Tabla de contenido

Format d´entrega 2
Exercicis 3
Exercici 1 3
Normalització per la diferència 4
Escalat decimal 5
Normalització basada en la desviació estàndard: estandardització de valors 6
Exercici 2 7
Mètodes de selecció d´atributs 8
Mètodes d´extracció d´atributs 10
Exercici 3 13
Reemplaçar els valors desconeguts per una constant 13
Reemplaçar el valor desconegut amb la mitjana 15
Reemplaçar el valor desconegut amb el valor més freqüent 16
Reemplaçar amb un valor aleàtori de la distribució de la variable 17
Exercici 4 18
Carrega i exàmen preliminar del conjunt de dades 18
Exploració i tractament de valors desconeguts 21
Discretització d´atributs 22
Reducció de la dimensionalitat 24
Exploració visual de les dades (EDA) 25
Bibliografia 29

Format d´entrega

Aquest document s´ha realitzat mitjançant **Markdown** [1] amb l´ajuda del entorn de desenvolupament **RStudio**[2] utilitzant les característiques que aquest ofereix per a la creació de documents **R** reproduibles.

La documentació generada en la realització de la pràctica es troba allotjada en **GitHub** al següent repositori:

- **https://github.com/rsanchezs/dataminig**

En aquest repositori es poden trobar els següents fitxers:

- Aquest document en formats **pdf** i **docx** amb el nom rsanchezs_PAC2.
- Un document **R Markdown**[3] que es pot utilitzar per a reproduir tots els exemples presentats a la PAC.
- El conjunt de dades utilitzades.

[1] **https://es.wikipedia.org/wiki/Markdown**

[2] **https://www.rstudio.com/**

[3] **https://rmarkdown.rstudio.com/**

1. **Nota:** Propietat intel·lectual

Sovint és inevitable, al produir una obra multimèdia, fer ús de recursos creats per terceres persones. És per tant comprensible fer-lo en el marc d'una pràctica dels Estudis, sempre que això es documenti clarament i no suposi plagi en la pràctica.
Per tant, al presentar una pràctica que faci ús de recursos aliens, s'ha de presentar juntament amb ella un document en quin es detallin tots ells, especificant el nom de cada recurs, el seu autor, el lloc on es va obtenir i el seu estatus legal: si l'obra està protegida pel copyright o s'acull a alguna altra llicència d'ús (Creative Commons, llicència GNU, GPL ...).
L'estudiant haurà d'assegurar-se que la llicència no impedeix específicament el seu ús en el marc de la pràctica. En cas de no trobar la informació corresponent haurà d'assumir que l'obra està protegida per copyright.
Hauríeu de, a més, adjuntar els fitxers originals quan les obres utilitzades siguin digitals, i el seu codi font si correspon.

Exercicis

Exercici 1

Explica el concepte de transformació de valors i dona tres exemple on es vegi la seva utilitat.

Per **transformació de valors** entenem modificacions dins el tipus de valors que poden prendre tots o alguns del atributs.

Les operacions més habituals són la **normalització** i la **discretització** de dades. Existeixen varies tècniques per a la transformació de dades, passem a examinar tres de les més importants. Considerarem X com al valor original del atribut i X^* com al valor del atribut normalitzat.

Normalització per la diferència

La normalització per la diferència tracta de compensar l´efecte de la distància del valor que tractem respecte al màxim dels valors observats. Es a dir,

$$X^* = \frac{X - min(X)}{rango(X)} = \frac{X - min(X)}{max(x) - min(x)}$$

Exemple de normalització per la diferència

Per a il·lustrar aquesta tècnica utilitzarem el conjunt de dades `cars`[4]del següent llibre *"Data Mining and Predictive Analytics"*[5]:

```
# Carregem les dades
library(readr)
cars <- read_csv("data/cars.txt")
```

Al següent fragment presentem un resum estadístic de la variable `weightlbs`:

```
# Resum estadístic variable `weight`
summary(cars$weightlbs)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1613    2246    2835    3005    3664    4997
```

Trobem els valors màxim i mínim de la variable `weightlbs`:

[4] Conjunt de dades disponibles en **http://www.dataminingconsultant.com**

[5] Daniel T. Larouse, Chantal D. Larouse: Data Mininig and Predictive Analytics.USA, John Wiley & Sons,2015,ISBN 978-1-118-11619-7

```
# Trobem els valors mínim i màxim de la variable `weight`
valor_min <- min(cars$weightlbs)
valor_max <- max(cars$weightlbs)
```

Finalment, trobem el valor normalitzat de la variable `weightlbs`:

```
# Valor normalitzat mitjançant la tècnica de la diferència
valors_norm_dif <- (cars$weightlbs - valor_min)/(valor_max - valor_
min)
head(valors_norm_dif)
```

```
## [1] 0.76713948 0.09219858 0.54255319 0.63475177 0.12943262 0.67
582742
```

Escalat decimal

La tècnica del **escalat decimal** ens garanteix que tots els valors normalitzats estiguin entre -1 i 1.

$$X^*_{decimal} = \frac{X}{10^d}$$

on d representa el nombre de dígits del valor de la dada amb el valor absolut més gran.

Exemple de normalització decimal

Per a explicar aquesta tècnica seguirem treballant amb el conjunt de dades `cars` de l´exemple anterior:

```
# Calculem el valor de decimals
max(abs(cars$weightlbs))
```

```
## [1] 4997
```

Un cop calculat el nombre de dígits (d), ens trobem amb condicions de transformar els valors de la variable `weight`:

```
# Valors transformats amb la tècnica decimal
valors_norm_dec <- cars$weightlbs/10^4
head(valors_norm_dec)

## [1] 0.4209 0.1925 0.3449 0.3761 0.2051 0.3900
```

Normalització basada en la desviació estàndard: estandardització de valors

El **mètode d´estandardització de valors** assegura que s´obtenen valors dins el rang escollit que tenen la propietat que la seva mitjana és zero i la seva desviació estàndard val 1.

Es a dir, l´estandardització consisteix en la diferència entre el valor de l´atribut i la seva mitjana, dividint aquesta diferència per la desviació estàndard dels valors de l´atribut. Es a dir:

$$Z - score \quad = \quad \frac{X \quad - \quad mean(X)}{SD(X)}$$

Exemple de normalització per estandardització

En primer lloc, ens caldrà calcular la mitjana i la desviació estàndard:

```
## Calculem la mitjana
m <- mean(cars$weightlbs)

## Calculem la desviació estandard
s <- sd(cars$weightlbs)
```

Finalment, apliquem la transformació mitjançant la formula presentada anteriorment:

```
# Estandardització de valors
valors_norm_z <- (cars$weightlbs - m)/s
head(valors_norm_z)
```

```
## [1]  1.4115004 -1.2672210  0.5201570  0.8860769 -1.1194457  1.0
490989
```

Exercici 2

Explica el concepte de reducció de nombre d'atributs i dona un exemple on es vegi la seva utilitat. De quines tècniques disposem per a comprovar que no estem perdent qualitat en aquests procés?

La reducció del nombre d'atributs consisteix a trobar un subconjunt dels atributs originals que permeti d'obtenir models de la mateixa qualitat que els que s'obtindrien utilitzant tots els atributs. Aquest problema s'anomena **problema de la selecció òptima d'atributs**.

Aquest problema pot tindre diferents enfocaments, com per exemple: escollir els millors atributsa partir d´un anàlisi preliminar, el.limanar atributs redundants o que aporten poca informació, o reduir la dimensionalitat de les dades generant nous atributs a partir dels existents. En tots aquests casos, la finalitat es reduir el cost computacional per a la creació de models.

Existeixen els següents mètodes per a tractar amb el problema de la selecció òptima d´atributs:

- Mètodes de selecció d´atributs
- Mètodes de reducció del nombre d´atributs

Tot seguit explicarem en que consisteixen cada un dels mètodes i presentarem algunes de les tècniques més importants per a la reducció d´atributs.

Mètodes de selecció d´atributs

La **selecció d´atributs** consisteix a escollir únicament atributs que són realment rellevants per a resoldre el problema, descartant aquells que no ens aporten informació rellevant per a resoldre el problema.

Depenent de si la selecció de característiques fa ús o no de la informació del mètode de classificació posterior, podem definir la següent taxonomia:

- Els **algoritmes filtre** (*filter*), on els atributs o conjunt d´atributs son evaluats de forma independent respecte del mètode de classificació que s´utilitzarà amb posterioritat.
- Els **algoritmes empotrats** (*wrapper*), on el mètode de selecció de característiques utilitza el classificador que usarà amb posterioritat.

A continuació passem a estudiar els diferents mètodes de selecció de característiques i els algoritmes utilitzats.

En primer lloc, explicarem breument els mètodes per a la selecció d´atributs individuals, coneguts com a **algoritmes univariants**:

- Selecció de màxima rellevància (*maxium relevance selection*), que utilitza el coeficient de correlació entre cada atribut.
- Selecció basada en la informació mútua, mesura la informació mútua entre variables aleatòries que modelen cada característica i les etiquetes de classificació.
- Mètodes basats en tests estadístics, apliquen tests estadístics de hipòtesi sobre les dades, com per exemple el **t-static** o el **chi-square**.

En segon lloc, trobem els mètodes de selecció de subconjunts d´atributs, coneguts com a **algoritmes multivariants**:

- Recerca exhaustiva (*exhaustive search*), consisteix en definir un espai de recerca i avaluar, mitjançant un funció de cost, totes les possibles combinacions. Només es aplicable a problemes de dimensionalitat reduïda.
- Selecció pas a pas (*stepwise selection*), consisteix en iterar per un algoritme en el que cada pas o be afegeix al conjunt d´atributs aquell atribut que augmenta el rendiment global del conjunt, o bé el.elimina aquell atribut que fa que el rendiment empitjori.
- Ramificació i poda (*branch and bound*), consisteix en aplicar la tècnica de recerca **branch and bound**.

Mètodes d´extracció d´atributs

La extracció d´atributs es tracta de calcular nous atributs a partir d´existents, amb l´objectiu de que els nous atributs resumeixin millor la informació que contenen, capturant la naturalesa de la estructura subjacent en les dades.

Anàlisi de Components Principals (PCA)

L´anàlisi de components principals (*Principal Component Analysis*, PCA) ens ajuda a solucionar problemes de reducció de dimensionalitat i extracció de característiques en les nostres dades de manera automàtica. El PCA es un algoritme molt conegut en l´àmbit de l`anàlisi de dades, i té moltes aplicacions diferents. Informalment, es pot definir com la tècnica que intenta aconseguir una representació d´un conjunt de dades a un espai de dimensionalitat més reduïda, minimitzant l´error quadràtic.

Exemple de Anàlisi de Components Principal (PCA)

Per a il.lustrar aquest exemple farem ús del *dataset* `houses`[6]:

```
# Realitzem la lectura de les dades
library(readr)
houses <- read_delim("data/houses.csv", ";",
    escape_double = FALSE, col_names = FALSE,
    trim_ws = TRUE, skip = 1)
```

A continuació preparem les dades per a realitzar l`ànalisi:

[6] Conjunt de dades disponible en StatLib: **http://lib.stat.cmu.edu/datasets/houses.zip**

```
# Donem nom als atributs
names(houses) <- c("MVAL", "MINC", "HAGE", "ROOMS", "BEDRMS", "POPN" ,
"HHLDS", "LAT", "LONG")
```

A continuació normalitzem les dades amb el **mètode d´estandardització de valors** que hem tractat en el **exercici 1**:

```
# Estanderditzem les variables

houses$MVAL_Z <- (houses$MVAL - mean(houses$MVAL))/(sd(houses$MVAL)
)
houses$MINC_Z <- (houses$MINC - mean(houses$MINC))/(sd(houses$MINC)
)
houses$HAGE_Z <- (houses$HAGE - mean(houses$HAGE))/(sd(houses$HAGE)
)
houses$ROOMS_Z <- (houses$ROOMS - mean(houses$ROOMS))/(sd(houses$RO
OMS))
houses$BEDRMS_Z <- (houses$BEDRMS - mean(houses$BEDRMS))/(sd(houses
$BEDRMS))
houses$POPN_Z <- (houses$POPN - mean(houses$POPN))/(sd(houses$POPN)
)
houses$HHLDS_Z <- (houses$HHLDS - mean(houses$HHLDS))/(sd(houses$HH
LDS))
houses$LAT_Z <- (houses$LAT - mean(houses$LAT))/(sd(houses$LAT))
houses$LONG_Z <- (houses$LONG - mean(houses$LONG))/(sd(houses$LONG)
)
```

Seleccionarem una mostra aleàtoria del 90% del conjunt de dades:

```
# Seleccionem aleatoriament el 90% de les dades per al joc de prov
es
dist_unif <- runif(dim(houses)[1], min = 0, max = 1)
test_houses <- houses[which(dist_unif < .1), ]
train_houses <- houses[which(dist_unif <= .1), ]
```

Per a realitzar el PCA utilitzarem el paquet `psych`[7]:

```
# Instal.lem el paquet
# install.packages("psych",dependencies=TRUE)
# Carreguem el paquet en la sesió
library(psych)

# Anàlisi de Components Principal (PCA)
pca_analysis <- principal(train_houses[, c(10:17)],
                          nfactors = 8,
                          rotate = "none",
                          scores = TRUE)
```

A continuació es mostren els resultats de l`ànalisi PCA:

```
# Resultas PCA
# Valors propis (eigen)
pca_analysis$values

## [1] 3.86450242 1.74531620 0.98446438 0.89384311 0.28371680 0.14
378879
## [7] 0.06629445 0.01807386

# Mostrem la matriu amb les variàncies
pca_analysis$loadings

##
## Loadings:
##          PC1    PC2    PC3    PC4    PC5    PC6    PC7    PC8
## MVAL_Z    0.907         0.207 -0.349
## MINC_Z    0.907  0.219         0.336
## HAGE_Z   -0.424        -0.329  0.833  0.135
## ROOMS_Z   0.956                              -0.174 -0.178
## BEDRMS_Z  0.969                0.115         -0.127
## POPN_Z    0.932 -0.104                 0.123  0.302
## HHLDS_Z   0.973                0.130                0.136
## LAT_Z            -0.261  0.902  0.333
```

[7] Podem consultar la documentació en **https://cran.r-project.org/web/packages/psych/index.html**

```
##
##                  PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8
## SS loadings    3.865 1.745 0.984 0.894 0.284 0.144 0.066 0.018
## Proportion Var 0.483 0.218 0.123 0.112 0.035 0.018 0.008 0.002
## Cumulative Var 0.483 0.701 0.824 0.936 0.971 0.989 0.998 1.000
```

Exercici 3

De quines tècniques disposem per resoldre el problema de la possible falta de valors d'un atributs? Dona almenys un exemple de cada tècnica.

Algunes de les opcions disponibles són les següents:

1. Reemplaçar els valors desconeguts per una constant, especificada per l´analista.
2. Reemplaçar el valor desconegut amb la mitjana (per a variables numèriques) o la moda (per a variables categòriques).
3. Reemplaçar els valors desconeguts amb un valor generat aleatòriament de la distribució de la variable.

Reemplaçar els valors desconeguts per una constant

En els següents exemples utilitzarem el *dataset* `cars` que ja hem utilitzat en apartats anteriors:

```
# Importem les dades
library(readr)
cars <- read_csv("data/cars.txt")
```

Realitzem un primer contacte amb el joc de dades, visualitzant la seva estructura i els 6 primers registres:

```
# Realitzem un exàmen preliminar del conjunt de dades
str(cars)
head(cars)
```

Per tal de simplificar l´exemple i millorar la llegibilitat del document només treballarem amb quatre variables:

```
# Seleccionem les variables mpg, cubicinches, hp i brand
my_cars <- cars[, c(1, 3, 4, 8)]
head(my_cars)
```

```
## # A tibble: 6 x 4
##     mpg cubicinches    hp brand
##   <dbl>       <int> <int> <chr>
## 1  14.0         350   165 US.
## 2  31.9          89    71 Europe.
## 3  17.0         302   140 US.
## 4  15.0         400   150 US.
## 5  30.5          98    63 US.
## 6  23.0         350   125 US.
```

Per tal de demostrar les diferents tècniques farem que el *dataframe* `my_cars` tingui valors desconeguts:

```
# Fem certs valors desconeguts
my_cars[2, 2] <- NA
my_cars[4, 4] <- NA
head(my_cars)
```

```
## # A tibble: 6 x 4
##     mpg cubicinches    hp brand
##   <dbl>       <int> <int> <chr>
## 1  14.0         350   165 US.
## 2  31.9          NA    71 Europe.
## 3  17.0         302   140 US.
## 4  15.0         400   150 <NA>
```

```
## 5  30.5          98    63 US.
## 6  23.0         350   125 US.
```

Tot seguit, es mostra com reemplaçar els valors desconeguts amb constants:

```
# Amb l´ajuda de un test lògic descobrim els valors desconeguts
missing_values_cubicinches <- is.na(my_cars$cubicinches)
missing_values_brand <- is.na(my_cars$brand)

# Reemplacem els valors desconeguts amb constants
my_cars$cubicinches[missing_values_cubicinches] <- 0
my_cars$brand[missing_values_brand] <- "Valor desconegut"
head(my_cars)
```

```
## # A tibble: 6 x 4
##     mpg cubicinches    hp brand
##   <dbl>       <dbl> <int> <chr>
## 1  14.0       350     165 US.
## 2  31.9         0      71 Europe.
## 3  17.0       302     140 US.
## 4  15.0       400     150 Valor desconegut
## 5  30.5        98.0    63 US.
## 6  23.0       350     125 US.
```

Reemplaçar el valor desconegut amb la mitjana

A continuació es mostra un exemple de com reemplaçar valors desconeguts amb la mitjana:

```
# Reemplacem els valors desconeguts amb la mitjana
my_cars$cubicinches[missing_values_cubicinches] <- mean(na.omit(my_
cars$cubicinches))
head(my_cars)
```

```
## # A tibble: 6 x 4
##     mpg cubicinches    hp brand
##   <dbl>       <dbl> <int> <chr>
## 1  14.0       350     165 US.
## 2  31.9       202      71 Europe.
```

```
## 3  17.0        302     140 US.
## 4  15.0        400     150 <NA>
## 5  30.5         98.0    63 US.
## 6  23.0        350     125 US.
```

Reemplaçar el valor desconegut amb el valor més freqüent

A diferència de altres mesures estadístiques, R no proporciona una funció definida per al càlcul de la moda. Es per això, que crearem una funció per a calcular el valor més freqüent en un conjunt de dades. Aquesta funció pren com a argument un vector i retorna el valor més freqüent:

```
# Funció per al càlcul de la moda
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}

# Trobem el valor més freqüent
moda <- getmode(my_cars$brand)
moda

## [1] "US."

# Reemplacem els valors desconeguts amb la moda
my_cars$brand[missing_values_brand] <- moda
head(my_cars)

## # A tibble: 6 x 4
##     mpg cubicinches    hp brand
##   <dbl>       <dbl> <int> <chr>
## 1  14.0        350     165 US.
## 2  31.9        202      71 Europe.
## 3  17.0        302     140 US.
## 4  15.0        400     150 US.
## 5  30.5         98.0    63 US.
## 6  23.0        350     125 US.
```

Reemplaçar amb un valor aleàtori de la distribució de la variable

Per últim, a continuació es mostra un exemple de com reemplaçar amb un valor aleatori de la distribució de la variable:

```
# Generem observacions aleàtories
random_cubinches_obs <- sample(na.omit(my_cars$cubicinches), 1)
random_brand_obs <- sample(na.omit(my_cars$brand), 1)
random_cubinches_obs
```

```
## [1] 260
```

```
random_brand_obs
```

```
## [1] "US."
```

```
# Reemplacem els valors desconeguts amb les observacions aleàtorie
s
my_cars$cubicinches[missing_values_cubicinches] <- random_cubinches
_obs
my_cars$brand[missing_values_brand] <- random_brand_obs
head(my_cars)
```

```
## # A tibble: 6 x 4
##     mpg cubicinches    hp brand
##   <dbl>       <dbl> <int> <chr>
## 1  14.0       350     165 US.
## 2  31.9       260      71 Europe.
## 3  17.0       302     140 US.
## 4  15.0       400     150 US.
## 5  30.5        98.0    63 US.
## 6  23.0       350     125 US.
```

Exercici 4

A partir del joc de dades disponible en el següent enllaç: http://archive.ics.uci.edu/ml/datasets/Adult realitza un estudi similar al que se ha realitzat amb el joc de dades "Titanic". Explica el procés que has seguit, quin coneixement has extret, quins objectius t'havies fixat i quins passos i tècniques has emprat.

Carrega i exàmen preliminar del conjunt de dades

En primer lloc, instal.larem el paquet `readr`[8] que forma part del ecosistema `tidyverse`[9] i que ens permetrà llegir les dades:

```
# La forma més sencilla de instal.lar readr es instal.lar tidyvers
e
##install.packages("tidyverse")

# Alternativament, podem instal.lar només readr
##install.packages("readr")
```

[8] Paquet per a la lectura de dades amb format rectangular: https://readr.tidyverse.org/

[9] La notació `paquet::funció` és la forma explicita de cridar una funció. Amb la funció `dplyr::mutate_if()` haguéssim pogut canviar totes les columnes.

Un cop instal.lat el paquet el carregarem a la sessió R mitjançant la següent línia de codi:

```
# Carrega de readr
##library(readr)

# Alternativament, com que forma part de tidyverse
library(tidyverse)
```

Observem que, hem fet ús de la segona opció que carrega tots els paquets de `tidyverse`, ja que utilitzarem per a la realització de la pràctica altres paquets, com per exemple: `dplyr` (per a la transformació de dades),`tibble` (per a un tractament més refinat de `data.frames`), `ggplot2` (per a la visualització de les dades), etc.

Un cop carregat el paquet a la sessió R, ja podem fer ús de les funcions. Per a importar les dades des de l´adreça utilitzarem la funció `read_csv()`:

```
# Llegim les dades
adult <- read_csv("http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data")
```

Convertim el conjunt de dades `adult` que és del tipus `data.frame` a `tibble`:

```
# Convertim el dataframe a tibble
as_tibble(adult)
```

```
## # A tibble: 32,560 x 15
##     `39` `Stat~ `77516` Bache~  `13` `Neve~ `Adm-~ `Not~ White
Male  `217~
##    <int> <chr>    <int> <chr>  <int> <chr>  <chr>  <chr> <chr>
<chr> <int>
##  1    50 Self-~   83311 Bache~    13 Marri~ Exec-~ Husb~ White
Male      0
##  2    38 Priva~  215646 HS-gr~     9 Divor~ Handl~ Not-~ White
```

```
Male      0
##  3    53 Priva~  234721 11th       7 Marri~ Handl~ Husb~ Black
Male      0
##  4    28 Priva~  338409 Bache~    13 Marri~ Prof-~ Wife  Black
Fema~     0
##  5    37 Priva~  284582 Maste~    14 Marri~ Exec-~ Wife  White
Fema~     0
##  6    49 Priva~  160187 9th        5 Marri~ Other~ Not-~ Black
Fema~     0
##  7    52 Self-~  209642 HS-gr~     9 Marri~ Exec-~ Husb~ White
Male      0
##  8    31 Priva~   45781 Maste~    14 Never~ Prof-~ Not-~ White
Fema~ 14084
##  9    42 Priva~  159449 Bache~    13 Marri~ Exec-~ Husb~ White
Male   5178
## 10    37 Priva~  280464 Some-~    10 Marri~ Exec-~ Husb~ Black
Male      0
## # ... with 32,550 more rows, and 4 more variables: `0` <int>, `
40` <int>,
## #   `United-States` <chr>, `<=50K` <chr>
```

Podem adonar-nos que, el conjunt de dades esta format per 32.560 observacions i 15 variables. A més, amb l´ajuda de `tibble` també podem observar el tipus per a cada columna.

Com que el nom de les columnes es poc descriptiu per alguns dels atributs, personalitzarem els noms mitjançant la següent línia de codi:

```
# Noms dels atributs
names(adult) <- c("age","workclass","fnlwgt","education","education-num
","marital-status","occupation","relationship","race","sex","capital-gai
n","capital-loss","hour-per-week","native-country","income")
```

Podem comprovar el nom de les columnes mitjançant la funció `colnames`:

```
# Comprovem es nom de les columnes
colnames(adult)
```

```
##  [1] "age"            "workclass"      "fnlwgt"         "educat
ion"
##  [5] "education-num"  "marital-status" "occupation"     "relati
onship"
##  [9] "race"           "sex"            "capital-gain"   "capita
l-loss"
## [13] "hour-per-week"  "native-country" "income"
```

Exploració i tractament de valors desconeguts

En tercer lloc, ens caldria comprovar que el nostre conjunt de dades no conté valors desconeguts:

```
# Estadístiques de valors buits.
sapply(adult, function(x) sum(is.na(x)))
```

```
##            age      workclass         fnlwgt      education  ed
ucation-num
##              0              0              0              0
0
## marital-status     occupation   relationship           race
sex
##              0              0              0              0
0
##   capital-gain   capital-loss  hour-per-week native-country
income
##              0              0              0              0
0
```

```
# Alternativament
colSums(is.na(adult))
```

```
##            age      workclass         fnlwgt      education  ed
ucation-num
##              0              0              0              0
0
## marital-status     occupation   relationship           race
sex
##              0              0              0              0
0
```

```
##   capital-gain   capital-loss  hour-per-week native-country
income
##              0              0              0              0
0
```

Passem a analitzar la variable `workclass` que representa la industria en que una persona està treballant:

```
# Resum dels valors que conté la variable workclass
unique(adult$workclass)
```

```
## [1] "Self-emp-not-inc" "Private"          "State-gov"
## [4] "Federal-gov"      "Local-gov"        "?"
## [7] "Self-emp-inc"     "Without-pay"      "Never-worked"
```

Com es pot observar la variable `workclass` conté el caràcter `?` per a representar valors desconeguts. Amb l´objectiu de fer aquest grup més descriptiu podríem canviar aquests valors per la constant `Unknown`:

```
# Amb l´ajuda de un test lògic descobrim els valors desconeguts
missing_values_workclass <- adult$workclass == "?"
# Reemplacem els valors desconeguts amb la  constant
adult$workclass[missing_values_workclass] <- "Unknown"
```

Discretització d´atributs

El següent pas seria discretitzar els atributs del nostre conjunt de dades en el cas de que fos necessari. Per a descobrir quines dades podrien ser discretitzades farem ús de la funció `apply()` que aplicarà la composició de les funcions `lenght(unique())` a cada columna retornant el nombre de observacions diferents per a cada variable:

```
# Per a quines variables tindria sentit un procés de discretitzaci
ó?
apply(adult,2, function(x) length(unique(x)))
```

```
##            age      workclass         fnlwgt      education  ed
ucation-num
##             73              9          21647             16
16
## marital-status     occupation   relationship           race
sex
##              7             15              6              5
2
##   capital-gain   capital-loss  hour-per-week native-country
income
##            119             92             94             42
2
```

Segons els resultats podríem discretitzar aquelles variables amb poques classes i canviar el seu tipus a `factor`, que és la manera que té R de tractar amb les variables de tipus categòric.

```
# Discretitzem les variables amb poques classes
cols <- c('workclass', 'education', 'marital-status', 'relationship', '
race',
          'sex', 'income')
adult <- mutate_at(adult, cols, as.factor)
adult
```

```
## # A tibble: 32,560 x 15
##      age workc~ fnlwgt educa~ `educ~ `marit~ occu~ rela~ race
sex   `cap~
##    <int> <fctr>  <int> <fctr>  <int> <fctr>  <chr> <fct> <fct>
<fct> <int>
##  1    50 Self-~  83311 Bache~     13 Marrie~ Exec~ Husb~ White
Male      0
##  2    38 Priva~ 215646 HS-gr~      9 Divorc~ Hand~ Not-~ White
Male      0
##  3    53 Priva~ 234721 11th        7 Marrie~ Hand~ Husb~ Black
Male      0
##  4    28 Priva~ 338409 Bache~     13 Marrie~ Prof~ Wife  Black
Fema~     0
##  5    37 Priva~ 284582 Maste~     14 Marrie~ Exec~ Wife  White
Fema~     0
##  6    49 Priva~ 160187 9th         5 Marrie~ Othe~ Not-~ Black
```

```
Fema~     0
##  7    52 Self-~ 209642 HS-gr~      9 Marrie~ Exec~ Husb~ White
Male      0
##  8    31 Priva~  45781 Maste~     14 Never-~ Prof~ Not-~ White
Fema~ 14084
##  9    42 Priva~ 159449 Bache~     13 Marrie~ Exec~ Husb~ White
Male   5178
## 10    37 Priva~ 280464 Some-~     10 Marrie~ Exec~ Husb~ Black
Male      0
## # ... with 32,550 more rows, and 4 more variables: `capital-los
s` <int>,
## #   `hour-per-week` <int>, `native-country` <chr>, income <fctr
>
```

Fixe-mos amb el codi anterior que hem fet ús de la funció `dplyr::mutate_at`[10] per a convertir les columnes de tipus `character` al tipus `factor`.

Reducció de la dimensionalitat

Per a la simplificació de l´anàlisi les següents variables són descartades:

```
# Reducció del nombre d´atributs
adult$fnlwgt <- NULL
adult$education <- NULL
adult$relationship <- NULL
```

Els motius són els següents:

- El atribut `fnlwgt` no és prou descriptiu per si mateix i no disposem de documentació del conjunt de dades.

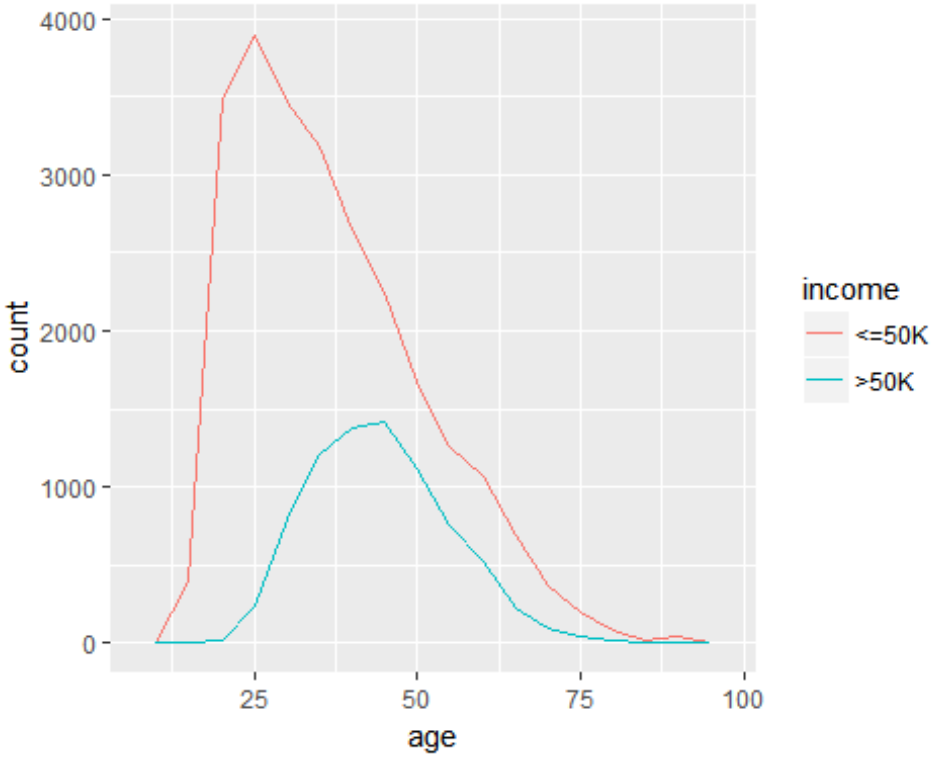
[10] La notació `paquet::funció` és la forma explicita de cridar una funció. Amb la funció `dplyr::mutate_if()` haguéssim pogut canviar totes les columnes.

- El atribut `education` pot ser el.limitat, ja que es pot conèixer pel nombre d´anys de formació acadèmica. En el conjunt de dades representat per la variable `education-num`.
- El atribut `relationship` pot ser el.limiat, degut a que es pot estimar a partir del gènere i l´estat civil. En el conjunt de dades representat per `marital-status` i `sex`, respectivament.

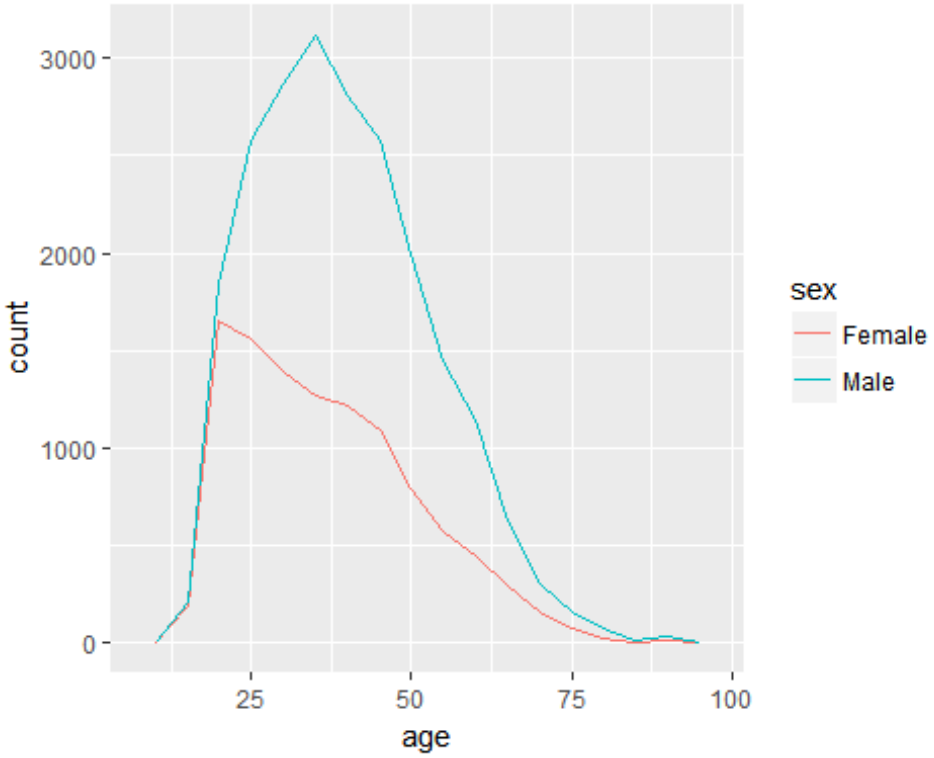
Exploració visual de les dades (EDA)

La primera variable de l´anàlisi és `age` que es tracta d´una variable continua. Podríem realitzar els següents histogrames que analitzen la correlació entre aquesta variable i la variable categòrica `income`:

```
# Histograma de la edat per grup d´ingresos
ggplot(data = adult, mapping = aes(x = age)) +
  geom_freqpoly(mapping = aes(color = income), binwidth = 5)
```




```r
# Histograma de la edat per grup de gènere
ggplot(data = adult, mapping = aes(x = age)) +
  geom_freqpoly(mapping = aes(color = sex), binwidth = 5)
```



El primer histograma ens indica que la majoria d´observacions reben una retribució per baix de 50.000$ a l´any. A més, aquells que reben una remuneració de més de 50.000 es troben a la mitat de la seva carrera professional.

Per altra banda, en el segon histograma el pot apreciar que les dones de qualsevol edat reben menys ingressos que els homes. També es pot observar que aquesta diferencia augmenta a mesura que són més grans.

Passem a analitzar la variable `workclass` que representa la industria en que una persona està treballant:

```
summary(adult$workclass)
```

```
##      Federal-gov        Local-gov     Never-worked          Pri
vate
##              960             2093                7             2
2696
```

```
##     Self-emp-inc Self-emp-not-inc         State-gov           Unk
nown
##             1116             2541              1297
1836
##      Without-pay
##               14
```

Podem observar que existeixen dos grups petits, `Never-worked` i `Without-pay`. Podríem combinar aquests grups amb `Unknowm`. A més, aquells que treballen per al govern estan distribuïts als grups *federal*, *state* i *local*. Per a facilitar el anàlisi, agruparem aquestes classes en una sola que anomenarem `Government`.

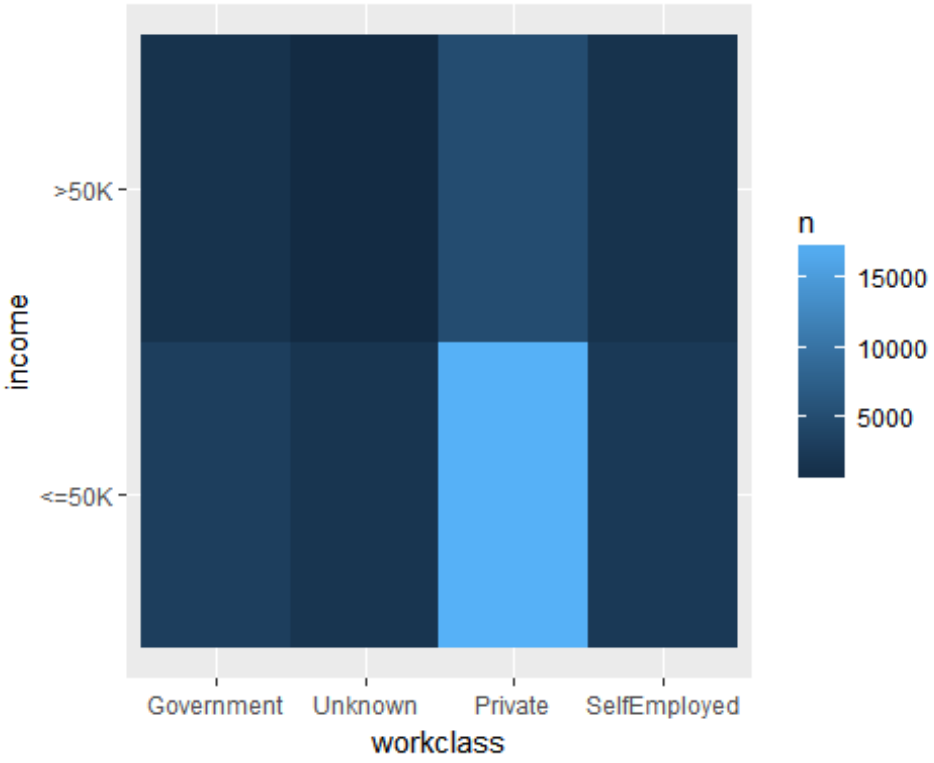
Per últim, aquells que són autonoms estàn distribuïts en *incorporated* i *not incorporated* i els combinarem en una variable amb el nom `Self-Employed`.

Per altra banda, cal anomenar que farem ús del paquet `forcats`que ens ajudarà a combinar les variables:

```
# Carreguem la llibreria
library(forcats)
# Combinem les classes
adult$workclass <- fct_collapse(adult$workclass,
                   Unknown = c("Never-worked", "Without-pay", "Unknowm
"),
                   Government = c("Federal-gov", "Local-gov", "State-g
ov"),
                   SelfEmployed = c("Self-emp-not-inc", "Self-emp-inc"
)
      )
```

Un cop reduït el nombre de classes ja podem comparar les variables `workclass` i `income`:

```
# Gràfic workclass vs income
adult %>%
  count(workclass, income) %>%
    ggplot(mapping = aes(x = workclass, y = income)) +
      geom_tile(mapping = aes(fill = n))
```



Bibliografia

[1] Daniel T. Larouse, Chantal D. Larouse: Data Mininig and Predictive Analytics.USA, John Wiley & Sons,2015,ISBN 978-1-118-11619-7

[2] Jordi Gironés Roig, Jordi Casas Roma, Julià Minguillón Alfonso, Ramon Caihuelas Quiles : Minería de Datos: Modelos y Algoritmos. Barcelona, Editorial UOC, 2017, ISBN: 978-84-9116-904-8.

[3] Jiawe Han, Michellie Chamber & Jian Pei: Data mining : concepts and techniques. 3º Edition. USA, Editorial Elsevier, 2012, ISBN 978-0-12-381479-1