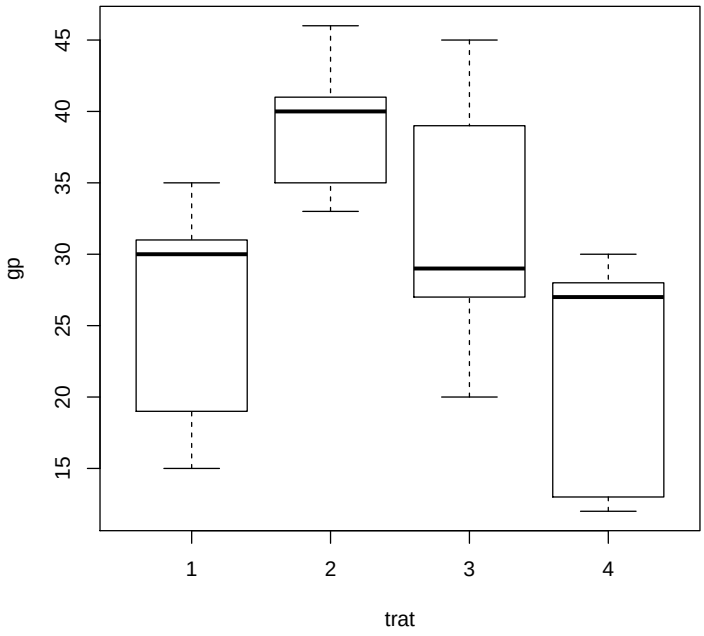
```
[1] TRUE
```

```
> # gráficos exploratórios
> par(mfrow = c(1,2))
> plot.design(pimen43,xlab="fatores",
+             ylab="médias de gp",ylim=c(20,42))
> plot.design(pimen43, fun = median,xlab="fatores",
+             ylab="medianas de gp",ylim=c(20,42))
> par(mfrow = c(1,1))
```



Os gráficos *boxplot* são gráficos bastantes interessantes de serem obtidos, principalmente se forem construídos para cada nível do fator *trat*. Fizemos isso e o resultado é apresentado na seqüência. Os 4 níveis dos tratamentos apresentam distribuições muito assimétricas em relação a mediana, o que pode ser um indicativo de não-normalidade ou de presença de *outliers* nos dados. Também podemos perceber um forte indicativo que a ração 2 difere da ração 4, pois não há sobreposição dos gráficos de ambos os tratamentos.

```
> # gráficos exploratórios
> plot(gp~trat,data=pimen43)
```



O programa R para obtenção da análise de variância do modelo (5.2.1) é dado por:

```
> # análise de variância dos dados de Pimentel Gomes p.43
> anava <- aov(gp~trat,data=pimen43)
> summary(anava) # resultado da anava

            Df  Sum Sq Mean Sq F value  Pr(>F)
trat         3  823.75  274.58  3.9939 0.02671 *
Residuals   16 1100.00   68.75
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Quando especificamos o nome do *data frame*, podemos referenciar suas colunas pelos nomes simplificados *trat* e *gp*. Assim, a sintaxe fica simplificada. Os principais resultados obtidos estão apresentados na seqüência de forma resumida. Nesse programa, modelamos o ganho de peso em função do fator rações. Não precisamos especificar nem o erro do modelo e nem a constante geral, pois isso é considerado por padrão na função *aov*. Os resultados da análise de variância estão apresentados nas Tabelas 5.2.

Tabela 5.2. Análise de variância para o delineamento inteiramente casualizado com um fator (rações) com quatro níveis e cinco repetições.

FV	G.L.	SQ	QM	F	$Pr > F$
Rações	3	823,7500	274,5833	3,99	0,0267
Erro	16	1100,0000	68,7500		
total corrigido	19	1923,7500			
CV	27,8708				
Média	29,7500				

Observando o resultado do teste F na análise de variância, devemos rejeitar a hipótese nula de igualdade de efeitos das rações, considerando um valor nominal de significância de 5%. Assim, pelo menos uma das médias de tratamento difere das demais. Devemos utilizar um teste de comparações múltiplas para identificar estas diferenças. Nesse exemplo utilizamos o teste de Tukey .

A função *TukeyHSD* é a principal função do R para objetos *aov*. Essa função deve ser utilizada com a seguinte sintaxe: *TukeyHSD(x, which, ordered = FALSE, conf.level = 0.95)*. Nesse caso, x é um objeto *aov*, *which* é um vetor de *strings*, caracteres, que indicam quais os fatores do modelo ajustado devem ser submetidos ao teste, *ordered* indica por meio de uma variável *booleana* se as médias devem ser ordenadas ou não e *conf.level* é o nível de significância.

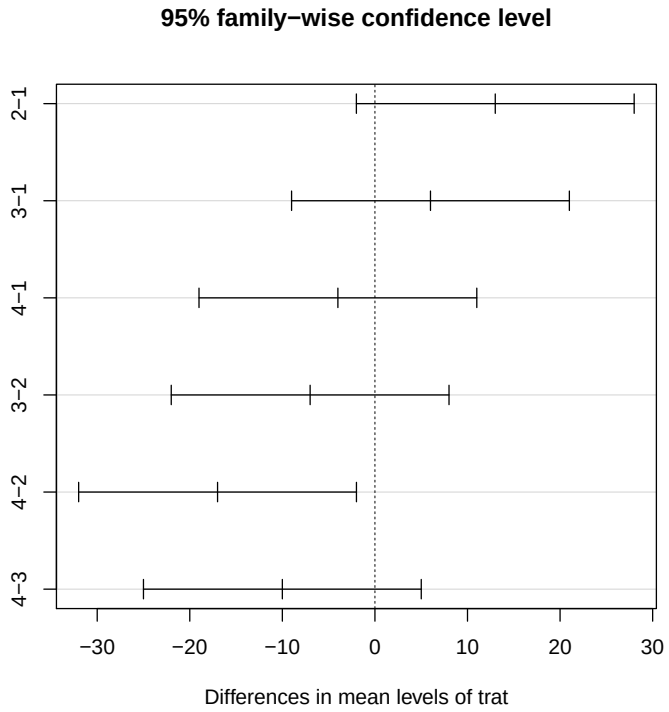
Podemos utilizar a função *glht*, do pacote *multcomp* para realizarmos tal tarefa. Podemos plotar o objeto gerado e, ainda, aplicar a função *confint* ao mesmo. O pacote *multcomp* pode ser utilizado em situações muito mais amplas, que envolvem outras classes de modelos ajustados, como *lm*, *glm* e

lme. Para a classe *lm* é utilizada a distribuição exata t multivariada e para os demais, é utilizada a distribuição assintótica normal multivariada. No programa a seguir ilustramos o uso da função *TukeyHSD*.

```
> # Comparações múltiplas para exemplo
> # de Pimentel Gomes p.43
> # aplica no objeto anava (aov)
> # o teste Tukey, com coef. confiança de 95%
> # chama a funçãoTukey
> attach(pimen43)
> THSD <- TukeyHSD(anava,wich="trat",
+                  ordered = F,conf.level = 0.95)
> THSD # imprime os resultados
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = gp ~ trat, data = pimen43)

$trat
    diff        lwr       upr     p adj
2-1   13  -2.003315 28.003315 0.1018285
3-1    6  -9.003315 21.003315 0.6687032
4-1   -4 -19.003315 11.003315 0.8698923
3-2   -7 -22.003315  8.003315 0.5553529
4-2  -17 -32.003315 -1.996685 0.0237354
4-3  -10 -25.003315  5.003315 0.2640642
```

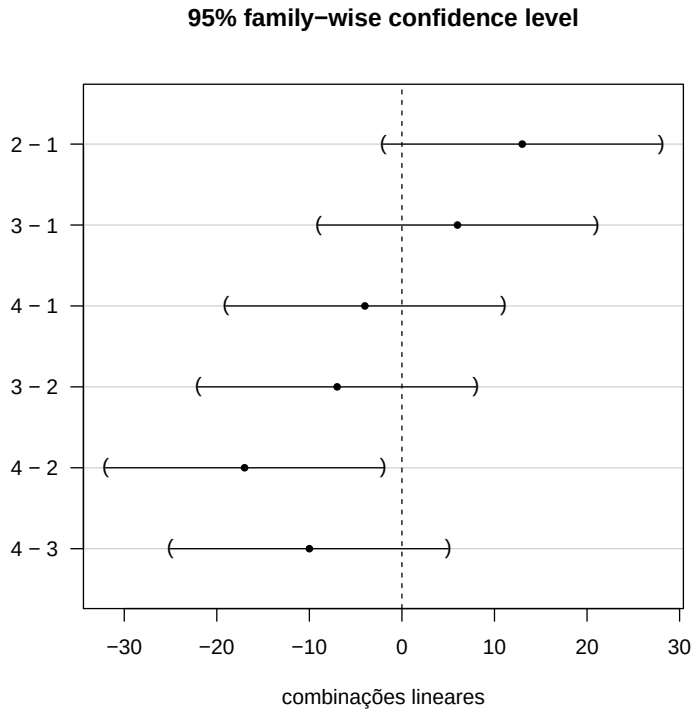
```
> plot(THSD) # obtém os gráficos das CM
> detach(pimen43)
```



Podemos obter o mesmo resultado, utilizando o pacote *multcomp*. Nesse caso a sintaxe geral é dada por *glht(model, linfct, alternative = c("two.sided", "less", "greater"),rhs = 0)*. A opção *model* deve receber um modelo ajustado das classes anteriormente mencionadas. A opção *linfct* nos permite especificar a hipótese linear a ser testada. Para as comparaçòes múltiplas em anava ou ancova, devemos utilizar a função *mcp*, que iremos explicar mais adiante. A opção *alternative*, nos permite escolher que tipo de hipótese alternativa iremos utilizar, uni ou bilateral, sendo essa última a opção padrão. Finalmente a opção *rhs*, nos permite especificar um vetor numérico que representa o valor do lado direito da hipótese formulada. A função *mcp* nos permite especificar uma matriz de contraste ou um efeito simbólico da descrição do contraste entre níveis de um fator. Essa descrição simbólica pode ser uma expressão ou um caractere, em que os níveis dos fatores sõ os nomes das variáveis. Podemos utilizar o teste de Tukey para todas as comparações emparelhadas ou de Dunnett para comparações com um nível controle.

```
> # Comparações múltiplas pelo multcomp
> # do exemplo de Pimentel Gomes p.43
> require(multcomp) # carregando o pacote multcomp
```

```
> # aplica no objeto anava (aov)
> # o teste Tukey, com coef. confiança de 95%
> mcHSD <- glht(anava,linfct = mcp(trat = "Tukey"))
> summary(mcHSD) # imprime o resultado

         Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Tukey Contrasts


Fit: aov(formula = gp ~ trat, data = pimen43)

Linear Hypotheses:
           Estimate Std. Error t value Pr(>|t|)
2 - 1 == 0   13.000      5.244   2.479   0.1015
3 - 1 == 0    6.000      5.244   1.144   0.6687
4 - 1 == 0   -4.000      5.244  -0.763   0.8699
3 - 2 == 0   -7.000      5.244  -1.335   0.5553
4 - 2 == 0  -17.000      5.244  -3.242   0.0237 *
4 - 3 == 0  -10.000      5.244  -1.907   0.2640
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Adjusted p values reported -- single-step method)

> plot(confint(mcHSD,level=0.95),
+      xlab="combinações lineares") # faz o gráfico do resultado
> confint(mcHSD,level = 0.95) # imprime os intervalos do gráfico

         Simultaneous Confidence Intervals

Multiple Comparisons of Means: Tukey Contrasts


Fit: aov(formula = gp ~ trat, data = pimen43)

Quantile = 2.8629
95% family-wise confidence level


Linear Hypotheses:
           Estimate lwr      upr
2 - 1 == 0  13.0000  -2.0133  28.0133
3 - 1 == 0   6.0000  -9.0133  21.0133
4 - 1 == 0  -4.0000 -19.0133  11.0133
3 - 2 == 0  -7.0000 -22.0133   8.0133
4 - 2 == 0 -17.0000 -32.0133  -1.9867
4 - 3 == 0 -10.0000 -25.0133   5.0133
```



Na Tabela 5.3 apresentamos o resultado do teste Tukey e as respectivas diferenças mínimas significativas (dms). As médias que possuem a mesma letra não são consideradas significativamente diferentes pelo teste SNK no nível nominal de significância de 5%. Nesse caso, as rações 2, 3 e 1 não são significativamente diferentes em média, como ocorre também com as rações 3, 1 e 4. No entanto, as rações 2 e 4 são significativamente diferentes ($P < 0{,}05$). Todas essas conclusões são facilmente obtidas pelos resultados anteriores: testes, intervalos de confiança para as diferenças de todos os pares e seus respectivos gráficos, que são visualmente apelativos. É muito comum na literatura a representação por letras unindo os níveis que não apresentam diferenças significativamente diferentes de zero, como os resultados da Tabela 5.3. Para obtermos resultados com as letras podemos utilizar o pacote *agricolae*.

O programa para obtermos o teste de média, utilizando o processo de letras para unir as médias é apresentado a seguir. Um aspecto importante do teste de Tukey refere-se ao fato de que ele controla o erro tipo I por experimento sob H_0 completa e sob a hipótese nula parcial e pode ser aplicado a hipóteses pós-experimentais.

Tabela 5.3. Teste de Tukey e médias para a fonte de variação rações juntamente com a diferença mínima significativa, dms.

Grupo	Média	r_i	Rações
A	39,000	5	2
A B	32,000	5	3
A B	26,000	5	1
B	22,000	5	4

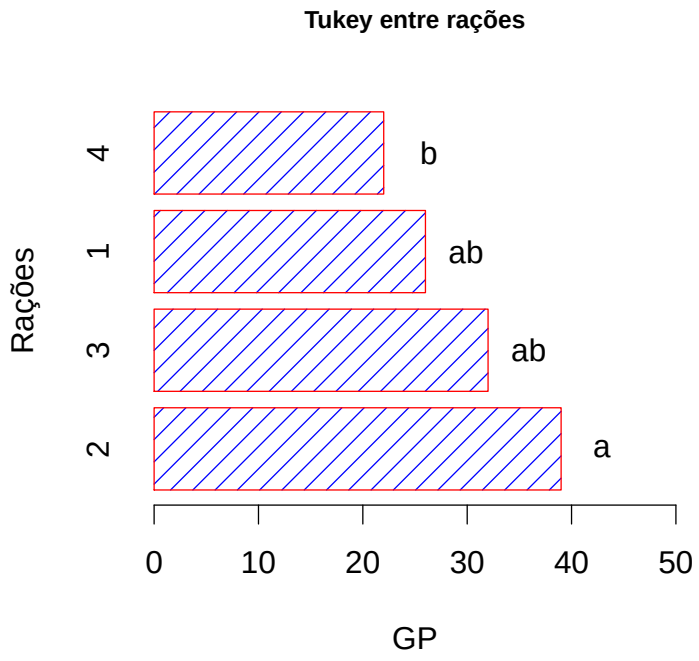
dms=15,003

```
> # Comparações múltiplas pelo agricolae
> # do exemplo de Pimentel Gomes p.43
> library(agricolae)
> attach(pimen43)
> df<-df.residual(anava)
> MSerror<-deviance(anava)/df
> Tuk <- HSD.test(gp,trat,df,MSerror, group=TRUE,
+                         main="Efeito rações nos GP")

Study: Efeito rações nos GP

HSD Test for gp

Mean Square Error:  68.75

trat,  means

  gp  std.err replication
1 26 3.820995           5
2 39 2.302173           5
3 32 4.449719           5
4 22 3.911521           5

alpha: 0.05 ; Df Error: 16
Critical Value of Studentized Range: 4.046093

Honestly Significant Difference: 15.00331

Means with the same letter are not significantly different.

Groups, Treatments and means
a          2          39
ab          3          32
```

```
ab          1          26
 b          4          22
> detach(pimen43)
> detach("package:agricolae")
```

Podemos realizar inferências de interesse sobre parâmetros decorrentes de uma combinação linear das médias por meio dos testes hipóteses e construindo intervalos de confiança. A realização de inferências sobre combinações lineares (usualmente contrastes) de médias, em geral, é o passo seguinte à rejeição da hipótese global apresentada em (5.0.1). Para identificarmos as médias que diferem entre si, no caso de comparações múltiplas, em que as combinações lineares são todos os contrastes de médias tomadas duas a duas, podemos ainda representar o resultado do teste de forma gráfica, utilizando o pacote *agricolae*, por:

```
> # Comparações múltiplas pelo agricolae
> # do exemplo de Pimentel Gomes p.43
> library(agricolae)
> par(cex=1.5)
> bar.group(Tuk,horiz=TRUE,density=8,col="blue",border="red",xlim=c(0,50))
> title(cex.main=0.8,main="Tukey entre rações",xlab="GP", ylab="Rações")
> detach("package:agricolae")
```



Como o teste F, que testa a hipótese global, não informa quais são as médias que diferem entre si, passamos, então, a realizar uma seqüência de testes de hipóteses sobre um conjunto de combinações lineares de médias utilizando os mesmos dados observados. A estes testes estão associados erros de decisão. Se a hipótese nula global for verdadeira e se uma destas hipóteses for rejeitada, estaremos cometendo o erro tipo I. O controle do erro tipo I, no caso de comparações múltiplas, envolve alguns conceitos diferentes. Se por outro lado não rejeitamos uma hipótese que deveria ser rejeitada, estaremos cometendo o erro tipo II. Acontece, também, que as taxas de erro dos tipos I e II, decorrentes da aplicação de um único teste, têm comportamentos diferentes daquelas associadas à aplicação de uma seqüência de testes.

Um grande número de estratégias existem para garantir uma taxa de erro global α para todas as comparações. Procedimentos de inferência que asseguram uma probabilidade conjunta $1 - \alpha$ contra o erro do tipo I são denominados procedimentos de *inferência simultânea* ou conjunta e procedimentos que asseguram proteção apenas para a comparação que está sendo realizada são denominados procedimentos de *inferência individual*. Nos procedimentos de inferência individual não é feito nenhum ajuste na probabilidade por causa da multiplicidade dos testes.

Algumas definições conduzem a uma taxa de erro que são dependentes da nulidade da hipótese global. Outras conduzem a uma taxa de erro dependente do número de inferências erradas em relação ao número total de inferências feitas. Assim, O'Neill e Wetherill (1971) definem duas maneiras básicas para calcularmos a taxa de erro do tipo I. Uma delas diz respeito à probabilidade de a família de testes conter pelo menos uma inferência errada e a outra, ao número esperado de inferências erradas na família.

De acordo O'Neill e Wetherill (1971) as possibilidades para as taxas de erro observadas são:

i. Taxa de erro por comparação (*comparisonwise error rate*):

$$\frac{\text{Número de inferências erradas}}{\text{Número total de inferências}}$$

ii. Taxa de erro por experimento (*experimentwise error rate*):

$$\frac{\text{Número de experimentos com pelo menos uma inferência errada}}{\text{Número total de experimentos}}$$

Os vários procedimentos de comparações múltiplas possuem diferentes controle do erro tipo I por experimento. O teste Tukey por exemplo, controla a taxa de erro por experimento sob H_0 nula e parcial, mas na medida em que o número de níveis do fator aumenta, o teste se torna mais conservador. Assim, esse teste possui elevadas taxas de erro tipo II, ou seja, baixo poder quando temos muitos níveis do fator. O teste Duncan e t de Student são muito liberais e apresentam elevadas taxas de erro tipo I por experimento, com baixas taxas de erro tipo II ou com elevado poder. Por causa de não haver controle do erro tipo I por experimento os elevados poderes não são vantajosos. O teste SNK, como já afirmamos, controla o erro tipo I sob a hipótese de nulidade completa, mas não sob a nulidade parcial. O teste t com proteção de Bonferroni é na maioria das vezes mais conservador do que o teste de Tukey, da mesma forma que ocorre com teste Scheffé quando utilizado no contexto de comparações múltiplas.

O R, entre os testes tradicionalmente utilizados, possui implementado, nesses pacotes mencionados anteriormente, apenas os testes Tukey, Waller-Duncan, LSD, Dunnett, e Bonferroni. Outras possibilidades de testes não-paramétricos ou envolvendo comparações das médias de cada nível do fator com a média geral de todos os seus níveis, podem ser realizados. Os leitores são convidados a ler a documentação das funções para obter maiores esclarecimentos.

Uma importante pressuposição na análise de variância é a homogeneidade de variâncias. Podemos testar hipóteses de igualdade de variâncias facilmente no R. Podemos utilizar a função *hov* do pacote *HH* para aplicarmos o teste de Brown e Forsythe, a função *bartlett.test* e a função *levene.test* do pacote *car*. Portanto, podemos aplicar todos esses testes comentados anteriormente e também o teste de Fligner-Killeen, quando estivermos utilizando funções pré implementadas no R. No entanto, podemos aplicar outros testes, se implementarmos nossas próprias funções. A hipótese de interesse no caso dos testes de homogeneidade de variâncias de grupos é dada por:

$$\begin{cases} H_0 : \sigma_1^2 = \sigma_2^2 = \cdots = \sigma_k^2 = \sigma^2 \\ H_1 : \text{pelo menos uma variância difere das demais} \end{cases} \tag{5.2.2}$$

em que k é o número de níveis do fator de interesse e σ_i^2 é a variância do i-ésimo nível, $i = 1, 2, \cdots, k$.

Existem vários testes para essa hipótese na literatura. O R possui implementado alguns deles, como já dissemos. Vamos descrever os testes de forma bastante simplificada. Maiores detalhes podem ser vistos em Ferreira (2005). O teste de Bartlett é um teste de razão de verossimilhanças. Para apresentarmos a estatística desse teste, devemos considerar que S_i^2 é o estimador da variância do i-ésimo nível do fator estudado em n_i repetições; $S_p^2 = \sum_{i=1}^{k}(n_i - 1)S_i^2/(n-k)$ é o estimador da variância comum das k populações (ou dos k níveis do fator); e $n = \sum_{i=1}^{k} n_i$ é total de parcelas experimentais. A estatística

$$\chi_c^2 = \frac{(n-k)\ln(S_p^2) - \sum_{i=1}^{k}\left[(n_i-1)\ln(S_i^2)\right]}{1 + \frac{1}{3(k-1)}\left[\sum_{i=1}^{k}\left(\frac{1}{n_i-1}\right) - \frac{1}{n-k}\right]} \tag{5.2.3}$$

sob H_0 possui distribuição assintoticamente de qui-quadrado com $\nu = k-1$ graus de liberdade. Assim, se o valor calculado da estatística for maior do que o quantil superior $100\alpha\%$ ($\chi^2_{\nu;\alpha}$) da distribuição de qui-quadrado com ν graus de liberdade, a hipótese nula (5.2.2) deve ser rejeitada.

Os demais testes que veremos na seqüência são os de Levene e Brown e Forsythe (Ferreira, 2005). Estes testes são baseados em uma análise de variância, onde os valores originais da variável resposta são substituídos por outra variável Z_{ij}. O teste F é aplicado e a sua estatística é dada pela razão da variação entre grupos e dentro de grupos. A diferença básica entre os procedimentos é determinada pela forma como os valores desta nova variável são obtidos. Para o teste de Levene, duas opções existem. A primeira é baseada nos desvios da i-ésima média, tomados em módulo, opção default da função correspondente no R. Assim, os valores para a variável $Z_{ij} = |Y_{ij} - \bar{Y}_{i.}|$ são obtidos e o teste F é aplicado. Para a segunda opção, devemos obter os valores da variável $Z_{ij} = (Y_{ij} - \bar{Y}_{i.})^2$, a qual refere-se aos desvios da média do i-ésimo nível do fator tomados ao quadrado. Para realizarmos o teste de Brown e Forsythe devemos obter esta variável por: $Z_{ij} = |Y_{ij} - \tilde{Y}_i|$, sendo $\tilde{Y}_i$ a mediana do i-ésimo nível do fator.

Obtidos os valores desta variável para as n observações amostrais, devemos utilizar a estatística do teste:

$$F_c = \frac{(n-k)\sum_{i=1}^{k} n_i \left(\bar{Z}_{i.} - \bar{Z}_{..}\right)^2}{(k-1)\sum_{i=1}^{k}\sum_{j=1}^{n_i} \left(Z_{ij} - \bar{Z}_{i.}\right)^2} \tag{5.2.4}$$

em que:

$$\bar{Z}_{i.} = \frac{\sum_{j=1}^{n_i} Z_{ij}}{n_i} \qquad \text{e} \qquad \bar{Z}_{..} = \frac{\sum_{i=1}^{k}\sum_{j=1}^{n_i} Z_{ij}}{n}$$

para testarmos a hipótese nula (5.2.2). Sob a hipótese de nulidade essa estatística possui distribuição aproximada F com $\nu_1 = k-1$ e $\nu_2 = n-k$ graus de liberdade. Devemos rejeitar a hipótese nula se F_c, de (5.2.4), for maior do que o quantil superior $100\alpha\%$ (F_{α,ν_1,ν_2}) da distribuição F.

O teste de Fligner-Killeen modificado é baseado nas estatísticas de ordem dos valores absolutos $Z_{ij} = |Y_{ij} - \tilde{Y}_i|$ e nos escores crescentes correspondentes à cada uma delas dado por:

$$a_{ij}^{\ell} = \Phi^{-1}\left(\frac{1 + \frac{\ell}{n+1}}{2}\right),$$

em que $\Phi^{-1}(.)$ é a função de distribuição inversa da normal padrão e $\ell = 1$, 2, ..., n é o posto correspondente da observação $Z_{ij} = |Y_{ij} - \tilde{Y}_i|$. Caso haja empates, devemos atribuir o valor da média dos postos que seriam atribuídos caso não houvesse empates às posições empatadas. Por exemplo, se o menor e o segundo menor valores forem iguais, os postos destas duas posições serão 1,5 e 1,5, ou seja, corresponderão à média dos postos 1 e 2, que seriam os postos caso não houvesse empates.

Após obtidos os valores de a_{ij}, devemos calcular a estatística do teste dada por:

$$\chi^2_c = \frac{\sum_{i=1}^{k} n_i \left(\bar{A}_i - \bar{a}\right)^2}{V^2},$$

em que $n = \sum_{i=1}^{k} n_i$,

$$\bar{A}_i = \frac{1}{n}\sum_{j=1}^{n_i} a_{ij}, \qquad \bar{a} = \frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{n_i} a_{ij}$$

$$V^2 = \frac{1}{n-1}\sum_{i=1}^{k}\sum_{j=1}^{n_i} (a_{ij} - \bar{a})^2 \qquad .$$

Sob a hipótese nula de homogeneidade de variâncias, a estatística χ^2_c possui distribuição assintótica qui-quadrado com $\nu = k - 1$ graus de liberdade. Segundo alguns autores, esse teste é muito robusto a desvios de normalidade, sendo considerado o mais robusto de todos.

As aplicações desses testes foram ilustradas nos programas R apresentados na seqüência para o experimento de ganho de peso das 4 rações. Para o teste Levene, utilizando os desvios da média do tratamento tomados ao quadrado, implementamos nossa própria função. na seqüência ilustra a aplicação do teste de Levene com desvios absolutos da média. Obtivemos um valor-p para as estatísticas dos testes, em todos os casos, superior ao valor nominal de 5% e tomamos a decisão de não rejeitar a hipótese de homogeneidade de variâncias.

```
> # Função para fazer o teste Levene com os valores quadráticos
> # dos resíduos
> Levene.Square <- function(y, group)
+ {
+      group <- as.factor(group)  # precaução
+      meds <- tapply(y, group, mean)
+      resp <- (y - meds[group])^2
+      anova(lm(resp ~ group))[1, 4:5]
+ }
> attach(pimen43) # prepara o data frame pimen43 - por garantia
> Levene.Square(gp,trat) # chamando a função que implementamos

      F value Pr(>F)
group  1.3927 0.2812
```

```
> # utilizando o teste Bartlett
> bartlett.test(gp ~ trat)

        Bartlett test of homogeneity of variances

data:  gp by trat
Bartlett's K-squared = 1.5284, df = 3, p-value =
0.6757

> # utilizando o teste Levene com
> # valores absolutos dos erros
> library(car) # carregando o pacote apropriado
> levene.test(gp, trat)

Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  3  0.3324  0.802
      16

> detach("package:car") # libera memória do pacote
> # utilizando o teste BF
> library(HH) # carregando o pacote apropriado
> hov(gp ~ trat, data=pimen43)

        hov: Brown-Forsyth

data:  gp
F = 0.3324, df:trat = 3, df:Residuals = 16,
p-value = 0.802
alternative hypothesis: variances are not identical

> detach("package:HH") # libera o pacote
> # utilizando o teste Fligner-Killeen
> fligner.test(gp ~ trat)

        Fligner-Killeen test of homogeneity of
        variances

data:  gp by trat
Fligner-Killeen:med chi-squared = 1.1534, df =
3, p-value = 0.7642

> detach(pimen43)
```

Da mesma forma que fizemos para o teste Levene, resolvemos implementar uma função para aplicar o teste Fligner-Killeen de homogeneidade de variâncias entre níveis de um fator. A função implementada, cujo objetivo foi apenas didático, está apresentada na seqüência.

```
> # Função para aplicar o
> # teste de Fligner-Killeen
> Fligner_Killeen <- function(y,group)
+ {
+   group <- as.factor(group)  # precaução
+   n <- length(y)
+   k <- length(names(summary(group)))
+   meds <- tapply(y, group, median)
+   ni <- tapply(y, group, length)
+   resp <- abs(y - meds[group])
+   zz <- data.frame(resp,group)
+   zz <- zz[order(resp),]
+   zz$resp <- rank(zz$resp)
+   zz$resp <- qnorm((1 + zz$resp/(n+1))/2)
+   Ai <- tapply(zz$resp, zz$group, mean)
+   abar <- mean(zz$resp)
+   V2 <- sum((zz$resp-abar)^2)/(n-1)
+   X2 <- sum(ni*(Ai-abar)^2)/V2
+   p.value <- 1-pchisq(X2,k-1)
+   return(list(X2=X2,p.value=p.value))
+ }
> Fligner_Killeen(pimen43$gp,pimen43$trat)

$X2
[1] 1.153388

$p.value
[1] 0.7642041
```

5.3 Estrutura Cruzada de Tratamentos

Em muitas situações experimentais temos delineamentos mais complexos que o inteiramente casualizado, ou mesmo para esse delineamento, podemos ter mais de um fator em estruturas mais intrincadas. Entre os delineamentos mais complexos, encontram-se os blocos casualizados, os quadrados latinos e os látices. Além da estrutura experimental ser mais complexa, a estrutura de tratamentos pode também ir além um simples fator. Uma estrutura muito comum é a cruzada, onde os fatores são combinados fatorialmente. Como a modelagem no R é bastante simples, independentemente das estruturas experimental e de tratamentos, vamos ilustrar o seu uso com um caso em que temos um delineamento em blocos casualizados com dois fatores quantitativos (adubo mineral e torta de filtro). Foram utilizados os

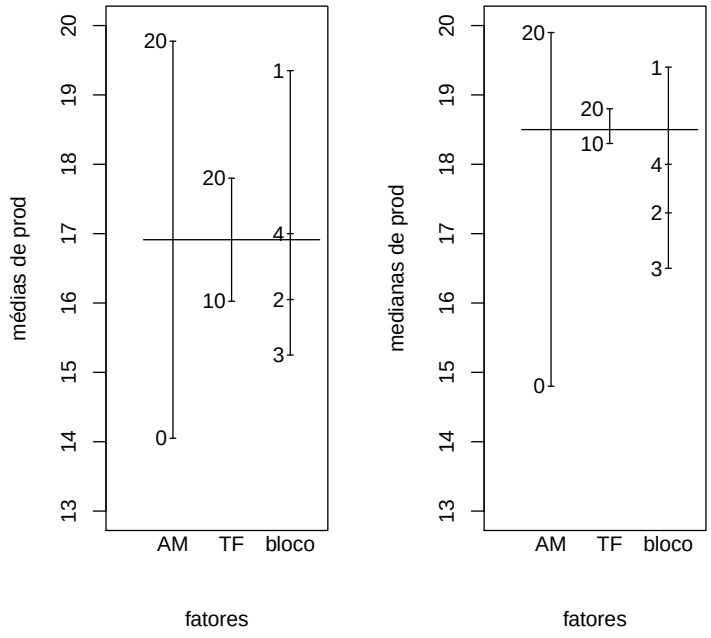
níveis 0 e 20 kg/ha de adubo mineral e 10% e 20% de torta de filtro. Cada combinação fatorial dos tratamentos foi repetida 4 vezes e a produtividade das plantas foi mensurada. O programa R para a análise de variância desse modelo está apresentado na seqüência. O modelo estatístico da análise de variação é dado por:

$$Y_{ijk} = \mu + \beta_i + \alpha_j + \tau_k + \delta_{jk} + \epsilon_{ijk} \tag{5.3.1}$$

em que μ é a constante geral do modelo, β_i é o efeito do i-ésimo bloco, α_j é o efeito do j-ésimo adubo mineral, τ_k é o efeito da k-ésima torta de filtro, δ_{jk} é o efeito da interação entre a j-ésima dose do adubo mineral e a k-ésima dose da torta de filtro e ϵ_{ijk} é o erro experimental suposto normal e independentemente distribuído com média 0 e variância σ^2.

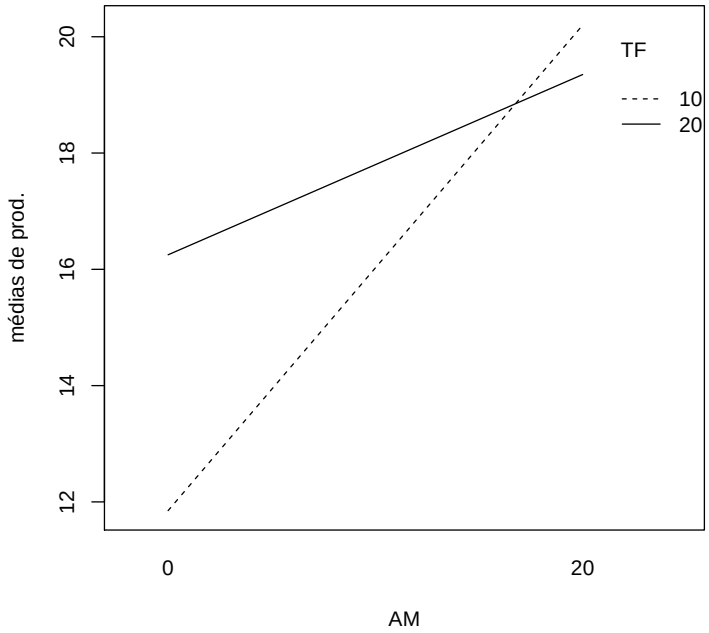
```
> # lê o arquivo fat.txt e o atribui ao objeto fat
> fat  <- read.table("C:/daniel/Cursos/RCursoTeX/fat.txt",
+                    header=TRUE)
> fat # imprime o data frame

   AM TF bloco prod
1   0 10     1 18.0
2  20 10     1 20.6
3   0 20     1 19.6
4  20 20     1 19.2
5   0 10     2  8.6
6  20 10     2 21.0
7   0 20     2 15.0
8  20 20     2 19.6
9   0 10     3  9.4
10 20 10     3 18.6
11  0 20     3 14.6
12 20 20     3 18.4
13  0 10     4 11.4
14 20 10     4 20.6
15  0 20     4 15.8
16 20 20     4 20.2

> fat$AM <- as.factor(fat$AM)
> fat$TF <- as.factor(fat$TF)
> fat$bloco <- as.factor(fat$bloco)
```

```
> # gráficos exploratórios
> par(mfrow = c(1,2))
> plot.design(fat,xlab="fatores",
+             ylab="médias de prod",
+             ylim=c(13,20),xlim=c(0,3.5))
> plot.design(fat, fun = median,xlab="fatores",
+             ylab="medianas de prod",
+             ylim=c(13,20),xlim=c(0,3.5))
> par(mfrow = c(1,1))
```



Um outro gráfico exploratório bastante útil é o gráfico da interação entre fatores. Podemos observar se existe interação entre os níveis dos fatores, o que poderá nos ajudar a explorar as relações entre os fatores, facilitando a interpretação e a recomendação após a aplicação dos testes. Podemos observar que para o nível 0 de adubo mineral, a produtividade média é maior para torta de filtro nível 20%. Quando passamos do nível 0 para o nível 20 de adubo mineral, verificamos que há uma inversão do comportamento, com a média de produtividade do nível 10% superando a do nível 20%. Esse comportamento é típico de interação, embora a hipótese de interação deva ser formalmente confirmada com um teste estatístico.

```
> attach(fat)
> interaction.plot(AM, TF, prod,xlab="AM",
+             ylab="médias de prod.")
> detach(fat)
```



Finalmente, vamos apresentar os comandos para obtermos a análise de variância. Optamos por utilizar a função *aov*.

```
> attach(fat)
> summary(aov(prod ~ bloco + AM + TF + AM:TF))

            Df  Sum Sq Mean Sq F value    Pr(>F)
bloco        3  37.827  12.609  3.0099 0.0871124 .
AM           1 131.103 131.103 31.2956 0.0003367 ***
TF           1  12.603  12.603  3.0084 0.1168637
AM:TF        1  27.563  27.563  6.5795 0.0304340 *
Residuals    9  37.703   4.189
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> detach(fat)
```

O resultado da análise de variação foi reapresentado na Tabela 5.4. Verificamos efeitos significativos para adubo mineral ($P < 0{,}05$) e para interação ($P < 0{,}05$).

Tabela 5.4. Análise da variação para o modelo fatorial (2 fatores) em um delineamento de blocos casualizados.

FV	G.L.	SQ	QM	F	$Pr > F$
Bloco	3	37,83	12,6100	3,01	0,09
AM	1	131,10	131,1000	31,30	0,00
TF	1	12,60	12,6000	3,01	0,12
AM*TF	1	27,55	27,5500	6,58	0,03
Erro	9	37,70	4,1889		
Total	15	246,80			

Poderíamos pensar, pelo menos inicialmente, em desdobrar a interação adubo mineral e torta de filtro $AM \times TF$, estudando o efeito do adubo mineral em cada nível de torta. Uma abordagem um pouco mais interessante consiste em utilizar um modelo de regressão contendo efeitos de ambos os fatores simultaneamente. Esse tipo de modelo é conhecido como superfície de resposta. Vamos utilizar um modelo com três parâmetros, sem considerar o intercepto. O modelo de análise de variância para as fontes de variação adubo mineral (AM), torta de filtro (TF) e interação adubo mineral e torta de filtro ($AM \times TF$) possui 3 graus de liberdade associados. O modelo a ser escolhido deve conter apenas 2 parâmetros, para que o grau de liberdade remanescente fosse utilizado para o teste da falta de ajuste do modelo. Nesse exemplo não podemos aplicar tal teste, pois esgotamos os três graus de liberdade disponíveis. O R^2 será igual à unidade, mostrando que obrigamos a superfície a passar exatamente sobre os pontos médios observados. Utilizaremos essa superfície apenas para ilustrar como recalcular determinadas quantidades, como R^2, erros padrões e as estatística e valores-p dos testes F e t, para as hipóteses de interesse. O modelo que ajustamos é:

$$\bar{Y}_{.jk} = \beta_0 + \beta_1 A_j + \beta_2 T_k + \beta_3 AT_{jk} + \bar{\epsilon}_{jk} \tag{5.3.2}$$

em que $\bar{Y}_{.jk}$ é a resposta média para os níveis j e k do adubo mineral e da torta de filtro, β_ℓ são os parâmetros da regressão, A_j é o nível j do adubo mineral, T_k é o k-ésimo nível da torta de filtro, AT_{jk} é o produto dos níveis j e k do adubo mineral e da torta de filtro e $\bar{\epsilon}_{jk}$ é o erro médio associado com variância σ^2/r, sendo $r = 4$.

Para ajustar o modelo da equação (5.3.2) foi utilizado a função *lm*, com todas as observações experimentais e não apenas as médias. Podemos, entretanto, utilizar somente as médias da interação para realizarmos esse ajuste. Nesse caso as somas de quadrados deveriam ser recalculadas para a escala original e optamos por não fazê-lo e utilizarmos todos os dados, evitando essa operação. Assim, criamos a variável $I(AM * TF)$ dada pelo produto dos níveis de AM pelos de TF. O programa R resultante é dado por:

```
> # conversão de fator para numérico
> # veja sintaxe - se converter direto retorna níveis
> # 1, 2, etc. Ex. as.numeric(fat$AM)
> fat$AM <- as.numeric(levels(fat$AM)[fat$AM])
> fat$TF <- as.numeric(levels(fat$TF)[fat$TF])
> attach(fat)
> # modelo para estimarmos os parâmetros de regressão
> # erros padrões incorretos, pois o resíduo incluiu
> # o efeito de blocos
> summary(anv1 <- lm(prod ~ I(AM) + I(TF) + I(AM*TF)))

Call:
lm(formula = prod ~ I(AM) + I(TF) + I(AM * TF))

Residuals:
   Min     1Q Median     3Q    Max
-3.250 -1.337 -0.300  0.500  6.150

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.45000    2.80494   2.656   0.0209 *
I(AM)        0.68000    0.19834   3.428   0.0050 **
I(TF)        0.44000    0.17740   2.480   0.0289 *
I(AM * TF)  -0.02625    0.01254  -2.093   0.0583 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 2.509 on 12 degrees of freedom
Multiple R-squared: 0.694,        Adjusted R-squared: 0.6174
F-statistic:  9.07 on 3 and 12 DF,  p-value: 0.002069
```

```
> anova(anv1)
```

```
Analysis of Variance Table

Response: prod
           Df  Sum Sq Mean Sq F value    Pr(>F)
I(AM)       1 131.103 131.103 20.8292 0.0006503 ***
I(TF)       1  12.602  12.602  2.0023 0.1824886
I(AM * TF)  1  27.563  27.563  4.3791 0.0583038 .
Residuals  12  75.530   6.294
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> # modelo para estimarmos os erros padrões adequados de regressão
> # erros padrões corretos, pois o efeito de bloco foi eliminado
> # do resíduo, mas o modelo não estima adequadamente o intercepto
> summary(anv2 <- lm(prod ~ bloco + I(AM) + I(TF) + I(AM*TF)))
```

```
Call:
lm(formula = prod ~ bloco + I(AM) + I(TF) + I(AM * TF))

Residuals:
    Min      1Q  Median      3Q     Max
-2.5875 -0.6000  0.0375  0.8000  3.7125

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.88750    2.45396   4.029  0.00298 **
bloco2      -3.30000    1.44727  -2.280  0.04855 *
bloco3      -4.10000    1.44727  -2.833  0.01963 *
bloco4      -2.35000    1.44727  -1.624  0.13888
I(AM)        0.68000    0.16181   4.202  0.00230 **
I(TF)        0.44000    0.14473   3.040  0.01401 *
I(AM * TF)  -0.02625    0.01023  -2.565  0.03043 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.047 on 9 degrees of freedom
```

```
Multiple R-squared: 0.8472,        Adjusted R-squared: 0.7454
F-statistic: 8.319 on 6 and 9 DF,  p-value: 0.002915
```

```
> anova(anv2)
```

```
Analysis of Variance Table

Response: prod
           Df  Sum Sq Mean Sq F value    Pr(>F)
bloco       3  37.827  12.609  3.0099 0.0871124 .
I(AM)       1 131.102 131.102 31.2956 0.0003367 ***
I(TF)       1  12.602  12.602  3.0084 0.1168637
I(AM * TF)  1  27.563  27.563  6.5795 0.0304340 *
Residuals   9  37.702   4.189
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> detach(fat)
```

Como fizemos as análises utilizando os dados originais, a soma de quadrados de modelo de regressão (171,2675), apresentada na Tabela 5.5, representa a soma das somas de quadrados de AM, TF e $AM \times TF$ (131,10, 12,60 e 27,55) obtidas na análise de variância (Tabela 5.4). A soma de quadrados do resíduo (75,53) desta análise contempla a soma de quadrados do erro puro (37,70) e a soma de quadrados de blocos (37,83), conforme pode ser observado na primeira parte da análise anterior. Também conteria a soma de quadrados do desvio do modelo ajustado, se não tivéssemos utilizado um modelo completo, como foi o caso. Como, nesse exemplo, esgotamos os graus de liberdade do modelo, não houve desvios. Devemos sempre isolar todos estes componentes *manualmente* ou utilizando um modelo completo, ou seja, um modelo que esgote todos os graus de liberdade de cada fator e de suas interações que foram considerados na regressão. O R não tem uma opção que nos possibilita ajustar o modelo dentro do contexto da análise de variância. Devemos utilizar a função *lm* e a função *anova* e os resultados obtidos devem ser corrigidos.

Não precisamos ajustar nenhum coeficiente de regressão na primeira parte da análise, mas devemos atentar para o fato de que os erros padrões e os testes associados, o R^2 do modelo e outros testes da análise de variância estão incorretos. O $R^2 = 0{,}6940$ utilizou a soma de quadrados de totais corrigido como denominador, mas deveria utilizar a soma de quadrados

Tabela 5.5. Análise da variação para o modelo de regressão para o exemplo fatorial da adubação com 2 fatores.

FV	G.L.	SQ	QM	F	$Pr > F$
Modelo	3	171,27	57,0900	9,070	0,002
Erro	12	75,53	6,2942		
Total	15	246,80			

de tratamentos $SQA + SQT + SQAT = 171{,}27$. Assim, o real valor do coeficiente de determinação é $R^2 = 1$. As estimativas dos parâmetros do modelo e os seus erros padrão estão apresentados na Tabela 5.6. Estes resultados referem-se as estimativas originais do programa R, obtidas na primeira parte do programa, as quais foram corrigidas na segunda parte do programa. Não podemos utilizar os testes, mas podemos utilizar o modelo ajustado.

Tabela 5.6. Estimativas dos parâmetros do modelo com seus erros padrões e teste da hipótese para $\beta_i = 0$ fornecidas originalmente pelo R na primeira parte do programa.

Parâmetro	GL	Estimativas	Erro padrão	t_c para $H_0 : \beta_i = 0$	$Pr > \|t\|$
β_0	1	7,4500	2,8049	2,66	0,021
β_1	1	0,6800	0,1983	3,43	0,005
β_2	1	0,4400	0,1774	2,48	0,029
β_3	1	-0,0263	0,0125	-2,09	0,058

O erro padrão de uma determinada estimativa é obtido pela expressão (3.3.6), ou seja, por $\sqrt{x_{ii}S^2}$, em que S^2 é o estimador da variância residual e x_{ii} a diagonal de $(X'X)^{-1}$. Como S^2 utilizada foi a variância contendo outros efeitos do modelo, como o efeito de blocos nesse caso, e em outros modelos pode incluir outros fatores do modelo, desvio de regressão e erro puro, então devemos obter o quadrado do erro padrão, multiplicar pela estimativa da variância do erro do modelo de regressão obtidos na primeira parte da análise e assim obter x_{ii}. O novo erro padrão é estimado multiplicando x_{ii} pelo QME da análise de variância (Tabela 5.4) e extraindo a raiz quadrada. Felizmente, podemos construir um modelo completo, utilizando o efeito de bloco nesse caso. Em outras situações devemos incluir todos os efeitos de outros fatores e de suas interações no modelo, para obtermos os

testes apropriados. Devemos incluir parâmetros referentes a cada fonte de variação até esgotarmos seus graus de liberdade. No caso só o efeito linear de AM, TF e de sua interação foi modelada na segunda parte, pois cada fonte de variação dessa possuía um grau de liberdade apenas.

Para ilustrarmos, vamos considerar o erro padrão da estimativa de β_0. Esse erro padrão foi igual a 2,8049. Devemos elevá-lo ao quadrado e dividi-lo por 6,2942, obtendo $2{,}8049^2/6{,}2942 = 1{,}25$. Esse valor deve ser multiplicado pelo quadrado médio do erro puro (4,1889) e em seguida extrair sua raiz quadrada. O valor obtido é $\sqrt{1{,}25 \times 4{,}1889} = 2{,}2883$. Repetindo esse processo para todos os demais parâmetros, encontramos os resultados apresentados na Tabela 5.7, após recalcularmos os valores-p da última coluna. Concluímos que todos os efeitos foram significativamente importantes na presença dos demais, o que não havia acontecido para $AM \times TF$ ou β_3, quando consideramos a primeira parte da análise, que está incorreta. Felizmente, não precisamos fazer isso manualmente, pois o R nos permite ajustarmos facilmente os dois casos e obtermos assim, os resultados dos testes de hipóteses corretos.

Tabela 5.7. Estimativas dos parâmetros do modelo com seus erros padrões e teste da hipótese para $\beta_i = 0$ devidamente corrigidas.

Parâmetro	GL	Estimativas	Erro padrão	t_c para $H_0 : \beta_i = 0$	$Pr > \|t\|$
β_0	1	7,4500	2,2882	3,26	0,010
β_1	1	0,6800	0,1618	4,20	0,002
β_2	1	0,4400	0,1447	3,04	0,014
β_3	1	-0,0263	0,0102	-2,58	0,030

A análise de variância para o modelo de regressão devidamente corrigida foi apresentada na Tabela 5.8. Não temos nesse caso graus de liberdade para o desvio de regressão, que nos possibilitaria aplicar o conhecido teste da falta de ajuste, um dos mais importantes testes na análise de regressão. O ideal é ajustarmos modelos que não esgotem os graus de liberdade de tratamentos, permitindo que haja pelo menos um grau de liberdade para realizarmos o teste da falta de ajuste.

Muitos pesquisadores não se atentam para estas correções da análise de regressão quando submetida ao R ou a outro programa de análise estatística, cujas funções de ajustes de modelos de regressão não são opções

Tabela 5.8. Análise da variação devidamente corrigida para o modelo de regressão do exemplo fatorial da adubação com 2 fatores.

FV	G.L.	SQ	QM	F	$Pr > F$
Modelo	3	171,27	57,0900	13,62	0,001
Desvios	0	-	-	-	-
Erro	9	37,70	4,1889		
Tratamento	3	171,27			

das funções que ajustam os modelos lineares de análise de variância. Assim, muitas inferências podem estar comprometidas e até mesmo incorretas. Outro aspecto interessante que devemos observar refere-se ao teste t para o efeito da torta de filtro e o teste F, para esse mesmo fator, aplicado na análise de variância. Como temos apenas 1 grau de liberdade e, portanto, um parâmetro é suficiente para modelar o efeito do fator no modelo de regressão, poderíamos pensar que os resultados dos dois testes seriam equivalentes. Infelizmente, essa percepção está incorreta. O teste t e o teste F deram resultados diferentes e isso decorre do fato de o teste t ser obtido a partir de reduções de modelos parciais e o teste F, seqüenciais. O t testa o efeito da torta de filtro ajustado para todos os outros efeitos, incluindo o efeito da interação com AM linear $\times$ TF linear e o F, apenas considera o ajuste da torta para o efeito da correção e do adubo mineral. Essa é uma questão que não tem sido considerada e, até mesmo, ignorada por muitos pesquisadores. Podemos constatar isso comparando os resultados do teste t e da função *Anova* do pacote *car*, conforme ilustrado a seguir.

```
> # Anava do tipo II para o modelo de Regressão
> attach(fat)
> library(car)
> Anova(anv2)

Anova Table (Type II tests)

Response: prod
           Sum Sq Df F value   Pr(>F)
bloco      37.828  3  3.0099 0.087112 .
I(AM)      73.984  1 17.6608 0.002298 **
I(TF)      38.720  1  9.2429 0.014013 *
I(AM * TF) 27.563  1  6.5795 0.030434 *
Residuals  37.702  9
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> detach("package:car")
> detach(fat)
```

O modelo ajustado é dado por:

$$\hat{\hat{Y}}_{jk} = 7{,}45 + 0{,}68A_j + 0{,}44T_k - 0{,}0263AT_{jk}$$

Na Figura 5.1 apresentamos a superfície de resposta ajustada para os valores médios dos níveis dos fatores AM e TF em relação a produção. Observamos que as respostas máximas foram obtidas quando se utilizou a dose 20 kg/ha de adubo mineral com a dose mínima de torta de filtro (10%).

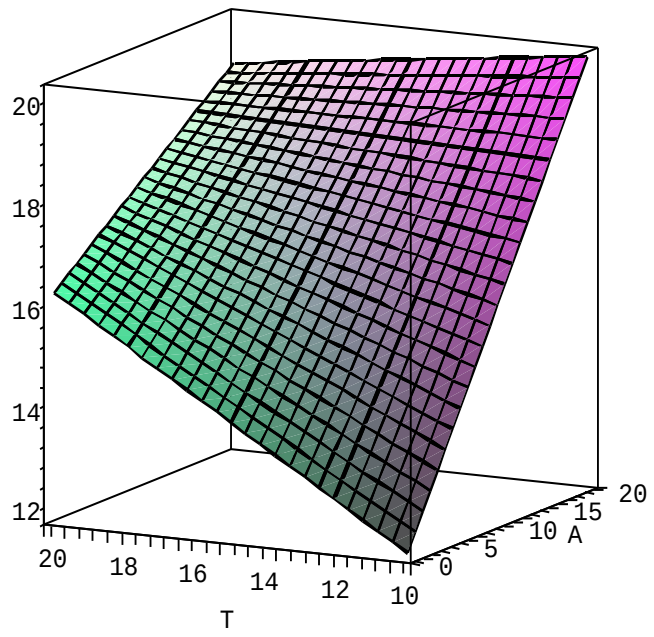


Figura 5.1. Modelo ajustado de superfície de resposta para os dados de produção em função da adubação mineral (AM) e da adubação orgânica com torta de filtro (TF).

Podemos observar que haverá uma queda acentuada da produtividade se não for utilizado adubo mineral. Na condição de ausência de adubo mineral, se passarmos do nível de 10% de torta para 20%, observamos um incremento na produtividade. No entanto, se estamos utilizando a dose de 20 kg/ha de adubo mineral, esse aumento de 10% para 20% na torta de filtro provoca uma redução da produtividade média. Assim, devemos recomendar as doses de 20 kg/ha de adubo mineral e 10% de torta de filtro para obtermos a máxima resposta.

5.4 Modelos Lineares Com Mais de Um Erro

Em algumas situações reais nos deparamos com modelos que contém mais de um erro experimental. Isso acontece em experimentos como o de parcelas subdivididas, sub-subdivididas ou em faixas. Um outro caso que ocorre normalmente é o de parcela subdividida no tempo. Nesse caso, o delineamento em geral é simples, como o inteiramente casualizado ou o de blocos casualizados e cada parcela ou unidade experimental é avaliada ao longo do tempo. Se pudermos supor que existe uma variância constante entre as observações ao longo do tempo e que a estrutura de correlação entre diferentes tempos é a mesma, então podemos fazer uma abordagem biométrica bastante simples, tratando esse modelo com um modelo de parcelas subdividas no tempo. Assim, mais de um erro irá aparecer no modelo e esse caso pode ser encaixado dentro dessa seção. Essa estrutura de correlação é denominada de simetria composta.

Vamos ilustrar esse tipo de modelo, contendo mais de um erro, com um exemplo de parcela subdividida no tempo. Um adubo mineral foi utilizado como fator principal, onde desejávamos comparar seus três níveis 0, 10 e 20 kg/ha. Essas três doses foram avaliadas em um delineamento em blocos completos casualizados com 2 repetições. O interesse era observar e mensurar o efeito do adubo mineral ao longo do tempo no crescimento das plantas. Assim, foram avaliadas as alturas das plantas durante 3 meses consecutivos. Os meses, embora seja um fator quantitativo, não pôde ser tratado assim nesse exemplo, uma vez que as medidas não foram de espaçadas com exatamente 30 dias, e o valor exato dos dias entre uma mensuração e outra não nos foi informado. Assim, tratamos o fator mês como qualitativo. O modelo estatístico para esse experimento é dado por:

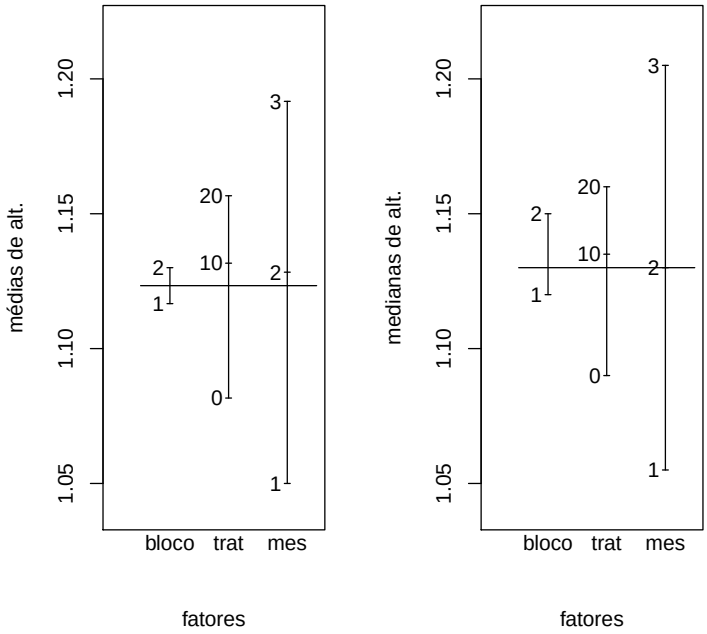
$$Y_{ijk} = \mu + \alpha_i + \beta_j + \epsilon_{ij} + \gamma_k + \epsilon_{jk} + \delta_{ik} + \epsilon_{ijk} \tag{5.4.1}$$

em que Y_{ijk} é a observação da altura das plantas em metros, μ é a constante geral do modelo, α_i é o efeito do i-ésimo nível da adubação química, β_j é o efeito do j-ésimo bloco, ϵ_{ij} é o efeito do erro experimental entre a i-ésima dose e o j-ésimo bloco, γ_k é o efeito do k-ésimo mês, ϵ_{jk} é efeito do erro experimental do j-ésimo bloco com o k-ésimo mês, δ_{ik} é o efeito da interação entre a i-ésima dose de adubo químico com o k-ésimo mês e ϵ_{ijk} é o erro experimental entre a i-ésima dose, j-ésimo bloco e k-ésimo mês.

O programa R contendo os dados experimentais (*data frame*) e a sintaxe para especificar os gráficos exploratórios é apresentado na seqüência.

```
> # lê o arquivo sub.txt e o atribui ao objeto sub
> sub  <- read.table("C:/daniel/Cursos/RCursoTeX/sub.txt",
+                    header=TRUE)
> sub # imprime o data frame

   bloco trat mes  alt
1      1    0   1 1.00
2      1   10   1 1.05
3      1   20   1 1.08
4      2    0   1 1.02
5      2   10   1 1.06
6      2   20   1 1.09
7      1    0   2 1.10
8      1   10   2 1.12
9      1   20   2 1.14
10     2    0   2 1.08
11     2   10   2 1.15
12     2   20   2 1.18
13     1    0   3 1.14
14     1   10   3 1.20
15     1   20   3 1.22
16     2    0   3 1.15
17     2   10   3 1.21
18     2   20   3 1.23

> sub$bloco <- as.factor(sub$bloco)
> sub$mes   <- as.factor(sub$mes)
> sub$trat <- as.factor(sub$trat)
> # gráficos exploratórios
> par(mfrow = c(1,2))
```

```
> plot.design(sub,xlab="fatores",
+             ylab="médias de alt.",
+             ylim=c(1.04,1.22),xlim=c(0,3.5))
> plot.design(sub, fun = median,xlab="fatores",
+             ylab="medianas de alt.",
+             ylim=c(1.04,1.22),xlim=c(0,3.5))
> par(mfrow = c(1,1))
```



O programa R contendo a sintaxe para especificar os erros do modelo e determinar os testes corretos é apresentado na seqüência. Os erros intermediários do modelo não são prontamente reconhecidos pelo R e devem ser indicados para que os testes de hipóteses sejam aplicados corretamente. Se esta indicação dos erros intermediários não for feita, os resultados dos testes de hipóteses serão incorretos. A adição do termo *Error(bloco:trat+bloco:mes)* ao modelo, permitirá à função *aov* calcular os erros correspondentes. Assim, será criado um erro de cada parcela e um terceiro, da interação tripla dos três fatores. A função *aov* ajustará apropriadamente a ordem dos fatores para serem testados com os erros apropriados, o que torna nosso trabalho mais fácil.

```
> library(agricolae) # carrega o pacote
> attach(sub)
```

```
> # modela primeiro sem a estrutura de três erros
> # artifício para aplicar o teste Tukey depois
> # A única forma encontrada para capturar os QME e GLE
> mod1 <- aov(alt~bloco + trat + bloco:trat+ mes +
+            bloco:mes + mes:trat)
> mm <- anova(mod1) # guarda anava no objeto mm
> mod.glb <- mm[5,1] # captura gleb
> mod.Eb <- mm[5,3]  # captura QMEb
> mod <- aov(alt~bloco + trat +  mes +
+            Error(bloco:trat+bloco:mes) + mes:trat)
> summary(mod)

Error: bloco:trat
          Df    Sum Sq   Mean Sq F value  Pr(>F)
bloco      1 0.0008000 0.0008000  6.8571 0.12012
trat       2 0.0175000 0.0087500 75.0000 0.01316 *
Residuals  2 0.0002333 0.0001167
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Error: bloco:mes
          Df   Sum Sq   Mean Sq F value    Pr(>F)
mes        2 0.060433 0.0302167    1813 0.0005513 ***
Residuals  2 0.000033 0.0000167
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Error: Within
          Df     Sum Sq    Mean Sq F value Pr(>F)
trat:mes   4 0.00016667 4.1667e-05     0.2 0.9259
Residuals  4 0.00083333 2.0833e-04

> HSD.test(alt, mes, mod.glb,mod.Eb, alpha = 0.05)

Study:

HSD Test for alt

Mean Square Error:  1.666667e-05

mes,  means

       alt    std.err replication
1 1.050000 0.01414214           6
2 1.128333 0.01470072           6
3 1.191667 0.01536591           6
```

```
alpha: 0.05 ; Df Error: 2
Critical Value of Studentized Range: 8.330783

Honestly Significant Difference: 0.01388464

Means with the same letter are not significantly different.

Groups, Treatments and means
a          3          1.191667
 b          2          1.128333
  c          1          1.05

> detach(sub)
> detach("package:agricolae") # libera o pacote
```

Se os níveis dos tratamentos fossem qualitativos, o que não é o caso desse exemplo, o comando *HSD.test(alt, trat, mod.gla,mod.Ea, alpha = 0.05)*, poderia ser utilizado. O quadrado médio e os graus de liberdade do erro A, devem ser apropriadas obtidos do objeto *mm*, criado para essa finalidade. Com esse comando, são requisitados o cálculo das médias de tratamento e a aplicação do teste de Tukey usando como erro o efeito de *bloco*× *trat*. Os testes de hipóteses sobre os efeitos dos fatores são aplicados corretamente, se for utilizada apropriadamente a função *aov* e a função *HSD.test* do pacote *agricolae*. Os resultados da análise de variância reorganizada está apresentada na Tabela 5.9.

Tabela 5.9. Análise da variação para o modelo de parcela subdividida no tempo.

FV	G.L.	SQ	QM	F	$Pr > F$
Bloco	1	0,00080000	0,00080000	6,86	0,1201
Trat	(2)	(0,01750000)	0,00875000	75,00	0,0132
RL	1	0,01687000	0,01687000	144,60	0,0068
Desvio	1	0,00062500	0,00062500	5,35	0,1468
Erro a	2	0,00023333	0,00011667		
Mês	2	0,06043333	0,03021667	1.813,00	0,0006
Erro b	2	0,00003333	0,00001667		
Trat*Mês	4	0,00016667	0,00004167	0,20	0,9259
Erro	4	0,00083333	0,00020833		
Total	17	0,08000000			

Ajustamos um modelo linear simples da variável resposta altura em função da adubação química utilizando a função *lm* e obtivemos o seguinte modelo: $\hat{\hat{Y}}_{i..} = 1{,}08583 + 0{,}00375A_i$, em que A_i é o i-ésimo nível do adubo químico. O coeficiente de determinação deve ser re-estimado por $R^2 = 0{,}01687/0{,}0175 = 0{,}964$. A análise de variância do modelo de regressão, apresentando o teste de falta de ajuste foi incorporado na Tabela 5.9. Nesse caso, aplicamos o teste de falta de ajuste que foi não significativo, um R^2 alto e o modelo de regressão com teste F significativo, ou seja, obtivemos resultados considerados ideais.

```
> # transforma trat em numérico e atribui a trat1
> attach(sub)
> trat1 <- as.numeric(levels(sub$trat)[sub$trat])
> trat2 <- as.factor(trat) # só por garantia
> # ajusta o modelo linear, embora os testes
> # e erros padrões estejam incorretos
> reg1 <- lm(alt~trat1)
> summary(reg1) # imprime o modelo ajustado

Call:
lm(formula = alt ~ trat1)

Residuals:
      Min        1Q    Median        3Q       Max
-0.085833 -0.065208  0.005417  0.057917  0.086667

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.085833   0.023409  46.386   <2e-16 ***
trat1       0.003750   0.001813   2.068   0.0552 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.06281 on 16 degrees of freedom
Multiple R-squared: 0.2109,        Adjusted R-squared: 0.1616
F-statistic: 4.277 on 1 and 16 DF,  p-value: 0.05519

> # ajusta o modelo linear, agora para testar a RL e o Desvio trat^2
> reg2 <- aov(alt~bloco+trat1+I(trat1*trat1)+Error(bloco:trat2))
> summary(reg2) # imprime a análise (considerar só primeira parte)

Error: bloco:trat2
                Df    Sum Sq   Mean Sq  F value
bloco            1 0.0008000 0.0008000   6.8571
trat1            1 0.0168750 0.0168750 144.6429
```

```
I(trat1 * trat1)  1 0.0006250 0.0006250   5.3571
Residuals         2 0.0002333 0.0001167
                   Pr(>F)
bloco             0.120117
trat1             0.006843 **
I(trat1 * trat1) 0.146680
Residuals
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Error: Within
          Df   Sum Sq   Mean Sq F value Pr(>F)
Residuals 12 0.061467 0.0051222
```

```
> detach(sub)
```

Consideramos ainda que os níveis de mês são qualitativos e não quantitativos e aplicamos o teste Tukey. Todas as médias diferiram entre si pelo teste de Tukey. Como comentado anteriormente, devemos selecionar o erro apropriado para realizarmos o teste de comparações múltiplas de Tukey. As maiores médias para a altura em relação ao mês, como era esperado, estavam associadas ao 3, seguidas pelo 2 e finalmente pelo 1.

5.5 Modelos lineares multivariados

Na pesquisa agropecuária e de outras áreas é comum as situações em que várias variáveis são mensuradas simultaneamente. Os fenômenos estudados respondem aos tratamentos não em relação a apenas uma variável, mas sofrem o efeito no conjunto total de variáveis associadas a eles. Nesses casos, duas aproximações podem ser feitas: a primeira, utiliza uma análise marginal de cada variável, produzindo uma grande quantidade de informações, além de não levar em consideração a estrutura de covariância entre as variáveis; a segunda, utiliza a análise multivariada, que considera a estrutura de covariância entre as variáveis sob estudo, conduzindo a resultados mais poderosos.

Para ilustrar como são realizados os ajustes dos modelos e obtidas as somas de quadrados e de produtos, vamos utilizar um modelo linear multivariado com m parâmetros associados a cada uma das p variáveis respostas. Diferentemente dos casos univariados, em que são calculadas apenas somas de quadrados, nos modelos lineares multivariados são obtidas somas de pro-

dutos entre as variáveis. Isso é feito para cada fonte de variação (ou fator) do modelo. As somas de quadrados e produtos são apresentadas em uma matriz $p \times p$ e os testes de hipóteses envolvem estatísticas que são relacionadas com razões de determinantes ou de funções dos autovalores das matrizes de somas de quadrados e produtos associadas à hipótese e ao erro.

Os modelos lineares multivariados podem ser escritos matricialmente por:

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{5.5.1}$$

em que $\boldsymbol{Y}$ é matriz das variáveis respostas com n linhas (observações) e p colunas (variáveis), $\boldsymbol{X}$ é a matriz de modelo com n linhas e m colunas (parâmetros do modelo), $\boldsymbol{\beta}$ é a matriz de parâmetros com m linhas e p colunas e $\boldsymbol{\epsilon}$ é a matriz de erros $n \times p$ supostos normal multivariados e independentemente distribuídos com média $\underset{\sim}{0}$ e covariância comum $\boldsymbol{\Sigma}$.

A solução de mínimos quadrados é obtida por:

$$\boldsymbol{\beta}^* = (\boldsymbol{X}'\boldsymbol{X})^g \boldsymbol{X}'\boldsymbol{Y} \tag{5.5.2}$$

A matriz de somas de quadrados e produtos do modelo especiicado em 5.5.1 é dada por:

$$H = R(\boldsymbol{\beta}) = \boldsymbol{\beta}^{*\prime}\boldsymbol{X}'\boldsymbol{Y} \tag{5.5.3}$$

A matriz de soma de quadrados e produtos do resíduo $\boldsymbol{E}$ é obtida por $\boldsymbol{E} = \boldsymbol{Y}'\boldsymbol{Y} - \boldsymbol{\beta}^{*\prime}\boldsymbol{X}'\boldsymbol{Y}$. Mediante reduções de modelos hierárquicos, aplicamos as expressões (5.5.2) e (5.5.3) para estimarmos as matrizes de somas de quadrados e produtos dos efeitos de um modelo ajustados para os efeitos de outros, da mesma forma como é feito para regressão e para modelos univariados. A diferença nesse caso é o resultado matricial obtido. Não apresentaremos nenhum outro resultado adicional nesse material, devido às dificuldades teóricas relacionadas a esse assunto.

Vamos ilustrar a utilização da *aov* e da *manova* para realizarmos uma análise de variância multivariada, com os respectivos testes de hipóteses.

O exemplo que vamos utilizar refere-se a três métodos de ensino diferentes aplicados a uma determinada série do ensino básico. As notas de duas disciplinas em cada método de ensino foram anotadas em amostras de diferentes tamanhos. O programa R, para testarmos a igualdade do vetor de médias dos três métodos de ensino (A, B e C), juntamente com os comandos das funções *manova* e *aov* são apresentados na seqüência.

```
> # lê o arquivo mult.txt e o atribui ao objeto mult
> mult  <- read.table("C:/daniel/Cursos/RCursoTeX/mult.txt",
+                     header=TRUE)
> mult # imprime o data frame

   met n1 n2
1    A 69 75
2    A 69 70
3    A 71 73
4    A 78 82
5    A 79 81
6    A 73 75
7    B 69 70
8    B 68 74
9    B 75 80
10   B 78 85
11   B 68 68
12   B 63 68
13   B 72 74
14   B 63 66
15   B 71 76
16   B 72 78
17   B 71 73
18   B 70 73
19   B 56 59
20   B 77 83
21   C 72 79
22   C 64 65
23   C 74 74
24   C 72 75
25   C 82 84
26   C 69 68
27   C 76 76
28   C 68 65
```

```
29   C 78 79
30   C 70 71
31   C 60 61
```

```
> mult$met <- as.factor(mult$met)
> attach(mult)
> mult1<-aov(cbind(n1,n2)~met)
> # análises univariadas
> summary(mult1)

 Response n1 :
            Df Sum Sq Mean Sq F value Pr(>F)
met          2  60.61  30.303  0.9095 0.4143
Residuals   28 932.88  33.317


 Response n2 :
            Df  Sum Sq Mean Sq F value Pr(>F)
met          2   49.74  24.868  0.5598 0.5776
Residuals   28 1243.94  44.426

> # análises multivariadas com exemplo de dois critérios
> anova(mult1,test = "Hotelling-Lawley")

Analysis of Variance Table

            Df Hotelling-Lawley approx F num Df den Df
(Intercept)  1          170.383  2300.17      2     27
met          2            0.469     3.05      4     52
Residuals   28
             Pr(>F)
(Intercept) < 2e-16 ***
met         0.02478 *
Residuals
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> anova(mult1,test = "Wilks")

Analysis of Variance Table

            Df   Wilks approx F num Df den Df  Pr(>F)
(Intercept)  1 0.00583  2300.17      2     27 < 2e-16
met          2 0.67310     2.95      4     54 0.02793
Residuals   28
```

```
(Intercept) ***
met         *
Residuals
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> # análises multivariadas para ilustrar a manova
> # utiliza todas os critérios multivariados
> mult2 <- manova(cbind(n1,n2)~met)
> summary(mult2,test = "Pillai")

          Df  Pillai approx F num Df den Df  Pr(>F)
met        2 0.33798    2.847      4     56 0.03215 *
Residuals 28
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> summary(mult2,test = "Wilks")

          Df  Wilks approx F num Df den Df  Pr(>F)
met        2 0.6731   2.9548      4     54 0.02793 *
Residuals 28
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> summary(mult2,test = "Hotelling-Lawley")

          Df Hotelling-Lawley approx F num Df den Df
met        2          0.46919   3.0497      4     52
Residuals 28
           Pr(>F)
met       0.02478 *
Residuals
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> summary(mult2,test = "Roy")

          Df     Roy approx F num Df den Df   Pr(>F)
met        2 0.43098   6.0337      2     28 0.006624
Residuals 28

met       **
```

```
Residuals
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

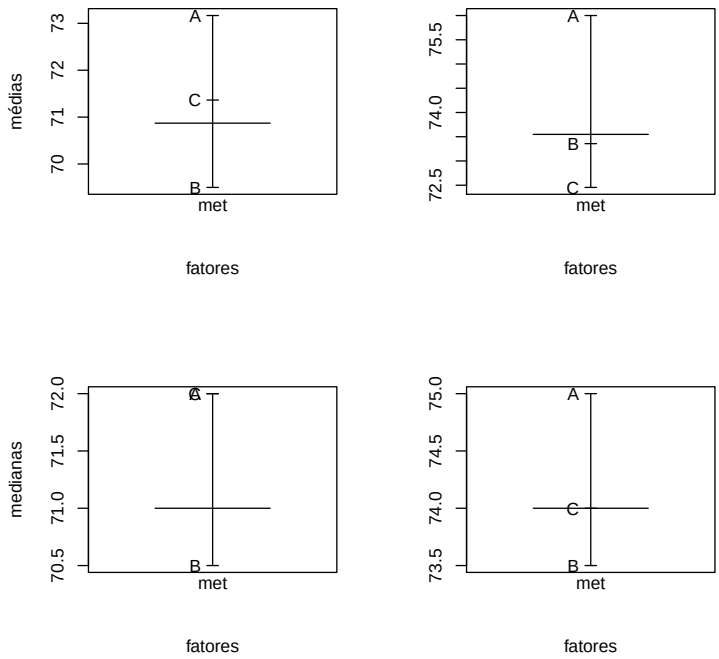
```
> # obtenção de E, H e autovalores
> mult.summ <- summary(mult2)
> mult.summ$SS
```

```
$met
         n1       n2
n1 60.60508 31.51173
n2 31.51173 49.73586

$Residuals
          n1       n2
n1  932.8788 1018.682
n2 1018.6818 1243.942
```

```
> mult.summ$Eigenvalues
```

```
         [,1]       [,2]
met 0.4309803 0.03821194
```

```
> detach(mult)
> # gráficos exploratórios
> par(mfrow = c(2,2))
> plot.design(mult,xlab="fatores",
+             ylab="médias")
> plot.design(mult, fun = median,xlab="fatores",
+             ylab="medianas")
> par(mfrow = c(1,1))
```



Os principais resultados dessa análise foram sumariados na seqüência. Inicialmente foram obtidas as análises de variâncias para cada uma das notas das matérias, para testarmos o efeito marginal de métodos. Os resultados para a variável 1 estão apresentados na Tabela 5.10. Observamos que não foram detectadas diferenças significativas entre os métodos.

Tabela 5.10. Análise da variação para nota da disciplina 1 para testar a hipótese de igualdade dos efeitos dos métodos de ensino.

FV	G.L.	SQ	QM	F	$Pr > F$
Métodos	2	60,6051	30,3025	0,91	0,4143
Erro	28	932,8788	33,3171		
Tratamento	30	993,4839			

Os resultados para a variável 2 estão apresentados na Tabela 5.11. Da mesma forma que ocorreu para a variável 1, observamos que não foram detectadas diferenças significativas entre os métodos.

Os objetos *mult.summ$SS* e *mult.summ$Eigenvalues* representam as matrizes de somas de quadrados e produtos do resíduo e de métodos e os autovalores, respectivamente. A partir do objeto referente a matriz de so-

Tabela 5.11. Análise da variação para nota da disciplina 2 para testar a hipótese de igualdade dos efeitos dos métodos de ensino.

FV	G.L.	SQ	QM	F	$Pr > F$
Métodos	2	49,7359	24,8679	0,56	0,5776
Erro	28	1243,9416	44,4265		
Tratamento	30	1293,6774			

mas de quadrados e produtos podemos obter as estimativas das correlações parciais entre as variáveis ajustadas paras as fontes de variação do modelo. As matrizes de soma de quadrados e produtos são:

$$E = \begin{bmatrix} 932{,}8788 & 1018{,}6818 \\ 1018{,}6818 & 1243{,}9416 \end{bmatrix} \quad \text{e} \quad H = \begin{bmatrix} 60{,}6051 & 31{,}5117 \\ 31{,}5117 & 49{,}7359 \end{bmatrix}$$

A matriz de correlações parciais acompanhada das probabilidade para os testes de hipóteses $H_0 : \rho = 0$, foram obtidas da seguinte forma:

```
> df  <- df.residual(mult1) # GL resíduo
> E   <- mult.summ$SS$Residuals # SS&P do resíduo
> # matriz diagonal com  inverso desv.pad.
> DHalf <- diag(1/diag(E)^0.5)
> R <- DHalf%*%E%*%DHalf # matriz correlação
> corr <- R[lower.tri(R)] # captura as correlações
> # calcula o valor-p
> valor.p <- 2*(1-pt(abs(corr*(df-1)^0.5/(1-corr^2)^0.5),df-1))
> # retorna os valores-p para diagonal inferior de R
> # assim, diag superior as correlações e diag inferior
> # os valores-p
> R[lower.tri(R)] <- valor.p
> R # imprime a matriz R modificada

             [,1]      [,2]
[1,] 1.000000e+00 0.9456403
[2,] 1.065814e-14 1.0000000
```

O resultado final, apresentado o teste apropriado, é:

$$R = \begin{bmatrix} 1{,}0000 & 0{,}94564 \\ & < 0{,}0001 \\ 0{,}945640 & 1{,}0000 \\ < 0{,}0001 & \end{bmatrix}$$

Concluímos que as duas variáveis são altamente correlacionadas, eliminando-se o efeito dos métodos de ensino. Os testes de hipóteses multivariados sobre a igualdade do vetor de médias são feitos basicamente por 4 critérios distintos. O critério de Wilks é um deles e é um teste via razão de verossimilhanças. Muitos pesquisadores preferem tomar a decisão de rejeitar a hipótese nula quando pelo menos 3 dos 4 critérios apresentarem estimativas significativas das estatísticas dos testes. Outros preferem utilizar o critério de Wilks para tomar esta decisão. Para testarmos a hipótese nula, qualquer que seja a opção escolhida, os valores dessas estatísticas são convertidos para F, que é a distribuição utilizada para aproximar as distribuições exatas, difíceis de serem obtidas. Em alguns casos, dependendo do número de tratamentos e de variáveis, a estatística F resultante possui distribuição F exata. Os resultados do teste de hipótese de igualdade dos vetores de médias dos três métodos foram apresentados na Tabela 5.12. Todos os critérios apresentaram valores correspondentes de F significativos.

Tabela 5.12. Testes de hipóteses multivariados para a igualdade dos efeitos dos métodos de ensino.

Estatística	Estimativa	F	GL num.	GL den.	$Pr > F$
Wilks' Lambda	0,67310	2,95	4	54	0,0279
Pillai's Trace	0,33798	2,85	4	56	0,0322
Hotelling-Lawley Trace	0,46919	3,05	4	52	0,0248
Roy's Greatest Root	0,43098	6,03	2	28	0,0066

Uma outra observação que pode ser feita nesse exemplo, refere-se ao fato de os níveis de significância multivariados terem sido muito menores que os univariados, representando um dos casos clássicos em que os testes univariados falham em detectar alguma diferença entre os tratamentos, mas

os multivariados não. Esse fato provavelmente pode ser, em parte, explicado pela alta correlação parcial entre as variáveis respostas.

5.6 Exercícios

1. Utilizar dados balanceados resultantes de pesquisas desenvolvidas em sua área e realizar análises de variâncias utilizando a função *aov*. Aplicar os testes de médias, se os níveis forem qualitativos, ou ajustar modelos de superfície de resposta ou de regressão, se os níveis dos fatores forem quantitativos.

2. Em sua opinião, qual foi a vantagem de se utilizar uma modelagem multivariada para o exemplo desse capítulo que comparava três métodos de ensino em relação a análise de variância univariada. Você utilizaria análises multivariadas de variância em sua área profissional?

Capítulo 6

Análise de Variância para Dados Não-Balanceados

Muitas vezes precisamos realizar inferência sobre a igualdade de médias de um determinado fator. Se o conjunto de dados for não-balanceado, apresentando perdas de parcelas ou até mesmo de caselas devemos utilizar um tipo especial de análise de variância. A análise de variância nesse caso deve ser realizada por meio de métodos matriciais para lidarmos com o não-balanceamento dos dados. A partição da variação entre às observações associadas aos fatores, que são definidos pelo esquema de classificação dos dados experimentais, pode ser realizada de diferentes formas. Assim, diferentes hipóteses podem ser testadas a partir de um mesmo conjunto de dados.

A função *aov* é apropriada para conjuntos de dados que sejam balanceados. A *lm* nos permite analisar conjuntos de dados não-balanceados, incluindo casos extremos de desconexão. Nesse capítulo aplicaremos a função *lm* a conjuntos de dados não-balanceados e obteremos a análise de variância por intermédio do pacote *car*, utilizando a função *Anova*. Estudaremos três tipos de somas de quadrados que podem ser estimados nos modelos lineares não-balanceados. No caso de delineamentos balanceados, estas somas de quadrados, são todas iguais, não havendo diferenças nas hipóteses que são testadas, exceto se para a soma de quadrados tipo I for utilizada uma ordem em que um efeito de interação aparece antes dos efeitos principais ou de interações de menor ordem envolvendo os efeitos principais que compõem a interação.

A soma de quadrados tipo I refere-se à soma de quadrados seqüencial. Esta soma de quadrado é obtida com a redução no modelo de um fator por vez, na ordem inversa à de entrada dos fatores no modelo. Para ilustrar, vamos considerar um modelo com dois fatores (α, β) e interação (δ) dado por:

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \delta_{ij} + \epsilon_{ijk} \tag{6.0.1}$$

em que Y_{ijk} é o valor observado da variável resposta, μ é a constante geral, α_i é o efeito do i-ésimo nível do fator α, β_j é o efeito do j-ésimo nível do fator β, δ_{ij} é o efeito da interação entre o i-ésimo nível do fator α com o j-ésimo nível do fator β e ϵ_{ijk} é o efeito do erro experimental suposto normal e independentemente distribuído com média 0 e variância comum σ^2.

As somas de quadrados tipo I, II e III para os efeitos do modelo da equação (6.0.1) estão apresentadas na Tabela 6.1.

Tabela 6.1. Tipos de somas de quadrados de um modelo de análise de variância contendo dois fatores α e β e interação δ.

FV	SQ Tipo I	SQ Tipo II	SQ Tipo III
α	$R(\alpha/\mu)$	$R(\alpha/\mu,\beta)$	$R(\alpha^*/\mu^*,\beta^*,\delta^*)$
β	$R(\beta/\mu,\alpha)$	$R(\beta/\mu,\alpha)$	$R(\beta^*/\mu^*,\alpha^*,\delta^*)$
δ	$R(\delta/\mu,\alpha,\beta)$	$R(\delta/\mu,\alpha,\beta)$	$R(\delta^*/\mu^*,\alpha^*,\beta^*)$

* indica parâmetros obtidos sob o uso de restrição paramétrica.

A soma de quadrado tipo II para um dado fator é obtida ajustando esta fonte de variação para todas as outras que não envolvam o efeito em questão. Assim, a soma de quadrados para α, não pode ser ajustada para a fonte de variação δ, uma vez que esse último efeito envolve o efeito de α, por ser a interação desse fator com β. A soma de quadrados tipo III, ou parcial, refere-se ao ajuste de cada fator para todos os demais efeitos do modelo sob restrição paramétrica do tipo soma de efeitos igual a zero.

As somas de quadrados do tipo I são dependentes da ordem de entrada dos fatores no modelo. As somas de quadrados do tipo II e III não dependem desta ordem de entrada. Como dissemos, os três tipos são iguais quando os dados são balanceados, tomando-se o cuidado de entrar com uma ordem dos efeitos no modelo, em que os fatores principais vêm antes das interações

de que participam. Caso contrário a soma de quadrados do tipo I irá diferir das demais.

A função *lm* é um dos procedimentos do R utilizados para lidar com estes casos não-balanceados. Ao objeto que recebeu o resultado dessa função devemos submetê-lo à função *Anova* para obtermos as somas de quadrados dos tipos II e III. Devemos salientar, que as nomenclaturas de somas de quadrados do tipo II e III, segundo as palavras de John Fox, criador e mantenedor do pacote car, foram emprestadas do SAS. A soma de quadrados do tipo II é exatamente equivalente a do SAS, mas a do tipo III, é obtida de forma diferente e não corresponde precisamente aquela obtida desse programa. Então, devemos nos atentar para o fato de que os resultados obtidos pelo SAS e pelo R, pacote car, para esse tipo de soma de quadrados não são necessariamente idênticos. Segundo Bill Venables, devemos nos preocupar com o tipo de hipóteses que queremos testar e não com o tipo de somas de quadrados que iremos utilizar. As sintaxes desses procedimentos são similares a da função *aov*.

Vamos utilizar alguns dos conjuntos de dados anteriores, provocando artificialmente algum tipo de não-balanceamento em algumas ocasiões e em outras utilizando os dados balanceados, para ilustrarmos as principais peculiaridades das funções *lm* e *Anova*. Infelizmente, para alguns casos de caselas vazias a função *Anova* retorna mensagens de erro e não obtém as somas de quadrados dos tipos II e III. Felizmente, podemos obter ao menos as somas de quadrados do tipo II, utilizando tanto o *aov* para obter as reduções entre dois modelos e quanto o conceito de somas de quadrados do tipo II, para escolhermos os modelos apropriados.

6.1 Delineamento Inteiramente Casualizado

No modelo inteiramente casualizado com um fator (equação 5.2.1), vamos considerar o mesmo conjunto de dados apresentados na Tabela 5.1, para ilustrarmos o uso de contrastes no *lm*. A variável resposta é o ganho de peso dos animais submetidos a quatro rações diferentes. Um delineamento inteiramente casualizado com 5 repetições foi utilizado. Vamos considerar a estrutura para os níveis dos tratamentos, estabelecida por diferentes firmas produtoras das rações e diferentes fontes de proteínas. Assim, a ração 1 é proveniente da firma A e as rações 2, 3 e 4 da firma B. A ração 2 possui

proteína animal na sua composição e as rações 3 e 4, proteína de origem vegetal. As rações 3 e 4 diferem quanto ao nível de energia que possuem.

Devido aos tratamentos serem estruturados é natural que façamos contrastes sugeridos por essa estrutura. Um conjunto de contrastes ortogonais que estaríamos interessados em testar é: 1 vs 2, 3, e 4, contrastando firma A contra firma B, 2 vs 3 e 4, contrastando proteína animal contra proteína vegetal e finalmente 3 vs 4, contrastando os níveis de energia. Como temos 3 graus de liberdade e 3 contrastes ortogonais, então, estamos fazendo uma decomposição ortogonal das somas de quadrados de tratamento. Para estimar os efeitos dos contrastes, testarmos as hipóteses de que seus efeitos são nulos e obtermos os intervalos de confiança, podemos utilizar a função *fit.contrast* do pacote *gregmisc*. O programa resultante, para estimarmos e testarmos os efeitos dos contrastes, é apresentado na seqüência.

```
> # lê o arquivo pimen43.txt e o atribui ao objeto pimen43
> pimen <- read.table("C:/daniel/Cursos/RCursoTeX/pimen43.txt",
+                         header=TRUE)
> pimen$trat <-   as.factor(pimen43$trat)
> library(gregmisc)
> attach(pimen)
> # cria a matrix de contrastes
> C <- rbind(" 1 vs 2, 3 e 4"= c(3,-1,-1,-1)/3,
+            " 2 vs 3 e 4"   = c(0, 2,-1,-1)/2,
+            " 3 vs 4"       = c(0, 0, 1,-1))
> C # mostra a matrix
               [,1]       [,2]       [,3]       [,4]
 1 vs 2, 3 e 4    1 -0.3333333 -0.3333333 -0.3333333
 2 vs 3 e 4       0  1.0000000 -0.5000000 -0.5000000
 3 vs 4           0  0.0000000  1.0000000 -1.0000000
> pimen.aov <- aov(gp ~ trat) # realiza a anava
> anova(pimen.aov) # mostra a anava
Analysis of Variance Table

Response: gp
          Df  Sum Sq Mean Sq F value  Pr(>F)
trat       3  823.75  274.58  3.9939 0.02671 *
Residuals 16 1100.00   68.75
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> # estima por ponto e intervalo
> # e testa os contrastes
```

```
> fit.contrast(pimen.aov, "trat",
+              C, conf=0.95 )
                    Estimate Std. Error   t value
trat 1 vs 2, 3 e 4       -5   4.281744 -1.167748
trat 2 vs 3 e 4          12   4.541476  2.642313
trat 3 vs 4              10   5.244044  1.906925
                     Pr(>|t|)   lower CI  upper CI
trat 1 vs 2, 3 e 4 0.26001588 -14.076892  4.076892
trat 2 vs 3 e 4    0.01774519   2.372502 21.627498
trat 3 vs 4        0.07464957  -1.116877 21.116877
> detach(pimen)
> detach("package:gregmisc") # libera memória do pacote
```

```
/* Exemplo da utilização do Proc GLM para testarmos contrastes em um DIC balanceado*/
data dic;
input racoes gp @@;
cards;
 1 35   1 19   1 31   1 15
 1 30   2 40   2 35   2 46
 2 41   2 33   3 39   3 27
 3 20   3 29   3 45   4 27
 4 12   4 13   4 28   4 30
  ;
proc glm;
    class racoes;
    model gp=racoes;
    means racoes / tukey alpha = 0.05 lines;
    lsmeans racoes / pdiff adjust = tukey;
    lsmeans racoes / pdiff = control("1") adjust = dunnett;
    contrast "1 vs 2, 3 e 4" racoes 3 -1 -1 -1;
    contrast "2 vs 3 e 4" racoes 0 2 -1 -1;
    contrast "3 vs 4" racoes 0 0 1 -1;
    estimate "1 vs 2, 3 e 4" racoes 3 -1 -1 -1/divisor=3;
    estimate "2 vs 3 e 4" racoes 0 2 -1 -1/divisor=2;
    estimate "3 vs 4" racoes 0 0 1 -1;
run; quit; /* fim do programa */
```

Utilizamos os comandos *means* e *lsmeans*, nesse exemplo, simplesmente para ilustrarmos as sintaxes, pois como os tratamentos são qualitativos estruturados, devemos utilizar contrastes para otimizarmos as comparações

realizadas. Ilustramos o uso de um teste de comparações múltiplas sobre médias não ajustadas e ajustadas e o teste de Dunnett bilateral, utilizando a ração 1 como controle. O objetivo foi de apresentar a sintaxe dos comandos para podermos obter médias ajustadas e para aplicarmos os testes de comparações múltiplas e de Dunnett. Todos estes resultados devem ser ignorados nesse exemplo e somente os resultados dos contrastes e das estimativas devem ser considerados. Somente o contraste entre os tipos de origem das proteínas na formulação das rações da firma B foi significativo ($P < 0{,}0177$). Como a estimativa é positiva, podemos afirmar que em média teremos um ganho superior em 12 kg/animal/período, se utilizarmos ração com proteína animal em vez de proteína de origem vegetal. Não solicitamos somas de quadrados de nenhum tipo, mas o padrão do *glm* é apresentar tanto a soma de quadrados do tipo I, quanto à do tipo III. Nos modelos lineares para os quais temos apenas um efeito, além do intercepto e do erro, não faz sentido diferenciar as somas de quadrados, pois todas elas são idênticas. Nesse caso, a soma de quadrados do tipo I para rações foi de 823,75, sendo o mesmo resultado obtido para as somas de quadrados dos tipos II e III.

Uma outra vantagem do *proc glm* é obter predições para os valores da variável resposta, que nesse caso, são as médias de caselas. Adicionalmente os valores residuais são preditos. Para isso basta substituir o comando *<model gp=racoes;>* por *<model gp=racoes/p;>*. Esse comando, além destas estimativas e predições, fornece a estatística de Durbin-Watson, para realizarmos testes de autocorrelação. Outra estimativa, que utilizamos com freqüência na análise de dados não-balanceados, é a da média ajustada. Em vez de utilizarmos o comando *<means racoes / tukey alpha=0.05 lines;>* podemos utilizar o comando *<lsmeans racoes / pdiff adjust=tukey;>*. Nesse caso, o R calculará os valores-p das comparações entre as lsmeans utilizando o procedimento ajustado de Tukey. Para comparação com o controle fazemos *pdiff = control('trat')* com o comando *adjust = opção*. A opção que devemos utilizar é a do teste de Dunnett, determinada por *dunnett*. Apesar de o natural ser a escolha do comando *adjust=dunnett*, podemos escolher outras formas de ajustes como Bon, Sidak, Scheffe, entre outras. É claro que para um delineamento inteiramente casualizado com um fator balanceado ou não-balanceado não existem diferenças entre as médias ajustadas e não-ajustadas. Mas, entre os testes utilizando as médias ajustadas e as mé-

dias não ajustadas existem diferenças nos casos não balanceados. Devemos optar por utilizar as médias ajustadas solicitando o teste apropriado.

6.2 Estrutura Cruzada de Tratamentos

Para ilustramos a análise de modelos mais complexos, onde temos conjuntos de dados não-balanceados, vamos retornar ao exemplo apresentado na seção 5.3, simulando algumas perdas de parcelas. Com esse exemplo, vamos mostrar as dificuldades existentes para realizar uma análise de dados não-balanceados e as diferenças entre os três tipos de somas de quadrados que estamos considerando. Posteriormente consideraremos, ainda, uma análise de covariância. Os dados apresentados na seção 5.3 com algumas perdas de unidades experimentais simuladas e o modelo da equação (5.3.1) foram utilizados. Temos um delineamento em blocos casualizados com 4 repetições e 2 fatores (adubo mineral e torta de filtro) com 2 níveis cada.

O programa ilustrando a análise de variância e os principais resultados alcançados estão apresentados na seqüência. Vamos destacar o uso da opção *slice* do comando *lsmeans* nesse programa, a qual possibilita que seja realizado o desdobramento de interações entre efeitos do modelo.

```
/* Exemplo da utilização do proc GLM para uma estrutura fatorial de tratamentos em
um DBC e não-balanceada*/
data Fat;
input A T bloco prod;
cards;
 0 10 1 18.0
20 10 1 20.6
 0 20 1 19.6
20 20 1 19.2
 0 10 2  8.6
 0 20 2 15.0
20 20 2 19.6
 0 10 3  9.4
20 10 3 18.6
 0 20 3 14.6
20 20 3 18.4
 0 10 4 11.4
 0 20 4 15.8
20 20 4 20.2
```

```
;
proc glm data=fat;
    class A T bloco;
    model prod = bloco A T A*T/ss1 ss2 ss3;
    means A T/Tukey;
    lsmeans A T/pdiff adjust=Tukey;
    lsmeans A*T/slice=A slice=T;
run; quit;
```

Inicialmente, observamos que uma análise de variação contendo as fontes de variação de modelo e de resíduos foi obtida. Estes resultados estão apresentados na Tabela 6.2. Na Tabela 6.3 apresentamos os três tipos de somas de quadrados solicitadas (I, II e III). Podemos observar um efeito significativo de A e de T para os três tipos de somas de quadrados, exceto para o efeito da torta de filtro com a soma de quadrado do tipo III. Em todos os casos (I, II e III) tivemos um efeito não significativo da interação, sendo as somas de quadrados tipo I, II e III para esse efeito iguais.

Tabela 6.2. Análise da variação para o modelo fatorial (2 fatores) em um delineamento de blocos casualizados, destacando-se as fontes de variação de modelo e erro.

FV	G.L.	SQ	QM	F	$Pr > F$
Modelo	6	180,89	30,15	6,75	0,0120
Erro	7	31,29	4,47		
Total	13	212,17			

CV = 12,92% $\bar{Y}_{...} = 16{,}36$

Houve uma diferença muito grande entre algumas das somas de quadrados, sendo que no efeito da adubação mineral, isto foi mais pronunciado. Era esperado, por exemplo, que as somas de quadrados do tipo I e do tipo II para efeito da torta de filtro fossem iguais, considerando a ordem que os fatores entraram no modelo. Dessa forma, podemos observar a importância de saber exatamente o que testamos, para interpretar adequadamente as saídas do *proc glm*. Detalhes técnicos a respeito das hipóteses associadas a estas somas de quadrados podem ser obtidos em publicações especializadas.

Se observarmos as saídas do R, podemos verificar que existem diferenças entre as médias ajustadas e não-ajustadas, destacando-se a importância de

Tabela 6.3. Resumo da análise da variação para o modelo fatorial (2 fatores) em um delineamento de blocos casualizados, destacando as somas de quadrados tipo I, II e III e as significâncias correspondentes.

FV	G.L.	SQ I	SQ II	SQ III
Bloco	3	53,1543ns	42,7233ns	42,7233ns
A	1	88,7520**	66,9780**	77,0133**
T	1	27,3780*	27,3780*	17,7633ns
A*T	1	11,6033ns	11,6033ns	11,6033ns

*, ** e ns: significativo a 5, 1% e não significativo, respectivamente.

utilizar o comando adequado para o caso balanceado. Nesse exemplo observamos que tanto para torta de filtro, como para a adubação mineral, obtivemos diferenças significativas para as médias. No entanto, quando utilizamos o teste com correção de Tukey sobre as médias ajustadas, somente detectamos diferenças significativas para adubo mineral, mas não para torta de filtro.

Finalmente o comando *slice* nos possibilita obter a análise do desdobramento da interação $A * T$. Solicitamos os dois tipos de desdobramento: o de A dentro dos níveis de T e o de T fixados os níveis de A. Nenhum destes dois casos serão apresentados, pois a interação foi não significativa. Assim, recomendamos utilizar a maior dose de adubo mineral (teste marginal significativo) e a menor porcentagem de torta de filtro (teste marginal não significativo).

Reiteramos que as somas de quadrados do tipo I são afetadas pela ordem dos efeitos na especificação do modelo. Podemos ver claramente que se alterarmos esta ordem, teremos diferentes somas de quadrados do tipo I, mas as mesmas somas de quadrados dos tipos II e III obtidas anteriormente. O caso mais crítico desta alteração ocorre quando colocamos o efeito da interação dos fatores antes dos efeitos principais. Como o espaço paramétrico da interação contém os espaços paramétricos dos efeitos principais, teremos resultados nulos para os graus de liberdade e somas de quadrados associados. O leitor é conclamado a verificar esse resultado para o modelo em questão.

Alguns outros aspectos interessantes da análise merecem destaques. Como todos os procedimentos são realizados por meio de álgebra matricial e

vetorial, podemos solicitar a matriz inversa, a matriz $X'X$, valores preditos, solução mínimos de quadrados, entre outras opções. Para isso bastaria substituir o comando <model prod = bloco A T A*T/ss1 ss2 ss3;> por <model prod = bloco A T A*T/ss1 ss2 ss3 p solution XPX I;>.

Outra grande vantagem do *proc glm* é a possibilidade de realizarmos análises de regressão. Um fator omitido do comando *class* será considerado variável regressora e não variável classificatória. Assim, temos a possibilidade de realizar análises de covariância. A análise de covariância ocorre quando temos variáveis classificatórias (fatores qualitativos) e variáveis regressoras (fatores quantitativos) no mesmo modelo. Em geral estas covariáveis devem ser mensuradas em todas as unidades experimentais e não devem ser influenciadas pelo tratamento. Por exemplo, se estamos testando diferentes cultivares, utilizar o estande final como covariável, pode não ser uma boa estratégia. Isso porque pode existir um efeito de cultivares no estande final, ou seja, o efeito de estande é influenciado pelo efeito de cultivares. Assim, uma análise como essa vai produzir um ajuste do efeito de cultivar pelo efeito de estande. Como os dois efeitos podem estar relacionados, como acabamos de discutir, teremos o efeito de cultivar ajustado, de forma indireta, para o próprio efeito de cultivar. Assim, devemos utilizar covariáveis que não sejam influenciadas pelos tratamentos. Nesse caso, poderíamos, por exemplo, ter tomado medidas da fertilidade do solo em cada parcela experimental, antes de as cultivares terem sido semeadas. Estas variáveis de fertilidade poderiam ser utilizadas como covariáveis.

Nesse exemplo fatorial foi simulada a avaliação de uma covariável em cada parcela, para podermos ilustrar uma análise de covariância. Assim, em cada parcela experimental foi avaliado o teor de nitrogênio. Uma amostra de cada unidade foi coletada e os níveis de nitrogênio do solo foram mensurados, antes da implantação dos tratamentos, correspondentes ao adubo mineral e a torta de filtro. Um aspecto da análise de covariância que empiricamente podemos mencionar, refere-se ao fato de que ao utilizarmos uma covariável e ajustarmos o efeito de tratamentos para essa covariável, estaríamos fazendo algo semelhante a ter um experimento cujas condições iniciais seriam homogêneas para os níveis desta covariável. Assim, é como se indiretamente estivéssemos utilizando um controle local.

No exemplo que se segue apresentamos a análise de covariância utilizando como covariável os níveis de nitrogênio nas unidades experimentais

mensurados anteriormente a implantação do experimento. A especificação de uma covariável no modelo é feita de maneira bastante simples. Para isso omitimos no comando *class* a covariável, mas a introduzimos no comando *model*. O *proc glm* irá reconhecer a variável omitida como uma variável regressora e o comando *lsmeans* irá ajustar as médias dos fatores para a covariável ou covariáveis presentes no modelo. O programa R, ilustrativo deste caso, é dado por:

```
/* Exemplo da utilização do proc GLM para uma estrutura fatorial dos tratamentos com
covariável em um DBC não-balanceado*/
data Fat;
input A T bloco prod N;
cards;
 0 10 1 18.0 3
20 10 1 20.6 4
 0 20 1 19.6 5
 0 10 2  8.6 3
 0 20 2 15.0 4
20 20 2 19.6 4
 0 10 3  9.4 6
20 10 3 18.6 5
 0 20 3 14.6 2
20 20 3 18.4 7
 0 10 4 11.4 4
 0 20 4 15.8 3
20 20 4 20.2 3
;
proc glm data=fat;
    class A T bloco;
    model prod = bloco A T A*T N/solution ss1 ss2 ss3;
    means A T/Tukey;
    lsmeans A T/pdiff adjust=Tukey;
    lsmeans A*T/slice=A slice=T;
run; quit;
```

Se realizarmos uma análise de variância com e sem a covariável podemos observar que os resultados para esse exemplo apresentam uma ligeira diferença nas somas de quadrados dos dois modelos. É claro que a soma de quadrados do tipo I não foi afetada, pois a covariável apareceu após todos

os demais efeitos do modelo. A opção *solution* permitiu que fosse apresentada a solução de mínimos quadrados. A covariável foi único efeito do modelo cuja estimativa era não viesada. As demais conclusões são similares às já apresentadas anteriormente para esse modelo de análise de variação.

6.3 Modelos Com Mais de Um Erro

Para analisarmos experimentos mais complexos, contendo mais de um erro e em estruturas não balanceadas, devemos definir quais tipos de somas de quadrados desejamos utilizar, tanto para o tratamento quanto para o resíduo. Além disso, temos que especificar quais são os testadores das fontes de variação do modelo e também qual tipo de soma de quadrados deve ser utilizada para realizar o teste de interesse. Vamos ilustrar esse tipo de análise considerando modelos que contenham mais de um erro, a partir do mesmo exemplo de parcela subdividida no tempo, apresentado na seção 5.4. Vamos provocar artificialmente um desbalanceamento no conjunto original de dados para ilustrarmos a análise almejada. Um adubo mineral foi utilizado como fator principal, onde desejávamos comparar seus três níveis 0, 10 e 20 kg/ha. Estas três dosagens foram submetidas a um delineamento em blocos completos casualizados com 2 repetições. O interesse focava o crescimento das plantas ao longo do tempo. Assim, foram avaliadas as alturas das plantas durante 3 meses consecutivos. O modelo estatístico para esse experimento é dado por:

$$Y_{ijk} = \mu + \alpha_i + \beta_j + \epsilon_{ij} + \gamma_k + \epsilon_{jk} + \delta_{ik} + \epsilon_{ijk} \qquad (6.3.1)$$

em que Y_{ijk} é a observação da altura das plantas em metros, μ é a constante geral do modelo, α_i é o efeito do i-ésimo nível da adubação química, β_j é o efeito do j-ésimo bloco, ϵ_{ij} é o efeito do erro experimental entre a i-ésima dose e o j-ésimo bloco, γ_k é o efeito do k-ésimo mês, ϵ_{jk} é efeito do erro experimental do j-ésimo bloco com o k-ésimo mês, δ_{ik} é o efeito da interação entre a i-ésima dose de adubo químico com o k-ésimo mês e ϵ_{ijk} é o erro experimental entre a i-ésima dose, j-ésimo bloco e k-ésimo mês.

O programa R contendo os dados experimentais modificados artificialmente para se tornarem não balanceado e a sintaxe para especificar os erros do modelo e determinar os testes corretos com o tipo de soma de quadrados

pretendida é apresentado na seqüência. O comando *test* deve ser utilizado e em suas opções devemos nos preocupar em indicar o tipo de soma de quadrados que utilizaremos. O programa resultante é dado por:

```
/* Programa para realizar análise de variância de um modelo contendo múltiplos erros.
O modelo escolhido foi o de parcela subdividida no tempo com dados não-balanceados.*/
data sub;
input bloco trat mes alt;
cards;
1   0 1 1.00
1 10 1 1.05
1 20 1 1.08
2 10 1 1.06
2 20 1 1.09
1   0 2 1.10
1 10 2 1.12
1 20 2 1.14
2   0 2 1.08
2 10 2 1.15
2 20 2 1.18
1   0 3 1.14
1 10 3 1.20
1 20 3 1.22
2 10 3 1.21
2 20 3 1.23
;
proc glm data=sub;
    class bloco trat mes;
    model alt = bloco trat bloco*trat mes bloco*mes mes*trat/ss1 ss2 ss3;
    test h=bloco trat e=bloco*trat / htype = 3 etype = 3;
    test h=mes e=bloco*mes /htype = 3 etype = 3;
    lsmeans trat/e=bloco*trat etype = 3 stderr;
    lsmeans mes/e=bloco*mes etype = 3 pdiff stderr adjust=Tukey;
    lsmeans trat*mes/ etype = 3 stderr slice = trat slice = mes;
run; quit;
```

Nessa análise podemos destacar que os testes são inicialmente realizados utilizando o erro do modelo (erro C) como testador. Somente com o uso do comando *test* é que esse problema foi corrigido. Assim, o teste para bloco e para tratamento foi realizado com o erro A (bloco*trat) e o efeito

de mês foi testado com erro B (bloco*mes). No comando *<test h=bloco trat e=bloco*trat / htype = 3 etype = 3;>* especificamos que iríamos utilizar as somas de quadrados do tipo III para tratamento e bloco e também para o resíduo. Comando similar é utilizado para o teste do efeito relativo a mês.

Os comandos solicitando as médias ajustadas de tratamento e de mês são acrescidos das opções para que sejam estipulados o erro e o tipo de somas de quadrados que serão utilizados. Também possibilitam obtermos os erros padrões dos efeitos e no caso de efeitos qualitativos, permitem realizarmos testes de comparações múltiplas com ajuste das probabilidade pelo método de Tukey-Kramer. No caso de efeitos de interação, permitem que sejam realizados desdobramentos com o comando *slice*. O problema do comando *<lsmeans trat*mes/ etype = 3 stderr slice = trat slice = mes;>* é não possibilitar que em alguns desdobramentos pudéssemos utilizar variâncias complexas, como é o caso destes dois tipos de desdobramento realizados. O R não permite que especifiquemos erros que são combinações de quadrados médios distintos. Então, apesar de as somas de quadrados estarem corretamente calculadas, os testes de hipóteses desta opção devem ser refeitos manualmente. Um outro problema é a impossibilidade de aplicar um teste de médias para algum desdobramento que tenha apresentado teste de hipótese significativo, utilizando o próprio programa.

6.4 Componentes de Variância

Podemos utilizar o *proc glm* para obtermos componentes de variância. Componentes de variância surgem quando alguns dos fatores que estamos estudando são aleatórios. Estes fatores são considerados aleatórios quando temos interesse na população de origem. Os níveis destes fatores são amostras aleatórias destas populações. Assim, temos interesse na média geral daquele efeito e principalmente na variância. Em geral, não temos nenhum interesse particular de comparar os níveis de fator aleatório.

A idéia de um dos métodos para estimarmos os componentes da variância dos efeitos aleatórios do modelo consiste em igualarmos as estimativas dos quadrados médios às suas esperanças $E(QM)$ e resolvermos as equações resultantes. Esse método é conhecido como método dos momentos. O *proc glm* permite que obtenhamos as esperanças dos quadrados médios por meio do comando *random*. Um modelo pode ser classificado como fixo, quando

todos os seus efeitos, excetuando a média geral e o erro, são fixos. Se todos os efeitos forem aleatórios, temos um modelo aleatório. Se por outro lado, tivermos efeitos fixos e efeitos aleatórios, teremos um modelo misto.

Quando temos efeitos aleatórios no modelo, os testes de hipóteses em muitas situações podem não ser feitos utilizando o quadrado médio do resíduo na obtenção da estatística. A decisão de qual deve ser o denominador da estatística do teste F, depende das esperanças dos quadrados médios. Nem sempre a especificação deste denominador é trivial, pois pode haver a necessidade de composição de quadrados médios. A opção *test* do comando *random* permite que testes F adequados sejam feitos nos modelos mistos ou aleatórios. Este comando (*random*) é essencialmente útil quando temos dados não balanceados.

Vamos ilustrar o uso do *proc glm* com um delineamento em blocos casualizados com 2 repetições. Uma amostra aleatória de 5 cultivares foi obtida pelo pesquisador e constituiu o fator de interesse da análise. Adicionalmente, esse experimento foi implantado em 2 locais. Assim, esse é um exemplo em que aplicaremos uma análise conjunta. Ocorreu, no experimento do local 1, uma perda de parcela. A repetição 1 da cultivar 5 foi perdida.

O interesse reside no componente de variância para cultivar, que foi considerada de efeito aleatório. O efeito de bloco, em geral, é considerado como aleatório na literatura. Pelo fato de o efeito de cultivar ter sido considerado aleatório e o de local fixo, a interação é considerada aleatória. Os comandos R, necessários para estimarmos os componentes de variância dos efeitos aleatórios, são dados por:

```
/* Programa para realizar análise de variância conjunta de um modelo misto.*/
data rand;
input cult bl local prod;
cards;
1 1 1 8.4
1 2 1 8.6
2 1 1 5.7
2 2 1 5.8
3 1 1 4.5
3 2 1 6.7
4 1 1 5.9
```

```
4 2 1 7.8
5 2 1 8.9
1 1 2 6.2
1 2 2 7.6
2 1 2 8.3
2 2 2 9.5
3 1 2 3.5
3 2 2 4.9
4 1 2 7.4
4 2 2 8.8
5 1 2 8.9
5 2 2 9.0
;
proc glm data=rand;
    class cult bl local;
    model prod = bl(local) cult local cult*local / e3 ss3;
    random bl(local) cult cult*local / test;
run; quit;
```

Merecem destaques alguns comandos e especificações de modelo utilizados. O comando *<model prod = bl(local) cult local cult*local / e3 ss3;>* possui o efeito de bloco hierarquizado em local. Não podemos especificar apenas o efeito de bloco, pois estaríamos ignorando o fato de que os blocos dos diferentes locais não são os mesmos. Assim, o bloco 1 do local 1 é diferente do bloco 1 do local 2. As opções *e3* e *ss3* indicam que as esperanças dos quadrados médios, utilizando somas de quadrados do tipo III, devem ser utilizadas. No comando *<random bl(local) cult cult*local / test;>*, que aparece após o comando *model*, indicamos ao *proc glm* quais são os efeitos aleatórios do modelo. Nesse exemplo foram os efeitos de bloco dentro de local, de cultivar e da interação cultivar $\times$ local.

Inicialmente o R apresenta o resultado da análise de variância do tipo III, cujo resumo apresentamos na Tabela 6.4. Se o modelo possui efeitos aleatórios, os testes de significância (teste F) apresentados nessa análise provavelmente podem estar incorretos. Nesse exemplo, como apenas o efeito de local é considerado fixo, sendo todos os demais aleatórios, a maioria dos testes F está incorreta. O correto é utilizar as esperanças dos quadrados médios para especificar os testes de hipóteses adequados e também para estimar os componentes de variância.

Tabela 6.4. Análise da variação para o modelo de análise conjunta (2 locais) em um delineamento de blocos casualizados.

FV	G.L.	SQ III	QM	F	$Pr > F$
Modelo	(11)	(52,9816)	4,8165	13,65	0,0011
bl(local)	2	5,4450	2,7225	7,72	0,0170
cult	4	27,4770	6,8693	19,47	0,0007
local	1	0,7111	0,7111	2,02	0,1987
cult*local	4	15,5483	3,8871	11,02	0,0038
Erro	7	2,4700	0,3529		
Total	18	55,4516			

CV = 8,27% $\bar{Y}_{...} = 7{,}1789$

Um segundo resultado apresentado pelo R, associado a análise de variação, refere-se as esperanças dos quadrados médios. Estes resultados estão sumariados na Tabela 6.5. Uma análise das esperanças dos quadrados médios mostra que o testador para bloco(local) e para a interação cultivar × local é o erro experimental. O testador para cultivar é a interação cultivar × local e o testador para local tem de ser obtido por uma combinação de quadrados médios. A opção *test* do comando *random* nos permite obter as estatísticas destes testes automaticamente.

Tabela 6.5. Esperança dos quadrados médios e resumo da análise da variação para o modelo de análise conjunta (2 locais) em um delineamento de blocos casualizados.

FV	G.L.	QM	E(QM)
bl(local)	2	2,7225	$\sigma^2 + 4{,}5\sigma^2_{b(L)}$
cult	4	6,8693	$\sigma^2 + 1{,}8333\sigma^2_{CL} + 3{,}6667\sigma^2_{C}$
local	1	0,7111	$\sigma^2 + 1{,}7778\sigma^2_{CL} + 4{,}4444\sigma^2_{b(L)} + Q_L$
cult*local	4	3,8871	$\sigma^2 + 1{,}8333\sigma^2_{CL}$
Erro	7	0,3529	σ^2

Q_L é a forma quadrática associada a local

A estimativa do componente de variância de cultivar pode ser obtida por: $\hat{\sigma}_C = (QMCult - QMCult \times Local)/3{,}6667 = 0{,}8133$. Os demais componentes de variância podem ser obtidos de maneira similar. Muitas vezes temos dificuldades em determinar qual é o quadrado médio que devemos subtrair do quadrado médio correspondente ao fator aleatório para o qual desejamos estimar o componente. Para a interação, isso foi ob-

tido de uma maneira bastante simples por $\hat{\sigma}_{CL} = (QMCult \times Local - QMErro)/1{,}8333 = 1{,}9278$. Quando precisamos combinar quadrados médios, o melhor indicativo para determinarmos esta combinação é fornecida pelo comando *test*. Por exemplo, se desejássemos testar a hipótese de que o efeito quadrático Q_L devido a local, que é fixo, seja nulo, poderíamos utilizar a seguinte combinação de quadrados médios como denominador da expressão da estatística do teste F:

$$0{,}9877\text{QMbl(local)} + 0{,}9697\text{QMcult} \times \text{local} - 0{,}9574\text{QMErro},$$

cujos graus de liberdade associados seriam obtidos pelo processo de Satterthwaite (1946).

Utilizando os testes adequados apenas os efeitos de bloco(local) e da interação cultivar × local foram significantes, indicando que os componentes de variância associados são diferentes de zero. Para cultivar não foi detectada significância estatística, sendo considerado nulo o componente de variância associado. Outras tipos de somas de quadrados podem ser utilizadas para estimarmos componentes de variância e para realizarmos os testes F. Para selecionarmos, por exemplo, as somas de quadrados do tipo II, bastaria trocar o comando *<model prod = bl(local) cult local cult*local / e3 ss3;>* por *<model prod = bl(local) cult local cult*local / e2 ss2;>*. Quando aplicamos esta mudança, os resultados dos testes são praticamente idênticos aos obtidos com as somas de quadrados do tipo III.

O R possui outros procedimentos para estimarmos componentes de variância. Podemos destacar o *proc mixed* e o proc *proc varcomp*. Estes procedimentos são muitas vezes mais adequados para estimarmos componentes de variância, além de oferecerem mais alternativas de métodos. Discutiremos o varcomp posteriormente nesse material. Os modelos mistos são uma generalização dos modelos lineares utilizados no *proc glm*.

6.5 Exercícios

1. Utilizar dados não balanceados resultantes de pesquisas desenvolvidas em sua área e realizar análises de variâncias utilizando o *proc glm*. Aplicar os testes de médias, se os níveis forem qualitativos, ou ajustar modelos de superfície de resposta ou de regressão, se os níveis dos fatores forem quantitativos.

2. Dar sua opinião sobre o fato de muitos autores ainda recomendarem estimação de parcelas, em conjuntos de dados onde foram perdidas uma ou mais delas. Como você lidaria com conjuntos de dados não balanceados? Estimaria os valores perdidos?

Referências Bibliográficas

BECKMAN, R. J.; TRUSSELL, H. J. The distribution of an arbitrary studentized residual and the effects of updating in multiple regression. *Journal of the American Statistical Association*, 69:179–201, 1974.

CHATTERJEE, S.; HADI, A. S. Influential observations, high leverage points, and outliers in linear regression. *Statistical Science*, 1(3):379–393, 1986.

FERREIRA, D. F. *Estatística básica*. Editora UFLA, Lavras, 2005. 676p.

GOMES, F. P. *Curso de estatística experimental.* Esalq/Usp, Piracicaba, 14 edition, 2000. 476p.

O'NEILL, R.; WETHERILL, G. B. The present state of multiple comparison methods. *Journal of the Royal Statistical Society*, 33(2):218–250, 1971.

SATTERTHWAITE, F. E. An approximate distribution of estimates of variance components. *Biometrics Bulletin*, 2(6):110–114, 1946.

SEARLE, S. R. *Linear models*. John Wiley, New York, 1971. 532p.

SEARLE, S. R. *Linear models for unbalanced models*. John Wiley, New York, 1987. 536p.

VANGEL, M. G. Confidence intervals for a normal coefficient of variation. *The American Statistician*, 15(1):21–26, 1996.

VELLEMAN, P. F.; WELSCH, R. E. Efficient computing of regression diagnostics. *The American Statistician*, 35(4):234–242, 1981.

Índice Remissivo