Pràctica 2 - Neteja i anàlisi de les dades

Tipologia i cicle de vida de les dades

Autor: Narcís Bustins Núñez

Desembre del 2019

Contents

1 Descripció del dataset

El joc seleccionat s'ha obtingut al repositori de machine learning UCI, el seu títol és "Hepatitis Data Set". Aquest dataset és molt interessant per aquesta pràctica, ja que permetrà aplicar algorismes de machine learining supervisats, disposa d'una etiqueta Class que indica a quina classe pertany cada observació. La classe indica si el pacient ha mort o ha sobreviscut. El dataset compta amb variables tant qualitatives com quantitatives i això aportarà varietat en el tractament de les dades. En l'àmbit mèdic és molt interessant, ja que permet fer models per tal de fer el seguiment de pacients d'hepatitis diagnosticar si és greu, si sobreviurà o morirà i descobrir quines variables són més importants és a dir, quines aporten més informació útil a l'hora de classificar el pacient.

També hi ha motius d'interès personal per haver seleccionat aquest dataset, ja que em sembla especialment interessant com la ciència de dades pot ajudar al camp mèdic fent estudis per tal de diagnosticar o predir diverses malalties i així poder ajudar a molts humans. Aquest dataset es pot utilitzar amb diversos propòsits, per exemple crear algorismes tant de deep learining com de machine learining per ajudar a fer el seguiment dels pacients amb hepatitis, es pot predir si el pacient viurà o morirà, això pot ajudar a comprovar si el tractament que s'està seguint està funcionant o no. També es poden buscar les relacions que tenen els atributs, quines influeixen més pel diagnòstic.

El dataset es pot obtenir en el següent link: https://archive.ics.uci.edu/ml/datasets/hepatitis

Aquests són els atributs presents al dataset, els valors de les variables qualitatives estan representats numèricament però en aquesta descripció indicarem el significat d'aquests.

- **Class**: Classe a la qual pertany el pacient, viu o mor (Die/Live).
- **AGE**: Indica l'edat del pacient.
- **SEX**: Indica el sexe del pacient(Male/Female).
- **STEROID**: Indica si el pacient ha pres esteroides (yes/no).
- **ANTIVIRALS**: Indica si el pacient ha pres antivirals (yes/no).
- **FATIGUE**: Indica si el pacient és sent fatigat o no (yes/no).
- **MALAISE**: Indica si el pacient sent malestar (yes/no).
- **ANOREXIA**: Indica si el pacient pateix anorèxia (yes/no).
- **LIVER BIG**: Indica si la mida del fetge ha augmentat (yes/no).
- **LIVER FIRM**: Indica si el fetge és manté ferm (yes/no).
- **SPLEEN PAL**: Indica si el pacient presenta esplenomegàlia, una ampliació de la melsa (yes/no).
- **SPIDERS**: Indica si el pacient presenta aranyes(vasos sanguinis engrandits) visibles (yes/no).
- **ASCITES**: Indica si hi ha presencia de líquid a la cavitat peritoneal (yes/no).
- **VARICES**: Indica si el pacient presenta varius (yes/no).
- **BILIRUBIN**: Indica el nivell de bilirubina del pacient.
- **ALK PHOSPH**: Indica el valor de la fosfatasa alcalina del pacient.
- **SGOT**: Valor obtingut amb l'anàlisi de sang del pacient després de la prova AST.
- **ALBUMIN**: Indica el valor de la proteïna albúmina del pacient.
- **PROTIME**: Indica el valor de la característica del pacient.
- **HISTOLOGY**: Indica el valor de'estudiar la histología (estudis microscopics)(yes/no).

2 Objectius de l'estudi

A partir del dataset definit es buscarà crear un models classificadors per tal d'identificar si un pacient està en risc de morir per hepatitis o no, es buscarà identificar les variables que influeixen més per obtenir aquest coneixement, és a dir les més significatives. Es plantejaran diferents hipòtesis a partir de les variables que es disposen per tal de buscar obtenir més coneixement i descobrir propietats d'interès. Aquest tipus d'anàlisi tenen molta importància en el sector mèdic, es poden crear algorismes de suport per tal d'ajudar als doctors a l'hora de fer el seguiment d'un pacient.

3 Neteja de les dades

Llegim el fitxer.

```
dhep <- read.csv("../data/hepatitis.csv")
dattrs <- c("Class","AGE","SEX","STEROID","ANTIVIRALS","FATIGUE","MALAISE","ANOREXIA",
            "LIVER_BIG","LIVER_FIRM","SPLEEN_PAL","SPIDERS","ASCITES","VARICES",
            "BILIRUBIN","ALK_PHOSPHATE","SGOT","ALBUMIN","PROTIME","HISTOLOGY")

names(dhep) <- dattrs
```

Com podem veure el dataset compta amb 154 files amb 20 variables (columnes).

```
dim(dhep)
```

```
## [1] 154  20
```

Primer mirem quin tipus s'ha assignat a cada columna. Es pot veure com hi ha variables que s'han llegit amb el tipus erroni, haurem de corregir el tipus assignat.

```
str(dhep)
```

```
## 'data.frame':    154 obs. of  20 variables:
##  $ Class        : int  2 2 2 2 2 1 2 2 2 2 ...
##  $ AGE          : int  50 78 31 34 34 51 23 39 30 39 ...
##  $ SEX          : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ STEROID      : Factor w/ 3 levels "?","1","2": 2 3 1 3 3 2 3 3 3 2 ...
##  $ ANTIVIRALS   : int  2 2 1 2 2 2 2 2 2 1 ...
##  $ FATIGUE      : Factor w/ 3 levels "?","1","2": 2 2 3 3 3 2 3 2 3 3 ...
##  $ MALAISE      : Factor w/ 3 levels "?","1","2": 3 3 3 3 3 3 3 3 3 3 ...
##  $ ANOREXIA     : Factor w/ 3 levels "?","1","2": 3 3 3 3 3 2 3 3 3 3 ...
##  $ LIVER_BIG    : Factor w/ 3 levels "?","1","2": 2 3 3 3 3 3 3 3 3 2 ...
##  $ LIVER_FIRM   : Factor w/ 3 levels "?","1","2": 3 3 3 3 3 3 3 2 3 2 ...
##  $ SPLEEN_PAL   : Factor w/ 3 levels "?","1","2": 3 3 3 3 3 2 3 3 3 3 ...
##  $ SPIDERS      : Factor w/ 3 levels "?","1","2": 3 3 3 3 3 2 3 3 3 3 ...
##  $ ASCITES      : Factor w/ 3 levels "?","1","2": 3 3 3 3 3 3 3 3 3 3 ...
##  $ VARICES      : Factor w/ 3 levels "?","1","2": 3 3 3 3 3 3 3 3 3 3 ...
##  $ BILIRUBIN    : Factor w/ 35 levels "?","0.30","0.40",..: 8 6 6 9 8 1 9 6 9 12 ...
##  $ ALK_PHOSPHATE: Factor w/ 84 levels "?","100","102",..: 19 84 51 1 83 1 1 1 1 73 ...
##  $ SGOT         : Factor w/ 85 levels "?","100","101",..: 54 48 62 31 44 1 1 60 8 46 ...
##  $ ALBUMIN      : Factor w/ 30 levels "?","2.1","2.2",..: 13 18 18 18 18 1 1 22 17 22 ...
##  $ PROTIME      : Factor w/ 45 levels "?","0","100",..: 1 1 42 1 38 1 1 1 1 44 ...
##  $ HISTOLOGY    : int  1 1 1 1 1 1 1 1 1 1 ...
```

Canviem el tipus de les variables numèriques que havien estat llegides com a factor.

```
quantattrs <- c("AGE","BILIRUBIN","ALK_PHOSPHATE","SGOT","ALBUMIN","PROTIME")
for (i in quantattrs){
  dhep[,i] <- as.numeric(dhep[,i])
}
```

Canviem el tipus de les variables factor que han estat llegides incorrectament.

```
factattrs <- c("Class","AGE","SEX")
for (i in factattrs){
  dhep[,i] <- as.numeric(dhep[,i])
}
```

Definim dos vectors amb les variables categòriques i numèriques.

```
quantattrs <- c("AGE","BILIRUBIN","ALK_PHOSPHATE","SGOT","ALBUMIN","PROTIME")
catattrs <- c("Class","SEX","STEROID","ANTIVIRALS","FATIGUE","MALAISE","ANOREXIA",
              "LIVER_BIG","LIVER_FIRM","SPLEEN_PAL","SPIDERS","ASCITES","VARICES","HISTOLOGY")
```

3.1 Valors desconeguts

En la primera insepecció hem pogut veure com hi ha valors desconeguts representats amb el signe interrogant. Mirarem quines columnes tenen valors desconeguts.

```
colSums(dhep == "?")
```

```
##         Class           AGE           SEX       STEROID    ANTIVIRALS
##             0             0             0             1             0
##       FATIGUE       MALAISE      ANOREXIA     LIVER_BIG    LIVER_FIRM
##             1             1             1            10            11
##    SPLEEN_PAL       SPIDERS       ASCITES       VARICES     BILIRUBIN
##             5             5             5             5             0
## ALK_PHOSPHATE          SGOT       ALBUMIN       PROTIME     HISTOLOGY
##             0             0             0             0             0
```

Mirem si hi ha algun valor NA. Podem veure que no.

```
colSums(is.na(dhep))
```

```
##         Class           AGE           SEX       STEROID    ANTIVIRALS
##             0             0             0             0             0
##       FATIGUE       MALAISE      ANOREXIA     LIVER_BIG    LIVER_FIRM
##             0             0             0             0             0
##    SPLEEN_PAL       SPIDERS       ASCITES       VARICES     BILIRUBIN
##             0             0             0             0             0
## ALK_PHOSPHATE          SGOT       ALBUMIN       PROTIME     HISTOLOGY
##             0             0             0             0             0
```

Pel que fa als valors quantitatius, podem veure que cap valor és 0.

```
colSums(dhep[,quantattrs] == 0)
```

```
##           AGE     BILIRUBIN ALK_PHOSPHATE          SGOT       ALBUMIN
##             0             0             0             0             0
##       PROTIME
##             0
```

Canviarem el valor de l'interrogant per NA. Guardem en un vector tots els atributs que contenen valors desconeguts i seguidament apliquem el canvi.

```
missingattrs <- names(which(colSums(dhep == "?") > 0))

for (m in missingattrs){
  dhep[which(dhep[,m] == '?'),m] <- NA
}
```

El primer que farem serà imputar els valors perduts que tenim al dataset, com que aquest dataset no és excessivament gran preferim aproximar els valors en comptes d'eliminar les observacions que compten amb mising values és a dir, assumirem un petit grau d'error a canvi de mantenir més observacions. A l'hora d'aplicar algorismes supervisats és interessant tenir grans datasets ja que es poden dividir en el dataset de train i el de test. En el primer apartat d'anàlisi exploratori hem comprovat com hi ha diversos atributs que tenen valors perduts. Per tal d'imputar els valors perduts utilitzarem el mètode missForest ja que últimament està guanyant popularitat i s'utilitza amb variables mixtes. Encara que el kNN és un dels més populars aquest, és molt sensible a la k que es tria.

Imputem els valors de les variables. Carreguem la llibreria i apliquem la funció.

```
library("missForest")
```

```
## Loading required package: randomForest

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

## Loading required package: foreach

## Loading required package: itertools

## Loading required package: iterators
```

```
mf.res <- missForest(dhep, variablewise = TRUE)
```

```
##   missForest iteration 1 in progress...done!
##   missForest iteration 2 in progress...done!
##   missForest iteration 3 in progress...done!
```

Podem veure com tots els valors NA han desaparegut, han estat imputats. També podem obtenir informació sobre l'error, es pot veure com les columnes que no tenien cap valor per imputar tenen un error de 0, les columnes on s'han imputat valors presenten els seus corresponents errors, això pot afectar a l'estudi ja que els valors no són reals simplement són suposicions.

```
dhep <- mf.res$ximp
colSums(is.na(dhep))
```

```
##         Class           AGE           SEX       STEROID    ANTIVIRALS
##             0             0             0             0             0
##       FATIGUE      MALAISE     ANOREXIA     LIVER_BIG    LIVER_FIRM
##             0             0             0             0             0
##    SPLEEN_PAL       SPIDERS       ASCITES       VARICES     BILIRUBIN
##             0             0             0             0             0
## ALK_PHOSPHATE          SGOT       ALBUMIN       PROTIME     HISTOLOGY
##             0             0             0             0             0
```

```
mf.res$OOBerror
```

```
##       MSE       MSE       MSE       PFC       MSE       PFC       PFC       PFC
## 0.0000000 0.0000000 0.0000000 0.3921569 0.0000000 0.2418301 0.1895425 0.1764706
##       PFC       PFC       PFC       PFC       PFC       PFC       MSE       MSE
## 0.1944444 0.2797203 0.2751678 0.2617450 0.1208054 0.1006711 0.0000000 0.0000000
##       MSE       MSE       MSE       MSE
## 0.0000000 0.0000000 0.0000000 0.0000000
```

3.2 Valors extrems

Mostrem una primera descripció estadística.

```
summary(dhep)
```

```
##      Class            AGE              SEX        STEROID   ANTIVIRALS
##  Min.   :1.000   Min.   : 7.00   Min.   :1.000   ?: 0    Min.   :1.000
##  1st Qu.:2.000   1st Qu.:32.00   1st Qu.:1.000   1:75    1st Qu.:2.000
##  Median :2.000   Median :39.00   Median :1.000   2:79    Median :2.000
##  Mean   :1.792   Mean   :41.27   Mean   :1.097           Mean   :1.844
##  3rd Qu.:2.000   3rd Qu.:50.00   3rd Qu.:1.000           3rd Qu.:2.000
##  Max.   :2.000   Max.   :78.00   Max.   :2.000           Max.   :2.000
##  FATIGUE MALAISE ANOREXIA LIVER_BIG LIVER_FIRM SPLEEN_PAL SPIDERS ASCITES
##  ?:  0   ?: 0    ?:  0    ?:  0     ?: 0       ?:  0      ?:  0   ?:  0
##  1:100   1:61    1: 32    1: 25     1:63       1: 30      1: 52   1: 21
##  2: 54   2:93    2:122    2:129     2:91       2:124      2:102   2:133
##
##
##
##  VARICES   BILIRUBIN      ALK_PHOSPHATE        SGOT          ALBUMIN
##  ?:  0   Min.   : 1.00   Min.   : 1.00   Min.   : 1.00   Min.   : 1.00
##  1: 18   1st Qu.: 6.00   1st Qu.: 8.25   1st Qu.:29.25   1st Qu.: 9.25
##  2:136   Median : 9.00   Median :40.50   Median :47.50   Median :17.00
##          Mean   :11.45   Mean   :40.13   Mean   :46.60   Mean   :14.70
##          3rd Qu.:14.00   3rd Qu.:71.00   3rd Qu.:66.75   3rd Qu.:20.00
##          Max.   :35.00   Max.   :84.00   Max.   :85.00   Max.   :30.00
##     PROTIME        HISTOLOGY
##  Min.   : 1.00   Min.   :1.000
##  1st Qu.: 1.00   1st Qu.:1.000
##  Median : 3.00   Median :1.000
##  Mean   :13.36   Mean   :1.455
##  3rd Qu.:25.75   3rd Qu.:2.000
##  Max.   :45.00   Max.   :2.000
```

Analitzem els valors extrems, comprovem quins atributs de tipus quantitatiu presenten possibles valors extrems. Com veiem els atributs AGE i BILIRUBIN tenen alguns valors extrems, en el proper apartat valorarem si realment són tan extrems i prendrem decissions sobre com tractar-ho.

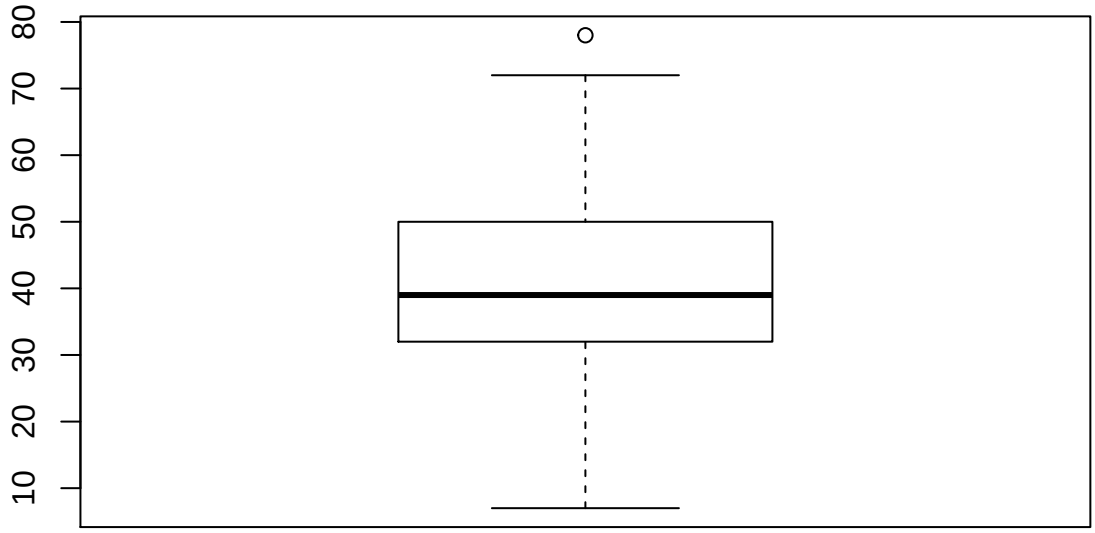
```
for (col in quantattrs){
  cat("col:",col,"Outliers:",length(boxplot.stats(dhep[,col])$out),"\n")
}
```

```
## col: AGE Outliers: 1
## col: BILIRUBIN Outliers: 13
## col: ALK_PHOSPHATE Outliers: 0
## col: SGOT Outliers: 0
## col: ALBUMIN Outliers: 0
## col: PROTIME Outliers: 0
```

Procedim a analitzar cadascun dels valors que es podria considerar outlier.

En aquest primer cas la variable AGE representa l'edat, un valor de 78 anys no es pot considerar un outiler, mantindrem aquest valor.

```
boxplot(dhep$AGE)
```



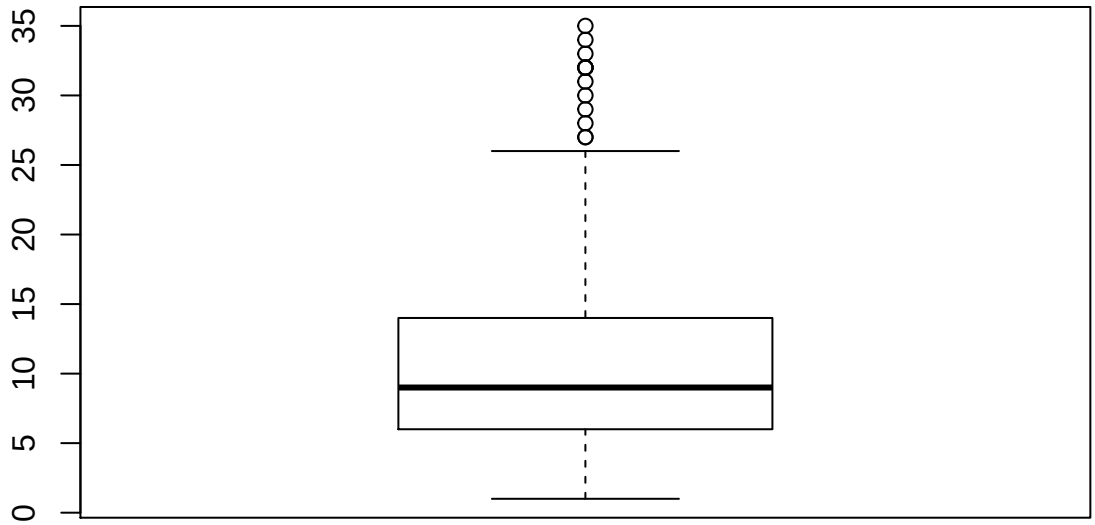

```
boxplot.stats(dhep$AGE)$out
```

```
## [1] 78
```

En aquest segon cas el nivell de bilirubina és més complex. Cal fer recerca sobre quins nivells màxims i mínims són possibles en pacient d'hepatitis. Segons les fonts consultades els valors més elevats de bilirubina indiquen problemes més greus, en propers apartats buscarem la relació de la bilirubina amb la vida o mort del pacient utilitzant testos d'estadística inferencial. Els nivells normals de bilirubina varien en un rang d'1 fins a 1.2 (mg/dL), a partir dels 2(mg/dL) la pell agafa un color groguenc. Sembla que aquests són valors molt elevats però dintre un rang possible, ja que s'indica que a partir dels 30 mg/dL el pacient es troba en estat molt crític.

Fonts:
- https://emedicine.medscape.com/article/775507-workup
- https://www.medicalnewstoday.com/articles/315086.php#1

```
boxplot(dhep$BILIRUBIN)
```




```
boxplot.stats(dhep$BILIRUBIN)$out
```

```
##  [1] 32 28 30 32 33 32 27 27 32 35 29 31 34
```

```
summary(dhep$BILIRUBIN)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    6.00    9.00   11.45   14.00   35.00
```

Com hem vist els valors dels nivells de les variables categòriques no són explicatius, actualment es troben representats amb valors numèrics, canviarem aquests valors, ja que pot ser útil quan apliquem futurs algorismes i vulguem extreure conclusions.

```
dhep[dhep$Class == 1,"Class"] <- 'Die'
dhep[dhep$Class == 2,"Class"] <- 'Live'
```

```
dhep[dhep$SEX == 1,"SEX"] <- 'Male'
dhep[dhep$SEX == 2,"SEX"] <- 'Female'

dhep$SEX <- as.factor(dhep$SEX)
dhep$Class <- as.factor(dhep$Class)
```

```
# Transformem a caràcter. canviem el valors dels nivells i tornem a transformar a factor, així eliminem
for (i in catattrs[- which (catattrs %in% list("Class","SEX"))] ){
  dhep[,i] <- as.character(dhep[,i])

  dhep[ dhep[,i] == 1,i] <- 'Yes'
  dhep[ dhep[,i] == 2,i] <- 'No'

  dhep[,i] <- as.factor(dhep[,i])
}
```

4 Anàlisi de les dades

4.1 Selecció dels grups a analitzar

En aquest apartat prepararem grups que poden ser d'interés per tal d'analitzar o comparar. En futurs apartats els utilitzarem per tal d'extreure conclusions.

Pacients que han mort d'hepatitis.

```
dhep.die <- dhep[dhep$Class == "Die",]
```

Pacients que han soberviscut a l'hepatitis.

```
dhep.live <- dhep[dhep$Class == "Live",]
```

4.2 Proves estadístiques

4.2.1 Comprovació de normalitat

Aplicarem un test de Shapiro-Wilk a tots els atributs quantitatius per tal de veure en quins casos podem assumir normalitat i en quins no. Utilitzarem aquest test, ja que es considera el més robust per tal de fer la prova de normalitat. Les hipòtesis que planteja el test són les següents:

- **H0**: La mostra prové d'una població amb distribució normal.
- **H1**: La mostra no prové d'una població amb distribució normal.

Si rebutgem la hipòtesi nul · la no podrem dir que les mostres vinguin d'una població amb distribució normal, d'altra forma si acceptem la hipòtsi nul · la és a dir, el p-valor és superior a 0.05, podrem assumir normalitat.

Carreguem una llibreria útil per fer taules.

```
library("kableExtra")
```

Com podem veure l'única variable que segueix una distribució normal és l'edat "AGE". Prendrem una conclusió conservadora i no assumirem la normalitat de les altres variables (no aplicarem el teorema del límit central).

```
mat <- NULL
alpha <- 0.05
for (atr in quantattrs){
  pv <- shapiro.test(dhep[,atr])$p.value
  mat <- rbind(mat,
               c(atr,pv,
                 ifelse(pv > alpha,"Yes","No"))
  )
}

colnames(mat) <- c("Variable","P-Value", "Dist. Normal")
kable(mat) %>% kable_styling()
```

Variable	P-Value	Dist. Normal
AGE	0.103350455455303	Yes
BILIRUBIN	9.51870650701998e-12	No
ALK_PHOSPHATE	4.00023337886688e-10	No
SGOT	0.000154910327946493	No
ALBUMIN	1.2353582877028e-06	No
PROTIME	1.72368503302654e-13	No

4.2.2 Testos sobre variables quantitatives

4.2.3 Edat

En aquest cas utilitzarem un test paramètric, hem pogut assumir normalitat després d'aplicar el test de shapiro wilk. Utilitzarem doncs el test de t-student.
Com podem veure el p-valor obtingut és inferior a 0.05, rebutgem la hipòtesi nul · la i acceptem l'alternativa. La mitjana d'edat dels pacients que moren és superior a l'edat mitjana dels que viuen.

```
t.test(dhep.die$AGE,dhep.live$AGE,alternative = 'greater')
```

```
##
##  Welch Two Sample t-test
##
## data:  dhep.die$AGE and dhep.live$AGE
## t = 3.1862, df = 61.099, p-value = 0.001136
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  3.195834      Inf
## sample estimates:
## mean of x mean of y
##  46.59375  39.87705
```

4.2.4 Bilirubina

Investigarem si la mitjana de bilirubina dels pacients de la classe DIE és superior als pacients de la classe LIVE. Com a hipòtesi nul · la tindrem que les dues classes tenen nivells de bilirubina iguals, com a alternativa

direm que la mitjana de bilirubina dels pacients que moren és superior als que viuen. Utilitzarem un test no paramètric, el wilcox test. Com hem comprovat el test de shapiro wilk no ens ha assegurat que puguem assumir normalitat. Segons el p-value obtingut rebutgem la hipòtesi nul · la i acceptem l'alternativa, ja que és menor a 0.05. Podem dir que la mitjana de bilirubina dels pacients de classe DIE és superior a la mitjana dels pacients de classe LIVE.

```
wilcox.test(dhep.die$BILIRUBIN,dhep.live$BILIRUBIN, paired = FALSE, alternative = 'greater')
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  dhep.die$BILIRUBIN and dhep.live$BILIRUBIN
## W = 2834, p-value = 3.957e-05
## alternative hypothesis: true location shift is greater than 0
```

4.2.5 Variables qualitatives que influeixen més en si el pacient soberviurà o morirà

En aquest apartat intentarem descobrir quines variables qualitatives influeixen més sobre si el pacient viurà o morirà. Utilitzarem el test de chi quadrat. Aquest test ens indica si dues variables són depenents o independents, quan rebutgem la hipòtesi nul · la direm que hi ha una relació entre les dues variables. Comprovarem si hi ha relació entre totes les variables categòriques i la classe, és a dir la variable que ens indica si el pacient viu o mor.

- **H0** : No hi ha relació entre les variables X i Y.
- **H1** : Hi ha relació entre les variables X i Y.

Aplicarem el test de chi quadrat a totes les variables categòriques. Com podem veure a la taula les variables FATIGUE, MALAISE, ANOREXIA, SPLEEN_PAL, SPIDERS, ASCITES, VARICES, HISTOLOGY tenen relació amb la classe a la que pertany el pacient. Hem utilitzat un alpha de 0.05 és a dir, en els casos on el p-valor obtingut és inferior a 0.05 rebutjem la Hipòtesi nul · la i determinem que hi ha relació.

```
mat <- NULL
alpha <- 0.05
for (atr in catattrs[-which(catattrs %in% "Class")]){
  freq.table <- table(dhep$Class,dhep[,atr])

  pv <- chisq.test(freq.table)$p.value
  mat <- rbind(mat,
               c(atr,pv,
                 ifelse(pv < alpha,"Yes","No"))
  )
}

colnames(mat) <- c("Variable","P-Value", "Relació")
kable(mat) %>% kable_styling()
```

Variable	P-Value	Relació
SEX	0.0796189439950327	No
STEROID	0.119734506305366	No
ANTIVIRALS	0.173247546792783	No
FATIGUE	0.000283623525807864	Yes
MALAISE	6.61626777671667e-05	Yes
ANOREXIA	0.162879415955493	No
LIVER_BIG	0.708242325065387	No
LIVER_FIRM	0.330469172143881	No
SPLEEN_PAL	0.00826818418064086	Yes
SPIDERS	9.036527779379e-07	Yes
ASCITES	4.45314447847816e-09	Yes
VARICES	2.93064976157634e-05	Yes
HISTOLOGY	7.16841274432988e-05	Yes

4.3 Models supervisats

En aquest apartat aplicarem la regressió logística per tal de crear un model de classificació supervisat.

```
library("pROC")
```

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

##
## Attaching package: 'pROC'

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

```
library("caret")
```

```
## Loading required package: lattice

## Loading required package: ggplot2

##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:randomForest':
##
##     margin
```

```
library("ggplot2")
```

Recodifiquem l'ordre dels nivells, ja que, entendrem Die com al cas positiu i Live com el cas negatiu seguint amb la lògica de cas positiu quan es detecta una malaltia.

```
dhep$Class <- relevel(dhep$Class,"Live")
```

4.4 Preparació del conjunt de train i test

```
library("rminer") # Carreguem la llibreria que utilitzarem per dividir el dataset.
```

Per tal de poder avaluar els models logístics haurem de dividir el dataset en dos, train i test. Per tal de dividir el conjunt utilitzarem un mètode simple anomenat mètode d'exclusió (holdout), dividirem el conjunt total en 2/3 per al train i 1/3 per al test.

```
h<-holdout(dhep$Class,ratio=2/3,mode="random", seed = set.seed(6))
data_train<-dhep[h$tr,]
data_test<-dhep[h$ts,]
```

Podem veure com la proporció de morts i vius es manté (la difèrencia és molt petita).

```
prop.train <- sum(data_train$Class == "Live")/sum(data_train$Class == "Die")
prop.ori <- sum(dhep$Class == "Live")/sum(dhep$Class == "Die")
prop.test <- sum(data_test$Class == "Live")/sum(data_test$Class == "Die")

c(prop.ori,prop.train,prop.test)
```

```
## [1] 3.812500 4.100000 3.333333
```

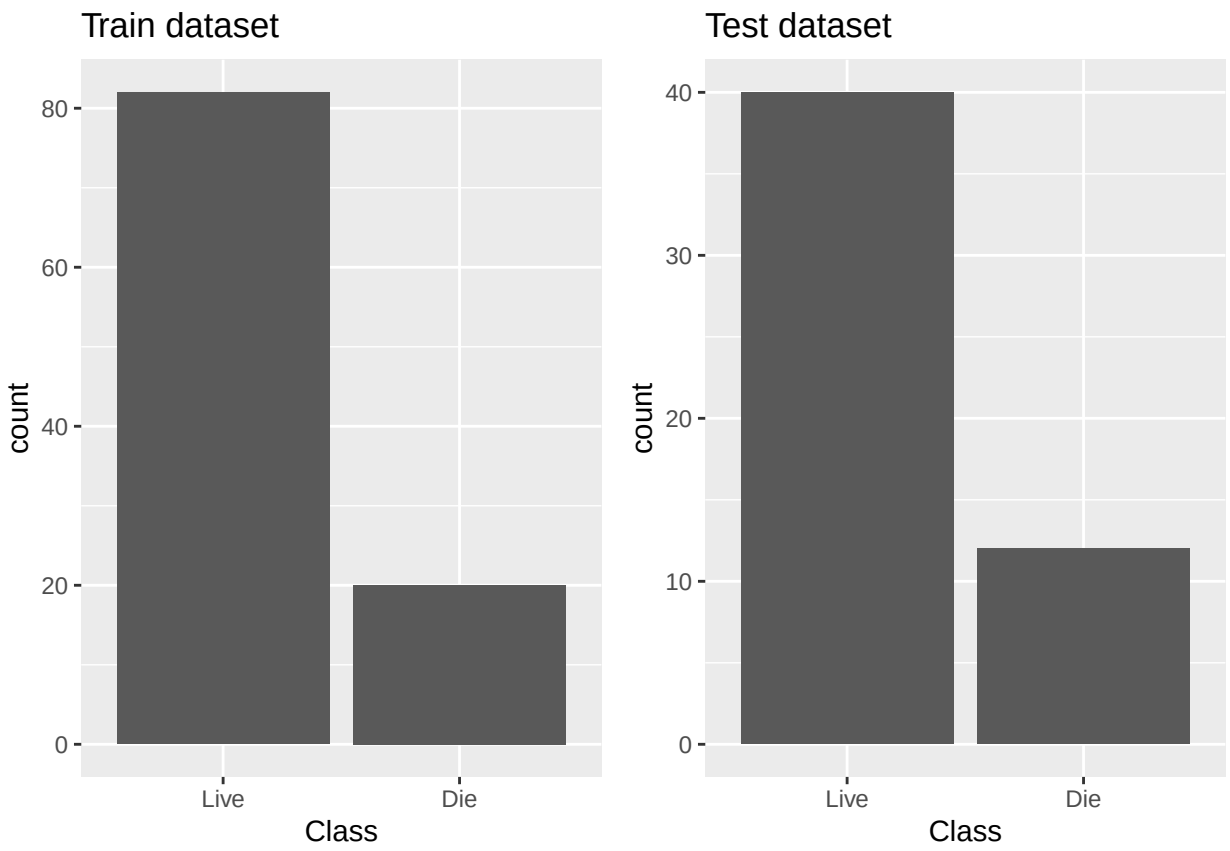
Podem veure com es distribueixen les classes en els dos datasets de forma gràfica.

```
library("gridExtra")
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:randomForest':
##
##     combine
```

```
p1 <- ggplot(data = data_train, aes(x = Class)) + geom_bar() + ggtitle("Train dataset")
p2 <- ggplot(data = data_test, aes(x = Class)) + geom_bar() + ggtitle("Test dataset")
grid.arrange(p1,p2, nrow = 1)
```



4.4.1 Regressió logística utilitzant únicament variables quantitatives.

Preparem un model amb les variables quantitatives que hem determinat que tenen relació amb Class. En el resum del model creat podem veure el p-valor del test de Wald que ens indica si el coeficient de la variable és significatiu o no pel model. Això serà interessant a l'hora de buscar un model més reduït que segueixi sent un bon classificador.

```
log.quant <- glm(data = dhep, Class~ AGE + BILIRUBIN+ ALK_PHOSPHATE + SGOT+ ALBUMIN+ PROTIME,
                 family = binomial(link = "logit") )
summary(log.quant)
```

```
##
## Call:
## glm(formula = Class ~ AGE + BILIRUBIN + ALK_PHOSPHATE + SGOT +
##     ALBUMIN + PROTIME, family = binomial(link = "logit"), data = dhep)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4099  -0.5397  -0.2992  -0.1906   2.6776
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.916611   1.130479  -1.695 0.090000 .
## AGE            0.031282   0.019557   1.600 0.109707
## BILIRUBIN      0.105696   0.027781   3.805 0.000142 ***
```

```
## ALK_PHOSPHATE -0.001488   0.008949  -0.166 0.867947
## SGOT          -0.008361   0.010196  -0.820 0.412161
## ALBUMIN       -0.143233   0.041110  -3.484 0.000494 ***
## PROTIME        0.003265   0.020832   0.157 0.875474
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 157.39  on 153  degrees of freedom
## Residual deviance: 109.86  on 147  degrees of freedom
## AIC: 123.86
##
## Number of Fisher Scoring iterations: 5
```

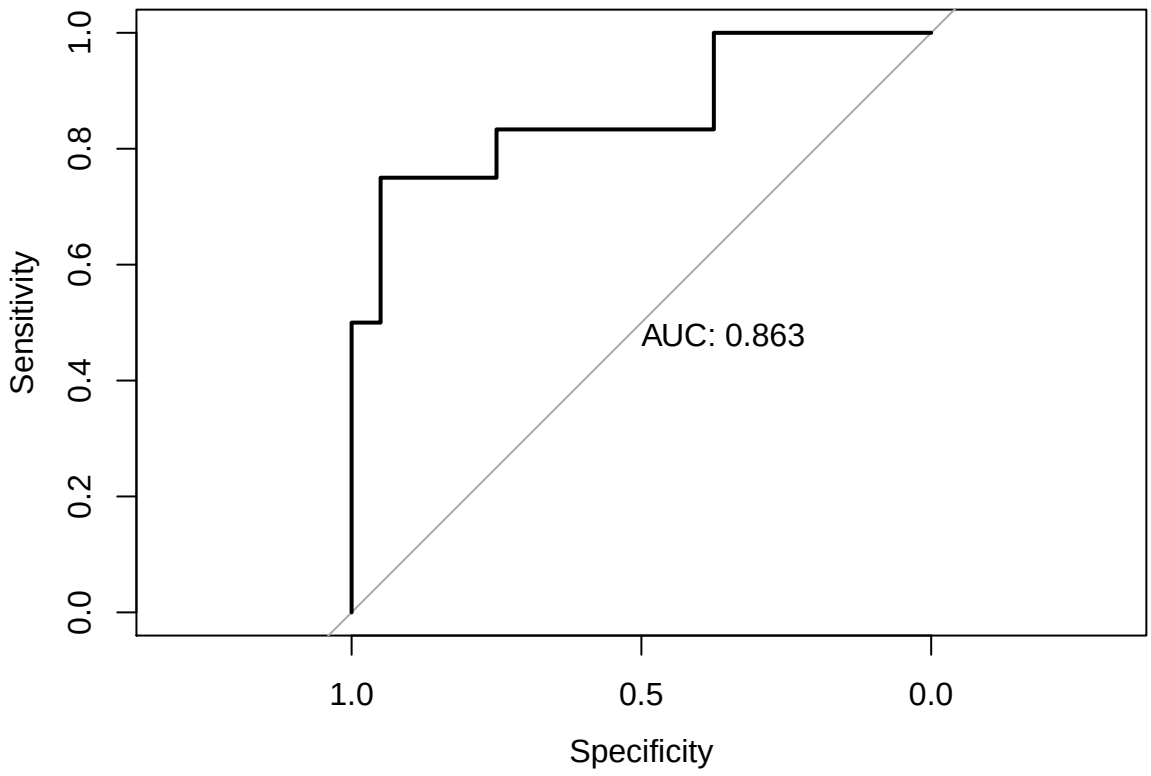
Dibuixem la corva de ROC. Podem veure com utilitzant simplement els atributs quantitatius obtenim un model bastant bo si ens guiem per l'AUC.

```
prediction.quant <- predict(log.quant, data_test, type = "response")

pROC.quant <- roc(data_test$Class, prediction.quant, plot = TRUE, print.auc = TRUE)
```

```
## Setting levels: control = Live, case = Die

## Setting direction: controls < cases
```



```r
coords(pROC.quant, x = "best",
                      ret = c("specificity","sensitivity","accuracy"),
                      transpose = F)
```

```
##      specificity sensitivity  accuracy
## best        0.95        0.75 0.9038462
```

4.4.2 Regressió logística utilitzant únicament variables qualitatives

En aquest apartat utilitzarem les variables qualitatives que hem determinat, tenen efecte sobre si el pacient viurà o morirà utilitzant la prova de chi quadrat. Observem el p-valor que s'obté, aquest correspon al test de Wald i ens indica si el coeficient és significativament diferent de 0. En aquest cas eliminarem les variables amb un p-valor superior a 0.10.

```r
log.qual <- glm(data = dhep, Class~ FATIGUE+MALAISE+SPLEEN_PAL+SPIDERS+ASCITES+VARICES+HISTOLOGY,
                family = binomial(link = "logit") )
summary(log.qual)
```

```
##
## Call:
## glm(formula = Class ~ FATIGUE + MALAISE + SPLEEN_PAL + SPIDERS +
##     ASCITES + VARICES + HISTOLOGY, family = binomial(link = "logit"),
##     data = dhep)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0079  -0.4352  -0.2686  -0.1971   2.5885
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -3.3144     0.8234  -4.025 5.69e-05 ***
## FATIGUEYes      0.6268     0.9552   0.656  0.51172
## MALAISEYes      0.9953     0.6451   1.543  0.12283
## SPLEEN_PALYes   0.8701     0.5901   1.474  0.14036
## SPIDERSYes      1.0067     0.5559   1.811  0.07012 .
## ASCITESYes      1.6882     0.6333   2.666  0.00768 **
## VARICESYes      0.7418     0.6630   1.119  0.26317
## HISTOLOGYYes   -0.6169     0.5961  -1.035  0.30074
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 157.39  on 153  degrees of freedom
## Residual deviance: 102.17  on 146  degrees of freedom
## AIC: 118.17
##
## Number of Fisher Scoring iterations: 6
```

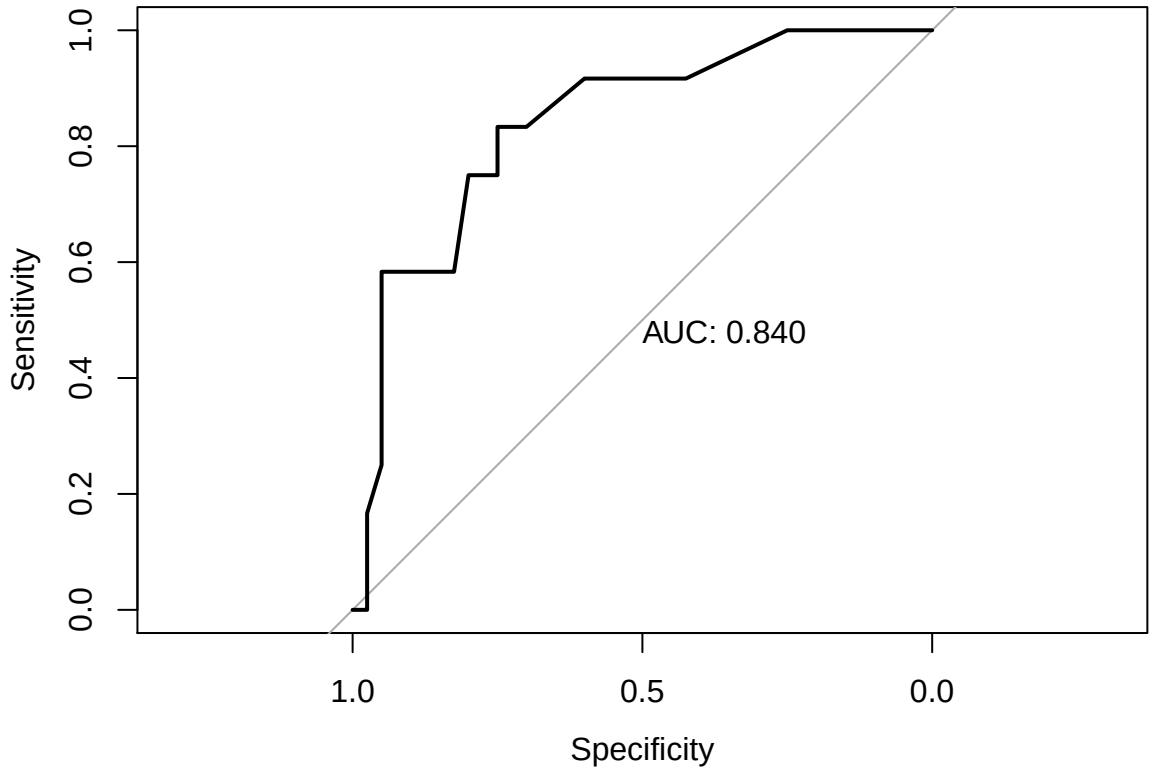
Sembla ser que també obtenim un bon model amb aquestes variables, el valor de AUC segueix sent molt elevat encara que lleugerament inferior al model anterior.

```
prediction.qual <- predict(log.qual, data_test, type = "response")

pROC.qual <- roc(data_test$Class, prediction.qual, plot = TRUE, print.auc = TRUE)
```

```
## Setting levels: control = Live, case = Die

## Setting direction: controls < cases
```




```
coords(pROC.qual, x = "best",
                      ret = c("specificity","sensitivity","accuracy"),
                      transpose = F)
```

```
##      specificity sensitivity  accuracy
## best        0.75   0.8333333 0.7692308
```

4.4.2.1 Reducció del model

Preparem el model reduït, hem eliminat les variables que tenien un p-valor superior a 0.10 al test de Wald i ens hem quedat amb SPIDERS i ASCITES.

```
log.qual.red <- glm(data = dhep, Class~SPIDERS+ASCITES,
                    family = binomial(link = "logit") )
summary(log.qual.red)
```

```
##
## Call:
## glm(formula = Class ~ SPIDERS + ASCITES, family = binomial(link = "logit"),
##     data = dhep)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8575  -0.3693  -0.3693  -0.3693   2.3321
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.6512     0.3862  -6.864 6.69e-12 ***
## SPIDERSYes    1.7483     0.4875   3.587 0.000335 ***
## ASCITESYes    2.4319     0.5884   4.133 3.58e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 157.39  on 153  degrees of freedom
## Residual deviance: 113.40  on 151  degrees of freedom
## AIC: 119.4
##
## Number of Fisher Scoring iterations: 5
```

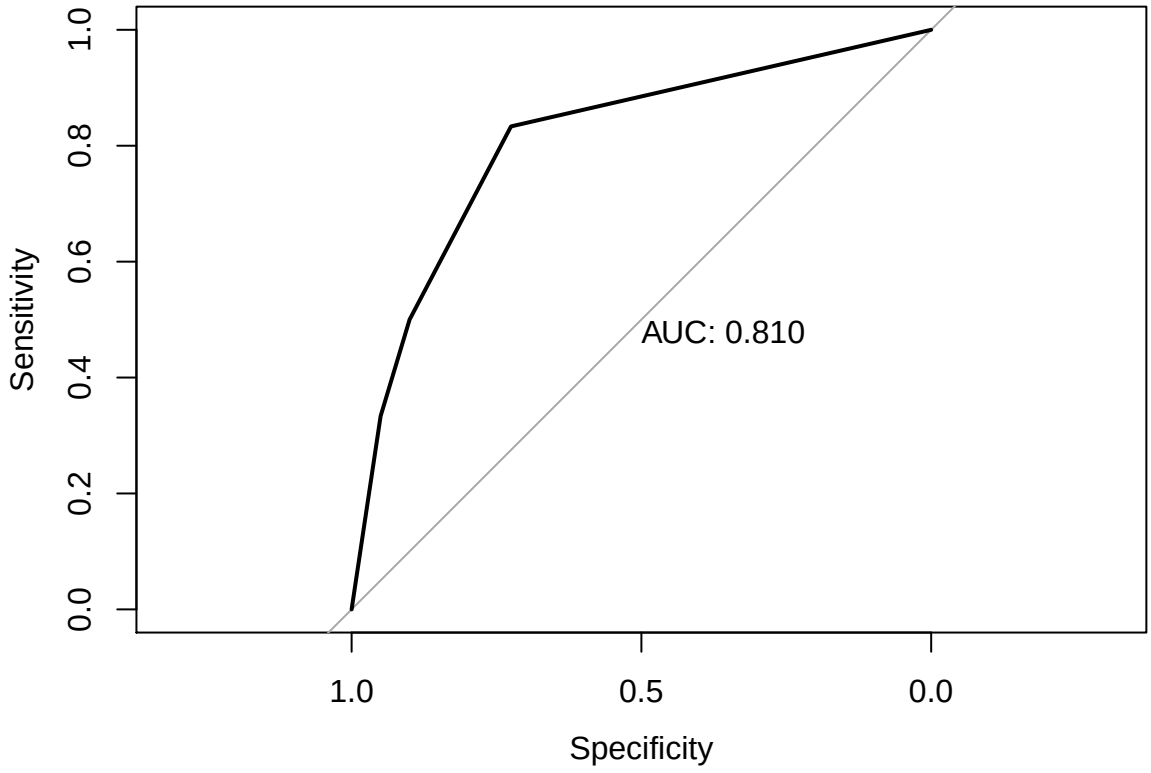
Obtenim uns molt bons resultats, l'AUC s'ha mantingut molt semblant al model on utilitzem tots els atributs però, en aquest cas hem reduït el model utilitzant només dues variables. Naturalment l'AUC ha disminuit minimament, ja que les altres variables aportaven una mica d'informació encara que no fos extremadament significativa. Amb aquesta versió reduïda hem aconseguit un model molt més petit que amb un AUC gairebé igual a l'anterior.

```
prediction.qual.red <- predict(log.qual.red, data_test, type = "response")

pROC.qual.red <- roc(data_test$Class, prediction.qual.red, plot = TRUE, print.auc = TRUE)
```

```
## Setting levels: control = Live, case = Die
```

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




```
coords(pROC.qual.red, x = "best",
                      ret = c("specificity","sensitivity","accuracy"),
                      transpose = F)
```

```
##      specificity sensitivity accuracy
## best       0.725   0.8333333     0.75
```

4.4.3 Regressió logística utilitzant variables mixtes

En aquest últim model utilitzarem les variables tant qualitatives com quantitatives que hem determinat com a més explicatives és a dir, les variables que han tingut més efecte en els models. Ens fixem en els resultats obtinguts del resum del model, veiem que totes les variables aporten, segons el test de Wald els coeficients que obtenim són significativament diferents de 0.

```
log.mixt<- glm(data = dhep, Class~SPIDERS+ASCITES+BILIRUBIN+ALBUMIN,
                family = binomial(link = "logit"))
summary(log.mixt)
```

```
##
## Call:
## glm(formula = Class ~ SPIDERS + ASCITES + BILIRUBIN + ALBUMIN,
##     family = binomial(link = "logit"), data = dhep)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -1.6758  -0.4979  -0.2057  -0.1363   2.8150
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.93419    0.68635  -2.818 0.004831 **
## SPIDERSYes   1.77657    0.56298   3.156 0.001601 **
## ASCITESYes   1.67104    0.63659   2.625 0.008665 **
## BILIRUBIN    0.08176    0.02969   2.753 0.005897 **
## ALBUMIN     -0.14088    0.04113  -3.426 0.000614 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 157.393  on 153  degrees of freedom
## Residual deviance:  90.992  on 149  degrees of freedom
## AIC: 100.99
##
## Number of Fisher Scoring iterations: 6
```

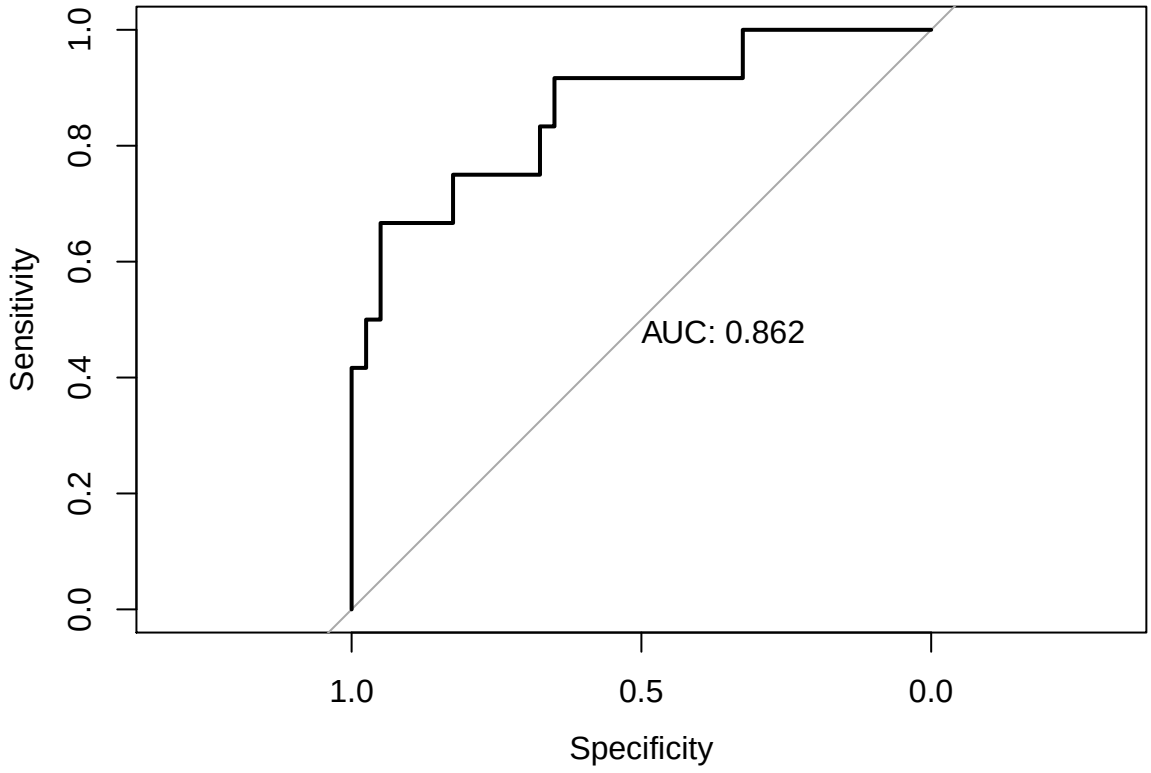
Podem veure com obtenim un molt bon resultat amb un AUC molt elevat. Hem aconseguit una bon model utilitzant variables mixtes.

```
prediction.mixt <- predict(log.mixt, data_test, type = "response")

pROC.mixt <- roc(data_test$Class, prediction.mixt, plot = TRUE, print.auc = TRUE)
```

```
## Setting levels: control = Live, case = Die
```

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




```
coords(pROC.mixt, x = "best",
                      ret = c("specificity","sensitivity","accuracy"),
                      transpose = F)
```

```
##      specificity sensitivity  accuracy
## best        0.95   0.6666667 0.8846154
```

4.4.4 Comparació dels models

Al llarg dels apartats anterior heme estat valorant els models unicament amb l'AUC però existeixen més mesures d'interés a l'hora de comparar models. En aquest apartat farem una comparativa dels models des de el punt de vista de sensibilitat, especificitat i exactitud.

```
# Vector amb els models
models <- NULL
models <- vector(mode="list", length=4)
models[[1]] <- pROC.quant
models[[2]] <- pROC.qual
models[[3]] <- pROC.qual.red
models[[4]] <- pROC.mixt

mat <- NULL
for (model in models){
  show(model)
```

```
  mat <- rbind(mat,
               coords(model, x = "best",
                      ret = c("sensitivity","specificity", "accuracy"),
                      best.method = "closest.topleft"
                      )
               )
}
```

```
##
## Call:
## roc.default(response = data_test$Class, predictor = prediction.quant,     plot = TRUE, print.auc = TR
##
## Data: prediction.quant in 40 controls (data_test$Class Live) < 12 cases (data_test$Class Die).
## Area under the curve: 0.8625
##
## Call:
## roc.default(response = data_test$Class, predictor = prediction.qual,     plot = TRUE, print.auc = TRU
##
## Data: prediction.qual in 40 controls (data_test$Class Live) < 12 cases (data_test$Class Die).
## Area under the curve: 0.8396
##
## Call:
## roc.default(response = data_test$Class, predictor = prediction.qual.red,     plot = TRUE, print.auc =
##
## Data: prediction.qual.red in 40 controls (data_test$Class Live) < 12 cases (data_test$Class Die).
## Area under the curve: 0.8104
##
## Call:
## roc.default(response = data_test$Class, predictor = prediction.mixt,     plot = TRUE, print.auc = TRU
##
## Data: prediction.mixt in 40 controls (data_test$Class Live) < 12 cases (data_test$Class Die).
## Area under the curve: 0.8625
```

```
rownames(mat) <- c("Quantitatius", "Qualitatius", "Qualitatius reduït", "Mixt")
```

La sensibilitat fa referència als casos classificats com a positius on realment són positius, en aquest cas serien els pacients que moriran classificats correctament com a pacients que moriran. L'especificitat és la taxa de casos classificats com a negatius que realment són negatius, en aquest cas els pacients classificats com que no moriran que realment no moren. Finalment l'exactitud fa referència al conjunt de registres classificats correctament. De la taula podem extreure diverses conclusions, per exemple quan comparem el model amb totes les variables qualitatives la pèrdua en les tres mesures és mínima, per tant tenim un model amb menys variables però segueix sent gairebé igual de bo a l'hora de classificar per tant, és millor, ja que és un model més petit, únicament format per variables que afecten en la classificació. Podem veure com el model mixt especificitat i exactitud als dos models qualitatius però no al que compta amb totes les variables quantitatives. Sembla que alguna de les variables quantitatives o varies en conjunt són una mica significatives, encara que ens hem quedat amb les que tenen més efecte.

```
kable(mat) %>% kable_styling()
```

	sensitivity	specificity	accuracy
Quantitatius	0.7500000	0.950	0.9038462
Qualitatius	0.8333333	0.750	0.7692308
Qualitatius reduït	0.8333333	0.725	0.7500000
Mixt	0.7500000	0.825	0.8076923