MAS: Trabalho de Grupo 3

Diogo Alexandre Alonso De Freitas

21 de março, 2023

Preencher a identificação do grupo:
NÚMERO DO GRUPO: Grupo 3

LISTA DE TODOS OS ELEMENTOS DO GRUPO (Número - nome):

103380 - Allan Kardec 104841 - Diogo Freitas 104782 - João Botas 104826 - Ricardo Ângelo

O Trabalho de Grupo de *Métodos de Aprendizagem Supervisionada* refere-se à análise do data set "Consumo.Jovens.csv".

Neste data set incluem-se 1523 registos e 28 atributos listados a seguir:

q0: País de residência
q1: Sexo
q2: Idade
q3: Situação estudantil
q10: Compra produtos de marca? (1-Sim; 2-Não)
q12b_a: Compra em centros comerciais? (1-Sim; 0-Não)
q12b_b: Compra em super/hipermercados? (1-Sim; 0-Não)
q12b_c: Compra no comércio local? (1-Sim; 0-Não)
q13a: Fidelidade a marcas? (1-Sim; 0-Não)
q13b: Fidelidade a lojas? (1-Sim; 0-Não)

Variáveis q14 na Escala 1-Nada Importante, 2, 3, 4, 5-Extremamente importante)
q14a: Preço
q14b: Necessidade do produto
q14c: Conveniência da localização da loja
q14d: Qualidade do produto
q14e: Imagem do produto
q14f: Imagem da loja
q14g: Características do produto
q14h: Promoção especial
q14i: Imagem da marca
q14j: Publicidade

Variáveis q19 na Escala 1-Discordo Completamente, 2, 3, 4, 5-Concordo Completamente)
q19_1: Alguns dos feitos + importantes da vida incluem adquirir bens materiais
q19_2: Não dou importância à quantidade de bens materiais
q19_3: Gosto de ter coisas para impressionar as pessoas

q19_4: Geralmente compro apenas aquilo de que preciso
q19_5: Gosto de gastar dinheiro em coisas que não são necessárias
q19_6: Comprar coisas dá-me imenso prazer
q19_7: Tenho todas as coisas de que preciso para ser feliz
q19_8: Seria mais feliz se tivesse dinheiro para comprar mais coisas

Notas:

1. Efetuar todos os Save com "Save with encoding UTF-8" de modo a manter palavras acentuadas e caracteres especiais**
2. A cotação está anexa a cada pergunta
3. **OS ALUNOS QUE NÃO SUBMETEREM PDF NO MOODLE TERÃO UMA PENALIZAÇÃO DE 1 VALOR; SE, O FICHEIRO ALTERNATIVO QUE SUBMETEREM (VIA EMAIL) REPORTAR ERROS NA COMPILAÇÃO, TERÃO UMA PENALIZAÇÃO ADICIONAL DE 1 VALOR**

```
# Remover tudo!
rm(list=ls(all=TRUE))# Remove everything!
# Incluir as libraries de que necessita
library(tree)
library(e1071)
library(knitr)
library(MASS) # with Boston data set
library(Metrics) # metrics for evaluation of results
library(FNN)
library(psych)# for some descriptives
library(nnet) # for Multinomial Logistic Regression
library(car)# to verify multicolinearity
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:psych':
##
##     logit
```

```
library(lsr)# for eta and Cramer's V measure of association
library(caret)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'

## The following objects are masked from 'package:Metrics':
##
##     precision, recall

library(ggplot2)
```

1. Leitura dos dados "Consumo.Jovens.csv" e análise preliminar dos mesmos

1.1) [1 valor] Leitura dos dados; apresentação de dimensão e estrutura dos dados; verificação do número de casos com dados em falta (para todos os atributos); sumário dos dados completos (depois de eliminação dos casos/linhas com dados omissos)

```
#Leitura dos dados (Nota: verifique sep no ficheiro de origem)
CJ<-read.csv("Consumo.Jovens.csv", header=TRUE, dec=".",na.strings="",
sep=";",stringsAsFactors = TRUE)

# apresentação de dimensão e estrutura dos dados.
Dimension <- c(nrow(CJ), length(CJ))
Dimension

## [1] 1523   28

#or
nrow(CJ) # Nº Linhas

## [1] 1523

length(CJ) # Nº Colunas

## [1] 28

#or
dim(CJ)

## [1] 1523   28

str(CJ)

## 'data.frame':    1523 obs. of  28 variables:
##  $ q0    : Factor w/ 6 levels "Alemanha","China",..: 6 6 6 6 6 6 6 6 6
6 ...
##  $ q1    : Factor w/ 2 levels "Feminino","Masculino": 1 2 1 2 1 2 1 2
2 2 ...
##  $ q2    : int  19 19 19 20 21 19 20 21 20 22 ...
##  $ q3    : Factor w/ 3 levels "Estudante-trabalhador",..: 2 2 2 2 2 2
```

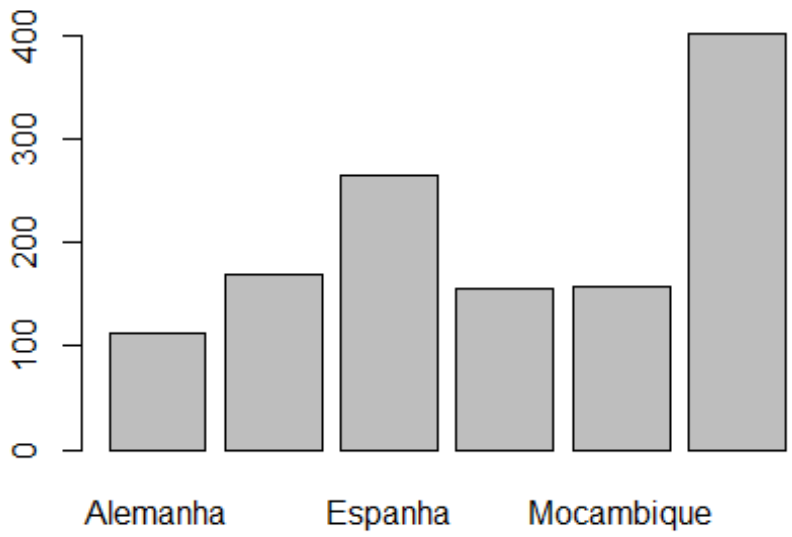
```
2 2 2 2 ...
##  $ q10   : Factor w/ 2 levels "Nao","Sim": 2 2 1 2 2 2 2 2 2 2 ...
##  $ q12b_a: int  0 1 0 1 1 1 1 0 1 1 ...
##  $ q12b_b: int  1 0 0 1 0 1 1 0 1 1 ...
##  $ q12b_c: int  0 1 1 0 0 0 0 1 0 0 ...
##  $ q13a  : int  0 1 0 1 0 0 1 1 0 1 ...
##  $ q13b  : int  0 1 1 0 0 0 0 0 1 NA ...
##  $ q14a  : int  5 4 5 3 5 3 3 5 3 3 ...
##  $ q14b  : int  3 3 5 5 3 4 5 3 5 3 ...
##  $ q14c  : int  NA 2 3 2 2 1 2 1 2 2 ...
##  $ q14d  : int  4 5 5 4 5 3 5 4 3 4 ...
##  $ q14e  : int  NA 3 3 2 4 2 3 2 3 3 ...
##  $ q14f  : int  NA 4 4 2 3 1 3 1 2 2 ...
##  $ q14g  : int  NA 5 5 3 4 4 4 3 4 2 ...
##  $ q14h  : int  NA 2 2 2 2 3 4 2 3 2 ...
##  $ q14i  : int  1 3 3 3 3 2 3 2 3 2 ...
##  $ q14j  : int  2 3 3 2 2 2 4 1 2 2 ...
##  $ q19_1 : int  NA 2 4 NA 3 1 NA 2 2 2 ...
##  $ q19_2 : int  NA 4 4 NA 5 4 NA 4 2 4 ...
##  $ q19_3 : int  NA 1 1 NA 2 1 NA 2 3 3 ...
##  $ q19_4 : int  NA 3 5 NA 3 4 NA 5 3 3 ...
##  $ q19_5 : int  NA 3 1 NA 3 1 NA 2 3 2 ...
##  $ q19_6 : int  NA 3 3 NA 3 3 NA 3 3 1 ...
##  $ q19_7 : int  NA 4 3 NA 5 2 NA 3 5 3 ...
##  $ q19_8 : int  NA 3 5 NA 3 4 NA 4 4 4 ...

# Verificação do número de casos com dados em falta (para todos os
atributos)
colSums(is.na(CJ))

##     q0     q1     q2     q3    q10 q12b_a q12b_b q12b_c   q13a   q13b
q14a
##      0      5      0     21     44      4      5      7     60     70
13
##   q14b   q14c   q14d   q14e   q14f   q14g   q14h   q14i   q14j  q19_1
q19_2
##     19     24     14     20     23     23     21     19     23     46
48
##  q19_3  q19_4  q19_5  q19_6  q19_7  q19_8
##     46     44     52     47     52     53

# eliminação dos casos/linhas com dados omissos
CJ<-na.omit(CJ)

# sumário dos dados completos

summary(CJ)

##           q0              q1            q2
##  Alemanha  :113   Feminino :727   Min.   :17.00
##  China     :170   Masculino:538   1st Qu.:20.00
```

```
##  Espanha   :266                   Median :21.00
##  Macau     :156                   Mean   :21.19
##  Mocambique:158                   3rd Qu.:23.00
##  Portugal  :402                   Max.   :25.00
##                           q3        q10          q12b_a
q12b_b
##  Estudante-trabalhador    : 116   Nao:556   Min.   :0.0000   Min.
:0.0000
##  Estudante a tempo inteiro:1044   Sim:709   1st Qu.:0.0000   1st
Qu.:0.0000
##  Outra                    : 105             Median :1.0000   Median
:0.0000
##                                             Mean   :0.5209   Mean
:0.3621
##                                             3rd Qu.:1.0000   3rd
Qu.:1.0000
##                                             Max.   :1.0000   Max.
:1.0000
##      q12b_c            q13a             q13b             q14a
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:3.000
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :4.000
##  Mean   :0.4791   Mean   :0.4198   Mean   :0.4806   Mean   :3.696
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:4.000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :5.000
##       q14b            q14c            q14d            q14e
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:3.000   1st Qu.:2.000   1st Qu.:4.000   1st Qu.:2.000
##  Median :4.000   Median :3.000   Median :4.000   Median :3.000
##  Mean   :3.704   Mean   :2.553   Mean   :4.029   Mean   :2.952
##  3rd Qu.:4.000   3rd Qu.:3.000   3rd Qu.:5.000   3rd Qu.:4.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##       q14f            q14g            q14h            q14i
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.000   1st Qu.:3.000   1st Qu.:2.000   1st Qu.:2.000
##  Median :2.000   Median :4.000   Median :3.000   Median :3.000
##  Mean   :2.544   Mean   :3.496   Mean   :2.651   Mean   :2.675
##  3rd Qu.:3.000   3rd Qu.:4.000   3rd Qu.:3.000   3rd Qu.:3.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000
##       q14j            q19_1           q19_2           q19_3
q19_4
##  Min.   :1.000   Min.   :1.00   Min.   :1.000   Min.   :1.000   Min.
:1.0
##  1st Qu.:2.000   1st Qu.:2.00   1st Qu.:3.000   1st Qu.:1.000   1st
Qu.:2.0
##  Median :2.000   Median :3.00   Median :4.000   Median :2.000   Median
:3.0
##  Mean   :2.192   Mean   :2.91   Mean   :3.404   Mean   :2.436   Mean
:3.3
##  3rd Qu.:3.000   3rd Qu.:4.00   3rd Qu.:4.000   3rd Qu.:3.000   3rd
```

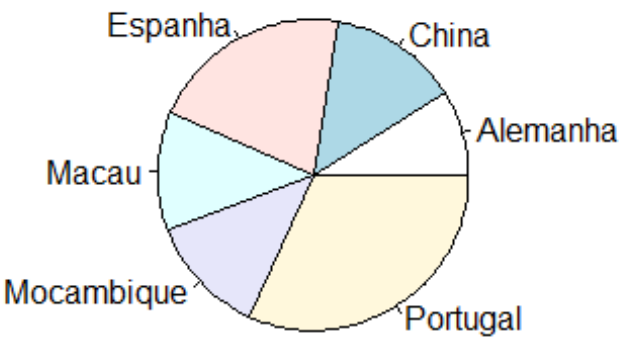
```
Qu.:4.0
##  Max.   :5.000   Max.   :5.00   Max.   :5.000   Max.   :5.000   Max.
:5.0
##      q19_5           q19_6          q19_7           q19_8
##  Min.   :1.000   Min.   :1.00   Min.   :1.000   Min.   :1.00
##  1st Qu.:2.000   1st Qu.:3.00   1st Qu.:2.000   1st Qu.:3.00
##  Median :2.000   Median :3.00   Median :3.000   Median :3.00
##  Mean   :2.387   Mean   :3.27   Mean   :2.947   Mean   :3.24
##  3rd Qu.:3.000   3rd Qu.:4.00   3rd Qu.:4.000   3rd Qu.:4.00
##  Max.   :5.000   Max.   :5.00   Max.   :5.000   Max.   :5.00
```

1.2) [1.5 valores] Breve análise descritiva de q0, q1, q2 e q3.

```
#q0: País de residência
table(CJ[,1])

##
##   Alemanha      China    Espanha      Macau Mocambique   Portugal
##        113        170        266        156        158        402

barplot(table(CJ[,1]))
```

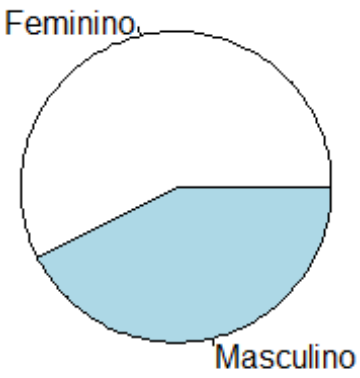



```
pie(table(CJ[,1]))
```



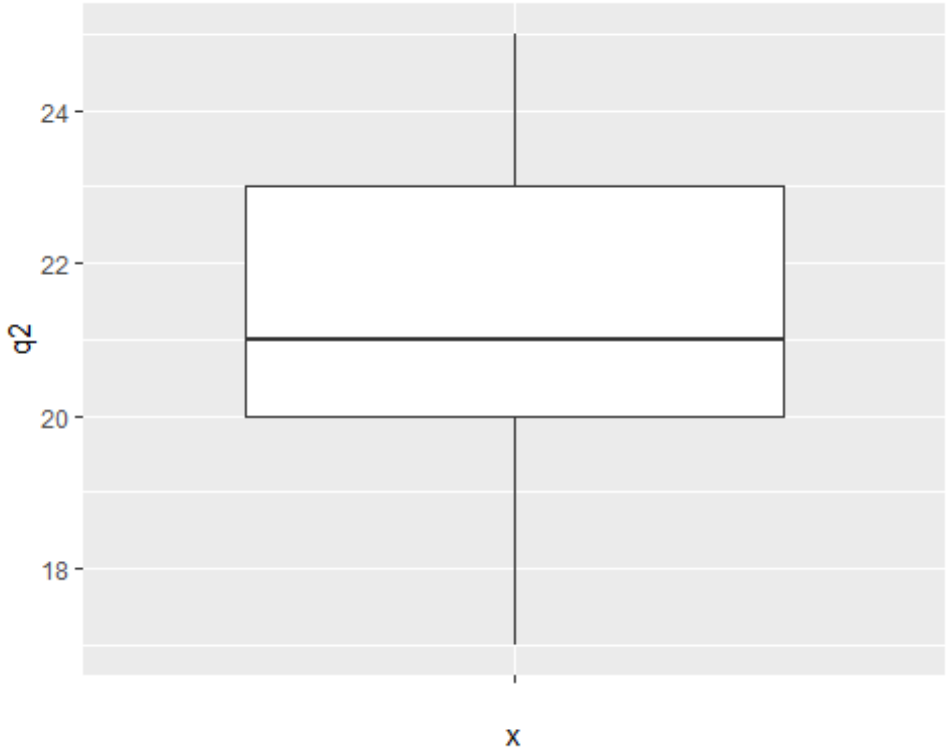

```
prop.table(table(CJ[,1]))

##
##   Alemanha      China    Espanha      Macau Mocambique   Portugal
## 0.08932806 0.13438735 0.21027668 0.12332016 0.12490119 0.31778656

# Nesta base de dados, existe um maior número de alunos com país de
residência em Portugal

#q1: Sexo
table(CJ[,2])

##
##  Feminino Masculino
##       727       538

barplot(table(CJ[,2]))
```


```
pie(table(CJ[,2]))
```




```
prop.table(table(CJ[,2]))
```

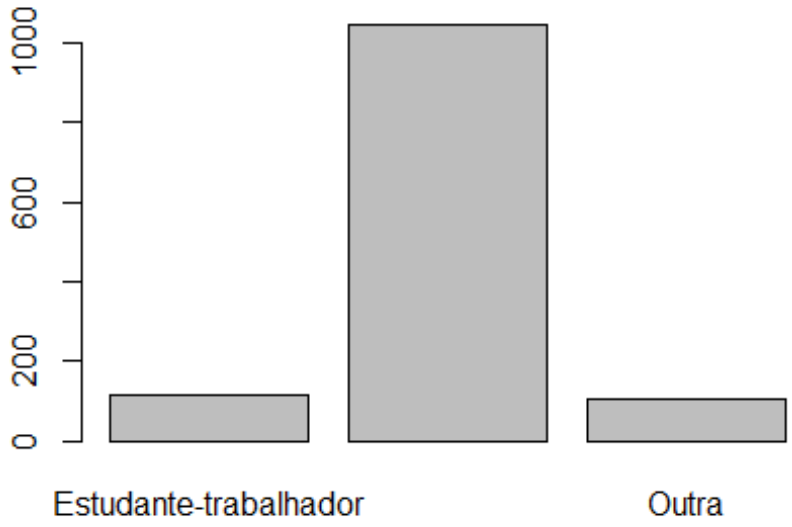
```
##
##  Feminino Masculino
## 0.5747036 0.4252964

# Nesta base de dados, existe mais alunos do sexo femenino do que
masculino

#q2: Idade
describe(CJ[,3])

##    vars    n  mean   sd median trimmed  mad min max range skew
kurtosis   se
## X1    1 1265 21.19 1.96     21   21.13 1.48  17  25     8 0.23    -
0.77 0.06

ggplot(CJ, aes(x = "", y = q2)) + geom_boxplot()
```



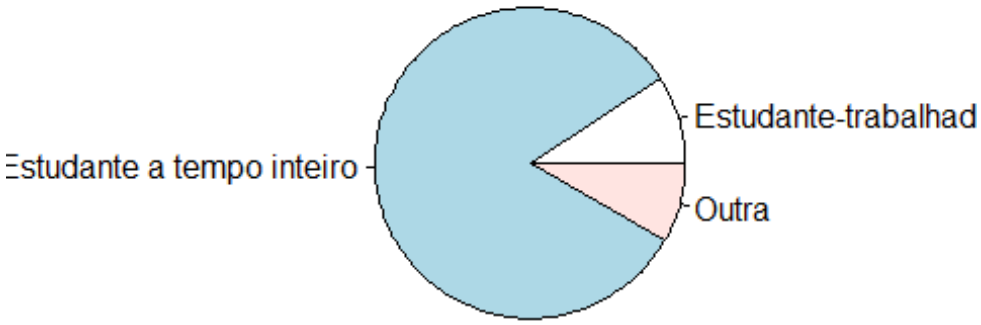

```
quantile(CJ$q2)

##   0%  25%  50%  75% 100%
##   17   20   21   23   25

# A média de idade dos alunos desta base de dados é de 21.19 anos, com um
desvio padrão de 1.96 alunos

#q3: Situação estudantil
table(CJ[,4])
```

```
##
##     Estudante-trabalhador Estudante a tempo inteiro
Outra
##                       116                      1044
105
```

```
barplot(table(CJ[,4]))
```




```
pie(table(CJ[,4]))
```




```
prop.table(table(CJ[,4]))

##
##     Estudante-trabalhador Estudante a tempo inteiro
Outra
##                0.09169960                0.82529644
0.08300395

# Mais de 80% dos alunos desta base de dados são estudantes a tempo
inteiro
```

1.3) [1.5 valores] Cálculo (e apresentação) de medidas de associação entre as variáveis: a) q14a…q14j; b) q0 e as variáveis q19_1…q19_8; c) q10 e q1

```
Eta_ <- function(y,x){
  freqk <- as.vector(table(x))
  l <- nlevels(x)
  m <- rep(NA,1)
  qual <- as.numeric(x)
  for (k in 1:1) {m[k] <- mean(y[qual == k])}
  return(sqrt(sum(freqk*(m-mean(y))^2)/sum((y-mean(y))^2)))}

#a) q14a…q14j

knitr::kable(corr.CJ<-round(cor(CJ[,11:20]),2))
```

	q14a	q14b	q14c	q14d	q14e	q14f	q14g	q14h	q14i	q14j

	q14a	q14b	q14c	q14d	q14e	q14f	q14g	q14h	q14i	q14j
q14a	1.00	0.08	0.05	-0.05	-0.07	-0.12	-0.01	0.15	-0.11	-0.06
q14b	0.08	1.00	0.07	0.15	0.00	-0.01	0.22	0.15	0.01	-0.09
q14c	0.05	0.07	1.00	0.10	0.13	0.20	0.05	0.11	0.12	0.19
q14d	-0.05	0.15	0.10	1.00	0.24	0.20	0.22	-0.04	0.19	0.10
q14e	-0.07	0.00	0.13	0.24	1.00	0.50	0.27	0.13	0.51	0.27
q14f	-0.12	-0.01	0.20	0.20	0.50	1.00	0.27	0.14	0.40	0.32
q14g	-0.01	0.22	0.05	0.22	0.27	0.27	1.00	0.20	0.18	0.08
q14h	0.15	0.15	0.11	-0.04	0.13	0.14	0.20	1.00	0.20	0.22
q14i	-0.11	0.01	0.12	0.19	0.51	0.40	0.18	0.20	1.00	0.44
q14j	-0.06	-0.09	0.19	0.10	0.27	0.32	0.08	0.22	0.44	1.00

```
#b) q0 e as variáveis q19_1…q19_8
cramersV(CJ$q0, CJ$q19_1)

## [1] 0.2079923

cramersV(CJ$q0, CJ$q19_2)

## [1] 0.1746758

cramersV(CJ$q0, CJ$q19_3) # Maior

## [1] 0.3678313

cramersV(CJ$q0, CJ$q19_4)

## [1] 0.1604731

cramersV(CJ$q0, CJ$q19_5)

## Warning in stats::chisq.test(...): Chi-squared approximation may be
incorrect

## [1] 0.1967741

cramersV(CJ$q0, CJ$q19_6)

## [1] 0.1271594

cramersV(CJ$q0, CJ$q19_7)

## [1] 0.242913

cramersV(CJ$q0, CJ$q19_8)

## [1] 0.1376108

# c) q10 e q1

cramersV(CJ$q10, CJ$q1)
```

```
## [1] 0.07398348
```

1.4) [1 valor] Divisão dos dados em amostra de treino (60%)- CJ.train - e de teste (40%) – CJ.test - usando set.seed(444);apresentação de tabela de frequências relativas de q1 em cada amostra

```
set.seed(444)

#CJ.train
ind_train <- sample(nrow(CJ),.60*nrow(CJ))
CJ.train <- CJ[ind_train,]
#CJ.test
CJ.test <- CJ[-ind_train,]

# tabela de frequencias de q1 para o conjunto de treino
prop.table(table(CJ.train$q1))

##
##  Feminino Masculino
## 0.5770751 0.4229249

# tabela de frequencias de q1 para o conjunto de teste
prop.table(table(CJ.test$q1))

##
##  Feminino Masculino
## 0.5711462 0.4288538
```

1.5) [1 valor] Completação das frases seguintes:

Inicialmente, o número de casos omissos na variável q1 era **5**. No conjunto de dados em análise (depois de eliminar os registos com observações omissas) o número de estudantes trabalhadores é igual a **116**. A correlação mais elevada entre o pares de variáveis q14 tem o valor **0.51**. A correlação maior entre a variável q0 e as variáveis q19_ regista-se para a variável q19_**3**

```
# Resposta 1
CJ_Delete<-read.csv("Consumo.Jovens.csv", header=TRUE,
dec=".",na.strings="", sep=";",stringsAsFactors = TRUE)
sum(is.na(CJ_Delete$q1))

## [1] 5

rm(CJ_Delete)

# Resposta 2
sum(grepl("Estudante-trabalhador", CJ$q3))

## [1] 116
```

```r
# Resposta 3
knitr::kable(corr.CJ<-round(cor(CJ[,11:20]),2)) # Visualizar os valores maiores
```

	q14a	q14b	q14c	q14d	q14e	q14f	q14g	q14h	q14i	q14j
q14a	1.00	0.08	0.05	-0.05	-0.07	-0.12	-0.01	0.15	-0.11	-0.06
q14b	0.08	1.00	0.07	0.15	0.00	-0.01	0.22	0.15	0.01	-0.09
q14c	0.05	0.07	1.00	0.10	0.13	0.20	0.05	0.11	0.12	0.19
q14d	-0.05	0.15	0.10	1.00	0.24	0.20	0.22	-0.04	0.19	0.10
q14e	-0.07	0.00	0.13	0.24	1.00	0.50	0.27	0.13	0.51	0.27
q14f	-0.12	-0.01	0.20	0.20	0.50	1.00	0.27	0.14	0.40	0.32
q14g	-0.01	0.22	0.05	0.22	0.27	0.27	1.00	0.20	0.18	0.08
q14h	0.15	0.15	0.11	-0.04	0.13	0.14	0.20	1.00	0.20	0.22
q14i	-0.11	0.01	0.12	0.19	0.51	0.40	0.18	0.20	1.00	0.44
q14j	-0.06	-0.09	0.19	0.10	0.27	0.32	0.08	0.22	0.44	1.00

```r
# Resposta 4
cramersV(CJ$q0, CJ$q19_3) # Maior
```

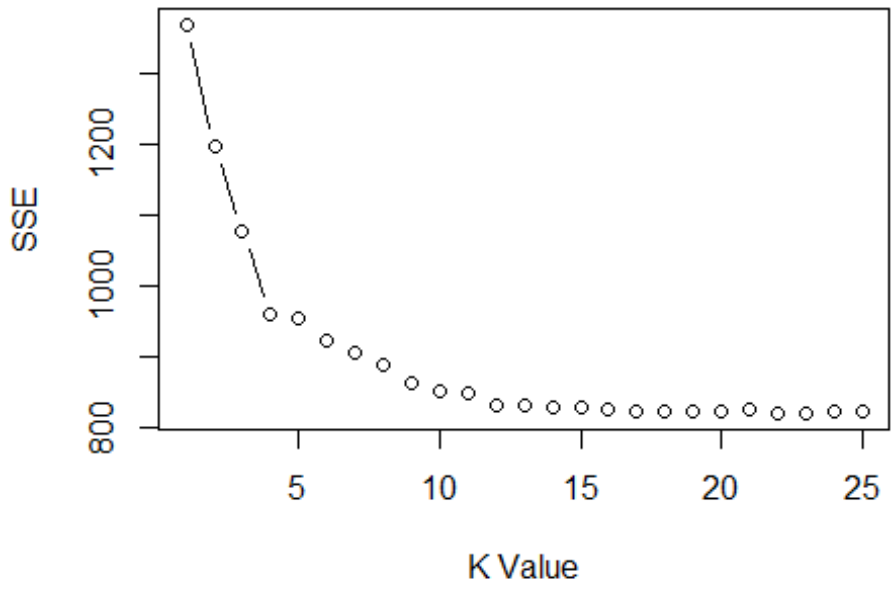
```
## [1] 0.3678313
```

2. Regressão: utilização do K-Nearest Neighbour para prever q19_8 com base nas variáveis q12b_a , q12b_b, q12b_c, q13a e q13b.

2.1) [2 valores] Aprendizagem sobre CJ.train[,c(6:10)] e considerando y=y=CJ.train$q19_8 recorrendo a one-hold-out validation; determinação de um "melhor" valor de K atendendo ao Sum of Squares Error

```r
# normalizar os dados
normalize_s <- function(x){
 return ((x -min(x)) / (max(x)-min(x)))}
# training set
CJ.train_s<-CJ.train
CJ.train_s [,6:10]<-sapply(CJ.train[,6:10],normalize_s)
# test set
CJ.test_s<-CJ.test
CJ.test_s [,6:10]<-sapply(CJ.test[,6:10],normalize_s)

# Para o knn é muito comum usar a normalização 0-1 (já estava de raiz na base de dados, mas foi aplicado à mesma)

k.sse<-matrix(NA,25,2)
for (i in 1:25){
 knn.CJ_k <- knn.reg(CJ.train_s[,c(6:10)],y=CJ.train_s$q19_8,k=i)
```

```
 k.sse[i,1]<-i
 k.sse[i,2] <- sse(CJ.train_s$q19_8, knn.CJ_k$pred) #
sse(actual,predicted)
}

plot(k.sse[,2], type="b", xlab="K Value",ylab="SSE")
```




```
(k.sse_sort<-k.sse[order(k.sse[,2],decreasing=FALSE),])
```

```
##       [,1]     [,2]
##  [1,]   22  820.7169
##  [2,]   23  821.0851
##  [3,]   19  821.8892
##  [4,]   24  822.6632
##  [5,]   17  822.7474
##  [6,]   20  823.0125
##  [7,]   25  823.1408
##  [8,]   18  823.3302
##  [9,]   21  825.3696
## [10,]   16  826.9844
## [11,]   15  828.1111
## [12,]   14  829.6582
## [13,]   13  830.8107
## [14,]   12  832.1389
## [15,]   11  848.3471
## [16,]   10  851.1600
## [17,]    9  862.7901
```

```
## [18,]    8  887.8438
## [19,]    7  906.8980
## [20,]    6  924.6389
## [21,]    5  954.0400
## [22,]    4  960.8125
## [23,]    3 1077.3333
## [24,]    2 1198.0000
## [25,]    1 1370.0000

(best_k<-k.sse_sort[1,1])

## [1] 22
```

2.2) [2 valores] Considerando o "melhor" valor de K (v. 2.1), obtenção de estimativas do alvo e listagem dos 6 primeiros valores estimados nos conjuntos CJ.train e CJ.test

```
knn.CJ_22_test <- knn.reg(CJ.train_s[,c(6:10)], CJ.test_s[,c(6:10)],
y=CJ.train_s$q19_8, k=best_k)
knn.CJ_22_train <- knn.reg(CJ.train_s[,c(6:10)], y=CJ.train_s$q19_8,
k=best_k)

# estimativas sobre CJ.test
knn.CJ_22_test$pred[1:6]

## [1] 3.227273 3.045455 3.227273 3.409091 3.090909 3.136364

# estimativas sobre CJ train
knn.CJ_22_train$pred[1:6]

## [1] 3.181818 3.000000 3.136364 3.090909 3.500000 3.181818
```

2.3) [2 valores] Determinação de Sum of Squares Error e de Root Mean Squared Error (RMSE) correspondentes às estimativas obtidas pelo KNN em 2.2) para as amostras CJ.train e CJ.test

```
# Métricas sobre CJ.train

# sse(actual, predicted) in R Metrics Library
sse(CJ.train_s$q19_8, knn.CJ_22_train$pred)

## [1] 820.7169

# rmse(actual, predicted) in R Metrics Library
rmse(CJ.train_s$q19_8, knn.CJ_22_train$pred)

## [1] 1.039862

# Métricas sobre CJ.test

# sse(actual, predicted) in R Metrics Library
sse(CJ.test_s$q19_8, knn.CJ_22_test$pred)
```

```
## [1] 570.3285

#rmse(actual, predicted) in R Metrics Library
rmse(CJ.test_s$q19_8, knn.CJ_22_test$pred)

## [1] 1.061664
```

2.4) [1 valor] Completação das frases seguintes:

O "melhor" valor de K, para K-NN, obtido segundo validação hold-one-out sobre a amostra de treino é **22**;o valor estimado do alvo para a 1ª observação do conjunto de teste é **3.227273**; neste conjunto obtém-se um RMSE de **1.061664** e um SSE de **570.3285**.

```
# Pergunta 1
(best_k<-k.sse_sort[1,1])

## [1] 22

# Pergunta 2
knn.CJ_22_test$pred[1]

## [1] 3.227273

# Pergunta 3
rmse(CJ.test_s$q19_8, knn.CJ_22_test$pred)

## [1] 1.061664

# Pergunta 4
sse(CJ.test_s$q19_8, knn.CJ_22_test$pred)

## [1] 570.3285
```

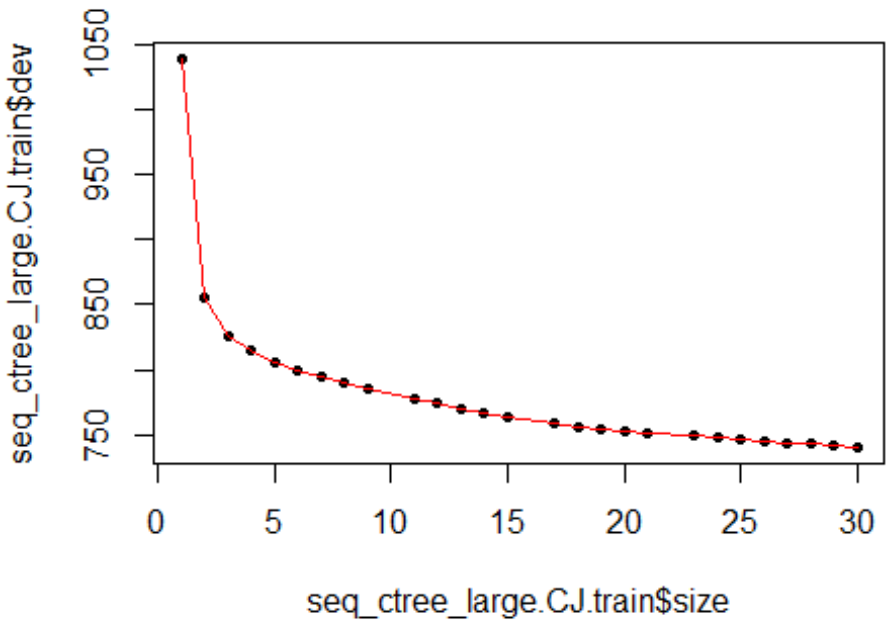
3. Classificação: utilização de uma Árvore para prever q10 (Compra ou não compra produtos de marca) considerando 4 preditores: q12b_a, q13a, q14e e q14i.

3.1) [2 valores] Construção de uma Árvore de classificação sobre CJ.train efetuando a sua poda de modo a fixar 15 nós folha (para prever q10 com base nos preditores q12b_a, q13a, q14e e q14i)

```
ctree_large.CJ.train<-tree(q10~q12b_a + q13a + q14e + q14i,
                           data=CJ.train,
                           control=tree.control(nrow(CJ.train),
                                                mincut = 1, minsize =
2, mindev = 0.001), split = "deviance")
summary(ctree_large.CJ.train)
```

```
##
## Classification tree:
## tree(formula = q10 ~ q12b_a + q13a + q14e + q14i, data = CJ.train,
##     control = tree.control(nrow(CJ.train), mincut = 1, minsize = 2,
##         mindev = 0.001), split = "deviance")
## Number of terminal nodes:  30
## Residual mean deviance:  1.015 = 740.1 / 729
## Misclassification error rate: 0.2437 = 185 / 759
```

```
# Prunning of Tree (teste de complexidade)
seq_ctree_large.CJ.train<-prune.tree(ctree_large.CJ.train)
plot(seq_ctree_large.CJ.train$size,seq_ctree_large.CJ.train$dev,pch =20)
lines(seq_ctree_large.CJ.train$size,seq_ctree_large.CJ.train$dev, col =
"red")
```



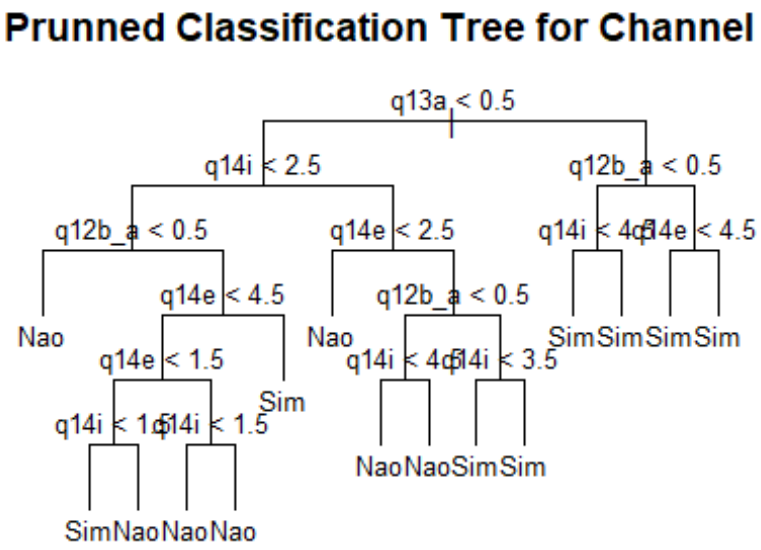

```
ctree.CJ.train<-prune.tree(ctree_large.CJ.train, best= 15)
```

3.2) [2 valores] Representações da Árvore de Classificação: a) Lista indentada; b) Gráfico da Árvore

```
# a)
print(ctree.CJ.train, indent = TRUE)
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 759 1040.000 Sim ( 0.43610 0.56390 )
##    2) q13a < 0.5 450  592.500 Nao ( 0.63111 0.36889 )
```

```
##      4) q14i < 2.5 246  279.900 Nao ( 0.74390 0.25610 )
##        8) q12b_a < 0.5 133  122.500 Nao ( 0.82707 0.17293 ) *
##        9) q12b_a > 0.5 113  146.900 Nao ( 0.64602 0.35398 )
##         18) q14e < 4.5 108  137.500 Nao ( 0.66667 0.33333 )
##           36) q14e < 1.5 10   13.860 Nao ( 0.50000 0.50000 )
##             72) q14i < 1.5 8   10.590 Sim ( 0.37500 0.62500 ) *
##             73) q14i > 1.5 2    0.000 Nao ( 1.00000 0.00000 ) *
##           37) q14e > 1.5 98  122.300 Nao ( 0.68367 0.31633 )
##             74) q14i < 1.5 20   13.000 Nao ( 0.90000 0.10000 ) *
##             75) q14i > 1.5 78  102.900 Nao ( 0.62821 0.37179 ) *
##         19) q14e > 4.5 5    5.004 Sim ( 0.20000 0.80000 ) *
##      5) q14i > 2.5 204  282.800 Sim ( 0.49510 0.50490 )
##       10) q14e < 2.5 41   52.640 Nao ( 0.65854 0.34146 ) *
##       11) q14e > 2.5 163  224.600 Sim ( 0.45399 0.54601 )
##         22) q12b_a < 0.5 72   98.920 Nao ( 0.55556 0.44444 )
##           44) q14i < 4.5 69   95.290 Nao ( 0.53623 0.46377 ) *
##           45) q14i > 4.5 3    0.000 Nao ( 1.00000 0.00000 ) *
##         23) q12b_a > 0.5 91  120.300 Sim ( 0.37363 0.62637 )
##           46) q14i < 3.5 68   92.790 Sim ( 0.42647 0.57353 ) *
##           47) q14i > 3.5 23   24.080 Sim ( 0.21739 0.78261 ) *
##    3) q13a > 0.5 309  263.500 Sim ( 0.15210 0.84790 )
##      6) q12b_a < 0.5 117  128.800 Sim ( 0.23932 0.76068 )
##       12) q14i < 4.5 112  126.000 Sim ( 0.25000 0.75000 ) *
##       13) q14i > 4.5 5    0.000 Sim ( 0.00000 1.00000 ) *
##      7) q12b_a > 0.5 192  124.000 Sim ( 0.09896 0.90104 )
##       14) q14e < 4.5 171  119.300 Sim ( 0.11111 0.88889 ) *
##       15) q14e > 4.5 21    0.000 Sim ( 0.00000 1.00000 ) *

# b)

plot(ctree.CJ.train, type="uniform")
text(ctree.CJ.train,pretty = 0,cex=0.8)
title(main = "Prunned Classification Tree for Channel")
```



3.3) [2 valores] Obtenção, sobre as amostras CJ.train e CJ.test, das "Matrizes de Confusão" e correspondentes medidas Accuracy associadas à Árvore de Classificação

```
# "Matriz de confusão" e accuracy sobre CJ.train #

probs.ctree.CJ.train<-predict(ctree.CJ.train,CJ.train,type="vector") #
the default type

## Confusion Matrix
pred.Train <-apply(probs.ctree.CJ.train,1,which.max)

pred.Train<-factor(pred.Train, levels = c(1,2), labels = c("Nao", "Sim"))

(confusion_mat_train<-table(CJ.train$q10,pred.Train))

##      pred.Train
##       Nao Sim
##   Nao 246  85
##   Sim 100 328

## Accuracy
(accuracy.Train<-sum(diag(confusion_mat_train))/sum(confusion_mat_train))
# 0.7562582

## [1] 0.7562582
```

```
# "Matriz de confusão"e accuracy sobre CJ.test #

probs.ctree.CJ.test<-predict(ctree.CJ.train,CJ.test,type="vector") # the
default type

## Confusion Matrix
pred.test <-apply(probs.ctree.CJ.test,1,which.max)

pred.test<-factor(pred.test, levels = c(1,2), labels = c("Nao", "Sim"))

(confusion_mat_test<-table(CJ.test$q10,pred.test))

##      pred.test
##       Nao Sim
##   Nao 158  67
##   Sim  78 203

## Accuracy
(accuracy.test<-sum(diag(confusion_mat_test))/sum(confusion_mat_test)) #
0.7134387

## [1] 0.7134387
```

3.4) [1 valor] Completação das frases seguintes:

A árvore obtida, classifica as observações do nó folha 73) na classe **Não**; o nó folha com o maior número de observações de treino é o nó **14**; no conjunto de teste o número de observações corretamente classificadas nas classes "Não" e "Sim" é **158** e **203**. respetivamente.

```
# Analisar com os seguintes outputs

# Pergunta 1 e 2
ctree.CJ.train

## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 759 1040.000 Sim ( 0.43610 0.56390 )
##    2) q13a < 0.5 450  592.500 Nao ( 0.63111 0.36889 )
##      4) q14i < 2.5 246  279.900 Nao ( 0.74390 0.25610 )
##        8) q12b_a < 0.5 133  122.500 Nao ( 0.82707 0.17293 ) *
##        9) q12b_a > 0.5 113  146.900 Nao ( 0.64602 0.35398 )
##         18) q14e < 4.5 108  137.500 Nao ( 0.66667 0.33333 )
##           36) q14e < 1.5 10   13.860 Nao ( 0.50000 0.50000 )
##             72) q14i < 1.5 8   10.590 Sim ( 0.37500 0.62500 ) *
##             73) q14i > 1.5 2    0.000 Nao ( 1.00000 0.00000 ) *
##           37) q14e > 1.5 98  122.300 Nao ( 0.68367 0.31633 )
##             74) q14i < 1.5 20   13.000 Nao ( 0.90000 0.10000 ) *
##             75) q14i > 1.5 78  102.900 Nao ( 0.62821 0.37179 ) *
##         19) q14e > 4.5 5    5.004 Sim ( 0.20000 0.80000 ) *
```

```
##      5) q14i > 2.5 204  282.800 Sim ( 0.49510 0.50490 )
##       10) q14e < 2.5 41   52.640 Nao ( 0.65854 0.34146 ) *
##       11) q14e > 2.5 163  224.600 Sim ( 0.45399 0.54601 )
##         22) q12b_a < 0.5 72   98.920 Nao ( 0.55556 0.44444 )
##           44) q14i < 4.5 69   95.290 Nao ( 0.53623 0.46377 ) *
##           45) q14i > 4.5 3    0.000 Nao ( 1.00000 0.00000 ) *
##         23) q12b_a > 0.5 91  120.300 Sim ( 0.37363 0.62637 )
##           46) q14i < 3.5 68   92.790 Sim ( 0.42647 0.57353 ) *
##           47) q14i > 3.5 23   24.080 Sim ( 0.21739 0.78261 ) *
##    3) q13a > 0.5 309  263.500 Sim ( 0.15210 0.84790 )
##      6) q12b_a < 0.5 117  128.800 Sim ( 0.23932 0.76068 )
##       12) q14i < 4.5 112  126.000 Sim ( 0.25000 0.75000 ) *
##       13) q14i > 4.5 5    0.000 Sim ( 0.00000 1.00000 ) *
##      7) q12b_a > 0.5 192  124.000 Sim ( 0.09896 0.90104 )
##       14) q14e < 4.5 171  119.300 Sim ( 0.11111 0.88889 ) *
##       15) q14e > 4.5 21    0.000 Sim ( 0.00000 1.00000 ) *
```

Pergunta 1 -» Observando o nó 73, é possivel verificar que este é um nó e coloca na classe não
Pergunta 2 -» Contando todas as observações dos nós folhas, o nó 14 possui 171 obs.

#Pergunta 3 e 4

```
confusion_mat_test
```

```
##      pred.test
##       Nao Sim
##   Nao 158  67
##   Sim  78 203
```

Pergunta 3 -» Contar os True Positive
Pergunta 4 -» Contar os True Negative