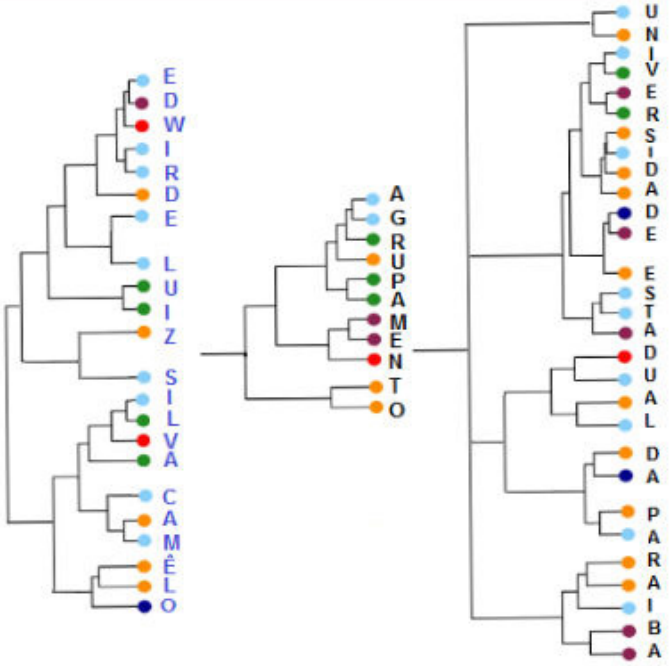
Edwirde Luiz Silva Camêlo
Paulo J. G. Lisboa
Ramón Gutiérrez Sánchez
Dalila Camêlo Aguiar

Principais Técnicas de Agrupamento

com aplicação em R



eduepb

EDITORA DA UNIVERSIDADE ESTADUAL DA PARAÍBA
Rua Baraúnas, 351 - Bairro Universitário - Campina Grande-PB - CEP 58429-500
Fone/Fax: (83) 3315-3381 - http://eduepb.uepb.edu.br - email: eduepb@uepb.edu.br

Edwirde Luiz Silva Camêlo
Paulo J. G. Lisboa
Ramón Gutiérrez Sánchez
Dalila Camêlo Aguiar

PRINCIPAIS TÉCNICAS DE AGRUPAMENTO

com aplicação em R

eduepb
Campina Grande - PB
2020

Editora da Universidade Estadual da Paraíba

Luciano Nascimento Silva | *Diretor*
Antonio Roberto Faustino da Costa | *Editor Assistente*
Cidoval Morais de Sousa | *Editor Assistente*

Expediente EDUEPB

Erick Ferreira Cabral | *Design Gráfico e Editoração*
Jefferson Ricardo Lima Araujo Nunes | *Design Gráfico e Editoração*
Leonardo Ramos Araujo | *Design Gráfico e Editoração*
Elizete Amaral de Medeiros | *Revisão Linguística*
Antonio de Brito Freire | *Revisão Linguística*
Danielle Correia Gomes | Divulgação

Depósito legal na Biblioteca Nacional, conforme decreto nº 1.825, de 20 de dezembro de 1907.

P957 Principais técnicas de agrupamento: com aplicações em R.[Livro eletrônico]./ Edwirde Luiz Silva Câmelo...[et al.]. Campina Grande: EDUEPB, 2020.
1400 Kb. - 144 p.: il. color.

ISBN 978-85-7879-616-7 (E-book)

1.Análise Multivariada. 2.Estatística computacional. 3.Estatística – Estudo e ensino. 4.Agrupamento – Aplicações em R. I.Camêlo, Edwirde Luiz Silva. II.Lisboa, Paulo J. G.. III.Sánchez, Ramón Gutiérrez. IV.Aguiar, Dalila Camêlo.

21. ed. CDD 512.2

Ficha catalográfica elaborada por Heliane Maria Idalino Silva – CRB-15ª/368

AGRADECIMENTOS

Gostaríamos de agradecer à EDUEPB, que muito tem apoiado e estimulado a divulgação dos trabalhos dos professores da UEPB. Devemos também agradecer ao trabalho profissional do Prof. Antônio de Brito Freire, que pacientemente abrilhantou este trabalho realizando as devidas correções gramaticais. Finalmente, nós autores não poderíamos deixar de agradecer aos alunos que nos permitiram a experiência para elaboração deste livro, aos colegas que contribuíram para a realização desta compilação e em especial ao **Eterno, Criador dos céus e da terra** que nos outorgou vida para concluir esta obra.

Sumário

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução ao *R*

R é um software estatístico, com ambiente integrado e com linguagem de programação especialmente desenvolvida para a análise de dados, cálculos estatísticos e representações gráficas.

É uma linguagem de programação muito simples, disponibilizada para diferentes plataformas (*Unix, MacOS, Windows*) e de fácil instalação. O melhor de tudo isso, é que se trata de um software gratuito e amplamente utilizado na pesquisa científica

Primeiro, você precisa instalar o *R* e, para isso, faça o *download* do site oficial http://cran.at.r-project.org/, de preferência a última versão estável, 2.15.0, clicando no *Windows* e depois no link *base* e, a partir daí, baixamos *R-2.15.0-win.exe*. E agora é só seguir os passos solicitados, em caso de dúvidas, pode consultar a instalação passo a passo em https://cran.r-project.org/doc/contrib/Itano-installation.pdf.

Após a instalação do *R*, instala-se o *RStudio*, que é um software livre de ambiente de desenvolvimento integrado ao *R*. É um ambiente mais amigável do que *R* para se trabalhar. Sua instalação

também simples e recomenda-se baixar a versão mais estável. O download pode ser feito em

https://www.rstudio.com/products/rstudio/download/.

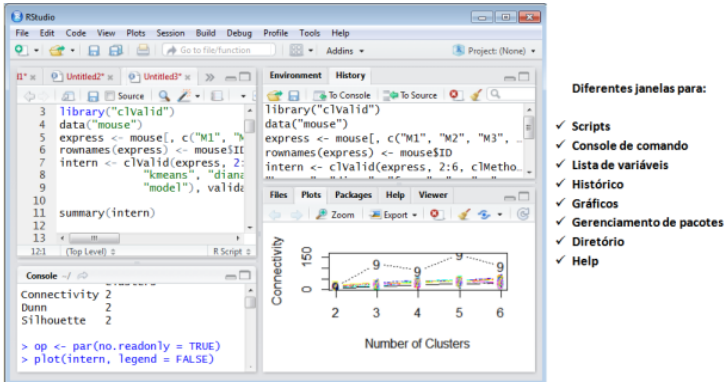


Figura 1.1: Interface de usuário para *RStudio*

Algumas vantagens do *RStudio*:

- Uma interface que permite uma simples manipulação, incluindo a visualização ou modificação dos dados brutos.
- Uma interface entre o usuário e os dados que podem executar várias operações ou aplicar vários testes para dados, além de apresentar resultados de diversas maneiras (gráficos, tabelas, etc).
- Para evitar a multiplicação de aplicações de software para a obtenção de dados utilizados na análise estatística.
- Para dados em 3D os dispositivos gráficos devem permitir alguma interatividade. Uma vez que a tela é plana, é preciso observar e girar dados 3D de uma forma convivial.

- Para obter resultados e armazená-los sob a forma de gráficos e tabelas.
- Para se adaptar e converter arquivos que foram tratados em outros softwares e arquivos de exportação para usuários que ainda relutam em usar o programa, assim como criar funções personalizadas.

Pode-se obter ajuda do próprio programa digitando o comando `help()`. Uma breve introdução ao uso do programa *R* pode ser encontrada em (LANDEIRO, 2011) no site https://cran.r-project.org/ em contribuição. Outra opção é ler o manual *R for Beginners* (PARADIS, 2005).

Para usar o *R* é necessário conhecer e digitar comandos. Em seguida apresentaremos alguns tópicos necessários.

1.1 Objetos

Os objetos são caracterizados por seus atributos. O modo e duração são atributos de todos os objetos em *R*. Se os elementos são dados, eles podem ter quatro modos diferentes: numérico, caráter, complexo e lógica (`FALSE` ou `TRUE`, alternativamente digitado como `F` ou `T`). O comprimento é o número de elementos do objeto e é devolvido digitando comprimento `lenght`(*objeto*). Finalmente, a função `str` mostra a estrutura interna de um objeto.

Exemplo 1.1.1. *Numérico, complexo, categórico e lógico*

```
a<-777; b<-3+4i; c<-"CD"; d<-TRUE
mode(a); mode(b); mode(c);  mode(d)
```

Também é possível verificar e coagir o modo de objetos usando as funções: `is.numeric`, `is.complex`, `is.character`, `is.logical` e `as.numeric`, `as.complex`, `as.character`, `as.logical`. O `"is"` antes dos modos significa uma pergunta, por exemplo o número dois é numérico? (`is.numeric(2)`), a resposta será `TRUE` (verdadeiro). E o `"as"` antes dos modos em algumas vezes significa transformar um modo, por exemplo, como caracter, `as.character(3)`, embora o 3 seja um numérico se torna aqui como um caracter, e para identificar entre aspas, "3".

```
#SE workspace está vazio?
ls()
# Se seu diretório de trabalho é o desejado?
verifique que está vazio, com o comando:
dir()
#Salve-o usando o comando:
save.image()
```

1.2 Encontrando uma função no *R*

Através do web site (http://rseek.org/) pode-se encontrar qualquer função no *R*. No quadro abaixo tem-se alguns prováveis links de ajuda no *R*.

http://www.r-project.org	*R* web site
http://www.cran.r-project.org	Downloads
http://www.rseek.org	Buscador de função
http://www.cran.r-project.org/web/views	Organizado por tarefa
www.tolstoy.newcastle.edu.au/R/	Discussão sobre o *R*

Através do comando `"apropos"`, é possível encontrar algumas funções que foram carregadas com os pacotes instalados. Também é possível pesquisar a documentação entre todos os pacotes instalados usando o comando `"help.search"`.

apropos("read")	Funções que se iniciam com *read*
apropos("mult")	Funções que se iniciam com *mult*
help.search(".matrix")	Funções que se iniciam com *matrix*
apropos(".test")	Busca as funções que terminan com .test

```
apropos(plot)
help.search(field="title","skew")
example(mean)
args(chisq.test) #lembrar do argumento da função
```

1.3 Instalando pacotes no *R*

Pacotes são conjuntos de funcionalidades (funções, dados e exemplos) distribuídos em conjunto para realizar tarefas específicas. Por exemplo, o pacote base carrega na sua área de trabalho (deixa disponível para uso) um conjunto de ferramentas básicas no *R*. É necessário entender as diferenças entre baixar (*download*) o pacote do repositório e carregar em sua área de trabalho. Para baixar algum pacote disponível no repositório CRAN do *R* é necessário utilizar o comando `install.packages`("pacote"). O *R* possui muitos pacotes (http://www.r-project.org/).

1.4 Operações básicas em vetores e matrizes

É muito fácil realizar operações básicas no *R*. A Tabela 1.1 mostra alguns exemplos.

```
A<- 3 + 5 + 3; A
B<- 3-8-7; B
C<- 3*4
D<- 9/7
2 + 6 # forma direta e o resultado
```

Tabela 1.1: Algumas operações básicas no *R*.

Log	$log_{base}(x)$	$log_{10}(2)$	log(2,base =10)=0.301
Raiz	sqrt(x)	$\sqrt{4}$	sqrt(4) =2
Log exp	$\log_e(2)$	$\log(2, exp(1))$	$\log(2, base = exp(1)) = 0.693$
sen(x)	$seno(\pi/4)$	$sen(\pi/4)$	$\sin(pi/4) = 0,707$

O *R* tem uma sintaxe de expressão aritmética convencional com aritmética habitual e operadores condicionais.

```
help(Arithmetic)
help(Comparison)
help(Syntax)
```

Os operadores aritméticos e condicionais são:

a + b	soma	a == b	a é igual a b?
a - b	divisão	a ! = b	a não é iqual a b?
a*b	multiplicação	a < b	a é menor que b?
a/b	divisão	a <= b	a é menor e igual a b?
a^b	Potenciação	a > b	a é maior que b?
-a	negação	a >= b	a é maior ou igual que b?

```
a=c(0,2,3,4,5,6,7)
b=c(1,5,2,8,12,10,10)
a+b #somando os elementos de a com os de b
[1]  1  7  5 12 17 16 17
a<b #  Testando os correspondentes elementos
TRUE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE
# Apenas o terceiro é maior que b
```

Na matriz aritmética teremos:

```
x=matrix(1:9,nrow=3, ncol=3)
# 9 elementos de 1 a 9 em 3 linhas e 3 colunas
xl=matrix(c(1,2,3,4,5,6,7,8,9), ncol=3,
byrow=TRUE)
# 3 colunas, leitura por linha
xc=matrix(c(1,2,3,4,5,6,7,8,9), ncol=3,
byrow=TRUE)
# 3 colunas, leitura por colunas igual a x
```

Uma variável de tipo caracter que mantém como valor uma cadeia de valores de dígitos pode ser manipulado por meio de funções especiais, vejamos um exemplo.

- `paste(..., sep="'')`. Concatena vetores depois de transformá-los em caracteres; o `sep="''` indica como as séries serão separadas (a definição padrão é um espaço em branco).

```
(nth<-paste0(1:5,c("ind","ind","ind",
rep("Novo",2))))
#[1] "1ind" "2ind" "3ind" "4Novo" "5Novo"
```

1.4.1 Matrizes

Uma matriz é uma coleção de elementos de dados dispostos em um layout retangular bidimensional. O seguinte exemplo é de uma matriz com 2 linhas e 3 colunas.

```
 H = matrix(
 c(7, 7, 3, 2, 4, 2),# os elementos da matriz
 nrow=2,             # número de linhas
 ncol=3,             # número de colunas
 byrow = TRUE)       # lendo os dados por linha
 H                   # "print" a matriz
#       [,1]  [,2] [,3]
# [1,]    2    4    3
# [2,]    1    5    7
```

Pode-se separar alguns elementos da matriz usando a expressão $H[m,n]$. Por exemplo:

```
H[2, 3] # Elemento da 2ª linha e 3ª coluna
# [1] 7
H[ ,c(1,3)] # 1ª e 3ª coluna
# [,1] [,2]
# [1,]    7    3
# [2,]    2    2
dimnames(H) = list(
    c("row1", "row2"), # nomes das linhas
    c("col1", "col2", "col3")) #das colunas
H
#       col1 col2 col3
# row1    7    7    3
# row2    2    4    2
A<-H[ ,c(1,3)]
#       col1 col3
# row1    7    3
# row2    2    2
```

Algumas funções de matrizes são apresentadas abaixo.

Funções	Significados	Comandos
t	transposta	$t(H)$
$diag$	diagonal	$diag(H)$
$\%*\%$	multiplicação	$H\%*\%H$
det	determinante	$det(H)$
$solve$	inversa	$solve(H)$
$eigen$	autovalores	$eigen(H)\$values$
$eigen$	autovetores	$eigen(H)\$vectors$
svd	decomposição de valores singulares	$svd(H)$
qr	descomposição QR	$qr(H)$
$chol$	decomposição Choleski	$chol(H)$

1.5 Criando fatores

Fatores representam variáveis categóricas e são usados também como indicadores de grupos. Usa-se a função "as.factor()" para criar um fator de um vetor e a função "as.numeric" e "levels" para ter os códigos internos dos fatores e legendas. Por exemplo:

```
x<-as.factor(c("Pulmonar","ExtraPulmonar",
"Urinária",
"Outras","Pulmonar","ExtraPulmonar"))
as.numeric(x)
levels(x)
table(x)


x<-c("Pulmonar","ExtraPulmonar","Urinária",
"Outras",
"Pulmonar","ExtraPulmonar")
as.factor(x)
x<-rep(6:1, each=2)
as.factor(x)
# Agrupando fatores
gl(4,3,labels=c("1", "2","3", "4"))
```

1.6 Operação de arredondamento e truncamento

O sistema utiliza basicamente 4 funções:

- floor(x), arredonda o valor passado no argumento para o próximo menor. *Floor*=piso.
- trunc(x), trunca o valor eliminando a componente decimal.

- `round(x, digits=0)`, arredonda para o número inteiro mais próximo. O arredondamento vem efetuado ao número decimal considerado.
- `ceiling(x)`, arredonda para o próximo superior. *Ceiling*=teto.

A Tabela 1.2 ilustra estes arredondamentos e truncamentos.

Tabela 1.2: Arredondamentos e truncamentos.

Valor	floor(x)	trumc(x)	round(x)	round(x,3)	ceiling(x)
7.4955	7	7	7	7.495	8
-7.4955	-7	-7	-7	-7.295	-7
7.5	7	7	7	7.5	8
-7.511	-8	-7	-8	-7.511	-7

1.7 Criando função no *R*

Para criar uma função no *R* usa-se a seguinte sintaxe: *Nome*<-*function(argumento/i)corpo da função*. Vamos encontrar as raízes de uma equação do segundo grau `x^2-3x+2=0`.

```
zero.funcao2<-function(a,b,c){
   delta<-b^2-4*a*c
   x1<-(-b+sqrt(delta))/(2*a)
   x2<-(-b-sqrt(delta))/(2*a)
   return(c(x1,x2))}
zero.funcao2(1,-3,2)
# [1] 2 1
```

```
prima.função<-function()
{cat("pi grego = "pi, "\n")}
Binomio<-function(a,b)
  {(a+b)/(a-b)}
Binomio(3,1)
# [1] 2  # (3+1)/(3-1)=2
```

1.8 Valores pré-determinados pelo sistema

O sistema utiliza valores pré-estabelecidos. Através da função `"options"` pode-se observar:

```
oldOp<-options() # armazena as configurações
                 # originais
oldOp #configurações presentes no sistema
ls(oldOp) # Lista todas as configurações
options(digits=4)
# [1] 3.142   valor pi com 4 dígitos
options(oldOp) # Configuração original
pi
# [1] 3.141593
```

1.9 Vetores

Um vetor é um conjunto ordenado de dados associados a um objeto. Se quisermos criar um vetor chamado de `a` com os valores 10, 20 e 30, usamos:

```
a<-c(10,20,30)
a
# [1] 10 20 30
```

Observe abaixo alguns vetores não métricos.

```
Nomes<-c("Jonas", "Ana")
Idades<-c(40,23,8)
Gêneros<-c("M", "F")
Irmãos<-c("Ivo", "Adan","Eva", "Edson",
          "Pedro", "Luisa")
Amigos<-c("Jorge","André", "José")
Todos<-c(Nomes, Irmãos, Amigos)
Todos
data.entry(Todos)
Todos
# [1] "Jonas" "Ana"   "Ivo"   "Adan"   "Eva"
# [6] "Edson"   "Pedro"   "Luisa"
# [9] "Jorge"   "André"   "José"
```

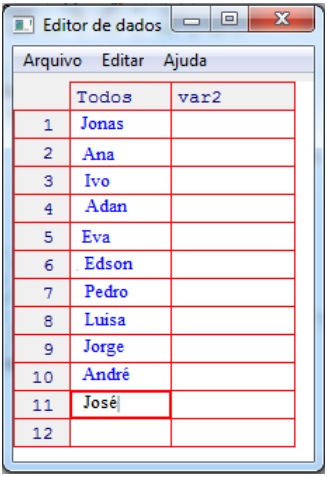


Figura 1.2: Acrescentando o nome José na tabela do editor de texto

Antes de voltar ao programa, não esquecer de fechar a tabela

`data.entra()`.

1.10 Tratando dados perdidos

Dados perdidos no *R* é indicado por `NA`. Criando um vetor numérico `A` com três valores perdidos, a função `mean()` retorna `NA`, indicando que o valor faltante não permite o cálculo da média.

```
A<-c(1,3,4,NA,1,0,NA,8,1,NA); A
# [1]  1  3  4 NA  1  0 NA  8  1 NA
mean(A)
# NA
```

Informando `na.rm=TRUE` no *R*, ignora-se os `NA` e calcula-se a média com os números restantes. Por exemplo:

```
mean(A, na.rm=TRUE)
# 2.571429
(1+3+4+1+0+8+1)/7
# 2.571429
```

Usando a função `cbind` (colunas (`c`) vinculadas (`bind`)), junta as colunas `x` e `y` no mesmo objeto `z` sem os `NA`.

```
x <- c(160, NA, 175, NA, 180)
y <- c(NA, NA, 65, 80, 70)
z<- cbind(x = x[!is.na(x) & !is.na(y)],
y = y[!is.na(x) & !is.na(y)]); z
#     x  y
# [1,] 175 65
# [2,] 180 70
```

1.11 *Looping* - fazer um laço

Quando se precisa executar uma operação que se repete, usa-se um `Looping` (laço), informando a ideia de um círculo que se repete até uma determinada conclusão.

1.11.1 *Conditional Statements and Branching* - As declarações condicionais e ramos

Utilizam-se as seguintes instruções:

- `switch(<expr:stat>,<expr:casel> = <codl>,<expr:case2> = <cod2>, etc).`

Na declaração acima `<expr:test>` é um número ou string (valor que liga a outros). Esta declaração retorna: `<codl> if <expr:test> values \verb<expr:casel>", <cod2> if <expr:test> values <expr:case2>`, etc. Se `<expr:test>` não é igual a qualquer um dos `<expr:case>`, a função `switch()` retorna `NULL`. Por exemplo:

```
r<-rnorm(10,0,1)
seja <- "valores"
switch(seja,valores = mean(x),mediane
         = median(x))
seja <- "mediane"
switch(seja,valores = mean(x),mediane
         = median(x))
seja <- "sd"
switch(seja,valores = mean(x),mediane
         = median(x))
```

- Instruções `if` e `else` (se, caso de outra maneira). A instrução `if` condicional é usado nas duas formas seguintes:

```
if "cond" <expr:vrai>
ou
if "cond" <expr:vrai> el se <expr:faux>
```

O parâmetro `"cond"` deve ser, portanto, uma lógica que leva a um dos valores `TRUE` ou `FALSE`. Se `cond` for `TRUE`, a instrução será executada. Mas se é falso, nada acontece. Em outras palavras, a condicional é realizada com `if` e `else`, tem-se a seguinte sintaxe:

```
if (test) {
executes something
} else {
executes something else
}
Gênero<-c("M", "F" ,"F", "M", "F", "M")
ifelse(Gênero=="M", "Masculino", "Feminino")
# [1] "Masculino" "Feminino"
# [3] "Feminino" "Masculino"
# [5] "Feminino" "Masculino"
ifelse(Gênero == "F","Feminino",
       "Masculino")
# [1] "Masculino" "Feminino"
# [3] "Feminino" "Masculino"
# [5] "Feminino""Masculino"


x<- 1:50
xt<-ifelse(x%%7==0, NA,x)
```

```
ris<-rt[!is.na(xt)]
ris[1:20]
# [1]  1  2  3  4  5  6  8  9 10 11 12 13
# 15 16 17 18 19 20 22 23
# Observa-se que os números múltiplos de 7
# foram eliminados
```

- Instrução `stop` e `return`

```
x <- TRUE
if(x) y <- 1 else y <- 0; y
# [1] 1
###### Fatorial de número positivo
Fatorial<-function(n){
if(n<0)
stop("Argumento negativo")
else{
if(n==0)
return(1)
else
return(prod(1:n))}
}
Fatorial(0)
# [1] 1
Fatorial(4)
# [1] 24
Fatorial(-1)


# Error in Fatorial(-1) : Argumento negativo
```

- Instruções `for`

```
for(n in (1:3)) print(n)
# [1] 1
# [1] 2
# [1] 3
for(n in seq(0,6, by=2)) print(n)
# [1] 0
# [1] 2
# [1] 4
# [1] 6
for(x in c("Jonas", "Ana")) print(x)
# [1] "Jonas"
# [1] "Ana"
for(f in c(log,log2,log10)) print(f(25))
# [1] 3.218876
# [1] 4.643856
# [1] 1.39794
```

- Instruções `return`

```
zero.funcao2<-function(a,b,c){
   delta<-b^2-4*a*c
   x1<-(-b+sqrt(delta))/(2*a)
   x2<-(-b-sqrt(delta))/(2*a)
   return(c(x1,x2))}


zero.funcao2(1,-3,2)
# [1] 2 1
```

- instrução `while`

A sintaxe da instrução é a seguinte: `while (<condition>) <expression>`.

A lógica `"while"` é que enquanto houver número inteiro e positivo se calcula o fatorial (`n!`).

```
 fatorial2<-function(n){
if(n<0)
stop("Argumento negativo")
ra<-1
while(n>0){
ra<-ra*n
n<-n-1
}
return(a)
}
 fatorial2(5)
# [1] 120
 fatotoril2(0)
# [1] 1
 prod(1:5) # 5!
# [1] 120


H <- 0
while (H < 1){

        H <- rnorm(1)
         cat(H,"\n")
      }
```

```
# -1.114971
# -0.7465893
# 1.740903
```

1.11.2 Instrução *repeat* (repetir) e *break* (interromper)

Observa-se abaixo que foram gerados números somado com 1 até encontrar o 4, e quando encontrou houve um `break` (stop).

```
j <- 0
repeat{
j<-j+1
if (j==4) break}; j
# [1] 4
```

1.12 Operação de leitura de dados

Os data.frame apresenta estrutura semelhante à de uma matriz, embora suporte valores numéricos e alfanuméricos. Normalmente, quando um estudo estatístico é realizado sobre os sujeitos ou objetos de uma amostra, a informação se organiza precisamente em um dataframe: uma folha de dados, em que cada linha corresponde a uma sujeito e cada coluna a uma variável.

1.12.1 Lendo um data.frame de um arquivo de texto

Usa-se a função `read.table()` para ler dados em formato de texto. Considere os seguintes arquivo no bloco de notas (arquivo do tipo `.txt` ou `.dat`).

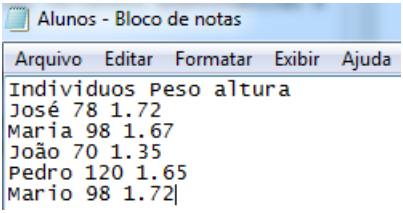


Figura 1.3: Nomes de alunos, peso e altura dos mesmos.

As colunas foram separadas por espaço no bloco de notas, se os espaços fossem separados por (;) teríamos que escrever `sep=";"`.

```
setwd("C:/LIVRO") # Criar um arquivo em C
                  # chamado LIVRO
getwd() # confirmar a mudança
#salvar o arquivo Alunos.txt
# dentro desse arquivo.
dados<-read.table("Alunos.txt",
header=TRUE, sep="")
dados
# Individuos Peso altura
# 1       José   78   1.72
# 2      Maria   98   1.67
# 3       João   70   1.35
# 4      Pedro  120   1.65
# 5      Mário   98   1.72
```

1.12.2 Lendo um data.frame de um arquivo de Excel

A abreviação `.csv` *comma-separated values* significa (valores separados por vírgula). O programa *Excel* também ler este tipo de arquivo. É um arquivo de apenas uma planilha que contém dados

separados por um sinal de pontuação, como vírgula, ponto e vírgula ou ponto.

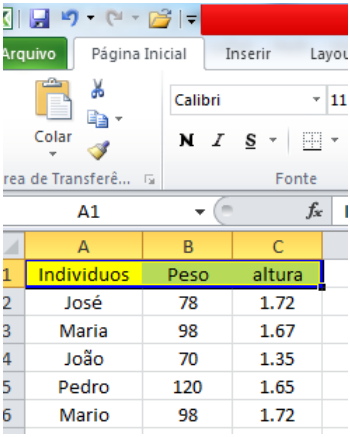


Figura 1.4: Alunos com peso e altura

```
setwd("C:/LIVRO")
dados<-read.csv("AlunosExcel.csv",
       header=TRUE, sep=";")
dados
# Individuos Peso altura
# 1       José   78   1.72
# 2      Maria   98   1.67
# 3       João   70   1.35
# 4      Pedro  120   1.65
# 5      Mário   98   1.72
```

Existe uma pequena diferença entre os comandos `read.csv()` e `read.csv2()`. O *default* (valor padrão) desses dois argumentos são: `sep` (separador de colunas) e `dec` (separador de casas deci-

mais). Na função `read.csv()` o default é `sep=","` e `dec="."`. Já em `read.csv2()` tem-se `sep=";"` e `dec=","`.

1.12.3 Lendo arquivos que usam um formato fixo

Neste tipo usa-se a função `read.fwf()`. Observa-se que os dados são separados em tamanho 6 (6 caracteres da palavra *ProdBB*), 4 para os anos, 2 código e 3 para índice (o número 5, o ponto e o número 7) considerando a primeira linha.

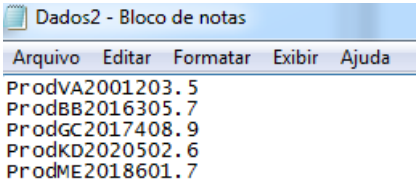


Figura 1.5: Produção, ano e códigos.

```
dados1<-read.fwf("Dados2.txt",
        widths=c(6,4,2,3), skip=1)
# Warning message:
# In readLines(file, n = thisblock) :
# incomplete final line found on 'Dados2.txt'
dados1
# V1 V2 V3 V4
# 1 ProdBB 2016 30 5.7
# 2 ProdGC 2017 40 8.9
# 3 ProdKD 2020 50 2.6
# 4 ProdME 2018 60 1.7
```

1.12.4 Leitura de arquivo direto da internet

Para ler um arquivo direto da internet, primeiro se deve clicar com o botão direito sobre o arquivo e selecionar a opção "Copiar endereço do link" para que possa ser escrito dentro da função *read.csv*2(). No *R*, basta colar o link do arquivo no local onde você normalmente colocaria o diretório de um arquivo do seu computador, assim observe um exemplo fictício:

```
read.csv2("http://www.estatistica777.com/
arquivos/dados.csv",sep=",", dec=".")
```

1.12.5 Como importar qualquer arquivo no *R*

O pacote *rio* serve para facilitar a importação de arquivos para o *R*. Com a função, *import*(), se pode detectar qual arquivo o usuário deseja abrir. O primeiro passo para usá-lo é instalar o pacote, que está no *CRAN*:

```
# Instalando o pacote 'rio'
install.packages(rio, dependencies = TRUE)
```

Após instalar o pacote com todas as dependências (outros pacotes que são necessários para que ele funcione), fica fácil carregar qualquer arquivo de dados, basta usar a função `import()` do pacote `rio` (CHAN *et al*., 2016).

```
# Carrega um arquivo em .csv
Dados <- import(file = "AlunosExcel.csv")

# Carrega um arquivo em .txt
Dados <- import(file = "Dados2.txt")
```

```
# Carrega um arquivo em .dta (Stata)
Dados <- import(file = "dados.dta")

# Carrega um arquivo em .sav (SPSS)
Dados <- import(file = "dados.sav")
```

vejamos alguns exemplos abaixo:

```
Dados_txt <- import(file = "dados3.txt")
Dados_txt
# G X1 X4
# 1   1  1  4
# 2   2  1  2
# 3   3  1  6
# 4   4  1  4
# 5   5  1  6
# 6   6  1  8
# 7   7  2  6
# 8   8  2  8
# 9   9  2  8
# 10 10  2  2
# 11 11  2  5
Dados_csv <- import(file = "AlunosExcel.csv")
Dados_csv
Indivíduos Peso altura
# 1       José   78   1.72
# 2      Maria   98   1.67
# 3       João   70   1.35
# 4      Pedro  120   1.65
# 5      Mário   98   1.72
```

1.13 Criando medidas repetidas

Considera-se dados multivariados de diferente natureza, ou seja, os dados resultantes das medições repetidas na mesma variável em cada unidade no conjunto de dados. Para entender o procedimento considere um pequeno conjunto de dados.

```
Dados<-read.table("MedidasRepetidas.txt",
         header=TRUE, sep="")
Dados
# Id G T.1 T.2 T.5 T.7
# 1 1 1 15 15 10 7
# 2 2 1 10 9 11 12
# 3 3 1 8 7 6 9
# 4 4 2 11 8 13 7
# 5 5 2 11 12 11 11
# 6 6 2 12 12 6 10
Rep<-reshape(Dados, direction="long",
idvar="Id", varying=colnames(Dados)[-(1:2)])
Rep
# ID G time T
# 1.1 1 1 1 15
# 2.1 2 1 1 10
# 3.1 3 1 1 8
# ...
# 5.7 5 2 7 11
# 6.7 6 2 7 10
```

1.14 Acrescentando colunas, linhas e grupos

Considere os mesmos dados trabalhado anteriormente.

```
dados<-read.table("dados3.txt", header=TRUE,
                  sep="")
dados
# G X1 X4
# 1   1  1  4
# 2   2  1  2
# 3   3  1  6
# 4   4  1  4
# 5   5  1  6
# 6   6  1  8
# 7   7  2  6
# 8   8  2  8
# 9   9  2  8
# 10 10  2  2
# 11 11  2  5
dadosU<- rep("G1",11) # repetindo G1
                      # onze vezes
dadosU# Acrescentando G1 na quarta coluna
# [1] "G1" "G1" "G1" "G1" "G1" "G1" "G1" "G1"
# [9] "G1" "G1" "G1"
dados[,4]<-dadosU[1]
# Alterando os nomes das colunas
colnames(dados)<-c("X2","X3","X4","G")
# Acrescentando elementos na 1a linha
dados[1,]<-c(10,22,47,"G1")
dados
```

```
colnames(dados, do.NULL=FALSE)
# [1] "X2" "X3" "X4" "G"
colnames(dados)<- c("X2","X3","X4","G")
dados<- cbind(1, dados)
colnames(dados)<- c("X1","X2","X3","X4","G")
dados
# X1 X2 X3 X4  G
# 1   1 10 22 47 G1
# 2   1  2  1  2 G1
# 3   1  3  1  6 G1
# 4   1  4  1  4 G1
# 5   1  5  1  6 G1
# 6   1  6  1  8 G1
# 7   1  7  2  6 G1
# 8   1  8  2  8 G1
# 9   1  9  2  8 G1
# 10  1 10  2  2 G1
# 11  1 11  2  5 G1
rownames(dados)<- rownames(dados[,1],
do.NULL = FALSE, prefix = "Ind.")
dados
# X1 X2 X3 X4 G
# Ind.1 1 10 22 47 G1
# Ind.2 1 2 1 2 G1
# Ind.3 1 3 1 6 G1
# Ind.4 1 4 1 4 G1
# Ind.5 1 5 1 6 G1
# Ind.6 1 6 1 8 G1
# Ind.7 1 7 2 6 G1
```

```
# Ind.8 1 8 2 8 G1
# Ind.9 1 9 2 8 G1
# Ind.10 1 10 2 2 G1
# Ind.11 1 11 2 5 G1
```

1.15 Janelas gráficas

Todos os gráficos criados em *R* são exibidos em janelas especiais, distinto do console, chamado "*R graphies: Device numero-device*", em que "número-device" é um número inteiro da janela (ou dispositivo). As diferentes instruções de *R* para realizar gráficos, que formam partes dos diferentes pacotes, se podem dividir em:

- Funções gráficas: Permitem realizar diferentes tipos de gráficos e têm seus próprios argumentos específicos.
- Gráficos complementares: São também funções que permitem acrescentar aos gráficos linhas, textos, flechas, legendas, etiquetas, etc., e também tem seus próprios argumentos específicos.
- Argumentos gerais: São argumentos que se podem usar nas funções e complementos gráficos anteriores.

Um pacote muito conhecido é o `graphics` (MURRELL, 2005). Para obter uma lista completa de funções com páginas de ajuda individuais, use `library(help=‘‘graphics’’)`, além de consultar o menu de ajuda da função:

```
library(graphics)
help(base)
```

Através do comando `demo(graphics)` se mostram bons exemplos de gráficos com seu correspondente código.

1.15.1 Principais *Scripts* e arquivos para geração de gráficos

abline: acrescenta linhas. Argumentos: `abline(a=NULL, b=NULL, h=NULL, v=NULL, reg=NULL, coef=NULL, untf=FALSE, ...)`.

- a e b: intersecção e inclinação da reta.
- h: valor de y em uma linha horizontal.
- v: valor de x em uma linha vertical.
- `reg`: um vetor do tipo $c(a,b)$ com a intersecção e inclinação da reta.
- `coef`: especifica-se uma regressão, $coef(lm(c(x \sim y)))$.
- `untf`: `FALSE` ou `TRUE`, se um eixo tem transformação `log`, com `TRUE`, representa a linha sem transformação.

arrows: representa flechas. Argumentos: `arrows(x0, y0, x1=x0, y1=y0, length=0.25, angle=30, code=2, col=par("fg"), lty=par("lty"), lwd=par("lwd"), ...)`.

- x_o e y_o: coordenadas da origem.
- x_1 e y_1: coordenadas finais.
- `lenght`: longitude da ponta da flecha.
- `angle`: ângulo em relação a ponta da flecha.
- `code`: tipo de flecha.

box: caixa de texto no gráfico. Argumentos: `box(which= "plot", lty="solid", ...)`. *Which* indica onde representa o quadro e pode ser: `plot, figure, "inner"` ou `"outer"`.

legend: acrescenta uma legenda. Principais Argumentos: `legend(x, y=NULL, legend, fill=NULL, col=par("col"), lty, lwd, pch, border="black", angle=45, density=NULL, bty="o", bg=par("bg"))`

- *x* e *y*: coordenadas da legenda.
- `legend`: texto da legenda.
- `fill`: cor do preenchimento.
- `ncol`: número de colunas.
- `horiz`: se for verdadeiro (`TRUE`) os textos são horizontais.
- A posição da legenda é especificada com coordenadas ou com palavras-chave:

`bottomright`: abaixo à direita.

`bottomleft`: abaixo à esquerda.

`topleft`: no topo à esquerda.

`topright`: no topo à direita.

`left`: à esquerda.

`right`: à direita.

`center`: no centro.

- `bty`: define o quadro da legenda: `"o"` com o quadro e `"n"` sem o quadro.

lines: acrescenta linhas ou pontos com linhas. Principais Argumentos: `lines(x, y=NULL, type="l", ...)`.

- *x* e *y*: coordenada dos pontos a unir mediante linhas.
- type: tipos de linhas: `"p"` para pontos, `"l"` para linhas, `"b"` desenha pontos ligados por curvas, `"c"` para as linhas da parte isolada de `"b"`, `"o"` desenha pontos com curvas sobrepostas `"overplotted"`, `"h"` para `"histograma"` com (ou "alta densidade") nas linhas verticais, `"s"` para passos de escada, `"S"` para outras etapas e `"n"` não desenha o gráfico, mas apresenta os eixos cujas coordenadas são determinadas de acordo com os dados.

locator: com auxílio do mouse, permite localizar a posição do objeto no painel gráfico e devolve as coordenadas. Argumentos: `locator(n=512, type="n", ...)`.

- `n`: o número máximo de objetos que se deseja localizar.
- `type`: com `"n"` localiza qualquer objeto, com `"p"` representa símbolos e com `"I"` localiza a linha unida aos pontos para o qual `n` deve ser maior que 1.

polygon: representa polígonos. Argumentos: `polygon(x, y=NULL, density=NULL, angle=45, border=NULL, col=NA, lty=par("lty"), ..., fill Odd Even=FALSE)`

- `x` e `y`: coordenadas dos vértices do polígono.
- `density`: densidade do sombreamento.
- `angle`: ângulo das linhas do sombreamento.
- `border`: cor do bordados, `NA` sem borda e `NULL` com a cor (preta) por *default*.

segments: desenha segmentos entre pontos. Argumentos: `segments(x0, y0, x1=x0, y1=y0, col=par("fg"), lty=par("lty"), lwd=par("lwd"), ...)`

- `x_{o}, y_{o}, x_{1}, y_{1}`: coordenadas de origem e final da linha.

text: permite acrescentar etiquetas nas coordenadas de um gráfico. Argumentos: `text(x, y=NULL, labels=seqalong(xx), adj=NULL, pos=NULL, offset=0.5, vfont=NULL, cex=1, col=NULL, font=NULL, ...)`

- `x` ou `y`: vetores numéricos de coordenadas onde os rótulos de texto devem ser escritos.
- `labels`: texto que deseja-se escrever ou variáveis com as etiquetas do texto dos pontos.
- `srt`: ângulo do texto, ou seja, o texto pode ser rotacionado usando este argumento.
- `offset`: posição do texto no eixo vertical em relação às coordenadas.
- `pos`: eixo em que se localizará o texto: abaixo (1), esquerda (2), acima (3) e direita (4).

title: permite acrescentar etiquetas as coordenadas de um gráfico. Argumentos: `title(main=NULL, sub=NULL, xlab=NULL, ylab=NULL, line=NA, outer=FALSE, ...)`

- `main`: texto do título do gráfico.
- `sub`: subtítulo.
- `xlab`: legenda do eixo `x`.
- `ylab`: legenda do eixo `y`.
- `line`: valor numérico que define a separação do texto em relação ao gráfico.

A Figura 1.6 mostra os valores de 1 a 25 que definem os símbolos do argumento `"pch"`.

Figura 1.6: Valores numéricos e símbolos da função do argumento `"pch"`.

A função `"colorTable()"` do pacote `fBasics` (WUERTZ, 2011) permite observar os números com diferentes cores.

1.16 Uso da função %>% (*pipe*)

O operador %>% (*pipe*) é usado para inserir um argumento em uma função. A ideia é usar o valor resultante da expressão do lado esquerdo como primeiro argumento da função do lado direito. Para utilizar o *pipe*, carregue o pacote `margrittr` (BACHE, 2014) utilizando o comando `library(magrittr)`. Um exemplo abaixo mostra que o operador do lado esquerdo é o primeiro argumento (1 a 20) e a função no lado direito, em seguida, o desvio padrão.

```
1:20 %>% sd
# [1] 5.91608 # é equivalente a
sd(1:20) # desvio padrão
# [1] 5.91608
```

Usa-se também para filtrar funções específicas dentro de pacotes existentes, como é o caso da função `cor.test` (um teste de correlação). Para o exemplo abaixo utilize o pacote `dplyr` (HADLEY *et al.*, 2018).

```
library(magrittr)
```

```
library(dplyr)
mtcars %>% filter(wt>2) %$% cor.test(hp, mpg)
```

A função `show.colors()` do pacote `DAAG` (MAINDONALD; BRAUN, 2015) permite ver os nomes dos diferentes tipos de cores: `"singles"` (simples) que não têm diferentes intensidades e `"shades"` (tons) apresenta diferentes tonalidades da cor cinza `"gray"`.

```
install.packages("fBasics")
library(fBasics)
colorTable(cex=1)
```

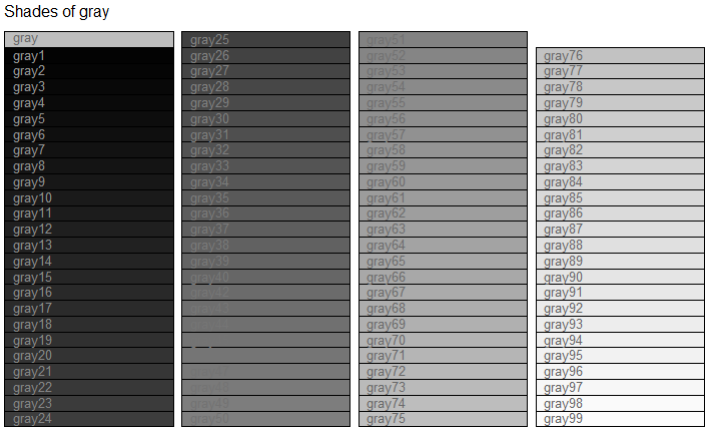


Figura 1.7: Tons de cinza.

```
show.colors("gray")
show.colors("singles")
show.colors("shades")
```

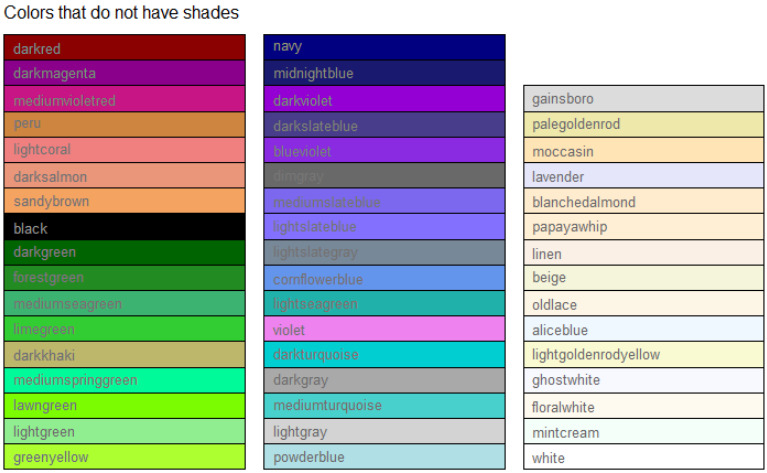


Figura 1.8: Cores sem tonalidades.

A Figura 1.9 mostra cores de diferentes tonalidades em cinco palhetas.

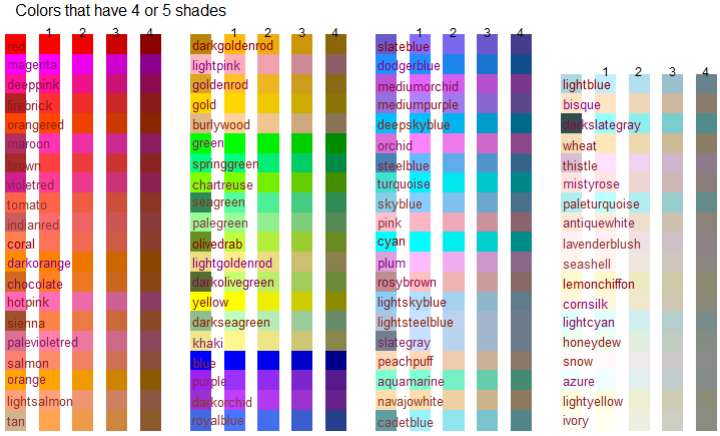


Figura 1.9: Cores que possuem 4 ou 5 tons.

1.17 Fórmulas matemáticas e caracteres especiais

Acrescenta fórmulas matemáticas ou textos com caracteres especiais nos gráficos gerados, como exemplo:

```
x<-seq(13.547,46.453, length=100)
plot(x,dnorm(x,mean=30, sd=5), xlab="x",
ylab="Densidade", main=expression(paste("y=",
frac(1,sigma*sqrt(2*pi))*e^{-frac(sum((x[i]
-mu))^2, 2*sigma^2)})), type="l",
sub=expression(paste(N(mu,sigma^2))))
```

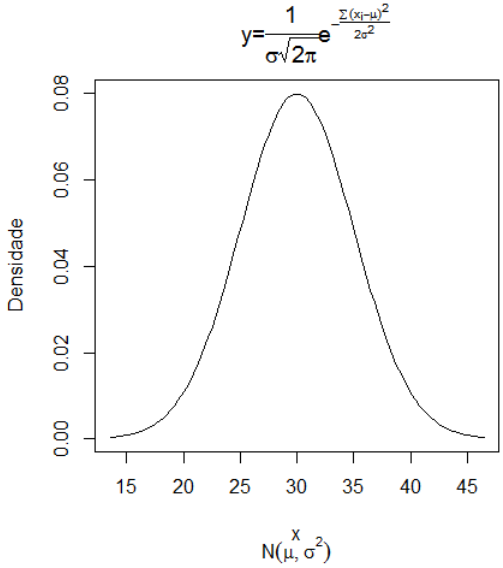


Figura 1.10: Normal com média zero ($\mu = 0$) e variância (σ^2)

```
plot(0,0,type="n",bty="n", xaxt="n", yaxt="n",
```

```
     xlab="", ylab="")
text(0,0.9,expression(chi^2==sum(sum(frac(((
O[mc] - E[mc]) - frac(1,2))^2, E[mc]), c-1,
i), m-1, j)))
```

$$\chi^2 = \sum_{m-1}^{j} \sum_{c-1}^{i} \frac{((O_{mc} - E_{mc}) - \frac{1}{2})^2}{E_{mc}}$$

Figura 1.11: Normal com média zero, $\mu = 0$ e variância, σ^2

Usaremos algumas fórmulas para exemplificar.

```
plot(0,0,type="n", bty="n", xaxt="n", yaxt="n",
xlab="",ylab="")
text(0,0.9,expression(L[t]==L[infinity](1
    -e^-k(t[f]-t[0]))))
text(0,0.5,expression(y[i] == sqrt(a[i]^2
+b[i]^2)))
text(0,0.1,expression("r"==paste(frac(
   paste(mu[max]*"S"), paste("K"[s]+"S")))))
text(0,-0.4,expression(bar(x) == frac(sum(
   x[i], n, i==1), n)))
```

Respectivamente, têm-se as seguintes fórmulas:

$$L_t = L_\infty\left(1 - e^{-k(t_r - t_0)}\right)$$

$$y_i = \sqrt{a_i^2 + b_i^2}$$

$$r = \frac{\mu_{max} S}{K_s + S}$$

$$\overline{x} = \frac{\sum_{n}^{i=1} x_i}{n}$$

Capítulo 2

Análise de agrupamento

A análise de agrupamento tem como objetivo agrupar elementos (indivíduos) em grupos homogêneos em função das similaridades entre eles. Agrupa-se os indivíduos, mas também pode agrupar variáveis. Geralmente, esses métodos são de classificação automática ou não supervisado, isto é, inexistem as respectivas saídas desejadas. Consequentemente, o próprio método aglomerativo deve se auto-organizar em relação às particularidades existentes entre os indivíduos (ou variáveis) do conjunto total da amostra, identificando subconjuntos (*clusters*) que contenham similaridades.

A quantidade de domínios de aplicação da análise de conglomerado é muito vasta. Segundo Rocha et al (2012), o termo grupo deve ser usado quando não existe qualquer informação sobre como é a organização dos dados. Assim, comumente, denomina-se agrupamento o processo pelo qual se estuda as relações de similaridade entre os indivíduos, determinando como estão organizados em grupos.

Definição 2.0.1. *É uma técnica multivariada que busca encontrar uma estrutura de* clusters *(grupos) nos dados em que os objetos pertencentes a cada* cluster *compartilhem alguma característica ou propriedade relevante para o domínio do problema em estudo, ou seja, são de alguma*

maneira similares.

Definição 2.0.2. *Pode ser entendida como um método aglomerativo que permite descobrir relações existentes entre exemplares de um conjunto de dados descritos por uma série de características (atributos descritivos).*

2.1 Técnicas de agrupamento

A maior parte de análise de agrupamento é realizada com objetivo de se tratar da heterogeneidade dos dados. A técnica de agrupamentos tem como objetivo encontrar uma estrutura de agrupamento (*cluster*). Por exemplo, a Figura 2.1 ilustra um conjunto de tipos de carnes (a) agrupados de três maneiras diferentes, em que os objetos (tipos de carnes) pertencentes a cada *cluster* compartilham alguma característica ou propriedade relevante para o domínio do problema em estudo, ou seja, são de alguma maneira similares. Visualmente identifica-se que uma das divisões dos tipos de carne e dois grupos agrupam as carnes pela forma (b) e a outra divide os tipos de carnes pelo preenchimento (c). A divisão em quatro grupos considera uma combinação dessas características (d). Cada uma dessas maneiras de agrupar os tipos de carnes é uma estrutura ou um modelo que descreve os dados e pode ter sido obtido por meio de uma algoritmo de agrupamento (FACELI *et al*., 2011).

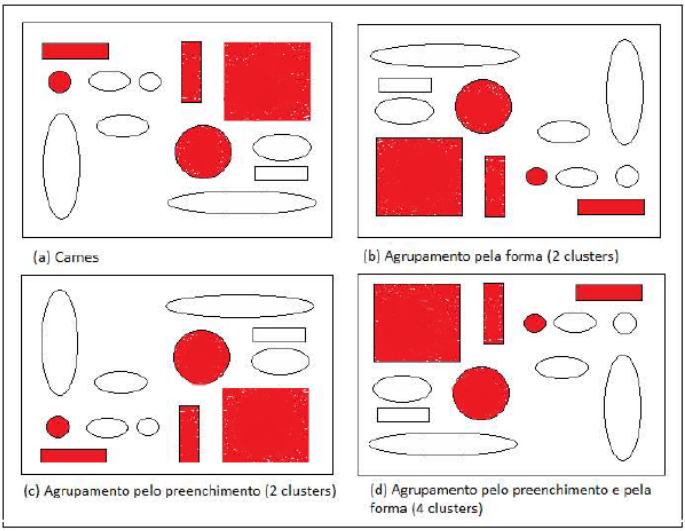


Figura 2.1: Objetos agrupados de diferentes maneiras. Fonte: Adaptado de Facelli, K et al, (2011).

2.1.1 Fases da análise de agrupamento

Na análise de agrupamento existem duas fases Figura 2.2. Na primeira fase, a partir da matriz de dados se constroi a matriz distância ou similaridades segundo se considera uma ou outra característica entre os dados.

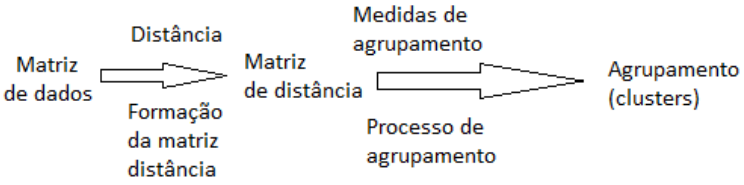


Figura 2.2: Fases da análise de agrupamento

Na segunda fase, partindo da matriz de proximidades se realiza o pro-

cesso de agrupamento de indivíduos. A Figura 2.3 apresenta três situações em que aparacem agrupamentos de diferentes formas e, ainda em todas elas aparece implícita uma noção de distância entre grupos, a segunda situação sugere que os indivíduos mais próximos não deveriam formar parte do mesmo agrupamento a menos que considere como distância entre agrupamentos outra distinta da tradicional distância euclidiana.

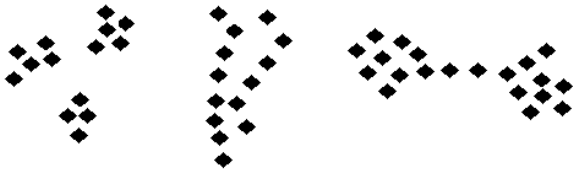


Figura 2.3: Proximidades de pontos para agrupamentos

Nesse exemplo mostra-se também o conveniente que são as representações gráficas na formação dos agrupamentos.

2.1.2 Proximidade

O sentido positivo e verdadeiro de proximidade pode ser melhor aquilatado se levarmos em conta a experiência de que a pessoa humana não nasce na natureza, mas do útero materno. Isto é, nasce em outro e é recebida em seus braços. A maior parte dos métodos de agrupamento requer que a matriz de proximidades entre objetos seja previamente obtida. A proximidade é o termo utilizado para indicar ou similaridade ou dissimilaridade, que é medida pelas distâncias.

As medidas de similaridade ou dissimilaridade satisfazem algumas propriedades. Vejamos algumas:

1. $d(x_i,x_i) = 0 \; \forall \; x_i$ (a distância entre um ponto do nível de ansiedade a ele mesmo é zero).
2. $d(x_i,x_j) = d(x_j,x_i)$ (simetria, ou seja, a distância entre x_1 e x_2 é mesma que x_2 e x_1.

3. $d(x_i,x_j) \geq 0 \ \forall \ x_i$ e x_j (Distância entre dois pontos num espaço é sempre positiva independente da direção).

4. $d(x_i,x_j) = 0$ somente se $x_i = x_j$ (distância entre dois objetos é zero, logo os dois objetos estão no mesmo ponto e posição).

5. $d(x_i,x_k) \leq d(x_i,x_j) + d(x_j,x_k) \ \forall \ x_i,x_j$ e x_k (desigualdade triangular).

Em geral, a palavra distância pode fazer referência de uma métrica, semi-métrica e quase-métrica. Vejamos cada uma delas:

- Semi-métrica é uma dissimilaridade que cumpre:
  $d(x_i,x_j) = d(x_i,x_k) + d(x_k,x_j)$.
- Métrica é uma semi-métrica que cumpre:
  $d(x_i,x_j) = 0 \Leftrightarrow i = j$, para todo i, j.
- Ultra-métrica é uma dissimilaridade que cumpre:
  $d(x_i,x_j) \neq max(x_{ik},x_{kj})$, para todo i, j, k.

Segundo a classificação de Sneath e Sokal (1973) existem quatro grandes tipos de medidas de similaridade:

Distâncias: trata-se das distintas medidas entre os pontos do espaço definido pelos indivíduos. Medidas inversas das similaridades, isto é, dissimilaridade.

Coeficiente de associação: utiliza-se quando se trabalha com dados qualitativos. Estas medidas são, basicamente, uma forma de medir a concordância ou conformidade entre os estados de duas colunas de dados.

Coeficiente angular: utiliza-se para medir a proporcionalidade e independência entre os vetores que definem os indivíduos. O mais comum é o coeficiente de correlação aplicado a variáveis contínuas.

Coeficiente de similaridade probabilística: mede a homogeneidade do sistema por participantes ou subpartições do conjunto dos indivíduos e incluem informações estatísticas. A ideia deste coeficiente baseia-se em relacioná-lo com diferentes classificações utilizando para elas critérios de bondade ou bons ajustes estatísticos. As principais propriedades destes coeficientes é que são aditivos, se distribuem em uma qui-quadrada e são probabilísticos. Esta última propriedade permite, naqueles casos em que é possível, estabelecer uma hipótese nula e contratá-la pelos métodos estatísticos tradicionais.

As medidas de similaridades mais utilizadas são: distância de Manhattan, distância euclidiana, distância euclidiana ao quadrado, distância de Chebyschev e Camberra.

Na caso dos coeficientes angulares seu campo de variação é -1 e +1. Os valores próximos a 0 indicam dissimilaridade entre os indivíduos e os próximos a +1 ou a -1 indicam similaridade positiva ou negativa, respectivamente.

É oportuno introduzir o conceito de proximidade entre elementos. As medidas de proximidade se diferenciam das distâncias em que quanto maior é a medida de proximidade, mais parecidos são os elementos. A proximidade de um conjunto de elementos é um número $\beta(x,y)$ que se põe em correspondência a cada par de elementos x e y e tem as seguintes propriedades:

1. $\beta(x,y)$ é uma função contínua das variáveis x e y, ou seja,

$$\lim_{x \to x_o, y \to y_o} \beta(x_o, y_o)$$

2. $\beta(x,y) = \beta(y,x)$

3. $0 \leq \beta(x,y) \leq 1$, e $\beta(x,y) = 1 \Leftrightarrow x = y$

Não confundir o elemento `x_{i}` com a coordenada x_i. As fórmulas dos principais métodos de proximidade entre os elementos $x = (x_1, x_2, \cdots, x_n)$ e $y = (y_1, y_2, \cdots, y_n)$ são:

- Cosseno: $$\beta(x,y) = \frac{\sum_{j=1}^{n} x_j y_j}{\sqrt{\sum_{j=1}^{n} x_j^2 \sum_{j=1}^{n} j_j^2}}$$

- Correlação de Pearson: $$\beta(x,y) = \frac{\sum_{j=1}^{n} (x_j - \bar{x})(y_j - \bar{y})}{(n-1)\sigma_x \sigma_y}$$

  Em que $\bar{x}$ e $\bar{y}$ são as médias dos objetos x e y, respectivamente, e σ_x e σ_y são os desvios padrões.

- Medida de Voronin: $$\beta(x,y) = \sum_{j=1}^{n} s_j \frac{\lambda_i}{n}$$

  Em que $\lambda_j = 1 - \frac{|x_j - y_j|}{max_j x_j - min_j y_j}$. É a medida de proximidade dos objetos segundo o j-*ésimo* criterio, s_j é o peso de informação do critério.

- Medida de proximidade de Zhuravliov:

$$\beta(x,y) = \sum_{j=1}^{n} \delta_{xy}^{j} \qquad \delta_{xy}^{i} = \begin{cases} 1, & \text{se} \quad |x_j - y_j| \leq \varepsilon_j, \\ 0, & \text{se} \quad c.c. \end{cases}$$

  O valor de ε_j é o umbral do j-*ésimo* critério.

2.2 Dados quantitativos em uma escala aproximadamente linear

Considere a seguinte matriz:

$$\begin{bmatrix} x_{11} & \dots & x_{1p} \\ \vdots & \vdots & \vdots \\ x_{a1} & \dots & x_{ap} \\ \vdots & \vdots & \vdots \\ x_{b1} & \dots & x_{bp} \\ \vdots & \vdots & \vdots \\ x_{n1} & \dots & x_{np} \end{bmatrix}_{n \times p}$$

Em que aparecem as $n \times p$ observações correspondentes as p variáveis (colunas) observadas nos n indivíduos (linhas). A partir desta matriz deve-se obter a matriz distância. Dita matriz distância entre os n indivíduos será uma matriz de dimensão $n \times n$ que tem a seguinte forma:

$$\begin{bmatrix} 0 & & & & \\ d_{2,1} & 0 & & & \\ d_{3,1} & d_{3,2} & 0 & & \\ \vdots & \vdots & \vdots & & \\ d_{n,1} & d_{n,2} & \dots & \dots & 0 \end{bmatrix}$$

Em que $d(i,j)$ representa uma distância entre os indivíduos i e j a qual não tem porque ser necessariamente uma distância no sentido matemático; basta que seja tal que $d(i,i) = 0$, $d(i.j) =\geq 0$ e $d(i,j) = d(j,i)$, para todo i, j. Pela simetria observa-se que $d(2,1) = d(1,2)$, $d(1,3) = d(3,1)$ etc.

$$d_{a,b} = \sqrt{\sum_{i=1}^{p}(x_{ai} - x_{bi})^2}$$

A distância Euclidiana entre os indivíduos $a = (x_{a1}, x_{a2}, \cdots, x_{ap})$ e $b = (x_{b1}, x_{b2}, \cdots, x_{bp})$ (ou seja, entre as linhas a e b).

Exemplo 2.2.1. *Considere duas variáveis métricas, altura e peso, de 5 indivíduos. Podendo escrever estas variáveis numa matriz de duas colunas de forma que*

$$\begin{bmatrix} 1.68 & 77 \\ 1.75 & 80 \\ 1.70 & 74 \\ 1.70 & 79 \\ 1.71 & 82 \end{bmatrix}$$

```
A<-matrix(c(1.68,77,1.75,80,1.70,74,1.70,79,1.71,
82), ncol=2, byrow = TRUE); A
#       [,1] [,2]
# [1,] 1.68   77
# [2,] 1.75   80
# [3,] 1.70   74
# [4,] 1.70   79
# [5,] 1.71   82
D<-dist(A, method="euclidean", diag=TRUE,
   upper=FALSE); D
#          1        2        3        4        5
# 1 0.000000
# 2 3.000817 0.000000
# 3 3.000067 6.000208 0.000000
```

```
# 4 2.000100 1.001249 5.000000 0.000000
# 5 5.000090 2.000400 8.000006 3.000017 0.000000
```

A distância euclidiana entre o indivíduo 1 e o indivíduo 2 será então:

$$\begin{aligned} d_{1,2} &= \sqrt{\sum_{i=1}^{2}(x_{11}-x_{21})^2+(x_{12}-x_{22})^2} \\ &= \sqrt{(1.68-1.75)^2+(77-80)^2} = 3.000817. \end{aligned}$$

De forma semelhante, calculam-se as distâncias entre os outros pares de indivíduos como se observa no objeto *D* na saída do *R*. Se `diag="FALSE"` não aparecia a diagonal nula na matriz distância. Outros métodos de distâncias seriam: `"maximum"`, `"manhattan"`, `"canberra"`, `"binary"` ou `"minkowski"`. Para o Manhattan seria:

```
DM<-dist(A,method="manhattan",diag=TRUE,upper=FALSE)
DM
#      1    2    3    4    5
# 1 0.00
# 2 3.07 0.00
# 3 3.02 6.05 0.00
# 4 2.02 1.05 5.00 0.00
# 5 5.03 2.04 8.01 3.01 0.00
abs(1.68-1.75) + abs(77-80)  #Distância de Manhattan
```

Já a distância de Manhattan, considerando também os indivíduos 1 e 2 seria:

$$\begin{aligned} d_{1,2} &= \sum_{i=1}^{2} | x_{11} - x_{21} | \\ &= | 1.68 - 1.75 | + | 77 - 80 | = 3.07 \end{aligned}$$

Definição 2.2.1. *Define-se esta distância de Mahalanobis entre um ponto e seu vetor de medias por:*

$$d_i^2 = (x_i - \bar{x}) S^{-1} (x_i - \bar{x})$$

A distância de Mahalanobis entre os indivíduos a e b tem a seguinte forma genérica:

$$D_{ab} = \sqrt{(X_a - X_b)' S^{-1} (X_a - X_b)}$$

Em que D_{ab} é a distância de Mahalanobis entre os indivíduos a e b, X_a vetor de característica do indivíduo a, X_b vetor de característica do indivíduo b e S^{-1} é a inversa da matriz de covariância amostral.

O método de Mahalanobis não apenas executa um processo de padronização sobre os dados, estabelecendo uma escala em termos de desvios-padrão, mas também soma a variância-covariância acumulada dentro dos grupos, o que ajusta as inter correlações entre as variáveis. Conjuntos de variáveis altamente inter-correlacionadas em análise de agrupamentos podem implicitamente dar mais peso a um conjunto de variáveis nos procedimentos de agrupamento. A função `dist` mostra como se encontra a distância de Mahalanobis no *R*.

```
dist <- function(X,matriz_distancia, casas_decimais)
 {
 if(!is.matrix(X))
 {
 alarm(); warning('X deve ser uma matriz numérica')
 }
 else
 {if((matriz_distancia=='Mahalanobis')||(
    matriz_distancia=='mahalanobis'))
 {
n< ncol(X)   # número de indivíduos (objetos)
p <- nrow(X) # número de variáveis
X <-round(sqrt(X),casas_decimais)
# Transformação da matriz
tX <- t(X)  # Transposta da matriz
sigma<-round(cov(tX),casas_decimais)
# matrix(NA,nrow=p,ncol=p)
inv.sigma <- solve(sigma) # matriz inversa
D2<-matrix(0,ncol=n,nrow=n) #D2 recebe as distâncias
A <- matrix(0,ncol=n,nrow=p) # Matriz de zeros
    for(j in 1:n){for(k in 1:n){
 D2[j,k] <-t(X[,j]-X[,k])%*%inv.sigma%*%(X[,j]
          -X[,k])
          }  #k
          }  #j
    return(list(D2=round(D2,casas_decimais)))
    }}
    }
# Devolve a matriz distancia de Mahalanobis
```

```
# Atribui o nº de linhas e colunas da matriz
X=matrix(X, nrow=?,ncol=?, byrow=TRUE)
dist(X, matriz_distanci='mahalanobis',
casas_decimais =1)
```

Observa-se que esta distância está em função de médias e variâncias das variáveis métricas. Para o caso considerado $p = 2$, pode-se escrever esta distância em função também da correlação entre as duas variáveis da seguinte forma:

$$\frac{1}{(1-r^2)}\begin{bmatrix} s_1^{-2} & -rs_1^{-1}s_2^{-1} \\ -rs_1^{-1}s_2^{-1} & s_2^{-2} \end{bmatrix}$$

A distância de Mahalanobis (ao quadrado) entre dois indivíduos seria:

$$d^2_{Man} = \frac{1}{(1-r^2)}\left[\frac{(x_{a1}-x_{a2})^2}{s_1^2}\frac{(x_{b1}-x_{b2})^2}{s_2^2} - 2r\frac{(x_{a1}-x_{a2})(x_{b1}-x_{b2})}{s_1 s_2}\right]$$

Se $r = 0$ esta distância reduz-se a distância euclidiana padronizada pelos desvios padrões.

```
A<-matrix(c(1.68,77,1.75,
80,1.70,74,1.70,79,1.71,82), ncol=2, byrow=T)
#      [,1] [,2]
# [1,] 1.68   77
# [2,] 1.75   80
# [3,] 1.70   74
# [4,] 1.70   79
# [5,] 1.71   82
mu=apply(A,2,mean)
sigma1=var(A); sigma1
```

```
#         [,1]  [,2]
# [1,] 0.00067 0.036
# [2,] 0.03600 9.300
for(i in 1:5){
   d<-matrix(c(A[i,]-mu), ncol=2)
   D[i]<-d %*% solve(sigma1)%*% t(d)
   D[1:5]
}
D
#           1         2         3         4         5
# 1 0.0000000
# 2 1.1716312 0.0000000
# 3 2.6913880 1.6620061 0.0000000
# 4 2.2354610 4.5434066 5.5125367 0.0000000
# 5 0.2395137 1.6394134 1.7090418 2.1420869 0.000000
```

A matriz distância apresenta em cada célula o valor do coeficiente calculado para os elementos posicionados nas respectivas linha e coluna. Essa matriz é quadrada, possuindo dimensão máxima $n \times n$, com n representando o número de elementos envolvidos. Possui diagonal principal nula (correspondente ao valor da distância de um elemento a si mesmo), assim pode ser representada numa matriz triangular superior ou inferior, cujos elementos são zeros ou vazios. Nessa representação é `dist(x, method="euclidean", diag=FALSE, upper=FALSE, p=2)`. *X* é a matriz de dados quantitativos, para diagonal `FALSE` não se apresenta zero na diagonal principal, para `upper=FALSE` a matriz será triangular inferior, pois o `upper` (superior) é falso.

2.3 Objetivos da aprendizagem

O agrupamento é um campo difícil de investigação em que as suas potenciais aplicações impõem requisitos específicos. Os requisitos principais que um algoritmo ideal de agrupamento satisfazem têm as seguintes características segundo (CATENA, A., 2003):

1. Distinguir análise de agrupamento das outras técnicas de redução de dados que permitem agrupar elementos (objetos) ou variáveis (componentes).

2. Conhecer em que contexto de pesquisa é apropriado usar análise de agrupamento.

3. Conhecer que tipo de questões de pesquisa a técnica de agrupamento pode ajudar.

4. Conhecer as medidas de semelhança, distância e associação.

5. Conhecer os tipos básicos de análise de agrupamento.

6. Conhecer alguns algoritmos fundamentais de agrupamento hierárquico e do não hierárquico.

7. Conhecer os procedimentos de valorização das soluções obtidas na análise.

8. Saber avaliar e interpretar os resultados da análise de agrupamento.

2.4 Procedimentos e técnicas na análise de agrupamento

Nas seções anteriores foram úteis para obtermos uma imagem sobre como ocorre um processo de agrupamento. Adicionando a essa experiência a literatura consultada é possível enumerar, nesse momento, as seguintes etapas a serem seguidas para a aplicação de análise de agrupamento (BUSSAB; MIAZAKI; ANDRADE, 1990, BARROSO; ARTES, 2003):

1. Definição dos objetivos da Análise de Agrupamento, obtenção dos dados e tratamento, se necessário, dos mesmos.

2. Escolha da Técnica de Agrupamento e da medida de distância (coeficiente de similaridade ou dissimilaridade) a ser utilizada.

3. Formação dos grupos a partir das definições efetuadas no item anterior.

4. Validação, avaliação e interpretação dos resultados obtidos.

No modelo hierárquico, os indivíduos vão sendo incluídos progressivamente no agrupamento, formando uma estrutura hierárquica de múltiplos níveis. Podem ter duas abordagens:

1. Aglomerativo: a cada instante forma seu próprio grupo e segue progressivamente mesclando com demais grupos, até existir somente um grupo restante.

2. Divisivo: inicia com todos os indivíduos e se divide em múltiplos grupos.

2.5 Sintaxes e parâmetros

O agruapemento hierárquico aglomerativo em R é realizado em dois passos. No primeiro deles, calcula-se a matriz de similaridade com o uso da função $dist(X, methods =)$, disponível no pacote stats (R Core Team, 20015), nativo no R. As sintaxe e parâmetros para as funções: $dist()$, *hclust*, *cutree* e *rect.hclust* são respectivamente:

```
MD<-  dist(x, method = "euclidean",
diag = FALSE, upper = FALSE)
```

A matriz X é um data.frame que contém dados métricos para realização dos cálculos de distâncias. O método é a medida de distância a ser aplicada que pode ser: Euclidiana, Mahalanobis, etc.

```
H<-hclust(MD, method = "", members = NULL)
```

Os métodos aglomerativos podem ser: single (ligação simples = vizinho mais próximo = distância menor), complete (vizinho mais distante = distância maior), ward.D2 (método de Ward), average (distância média), mcquitty, median (método da mediana) e centroid. A visualização do resultado de agrupamento por meio de dendrogramas pode ser feita com o uso da função $plot()$:

```
plot(H, main = "Método aglomerativo",
 xlab="", ylab="")
```

Ainda, é possível representar os grupos no dendrograma usando a função *cutree*. O agrupamento é a variável resultante da aplicação da função *hclust*. O número inteiro *k* representa a quantidade de grupinhos desejados no dendrograma, geralmente é representado por uma linha horizontal na coordenada de distância no dendro-

grama.

```
Grupos_dentro_agrupamento<- cutree (H, k)
```

A função *rect.hclust* imprime o dendrograma.

```
Representação_agrupamento<- rect.hclust(H, k,
border="blue")
```

2.5.1 Algoritmo hierárquico

Passos para o algoritmo hierárquico:

Passo 1 Cada objeto (elemento) é declarado um grupo inicial. Estes são os grupos da *etapa 0*.

Passo 2 Os dois grupos mais próximos se unem e esta fusão é declarada um grupo. Como distância entre os grupos utiliza-se a distância entre os objetos. Os grupos obtidos (o novo grupo e todos os grupos declarados anteriormente) são grupos da *etapa 1*.

Passo n Suponha que estão dados os grupos da *(n-1)-ésima etapa*. Os dois grupos mais próximos se unem para formar um novo grupo. Como distância entre os grupos utiliza-se distância entre grupos escolhida previamente. Deste modo obtém-se os grupos da *n-ésima etapa*.

Passo N Todos os objetos ficam unidos em um único grupo.

Neste algoritmo, o usuário pode eleger o número de grupos necessários. O processo de agrupamento detém-se em quanto

se obtém g grupos. É cômodo representar o processo de formação dos grupos em forma de dendrograma, no qual pode-se apreciar a união dos objetos para formação dos grupos.

Algoritmo hierárquico - A Glomerative NESting (AGNES)

É implementada a partir da abordagem aglomerativa. O algoritmo para agrupamento hierárquico aglomerativo - métodos AGNES é o seguinte (SILVA, L.A et al, 2016):

Parâmetros de entrada. Passos:

- X_tr: conjunto numérico, ou seja, $X_{tr} = (x_{tr}), i = 1, \cdots, n$.
- dist: Método de medida de distância. Abordagem para cálculo da distância entre grupos;

Parâmetros de saída: H dendrograma. Passos:

- Passo 1: calcula a matriz de similaridade;
- Passo 2: aloque cada elemento x_i de X_{tr}, em grupos distintos, criando os nós folhas da árvore D;
- Passo 3: **enquanto** há possibilidade de fusão de grupos **faça!**.

  Passo 3.1 verifique a distância entre todos os pares de grupos, usando a matriz de similaridade ou calculando a distância entre os centros de gravidade dos grupos (a depender da abordagem escolhida);

Passo 3.2 encontre o par de grupos mais similares e os transforme em uma único grupo, criando um nó interno na hierarquia da árvore **D**;

Ainda o algoritmo hierárquico aglomerativo de forma bem sintetizado foi apresentado em (FACELI, K *et al*,2011):

Entrada: Uma matriz distância $(D_{n\times n})entreosparesdeelementos; Umahierarqu$

Saída:

1. Alocar cada elemento em um cluster
2. **enquanto** há clusters para agrupar **faça**
3. Calcular a matriz de distância entre os pares de clusters disponíveis, utilizando uma métrica de integração
4. Combinar o par de clusters C_i e C_j mais próximos, gerando um único cluster C_{ij}
5. **fim**

Nesse algoritmo, o usuário pode eleger o número de grupos necessários. O processo de agrupamento detém-se em quanto se obtém g grupos. É cômodo representar o processo de formação dos grupos em forma de dendrograma, no qual pode-se apreciar a união dos elementos para formação dos grupos.

2.5.2 Dendrograma

Visualizar um resultado de um método é uma parte essencial de qualquer processo de análise de aglomerados. Os dendrogramas são estruturas arborescente, utilizadas para representar as junções (métodos hierárquicos) ou divisões (métodos de partição) de acordo

com diferentes níveis de hierarquia. São criadas a partir de valores provenientes de uma matriz de distâncias. Os dendrogramas são utilizados, principalmente para a observar os "saltos" (distâncias) que ocorrem na formação dos grupos, buscando detectar a formação de grupos heterogêneos. O número ideal de grupos também pode ser inferido do dendrograma, como se observa na Figura 2.4.

O passo final, numa análise de agrupamento com a técnica de hierarquização, é a interpretação do dendrograma identificando os grupos de espécies, objetos, indivíduos, etc. Há uma grande subjetividade na tomada de decisão para destacar e interpretar os grupos formados, o que é preciso muito conhecimento da realidade do estudo envolvido. Uma interpretação básica seria que os indivíduos C e D se agrupam no grupo 1, o F e G no grupo 2, os indivíduos (A,B,C,D) no grupo 5, e o grupo 4 que envolve todos os 8 indivíduos.

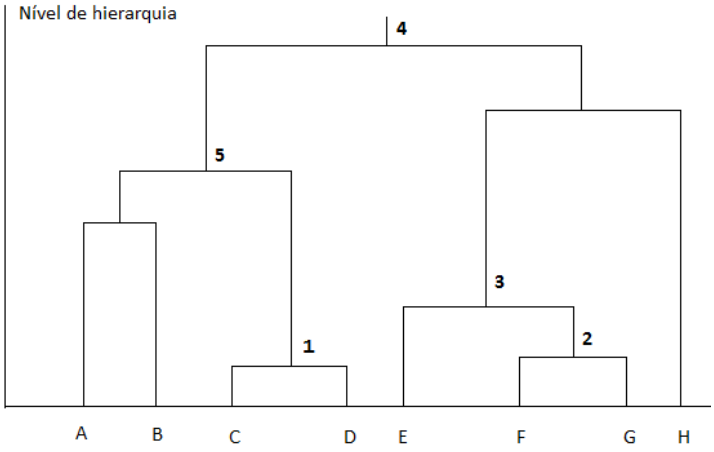


Figura 2.4: Representação de dendrograma para estudo dos indivíduos

2.5.3 Método do vizinho mais próximo ou método da união simples (*single linkage clustering ou Nearest Neighbour*)

Um indivíduo com um grupo se tiver a maior similaridade com qualquer dos elementos individuais desse grupo. Como ilustração a Figura 2.5. Dado dois grupos (NA,T) e (P,TS), a distância entre os dois é a menor das distâncias entre os dois grupos.

$$d_{(NA,T),(P,TS)} = \min(d_{NA,P}, d_{NA,T}, d_{T,P}, d_{T,TS}) = d$$

Deste modo, qualquer grupo de carnes (como exemplo) é definido como um conjunto de casos em que qualquer elemento é mais semelhante a pelo menos um outro elemento do mesmo grupo de que a qualquer elemento de outro grupo. A Figura 2.5 ilustra esta situação.

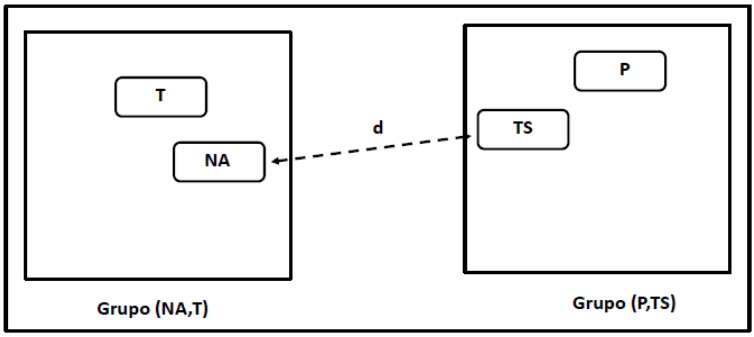


Figura 2.5: Método do vizinho mais próximo

A distância entre *clusters* é dada pela distância entre os tipos de preparo de carnes dos dois *clusters* que estão mais próximo, ou seja, a distância mínima entre quaisquer dois objetos, um de cada

clusters. A Figura 2.6(a) ilustra uma situação em que as variáveis (NA,TS) e (P,TS) estão em grupos separados, enquanto que a Figura 2.6(b) observa-se uma estrutura com 4 agrupamentos.

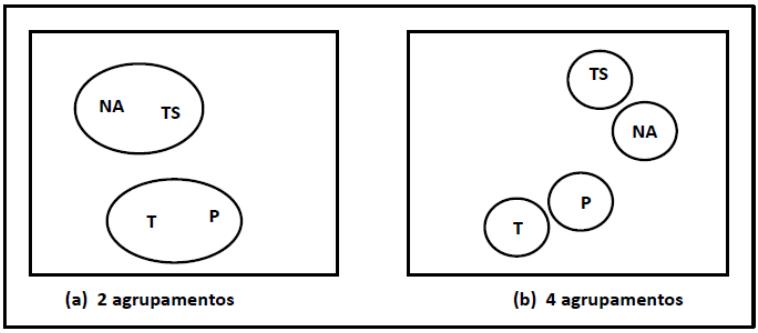


Figura 2.6: Agrupamentos em diferentes níveis

2.5.4 Método do vizinho mais distante (*Complete linkage clustering ou Furthest Neighbour*)

Muitas vezes é interessante agrupar os indivíduos mais distantes. Este método baseia-se na distância máxima (vizinho mais distante). Suponha que deseja-se agrupar indivíduos que apresentam características mais distintas. Este técnica ajudará o pesquisador a localizar as diferenças mais acentuadas. O objetivo é agrupar as maiores distâncias da matriz de distância em determinado grupo, assim é possível visualizar os elementos mais distantes agrupados nos clusters, ou seja, segmentar indivíduos ou variáveis em grupos homogêneos (de estar distantes)com base em suas próprias características, buscando assim, uma estrutura "natural distante" desses indivíduos.

$$d_{(NA,T),(P,TS)} = max(d_{NA,P}, d_{NA,T}, d_{T,P}, d_{T,TS}) = d$$

Deste modo, qualquer grupo de carnes é definido como o conjunto de casos em que qualquer elemento é mais semelhante a pelo menos um outro elemento do mesmo grupo de que a qualquer elemento de outro grupo. A Figura 2.5 ilustra esta situação.

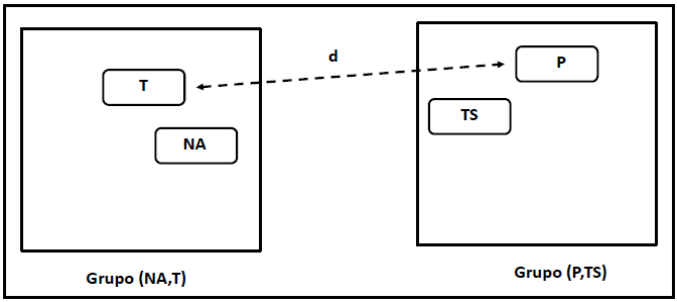


Figura 2.7: Esquema do método do vizinho mais distante

2.5.5 Método do centroide

Este método calcula a distância entre dois agrupamento como a distância entre médias para todas as variáveis. Baseia na distância entre os centroides priorizando a menor distância entre eles. Identifica os dois grupos separados (NA, T) e (P, TS) pela menor distância entre os dois mais próximos e os coloca no mesmo agrupamento. Segundo (HAIR *et al*, 2005) os centroides são valores médios das observações sobre as variáveis estatística de agrupamento. Nesse método, toda vez que indivíduos são reunidos, um novo centroide é considerado

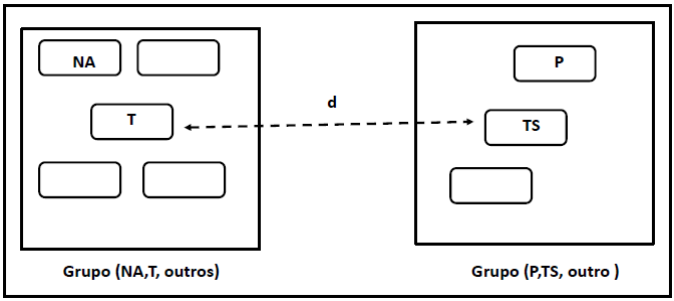


Figura 2.8: Esquema do método do centroide

Os pesos são proporcionais aos tamanhos dos agrupamentos, assim o número de indivíduos de NA, T e TS, por exemplo, que são n_i, n_j e n_k, respectivamente. A distância é definida da seguinte maneira.

$$d_{(TS),(NA,T)} = \left(\frac{n_i}{n_i+n_j}\right) d(n_i,n_k) + \left(\frac{n_j}{n_i+n_j}\right) - \left(\frac{n_i n_j}{(n_i+n_j)^2}\right) d(n_i,n_j)$$

2.5.6 Método de ward ou variância mínima

Este método considera a distância euclidiana ao quadrado como medida de dissimilaridade, ou seja, aqueles indivíduos que proporcionam a menor soma de quadrado dos desvios. Pode-se escrever o quadrado da distância de um ponto z a um centro de agrupamento x_i e x_j da seguinte forma:

$$d(x-c) = \left(\frac{1}{(m_i+m_j)(m_i d(x_i,z))^2}\right) + m_j d(x_i,z)^2 - \left(\frac{m_i m_j}{m_i+m_j}\right) - d(x_i,x_j)^2 \left(\frac{n_i n_j}{(n_i+n_j)^2}\right) d(n_i,n_j)$$

Em que x_i e x_j são dois elementos de massas m_i e m_j, respecti-

vamente.

Este método consiste em encontrar os indivíduos x_i e x_j com a condição de que tenha variância mínima em lugar de ser os indivíduos mais próximos.

$$MI_{ij} = \left(\frac{(m_i m_j)}{(m_i + m_j)}\right) \|x_i - c\|^2 = \left(\frac{(m_i m_j)}{(m_i + m_j)}\right) d(x_i - x_j)^2$$

2.6 Pressupostos da análise de agrupamento

Os pressupostos básicos a ser considerados na análise de agrupamento serão: a representatividade da amostra, o impacto da multicolinearidade entre as variáveis e outliers.

2.6.1 Pontos extremos (*outliers*)

Os pontos extremos podem muito afetar os valores da matriz distância e consequentemente os métodos de agrupamento. É importante verificar a existência destes pontos e eliminá-los. Usando o pacote `extremevalues` (VAN DER LOO, 2010) e fazendo uma simples mudança em `y<-c(min(y), y, max(y))` para `y<-c(y)` pode-se detectar os *outliers* facilmente. A Figura 2.9 ilustra a situação em que se apresentam dois pontos extremos, gráfico quantil-quantil *Q-Q plot* e o gráfico dos resíduos.

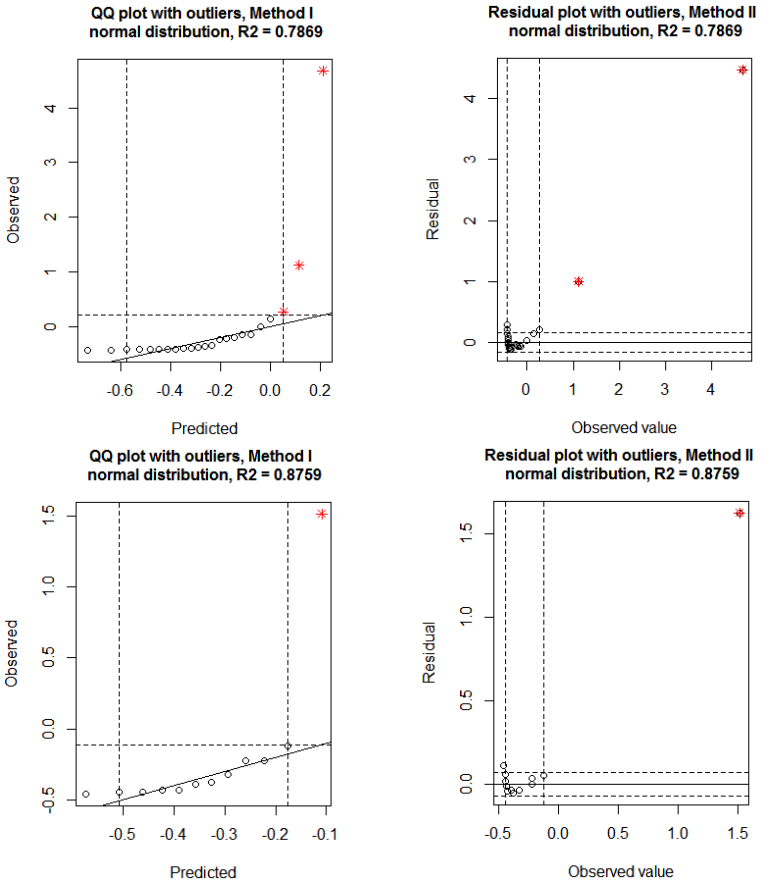


Figura 2.9: Detecção de outliers

```
# Detectar outliers
library(extremevalues)
D1<-Ds1; D1 # variável D1
# [1] -0.4388 -0.1548 -0.4299 -0.3903 -0.3975
# [6] -0.01101 -0.4245  4.6795 -0.3472 -0.2285
# [12]-0.4191 -0.4173 -0.2015 -0.2411 1.1180
# [17]-0.4227  0.2694  0.1400 -0.4209 -0.3921
```

```
# [23]-0.4263 -0.1476 -0.3705
y1 <- c(D1); y1
# [1] -0.4388 -0.1548 -0.4299 -0.3903 -0.3975
# [6] -0.0110 -0.4245  4.6795 -0.3472 -0.2285
# [12]-0.4191 -0.4173 -0.2015 -0.2411 1.1180
# [17]-0.4227  0.2694  0.1400 -0.4209
# [22]-0.3921 -0.4263 -0.1476 -0.3705
K<-getOutliers(y1,method="I",distribution="normal")
L<-getOutliers(y1,method="II",distribution="normal")
par(mfrow=c(1,2))
outlierPlot(y1,K,mode="qq")
outlierPlot(y1,L,mode="residual")
D2<-Ds2; D2 # variável D2
# [1] -0.3779 -0.4605 -0.2225 -0.1229 -0.2249
# [6] -0.4289 -0.3245 -0.4435 -0.4459  1.5136
# [11]-0.4337 -0.3900
y2 <- c(D2); y2
# [1] -0.3779 -0.4605 -0.2225 -0.1229 -0.2249
# [6] -0.4289 -0.3245 -0.4435 -0.4459  1.5136
# [11]-0.4337 -0.3900
K<-getOutliers(y2,method="I",distribution="normal")
L<-getOutliers(y2,method="II",distribution="normal")
par(mfrow=c(1,2))
outlierPlot(y2,K,mode="qq")
outlierPlot(y2,L,mode="residual")
```

Os efeitos de um só ponto atípico podem ser graves, pois distorcem as médias e o desvio padrão das variáveis e destroem as relações existentes entre elas influenciando posteriormente fortemente nos dendrogramas. Para ilustrar o problema do valor atípico, suponha que em uma amostra multivariada de tamanho n introduz um valor atípico, **a**, que é um vetor de falsas observações. Denomina-se

$\bar{x}$ e S o vetor de médias e matriz de covariância sem o dado atípico e $\bar{x}_k$ e S_k aos da amostra contaminada com este dado atípico. Assim, a média e a matriz de covariância são:

$$\bar{x}_c = \bar{x} + \frac{(a-\bar{x})}{n+1} \tag{2.1}$$

$$S_c = \frac{n}{n+1}S + \frac{(a-\bar{x})(a-\bar{x})'}{n+1}\left(\frac{n}{(n+1)}\right) \tag{2.2}$$

As fórmulas 2.1e 2.2 indicam que apenas um dato atípico pode afetar muito o valor da média e todas as variâncias e covariâncias entre as variáveis. No exemplo abaixo ilustra-se tal situação.

```
n<-4 #indivíduos com 3 variáveis
M<-matrix(c(1,2,4,3,6,5,4,3,4,2,3,5),ncol=3,byrow=T)
#       [,1] [,2] [,3]
# [1,]    1    2    4
# [2,]    3    6    5
# [3,]    4    3    4
# [4,]    2    3    5
Media_sem_atipico<-apply(M,2,mean);Media_sem_atípico
# [1] 2.5 3.5 4.5
Msa<- Media_sem_atipico;Msa
# [1] 2.5 3.5 4.5
a<-matrix(c(300,245,700), ncol=3, byrow=T);a
#      [,1] [,2] [,3]
# [1,]  300  245  700
Ma<-rbind(M,a); Ma  #com atípicos
#       [,1] [,2] [,3]
# [1,]    1    2    4
# [2,]    3    6    5
```

```
# [3,]    4    3    4
# [4,]    2    3    5
# [5,]  300  245  700
Media_com_atipico<-apply(Ma,2,mean);Media_com_atípico
# [1]  62.0  51.8 143.6
Xc<-Media_contaminada<-Msa+((Ma-Msa)/(n+1));Xc
# [,1] [,2]  [,3]
# [1,]  2.2  4.0   3.6
# [2,]  3.4  3.2   4.6
# [3,]  4.4  3.4   2.8
# [4,]  2.4  4.2   3.8
# [5,] 62.8 51.0 143.6
```

Observa-se como o valor atípico influencia os dados originais. Semelhantemente, a matriz de covariância, pois a média influencia também a variância.

2.6.2 Contraste de dados atípicos

No teste de igualdade de média pode aplicar-se para verificar se uma observação de uma amostra de dados normais tem algum valor atípico. A hipótese nula será que todos provem de uma distribuição normal. A hipótese alternativa será que um dado suspeito (x_i) tenha sido gerado por outra população desconhecida. Para caracterizar a população da hipótese alternativa podemos supor que a média é distinta e a variância é a mesma, ou que a média é a mesma e a variância distinta. Vamos considerar apenas o caso em que a média é distinta, mas a matriz de covariância é a mesma. A hipótese a ser testada será:

$$\begin{cases} H_0 : E(x_i) = \mu \\ H_1 : E(x_i) = \mu_i = \mu \end{cases}$$

A função de máxima verossimilhança sob H_o será:

$$D^2_{(x_i,\bar{x}_{(i)})} = (x_i - \bar{x}_{(i)})' S^{-1}_{(i)} (x_i - \bar{x}_{(i)})$$

Sob H_a como a estimação de μ é x_i, a estimação da variância será:

$$S_{(i)} = \frac{1}{n-1} W_i \qquad \text{e} \qquad W_{(i)} = \sum_{j=1, j\neq i}^{n} (x_j - \bar{x}_{(i)})(x_j - \bar{x}_{(i)})'$$

Isto é estimação da soma de quadrado dos resíduos, e $\bar{x}_{(i)}$ é a média das observações, em ambos casos eliminando a observação x_i. A diferença que contribui será então, particularizada em:

$$\lambda_o = m \log \frac{T}{W} = m \log \frac{W+B}{W} = m \log |I + W^{-1} B|$$

Como $|I + A| = \Pi(1 + \lambda_i)$, em que λ_i são os autovetores de A, esta estatística se reduz a

$$\lambda_o = m \sum \log(1 + \lambda_i) \tag{2.3}$$

Em que λ_i são os autovalores da matriz $W^{-1}B$.

2.7 Cálculo da variação dentro do grupo, entre grupo e Total

Facilmente no R, se encontra a matriz de variação cruzada entre (B) grupo, dentro (W) do grupo, inversa de W, autovalores e autovetores associados. Vamos mostrar como se calculam as matrizes B, W e os autovalores e autovetores de $W^{-1}B$ (cf. PENÃ, 2002).

Supondo que se tem extraído uma amostra em cada grupo, e, mediante agregação, obtemos o total da amostra designadas por $n = n_1 + n_2 + \ldots + n_G$. O vetor das médias amostras globais $\bar{y}$(ou seja, de todos os grupos) se obtém somando para cada variável todos os valores da amostra e dividindo pelo total da amostra. Dentro de cada grupo poder-se obter o correspondente vetor de médias amostrais $\bar{y}_g$.

F. de variação	G.L.	Matriz de soma de quadrado
Between (B)	G -1	$B = n_g \sum_g (\bar{X}_g - \bar{X})(\bar{X}_g - \bar{X})'$
Within (W)	n - G	$W = \sum_g \sum_u (\bar{X}_{gu} - \bar{X}_g)(\bar{X}_{gu} - \bar{X}_g)'$
Total	T	$T = \sum_g \sum_u (\bar{X}_{gu} - \bar{X})(\bar{X}_{gu} - \bar{X})'$

A matriz da soma dos quadrados e produtos cruzados entre os grupos se deve a influência do fator. Alguns autores o chamam de matriz da soma dos quadrados e produtos cruzados do fator. Já a matriz da soma dos quadrados e produtos cruzados dentro dos gru-

pos (W, do inglês *within)*, a componente intra grupo é a matriz da soma de quadrado e produto cruzados dos desvios entre cada dado e a média do seu grupo. Chama-se também de soma de quadrados e produtos cruzados residual.

A matriz W pode obter por agregação das matrizes da soma de quadrado e produtos cruzados calculados para cada grupo: $W = W_1 + W_2 + \ldots + W_G$, sendo que W_g é a soma de quadrado e produto cruzado no grupo $g = 1, \ldots, G$.

A decomposição da soma de quadrado e produto cruzado total, ou matriz T, se apresenta da seguinte forma:

$$T = B + W \tag{2.4}$$

Exemplo 2.7.1. *Com objetivo de exemplificar como se calcula estas três matrizes. Considere duas variáveis Y_1 e Y_2 e três grupos retirados de tamanhos aleatórios: $n_1 = 4$, $n_2 = 3$ e $n_3 = 5$ conforme se observa na Tabela 2.1.*

Tabela 2.1: Dados hipotéticos de duas variáveis em três grupos

Grupo I		Grupo II		Grupo III	
Y_1	Y_2	Y_1	Y_2	Y_1	Y_2
4	5	6	6	11	8
2	3	8	8	9	7
6	3	8	3	10	12
4	5	2	9	8	6
6	2	5	9	7	11
8	9			10	8
				9	9.5
$\bar{y}_{11} = 5$	$\bar{y}_{21} = 4.5$	$\bar{y}_{12} = 5.8$	$\bar{y}_{22} = 7$	$\bar{y}_{13} = 9.1428$	$\bar{y}_{23} = 8.7857$

```
manova.data<-data.frame(group=as.factor(rep(1:3,
             c(6, 5, 7))); manova.data
X1<-c(4,2,6,4,6,8,6,8,8,2,5,11,9,10,8,7,10,9)
X4<-c(5,3,3,5,2,9,6,8,3,9,9,8,7,12,6,11,8,9.5))
with(manova.data, tapply(X1, group, mean))
with(manova.data, tapply(X4, group, mean))
```

Vamos encontrar o produto de matrizes: $A = W^{-1}B$.

- Cálculo da soma de quadrado e produto cruzado dentro do grupo (W)

Para o caso de três grupos:

$$W = W_1 + W_2 + W_3$$

Sendo W_g, com $g = 1,2,3$ a soma de quadrados e produtos cruzados para os 3 grupos, respectivamente.

$$W_1 = \sum_{i=1}^{6} (Y_{1g_i} - \bar{Y}_1)(Y_{1g_i} - \bar{Y}_1)'$$

Então temos:

$$\begin{aligned}
W_1 = & \begin{bmatrix} 4-5 \\ 5-4.5 \end{bmatrix} \begin{bmatrix} 4-5 & 5-4.5 \end{bmatrix} + \begin{bmatrix} 2-5 \\ 3-4.5 \end{bmatrix} \begin{bmatrix} 2-5 & 3-4.5 \end{bmatrix} \\
& + \begin{bmatrix} 6-5 \\ 3-4.5 \end{bmatrix} \begin{bmatrix} 6-5 & 3-4.5 \end{bmatrix} + \begin{bmatrix} 4-5 \\ 5-4.5 \end{bmatrix} \begin{bmatrix} 4-5 & 5-4.5 \end{bmatrix} \\
& + \begin{bmatrix} 6-5 \\ 2-4.5 \end{bmatrix} \begin{bmatrix} 6-5 & 2-4.5 \end{bmatrix} + \begin{bmatrix} 8-5 \\ 9-4.5 \end{bmatrix} \begin{bmatrix} 8-5 & 9-4.5 \end{bmatrix} \\
W_1 = & \begin{bmatrix} 22 & 13 \\ 13 & 31.5 \end{bmatrix}
\end{aligned}$$

No *R*, tem-se:

```
# [1] 13
Sa<-matrix(c(4-5, 5-4.5), ncol=2, byrow=TRUE)
Sa<-t(Sa) %*% Sa
Sb<-matrix(c(2-5, 3-4.5), ncol=2, byrow=TRUE)
Sb<-t(Sb) %*% Sb
Sc<-matrix(c(6-5, 3-4.5), ncol=2, byrow=TRUE)
Sc<-t(Sc) %*% Sc
Sd<-matrix(c(4-5, 5-4.5), ncol=2, byrow=TRUE)
Sd<-t(Sd) %*% Sd
Se<-matrix(c(6-5, 2-4.5), ncol=2, byrow=TRUE)
Se<-t(Se) %*% Se
Sf<-matrix(c(8-5, 9-4.5), ncol=2, byrow=TRUE)
Sf<-t(Sf) %*% Sf
```

```
S<- Sa+Sb+Sc+Sd+Se+Sf; S
#      [,1] [,2]
# [1,]   22 13.0
# [2,]   13 31.5
```

Calculando as variâncias e covariâncias do grupo 1, tem-se:

```
S11<-(4-5)^2 + (2-5)^2 + (6-5)^2 + (4-5)^2 +
(6-5)^2 + (8-5)^2; S11
[1] 22
S22<- (5-4.5)^2 + (3-4.5)^2 + (3-4.5)^2 +
(5-4.5)^2 + (2-4.5)^2 + (9-4.5)^2; S22
[1] 31.5
SS21<-(4-5)*(5-4.5)+(2-5)*(3-4.5)+(6-5)*(3-4.5)+
(4-5)*(5-4.5)+(6-5)*(2-4.5)+(8-5)*(9-4.5); SS21
[1] 13
```

Semelhantemente, calcula-se as matrizes W_2 e W_3:

```
S21<-(6-5.8)^2 + (8-5.8)^2 + (8-5.8)^2 +
     (2-5.8)^2 + (5-5.8)^2); S21
# [1] 24.8
S21<-(6-5.8)^2 + (8-5.8)^2 + (8-5.8)^2 +
     (2-5.8)^2 + (5-5.8)^2; S21
# 24.8
S22<-(6-7)^2+(8-7)^2+(3-7)^2+(9-7)^2+(9-7)^2;S22
[1] 26
SS22<-(4-5.8)*(6-7)+(8-5.8)*(8-7)+(8-5.8)*(3-7)
      +(2-5.8)*(9-7)+(5-5.8)*(9-7); SS22
[1] -14
SS22<-(6-5.8)*(6-7)+(8-5.8)*(8-7)+(8-5.8)*(3-7)
      +(2-5.8)*(9-7)+(5-5.8)*(9-7); SS22
```

```
# [1] -16
```

Encontrando W_2 e W_3, respectivamente tem-se

$$W_2 = \begin{bmatrix} 24,8 & -16 \\ -16 & 26 \end{bmatrix} \quad \text{e} \quad W_2 = \begin{bmatrix} 24,8 & -16 \\ -16 & 26 \end{bmatrix}$$

Finalmente a matriz agregada de soma de quadrados e produtos cruzados dentro dos grupos (residual) será:

$$W = W_1 + W_2 + W_3 = \begin{bmatrix} 57.6571 & -3.7857 \\ -3.7857 & 85,4285 \end{bmatrix}$$

- Cálculo da soma de quadrados e produtos cruzados entre grupos.

O cálculo da matriz B é feito da seguinte forma:

$$B = \sum_{g=1}^{G} n_g (\bar{Y}_{1g} - \bar{Y}_1)(\bar{Y}_{pg} - \bar{Y}_p)'$$

o vetor de médias seria

$\bar{Y}_1$=123/18=6,8333 e $\bar{Y}_2$=123.5/18=6.8611 ou $\bar{Y}=\begin{bmatrix} 6,8333 \\ 6.8611 \end{bmatrix}$

A matriz B pressupõe o conhecimento das médias globais das variáveis 1 e 2 de cada grupo. Pode-se facilmente encontrar os

vetores de médias de cada grupo no *R* da seguinte forma:

```
b1<-matrix(c(5-6.8333,4.5-6.8611),ncol=1,byrow=TRUE)
b2<-matrix(c(5.8-6.8333,7-6.8611),ncol=1,byrow=TRUE)
b3<-matrix(c(9.1428-6.8333, 8.7857-6.8611),
           ncol=1, byrow=TRUE)
B<-6*(b1%*%t(b1))+5*(b2%*% t(b2))+7*(b3%*%t(b3));B
#           [,1]      [,2]
# [1,] 62.84101 56.36805
# [2,] 56.36805 59.47382
```

Portanto, a matriz *B* será:

$$
\begin{aligned}
B &= 6\begin{bmatrix} 5-6.8333 \\ 4.5-6.8611 \end{bmatrix}\begin{bmatrix} 5-6.8333 & 4.5-6.8611 \end{bmatrix} + \\
&\quad 5\begin{bmatrix} 5.8-6.8333 \\ 7-6.8611 \end{bmatrix}\begin{bmatrix} 5.8-6.8333 & 7-6.8611 \end{bmatrix} + \\
&\quad 7\begin{bmatrix} 9.1428-6.8333 \\ 8.7857-6.8611 \end{bmatrix}\begin{bmatrix} 9.1428-6.8333 & 8.7857-6.8611 \end{bmatrix} \\
B &= \begin{bmatrix} 62.84101 & 56.36805 \\ 56.36805 & 59.47382 \end{bmatrix}
\end{aligned}
$$

- Assim, a matriz de soma de quadrado e produto cruzado total (T) será:

$$T = B + W = \begin{bmatrix} 62.84101 & 56.36805 \\ 56.36805 & 59.47382 \end{bmatrix} + \begin{bmatrix} 57.6571 & -3.7857 \\ -3.7857 & 85.4285 \end{bmatrix}$$

$$T = \begin{bmatrix} 120.49811 & 52.58235 \\ 52.58235 & 144.90232 \end{bmatrix}$$

Através do seguinte código é possível encontrar a média e o desvio padrão dos grupos.

```
Media_Desvio_Grupos
<- function(Variaveis,GruposVariaveis)
{
  # Nomes das variáveis em todos os grupos
  NomesVariaveis <-names(GruposVariaveis),
  names(as.data.frame(Variaveis)))
  # cada variável dentro do grupo
  GruposVariaveis <- GruposVariaveis[,1]
  means <- aggregate(as.matrix(Variaveis)
  ~ GruposVariaveis, FUN = mean)
  names(means) <- NomesVariaveis
  print(paste("Means:"))
  print(means)
  #Matriz dentro dos grupos.
  #Encontra cada desvio padrão.
  DPs <- aggregate(as.matrix(Variaveis) ~
  GruposVariaveis, FUN = sd)
  names(DPs) <- NomesVariaveis
  print(paste("Desvio padrão:"))
  print(DPs)
  #Dentro de cada grupo se tem o tamanho da amostra
```

```
  samplesizes<-aggregate(as.matrix(Variaveis)
              ~GruposVariaveis, FUN = length)
  names(samplesizes) <- NomesVariaveis
  print(paste("Tamano da amostra:"))
  print(samplesizes)
}
colnames(dados)
Media_Desvio_Grupos(dados[2:3],dados[1]
```

Abaixo temos o código para o cálculo da variância dentro do grupo (W).

```
Variancia_Dentro_Grupo
<- function(variavel,GrupoVariavel)
{
  GrupoVariavel2
  <- as.factor(GrupoVariavel[[1]])
  levels <- levels(GrupoVariavel2)
  Num_nives <- length(levels)
  # Média e desvio de cada grupo
  Num_total <- 0
  denomtotal <- 0
  for (i in 1: Num_nives)
  {
    leveli <- levels[i]
    levelidata <- variavel[GrupoVariavel==leveli,]
    levelilength <- length(levelidata)
    # Encontrando o desvio padrão do grupo i
    Sdi <- sd(levelidata)
    Numi <- (levelilength - 1)*(Sdi * Sdi)
    Denomi <- levelilength
    Num_total <- Num_total + Numi
```

```
    denomtotal <- denomtotal + Denomi
  }
  # Cáculo da variância dentro dos grupos
  Var_dentro <- Num_total / (denomtotal - Num_nives)
  return(Var_dentro)
}
Variancia_Dentro_Grupo(dados[2],dados[1])
Variancia_Dentro_Grupo(dados[3],dados[1])
```

Encontra-se a variância para o caso de duas variáveis com o seguinte código:

```
Calculo_Var_dentro_Grupos
<- function(variavel1,variavel2,GrupoVariavel)
{
  GrupoVariavel2 <- as.factor(GrupoVariavel[[1]])
  levels <- levels(GrupoVariavel2)
  Num_niveis <- length(levels)
  # Encontrando a variancia 1 e 2 de cada grupo
  Covw <- 0
  for (i in 1:Num_niveis)
  {
    leveli <- levels[i]
    levelidata1 <-variavel1[GrupoVariavel==leveli,]
    levelidata2 <-variavel2[GrupoVariavel==leveli,]
    media1 <- mean(levelidata1)
    media2 <- mean(levelidata2)
    levelilength <- length(levelidata1)
    # get the covariance for this group:
    term1 <- 0
    for (j in 1:levelilength)
    {
```

```
      term1 <- term1 + ((levelidata1[j] - media1)*
      (levelidata2[j] - media2))
    }
    Cov_groupi <- term1 # covariance for this group
    Covw <- Covw + Cov_groupi
  }
  totallength <- nrow(variavel1)
  Covw <- Covw / (totallength - Num_niveis)
  return(Covw)
}
Calculo_Var_dentro_Grupos(dados[2],dados[3],dados[1])
```

Código para encontrar a variância entre grupos (B) será:

```
Calculo_Covariancia_dentro_Grupos
<- function(variavel1,variavel2,GrupoVariavel)
{
  GrupoVariavel2 <- as.factor(GrupoVariavel[[1]])
  levels <- levels(GrupoVariavel2)
  Num_niveis <- length(levels)
  # Calculo das médias
  mediavariavel1 <- mean(variavel1)
  mediavariavel2 <- mean(variavel2)
  # Calculando a matriz de covariância
  Covb <- 0
  for (i in 1:Num_niveis)
  {
    leveli <- levels[i]
    levelidata1 <- variable1[GrupoVariavel==leveli,]
    levelidata2 <- variable2[GrupoVariavel==leveli,]
    media1 <- mean(levelidata1)
    media2 <- mean(levelidata2)
```

```
    levelilength <- length(levelidata1)
    term1 <- (media1 - mediavariavel1)
    *(media2 - mediavariavel2)*(levelilength)
    Covb <- Covb + term1
  }
  Covb <- Covb / (Num_niveis - 1)
  Covb <- Covb[[1]]
  return(Covb)
}
attach(dados)
dados
Calculo_Covariancia_dentro_Grupos(X1,X4,G)
```

O valor da estatística Λ de Wilks será:

$$\Lambda = \frac{|W|}{|B+W|} = \frac{\begin{vmatrix} 57.6571 & -3.7857 \\ -3.7857 & 85.4285 \end{vmatrix}}{\begin{vmatrix} 120.4981 & 52.5823 \\ 52.58235 & 144.9023 \end{vmatrix}}$$

$$= \frac{57.6571 \cdot (85.4285) - 3.7857^2}{120.4981 \cdot (144.9023) - 52.5823^2}$$

$$= 0.3341$$

Por fim, o valor de $W^{-1}B$ no R será:

```
W<-matrix(c(57.6571,-3.7857,-3.7857,85.4285),
          ncol=2, byrow=TRUE); W
B<-matrix(c(62.8410,56.3680,56.3680,59.4738),
          ncol=2, byrow=TRUE); B
```

```
A<- solve(W) %*% B; A
Au<-eigen(A)
Au$values
Au$vectors
```

Da equação 2.3 demonstra-se a seguinte relação de contraste de valores atípicos.

$$\frac{B+W}{|W_{(i)}|} = 1+\frac{1}{n}D^2(x_i,\bar{x}_{(i)})$$

Em que $D^2((x_i,\bar{x})_{(i)})$ é a distância de Mahalanobis: $D^2((x_i,(\bar{x}))_{(i)}) = (x_i-\bar{x}_{(i)})'S^{-1}_{(i)}(x_i-\bar{x}_{(i)})$.

Demonstração 2.7.1. *A relação entre T e $W_{(i)}$ pode ser obtido da seguinte forma:*

$$T=\sum_{j=1}^{n}(x_i-\bar{x}_i+\bar{x}_i-\bar{x})'$$

. Que resulta em:

$$T=\sum_{j=1}^{n}(x_i-\bar{x}_i+\bar{x}_i-\bar{x})'+H+H'$$

. Em que:

$$H=\sum_{j=1}^{n}(x_j-\bar{x}_i+\bar{x}_i-\bar{x})'$$

. O primeiro termo de T se pode escrever da seguinte forma

$$\sum_{j=1}^{n}(x_i-\bar{x}_i+\bar{x}_i-\bar{x})' = W_{(i)}+(x_i-\bar{x}_i+\bar{x}_i-\bar{x})'$$

Sabendo-se que:

$$\bar{x}_{(i)}-\bar{x} = \bar{x}_i-\frac{(n-1)\bar{x}_i+x_i}{n} = \frac{1}{n}(\bar{x}_{(i)}-x_i)$$

Substituindo todos os termos $(\bar{x}_{(i)}-x_i)$ *por* $(\bar{x}_{(i)}-x_i)/n$*, obtém-se que:*

$$T = W_i+\frac{n-1}{n}(\bar{x}_{(i)}-x_i)(\bar{x}_{(i)}-x_i)'$$

Assim,

$$T = |W_{(i)}|\left|1+\frac{n-1}{n}W_{(i)}^{-1}(x_i-\bar{x}_{(i)})(x_i-\bar{x}_{(i)})'\right|$$

. E sabendo que:

$$\frac{S_o}{S} = 1+(\bar{x}-\mu_o)'S^{-1}(\bar{x}-\mu_o) = 1+\frac{T^2}{n-1}$$

Lema 2.7.1. *Se A é uma matriz não singular e z um vetor, então* $|I+Azz'| = 1+z'Az$*. Prova: Chamando* λ *o autovalor não nulo e* ν *o autovetor associado a este autovalor, como* $Azz'\nu = \lambda\nu$*, multiplicando por* z' *obtém-se que* $\lambda = z'Az$*. Então, a matriz* $I+Azz'$ *terá um autovalor igual a* $1+\lambda$ *e o resto será a unidade.(Cf. Penã,*

2002).

Logo, tem-se:

$$\frac{T}{|W_{(i)}|} = 1 + \frac{1}{n}(x_i - \bar{x}_{(i)})' S^{-1} (x_i - \bar{x}_{(i)})$$

Em que $S_{(i)}^{-1} = (n-1)W_{(i)}^{-1}$. *Assim,*

$$\frac{B+W}{|W_{(i)}|} = 1 + \frac{1}{n} D^2(x_i, \bar{x}_{(i)})$$

Portanto, para o teste calcula-se a distância de Mahalanobis, que se distribuirá, se H_o *é certo, para amostras grandes como uma* χ_p^2. *Na prática, para detectar valores atípicos calcula-se o máximo das distâncias* $D^2(x_i - \bar{x}_{(i)})$ *e este valor compara-se com os percentis 0.95 ou 0.99 da tabela de percentil do máximo de uma* χ_p^2.

No R, calcula-se estes percentis:

```
P<-pchisq(c(0.95,0.99),5); P
qchisq(P, 5)
```

Ordena-se todos os valores suspeitos por $D^2(x_i - \bar{x}_{(i)})$ *e se contrasta o mais próximo por ser incorporado à amostra. Se rejeita esta incorporação o procedimento termina e todos os dados suspeitos são declarados atípicos.*

2.7.1 Multicolinearidade

Dependência direta entre pares: correlações parciais

A dependência entre duas variáveis controlando o efeito de outras é medida pelo coeficiente de correlação parcial. Define-se o coeficiente de correlação parcial de duas (X_1, X_2), dada as variáveis $(X_3, \cdots, X_p)$, e se denota por $r_{12.3\cdots p}$ como coeficiente de correlação entre X_1 e X_2 quando se eliminam destas duas variáveis os efeitos das variáveis $(X_3, \cdots, X_p)$. Denota-se σ_{12} os elementos da inversa da matriz de covariância, S^{-1}, o coeficente de corelação parcial entre as variáveis $X_a, X_b \in X$. obtém-se por:

$$r_{ab.12,\ldots,p} = -\frac{\sigma_{ab}}{\sqrt{\sigma_{aa}\sigma_{bb}}} \tag{2.6}$$

Em que $X = [X_1, \ldots, X_p]$, $i = (1, \ldots, p)$ são as variáveis métricas por colunas.

Os coeficientes de correlação parcial podem ser calculado também a partir dos coeficientes de correlação múltipla mediante a fórmula:

$$1 - r_{123\cdots p} = \frac{1 - R^2_{1\dot{2},\cdots,p}}{1 - R^2_{1\dot{3},\cdots,p}}$$

Em que $r^2_{123\cdots p}$ é o quadrado do coeficiente de correlação parcial entre as variáveis (X_1, X_2) quando se controlam as variáveis

$(X_3, \cdots, X_p)$.

O teste de multicolinearidade de Farrar e Glauber (1967) ($F-G$) verifica se existe variável altamente correlacionada. Através da matriz de correlação de Pearson entre pares de variáveis. O teste $F-G$ é, de fato, um conjunto de testes de multicolinealidade. O teste do qui-quadrado para a detecção da existência multicolinearidade é uma função com várias variáveis métricas. A hipótese a ser testada é:

$$\begin{cases} H_0 : \textit{As variáveis X são ortogonais} \\ H_a : \textit{Caso contrário} \end{cases}$$

Usando as variáveis métricas padronizadas, um determinante padronizado é formado. Esse determinante padronizado, também chamado de determinante de correlação, é escrito considerando que os elementos da diagonal principal são iguais à unidade e os elementos fora da diagonal são os coeficientes de correlação simples entre as variáveis. O determinante padronizado considerando quatro variáveis é escrito como

$$\begin{vmatrix} 1 & r_{x_1x_2} & r_{x_1x_3} & r_{x_1x_4} & r_{x_2x_3} & r_{x_2x_4} & r_{x_3x_4} \\ r_{x_1x_2} & 1 & \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 1 \end{vmatrix}$$

A matriz de correlação parcial, CP, contém os coeficientes de correlação parcial entre cada pares de variáveis eliminando os

efeitos das restantes. Para as quatro variáveis temos:

$$CP = \begin{vmatrix} 1 & r_{x_{12.34}} & r_{x_{13}x_{24}} & r_{x_{14}x_{23}} \\ r_{x_{21.34}} & 1 & r_{x_{23}x_{14}} & r_{x_{24}x_{13}} \\ r_{x_{31.24}} & r_{x_{32}x_{14}} & 1 & r_{x_{34}x_{12}} \\ r_{x_{41.23}} & r_{x_{42}x_{13}} & r_{x_{43}x_{12}} & 1 \end{vmatrix}$$

Em que, por exemplo $r_{x_{21.34}}$ é a correlação entre as variáveis 2 e 1 quando se elimina o efeito das variáveis 3 e 4; $r_{x_{42.31}}$ é a correlação entre as variáveis 4 e 2 quando se elimina o efeito das variáveis 3 e 1 e assim por diante. Como se observa a notação 1.23 *(por exemplo) corresponde a $(X_1.X_2X_3) \in X$.*

Agora, no caso de multicolinearidade perfeita, os coeficientes de correlação simples são iguais à unidade e, portanto, o determinante acima se torna zero. Isso é, de acordo com a Equação 2.6 a matriz CP será:

$$CP \quad = \quad (-1)^{diag} D(S^{-1})^{-1/2} S^{-1} D(S^{-1})^{-1/2}$$

Em que $D(S^{-1}$ é a matriz ortogonal obtida selecionando os elementos diagonal da matriz S^{-1} e o $(-1)^{diag}$ indica a mudança do sinal de todos os elementos da matriz menos dos elementos diagonais que serão a unidade.

Considere o caso em que todos os elementos da matriz é 1. O

valor do determinante será:

$$\begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{vmatrix} = 0$$

Se todas as covariâncias forem 1 significa que existe uma correlação perfeita entre as variáveis, logo o valor do determinante será zero.

Por outro lado, no caso da ortogonalidade das variáveis métricas, os coeficientes de correlação simples são todos iguais a zero e, portanto, o determinante padronizado será a unidade (produto dos elementos da diagonal principal). Isso é:

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = 1$$

Na prática, no entanto, como a multicolinearidade perfeita ou ortogonalidade é muito rara, o determinante varia entre zero e unidade, mostrando que existe algum grau de multicolinearidade no modelo.

Assim, o problema da multicolinearidade pode ser considerado como a partida da ortogonalidade. Quanto maior a distância da ortogonalidade (ou mais próximo do valor do determinante padronizado para zero), mais forte é o grau de multicolinearidade e vice-

versa.

A partir desta noção Farrar e Glauber (1967) desenvolveram o teste Qui-quadrado para detectar a força da multicolinearidade em todo o conjunto de variáveis explicativas. A estatística do teste Qui-quadrado é dada por:

$$\chi^2 = -\left[(n-1)\frac{1}{6}(2p+5)\right] log_e(\lambda)$$

Em que λ é o valor do determinante, p número de variáveis e n número de indivíduo.

Sob a hipótese nula verdadeira, a estatística de teste seguirá uma distribuição Qui-quadrado com $(df = \frac{1}{2}p(p-1))$.

*Se o valor observado da estatística do teste Qui-quadrado for maior do que o valor crítico do Qui-quadrado no nível de significância desejado, rejeita-se a hipótese de ortogonalidade e aceita-se a presença de multicolinearidade. Caso contrário, se o valor observado da estatística do teste Qui-quadrado for inferior ao valor crítico do Qui-quadrado no nível de significância desejado, aceita-se que não há problema de multicolinearidade. Se o valor P é, no máximo, igual a α, dizemos que os dados são **estatisticamente significativos ao nível** α. A palavra "significante" no sentido estatístico significa que é pouco provável ocorrer multicolinearidade apenas por acaso.*

1. *Farar - Glauber é um teste F para multicolinearidade. Inicial-*

mente, calculam-se os coeficientes de correlações múltiplos entre as variáveis métricas. A estatística de teste será:

$$F = \frac{\left(R^2_{1.23\cdots x_p}\right)/(p-1)}{\left(1-R^2_{1.23\cdots x_p}\right)/(n-p)}$$

Em que n = tamanho da amostra e p = número de variáveis.

Se o valor observado de F for maior que o valor teórico com graus de liberdade no nível de significância α, aceita-se que a variável X_i multicolinear. Por outro lado, se o valor observado de F for inferior ao valor teórico de F, aceita-se que a variável X_i não será multicolinear.

1. *O teste de Farrar - Glauber detecta as variáveis que causam multicolinearidade. Os coeficientes de correlação parcial entre as variáveis métricas são calculados e sua significatividade é testada de acordo com o teste. Para a modelagem de três variáveis, os coeficientes de correlação parcial são dados pelas fórmulas:*

$$r^2_{12.3} = \frac{(r_{12}-r_{13}r_{23})^2}{(1-r^2_{23})(1-r^2_{13})}$$

$$r^2_{13.2} = \frac{(r_{13}-r_{12}r_{23})^2}{(1-r^2_{23})(1-r^2_{12})}$$

$$r^2_{23.1} = \frac{(r_{23}-r_{12}r_{13})^2}{(1-r^2_{13})(1-r^2_{12})}$$

Após calcular todos os coeficientes de correlação parciais, sua

significância estatística é testada individualmente para cada uma delas usando a seguinte estatística de teste.

$$L = \frac{\left(r_{x_a x_b.12\cdots x_p}\right)\sqrt{n-p}}{\sqrt{1-r^2_{x_a x_a.12\cdots x_p}}}$$

A estatística de teste L tem uma distribuição t com $(n-p)$ *graus de liberdade. Assim, se o valor calculado de estatística t for maior que o valor teórico de t com* $(n-p)$ *graus de liberdade no nível desejado de significância, aceitamos que as variáveis* X_a *e* X_b *são responsáveis pela multicolinearidade no modelo, caso contrário, as variáveis não são a causa da multicolinearidade, pois seu coeficiente de correlação parcial não é estatisticamente significativo.*

No R *é necessário carregar os pacotes:* `readxl` *(HADLEY; BRYAN, 2016),* `mctest` *(IMDADULLAH; ASLAM, 2016) e* `GGally` *(BARRET* et al*, 2016).*

```
dados<-read.csv2("dados.csv",header=TRUE,
       encoding="latin")
dim(dados)
dados<-dados[,1:10]
colnames(dados)
attach(dados)
dados1<-as.matrix(dados[,2:10])  #Transf em matriz
dim(dados1)   #  27 x 10 (27 estados e 10 variáveis)
rownames(dados1)<-dados[,1]
colnames(dados1)
Deu<- dist(dados1, method="euclidea")
```

```
#cada linha representa os estados
# dendrograma tridimensional usando a
# distância euclidiana
colnames(dados1)
#install.packages("readxl")
library(readxl)
#wagesmicrodata <- read_excel(file.choose(),
sheet = "Data", skip = 0)
View(wagesmicrodata)
attach(wagesmicrodata)
install.packages("mctest")
library(mctest)
attach(dados)
omcdiag(as.matrix(dados1),dados1[,1])
library(GGally)
ggpairs(data.frame(dados1))
```

A Figura 2.10 mostra a dispersão entre cada par de variáveis e as correlações.

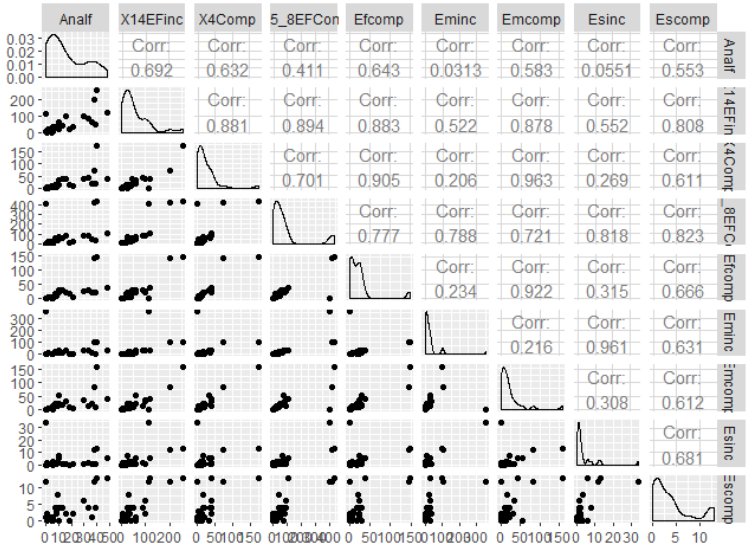


Figura 2.10: Gráfico de dispersão entre as variáveis e correlações.

Na saída do R *observa-se que foram detectados multicolinearidades principalmente no teste de Farrar e no determinante. Cabe ao leitor pesquisar exemplos sobre outros testes. Carregue o pacote* `corpcor` *(SCHÄFER* et al*, 2015).*

```
omcdiag(as.matrix(dados1),dados1[,1])
# Call:
omcdiag(x = as.matrix(dados1), y = dados1[, 1])
# Overall Multicollinearity Diagnostics
#                        MC Results detection
# Determinant |X'X|:          0.0000         1
# Farrar Chi-Square:        429.9720         1
# Red Indicator:              0.6673         1
# Sum of Lambda Inverse:    459.0107         1
```

```
# Theil's Method:             0.4178         0
# Condition Number:          58.0694         1
# 1 --> COLLINEARITY is detected
# 0 --> COLLINEARITY in not detected by the test
# ===================================
# Eigvenvalues with INTERCEPT
#   Intercept Analf X14EFinc X4Comp X5_8EFCom Efcomp
#    7.4105 1.4209   0.6744 0.2360    0.1089 0.0793
#    1.0000 2.2837   3.3148 5.6035    8.2487 9.6666
#                    Eminc  Emcomp   Esinc  Escomp
# Eigenvalues:        0.0387  0.0205  0.0086  0.0022
# Condition Indeces: 13.8377 19.0090 29.3851 58.0694
# Correlaçoes
install.packages("corpcor")
library(corpcor)
DD<-cor2pcor(cov(dados1))
```

Definição 2.7.1. *O coeficiente de dependência efetiva será*

$$D(CP) \quad = \quad 1 - |CP|^{1/p}$$

Em que CP é a matriz de correlação parcial e p é o número de variáveis.

Este coeficiente é uma boa medida global de dependência nos dados (PEÑA; RODRÍGUEZ, 2003). Por exemplo, para $p = 2$ como $|R_2| = 1 - R_{12}^2$ esta medida coincide com o quadrado de correlação de Pearson entre duas variáveis. Para $p > 2$ pode-se escrever

$$|R_p| \quad = \quad (1-R^2_{p\cdot 1\ldots p-1})(1-R^2_{p-1\cdot 1\ldots p-2})\ldots(1-R^2_{2\cdot 1})$$

e $D(R_p) = 1-|R_p|^{1/p}$ *da seguinte forma:*

$$1-D(R_p) \quad = \quad [(1-R^2_{p\cdot 1\ldots p-1})(1-R^2_{p-1\cdot 1\ldots p-2})\ldots(1-R^2_{2\cdot 1})]^{1/(p-1)}$$

Observa-se que a dependência efetiva é o coeficiente de correlação necessário para que a variabilidade não explicada no problema seja igual à média geométrica de todas as possíveis variabilidades não explicada. Usando as funções do pacote ppcor *(KIM, 2015) pode-se encontrar a correlação parcial bem como o teste de hipóteses para a matriz de correlação para verificar se as p variáveis são independentes ou não.*

```
# correlação parcial
install.packages("ppcor")
library(ppcor)
pcor(dados1)
library(corpcor)
CP<- cor2pcor(cov(dados1))
detCP<-det(CP); detCP
p<- 9 # variávies
Coef_Dep_Efe<- 1-(det(CP))^{1/(p-1)}; Coef_Dep_Efe
# [1] -0.1234
```

Para uma medida de dependência global, o determinante de CP é 2.8511 e o coeficiente de dependência efetiva é -0.12. Pode-se concluir que a dependência linear negativa explica 12,34 % da variabilidade desse conjunto de dados.

2.8 Número de agrupamento

É difícil fornecer uma solução clara sobre como escolher o "Melhor" número de clusters, k, seja qual for o método de agrupamento. Pode-se optar por colocar o limiar onde os clusters são também distantes (isto é, há um grande salto entre os dois níveis adjacentes). Se n indicar o tamanho da amostra, uma simples regra de polegar (Cf. MARDIA; KENT; BIBBY, 1979) é $k = \sqrt{n/2}$. *No "método do cotovelo"ajudar a encontrar o número de agrupamento diante de k agrupamentos.*

Nesse conjunto de dados, observa-se a composição de diferentes grupos. Dado um conjunto de observações $(x_1, x_2, \cdots, x_n)$, *em que cada observação é um vetor real de dimensão p, o agrupamento k-means visa particionar as observações n em* $(k \leq n) S = (S_1, S_2, \cdots, S_n)$ *para minimizar a soma de quadrados dentro do agrupamento (WCSS* - within-cluster sum of squares*).*

$$WCSS(k) \quad = \quad \sum_{j=1}^{k} \sum_{x_j \in A_i} \|x_j - \bar{x}_i\|^2$$

Em outras palavras, o objetivo é encontrar:

$$arg\min_{A} \quad = \quad \sum_{j=1}^{k} \sum_{x_j \in A_i} \|x_j - \bar{x}_i\|^2$$

Em que $\bar{x}_i$ *é a média amostral no agrupamento* A_i. *O eixo da coordenada* $WCSS(k)$ *contra o número k de agrupamento. O problema de otimização é resolvido com a função* `kmeans` *em* R.

Na Figura 2.11 escolhe-se o número de três agrupamento observando ponto de inflexão (cotovelo). Carregue o pacote `GMD` *(ZHAO* et al*, 2011).*

```
install.packages("GMD")
library("GMD")
D1<-dist(dados1)
hc <-hclust(dist(dados1, method="euclidea"),
method="single")
plot(hc)
```

```
css1 <- css.hclust(D1,hc)
plot(css1)
summary(css1)
plot(css1$k,css1$tss-css$totbss,pch=16,
    ylab="Soma de quadrado dentro do grupo (WCSSk)",
    xlab="Número de agrupamentos",cex.lab=1)
par (mar = c(5, 10, 3, 1) + 0.1)
```

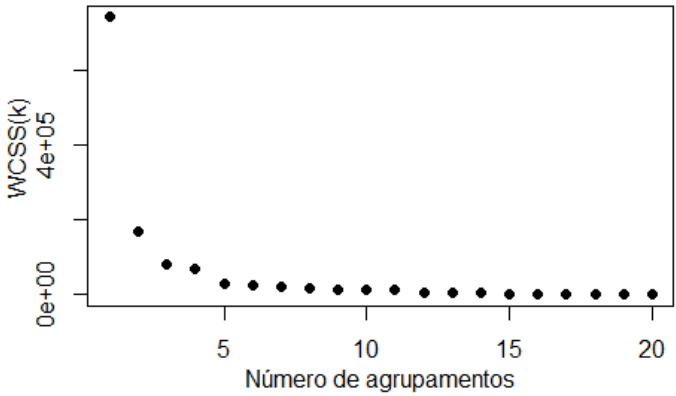


Figura 2.11: Gráfico da função $WCSS(k)$ *versus* número de agrupamento

2.8.1 Matriz cofenética

Para agrupamento hierárquico pode-se utilizar uma medida bastante comum, que é a correlação cofenética. O coeficiente cofenético mede o grau de preservação das distâncias emparelhadas pelo dendrograma resultante do agrupamento em relação às distâncias originais (SNEATH & SOKAL, 1973).

Um método é melhor que outro quando o dendrograma fornece uma imagem menos distorcido da realidade. Pode-se avaliar o grau de deformação provocado pela construção do dendrograma através do "coeficiente de correlação cofenético", que serve para medir o grau de ajuste entre a matriz de dissimilaridade (euclidiana) e a matriz resultante da simplificação proporcionada pelo método de

agrupamento (matriz cofenética MC).

Esse coeficiente de correlação cofenético é o coeficiente r de Pearson, sendo calculado entre índices de similaridade ou dissimilaridade da matriz original e os índices reconstituídos com base no dendrograma. Logo, quanto maior for o r, menor será a distorção. Há sempre um certo grau de distorção, pois o r nunca será igual a 1.

$$CC = \frac{\sum_{i=1}^{n-1}\sum_{j=1+1}^{n}(c_{ij}-\bar{c})(d_{ij}-\bar{d})}{\sqrt{\sum_{i=1}^{n-1}((c_{ij}-\bar{c}))}\sqrt{\sum_{i=1}^{n-1}(d_{ij}-\bar{d})}}$$

Em que,

- c_{ij} *valor de dissimilaridade entre os indivíduos i e j, obtidos a partir da matriz cofenética.*
- d_{ij} *valor de dissimilaridade entre os indivíduos i e j, obtidos a partir de dissimilaridade.*
- $\bar{c} = \frac{2}{n(n-1)}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} c_{ij}$
- $\bar{d} = \frac{2}{n(n-1)}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij}$

Para que seja possível calcular os valores da matriz cofenética, faz-se necessário estabelecer a medida de distância que será utilizada na análise. No exemplo abaixo, utilizar-se-á o método simples, sendo este uma medida da distância euclidiana e os métodos de agrupamentos: simples, completo, mediana e Ward. No R *considere os pacotes* `ggplot2` *(WICKHAM, 2009) e* `ggdendro` *(VRIES; RIPLEY, 2016).*

```
library(ggplot2)
library(ggdendro)
dados<-read.csv2("Abandono_EscolaridadeTB20131.csv",
header=TRUE, encoding="latin")
```

```
dados<-dados[,1:10]
colnames(dados)
attach(dados)
dados1<-as.matrix(dados[,2:10])
# Inserindo os nomes dos estados
# na primeira coluna
rownames(dados1)<-dados[,1]
Deuc<- dist(dados1, method="euclidea")
hsingle <- hclust(Deuc, "single")
ggdendrogram(hsingle, rotate = FALSE, size = 1)
dsingle <- cophenetic(hsingle)
cor(Deuc, dsingle)
hcompleta <- hclust(Deuc, "complete")
ggdendrogram(hcompleta, rotate = FALSE, size = 2)
dcomplete <- cophenetic(hcompleta)
cor(Deuc, dcomplete)
hmedian<- hclust(Deuc, "median")
ggdendrogram(hmedian, rotate = FALSE, size = 2)
dmedian <- cophenetic(hmedian)
cor(Deuc, dmedian)
hWard2 <- hclust(Deuc, "ward.D")
ggdendrogram(hWard2, rotate = FALSE, size = 2)
dward2 <- cophenetic(hWard2)
cor(Deuc, dward2)
```

A Tabela 2.2 mostra os valores dos coeficiente cofenéticos considerando os métodos propostos.

Tabela 2.2: Valores dos coeficientes cofenéticos

Métodos	Coeficientes
Simples	0.9793
Completa	0.9786
Mediana	0.9652
Ward	0.9567

Nesse exemplo hipotético observa-se que o método de ligação completa teve um maior coeficiente cofenético, assim afirma-se basicamente que o agrupamento teve boa qualidade.

2.8.2 Validação dos agrupamentos

Para determinar se os grupos são significativos ou não, o resultado do agrupamento é validado para aferir a qualidade da solução encontrada. Estes índice avaliam a qualidade dos grupos formados, com base na ideia de que, se eles refletirem a estrutura dos dados, então os índices de validação devem indicar um bom resultado. Em validação, é geralmente desenvolvido no contexto de um dos três tipos diferentes de testes de validação: externos, internos e relativos

- *Os testes externos comparam um agrupamento ou parte dele com informação que não é usada para construir o agrupamento. Neste caso, o resultado do algoritmo é avaliado*
  comparando-se com uma estrutura pré-definida que é imposta ao conjunto de dados, refletindo a estrutura real em grupos que se sabe ou que se pensa afetar os elementos do conjunto.
- *Os testes internos comparam um agrupamento ou parte de um grupo com o conjunto de dados original, usando somente informação obtida a partir do processo do grupo, medindo-se essencialmente o desvio entre a estrutura gerada pelo algoritmo aplicado e os dados.*
- *Os testes relativos comparam várias estruturas de agrupamento do mesmo conjunto de dados, resultantes da aplicação do algoritmo com diferentes valores de parâmetros de entrada.*

Os seguintes índices podem ser utilizados tanto para medir a qualidade dos agrupamentos gerados quanto para compará-los.

Suponha que N seja o número de objetos a classificar formando K grupos, em relação a n variáveis $X_1, X_2, \cdots, X_n$. Sejam B, W e T as matrizes de dispersão dentro do grupo, entre grupos e total, respectivamente. Como $T = B + W$ não depende da forma que tem sido agrupado os sujeitos, um critério razoável de classificação consiste em construir K grupos de forma que B seja máximo ou W seja mínimo, seguindo algum critério apropriado. Alguns destes critérios são:

(a) *Minimizar $tr(W)$, ou seja o traço de W.*

(b) *Minimizar o determinante de W.*

(c) *Minimizar $\frac{|W|}{|T|}$.*

(d) *Maximizar o $tr(W^{-1}B)$.*

(e) *Minimizar* $\sum_{i=1}^{A} \sum_{k=1}^{N_i} (X_{ik} - \bar{X}_i)' S_i^{-1} (X_{ik} - \bar{X}_i)$.

Os critérios (a) e (b) se justificam porque tratam de minimizar a magnitude da matriz W. O critério (c) é chamado critério de Wilks é equivalente a (b) porque o $det(W)$ é constante. O caso, (d) é chamado critério de Hotelling e o critério (e) representa a soma das distâncias de Mahalanobis de cada sujeito ao centro de gravidade do grupo ao que é indicado.

O número de formas de agrupar os N sujeitos (objetos) nos K grupos é de ordem de $K^N k!$. Por exemplo, para três grupos $K = 3$ e 10 elementos $N = 10$, tem-se:

```
#$K^{N} k!$
K<-3
N<-10 # (elementos)
F<-K^{N}* prod(1:K); F
```

O valor será de 354294 formas de agrupar em três grupos os 10 elementos. Uma vez elegido o critério de otimização, é necessário seguir algum algoritmo adequado de classificação para evitar o número elevado de agrupamento.

2.9 Consistência do agrupamento

É feita após ter obtido o dendrograma. Com a formação do dendrograma pode ocorrer considerável simplificação das informações originais e podem ser geradas algumas distorções sobre o padrão de dissimilaridade entre os indivíduos estudados. Assim, é necessário verificar a adequação do resultado do dendrograma. Em (1962) Sokal e Rohlf propuseram o uso do coeficiente de correlação cofenético. Este coeficiente mede a correlação entre as distâncias iniciais, tomadas a partir dos dados originais e as distâncias com as quais os indivíduos se uniram durante o desenvolvimento do método considerado. Quanto menor o valor, do coeficiente de correlação cofenética, menor será a distorção provocada ao agrupar os indivíduos. Os resultados do coeficiente cofenético são mostrados a seguir:

Tabela 2.3: Distâncias recuperadas do dendrograma e as distâncias originais

Indivíduos	Indivíduos	d_{ij}	Elemento da matriz cofenética
1	2	5,477	5,477
1	3	8,185	5,477
2	3	5,196	5,196

Assim, a correlação de Pearson entre os elementos da matriz de $D_{3\times3}$ e da matriz cofenética, $Cof_{3\times3}$ teve um valor 0.56. Também este valor não é significativo a 5% de probabilidade. Recomenda-se avaliar a consistência de outro padrão de agrupamento. Considerando apenas, 3 indivíduos como forma didática, pode-se afirmar que o agrupamento não teve uma boa qualidade.

2.10 Dendrograma bi e tridimensional

A análise de agrupamento é um conjunto de técnicas multivariadas que, ainda têm diversas aplicações na área de saúde pública, e estão orientadas fundamentalmente para classificação de elementos de uma população ou amostra (estados) em distintos grupos nos que os integrantes do mesmo grupo ou agrupamento po-

dem ser considerados parecidos entre si e distintos dos demais. Suponha que se dispõem dos grupos de estados r e s brasileiros, em que um deles contém n_a e n_b estados. Considere também as p variáveis representadas por diversos níveis de escolaridades nos estados, como se observa na Tabela 2.4.

2.10.1 Clasificação Hierárquica dendrograma)

É o método mais utilizado. Obtém-se uma sequência de partições (classificações) organizadas hierarquicamente, deste uma única classe que contém todos os elementos, até n classes, cada uma delas com o um só elemento. A sequência pode ser representada graficamente mediante um dendrograma, que permite ver o processo completo de classificação desde seu início com cada elemento da amostra (GONZÁLEZ; LISTE, 2012). No dendrograma se observa todas as partições ou classificações obtidas e sua relação hierárquica.

Na função `hclust()` *o primeiro argumento serve para definir os dados e o tipo de distância ("Euclidiana", "Manhattan", ...); com o "method"define-se o método de aglomeração que pode ser alguma das seguintes opções:* `Ward, single, complete, average, mcquitty, median` *e* `centroid`.

2.10.2 Função: draw.dendrogram3d()

Com as funções do pacote `NeatMap()` *(RAJARAM; OONO, 2014) é possível realizar dendrogramas tridimensionais e mapa de calor, respectivamente. Com a função* `heatmap` *se constroi mapa de calor, neste mapa também mostra distintas cores e tonalidades indicando a intensidade da relação. Aplicada a uma matriz de dados, com elementos (objetos) e variáveis (colunas), classificam-se ambos simultaneamente elaborando um dendrograma marginal para as linhas e outros para as colunas, ordenados conjuntamente. A cor branca indica máxima relação, passando pelo amarelo, laranja e vermelho intensidade mínima, as cores por defeito (*default*) podem ser modificadas livremente pelo usuário.*

Pode-se fazer também dendrogramas tridimensionais usando ainda o pacote `NeatMap`. *Obtém-se previamente as coordenadas tridimensionais com a função* `nMDS()`. *O argumento* `"emded.dim"` *define a dimensão do espaço euclidiano em que se obtém o gráfico. Com* `"n.iters"` *especifica-se o número de interações. O argumento* `"metric"` *especifica o tipo de distância, que pode ser "Pearson" ou "Euclidiana". Uma vez gerado o dendrograma tridimensional pode-se mudar seu*

tamanho de modo iterativo utilizando o mouse convenientemente para encontrar uma ótima posição que revele melhor os agrupamentos no espaço 3D.

O exemplo a seguir são casos de abandono do tratamento da tuberculose em 2013 nos estados brasileiros por níveis de escolaridade retirada do site do Sistema de Informação de Agravos de Notificação (SINAN). [1] *As variáveis educativas são: analfabeto (Anal f), escolaridade primária incompleta (EF_1a4), escolaridade primária completa (EF_4), ensino fundamental incompleto (EF_5a8), ensino fundamental completo (EFcomp), educação secundária incompleta (EM_inc), educação secundária completa (EM_comp), educação superior incompleta (ES_inc) e educação superior completa (ES_comp).*

[1] O uso dos dados coletados do SINAN foi aprovado pelo Comitê de Ética da Universidade Estadual da Paraíba (UEPB), sob o registro de Nº 45954315.5.0000.5187, de acordo com a resolução 466/2012 do Conselho Nacional de Saúde.

Tabela 2.4: Casos de abandono do tratamento da tuberculose por níveis de escolaridade, 2013

	UF	Analf	*EF_1a4*	*EF_4*	*EF_5a8*	EFcomp	*EM_inc*	*EM_comp*	*ES_inc*	*ES_comp*
1	RO	2	23	1	19	6	10	7	1	1
2	AC	1	0	0	4	1	0	1	0	0
3	AM	10	61	34	72	29	31	54	6	0
4	RR	0	5	0	3	2	2	0	0	1
5	PA	16	98	38	86	24	36	31	1	2
6	AP	5	3	2	5	1	1	3	1	0
7	TO	1	3	2	2	3	0	2	0	0
8	MA	36	61	20	51	24	17	22	1	1
9	PI	6	9	4	2	4	4	2	0	1
10	CE	33	88	41	88	29	28	13	2	4
11	RN	19	18	14	16	19	4	8	0	0
12	PB	39	48	19	40	24	6	9	1	4
13	PE	48	120	38	111	38	30	42	6	13
14	AL	21	33	8	26	7	7	6	1	2
15	SE	6	37	8	25	7	4	12	3	0
16	BA	30	98	38	88	25	29	34	1	6
17	MG	13	46	37	60	31	16	19	1	4
18	ES	8	30	4	48	10	18	14	1	3
19	RJ	40	255	175	439	149	101	157	13	13
20	SP	0	112	0	406	0	356	0	34	12
21	PR	7	35	15	57	19	20	21	2	4
22	SC	10	24	15	68	24	10	21	8	4
23	RS	38	201	74	424	145	102	83	12	12
24	MS	5	21	7	23	5	3	5	2	1
25	MT	9	30	15	39	22	20	15	3	8
26	GO	8	25	14	33	11	9	14	1	2
27	DF	1	3	5	2	2	1	1	2	2

```
library(NeatMap)
dados<-read.csv2("Abandono_EscolaridadeTB20131.csv",
```

```
       header=TRUE, encoding="latin"); dados
library(xtable)
xtable(dados)
dados<-dados[,1:10]
colnames(dados)
attach(dados)
dados1<-as.matrix(dados[,2:10])  # Transf em matriz
dim(dados1)   #  27 x 10 (27 estados e 10 variáveis)
rownames(dados1)<-dados[,1]
#cada linha representa os estados
#dendrograma tridimensional usando dist. euclidiana
hc.MDS<-nMDS(dados1,embed.dim=3,n.iters=100,
        metric="euclidean")
hc.CLUSTER<-hclust(dist(dados1, method="euclidea"),
            method="single")
plot(hc.CLUSTER)
draw.dendrogram3d(hc.CLUSTER,hc.MDS$x,
labels=rownames(dados1), label.size=0.5)
```

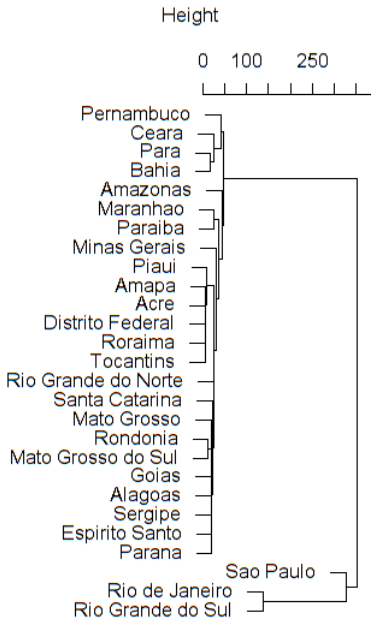


Figura 2.12: Dendrograma conforme critério do Vizinho mais próximo - distância euclidiana.

Seguindo o mesmo critério na Figura 2.13 observa-se o dendrograma 3D. Comparando com a Figura 2.12 claramente SP, RJ e Rio Grande do Sul tem uma certa distância em relação aos demais estados brasileiros.

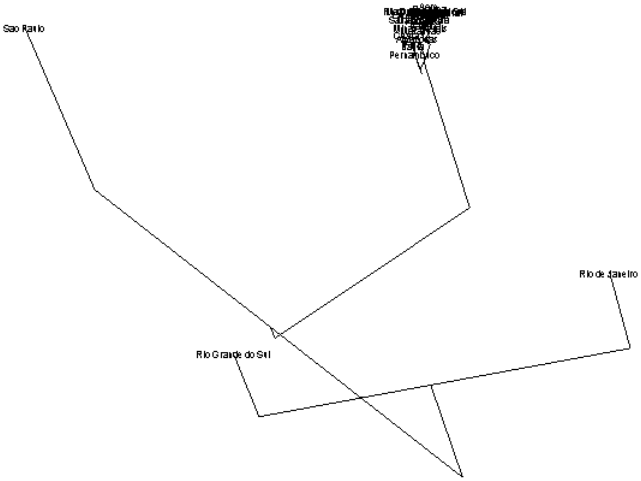


Figura 2.13: Dendrograma 3D conforme critério do Vizinho mais próximo

2.10.3 Cálculo do índice de Fowlkes-Mallows para semelhança de dois dendrogramas.

O índice de Fowlkes e Mallows (1983) é um método de avaliação externo usado para determinar a semelhança entre dois agrupamentos (clusters obtidos após um algoritmo de agrupamento). Essa medida de similaridade poderia ser entre dois agrupamentos hierárquicos ou um agrupamento e uma classificação de referência. Este índice varia de 0 a 1, quanto mais próximo de um, maior é a semelhança entre as partições criadas pelos pares de dendrogramas.

```
library(dendextend)
library(NMF)
x<-dados1
x<-as.matrix(x)
Rowv <- x %>% dist %>% hclust %>% as.dendrogram %>%
set("branches_k_color", k=6)%>%set("branches_lwd",4)
%>% ladderize
```

```
# rotate_DendSer(ser_weight = dist(x))
Colv<-x%>%t%>%dist%>%hclust%>%as.dendrogram%>%
set("branches_k_color", k=6)%>%set("branches_lwd",
    4)%>% ladderize
hc1 <- hclust(dist(x), "ward.D2")
hc2 <- hclust(dist(x), "single")
dend1 <- as.dendrogram(hc1)
dend2 <- as.dendrogram(hc2)
Bk(hc1,hc2,k = 6)
```

Na Figura 2.14, a linha pontilhada mostra os valores Bk. A linha tracejada (inferior) mostra os valores de Bk esperado sob a hipótese H_o (não existe relação entre os clusters nos dois dendrogramas considerados) e a hipótese alternativa (existe relação entre eles). A linha contínua (linha vermelha sólida) mostra os valores Bk críticos superiores para rejeitar H_o. O $k = 6$ significa o número de clusters considerado nos grupos.

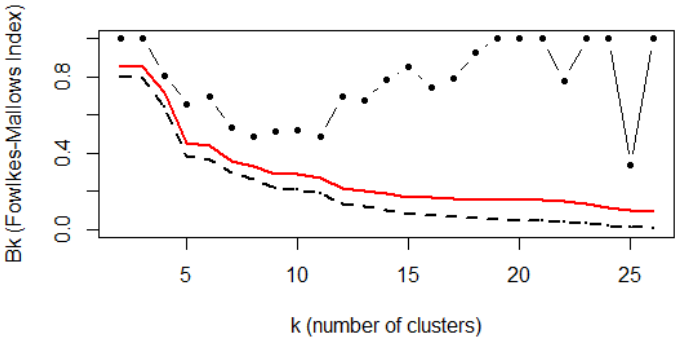


Figura 2.14: Comparação dos métodos vizinho mais próximo e método de Ward.

Vamos considerar diversos agrupamentos: ligação completa (vizinho mais distante), ligação simples (vizinho mais próximo), average *e método de Ward. Calcular o Bk entre cada par e a diferença média relativa entre pares de dendrogramas.*

```
Completa<-x%>%dist%>%hclust("com")%>%as.dendrogram
Simples<-x%>%dist%>%hclust("single")%>%as.dendrogram
Ave<-x%>%dist%>%hclust("ave")%>%as.dendrogram
Centroide<-x%>%dist%>%hclust("centroid")
          %>%as.dendrogram
dend1234<-dendlist("Completa"=dend1,"Single"
     =Simples,"Average"=Ave, "Centroid"=Centroide)
```

A Figura 2.15 ilustra quatro dendrogramas com os métodos ligação completa, simples, average e centroide.

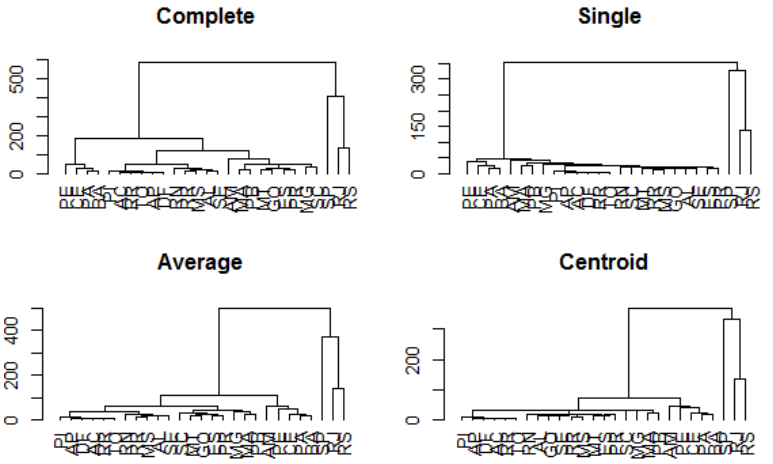


Figura 2.15: Dendrogramas. Usando a distância Euclidiana

A diferença das distâncias dos métodos de agrupamento através da função `all.equal()` *É visto na Tabela 2.5. abaixo.*

Tabela 2.5: Diferença das distâncias dos métodos de agrupamento.

Agrupamentos		Diferenças
Simples	Centroide	0,0736
Simples	Ave	0,3299
Simples	Completo	1,1118

Comparações de pares de dendrogramas

Numa interpretação básica, cada elemento do dendrograma se corresponderá com o outro par de forma que é possível visualizar a relação entre elementos (no caso estados brasileiros). Também ajudar a entender melhor a posição dos elementos nos grupos criados pelos dois dendrogramas paralelos. Usando as funções do pacote `dendextend` *(GALILI, 2015) é possível visualizar dois métodos de agrupamentos paralelamente em forma de dendrogramas, bem como a correspondência dos elementos através de linhas coloridas.*

```
dados<-read.csv2("Abandono_EscolaridadeTB20131.csv",
       header=TRUE, encoding="latin")
dados<-dados[,1:10]
colnames(dados)
attach(dados)
dados1<-as.matrix(dados[,2:10])  # Transf em matriz
dim(dados1)   #  27 x 10 (27 estados e 10 variáveis)
rownames(dados1)<-dados[,1]
dim(dados1)
colnames(dados1)
set.seed(23235)
Completa<-dados1%>%dist%>%hclust("complete")
         %>%as.dendrogram
Simples<-dados1 %>%dist %>%hclust("single")
        %>%as.dendrogram
Media<-dados1%>%dist%>%hclust("ave")%>%as.dendrogram
Centroide<-dados1%>%dist%>%hclust("centroid")
```

```
            %>%as.dendrogram
Juntos<-dendlist("Complete"=Completa,"Single"
     =Simples,"Average"=Media,"Centroid"=Centroide)
par(mfrow = c(2,2))
plot(Completa, main = "Complete")
plot(Simples, main = "Single")
plot(Media, main = "Average")
plot(Centroide, main = "Centroid")
Juntos23 %>% tanglegram(which = c(2,3))
Juntos24 %>% tanglegram(which = c(2,4))
Juntos12 %>% tanglegram(which = c(1,2))
```

- *A Figura 2.16 se observa que RS, SP e RJ são os mesmos nos dendrogramas gerados pelos métodos vizinho mais próximo e average, respectivamente. os Estados de BA, PA, CE e PE formam praticamente o mesmo grupos comparando os dois métodos de conglomerados. As linhas com cores diferentes fazem a correspondência dos mesmos elementos nos dois dendrogramas.*

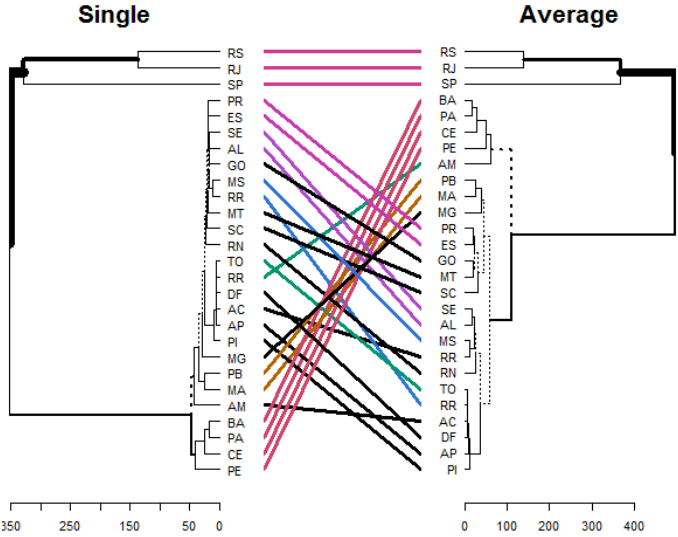


Figura 2.16: Comparações dos dendrogramas pelos métodos vizinho mais próximo e Average.

- *A Figura 2.17 se observa que RS, SP e RJ são os mesmos nos dendrogramas gerados pelos métodos vizinho mais próximo e centroide, respectivamente. Praticamente se tem a mesma distância máxima nos dois dendrogramas.*

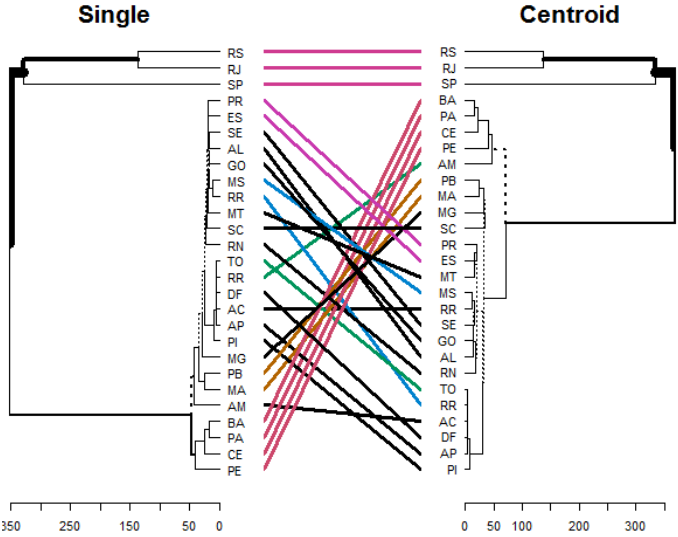


Figura 2.17: Comparações dos dendrogramas pelos métodos vizinho mais próximo e centroide

- *Na Figura 2.18 o mesmo ocorre com RS, SP e RJ. Torna-se mais difícil relacionar os Estados comparando estes dois dendrogramas. Nesse exemplo hipotético, na ligação completa o nível de hierarquia é superior a 500, já na ligação simples tem-se uma distância de 350.*

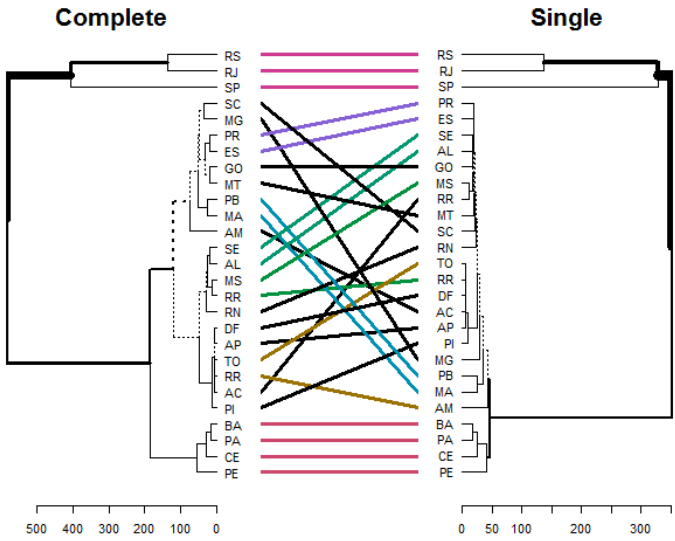


Figura 2.18: Comparações dos dendrogramas pelos métodos: vizinho mais próximo e completa.

2.11 Agrupamento por variável

A análise de agrupamento de variáveis é um procedimento exploratório que pode sugerir procedimentos de redução de dimensão, como as técnicas de redução como análise fatorial, correlação canônica por exemplo. A matriz de covariância e de correlação também serve para verificar relações lineares entre as variáveis. A distância entre duas variáveis, por exemplo, X_1 e X_2 representando cada variável como um ponto em R^n. Numa matriz as variáveis são as colunas e as linhas dos objetos, assim cada objeto é representado por estas duas variáveis. A distância entre estas duas variáveis é:

$$\begin{aligned} d_{12}^2 &= \sum_{j=1}^{n} (x_{j1} - x_{j2})^2 \\ &= \sum_{j=1}^{n} x_{j1}^2 \sum x_{j2}^2 - 2 \sum x_{j1} x_{j2} \end{aligned}$$

em que $X_i, i = 1,2$ *variáveis e* $j = 1,2$ *indivíduos.*

Para que a distância não dependa das unidades de medidas, as duas variáveis devem ser padronizadas com média zero e variância. Com esta padronização a distância 2.7 será $d_{12}^2 = 2n(1 - r_{12})$*. Em que* r_{12} *é a correlação entre as variáveis 1 e 2. Considere os valores 1,0 e* < 0 *desta correlação, visto que a mesma pode variar de* $-1 < r_{12} < 1$*.*

(a) *Se* $r_{12} = 1$*, a distância,* $d_{12}^2 = 2n(1-1) = 0$*. Indicando que as duas variáveis são idênticas.*

(b) *Se* $r_{12} = 0$*, a distância,* $d_{12}^2 = 2n(1)$*, logo,* $d_{12} = \sqrt{2n}$*. As duas variáveis estão não correlacionadas*

(c) *Se* $r_{12} < 0$*, por exemplo* $r_{12} = -0.5$*, a distância,* $d_{12} = \sqrt{3n}$*.*

Considerando $|r_{i2}|$ *as distâncias entre as variáveis não dependem do sinal da correlação.*

```
Deuc<- dist(t(dados1), method="euclidea")
hsingle <- hclust(Deuc, "single")
ggdendrogram(hsingle, rotate=FALSE, size=1)
dsingle <- cophenetic(hsingle)
cor(Deuc, dsingle)
```

A Figura 2.19 mostra o dendrograma do agrupamento das variáveis dos dados1.

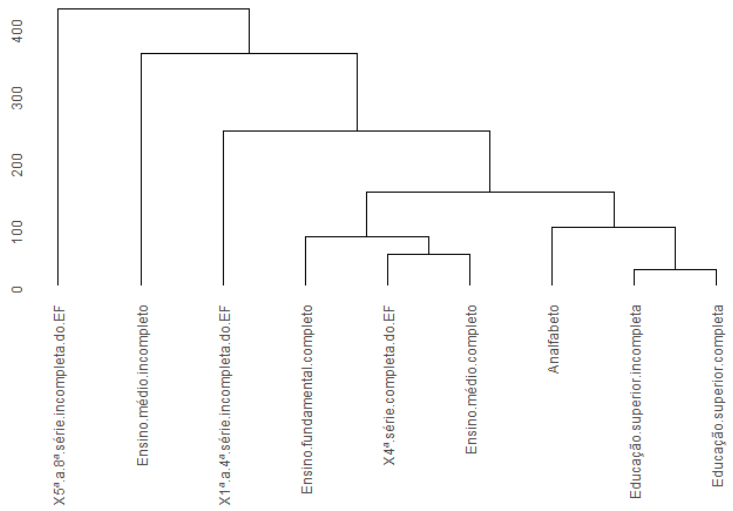


Figura 2.19: Agrupamento por variáveis. Método do vizinho mais próximo.

A Tabela 2.6 mostra os valores dos coeficiente cofenéticos considerando os métodos propostos para o conglomerado por variável.

Tabela 2.6: Valores dos coeficientes cofenéticos

Métodos	Coeficientes
Simples	0.9136
Completa	0.9478
Mediana	0.9458
Ward	0.8736

Exemplo 2.11.1. *Suponha que temos medido dez indivíduos nos grupos I e II, como se observa na Tabela 2.7. Suponha que temos medido 10 sujeitos nas duas variáveis de personalidade (nível de ansiedade (NA), nível de tristeza (T) e nas duas variáveis cognitivas (velocidade para resolver problema de matemática (P) e capacidade de realização de tarefas simultâneas (TS) e os dados foram o*

seguinte:

Tabela 2.7: Dados sobre as características de um único grupo X.

Indiv	NA	T	P	TS
1	11	6	52	19
2	6	7	52	17
3	7	11	51	14
4	6	14	49	21
5	19	17	47	20
6	15	18	51	27
7	20	20	47	26
8	10	17	54	26
9	12	12	47	26
10	14	8	50	24
Total	120	130	500	220
Media	12	13	50	22

O valor de $d_{1,2} = \sqrt{(11-6)^2 + (6-7)^2 + (52-52)^2 + (19-17)} = 5,477$ *é a matriz distância euclidiana entre os indivíduos 1 e 2. Observa-se que o segundo e o terceiro indivíduo tiveram a menor distância euclidiana,* $d_{2,3} = d_{3,2} = 5,196$*. A maior distância foi entre o indivíduo sete e o segundo,* $d_{2,7} = d_{7,2} = 21,703$*. Verifica-se também que as distâncias entre um ponto a ele mesmo é zero, ou seja,* $d_{1,1} = d_{2,2} = d_{3,3} = \ldots = d_{10,10} = 0$*.*

Tabela 2.8: Matriz de distância euclidiana entre os 10 indivíduos

0	5,477	8,185	10,100	14,526	15,000	18,735	13,229	10,536	6,481
5,477	0	5,196	8,602	17,407	17,407	21,703	14,177	12,923	10,863
8,185	5,196	0	7,937	15,232	16,793	20,248	14,071	13,638	12,610
10,100	8,602	7,937	0	13,528	11,705	16,155	8,660	8,307	10,488
14,526	17,407	15,232	13,528	0	9,055	6,782	12,884	10,488	11,446
15,000	17,407	16,793	11,705	9,055	0	6,782	6,000	7,874	10,536
18,735	21,703	20,248	16,155	6,782	6,782	0	12,570	11,314	13,892
13,229	14,177	14,071	8,660	12,884	6,000	12,570	0	8,832	10,817
10,536	12,923	13,638	8,307	10,488	7,874	11,314	8,832	0	5,745
6,481	10,863	12,610	10,488	11,446	10,536	13,892	10,817	5,745	0

Como ilustração usaremos a técnica do vizinho mais próximo. Para simplificar a ilustração vamos considerar apenas 3 indivíduos.

Assim, a matriz de distância euclidiana será:

Tabela 2.9: Matriz de distância euclidiana entre os 3 primeiros indivíduos.

$D_{3x3} =$

	(1)	(2)	(3)
(1)	0	5,477	8,185
(2)	5,477	0	5,196
(3)	8,185	5,196	0

Aplicando o método do vizinho, considera-se a menor distância euclidiana, $d_{2,3} = 5,196$. *Assim,* $d_{(2,3)(1)} = min(d_{2,1}; d_{3,1}) = min(5,477; 8,185) = 5,477$. *A nova matriz distância* $D_{2\times 2} =$ *será então:*

Tabela 2.10: Matriz de distância euclidiana.

	(1)	(2,3)
(1)	0	5,477
(2,3)	5,477	0

O histórico de agrupamento para a etapa 1 será então:

Tabela 2.11: Processo de agrupamento hierárquico dos 3 indivíduos agrupados.

Etapa	Distância	Grupos	Número de agrupamento
1	5,477	(2,3)(1)	2

Pelo dendrograma, é possível observar que os indivíduos 2 e 3 estão unidos abaixo de uma distância de 5,196. A medida que nos afastamos da origem junta-se a esse bloco (2,3) o indivíduo 1. Visualiza-se que o indivíduo 1 está separado dos indivíduos (2,3) aproximadamente a uma distância de (5,477-5,196=0,281). Assim, observa-se que o indivíduo 2 e 3 apresentam características como quando ao nível de ansiedade (NA), nível de tristeza (T), problemas de matemática (P), e capacidade de tarefas simultâneas (TS) mais similares que o indivíduo 1.

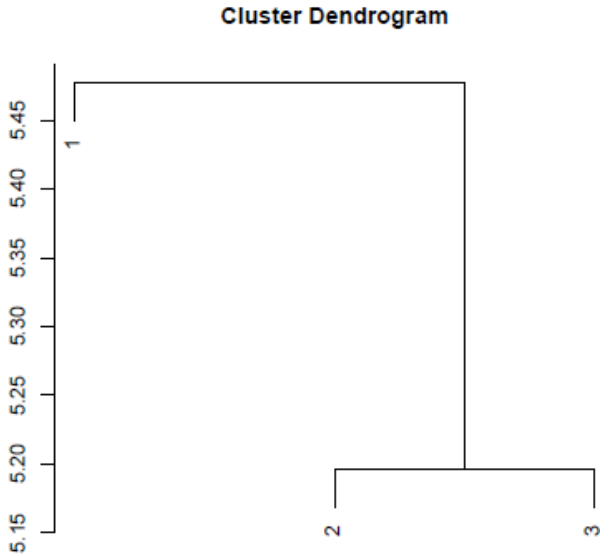


Figura 2.20: Dendrograma conforme critério do vizinho mais próximo - distância euclidiana.

Exercícios

1. *Dada a seguinte matriz de distâncias, realizar uma análise de agrupamento e o dendrograma correspondente com a metodologia do método do vizinho mais próximo (*linkaje simple aglomerativo*). Use o R!*

	A	*B*	*C*	*D*	*E*	*F*	*G*
A	*0*						
B	*2.15*	*0*					
C	*0.9*	*1.43*	*0*				
D	*1.07*	*1.14*	*0.43*	*0*			
E	*0.85*	*1.36*	*0.21*	*0.29*	*0*		
G	*1.16*	*1.02*	*0.55*	*0.32*	*0.51*	*0*	
G	*1.46*	*2.49*	*2.06*	*2.01*	*1.99*	*1.90*	*0*

2. *Dada a seguinte matriz distâncias, realizar uma análise de agrupamento e o dendrograma correspondente com a metodologia do método do vizinho mais distante (linkage completo aglomerativo).*

	A	*B*	*C*	*D*	*E*	*F*	*G*
A	*0*						
B	*5.65*	*0*					
C	*2.97*	*1.83*	*0*				
D	*3.80*	*2.54*	*1.42*	*0*			
E	*3.72*	*3.88*	*1.57*	*1.21*	*0*		
G	*5.89*	*4.33*	*5.35*	*5.13*	*4.08*	*0*	
G	*4.07*	*3.59*	*4.02*	*3.14*	*2.98*	*3.87*	*0*

3. *Considere os dados abaixo relativos a uma amostra de 6 indivíduos e as variáveis A, B e C. Pede-se:*

Indivíduos	***A***	***B***	***C***
1	*10*	*28*	*15*
2	*8*	*31*	*12*
3	*2*	*42*	*20*
4	*18*	*38*	*24*
5	*4*	*25*	*16*
6	*6*	*41*	*16*

(a) Preencher a Tabela do histórico de agrupamento - Apenas usando o método do vizinho mais próximo (ligação simples, nearest neighbor*).*

Passo	*No. grupos*	*Fusão*	*Distância (nível)*
1			
2			
3			
4			
5			

Algoritmo no R

```
A<-matrix(c(10,8,2,18,4,6), ncol=1, byrow=T)
B<-matrix(c(28,31,42,38,25,41),ncol=1,by=T)
C<-matrix(c(15,12,20,24,16,16),ncol=1,byrow=T)
X<-cbind(A,B,C)
scatterplot3d(A,B,C)
#Instalar o pacote scatterplot3d
plclust(hclust(dist(X), method="single"))
individuos<-row.names(X)
par(mfrow=c(1,2))
plclust(hclust(dist(X), method="single"),
labels=individuos, ylab="Distância");
title("(a) Vizinho mais próximo")
plclust(hclust(dist(X), method="complete"),
labels=individuos, ylab="Distância");
title("(b) Vizinho mais distante")
```

(a) A matriz de distância de Mahalanobis.

(b) A Matriz de distância Euclidiana.

(c) Preencher a mesma Tabela do histórico de agrupamento - Apenas usando o método do centroide.

(d) Usando as funções do pacote `dendextend` *compare os dois dendrogramas gerados acima.*

4. *Considerando dados de casos novos de uma enfermidade por grau de escolaridade nos Estados, mostrado na Tabela abaixo.*

Tabela 2.12

	UF	Analf	X1a4IncEF	X4CompEF	X5a8EFinc	EFC	EMInc	EMC	EdSupInc	EdSupComp
1	RO	38	111	32	122	47	56	79	16	16
2	AC	43	60	20	55	23	31	56	10	10
3	AM	131	431	158	413	193	262	476	71	86
4	RR	15	23	6	18	7	10	24	1	6
5	PA	205	631	218	615	215	327	534	64	99
6	AP	18	29	10	24	6	18	41	7	9
7	TO	13	44	17	21	9	10	19	6	11
8	MA	251	385	168	304	113	148	301	36	41
9	PI	125	184	44	110	50	57	104	18	37
10	CE	275	586	241	470	224	214	311	52	67
11	RN	115	173	77	123	121	53	113	16	28
12	PB	139	179	75	143	68	44	115	16	42
13	PE	311	610	287	624	211	222	345	57	120
14	AL	119	171	61	115	58	58	97	16	25
15	SE	40	150	32	124	37	42	77	16	21
16	BA	364	881	296	657	234	297	496	82	104
17	MG	182	485	246	350	191	155	281	49	98
18	ES	68	182	74	235	91	84	173	22	37
19	RJ	175	1158	735	1794	717	747	1442	233	433
20	SP	39	1495	0	4871	0	4955	0	952	459
21	PR	94	371	173	434	235	169	247	47	76
22	SC	53	225	166	404	194	136	207	50	75
23	RS	114	629	304	1339	458	409	454	83	134
24	MS	39	126	55	160	62	68	57	13	19
25	MT	108	228	124	298	144	157	206	36	83
26	GO	49	169	77	115	47	53	105	11	30
27	DF	18	51	39	63	21	27	39	10	32
28	Total	3141	9767	3735	14001	3776	8809	6399	1990	2198

Segue a legenda da Tabela de dados:

Tabela 2.13

UF	Unidade da Federação
Analf	Analfabeto
X1a4IncEF	1^a a 4^a ano do ensino fundamental incompleto
X4CompEF	1^a a 4^a ano do ensino fundamental completo
X5a8EFinc	5^a a 8^a ano do ensino fundamental completo
EFC	Ensino Fundamental Completo
EMInc	Ensino Fundamental incompleto
EMC	Ensino Médio Completo
EdSupInc	Ensino Superior Incompleto
EdSupComp	Ensino Superior Completo

Pede-se no R*:*

a) *Encontrar a dispersão gráfica das três variáveis (instalar o pacote:* `scartterplot3d`*).*

b) *Encontrar a matriz distância usando a distância Euclidiana entre as variáveis A, B e C (use o comando* `dist(X)`*).*

c) *Elaborar o dendrograma usando o método do vizinho mais próximo.*

d) *Elaborar o dendrograma usando o método do vizinho mais distante.*

e) *Elaborar o dendrograma usando o método do ward.*

f) *Usando as funções do pacote dendextend compare os dois dendrogramas gerados acima.*

g) *Encontrar o coeficiente cofenético (método do vizinho mais próximo).*

h) *Encontrar o coeficiente cofenético (método do vizinho mais distante).*

Referências Bibliográficas

ANDERBERG, M. R. ***Cluster analysis for applications****. London: Academic Press, p.359, 1973.*

ARABIE, P.; HUBERT, L. ***An overview of combinatorial data analysis, Clustering and Classification****. World Scientific Publishing Co., p.5-63, 1996.*

BACHE, S. M.; WICKHAM, H. ***magrittr: A Forward-Pipe Operator for R****. R package version 1.5. <https://CRAN.R-project.org/package=magrittr>. 2014.*

BARRET, Schloerke et al. ***GGally: Extension to ggplot2****. R package version 1.0.1. http://CRAN.R-project.org/package=GGally. 2016.*

BARROSO, L. P.; ARTES, R. ***Análise Multivariada****. In: REUNIÃO ANUAL DA RBES E SEAGRO, 48., 100, Curso. Lavras-MG: Departamento de Ciências Exatas, p.155, 2003.*

BIVAND, Roger S.; PEBESMA, Edezer; GÓMEZ-RUBIO, Virgilio. ***Applied Spatial Data Analysis with R****. Springer: USA, 2008.*

BUSSAB, W. O.; MIAZAKI, E. S.; ANDRADE, D. ***Introdução à análise de agrupamentos****. 9º Simpósio Nacional de Probabilidade e Estatística (Sinape), São Paulo. Associação Brasileira de Estatística, p.105, 1990.*

CATENA, A.; RAMOS, M.M.; TRUJILLO, H.M. ***Análisis multivariado. Un manual para investigadores****. Munuales Universidad. Biblioteca Nueva: Madrid, 2003.*

CHAN, C.; CH CHAN, G.; LEEPER, Thomas J.; BECKER, Jason. ***rio: A Swiss-army knife for data file I/O****. R package version 0.4.0, 2016.*

DAHL, David B. ***xtable: Export Tables to LaTeX or HTML****. R package version 1.8-2, 2016. http://CRAN.R-project.org/package=xtable*

FARRAR, D.E.; GLAUBER, R.R. ***Multicollinearity in regression analysis****. Review of Economics and Statistics, v.49, p.92-107, 1967.*

FACELI, K.; LORENA; A. C., GAMA, J.; CARVALHO, A. C. P. L. F. ***Inteligência artificial: Uma abordagem de aprendizado de máquina****. Rio de Janeiro: LTC, 2011.*

FERREIRA, D. F. ***Estatística multivariada****. Lavras: Editora Ufla, 2008.*

FOWLKES, E.B.; MALLOWS, C.L. ***A Method for Comparing Two Hierarchical Clusterings****. Journal of the American Statistical Association, 78 (383): 553, (1 September 1983).*

GALILI, Tal. ***dendextend: an R package for visualizing, adjusting, and comparing trees of hierarchical clustering****. Bioinformatics, 2015. DOI: 10.1093/bioinformatics/btv428*

GONZÁLEZ, Cástor G.; LISTE, Antonio V. ***Gráficos estadísticos y mapas con R****. Espanha: Diaz de Santos, 2012.*

HADLEY, Wickham; BRYAN, Jennifer. ***readxl: Read Excel Files****. R package version 0.1.1. http://CRAN.R-project.org/package=readxl. 2016.*

HADLEY, W.; FRANÇOIS, R.; HENRY, L.; MÜLLER, K. ***dplyr: A Grammar of Data Manipulation****. R package version 0.7.6. https://CRAN.R-project.org/package=dplyr. 2018.*

HAIR, J.F. et al. ***Análise Multivariada de Dados****. Tradução de Adonai Schlup Sant'Ana e Anselmo Chaves Neto. 5ª ed., p. 381-420. Porto Alegre: Bookman, 2005.*

HARTIGAN, John A. ***Clustering Algorithms*** *Probability & Mathematical Statistics. Jonh Wiley & Sons Inc. 1975.*

IMDADULLAH, Muhammad; ASLAM, Muhammad. ***mctest: Multicollinearity Diagnostic Measures****. R package version 1.1. <https://CRAN.R-project.org/package=mctest>. 2016.*

JOBSON, J. D. ***Applied Multivariate Data Analysis****. Categorical and Multivariate Methods. New York: Springer Verlag. Vol. 2, 1992.*

KAUFMAN, L.; ROUSSEEUW, P. J. ***Finding groups in data: an introduction to cluster analysis****. New York: John Wiley and Sons. 1990.*

KIM, Seongho. ***ppcor: Partial and Semi-Partial (Part) Correlation****. R package version 1.1. http://CRAN.R-project.org/package=ppcor. 2015.*

JIMENEZ, E. U.; MANZANO, J. A. ***Análisis Multivariante Aplicado****. Thomson, España, 2005.*

KING, Ronald S. ***Cluster Analysis and Data Mining: An Introduction****. Transatlantic Publishers: 2015*

LANDEIRO, V. M. ***Introdução ao uso do programa R****.*

MAINDONALD, J. H.; BRAUN, W. J. ***DAAG: Data Analysis and Graphics Data and Functions****. R package version 1.22, 2015. http://CRAN.R-project.org/package=DAAG*

MARDIA, K. V.; KENT, J. T.; BIBBY, J. M. ***Multivariate analysis****. Academic Press: London, New York, Toronto, Sydney, San Francisco. 1979.*

MURRELL, P. ***R Graphics****. Chapman & Hall/CRC Press. 2005.*

PARADIS, E. ***R for Beginners*** *. Institut des Sciences de l'Évolution.*

PEÑA, Daniel. ***Análisis de datos multivariantes****. McGraw-Hill: España, 2002.*

PEÑA, Daniel; RODRÍGUEZ, J. ***Descreptive Measures of Multivariate Scatter and Linear Dependences****. Journal of Multivariate Analysis, 2003.*

R CORE TEAM. ***R: A Language and Environment for Statistical Computing****. R Foundation for Statistical Computing, Vienna: Austria, 2017. http://www.R-project.org/*

R CORE TEAM. ***R: A Language and Environment for Statistical Computing****. R Foundation for Statistical Computing, Vienna: Austria, 2015. http://www.R-project.org/*

RAJARAM, Stawik; OONO, Yoshi. ***NeatMap: Non-clustered heat-map alternatives****. R package version 0.3.6.2, 2014. http://CRAN.R-project.org/package=NeatMap*

VASCONCELLOS, M.A.S., ALVES, D (editores). ***Manual de Econometria: nível intermediário****. São Paulo: Atlas, 2000.*

RENAUD GAUJOUX, Cathal Seoighe. ***flexible R package for non-negative matrix factorization****. BMC Bioinformatics 2010, 11:367. [http://www.biomedcentral.com/1471-2105/11/367]*

SCHÄFER, Juliane; et al. ***corpcor: Efficient Estimation of Covariance and (Partial) Correlation****. R package version 1.6.8. http://CRAN.R-project.org/package=corpcor. 2015.*

SILVA,I.N.; SPATTI,D.H.; FLAUZINO, R.A. ***Redes Neurais Artificiais. Para engenheiro e ciências aplicadas****. São Paulo: ArtLiber, 2010.*

SNEATH, P. H. A; SOKAL, R. R. ***Numeric taxonomy: the principles e practice of numerical classification****. San Francisco: W. H. Freeman, p.573, 1973.*

Timm, N. H. ***Applied multivariate ANALYSIS****. 2nd Edn., New York, SpringerVerlag. 2002.*

VAN DER LOO, M.P.J. ***extremevalues, an R package for outlier detection in univariate data****. R package version 2.3, 2010.*

VRIES, Andrie de; RIPLEY, Brian D. ***ggdendro: Create Dendrograms and Tree Diagrams Using 'ggplot2'****. R package version 0.1-20, 2016. http://CRAN.R-project.org/package=ggdendro*

WICKHAM, Hadley. ***ggplot2: Elegant Graphics for Data Analysis****. Springer-Verlag New York, 2009.*

WHITTAKER, Joe. ***Graphical Models in Applied Multivariate Statistics****. Wiley, 1990.*

WUERTZ, D.; SETZ, T.; CHALABI, Y. ***fBasics: Rmetrics - Markets and Basic Statistics****. RMETRICS Core Team: R package version 3011.87. <http://CRAN.R-project.org/package=fBasics>. 2014.*

ZHAO et al. ***Systematic Clustering of Transcription Start Site Landscapes****. PLoS ONE 6(8): e23409, 2011.*

Organizador e Autores

Edwirde Luiz Silva Camêlo (Brasil) *- (Organizador) Professor Associado da Universidade Estadual da Paraíba (UEPB). Pós-doutorado em* Estadística Aplicada *(2016) e Doutor em* Estadística e Investigación Operativa *(2007) pela* Universidad de Granada*. Mestrado em Biometria e Estatística Aplicada (2001) pela Universidade Federal Rural de Pernambuco (UFRPE). Técnicas de estatística multivariada aplicada em diversas áreas tem sido sua principal linha de pesquisa. E-mail: edwirde@uepb.edu.br*

Paulo J. G. Lisboa (Inglaterra) *- Professor e chefe do Departamento de Matemática Aplicada da* Liverpool John Moores University *(LJMU). Estudou física matemática na Universidade de Liverpool, onde obteve um doutorado em física teórica de partículas em 1983. Foi nomeado para a cátedra de Matemática Industrial na (LJMU) em 1996 e Chefe de Graduação em 2002. PhD em Física de Partículas (1983) e bacharel em Física matemática (1979) pela Universidade de Liverpool. Aplica ciência de dados para medicina personalizada, saúde pública, análise esportiva e marketing digital. Vice-presidente do Grupo Consultivo* Horizon2020 *para o Desafio Societário 1: Saúde, Mudança Demográfica e Bem-estar, fornecendo conselhos científicos a um dos maiores programas de pesquisa coordenada do mundo em saúde. Membro do Conselho do Instituto de Matemática e suas Aplicações. Presidente da Equipe de Tarefa de Análise de Dados Médicos no Comitê Técnico de Mineração de Dados do* IEEE*. Presidente do Comitê de Prêmio* JA Lodge *e presidente da Rede Profissional de Tecnologias de Saúde na*

Instituição de Engenharia e Tecnologia. Assessora o Group of Performance.Lab at Prozone *e tem papéis de revisão editorial e de pares em várias revistas e órgãos de financiamento da pesquisa, incluindo EPSRC. E-mail: P.J.Lisboa@ljmu.ac.uk*

Ramón Gutiérrez Sánchez (Espanha) *- Professor da* Universidad de Granada *(UGR), Secretário do Departamento de* Estadística e IO. *Doutor desde 2005, pertence à linha de pesquisa de Análise Multivariada e Processos Estocásticos. Publicou mais de 40 artigos em* JCR *na área de Estatística. Também colabora com o Departamento de Parasitologia da UGR. E-mail: ramongs@ugr.es*

Dalila Camêlo Aguiar (Brasil) *- Doutoranda em* Estadística Matemática y Aplicada *pela* Universidad de Granada *(UGR), Mestra em* Estadística Aplicada *(2016) pela UGR, Especialista em Estatística Aplicada (2011) pela Fundação de Apoio, Pesquisa e Extensão (FURNE) e Bacharela em Estatística (2010) pela Universidade Estadual da Paraíba (UEPB). Pesquisadora na área de estatística multivariada aplicada. E-mail: dalilacamelo@correo.ugr.es*