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 coronàries i això pot contribuïr al seu diagnòstic i prevenció. Aquests objectius són importants: segons l'Organització Mundial de la Salut les malalties cardiovasculars van ser responsables del 31% de les morts registrades al món durant l'any 2015 [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. Posteriorment es tractaràn de forma diferenciada de les quantitatives:

```
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 ...
```

3.1 Elements buits

El següent codi R mostra que el joc de dades conté valors buits:

```
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      num
##        0        0        0        4        2        0
```

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 com els k-veïns més propers o *random forests*. No obstant, com en aquest cas el nombre de registres afectats és molt reduit i es perd poca informació (6 registres de 303, menys d'un 2%) optarem per descartar-los. El següent codi R filtra els registres amb valors buits:

```
data <- data[!is.na(data$ca),]
data <- data[!is.na(data$thal),]
nrow(data)
```

```
## [1] 297
```

S'observa que el número de registres del joc de dades ha passat a ser 297.

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
```

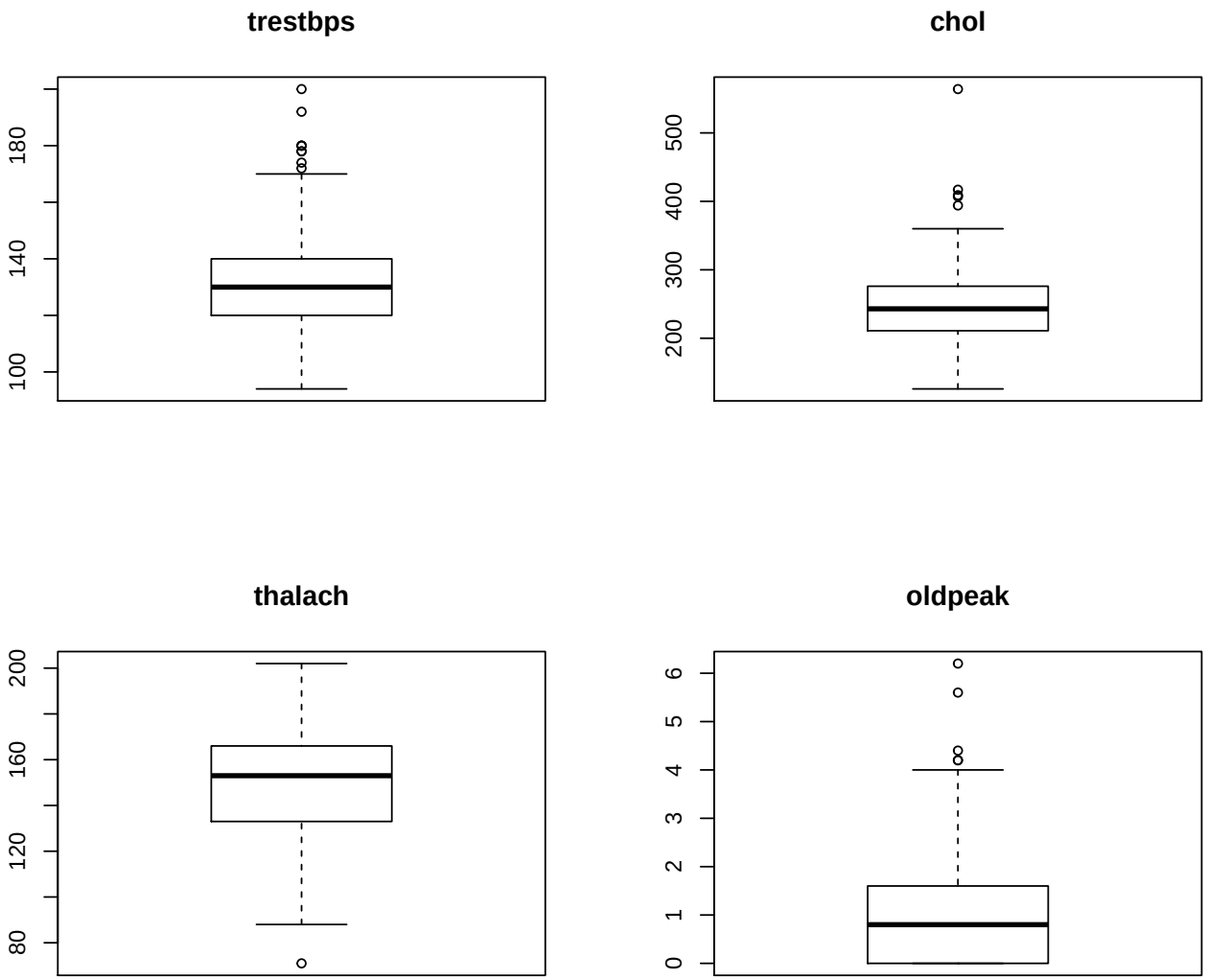
Variable	#Outliers	%Outliers	Q1-1.5IQR	Q3+1.5IQR	Valors outliers
trestbps	9	3.0303030	90.0	170.0	172,180,200,174,178,192,180,178,180
chol	5	1.6835017	113.5	373.5	417,407,564,394,409
thalach	1	0.3367003	83.5	215.5	71
oldpeak	5	1.6835017	-2.4	4.0	6.2,5.6,4.2,4.2,4.4

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 usarem la tècnica dels 3 veïns més propers segons la resta d'atributs, aplicant la funció `kNN` de la llibreria VIM [3]. Com que aquesta funció assumeix que els registres a imputar tenen valor buit, abans de cridar-la haurem d'assignar el valor `NA` als outliers:

```
for(variable in df_outliers$Variable) {
  values <- data[,variable]
  outliers <- boxplot.stats(values)$out
  data[values %in% outliers, variable] <- NA
}
colSums(is.na(data))
```

```
##      age      sex       cp trestbps     chol      fbs  restecg  thalach
##        0        0        0        9        5        0        0        1
##    exang  oldpeak    slope       ca     thal      num
##        0        5        0        0        0        0
```

Ara podem usar la funció `kNN` de la llibreria VIM:

```
library(VIM)
data <- kNN(data, variable=num_vars, k=3)
```

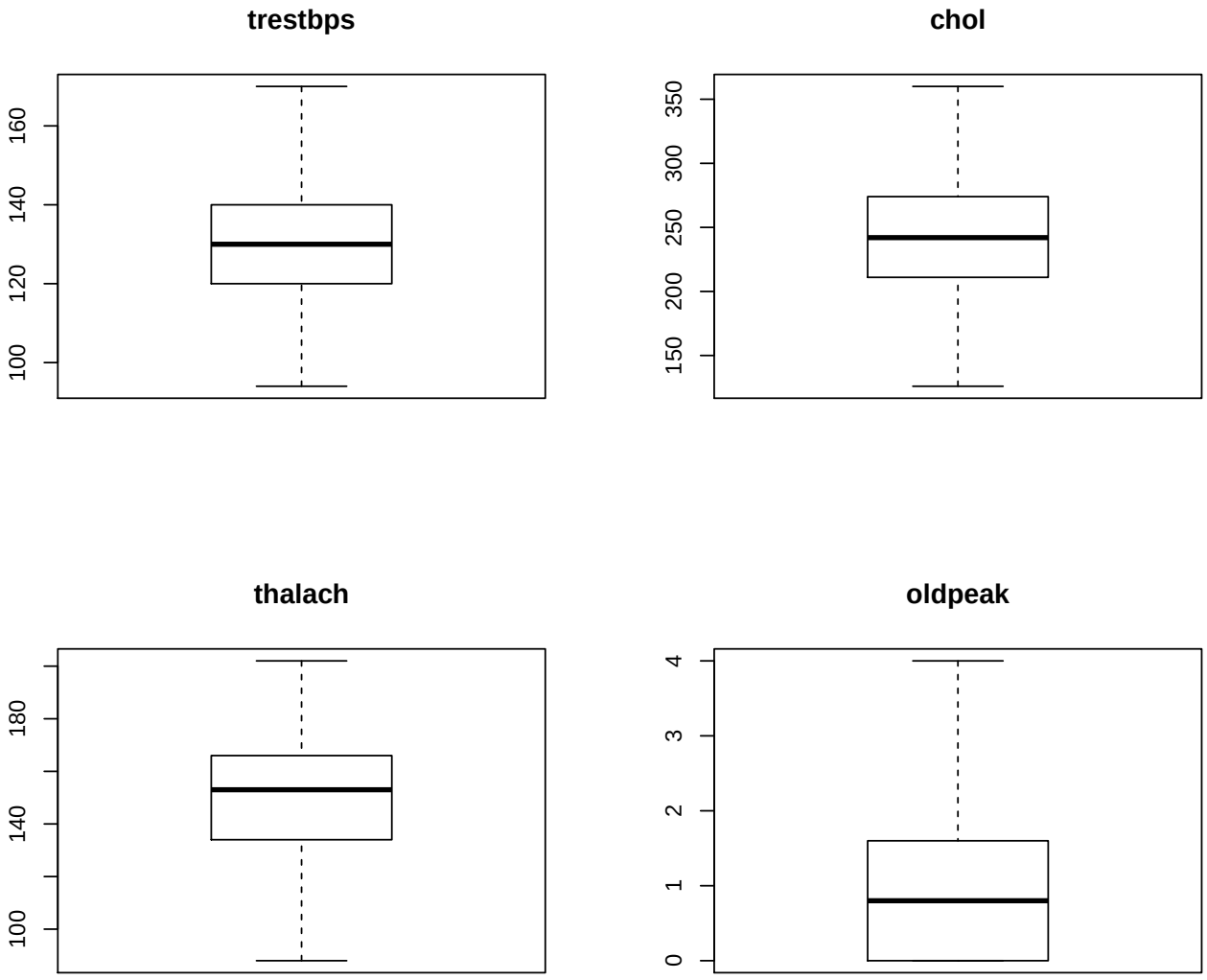
La funció crea variables booleanes "_imp" per a indicar si s'ha imputat un valor a un registre. Això facilita observar els valors imputats. Per exemple, per a la variable `chol`:

```
data[data$chol_imp==TRUE,]$chol
```

```
## [1] 283 225 250 264 282
```

Si dibuixem els boxplots s'observa que ja no es detecten valors extrems:

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



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.

Per aquest motiu substituïrem la variable `num` per una variable qualitativa binomial `disease` que valdrà 1 si el pacient presenta malaltia i 0 en cas contrari. Segons [1] totes les recerques publicades s'han basat en aquest criteri.

```
data$disease <- ifelse(data$num == 0, 0, 1)
data$num <- NULL
```

Ambdues classes estan prou equilibrades al joc de dades: hi ha 137 pacients malalts i 160 sans.

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

```
##
##   0   1
## 160 137
```

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:

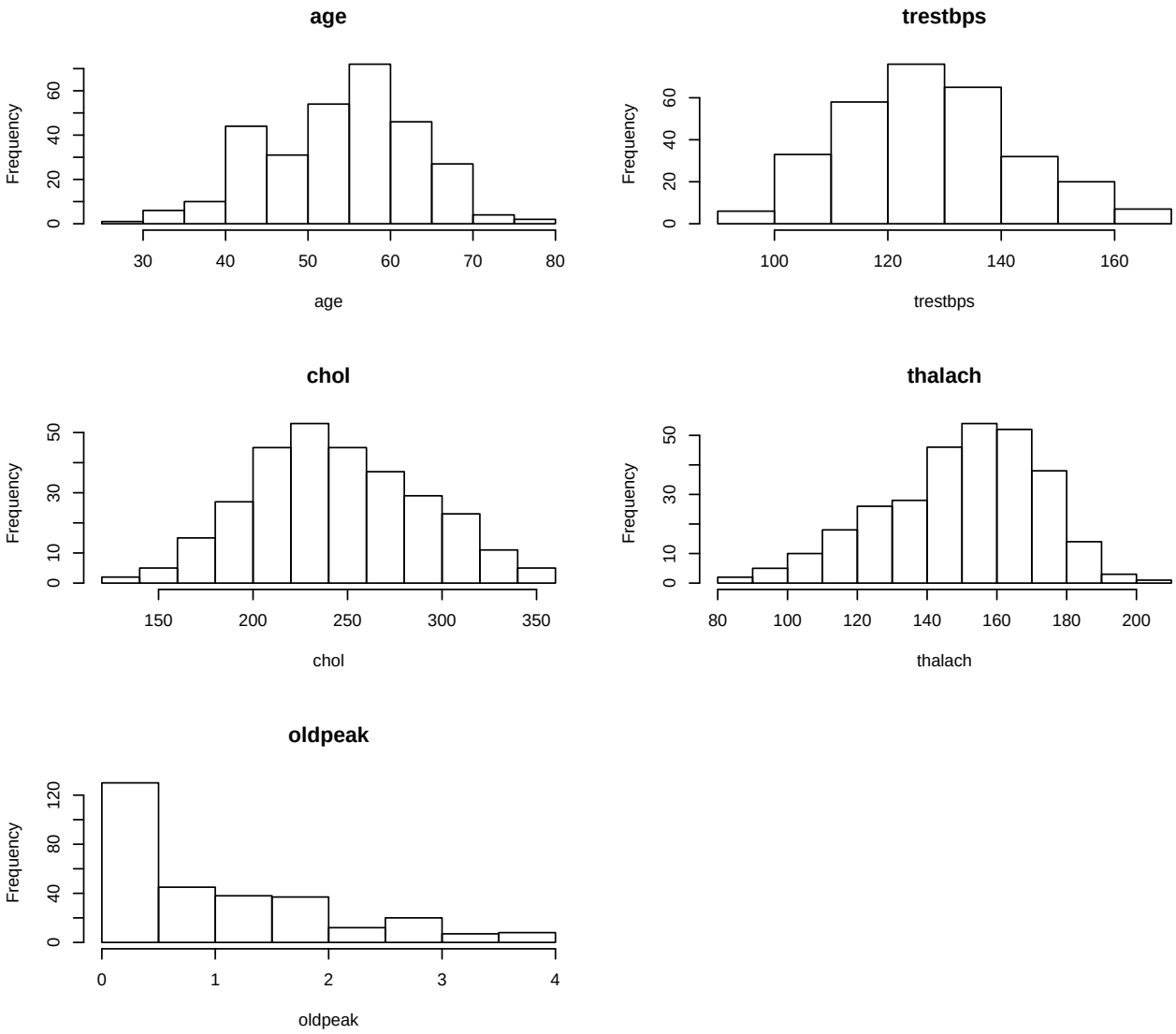
```
for(variable in num_vars) {
  results <- shapiro.test(data[,variable])
  print(paste("Variable", variable, "p-valor", results$p))
}
```

```
## [1] "Variable age p-valor 0.00542394906133735"
## [1] "Variable trestbps p-valor 0.00161312633483421"
## [1] "Variable chol p-valor 0.337571660944917"
## [1] "Variable thalach p-valor 0.000120807091594364"
## [1] "Variable oldpeak p-valor 1.89083249257303e-15"
```

Per a totes les variables excepte `chol` 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. En el cas de `chol` el p-valor major que 0.05 indica que no es pot rebutjar la hipòtesi de normalitat.

Mostrarem els histogrames univariants de cada variable on s'aprecia que, efectivament la variable `chol` sembla normalment distribuïda però la resta presenten més desviacions de la normalitat (biaixos o la esquerra o dreta), especialment `oldpeak` que és la que ha obtingut el p-valor més baix al test de Shapiro-Wilk:

```
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 usarem el test de Fligner-Killeen. Ens interessarà conèixer la variació en la reposta (variable `disease`) en funció de diversos factors:

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

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  disease by sex
## Fligner-Killeen:med chi-squared = 9.1056, df = 1, p-value =
## 0.002548
```

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

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  disease by exang
## Fligner-Killeen:med chi-squared = 1.9197, df = 1, p-value = 0.1659
```

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

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  disease by fbs
## Fligner-Killeen:med chi-squared = 0.0029685, df = 1, p-value =
## 0.9565
```

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 homes i dones tractats per afeccions coronàries o si hi ha diferències significatives segons el sexe.

Estratificarem per la variable `sex` per a crear dues submostres, una d'homes i altra de dones, 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 d'homes i per μ_2 la de les dones, 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 297 (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.female <- data[data$sex == 0,]
data.male <- data[data$sex == 1,]
t.test(data.female$chol, data.male$chol)
```

```
##
##  Welch Two Sample t-test
##
## data:  data.female$chol and data.male$chol
## t = 2.2051, df = 168.46, p-value = 0.0288
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   1.334706 24.156899
## sample estimates:
## mean of x mean of y
##  252.9896  240.2438
```

El p-valor menor de 0.05 indica que amb les dades disponibles i per a un nivell de confiança del 95%, podem rebutjar la hipòtesis nul.la d'igualtat de mitjanes i acceptar que hi ha diferències estadísticament significatives en el nivell de colesterol segons el sexe.

4.3.2 Correlació

En primer lloc analitzarem la correlació entre les variables numèriques i la resposta `disease`.

```
df <- data.frame("", 0., 0., stringsAsFactors=FALSE)
colnames(df) <- c("Variable", "Estimate", "p-value")
row <- 1
for(variable in num_vars) {
  spearman_test = cor.test(data[,variable], data$disease, method="spearman", exact=FALSE)
  corr_coef <- spearman_test$estimate
  p_val <- spearman_test$p.value
  df[row,] <- list(variable, corr_coef, p_val)
  row <- row+1
}
df
```

Variable	Estimate	p-value
age	0.2399263	0.0000293
trestbps	0.1090234	0.0605800
chol	0.1182148	0.0417674
thalach	-0.4226073	0.0000000
oldpeak	0.4169772	0.0000000

Per a les variables qualitatives podem usar el test Chi-quadrat.

```
cat_vars <- c('sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal')
for(variable in cat_vars) {
  tab <- table(data$disease, data[,variable])
  results <- fisher.test(tab)
  print(paste("Variable", variable, "p-valor", results$p.value))
}
```

```
## [1] "Variable sex p-valor 1.80354898631261e-06"
## [1] "Variable cp p-valor 1.29860544246526e-17"
## [1] "Variable fbs p-valor 1"
## [1] "Variable restecg p-valor 0.00483362703047518"
## [1] "Variable exang p-valor 4.17516086446576e-13"
## [1] "Variable slope p-valor 1.08373960645722e-10"
## [1] "Variable ca p-valor 1.48834446841864e-16"
## [1] "Variable thal p-valor 6.49640400399579e-20"
```

Els p-valors menors que 0.05 indiquen que hi ha relació entre les variables qualitatives i la resposta malaltia, excepte en el cas de `fbs` on el p-valor és igual a 1 indicant que no s'aprecia relació entre aquesta variable i el 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]))
```

```
}

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

```
## [1] 3.573933
```

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.

```
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 <- ifelse(probs < 0.5, 0, 1)
  errors <- ifelse(test_df$disease-preds != 0, 1, 0)
  df <- data.frame(test_df$disease, preds, probs, errors)
  colnames(df) <- c("Realitat", "Prediccio", "Probabilitat","Errors")
  return(list("df"=df,
              "accuracy"=1-sum(df$Errors)/nrow(df)))
}
```

Abans de construïr el model, crearem variables dummy per a codificar les categories descartant els valors de referència com s'indica a [1]. També eliminarem els atributs "_imp" generades per la funció `kNN` de `VIM`:

```
data2 <- data
# eliminar atributs "_imp"
for(variable in colnames(data2)) {
  imp_var <- paste(variable, "_imp", sep="")
  data2[,imp_var] <- NULL
}
# crear variables dummy per a els atributs categòrics
for(variable in cat_vars) {
  var_values <- as.factor(data2[,variable])
  var_levels <- levels(var_values)
  # descartar valor de referència
  if(variable == 'cp') {
    ref_level_index <- 4
  } else {
    ref_level_index <- 1
  }
  for(level in var_levels[-ref_level_index]) {
      new_var <- paste(variable, level, sep="")
      data2[,new_var] <- ifelse(var_values == level, 1, 0)
  }
  data2[,variable] <- NULL
}
```

```r
str(data2)
```

```
## 'data.frame':    297 obs. of  21 variables:
##  $ age     : num  63 67 67 37 41 56 62 57 63 53 ...
##  $ trestbps: num  145 160 120 130 130 120 140 120 130 140 ...
##  $ chol    : num  233 286 229 250 204 236 268 354 254 203 ...
##  $ thalach : num  150 108 129 187 172 178 160 163 147 155 ...
##  $ oldpeak : num  2.3 1.5 2.6 3.5 1.4 0.8 3.6 0.6 1.4 3.1 ...
##  $ disease : num  0 1 1 0 0 0 1 0 1 1 ...
##  $ sex1    : num  1 1 1 1 0 1 0 0 1 1 ...
##  $ cp1     : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ cp2     : num  0 0 0 0 1 1 0 0 0 0 ...
##  $ cp3     : num  0 0 0 1 0 0 0 0 0 0 ...
##  $ fbs1    : num  1 0 0 0 0 0 0 0 0 1 ...
##  $ restecg1: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ restecg2: num  1 1 1 0 1 0 1 0 1 1 ...
##  $ exang1  : num  0 1 1 0 0 0 0 1 0 1 ...
##  $ slope2  : num  0 1 1 0 0 0 0 0 1 0 ...
##  $ slope3  : num  1 0 0 1 0 0 1 0 0 1 ...
##  $ ca1     : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ ca2     : num  0 0 1 0 0 0 1 0 0 0 ...
##  $ ca3     : num  0 1 0 0 0 0 0 0 0 0 ...
##  $ thal6   : num  1 0 0 0 0 0 0 0 0 0 ...
##  $ thal7   : num  0 0 1 0 0 0 0 0 1 1 ...
```

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

```r
set.seed(123)
res <- train.test.split(data2)
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
## -2.8855  -0.4580  -0.1078   0.3164   2.7002
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -7.710580   3.637358  -2.120 0.034020 *
## age         -0.005952   0.029422  -0.202 0.839673
## trestbps     0.028458   0.015392   1.849 0.064472 .
## chol         0.009776   0.005570   1.755 0.079235 .
## thalach     -0.010850   0.014011  -0.774 0.438675
## oldpeak      0.711033   0.299168   2.377 0.017468 *
## sex1         1.221567   0.611240   1.999 0.045662 *
## cp1         -2.476299   0.779833  -3.175 0.001496 **
## cp2         -0.920238   0.623247  -1.477 0.139804
## cp3         -2.293732   0.636408  -3.604 0.000313 ***
```

```
## fbs1        -0.702869   0.732706  -0.959 0.337418
## restecg1     1.390640   2.862410   0.486 0.627089
## restecg2     0.833772   0.471428   1.769 0.076959 .
## exang1       0.359852   0.531300   0.677 0.498212
## slope2       1.039911   0.550504   1.889 0.058890 .
## slope3       0.983477   1.018790   0.965 0.334376
## ca1          2.204043   0.568306   3.878 0.000105 ***
## ca2          2.996297   0.857638   3.494 0.000476 ***
## ca3          3.248777   1.880234   1.728 0.084014 .
## thal6        0.463018   0.987352   0.469 0.639105
## thal7        2.196840   0.535060   4.106 4.03e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 327.33  on 236  degrees of freedom
## Residual deviance: 143.44  on 216  degrees of freedom
## AIC: 185.44
##
## Number of Fisher Scoring iterations: 6
```

La sortida de `summary` marca les variables estadísticament significatives amb asteriscs. El model final es construirà considerant aquestes variables explicatives i descartant la resta:

```
model <- glm(disease ~ oldpeak+sex1+cp1+cp3+slope2+ca1+ca2+thal7,
             data=train, family=binomial(link="logit"))
summary(model)
```

```
##
## Call:
## glm(formula = disease ~ oldpeak + sex1 + cp1 + cp3 + slope2 +
##     ca1 + ca2 + thal7, family = binomial(link = "logit"), data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6385  -0.5201  -0.1347   0.4786   3.0601
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.3198     0.5546  -5.986 2.15e-09 ***
## oldpeak       1.0318     0.2353   4.385 1.16e-05 ***
## sex1          0.8647     0.4794   1.804 0.071274 .
## cp1          -2.2175     0.6888  -3.219 0.001285 **
## cp3          -2.2035     0.5371  -4.103 4.09e-05 ***
## slope2        1.3219     0.4360   3.032 0.002433 **
## ca1           2.2104     0.5259   4.203 2.63e-05 ***
## ca2           2.5284     0.7347   3.441 0.000579 ***
## thal7         2.1661     0.4334   4.998 5.81e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 327.33  on 236  degrees of freedom
```

```
## Residual deviance: 168.78  on 228  degrees of freedom
## AIC: 186.78
##
## Number of Fisher Scoring iterations: 6
```

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: 0.85"
```

S'observa que el model de regressió logística ha assolit una precisió del 85% 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 20 primeres observacions de la mostra de test:

```
df <- results$df
head(df, 20)
```

	Realitat	Prediccio	Probabilitat	Errors
2	1	1	0.6021644	0
3	1	1	0.9980617	0
12	0	0	0.3415197	0
15	0	0	0.1216840	0
19	0	0	0.0048840	0
38	1	1	0.8450456	0
44	0	0	0.0470800	0
47	0	0	0.0173014	0
49	0	0	0.0767473	0
56	1	1	0.9959835	0
62	0	0	0.0166476	0
65	1	1	0.9908775	0
68	0	0	0.3012056	0
80	1	1	0.6309871	0
82	0	0	0.1700633	0
92	0	1	0.6652272	1
96	1	1	0.9888099	0
97	1	1	0.9954104	0
99	0	0	0.0790642	0
100	0	0	0.0790642	0

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

La següent matriu de confusió mostra quants falsos positius i falsos negatius (errors de 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:

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

```
##
##      0  1
##   0 32  1
##   1  8 19
```

S'observa que el model de regressió logística no ha comés gairebè errors de tipus I però ha comés molts errors de tipus II. Podriem rebaixar els errors de tipus II reduïnt el nivell de probabilitat del 0.5 per als negatius, però això és una dada que ha de contrastar-se.

6 Conclusions

S'ha seleccionat un joc de dades complex..

7 Codi