Tipologia i Cicle de Vida de les Dades: PRA2

Estudiant: David Gil del Rosal

1 Descripció del dataset

L'objectiu d'aquesta pràctica és el preprocessament i anàlsi preliminar del joc de dades "Heart Disease Dataset" de l'*UCI Machine Learning Repository* [1]. Aquest dataset recopila analítiques sobre pacients tractats a diversos centres mèdics amb l'objectiu de predir si han sigut diagnosticats amb una malaltia cardiovascular.

L'interès d'aquest joc de dades és que el seu anàlisi ajuda a conèixer quins factors alerten sobre la presència d'una malaltia coronària i això pot contribuïr al seu diagnòstic i prevenció. Aquests objectius són importants: segons l'Organització Mundial de la Salut les malalties cardiovasculars són la primera causa global de mortalitat [2].

Des del punt de vista del seu preprocessament aquest joc de dades és interessant perque presenta atributs quantitatius i qualitatius, així com valors perduts i extrems.

2 Integració i selecció de les dades a analitzar

El joc de dades complet consta de 4 fitxers corresponents a diversos hospitals i centres mèdics. Segons [1] les úniques dades usades a les recerques prèvies publicades han sigut les de la *Cleveland Clinic Foundation*, per la qual cosa són les que usarem a aquest treball.

El joc de dades original consta de 76 variables, però tots els estudis publicats han analitzat els 14 atributs més importants que es presenten a continuació. S'indica si són quantitatius (numèrics) o qualitatis (categòrics). Tots els atributs categòrics estàn codificats mitjançant números per als quals s'assenyala els nivells i el valor de referència que és el que presenta menys risc de malaltia cardiovascular i que, excepte se s'indica el contrari, ès el primer nivell.

Atribut	Descripció	Tipus
age	Edat en anys	Num.
sex	Sexe	Cat.: 0=dona, 1=home
cp	Tipus de dolor en el pit	Cat.: 1,2,3,4; 4=asimptomàtic
trestbps	Pressió de la sang en repòs en mm/Hg	Num.
chol	Sèrum de colesterol en mg/dl	Num.
fbs	Nivell de sucre en sang en dejuni > 120 mg/dl	Cat.: 0=no, 1=sí
restecg	Resultats de l'electrocardiograma en repòs	Cat.: 0,1,2; 0=normal
thalach	Velocitat màxima de pulsacions registrada	Cat.: 0,1,2,3; 0=normal
exang	Angina de pit induïda per exercici	Cat.: 0=no, 1=sí
oldpeak	Depressió en el segment ST de l'electrocardiograma	Num.
slope	Tipus de pendent del segment ST	Cat.: 1=avall, 2=pla, 3=amunt
ca	Venes majors acolorides amb fluoroscopi	Cat.: 0,1,2,3
thal	Defecte congènit de sang (talassèmia)	Cat.: 3=no,6=inactiu,7=actiu
num	Diagnòstic (valor a predir)	Cat.: 0=sa, 1-4=malalt

3 Neteja de les dades

Les dades són en un fitxer CSV delimitat per comes al lloc web de l'*UCI Machine Learning Repository* [1]. El següent codi R llegeix el fitxer i assigna el nóm dels atributs. Els valors buits estan codificats amb el caràcter "?" al fitxer:

```
data <- read.csv('processed.cleveland.data', header=FALSE,
                 sep=",", na.strings="?")
colnames(data) <- c('age','sex','cp','trestbps','chol','fbs',
                    'restecg','thalach','exang','oldpeak',
                    'slope','ca','thal','num')
```

El joc de dades conté 303 registres amb 14 variables. Totes s'han interpretat com numèriques ja que, com s'ha dit, les categòriques estan codificades mitjançant números:

```
str(data)
```

```
## 'data.frame':    303 obs. of  14 variables:
##  $ age     : num  63 67 67 37 41 56 62 57 63 53 ...
##  $ sex     : num  1 1 1 1 0 1 0 0 1 1 ...
##  $ cp      : num  1 4 4 3 2 2 4 4 4 4 ...
##  $ trestbps: num  145 160 120 130 130 120 140 120 130 140 ...
##  $ chol    : num  233 286 229 250 204 236 268 354 254 203 ...
##  $ fbs     : num  1 0 0 0 0 0 0 0 0 1 ...
##  $ restecg : num  2 2 2 0 2 0 2 0 2 2 ...
##  $ thalach : num  150 108 129 187 172 178 160 163 147 155 ...
##  $ exang   : num  0 1 1 0 0 0 0 1 0 1 ...
##  $ oldpeak : num  2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 ...
##  $ slope   : num  3 2 2 3 1 1 3 1 2 3 ...
##  $ ca      : num  0 3 2 0 0 0 2 0 1 0 ...
##  $ thal    : num  6 3 7 3 3 3 3 3 7 7 ...
##  $ num     : int  0 2 1 0 0 0 3 0 2 1 ...
```

Abans de proseguir l'anàlisi, farem dues tasques de preprocessament. Com que l'anàlisis es centrarà en detectar la presència o absència de malaltia i no el seu tipus, substituïrem la variable `num` (factor amb nivells 0 a 4) pel factor binari `disease` i convertirem la resta de variables categòriques en factors:

```
data$disease <- as.factor(ifelse(data$num == 0, 0, 1))
data$num <- NULL
cat_vars <- c('sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal')
for(cat_var in cat_vars) {
  data[,cat_var] <- as.factor(data[,cat_var])
}
str(data)
```

```
## 'data.frame':    303 obs. of  14 variables:
##  $ age     : num  63 67 67 37 41 56 62 57 63 53 ...
##  $ sex     : Factor w/ 2 levels "0","1": 2 2 2 2 1 2 1 1 2 2 ...
##  $ cp      : Factor w/ 4 levels "1","2","3","4": 1 4 4 3 2 2 4 4 4 4 ...
##  $ trestbps: num  145 160 120 130 130 120 140 120 130 140 ...
##  $ chol    : num  233 286 229 250 204 236 268 354 254 203 ...
##  $ fbs     : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 2 ...
##  $ restecg : Factor w/ 3 levels "0","1","2": 3 3 3 1 3 1 3 1 3 3 ...
##  $ thalach : num  150 108 129 187 172 178 160 163 147 155 ...
##  $ exang   : Factor w/ 2 levels "0","1": 1 2 2 1 1 1 1 2 1 2 ...
##  $ oldpeak : num  2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 ...
```

```
##  $ slope   : Factor w/ 3 levels "1","2","3": 3 2 2 3 1 1 3 1 2 3 ...
##  $ ca      : Factor w/ 4 levels "0","1","2","3": 1 4 3 1 1 1 3 1 2 1 ...
##  $ thal    : Factor w/ 3 levels "3","6","7": 2 1 3 1 1 1 1 1 3 3 ...
##  $ disease : Factor w/ 2 levels "0","1": 1 2 2 1 1 1 2 1 2 2 ...
```

3.1 Elements buits

El següent codi R usa la funció summary() per a mostrar els principals estadístics de les variables del joc de dades. També s'aprecia la presència de valors buits NA:

```
summary(data)
```

```
##       age        sex     cp        trestbps          chol       fbs
##  Min.   :29.00   0: 97   1: 23   Min.   : 94.0   Min.   :126.0   0:258
##  1st Qu.:48.00   1:206   2: 50   1st Qu.:120.0   1st Qu.:211.0   1: 45
##  Median :56.00           3: 86   Median :130.0   Median :241.0
##  Mean   :54.44           4:144   Mean   :131.7   Mean   :246.7
##  3rd Qu.:61.00                   3rd Qu.:140.0   3rd Qu.:275.0
##  Max.   :77.00                   Max.   :200.0   Max.   :564.0
##  restecg    thalach      exang      oldpeak     slope      ca
##  0:151   Min.   : 71.0   0:204   Min.   :0.00   1:142   0   :176
##  1:  4   1st Qu.:133.5   1: 99   1st Qu.:0.00   2:140   1   : 65
##  2:148   Median :153.0           Median :0.80   3: 21   2   : 38
##          Mean   :149.6           Mean   :1.04           3   : 20
##          3rd Qu.:166.0           3rd Qu.:1.60           NA's:  4
##          Max.   :202.0           Max.   :6.20
##    thal     disease
##  3   :166   0:164
##  6   : 18   1:139
##  7   :117
##  NA's:  2
##
##
```

Hi ha 6 registres amb valors buits: 4 per a l'atribut `ca` i 2 per a l'atribut `thal`. Per a imputar el seu valor podriem usar una mesura de tendència central (la moda atès que tots dos atributs són categòrics) o bé predir-lo emprant un algorisme de mineria de dades. Optarem per la darrera opció, imputant-los amb els 3 veïns més propers emprant la funció `kNN` de la llibreria `VIM`:

```
library(VIM)
data <- kNN(data, variable=c("ca","thal"), k=3, imp_var=FALSE)
colSums(is.na(data))
```

```
##      age      sex       cp trestbps     chol      fbs  restecg  thalach
##        0        0        0        0        0        0        0        0
##    exang  oldpeak    slope       ca     thal  disease
##        0        0        0        0        0        0
```

S'observa que ja no hi ha registres amb valors buits.

3.2 Valors extrems

Per a detectar els valors extrems (*extreme scores*) de les variables numèriques, usarem el criteri convencional de considerar com outliers els valors inferiors a Q1-1.5IQR o superiors que Q3+1.5IQR on Q1, Q3 són el primer i el tercer quartil de la distribució de valors de la variable corresponent, i IQR és el rang interquartílic.

El següent codi R implementa una funció que, donat el dataframe que conté el joc de dades i un vector amb el nóm de les variables a testejar, retorna un dataframe les files del qual corresponen a les variables amb outliers, indicant el número, percentatge de registres afectats i valors extrems.

```
get.outliers <- function(data, variables) {
  df <- data.frame("Variable", 0, 0, 0, 0, "Valors", stringsAsFactors=FALSE)
  colnames(df) <- c("Variable", "#Outliers", "%Outliers",
                    "Q1-1.5IQR", "Q3+1.5IQR", "Valors outliers")
  row <- 1
  for(variable in variables) {
    values <- data[,variable]
    q <- quantile(values)
    iqr <- IQR(values)
    outliers <- boxplot.stats(values)$out
    n_outliers <- length(outliers)
    pct_outliers <- n_outliers*100/nrow(data)
    if(n_outliers > 0) {
      df[row,] <- list(variable, n_outliers, pct_outliers,
                       q[2]-1.5*iqr, q[4]+1.5*iqr,
                       paste(outliers, sep='', collapse=','))
      row <- row + 1
    }
  }
  return(df)
}

num_vars <- c('age', 'trestbps', 'chol', 'thalach', 'oldpeak')   # variables numèriques
df_outliers <- get.outliers(data, num_vars)
df_outliers
```

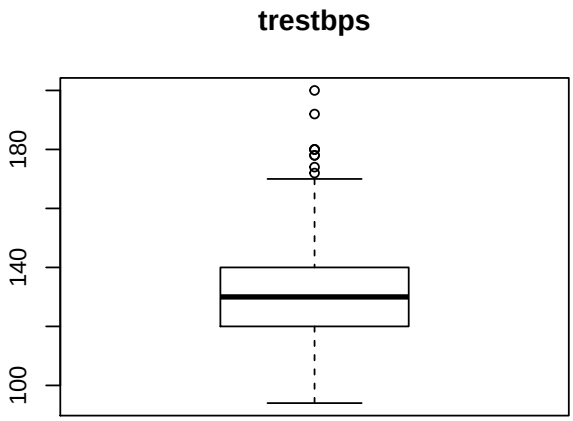
Variable	#Outliers	%Outliers	Q1-1.5IQR	Q3+1.5IQR	Valors outliers
trestbps	9	2.970297	90.00	170.00	172,180,200,174,178,192,180,178,180
chol	5	1.650165	115.00	371.00	417,407,564,394,409
thalach	1	0.330033	84.75	214.75	71
oldpeak	5	1.650165	-2.40	4.00	6.2,5.6,4.2,4.2,4.4

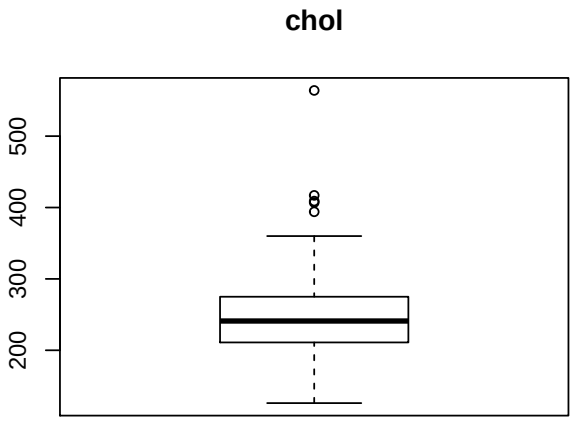
S'observa que les següents variables presenten valors extrems:

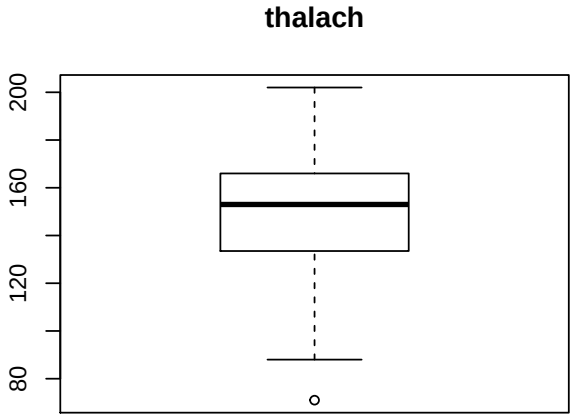
- trestbps: pressió de la sang en repós
- chol: nivell de colesterol
- thalach: velocitat màxima de pulsacions
- oldpeak: depressió en el segment ST de l'electrocardiograma

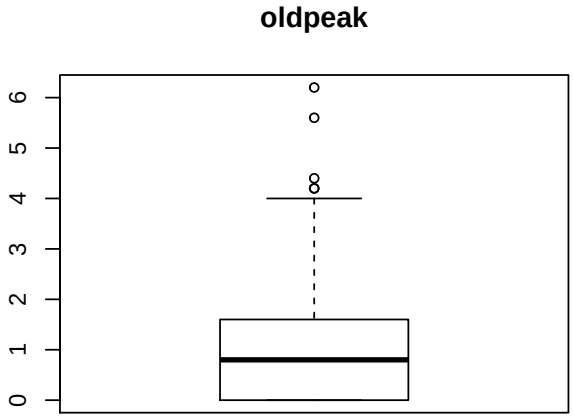
Els següents *boxplots* mostren els outliers detectats:

```
par(mfrow=c(2,2))
for(variable in df_outliers$Variable) {
  boxplot(data[,variable], main=variable)
}
```









Per a corregir-los podriem usar les mateixes tècniques que amb els valors perduts però com semblen valors legítims en el domini, optarem per no corregir-los deixant-los tal com són.

4 Anàlisi de les dades

4.1 Selecció

L'objectiu dels anàlisis serà determinar quins factors influeixen en la presència de malalties cardiovasculars i tractar de predir-les. Segons [1] totes les recerques publicades s'han basat en aquest criteri.

Ambdues classes estan prou equilibrades al joc de dades: hi ha 139 pacients malalts i 164 sans.

```
table(data$disease)
```

```
##
##   0   1
## 164 139
```

4.2 Normalitat i homocedasticitat

Per a comprovar la normalitat de les variables numèriques usarem el test de Shapiro-Wilk. És un contrast estadístic on la hipòtesi nul.la és que els valors provenen d'una distribució normal. El següent codi R aplica aquest test a les variables numèriques:

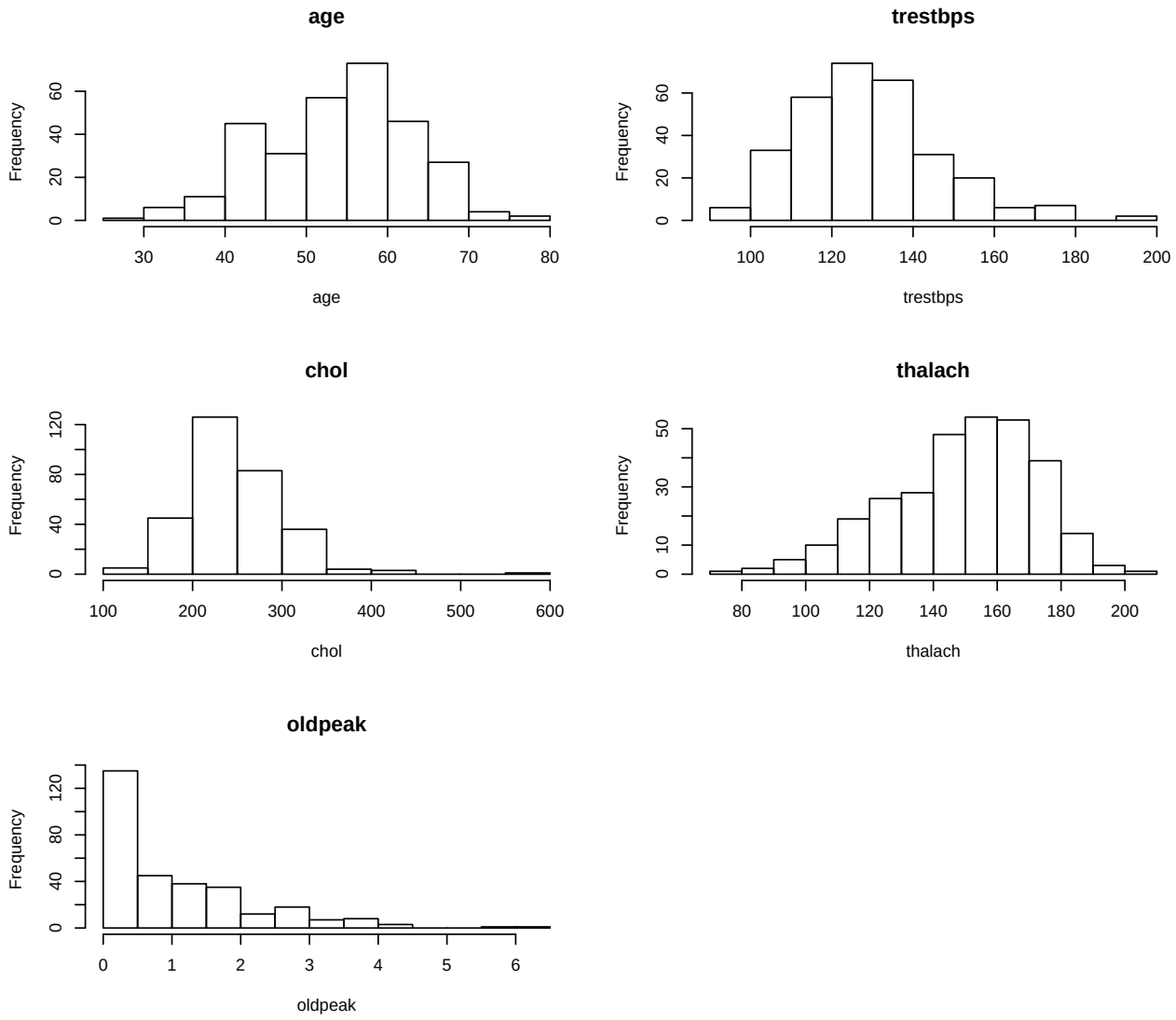
```
for(variable in num_vars) {
  results <- shapiro.test(data[,variable])
  print(paste("Variable", variable, "p-valor", results$p))
}
```

```
## [1] "Variable age p-valor 0.00606864230991071"
## [1] "Variable trestbps p-valor 1.80206438980931e-06"
## [1] "Variable chol p-valor 5.9115205752249e-09"
## [1] "Variable thalach p-valor 6.9964709790962e-05"
## [1] "Variable oldpeak p-valor 8.18337828923442e-17"
```

Per a totes les variables els p-valors són menors que 0.05 per la qual cosa, a un nivell de confiança del 95%, rebutjem la hipòtesi nul.la de que els valors estan normalment distribuïts.

Mostrarem els histogrames univariants de cada variable on s'aprecia que totes les variables presenten importants desviacions de la normalitat com biaixos esquerra-dreta o indicis de multimodalitat.

```
par(mfrow=c(3,2))
for(variable in num_vars) {
  hist(data[,variable], main=variable, xlab=variable)
}
```



Respecte a l'homocedasticitat, o igualtat de les variàncies entre els diferents grups de dades a comparar, per a comprovar-la s'estudiarà com varien els valors de les variables numèriques en funció de la resposta `disease`. Aplicarem el test de Fligner-Killeen la hipòtesi nul.la del qual és que les variàncies dels grups són iguals:

```
fligner.test(age ~ disease, data=data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  age by disease
## Fligner-Killeen:med chi-squared = 7.2746, df = 1, p-value =
## 0.006994
```

```
fligner.test(trestbps ~ disease, data=data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  trestbps by disease
## Fligner-Killeen:med chi-squared = 1.5023, df = 1, p-value = 0.2203
```

```r
fligner.test(chol ~ disease, data=data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  chol by disease
## Fligner-Killeen:med chi-squared = 0.76597, df = 1, p-value =
## 0.3815
```

```r
fligner.test(thalach ~ disease, data=data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  thalach by disease
## Fligner-Killeen:med chi-squared = 5.3987, df = 1, p-value =
## 0.02015
```

```r
fligner.test(oldpeak ~ disease, data=data)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  oldpeak by disease
## Fligner-Killeen:med chi-squared = 31.621, df = 1, p-value =
## 1.874e-08
```

En el cas de `age`, `thalach` i `oldpeak` el p-valor menor que 0,05 indica que poden rebutjar la hipòtesi nul.la i concloure que les variàncies són diferents en els grups de pacients sans i malalts. En el cas de `trestbps` i `chol` no podem rebutjar la hipòtesi nul.la de que els grups són homocedàstics.

4.3 Proves estadístiques

4.3.1 Contrast d'hipòtesis

A aquest apartat realitzarem un contrast d'hipòtesis per a determinar si el nivell de colesterol és similar en els pacients diagnosticats positiva i negativament o si hi ha diferències significatives entre ambdós tipus de pacients.

Estratificarem per la variable `disease` per a crear dues submostres, segons diagnosi positiu i negatiu, que són les que contrastarem. Usarem el test de Welch (derivat del test T de Student) per a comparar les mitjanes d'ambdós poblacions. Si denotem per μ_1 la mitjana del nivell de colesterol de la població de pacients malalts i per μ_2 la de la resta, les hipòtesis del test seran:

- Hipòtesi nul.la. $H_0 : \mu_1 - \mu_2 = 0$
- Hipòtesi alternativa. $H_a : \mu_1 - \mu_2 \neq 0$

Com que compararem mitjanes i el tamany de la mostra és 303 (major que el valor convencional de 30), pel Teorema del Límit Central podem assumir que la distribució de mitjanes és aproximadament normal, així que podem aplicar el test T amb garanties.

El següent codi obté les submostres i aplica el test T:

```r
data.disease <- data[data$disease == 1,]
data.non_disease <- data[data$disease == 0,]
t.test(data.disease$chol, data.non_disease$chol)
```

```
##
##  Welch Two Sample t-test
##
## data:  data.disease$chol and data.non_disease$chol
## t = 1.4924, df = 298.64, p-value = 0.1366
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.815018 20.484170
## sample estimates:
## mean of x mean of y
##  251.4748  242.6402
```

El p-valor major que 0,1 indica que, per a qualsevol nivell de confiança major que 90%, no es pot rebutjar la hipòtesis nul.la d'igualtat de mitjanes. És a dir: les dades disponibles no recolcen que hi ha diferències estadísticament significatives entre el nivell de colesterol dels pacients malalts i els sans, i són un indici de aquesta variable no és molt rellevant per a predir la presència de malaltia cardiovascular.

4.3.2 Independència entre la resposta i les variables qualitatives

Per a analitzar la correlació entre les variables qualitatives i la resposta podem usar el test Chi-quadrat. A aquest test la hipòtesi nul.la és que les variables són independents.

```
for(variable in cat_vars) {
  tab <- table(data$disease, data[,variable])
  results <- chisq.test(tab)
  print(paste("Variable", variable, "p-valor", results$p.value))
}
```

```
## [1] "Variable sex p-valor 2.66671234818094e-06"
## [1] "Variable cp p-valor 1.25171060078375e-17"
## [1] "Variable fbs p-valor 0.781273406706379"
## [1] "Variable restecg p-valor 0.00656652381421735"
## [1] "Variable exang p-valor 1.41378809671808e-13"
## [1] "Variable slope p-valor 1.1428845467527e-10"
## [1] "Variable ca p-valor 1.17425942803183e-15"
## [1] "Variable thal p-valor 3.69952147693275e-19"
```

Els p-valors menors que 0.05 indiquen que podem rebutjar la hipòtesi d'independència i assumir que hi ha relació entre les var1iables qualitatives i la resposta malaltia, excepte en el cas de `fbs` on l'alt p-valor indica que es pot acceptar que aquesta variable és independent del diagnòstic.

4.3.3 Regressió

A aquest apartat generarem un model de regressió logística que permetrà predir la presència de malaltia coronària en funció de diverses variables explicatives quantitatives i qualitatives.

La regressió logística està vinculada al concepte d'odds-ratio (OR) que mesura l'increment de probabilitat d'una resposta (en el nostre cas malaltia coronària) en funció d'un factor. Per a les variables binàries l'odd-ratio es pot calcular amb la taula de contingència. Per exemple, el següent codi calcula l'OR de malaltia coronària segons el sexe (recordem que la variable `sex` val 0 per a les dones i 1 per als homes).

```
odds.ratio.binary <- function(x, y) {
  tab <- table(x,y)
  return(tab[1,1]*tab[2,2]/(tab[1,2]*tab[2,1]))
}
```

```r
odds.ratio.binary(data$disease, data$sex)
```

```
## [1] 3.568696
```

L'OR indica que és 3.57 vegades més probable patir una malaltia coronària si se és home.

Per a construïr el model de regressió logística aplicarem una tècnica anomenada selecció de variables cap enrere (backward selection). Construïrem un model amb totes les variables independents i després eliminarem les estadísticament no significatives. Com és habitual a l'àmbit del *machine learning*, particionarem les dades en un conjunt d'entrenament i test per a validar l'efectivitat del model construït.

```r
train.test.split <- function(data, train_size=0.8) {
  smp_size <- floor(train_size * nrow(data))
  train_ind <- sample(seq_len(nrow(data)), size=smp_size, replace=FALSE)
  train <- data[train_ind,]
  test <- data[-train_ind,]
  return(list("train"=train, "test"=test))
}

test.model <- function(model, test_df) {
  probs <- predict(model, test_df, type="response")
  preds <- as.factor(ifelse(probs < 0.5, 0, 1))
  errors <- ifelse(test_df$disease==preds, 0, 1)
  df <- data.frame(test_df$disease, preds, probs, errors)
  colnames(df) <- c("Realitat", "Prediccio", "Probabilitat","Errors")
  return(list("df"=df,
              "accuracy"=1-sum(df$Errors)/nrow(df)))
}
```

Finalment construïm el model de regressió logística amb totes les variables explicatives:

```r
set.seed(123)
res <- train.test.split(data)
train <- res$train
test <- res$test
model <- glm(disease ~ ., data=train, family=binomial(link="logit"))
summary(model)
```

```
##
## Call:
## glm(formula = disease ~ ., family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9790  -0.5375  -0.1399   0.4316   2.7367
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.100852   3.269342  -1.254 0.209720
## age         -0.025433   0.025960  -0.980 0.327240
## sex1         1.493321   0.584413   2.555 0.010611 *
## cp2          1.041802   0.837246   1.244 0.213382
## cp3          0.436958   0.715180   0.611 0.541215
## cp4          2.202510   0.704711   3.125 0.001776 **
## trestbps     0.021484   0.012631   1.701 0.088954 .
```

```
## chol         0.002931   0.004251   0.689 0.490538
## fbs1        -0.497026   0.618844  -0.803 0.421886
## restecg1     0.885653   2.415596   0.367 0.713888
## restecg2     0.638122   0.420301   1.518 0.128952
## thalach     -0.022285   0.013703  -1.626 0.103896
## exang1       0.800569   0.474149   1.688 0.091328 .
## oldpeak      0.362455   0.246965   1.468 0.142203
## slope2       1.117982   0.504230   2.217 0.026609 *
## slope3       0.514305   0.966492   0.532 0.594632
## ca1          1.951149   0.538482   3.623 0.000291 ***
## ca2          2.843190   0.800578   3.551 0.000383 ***
## ca3          1.839818   0.939805   1.958 0.050270 .
## thal6       -0.289321   0.841602  -0.344 0.731016
## thal7        1.335227   0.466457   2.862 0.004203 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 334.42  on 241  degrees of freedom
## Residual deviance: 166.58  on 221  degrees of freedom
## AIC: 208.58
##
## Number of Fisher Scoring iterations: 6
```

La sortida de **summary** mostra que s'han creat variables dummy per als factors, i marca les variables estadísticament significatives amb asteriscs. El model final es construirà considerant aquestes variables explicatives significatives i descartant la resta:

```
model <- glm(disease ~ oldpeak+sex+cp+slope+ca+thal,
             data=train, family=binomial(link="logit"))
summary(model)
```

```
##
## Call:
## glm(formula = disease ~ oldpeak + sex + cp + slope + ca + thal,
##     family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6282  -0.5784  -0.1713   0.4911   2.6806
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -4.62273    0.86507  -5.344 9.10e-08 ***
## oldpeak      0.50887    0.22388   2.273 0.023027 *
## sex1         1.05794    0.49140   2.153 0.031327 *
## cp2          0.65040    0.79407   0.819 0.412748
## cp3          0.05233    0.67603   0.077 0.938294
## cp4          2.22620    0.63918   3.483 0.000496 ***
## slope2       1.31049    0.46786   2.801 0.005094 **
## slope3       0.57047    0.84103   0.678 0.497583
## ca1          1.97457    0.49313   4.004 6.22e-05 ***
## ca2          2.22261    0.71478   3.109 0.001874 **
## ca3          1.79655    0.89673   2.003 0.045129 *
```

```
## thal6       -0.09523    0.75797  -0.126 0.900017
## thal7        1.37326    0.43020   3.192 0.001412 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 334.42  on 241  degrees of freedom
## Residual deviance: 179.58  on 229  degrees of freedom
## AIC: 205.58
##
## Number of Fisher Scoring iterations: 5
```

Finalment comprovarem l'efectivitat del model de regressió logística amb el conjunt de dades de test:

```
results <- test.model(model, test)
print(paste("Precisió: ", results$accuracy, sep=""))
```

```
## [1] "Precisió: 0.950819672131147"
```

S'observa que el model de regressió logística ha assolit una precisió del 95% classificant els casos de malaltia del joc de dades de test.

5 Representació dels resultats

El model de regressió logística ha assolit una bona precisió identificant els casos de malaltia al joc de dades de test. La següent taula mostra els resultats per a les observacions amb error de diagnòstic:

```
df <- results$df
df[df$Errors==1,]
```

	Realitat	Prediccio	Probabilitat	Errors
172	0	1	0.5086649	1
262	1	0	0.1480825	1
291	1	0	0.3961323	1

Als estudis clínics és molt important conèixer els tipus d'errors comesos. Els errors són de dos tipus:

- Errors de tipus I o falsos positius: diagnòstic positiu sense malaltia.
- Errors de tipus II o falsos negatius: malaltia sense diagnòstic positiu.

La següent matriu de confusió mostra quants falsos positius i falsos negatius s'han comès amb el model de regressió logística generat a aquest exercici. Les files de la matriu indiquen els diagnòstics reals i les columnes les prediccions generades pel model:

```
table(df$Realitat, df$Prediccio)
```

```
##
##      0  1
##   0 34  1
##   1  2 24
```

S'observa que el model de regressió logística ha comés 1 error de tipus I i 2 errors de tipus II.

6 Conclusions

S'ha seleccionat un joc de dades reial i s'han preprocessat les dades per a permetre respondre a diverses preguntes d'interès analític relacionades amb quins són els factors que més influència semblen tenir en la presència d'una malaltia cardiovascular.

En concret, s'ha generat un model de regresssió logística que ha permès identificar les variables més significatives i, a més, ha assolit una alta precisió predint nous diagnòstics.

Els resultats dels diferents anàlisis s'han presentat mitjançant gràfics i taules.

7 Codi

El codi R markdown, així com els fitxers CSV original i preprocessat i aquest fitxer PDF són al repositori https://github.com/dgilros/PRA2_Tipologia

Per a generar el fitxer amb les dades preprocessades s'ha usat el següent codi R:

```
write.csv(data, file="heart.csv", sep=",", row.names=FALSE)
```

Referències

1) https://archive.ics.uci.edu/ml/datasets/Heart+Disease
2) https://www.who.int/health-topics/cardiovascular-diseases/#tab=tab_1