

Practica 2

Marc González Planes i Maria Sunyer Rigau

- LLiberies necessaries
- Descripció del Dataset
- Integració i selecció de dades
- Neteja de dades
 - KNN per suplir valors
- Identifica i gestiona valors extrems
 - Selecció de dades
 - Comprovació de la normalitat i homogeneïtat de la variància:
 - Comprovació de la normalitat
 - Comprovació Homoscedasticitat
- Proves estadístiques:
 - Contrast d'hipotesis
 - Models
 - Preperació de les dades
 - Naive Bayes
 - KNN
- Extracció Dataset
- Conclusió:
- Contribucions

LLiberies necessaries

```
if (!require('VIM')) install.packages('VIM');library('VIM')
if (!require('dplyr')) install.packages('dplyr');library('dplyr')
if (!require('missForest')) install.packages('missForest');library('missForest')
if (!require('car')) install.packages('car');library('car')
if (!require('psych')) install.packages('psych');library('psych')
if (!require('ggplot2')) install.packages('ggplot2');library('ggplot2')
if (!require('caTools')) install.packages('caTools');library('caTools')
if (!require('e1071')) install.packages('e1071');library('e1071')
if (!require('caret')) install.packages('caret');library('caret')
```

Descripció del Dataset

El dataset utilitzat intenta respondre la pregunta de quins factors influeixen en la salut cardiovascular. Aquest consisteix en les dades mèdiques d'un conjunt de pacients. El dataset utilitzat consisteix en diferents parametres mèdics els quals s'utilitzen per determinar riscos cardiovasculars. El nostre anàlisis intenta donar resposta a dues preguntes: - Quins són els parametres més i menys rellevants? - Podem predir el risc d'un pacient amb les dades actuals?

Age : Age of the patient

Sex : Sex of the patient

exng: exercise induced angina (1 = yes; 0 = no)

caa: number of major vessels (0-3)

cp : Chest Pain type

Value 1: typical angina

Value 2: atypical angina

Value 3: non-anginal pain

Value 4: asymptomatic

trtbps : resting blood pressure (in mm Hg)

chol : cholestoral in mg/dl fetched via BMI sensor

fbs : (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)

rest_ecg : resting electrocardiographic results

- Value 0: normal
- Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)
- Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria

thalach : maximum heart rate achieved

target : 0 = less chance of heart attack 1 = more chance of heart attack

Integració i selecció de dades

```
data_heart <- read.csv("./heart.csv")

num_fil <- dim(data_heart)[1]
num_col <- dim(data_heart)[2]

print("El data set te una mida:")
```

```
## [1] "El data set te una mida:"
```

```
sprintf("Files: %d", num_fil)
```

```
## [1] "Files: 303"
```

```
sprintf("Columnes: %d", num_col)
```

```
## [1] "Columnes: 14"
```

Neteja de dades

Les dades contenen zeros o elements buits:

```
# Calculem si tenim dades nulles
colSums(is.na(data_heart))
```

```
##      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
```

```
print("No hi ha valors nulls")
```

```
## [1] "No hi ha valors nulls"
```

En cas de rebre noves dades, aquestes podrien contenir valors nulls o elements buits. Per tant, procedim a “inventar” valors nulls per tal de fer un tractament de dades que en un futur ens permeti acceptar i tractar dades no completes.

KNN per suplir valors

Utilitzarem el metode KNN per suplir aquest valors i no “descartar” les dades.

```
# Enllaç de referencia: https://rpubs.com/harshaash/KNN_imputation
# Enllaç dos: https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputin
# g-missing-values/

# Introduim valors aleatoris: https://search.r-project.org/CRAN/refmans/missForest/html/prodN
# A.html
data_nulls <- data_heart

# 1% de valors nulls
data_nulls <- prodNA(data_nulls, noNA = 0.01)

print("Data amb valors nulls")
```

```
## [1] "Data amb valors nulls"
```

```
colSums(is.na(data_nulls))
```

```
##      age      sex      cp  trtbps      chol      fbs  restecg thalachh
##      3        3        5      3        3        6        2        2
##      exng  oldpeak    slp      caa    thall    output
##      3        1        2      3      3        3
```

```
# Realitzem els calculs de KNN
data_knn_filled <- knn(data_nulls, variable = names(data_nulls), k = 10)
```

```
## Warning in `[<-data.table`(`*tmp*`, indexNA2s[, variable[j]], variable[j], :
## 47.500000 (type 'double') at RHS position 1 truncated (precision lost) when
## assigning to type 'integer' (column 1 named 'age')
```

```
## Warning in `[<-.data.table`(`*tmp*`, indexNA2s[, variable[j]], variable[j], :  
## 121.500000 (type 'double') at RHS position 3 truncated (precision lost) when  
## assigning to type 'integer' (column 4 named 'trtbps')
```

```
## Warning in `[<-.data.table`(`*tmp*`, indexNA2s[, variable[j]], variable[j], :  
## 248.500000 (type 'double') at RHS position 1 truncated (precision lost) when  
## assigning to type 'integer' (column 5 named 'chol')
```

```
## Warning in `[<-.data.table`(`*tmp*`, indexNA2s[, variable[j]], variable[j], :  
## 172.500000 (type 'double') at RHS position 1 truncated (precision lost) when  
## assigning to type 'integer' (column 8 named 'thalachh')
```

```
## Warning in `[<-.data.table`(`*tmp*`, indexNA2s[, variable[j]], variable[j], :  
## 1.500000 (type 'double') at RHS position 3 truncated (precision lost) when  
## assigning to type 'integer' (column 12 named 'caa')
```

```
colSums(is.na(data_knn_filled))
```

```
##      age      sex      cp      trtbps      chol      fbs  
##      0        0        0        0        0        0  
##  restecg  thalachh      exng    oldpeak      slp      caa  
##      0        0        0        0        0        0  
##   thall    output    age_imp  sex_imp    cp_imp  trtbps_imp  
##      0        0        0        0        0        0  
##  chol_imp  fbs_imp  restecg_imp  thalachh_imp  exng_imp  oldpeak_imp  
##      0        0        0        0        0        0  
##   slp_imp  caa_imp   thall_imp  output_imp  
##      0        0        0        0
```

```
# Observem que data_knn_filled augmenta les variables a 28.  
# Ens quedem amb les dades utils, les que coincideixen amb el dataframe inicial  
data_knn_filtred <- subset(data_knn_filled, select = names(data_nulls))  
summary(data_knn_filtred)
```

```
##           age           sex           cp           trtbps
## Min.      :29.00   Min.      :0.0000   Min.      :0.0000   Min.      : 94.0
## 1st Qu.:47.00   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:120.0
## Median :55.00   Median :1.0000   Median :1.0000   Median :130.0
## Mean      :54.34   Mean      :0.6865   Mean      :0.9637   Mean      :131.6
## 3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:2.0000   3rd Qu.:140.0
## Max.      :77.00   Max.      :1.0000   Max.      :3.0000   Max.      :200.0
##           chol           fbs           restecg           thalachh
## Min.      :126.0   Min.      :0.0000   Min.      :0.0000   Min.      : 71.0
## 1st Qu.:211.0   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:133.5
## Median :241.0   Median :0.0000   Median :1.0000   Median :153.0
## Mean      :246.2   Mean      :0.1419   Mean      :0.5248   Mean      :149.7
## 3rd Qu.:274.0   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:166.0
## Max.      :564.0   Max.      :1.0000   Max.      :2.0000   Max.      :202.0
##           exng           oldpeak           slp           caa
## Min.      :0.0000   Min.      :0.000   Min.      :0.000   Min.      :0.0000
## 1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:1.000   1st Qu.:0.0000
## Median :0.0000   Median :0.800   Median :1.000   Median :0.0000
## Mean      :0.3234   Mean      :1.039   Mean      :1.399   Mean      :0.7228
## 3rd Qu.:1.0000   3rd Qu.:1.600   3rd Qu.:2.000   3rd Qu.:1.0000
## Max.      :1.0000   Max.      :6.200   Max.      :2.000   Max.      :4.0000
##           thall           output
## Min.      :0.000   Min.      :0.0000
## 1st Qu.:2.000   1st Qu.:0.0000
## Median :2.000   Median :1.0000
## Mean      :2.317   Mean      :0.5446
## 3rd Qu.:3.000   3rd Qu.:1.0000
## Max.      :3.000   Max.      :1.0000
```

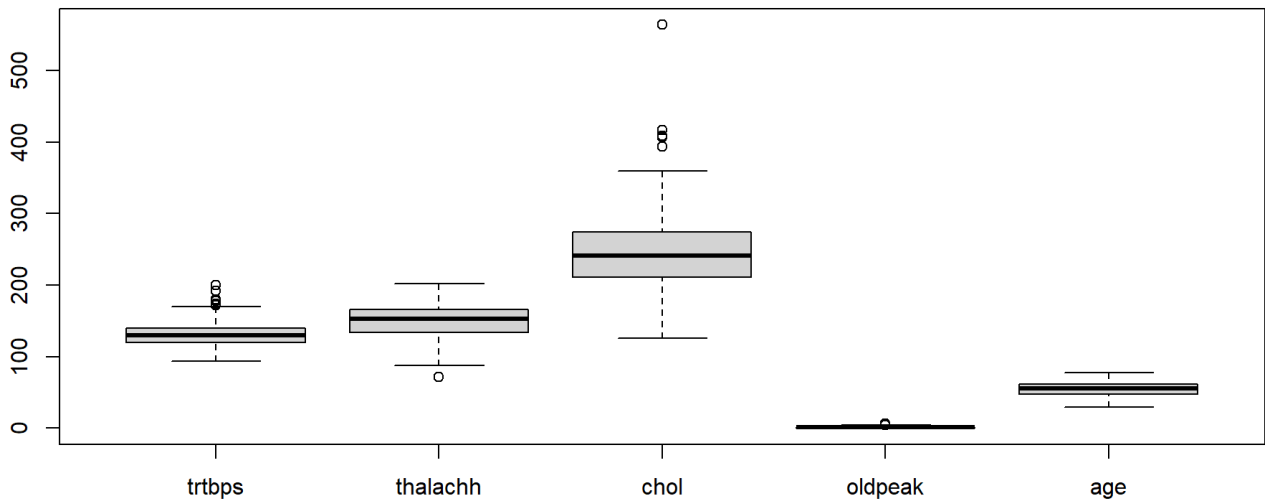
Identifica i gestiona valors extrems

Per trobar els valors extrems, hem considerat que quan un valor es troba allunyat 3 desviacions estàndard respecte a la mitjana del conjunt és un outlier. Per trobar-los hem representant les dades en boxplots. Hem realitzat aquesta comprovació a les variables numèriques següent:

```
data_numerical <- as.data.frame(data_knn_filtred %>%
  select("trtbps", "thalachh", "chol", "oldpeak", "age"))
```

BoxPlot:

```
boxplot <- boxplot(data_numerical)
```



Com es pot veure, les columnes oldpeak, chol, thalachh i trtbps tenen possibles outliers. Per eliminar-los, he creat la següents funcions per seleccionar aquells valor 3 desviacions standards allunyats:

```
outliers = function(x) {

  standardD <- sd(x)
  mitjana <- mean(x)

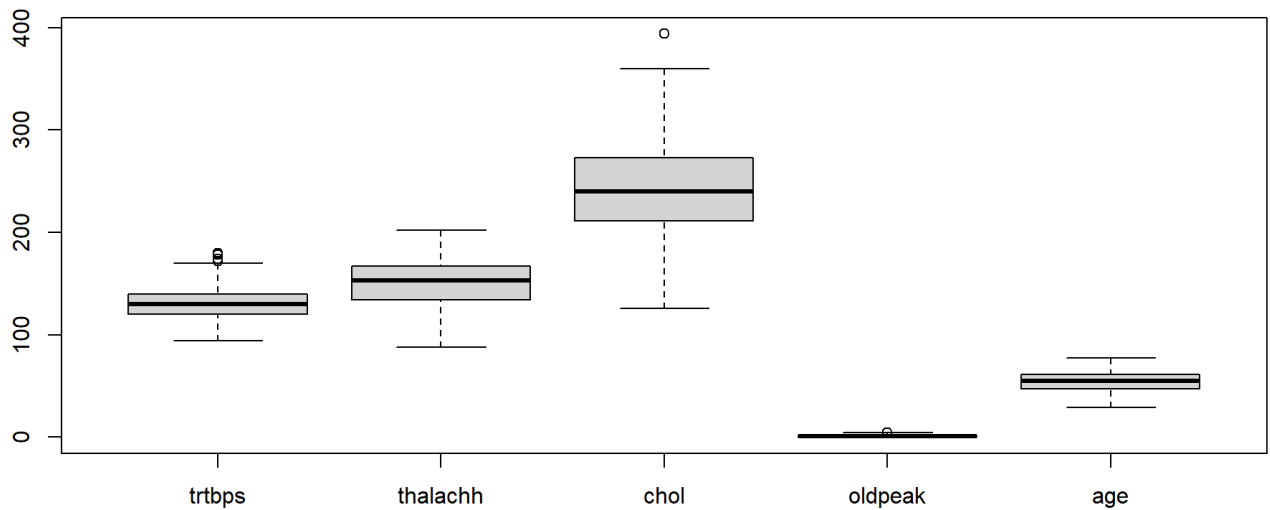
  #Upper Range
  upper_limit = mitjana + 3*standardD
  #Lower Range
  lower_limit = mitjana - 3*standardD

  x > upper_limit | x < lower_limit
}

# remove the outliers
elimina_outliers <- function(df_outliers, cols = names(df_outliers)) {
  for (col in cols) {
    df_outliers<- df_outliers[!outliers(df_outliers[[col]]),]
  }
  df_outliers
}
```

Apliquem aquesta funció i mostrem els nous boxplots, a on es pot veure la disminució de outliers:

```
data_heart<-elimina_outliers(data_knn_filtred,c("trtbps","oldpeak" ,"thalachh", "chol"))
data_numerical <- as.data.frame(data_heart %>%
  select("trtbps","thalachh","chol","oldpeak", "age"))
boxplot <- boxplot(data_numerical)
```



Anàlisi de les dades

Selecció de dades

Obtenim les dades estadístiques bàsiques del dataset

```
summary(data_heart)
```

```
##      age      sex      cp      trtbps
## Min.   :29.00   Min.   :0.0000   Min.   :0.0000   Min.    : 94.0
## 1st Qu.:47.00   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:120.0
## Median :55.00   Median :1.0000   Median :1.0000   Median :130.0
## Mean   :54.15   Mean    :0.6973   Mean    :0.9762   Mean    :131.1
## 3rd Qu.:61.00   3rd Qu.:1.0000   3rd Qu.:2.0000   3rd Qu.:140.0
## Max.    :77.00   Max.    :1.0000   Max.    :3.0000   Max.    :180.0
##      chol      fbs      restecg      thalachh
## Min.    :126.0   Min.    :0.0000   Min.    :0.000   Min.     : 88.0
## 1st Qu.:211.0   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:134.5
## Median :240.0   Median :0.0000   Median :1.000   Median :153.0
## Mean    :243.6   Mean     :0.1395   Mean     :0.534   Mean     :150.0
## 3rd Qu.:272.5   3rd Qu.:0.0000   3rd Qu.:1.000   3rd Qu.:166.8
## Max.    :394.0   Max.     :1.0000   Max.     :2.000   Max.     :202.0
##      exng      oldpeak      slp      caa
## Min.    :0.0000   Min.    :0.000   Min.    :0.000   Min.    :0.0000
## 1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:1.000   1st Qu.:0.0000
## Median :0.0000   Median :0.600   Median :1.000   Median :0.0000
## Mean    :0.3231   Mean     :0.985   Mean     :1.415   Mean     :0.7041
## 3rd Qu.:1.0000   3rd Qu.:1.600   3rd Qu.:2.000   3rd Qu.:1.0000
## Max.    :1.0000   Max.     :4.400   Max.     :2.000   Max.     :4.0000
##      thall      output
## Min.    :0.000   Min.    :0.0000
## 1st Qu.:2.000   1st Qu.:0.0000
## Median :2.000   Median :1.0000
## Mean    :2.303   Mean     :0.5544
## 3rd Qu.:3.000   3rd Qu.:1.0000
## Max.    :3.000   Max.     :1.0000
```

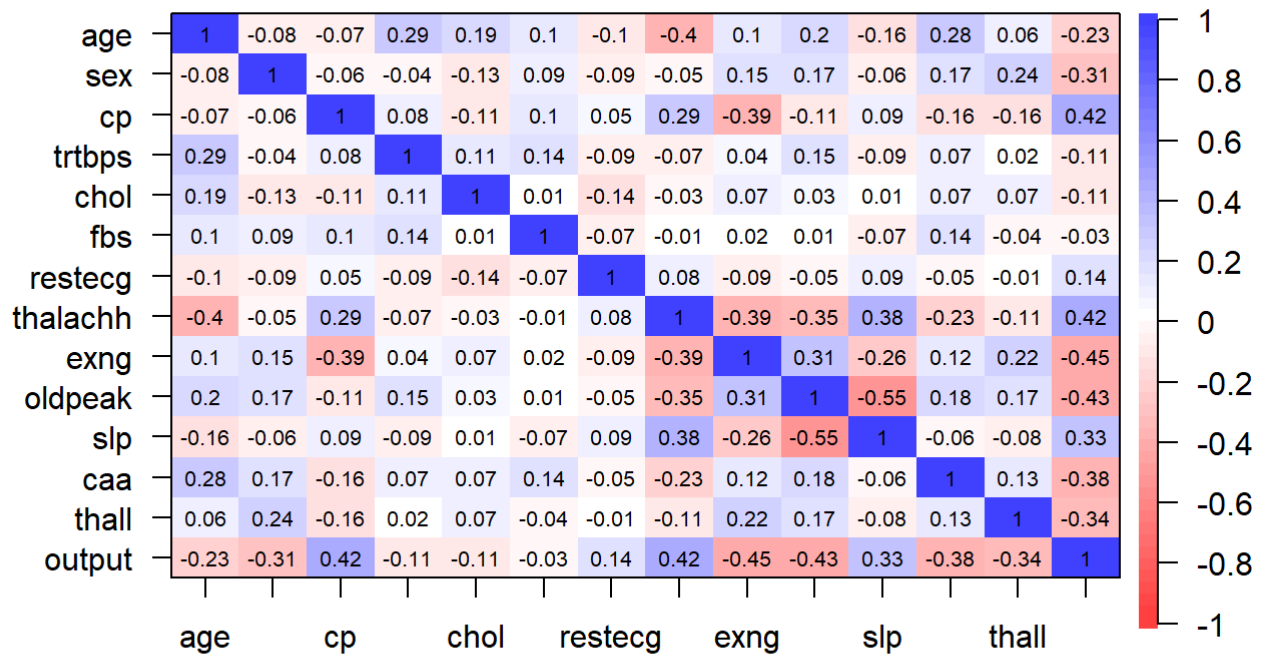
```
# Mostrem els primers 5 valors
head(data_heart, n = 5)
```

```
##   age sex cp trtbps chol fbs restecg thalachh exng oldpeak slp caa thall output
## 1  63  1  3   145  233   1      0     150    0    2.3   0   0    1    1
## 2  37  1  2   130  250   0      1     187    0    3.5   0   0    2    1
## 3  41  0  1   130  204   0      0     172    0    1.4   2   0    2    1
## 4  56  1  1   120  236   0      1     178    0    0.8   2   0    2    1
## 5  57  0  0   120  354   0      1     163    1    0.6   2   0    2    1
```

Aprofitarem que totes les variables estan introduïdes com a Integers per tal d'obtenir una matriu de correlació i veure quins atributs tenen major correlació amb el resultat i per tant, semblen més rellevants a analitzar.

```
# Enllaç: https://sparkbyexamples.com/r-programming/r-select-function-from-dplyr/
# Anàlisi de correlació, quines dades tenen més relació entre elles i el output?
corr_df <- cor(data_heart)
corPlot(corr_df, main = "Matriu de correlació")
```


Matriu de correlació



Observem en la matriu de correlació que: - La variable "sex" no té força correlació amb cap dels atributs, sembla que no es un factor rellevant. Tot i així, sorpren que a nivell "general" la percepció de la població és que hi ha més homes amb aquest problema. - La variable "edad" sembla no influir gaire en les variables 'categoriques'. - La variable "output" esta correlativament influenciada per "cp" i "thalachh".

Per tant, semblen tenir algunes hipotesis que podem comprobar, ¿Hi ha més problemes cardiovasculars entre homes o entre dones?

Seguim amb l'anàlisi preliminar per visualitzar gràficament les variables categoriques i veure si hi ha relació o no amb l'output.

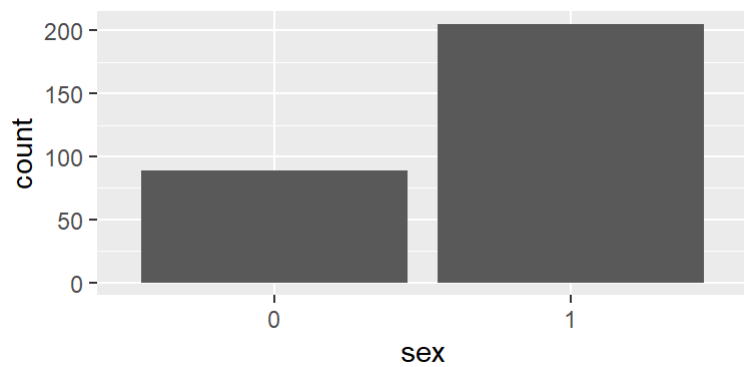
```
# Reestablim les dades ara sense els outliers.
data_numerical <- as.data.frame(data_heart %>%
  select("trtbps","thalachh","chol","oldpeak", "age"))

# Obtenim les variables categoriques i les transformem al tipus necessari
# Link : https://gist.github.com/ramhiser/93fe37be439c480dc26c4bed8aab03dd

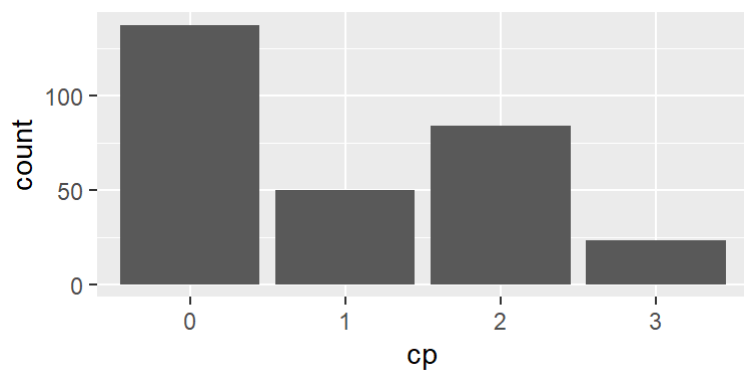
data_categorical <- as.data.frame(data_heart %>%
  select("sex","cp","fbs","restecg", "exng", "slp", "caa", "thall", "output"))

data_categorical <- data_categorical %>% mutate_if(is.numeric, as.factor)

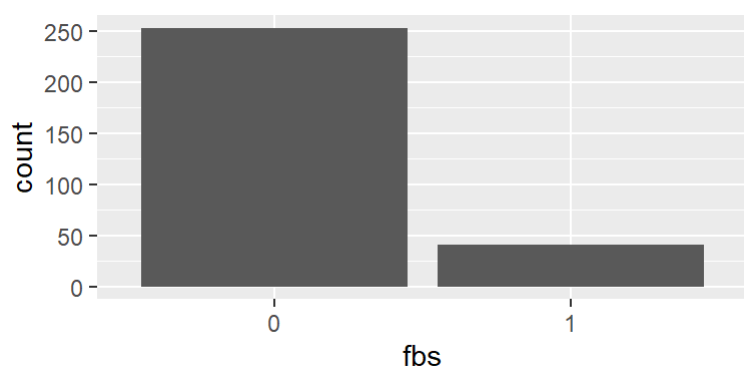
# Representem graficament les diferents variables categoriques
ggplot(data = data_categorical, aes(x = sex))+geom_bar()
```



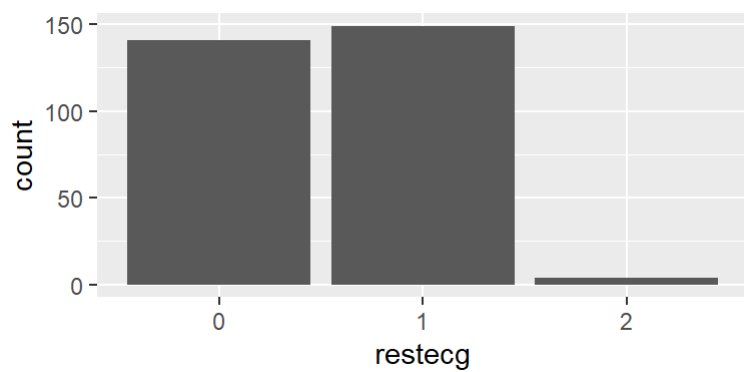
```
ggplot(data = data_categorical, aes(x = cp))+geom_bar()
```



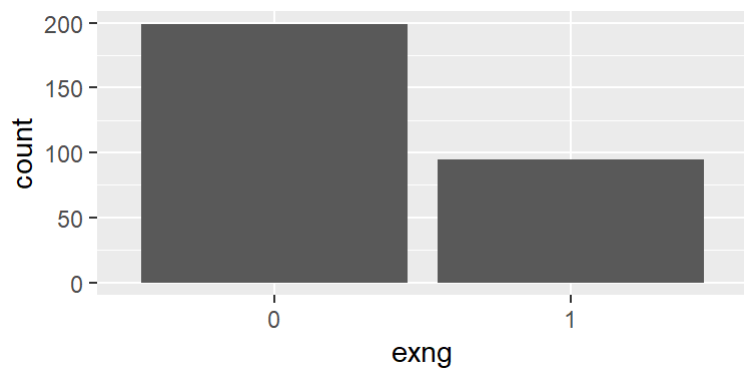
```
ggplot(data = data_categorical, aes(x = fbs))+geom_bar()
```



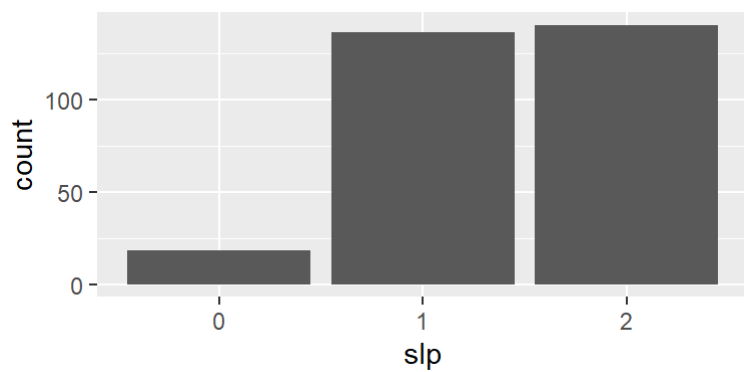
```
ggplot(data = data_categorical, aes(x = restecg))+geom_bar()
```



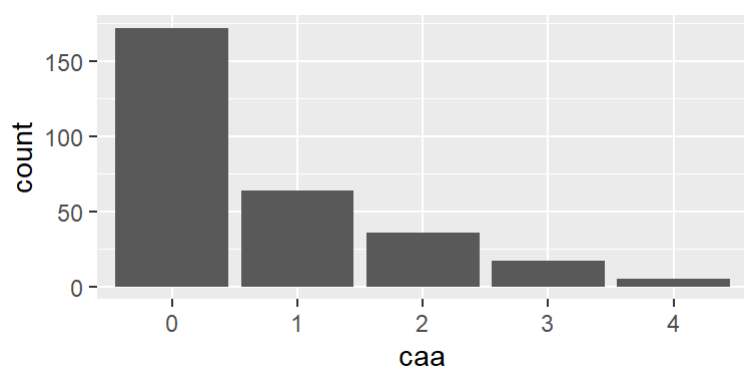
```
ggplot(data = data_categorical, aes(x = exng))+geom_bar()
```



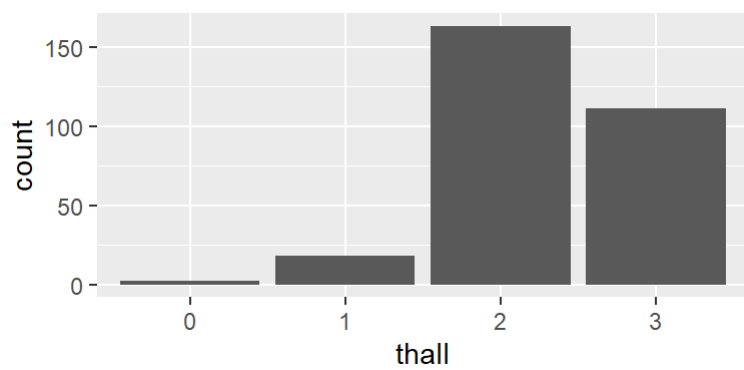
```
ggplot(data = data_categorical, aes(x = slp))+geom_bar()
```



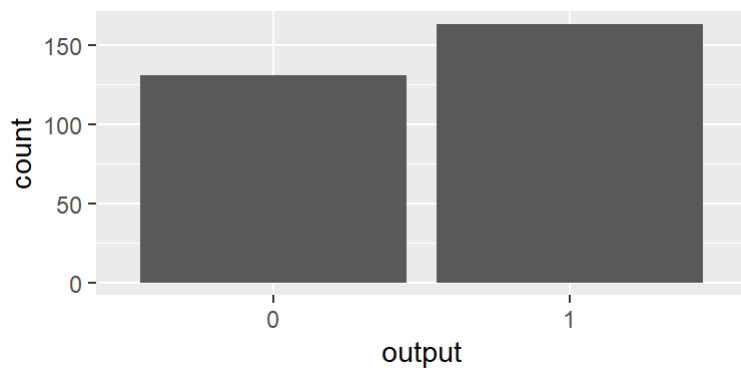
```
ggplot(data = data_categorical, aes(x = caa))+geom_bar()
```



```
ggplot(data = data_categorical, aes(x = thall))+geom_bar()
```



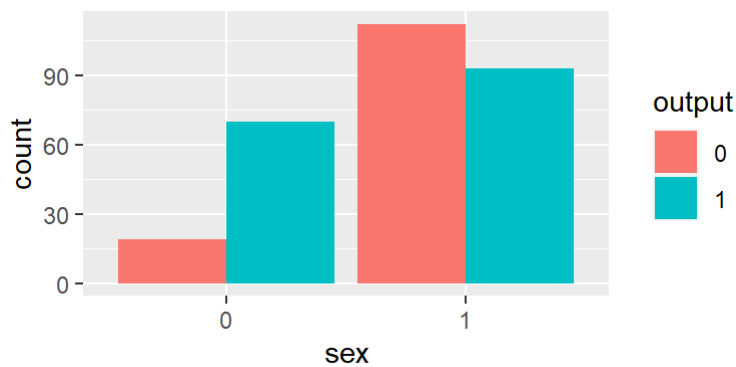
```
ggplot(data = data_categorical, aes(x = output))+geom_bar()
```



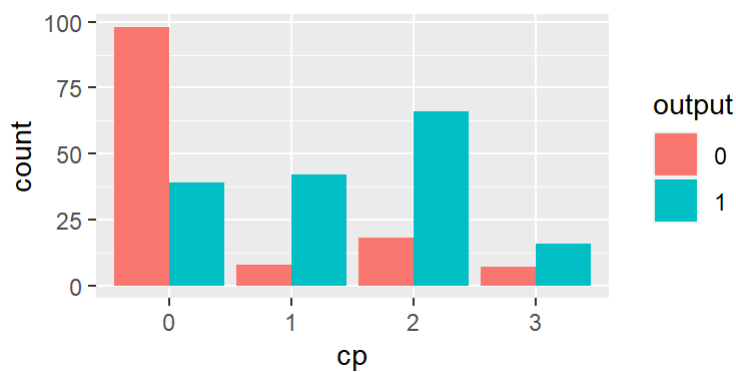
Representem les variables dividides per genere per veure si hi ha diferencies en les variables que sortissin a la corrMatrix (?):

Error: <https://stackoverflow.com/questions/24895575/ggplot2-bar-plot-with-two-categorical-variables>

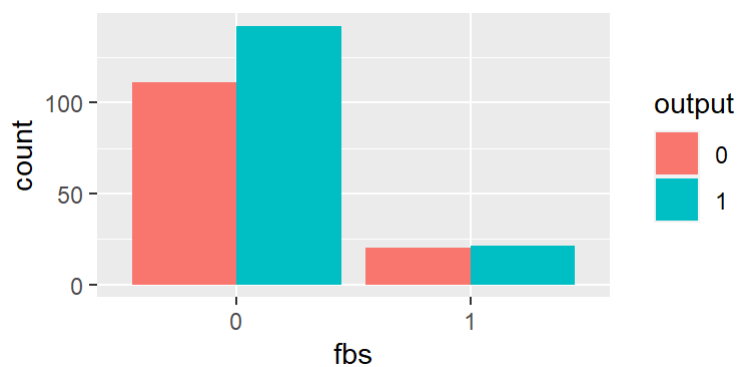
```
ggplot(data = data_categorical, aes(x = sex, after_stat(count) ))+geom_bar(aes(fill = output), position = "dodge")
```



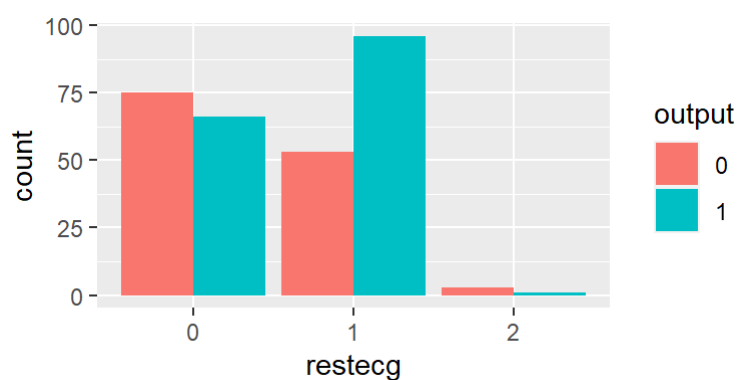
```
ggplot(data = data_categorical, aes(x = cp, after_stat(count) ))+geom_bar(aes(fill = output), position = "dodge")
```



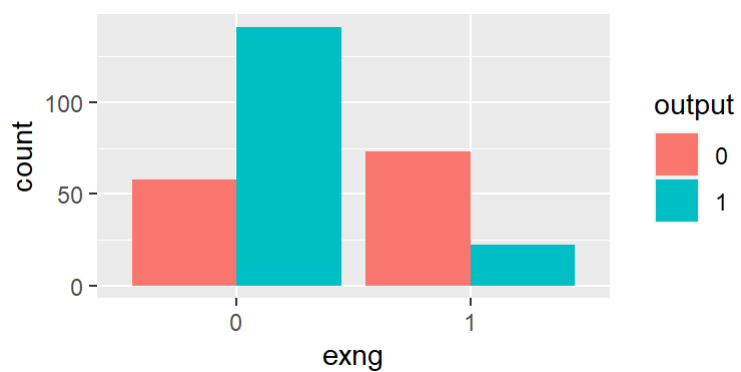
```
ggplot(data = data_categorical, aes(x = fbs, after_stat(count) ))+geom_bar(aes(fill = output), position = "dodge")
```



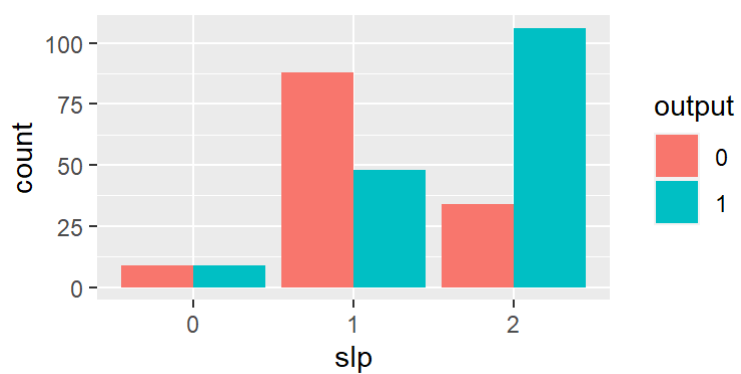
```
ggplot(data = data_categorical, aes(x = restecg, after_stat(count) ))+geom_bar(aes(fill = output), position = "dodge")
```



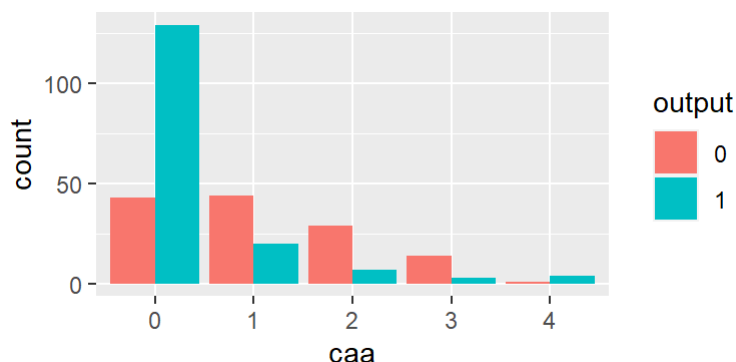
```
ggplot(data = data_categorical, aes(x = exng, after_stat(count) ))+geom_bar(aes(fill = output), position = "dodge")
```



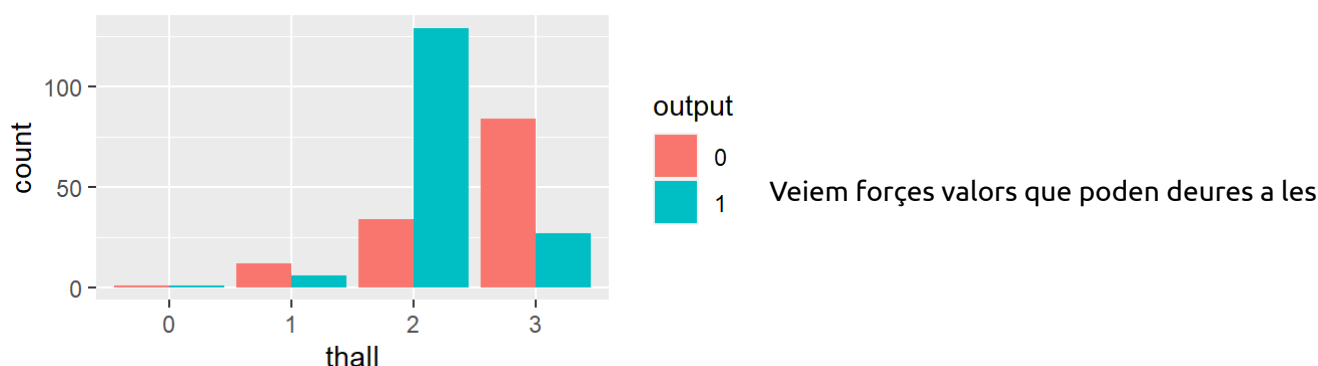
```
ggplot(data = data_categorical, aes(x = slp, after_stat(count) ))+geom_bar(aes(fill = output), position = "dodge")
```



```
ggplot(data = data_categorical, aes(x = caa, after_stat(count) ))+geom_bar(aes(fill = output), position = "dodge")
```



```
ggplot(data = data_categorical, aes(x = thall, after_stat(count) ))+geom_bar(aes(fill = output), position = "dodge")
```



poques mostres del dataset (14 variables per 300 mostres, no són gaires), tot i així sembla que hi ha més del doble de pacients femenins que masculins, per tant les dades son 100~200).

Comprovació de la normalitat i homogeneïtat de la variància:

Comprovació de la normalitat

```
columns = c("name","normalitat")
df = data.frame(matrix(nrow = 0, ncol = length(columns)))
colnames(df) = columns
data_numerical <- as.data.frame(data_heart %>% select("trtbps","thalachh","chol","oldpeak",
"age"))
for (col in names(data_numerical)) {
  normalitat <- TRUE
  resultat <- shapiro.test(data_numerical[[col]])
  if(resultat$p.value<0.05){
    #No Normalitat
    normalitat<-FALSE
  }
  df[nrow(df)+1,1] = col
  df[nrow(df),2] = normalitat
}
print(df)
```

```
##      name normalitat
## 1   trtbps      FALSE
## 2 thalachh      FALSE
## 3     chol       TRUE
## 4  oldpeak      FALSE
## 5     age       FALSE
```

Comprovació Homoscedasticitat

Comprovació homoscedasticitat per la variable chol (única variable amb normalitat en la variància) amb els següents grups: sex, cp, fbs, restecg, exng, slp, thall, output

```
columns = c("name1", "name2", "homoscedasticitat")
df = data.frame(matrix(nrow = 0, ncol = length(columns)))
colnames(df) = columns
chol <- data_heart[["chol"]]
for (col in names(data_categorical)) {
  homogeneitat <- TRUE
  test <- leveneTest(chol~data_categorical[[col]])
  if(test$`Pr(>F)`[1]<0.05){
    homogeneitat <- FALSE
  }
  df[nrow(df)+1,1:2] <- c("chol", col)
  df[nrow(df),3] <- homogeneitat
}
print(df)
```

```
##   name1  name2 homoscedasticitat
## 1  chol    sex                TRUE
## 2  chol    cp                TRUE
## 3  chol    fbs                TRUE
## 4  chol restecg              TRUE
## 5  chol    exng              TRUE
## 6  chol    slp              TRUE
## 7  chol    caa              TRUE
## 8  chol    thall            TRUE
## 9  chol  output            TRUE
```

```
data_numerical_nn <- as.data.frame(data_heart %>%
  select("trtbps", "thalachh", "oldpeak", "age"))
for(numerical in names(data_numerical_nn)){
  for (col in names(data_categorical)) {
    homogeneitat <- TRUE
    test <- fligner.test(data_numerical_nn[[numerical]]~data_categorical[[col]])

    if(test$p.value<0.05){
      homogeneitat <- FALSE
    }
    df[nrow(df)+1,1:2] <- c(numerical,col)
    df[nrow(df),3] <- homogeneitat
  }
}
print(df)
```

##	name1	name2	homoscedasticitat
## 1	chol	sex	TRUE
## 2	chol	cp	TRUE
## 3	chol	fbs	TRUE
## 4	chol	restecg	TRUE
## 5	chol	exng	TRUE
## 6	chol	slp	TRUE
## 7	chol	caa	TRUE
## 8	chol	thall	TRUE
## 9	chol	output	TRUE
## 10	trtbps	sex	TRUE
## 11	trtbps	cp	FALSE
## 12	trtbps	fbs	TRUE
## 13	trtbps	restecg	TRUE
## 14	trtbps	exng	TRUE
## 15	trtbps	slp	TRUE
## 16	trtbps	caa	TRUE
## 17	trtbps	thall	TRUE
## 18	trtbps	output	TRUE
## 19	thalachh	sex	FALSE
## 20	thalachh	cp	TRUE
## 21	thalachh	fbs	TRUE
## 22	thalachh	restecg	TRUE
## 23	thalachh	exng	TRUE
## 24	thalachh	slp	FALSE
## 25	thalachh	caa	TRUE
## 26	thalachh	thall	TRUE
## 27	thalachh	output	TRUE
## 28	oldpeak	sex	FALSE
## 29	oldpeak	cp	FALSE
## 30	oldpeak	fbs	TRUE
## 31	oldpeak	restecg	FALSE
## 32	oldpeak	exng	FALSE
## 33	oldpeak	slp	FALSE
## 34	oldpeak	caa	FALSE
## 35	oldpeak	thall	FALSE
## 36	oldpeak	output	FALSE
## 37	age	sex	TRUE
## 38	age	cp	TRUE
## 39	age	fbs	FALSE
## 40	age	restecg	TRUE
## 41	age	exng	TRUE
## 42	age	slp	TRUE
## 43	age	caa	FALSE
## 44	age	thall	FALSE
## 45	age	output	FALSE

Proves estadístiques:

Contrast d'hipotesis

El contrast d'hipòtesis el farem per determinar el si el gènere influeix.

Es planteja el següent contrast d'hipòtesis:

- Hipòtesi nul·la: La probabilitat de patir malaltia és independent al gènere.
- Hipòtesi alternativa: La probabilitat de patir malalties és major en homes.

```
home_mostra <- data_heart[data_heart$sex == 0,]$output
dona_mostra <- data_heart[data_heart$sex == 1,]$output
t.test(dona_mostra, home_mostra, alternative = 'less')
```

```
##
## Welch Two Sample t-test
##
## data: dona_mostra and home_mostra
## t = -5.9562, df = 200.65, p-value = 5.698e-09
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf -0.2405111
## sample estimates:
## mean of x mean of y
## 0.4536585 0.7865169
```

Com es pot veure, es pot confirmar la hipòtesis plantejada ja que el gènere influeix. Aquest resultat es podia intuir en els gràfic entre sex i output.

Models

Preparació de les dades

```
# Recuperem les dades numèriques i les normalitzem
data_numerical_norm <- scale(data_numerical)

# combinem les dades i creem el dataframe definitiu amb el que farem els models
data_model = cbind(data_numerical_norm, data_categorical)

# Dividirem el dataset en train - test
set.seed(123)
split = sample.split(data_model$output, SplitRatio = 0.80)
train_set = subset(data_model, split == TRUE)
test_set = subset(data_model, split == FALSE)
```

Naive Bayes

Utilitzem el model Naive Bayes per predir la classificació dels pacients. Per tant podríem donar resposta a la pregunta formalitzada a l'inici de la pràctica "Podem predir el risc d'un pacient de patir problemes cardíacs?".

```
# https://www.rdocumentation.org/packages/e1071/versions/1.7-12/topics/naiveBayes
set.seed(123)
model_nB = naiveBayes(output~., data=train_set)
y_pred = predict(model_nB, newdata = test_set)
y_pred
```

```
## [1] 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 0 0 1 0 0
## [39] 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1
## Levels: 0 1
```

```
cm_results = confusionMatrix(table(test_set$output, y_pred))
print(cm_results)
```

```
## Confusion Matrix and Statistics
##
##      y_pred
##      0  1
## 0 22  4
## 1  4 29
##
##                Accuracy : 0.8644
##                95% CI : (0.7502, 0.9396)
##      No Information Rate : 0.5593
##      P-Value [Acc > NIR] : 5.256e-07
##
##                Kappa : 0.7249
##
##  Mcnemar's Test P-Value : 1
##
##      Sensitivity : 0.8462
##      Specificity : 0.8788
##      Pos Pred Value : 0.8462
##      Neg Pred Value : 0.8788
##      Prevalence : 0.4407
##      Detection Rate : 0.3729
##      Detection Prevalence : 0.4407
##      Balanced Accuracy : 0.8625
##
##      'Positive' Class : 0
##
```

Com podem veure, amb aquest model hem aconseguit una accuracy de 0.7195.

KNN

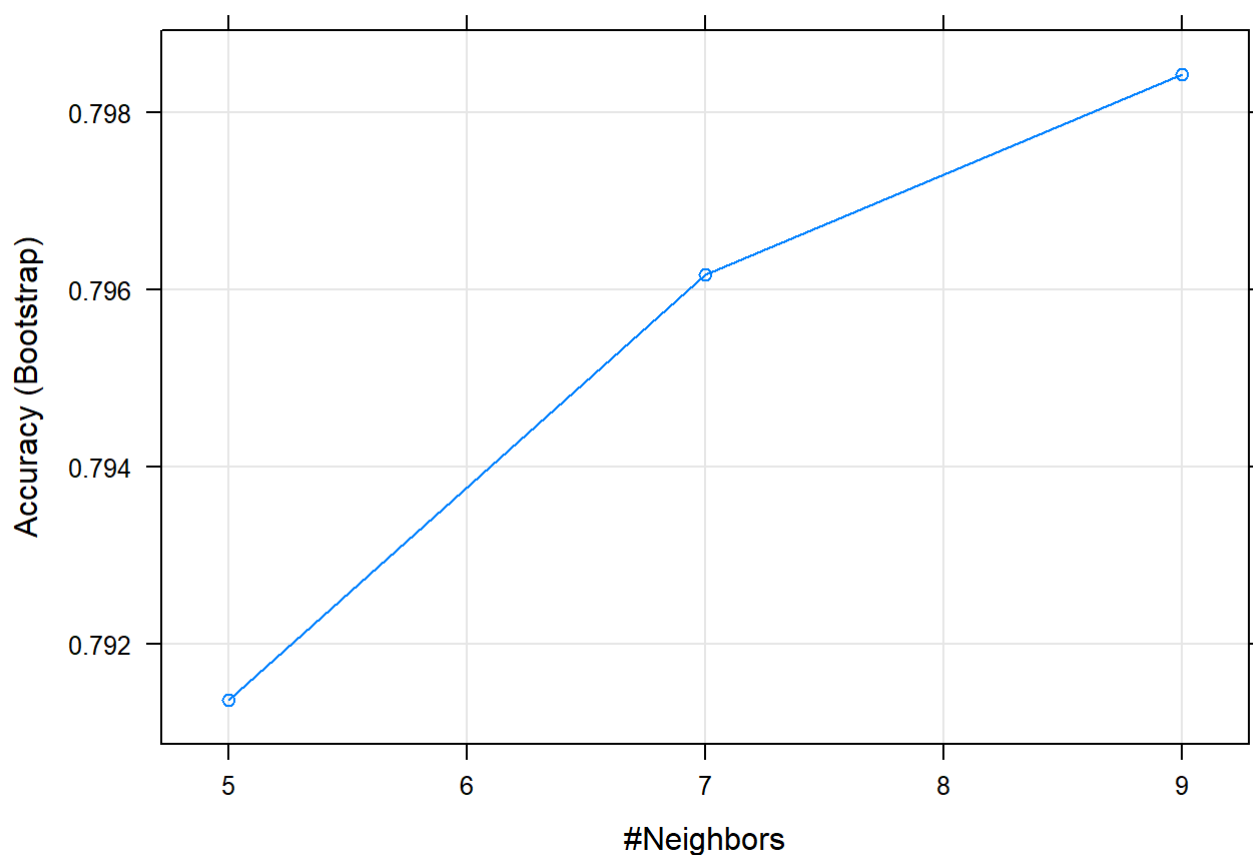
Utilitzem el model KNN per predir la classificació dels pacients. Per tant podríem donar resposta a la pregunta formalitzada a l'inici de la pràctica "Podem predir el risc d'un pacient de patir problemes cardíacs?". Utilitzem aquest algoritme perquè és un algoritme classificador i ens podria servir per agrupar diferents tipus de pacients i fer diferents seguiments mèdics.

```
# https://rpubs.com/njvijay/16444
set.seed(123)

knn_fit <- train(output ~., data = train_set, method = "knn")
knn_fit
```

```
## k-Nearest Neighbors
##
## 235 samples
## 13 predictor
## 2 classes: '0', '1'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 235, 235, 235, 235, 235, ...
## Resampling results across tuning parameters:
##
## k Accuracy Kappa
## 5 0.7913686 0.5778019
## 7 0.7961744 0.5873663
## 9 0.7984314 0.5918746
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 9.
```

```
plot(knn_fit)
```



Extracció Dataset

```
res <- c(data_heart, "./heartClean.csv", row.names=TRUE)
```

Conclusió:

Inicialment volíem donar resposta als parametres més importants i si podem preveure nous pacients amb èxit.

Creiem que amb els resultats de la matriu de correlació i els anàlisis fets a les variables tant numèriques com categòriques, sabem que els atributs cp i thalachh tenen més impacte.

En quant a preveure nous pacients, concluïm creient que és un camí possible sempre que hi hagi col·laboració d'experts en el domini, i que les mostres de dades siguin molt més representatives, ja que hi havia força dones en comparació i les franjes d'edat estaven poc representades (impedint comprovar que passa en diferents franjes d'edat).

Considerem interessant aplicar diferents algorismes com el KNN per pacients amb una determinada edat i veure si l'accuracy augmenta o no..

Hi ha per tant moltes preguntes a fer amb aquestes dades, esperem que amb el temps es doni resposta i la ciència de dades sigui clau en el desenvolupament de la salut de les persones.

Contribucions

Contribucions	Firma
Investigació prèvia	Marc González i Maria Sunyer
Redacció de les respostes	Marc González i Maria Sunyer
Desenvolupament del codi	Marc González i Maria Sunyer
Participació al vídeo	Marc González i Maria Sunyer