Tipologia Pac 2

Alexandre Fló Cuesta i Arnau Gironella

12/01/2022

1. Descripció del dataset.

Perquè és important i quina pregunta/problema pretén respondre?

Hem decidit agafar el conjunt de dades de mostra proposat per el professorat ja que és prou interessant i conté tant variables categòriques com numèriques.

La nostra pregunta i/o problema és esbrinar si conjunt de dades serveix per predir de forma efectiva un atac de cor, i en cas que sigui efectiu, volem comprovar si la freqüència cardíaca màxima que pot arribar una persona està relacionada amb el risc coronari.

2. Integració i selecció de les dades d'interès a analitzar.

En aquest cas hem decidit mantenir totes les variables fins que poguem analitzar quines són relament significatives per resoldre el problema:

age: L'edat de la persona en anys

sex: El sexe de la persona (1 = home, 0 = dona)

cp: tipus de dolor toràcic

- Valor 0: asimptomàtic
- Valor 1: angina atípica
- Valor 2: dolor no anginós
- Valor 3: angina típica

trtbps: La pressió arterial en repòs de la persona (mm Hg a l'ingrés a l'hospital)

chol: Mesura del colesterol de la persona en mg/dl

fbs: Glucèmia en dejú de la persona (> 120 mg/dl, 1 = veritable; 0 = fals)

restecg: resultats de l'electrocardiograma en repòs:

- Valor 0: hipertròfia ventricular esquerra probable o definida segons els criteris de Estes
- Valor 1: normal
- Valor 2: anomalia de l'ona ST-T (inversió de l'ona T i/o elevació o depressió del ST > 0,05 mV)

thalachh: La freqüència cardíaca màxima aconseguida per la persona

exng: Angina induïda per l'exercici (1 = sí; 0 = no)

oldpeak: Depressió del ST induïda per l'exercici en relació amb el repòs

slp: el pendent del segment ST màxim de l'exercici

- 0: descendent
- 1: pla
- 2: ascendent

ca: El nombre de vasos principals (0-3)

thall: Un trastorn sanguini anomenat talasemia

- Valor 1: defecte fix (absència de flux sanguini en alguna part del cor)
- Valor 2: flux sanguini normal
- Valor 3: defecte reversible (s'observa un flux sanguini però no és normal)

output: cardiopatia (1 = no, 0= sí)

3. Neteja de les dades

3.1. Les dades contenen zeros o elements buits? Gestiona cadascun d'aquests casos.

Primerament carreguem les dades, en aquest cas és "heart.csv" i comprovem si hi ha alguna columna que tingui valors nuls o valors buits.

```
df <- read.csv("/Users/albert/Desktop/heart.csv")
df2 <- read.csv("/Users/albert/Desktop/heart.csv")

str(df)
```

```
## 'data.frame':    303 obs. of  14 variables:
##  $ age     : int  63 37 41 56 57 57 56 44 52 57 ...
##  $ sex     : int  1 1 0 1 0 1 0 1 1 1 ...
##  $ cp      : int  3 2 1 1 0 0 1 1 2 2 ...
##  $ trtbps  : int  145 130 130 120 120 140 140 120 172 150 ...
##  $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
##  $ fbs     : int  1 0 0 0 0 0 0 0 1 0 ...
##  $ restecg : int  0 1 0 1 1 1 0 1 1 1 ...
##  $ thalachh: int  150 187 172 178 163 148 153 173 162 174 ...
##  $ exng    : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
##  $ slp     : int  0 0 2 2 2 1 1 2 2 2 ...
##  $ caa     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ thall   : int  1 2 2 2 2 1 2 3 3 2 ...
##  $ output  : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
colSums(is.na(df))
```

```
##      age      sex       cp   trtbps     chol      fbs  restecg thalachh
##        0        0        0        0        0        0        0        0
##    exng  oldpeak      slp      caa    thall   output
##        0        0        0        0        0        0
```

```
colSums(df == "")
```

```
##      age      sex       cp   trtbps     chol      fbs  restecg thalachh
##        0        0        0        0        0        0        0        0
##    exng  oldpeak      slp      caa    thall   output
##        0        0        0        0        0        0
```

Posteriorment transformem algunes de les dades categòriques a factor.

```
df$sex <- factor(df$sex)
df$exng <- factor(df$exng)
df$cp <- factor(df$cp)
df$fbs <- factor(df$fbs)
df$restecg <- factor(df$restecg)
df$output <- factor(df$output)
levels(df$output)=c("Yes","No")

str(df)
```

```
## 'data.frame':    303 obs. of  14 variables:
##  $ age     : int  63 37 41 56 57 57 56 44 52 57 ...
##  $ sex     : Factor w/ 2 levels "0","1": 2 2 1 2 1 2 1 2 2 2 ...
##  $ cp      : Factor w/ 4 levels "0","1","2","3": 4 3 2 2 1 1 2 2 3 3 ...
##  $ trtbps  : int  145 130 130 120 120 140 140 120 172 150 ...
##  $ chol    : int  233 250 204 236 354 192 294 263 199 168 ...
##  $ fbs     : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 2 1 ...
##  $ restecg : Factor w/ 3 levels "0","1","2": 1 2 1 2 2 2 1 2 2 2 ...
##  $ thalachh: int  150 187 172 178 163 148 153 173 162 174 ...
##  $ exng    : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 1 1 ...
##  $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
##  $ slp     : int  0 0 2 2 2 1 1 2 2 2 ...
##  $ caa     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ thall   : int  1 2 2 2 2 1 2 3 3 2 ...
##  $ output  : Factor w/ 2 levels "Yes","No": 2 2 2 2 2 2 2 2 2 2 ...
```

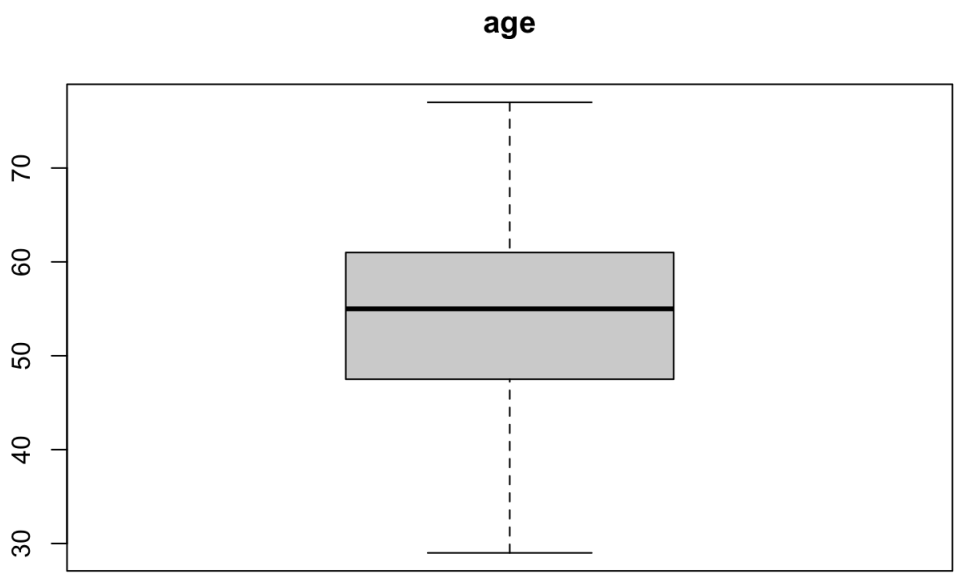
3.2. Identifica i gestiona els valors extrems.

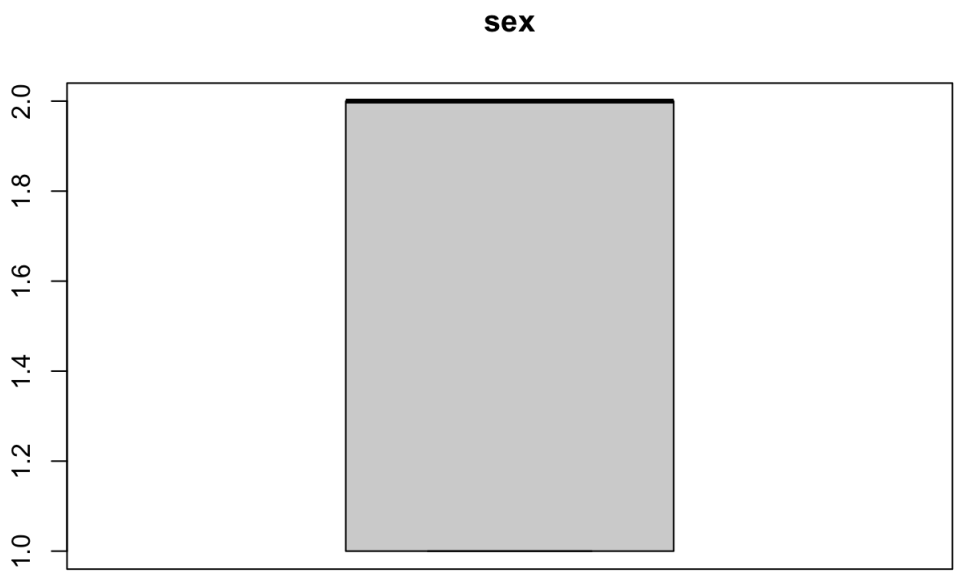
Amb la funció de boxplot de R podem veure fàcilment quins son els valors atípics de cadascuna de les variables.

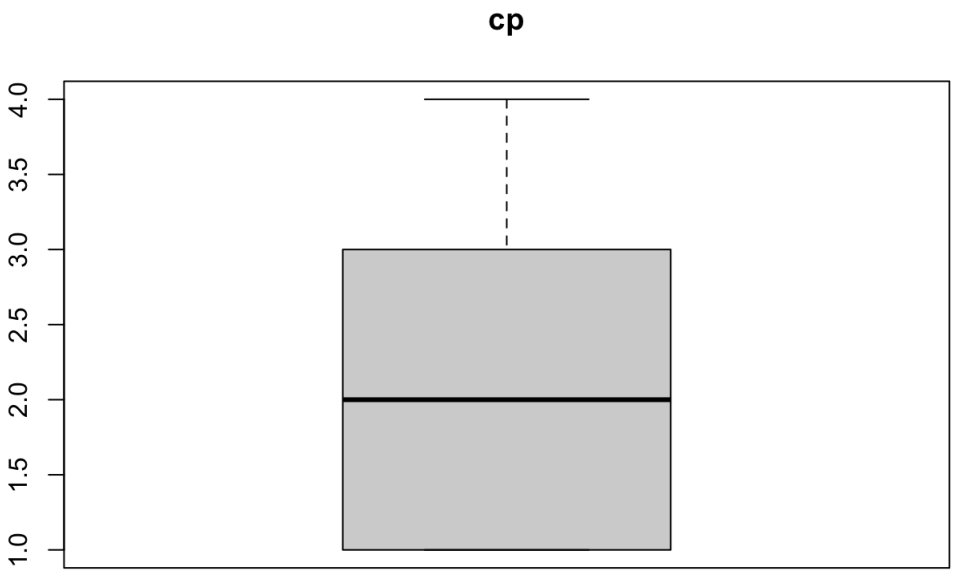
```
# crear lista con nombres de las variables
var_names <- names(df)[1:14]

var_names
```

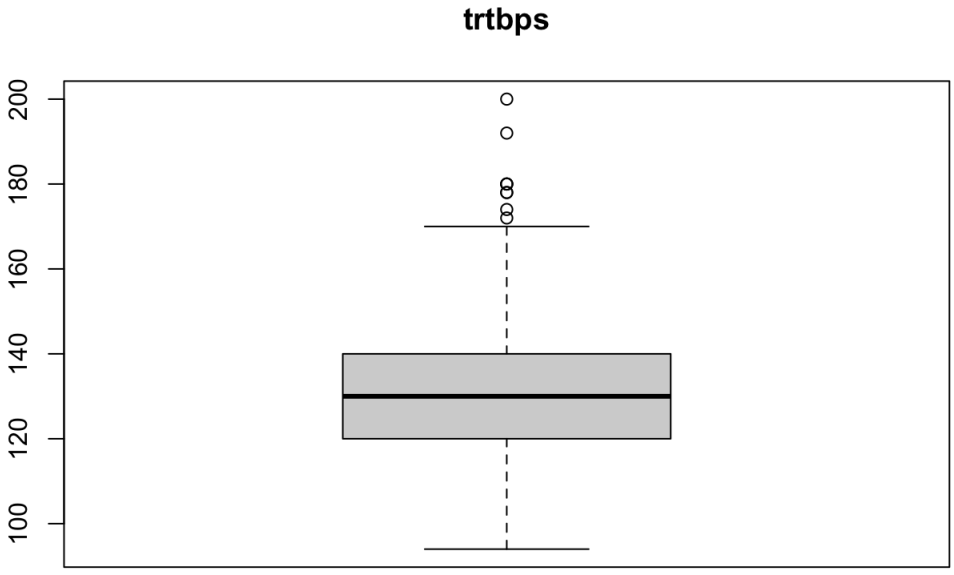
```
##  [1] "age"      "sex"      "cp"       "trtbps"   "chol"     "fbs"
##  [7] "restecg"  "thalachh" "exng"     "oldpeak"  "slp"      "caa"
## [13] "thall"    "output"
```

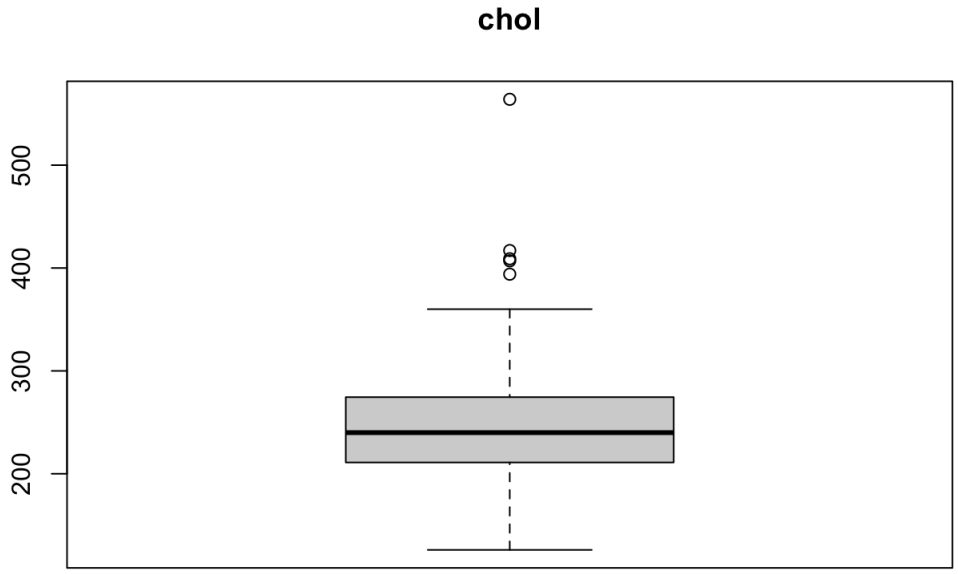
```
# loop para graficar boxplots de cada variable
for (i in var_names) {
  boxplot((df[[i]]), main = i)
}
```

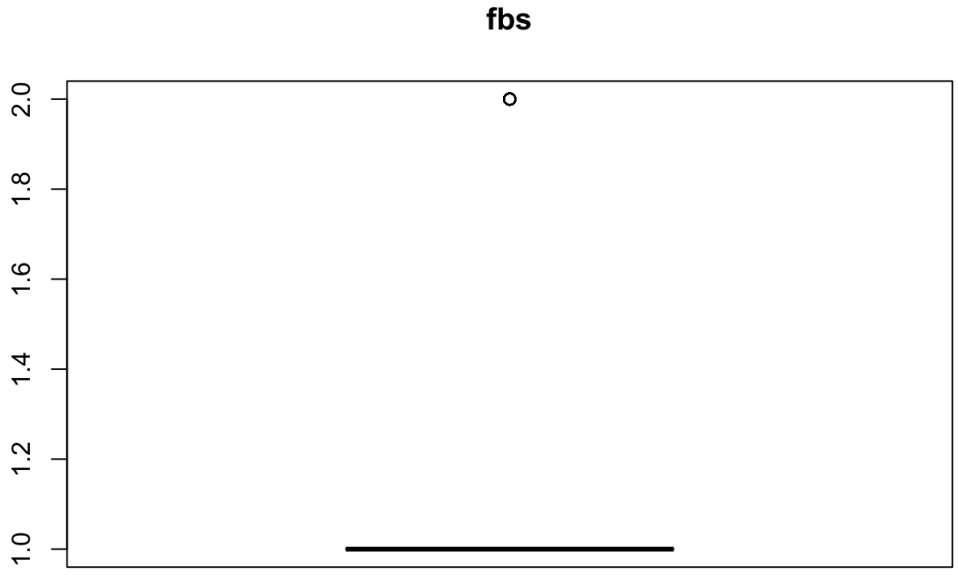


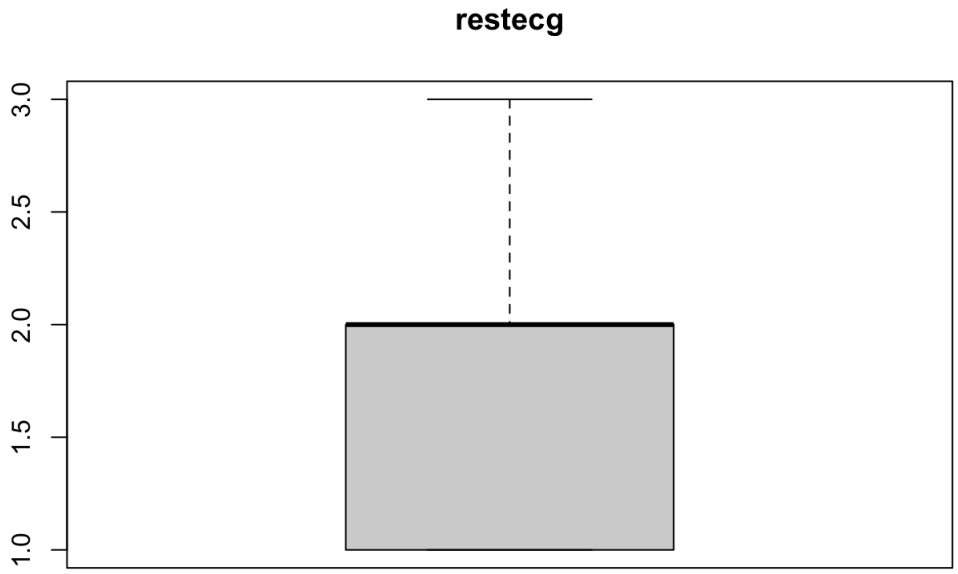


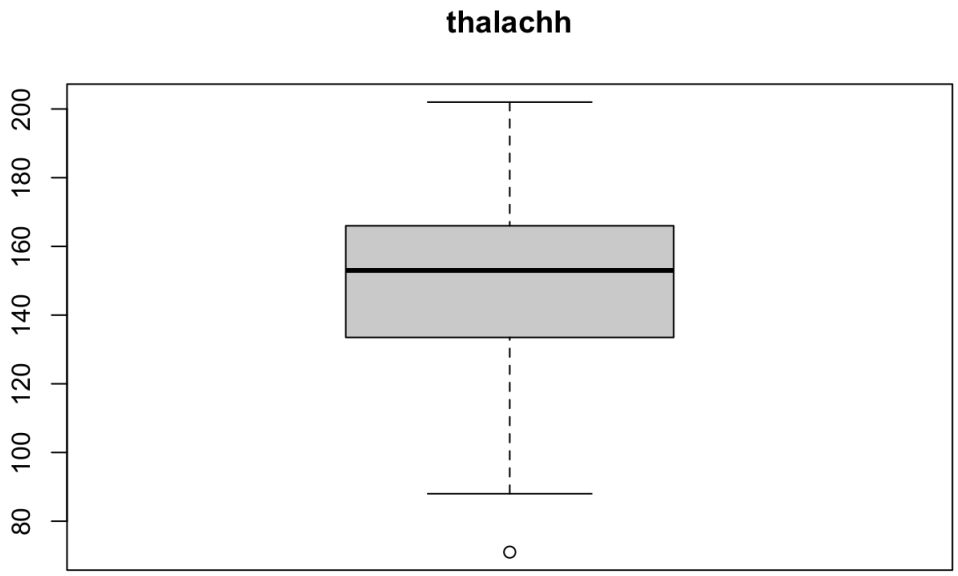

cp
1.0
1.5
2.0
2.5
3.0
3.5
4.0

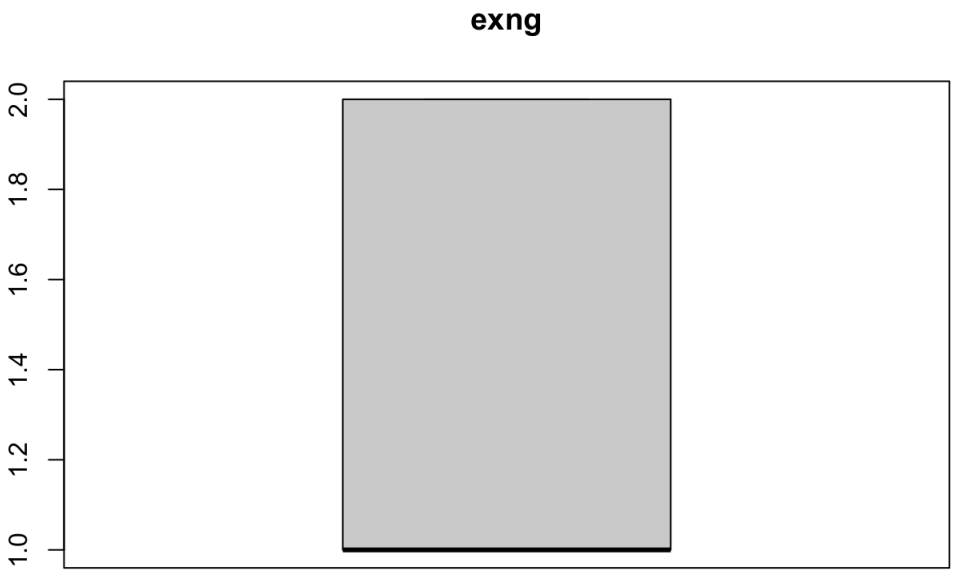

trtbps
100
120
140
160
180
200

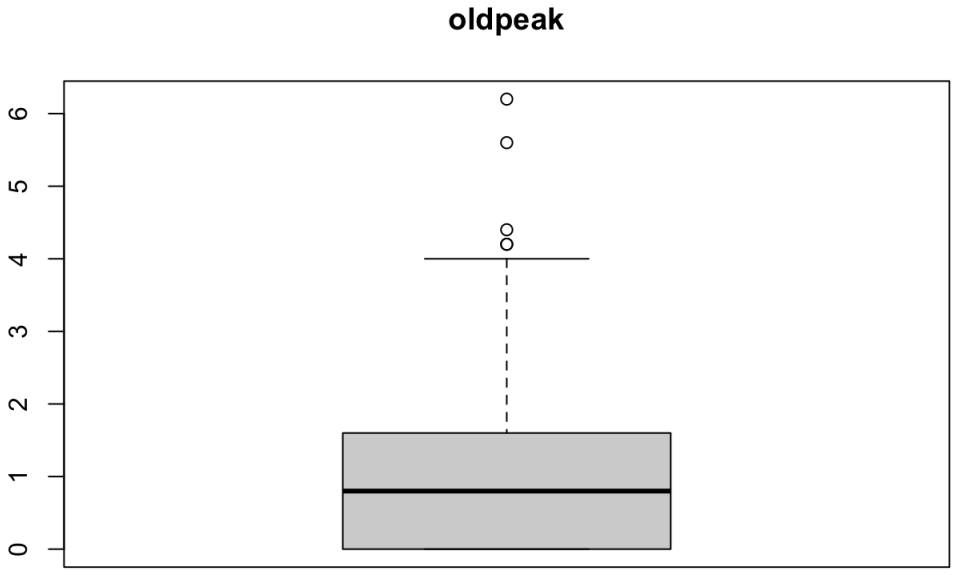

chol
500
400
300
200

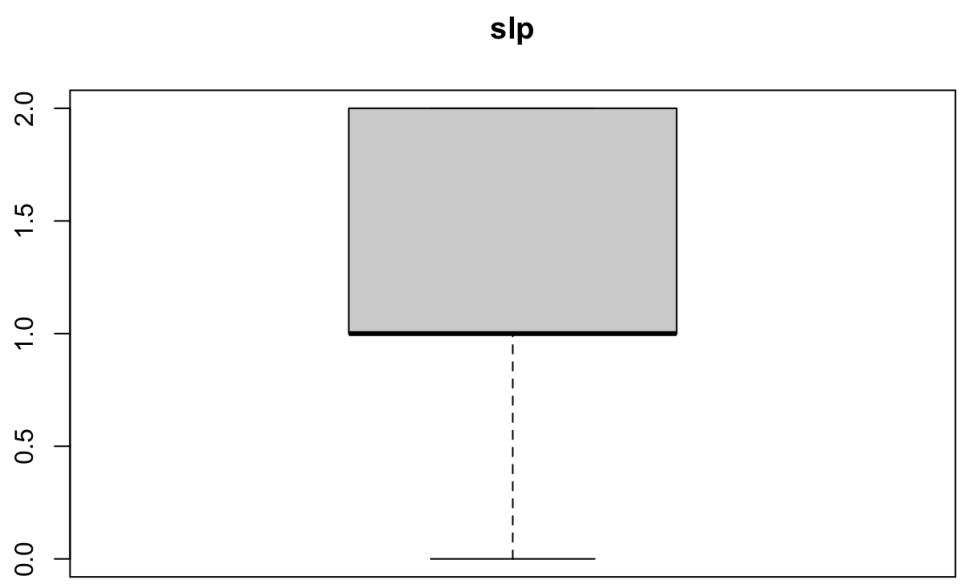

fbs
2.0
1.8
1.6
1.4
1.2
1.0

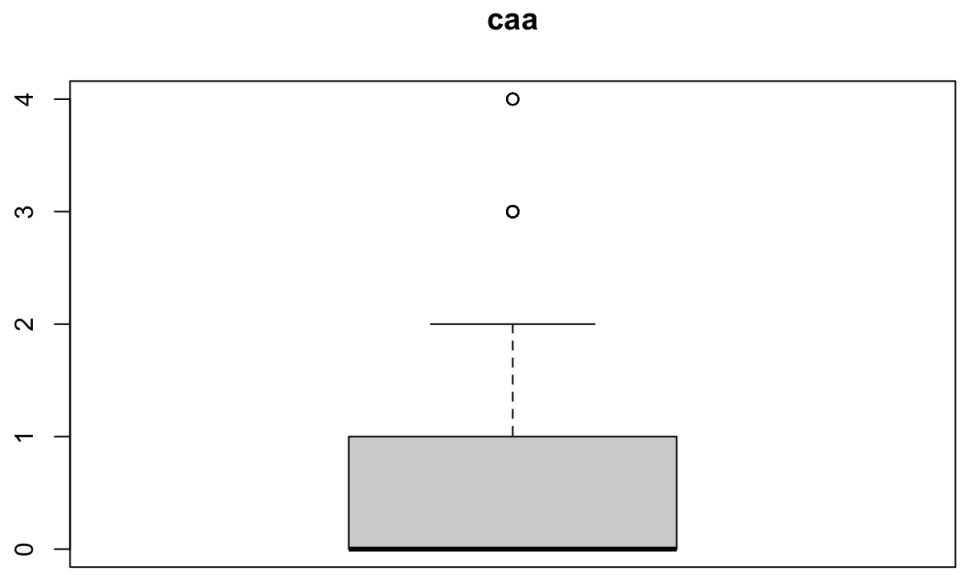

restecg
3.0
2.5
2.0
1.5
1.0

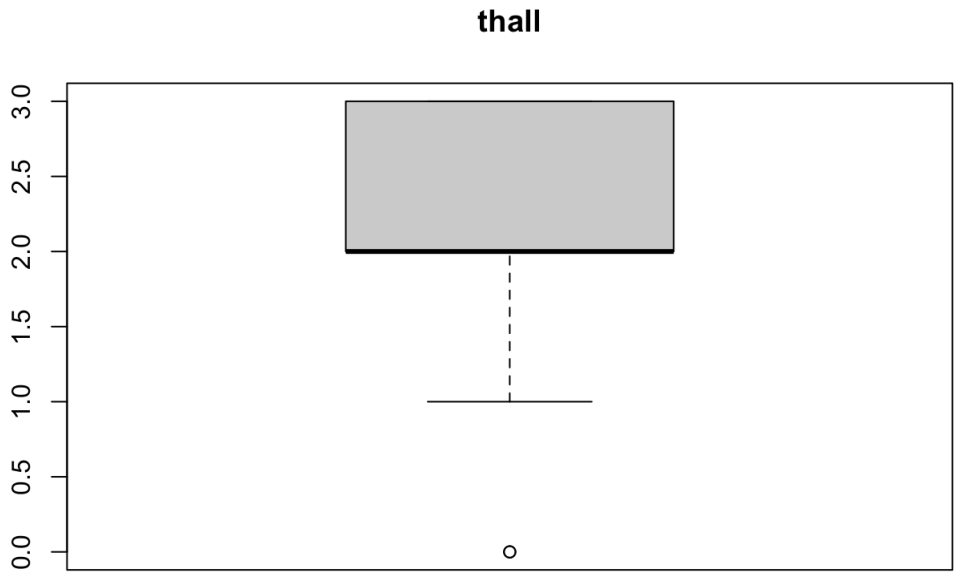

thalachh
200
180
160
140
120
100
80

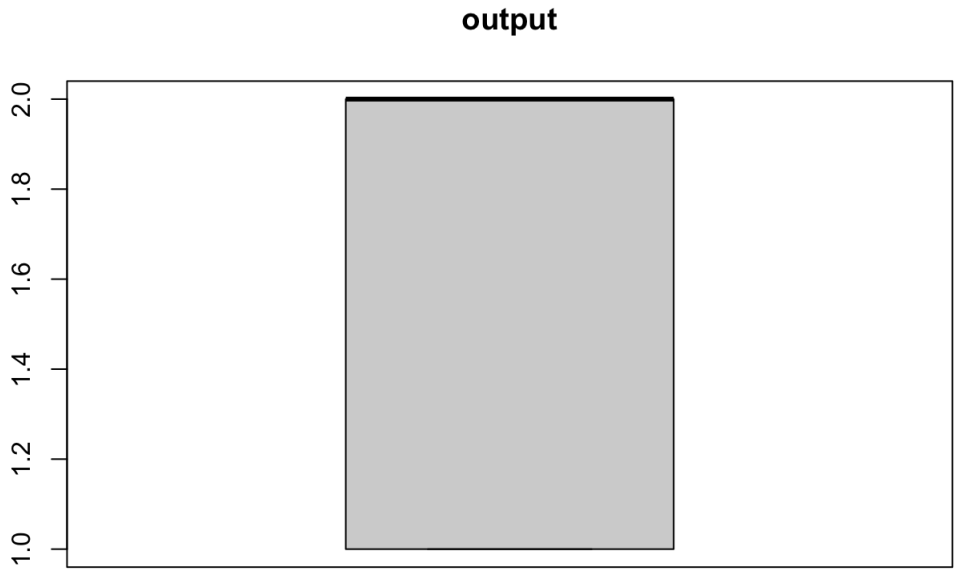

exng
2.0
1.8
1.6
1.4
1.2
1.0


oldpeak
6
5
4
3
2
1
0


slp
0.0
0.5
1.0
1.5
2.0


caa
0
1
2
3
4





Observem que en la gran majoria no tenen valors atípics, i en les variables on sí que en tenen, creiem que són necessàries per l'estudi. Tot i que són valors extrems no són anòmals, per tant, no provenen d'un error sinó que són necessaris per l'anàlisi posterior.

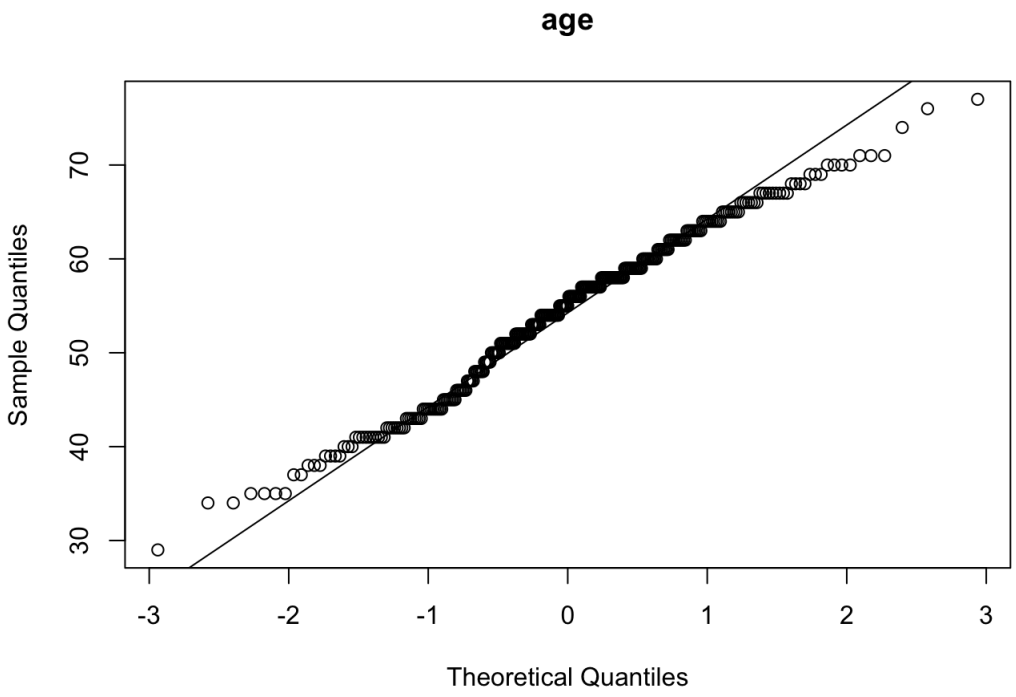
4. Anàlisi de les dades.

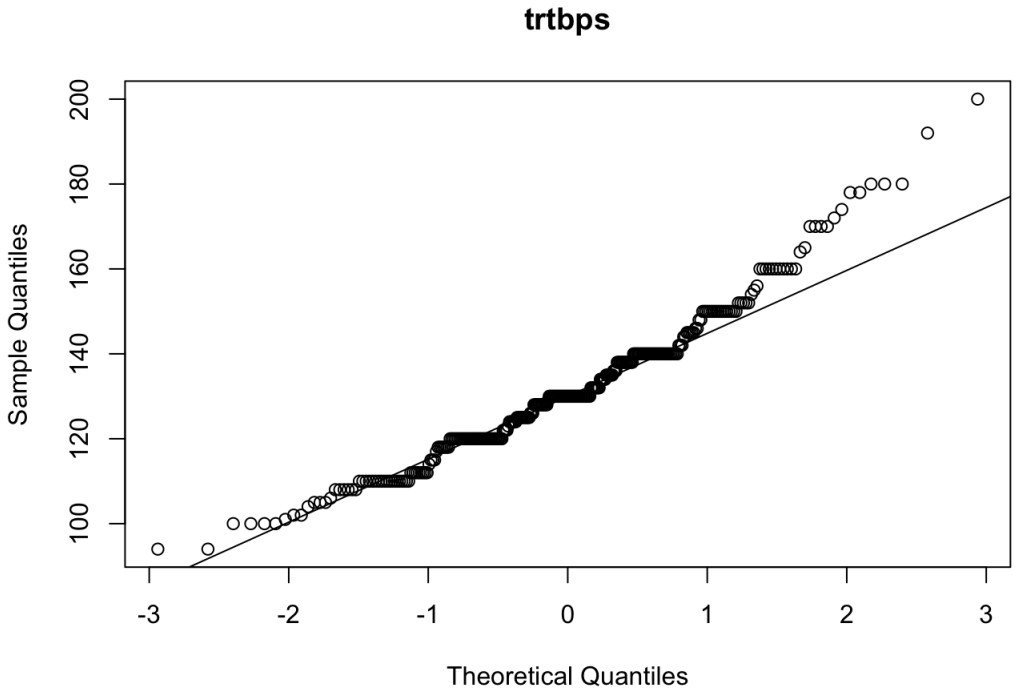
4.1. Selecció dels grups de dades que es volen analitzar/comparar (p. e., si es volen comparar grups de dades, quins són aquests grups i quins tipus d'anàlisi s'aplicaran?).

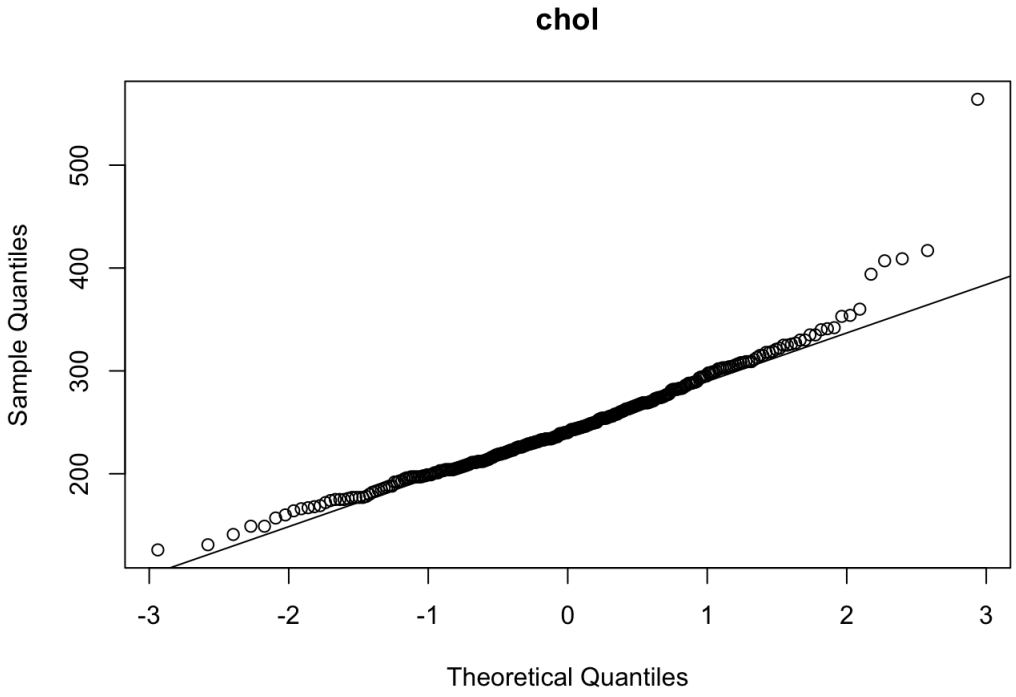
En aquest cas realitzarem 3 tipus d'anàlisis: Un contrast d'hipòtesis per a la variable `thalachh`, una correlació entre les variables per entendre quant forta és la relació entre les variables independents i el `output` i, finalment una classificació per determinar si amb aquest conjunt de dades es poden treure prediccions prou bones i en cas que sigui així .

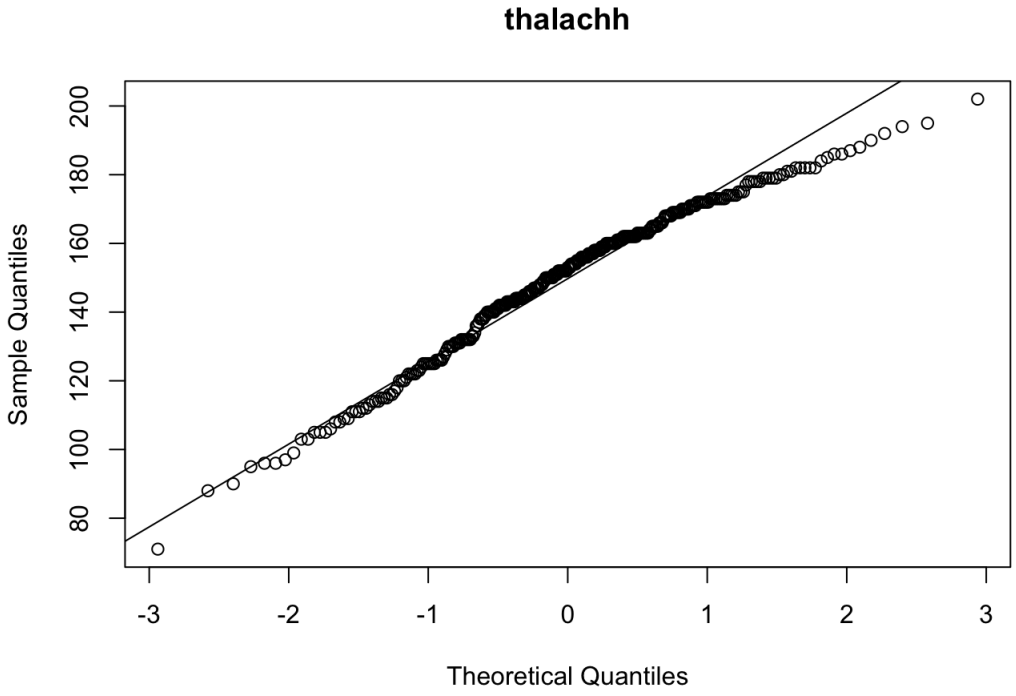
4.2. Comprovació de la normalitat i homogeneïtat de la variància.

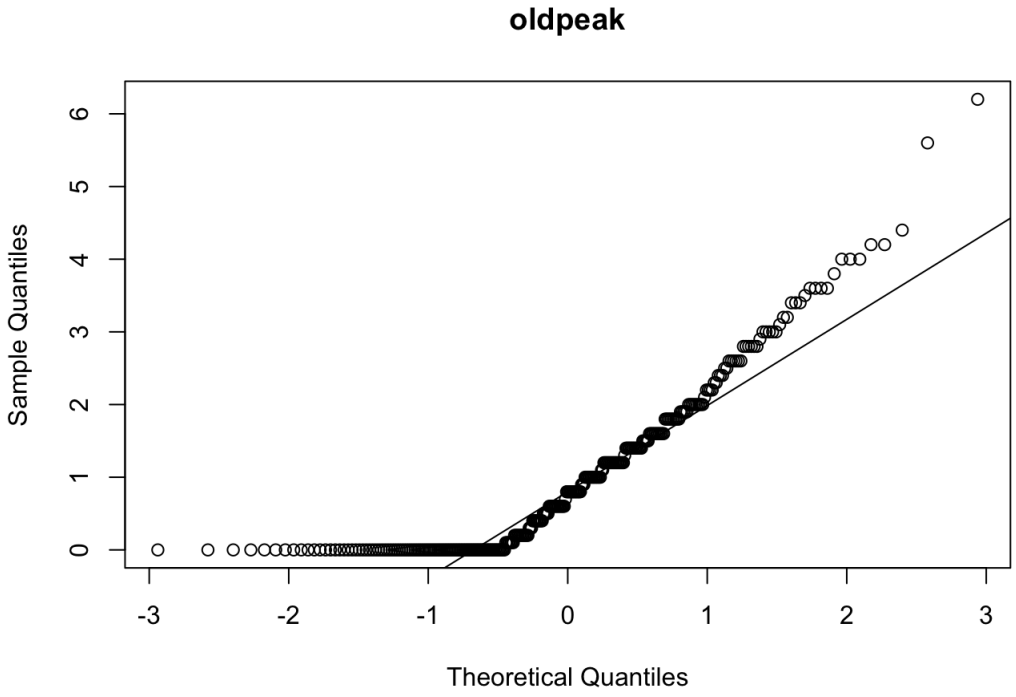
```
var_names <- names(df)[1:14]
var_num <- var_names[sapply(df[var_names], is.numeric)]
for (i in var_num) {
  qqnorm(na.omit(df[[i]]), main = i)
  qqline(na.omit(df[[i]]))

}
```

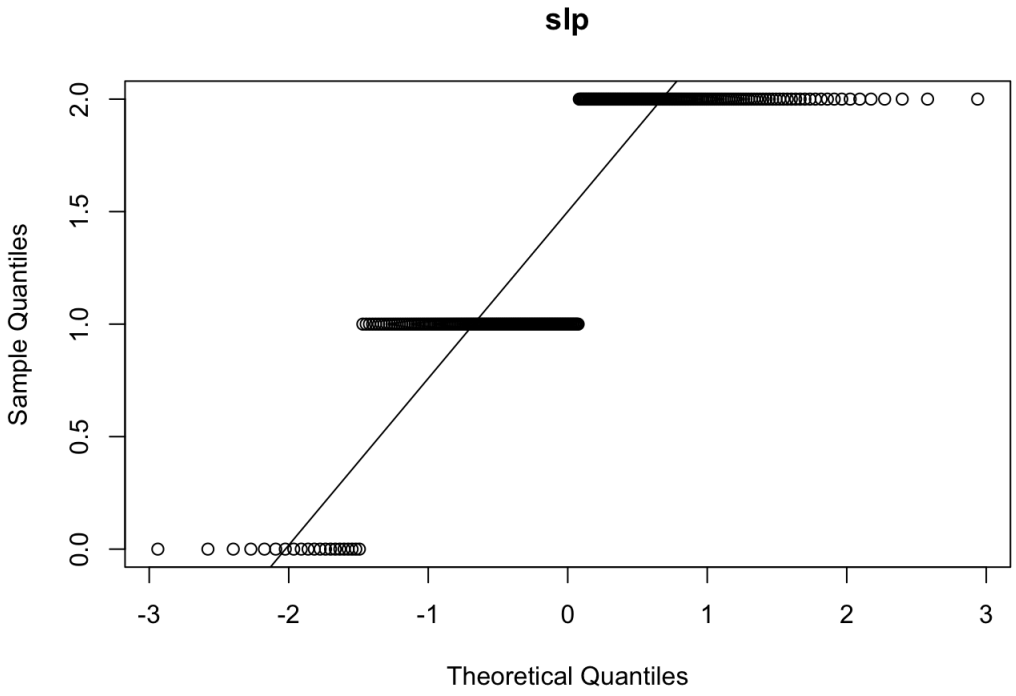

age
Sample Quantiles
30
40
50
60
70
-3
-2
-1
0
1
2
3
Theoretical Quantiles

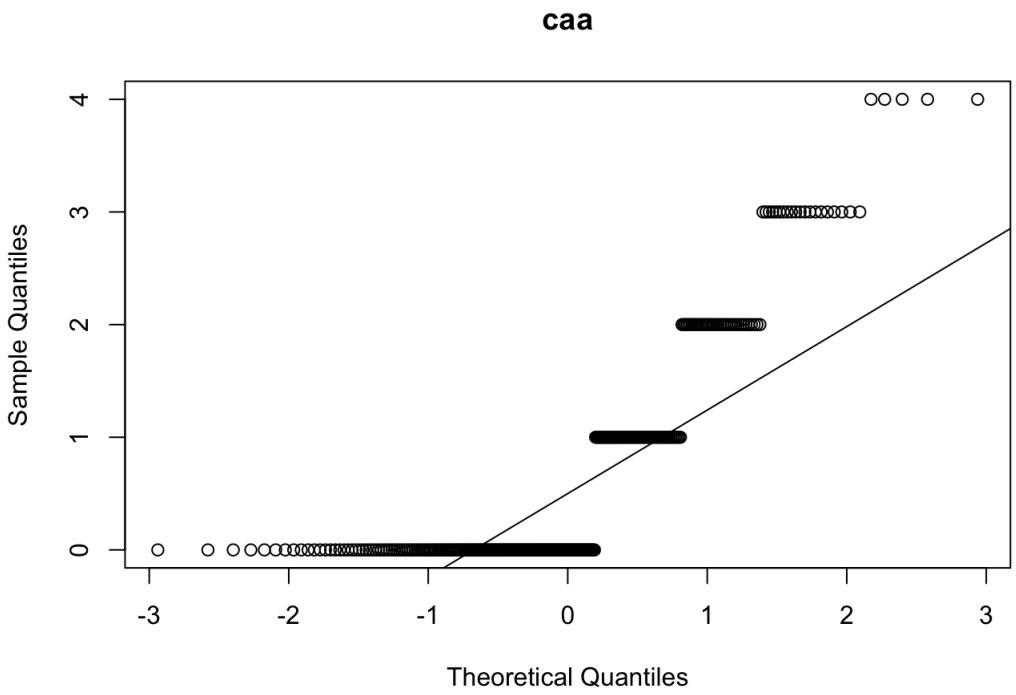

trtbps
Sample Quantiles
100
120
140
160
180
200
-3
-2
-1
0
1
2
3
Theoretical Quantiles

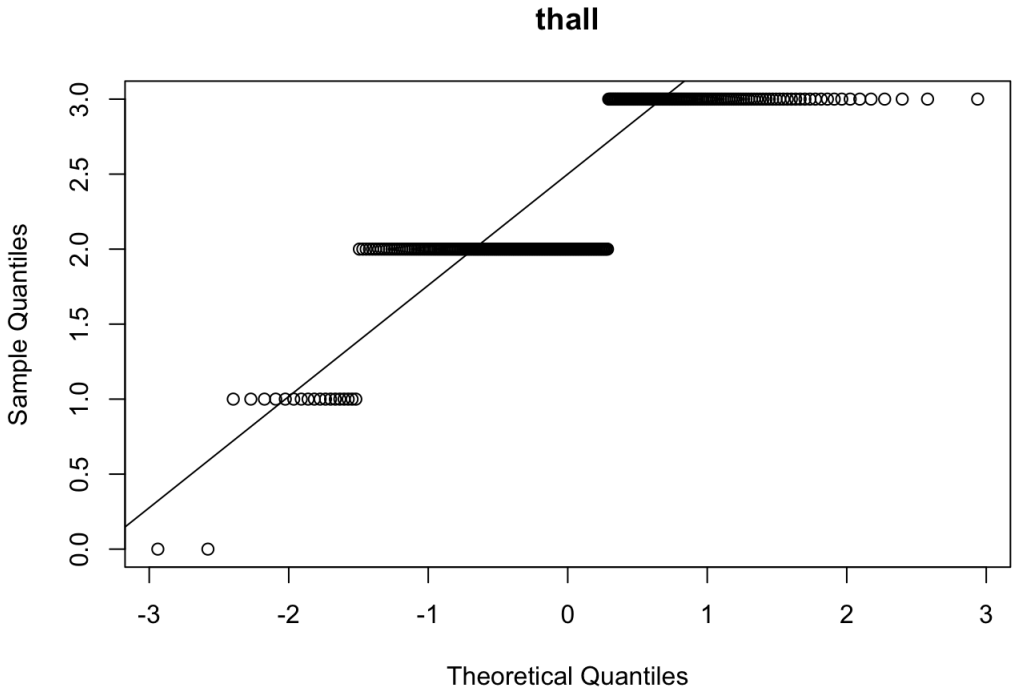

chol
Sample Quantiles
200
300
400
500
-3
-2
-1
0
1
2
3
Theoretical Quantiles


thalachh
Sample Quantiles
80
100
120
140
160
180
200
-3
-2
-1
0
1
2
3
Theoretical Quantiles


oldpeak
Sample Quantiles
Theoretical Quantiles
0
1
2
3
4
5
6
-3
-2
-1
0
1
2
3


slp
Sample Quantiles
Theoretical Quantiles
0.0
0.5
1.0
1.5
2.0
-3
-2
-1
0
1
2
3





Veiem que totes les gràfiques segueixen una distribució normal.

Per a la comprovar la homogeneïtat de la variança:

```
for (i in var_names) {
  if(is.numeric(df[[i]])){
    levene_test <- var.test(df[[i]] ~ df$output)
    print(paste(i, ": p-value = ", levene_test$p.value))
  }
}
```

```
## [1] "age : p-value =  0.0280731363393591"
## [1] "trtbps : p-value =  0.0714572799576749"
## [1] "chol : p-value =  0.33533240236138"
## [1] "thalachh : p-value =  0.0439333150911281"
## [1] "oldpeak : p-value =  5.9760818515997e-10"
## [1] "slp : p-value =  0.49875624806208"
## [1] "caa : p-value =  0.0114463984756439"
## [1] "thall : p-value =  2.5668326681938e-06"
```

Observant els valors de p-value, totes les variables que tenen un p-value superior a 0.05 tenen variàncies similars a la nostre variable d'estudi `output` .

4.3 Aplicació de proves estadístiques per comparar els grups de dades.

4.3.1 Test d'hipòtesis

Hipòtesi nul·la (H0): No hi ha diferència significativa en la mitjana de "thalachh" entre els pacients que tenen una major probabilitat de patir un atac de cor (target = 1) i els pacients que tenen una menor probabilitat (target = 0).

Hipòtesi alternativa (Ha): Hi ha una diferència significativa en la mitjana de "thalach" entre els pacients que tenen una major probabilitat de patir un atac de cor i els pacients que tenen una menor probabilitat.

Abans de fer una prova t per comparar les mitjanes de dos grups, és recomanable comprovar si les variàncies són iguals o diferents entre els dos grups. Això es coneix com la suposició d'igualtat de variàncies i és necessària per aplicar una prova t estandarditzada.

Com que el valor p obtingut a la prova d'abans, és menor que el nivell de significància (0.05), es rebutja la hipòtesi nul·la d'igualtat de variàncies i s'ha d'utilitzar una prova t no paramètrica com la prova de Welch.

```
t.test(thalachh ~ output, data=df, var.equal=FALSE)
```

```
##
##  Welch Two Sample t-test
##
## data:  thalachh by output
## t = -7.953, df = 269.9, p-value = 5.019e-14
## alternative hypothesis: true difference in means between group Yes and group No is not equal to 0
## 95 percent confidence interval:
##  -24.15912 -14.57132
## sample estimates:
## mean in group Yes  mean in group No
##          139.1014          158.4667
```

La sortida de l'output de la prova t de Welch indica el següent:

- El valor t obtingut és -7.953, cosa que indica una diferència significativa entre les mitjanes de "thalach" per als dos grups (target=0 i target=1).
- El valor p és molt petit (5.019e-14), cosa que indica que la probabilitat d'obtenir un valor t tan extrem o més extrem sota la hipòtesi nul·la (sense diferència significativa) és molt petita. Per tant, es rebutja la hipòtesi nul·la.
- Els valors mitjans de la variable "thalach" per als dos grups són: 139.1014 per a target=0 i 158.4667 per a target=1.

En resum, els resultats obtinguts en aquesta prova suggereixen que hi ha una diferència significativa en la mitjana de "thalach" entre els pacients que tenen una major probabilitat de patir un atac de cor (target = 1) i els pacients que tenen una menor probabilitat (target = 0).

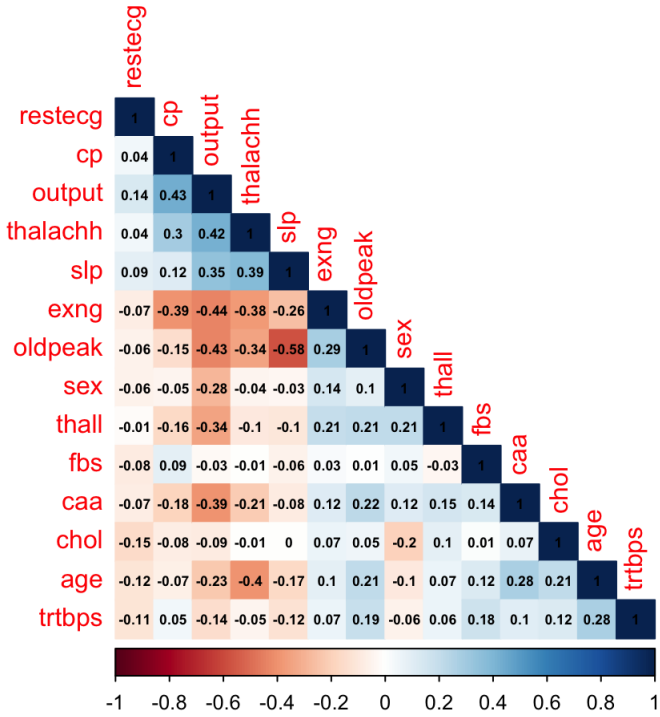
4.3.2 Correlació

```
if (!require('corrplot')) install.packages('corrplot'); library(corrplot)
```

```
## Loading required package: corrplot
```

```
## corrplot 0.92 loaded
```

```
cor <- cor(df2)

corrplot(cor,method = 'color',type = 'lower',
         order = 'hclust',cl.ratio = 0.2,
         addCoef.col = 'black',
         number.cex = 0.55)
```



Observant la matriu de correlacions

veiem que les 4 variables amb una correlació més forta amb `output` són: `exng` , `oldpeak` , `thalachh` i `caa` . En el cas de `exng` i `oldpeak` es tracta d'una correlació negativa, és a dir, quan els valors d'aquesta disminueixen el valor de `output` augmenta. I en el cas de les altres dues, passa quan més augmenta el valor més augmenta `output` .

4.3.3 Classificació

A continuació provarem un model de GradientBoosting amb la llibreria Caret per tal de fer una primera aproximació de la variable `output` per tal de veure si amb aquest conjunt de dades podem predir de forma eficaç el risc coronari, i en cas que així sigui, quines son les variables més importants per el model.

```
if (!require('caret')) install.packages('caret'); library(caret)
```

```
## Loading required package: caret
```

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

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

```
if (!require('gbm')) install.packages('gbm'); library(gbm)
```

```
## Loading required package: gbm
```

```
## Loaded gbm 2.1.8
```

```
set.seed(619)
indexes <- createDataPartition(df$output,p = 0.75,list = FALSE)
```

```
trainData <-  df[indexes,]
testData <-  df[-indexes,]
```

```
fitControl4 <-  trainControl(method = 'repeatedcv',
                             number=10,
                             repeats=1,
                             classProbs=TRUE,
                             summaryFunction = twoClassSummary)
```

```
gbmGrid4 <-  expand.grid(interaction.depth = c(1:9),
                         n.trees = (1:30)*10,
                         shrinkage = 0.1,
                         n.minobsinnode = c(10,20,30,40,50))
```

```
xgbe4 <-  train(output~.,
              data = df,
              method = 'gbm',
              verbose=FALSE,
              tuneGrid = gbmGrid4,
              trControl = fitControl4,
              metric='ROC')
```

```
preds4 <-  predict(xgbe4, testData)
levels(preds4)=c("Yes","No")
confusionMatrix(preds4, testData$output)
```

```
## Confusion Matrix and Statistics
##
##          Reference
## Prediction Yes No
##        Yes  28  2
##        No    6 39
##
##               Accuracy : 0.8933
##                 95% CI : (0.8006, 0.9528)
##    No Information Rate : 0.5467
##    P-Value [Acc > NIR] : 9.35e-11
##
##                  Kappa : 0.7826
##
##  Mcnemar's Test P-Value : 0.2888
##
##             Sensitivity : 0.8235
##             Specificity : 0.9512
##          Pos Pred Value : 0.9333
##          Neg Pred Value : 0.8667
##              Prevalence : 0.4533
##          Detection Rate : 0.3733
##    Detection Prevalence : 0.4000
##       Balanced Accuracy : 0.8874
##
##        'Positive' Class : Yes
##
```

Tal i com indica la matriu de confusió. El model té una alta capacitat de discernir entre els valors positius i negatius de la variable `output`. És especialment bo, a l'hora d'encertar el nombre de negatius amb un valor per sobre de 95. Segurament la capacitat de ditingir entre veritables i falsos positius no sigui gaire bona tenint en compte el context del problema. En aquest cas seria distingir entre persones amb risc coronari o no.

```
xgbe4$results[best(xgbe4$results, metric='ROC', maximize=TRUE),]
```

```
##     shrinkage interaction.depth n.minobsinnode n.trees       ROC      Sens
## 252       0.1                 2             40     120 0.9151665 0.7879121
##          Spec      ROCSD    SensSD    SpecSD
## 252 0.8533088 0.05520859 0.1530198 0.1203535
```

Segons el model entrenat ambc aret. En la millor parametrització del amteix s'arriba a un roc de 0.91. Mentre que en el següent gràfic realitzat amb la llibreria pROC el ROC és de 0.88. Un ROC especialment alt.

```
if (!require('pROC')) install.packages('pROC'); library(pROC)
```

```
## Loading required package: pROC
```

```
## Type 'citation("pROC")' for a citation.
```

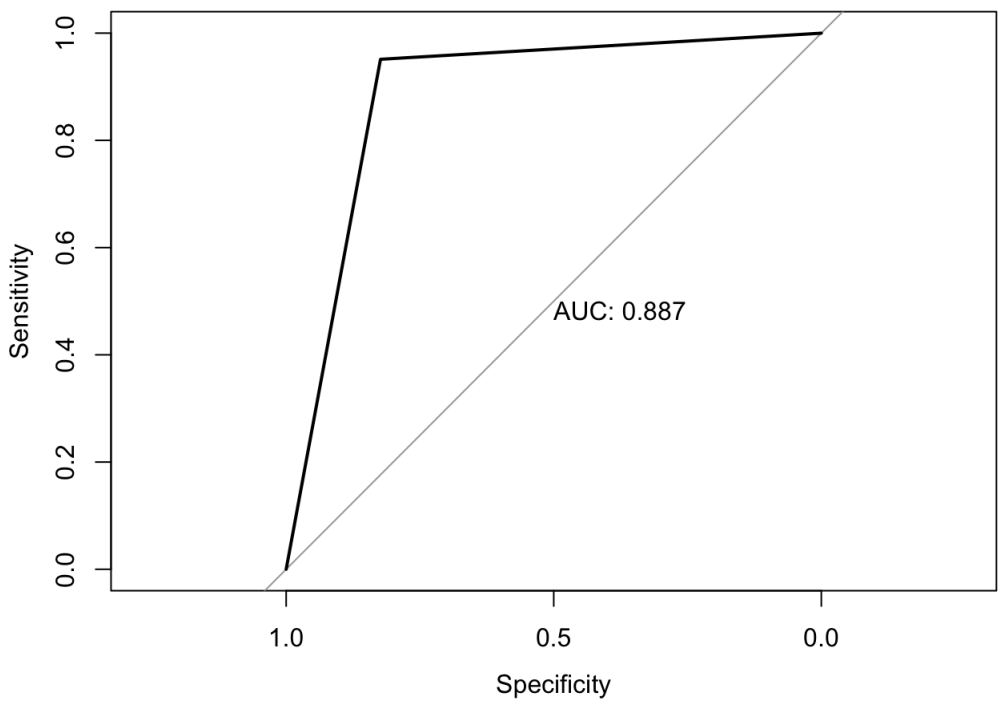
```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
roc <- roc(testData$output,as.numeric(preds4),plot=TRUE,print.auc=TRUE)
```

```
## Setting levels: control = Yes, case = No
```

```
## Setting direction: controls < cases
```




```
featureImportance <-  varImp(xgbe4, scale=FALSE)
featureImportance
```

```
## gbm variable importance
##
##          Overall
## thall     39.117
## caa       37.083
## exng1     21.087
## thalachh  20.680
## oldpeak   19.328
## age       15.533
## slp       11.495
## chol      10.966
## cp2        8.673
## sex1       7.714
## trtbps     7.028
## restecg1   2.752
## fbs1       0.000
## restecg2   0.000
## cp1        0.000
## cp3        0.000
```

Finalment observem la importància de les variables que ha utilitzat el model. En aquest cas la variable `thalachh` és la quarta en ordre d'importància, per tant, si bé té una importància signifcativa, hi ha altres variables com `thall` -existència o no d'un trastorn sanguini- o `caa` -nombre de vasos principals, que tenen més importància que la freqüència cardíaca màxima d'una persona.

Investigació prèvia : Alexandre,Arnau
Redacció de les respostes : Alexandre,Arnau
Desenvolupament del codi: Alexandre,Arnau
Participació al video: Alexandre,Arnau

Link github: https://github.com/PandaPandula/PRACTICA2TCVD Link video: https://drive.google.com/file/d/1__YF6e_OVsVKE2Zr6Ux8QHB8MsayjhvQ/view?usp=share_link