A1 - Preprocés de dades

Xavier Vizcaino Gascon

21 de marzo, 2022

Contents

1. Càrrega de l'arxiu

Carregueu el fitxer de dades i examineu el tipus de dades amb què R ha interpretat cada variable. Examinar també els valors resum de cada tipus de variable.

Carreguem el fitxer de dades amb la següent comanda i generem un *dataset* que anomenarem **CIds**:

```
CIds <- read.csv2("CensusIncomedataset.csv", stringsAsFactors=TRUE)
```

S'ha utilitzat *read.csv2* ja que el separador és el punt i coma (;). També s'ha utilitzat la opció *stringsAsFactors=TRUE*, doncs permet tenir una primera visió dels *strings* repetits en diferents registres.

Examinem el tipus de dades amb que R ha interpretat cada variable, per fer-ho apliquem a través de *sapply()* la funció *class()* en tot el *dataset*.

```
sapply(CIds,class)
```

```
##          CS_ID            age      workclass         fnlwgt  education_num
##       "factor"      "integer"       "factor"      "integer"      "integer"
## marital_status   relationship     occupation           race            sex
##       "factor"       "factor"       "factor"       "factor"       "factor"
##   capital_gain   capital_loss hours_per_week         income
##      "integer"      "integer"       "factor"       "factor"
```

A continuació examinem els valors resum, de cada tipus de variable amb la funció *summary()* aplicada a tot el *dataset*:

```
summary(CIds)
```

```
##      CS_ID            age                  workclass         fnlwgt
##  CS23654:    2   Min.   : 17.00    Government   : 4349   Min.   :  12285
##  CS624  :    2   1st Qu.: 28.00    Other/Unknown: 1855   1st Qu.: 117824
##  CS7163 :    2   Median : 37.00    Private      :22692   Median : 178356
##  CS7453 :    2   Mean   : 38.58    Self-Employed: 3657   Mean   : 189775
##  CS8017 :    2   3rd Qu.: 48.00   NA's          :    7   3rd Qu.: 237046
##  CS8087 :    2   Max.   :650.00                          Max.   :1484705
##  (Other):32548   NA's   :7
##  education_num      marital_status          relationship
##  Min.   : 1.00    Divorced : 4443   Husband       :13192
##  1st Qu.: 9.00    Married  :15417   Not-in-family : 8303
##  Median :10.00    Separated: 1025   Other-relative:  981
##  Mean   :10.08    Single   :10682   Own-child     : 5067
##  3rd Qu.:12.00    Widowed  :  993   Unmarried     : 3444
##  Max.   :16.00                      Wife          : 1566
##  NA's   :7                          NA's          :    7
##           occupation                    race           sex
##   Blue-Collar  :10060    Amer-Indian-Eskimo:  311    Male  :19789
##   Other/Unknown: 1850    Asian-Pac-Islander: 1039    Female: 8771
##   Professional : 4139    Black             : 3123    female: 1100
##   Sales        : 3649    Other             :  271    male  : 1100
##   Service      : 5021    White             :27809    F     :  500
##   White-Collar : 7834   NA's               :    7    m     :  500
##  NA's          :    7                                (Other):  800
##    capital_gain    capital_loss     hours_per_week                  income
##  Min.   :    0   Min.   :   0.00   40,5 h : 8325   52,37 Milers d'euros:   32
##  1st Qu.:    0   1st Qu.:   0.00   40 h   : 6891   52,71 Milers d'euros:   32
##  Median :    0   Median :   0.00   50,5 h : 1575   54,02 Milers d'euros:   30
##  Mean   : 1077   Mean   :  87.31   50 h   : 1244   54,35 Milers d'euros:   29
##  3rd Qu.:    0   3rd Qu.:   0.00   45,5 h : 1039   52,27 Milers d'euros:   28
##  Max.   :99999   Max.   :4356.00   60,5 h :  830   53,76 Milers d'euros:   28
##                                    (Other):12656   (Other)             :32381
```

En aquest punt, comprovem que ,durant la importació de dades amb la utilització de l'opció *stringsAsFactors=TRUE*, totes les variables que contenen cadenes de text s'han carregat com a *factor*.

Veiem que tenim algunes variables de tipus *factor* amb un nombre elevat de *levels* fet que indica que aquest tipus no és el més adequat i hauriem de canviar el tipus de variable a *char*.

Així doncs, realitzem les següents modificacions addicionals. Primer, generant un vector amb els noms de les variables a modificar i posteriorment recorrent amb un *for* cada una de les variables realitzant la modificació amb la funció *as.character()*:

```
#Vector de variables a modificar
t_vector<-c("CS_ID","hours_per_week","income")

#Loop
for (i in t_vector){
  #Canvi de tipus a char
  CIds[,i]<-as.character(CIds[,i])
}
```

```
#Analitzem novament la classe de cada variable
sapply(CIds,class)
```

```
##          CS_ID            age      workclass         fnlwgt  education_num
##    "character"      "integer"       "factor"      "integer"      "integer"
## marital_status   relationship     occupation           race            sex
##       "factor"       "factor"       "factor"       "factor"       "factor"
##   capital_gain   capital_loss hours_per_week         income
##      "integer"      "integer"    "character"    "character"
```

2. Obtenció del *dataset* per fer l'estudi

En aquest cas, es vol eliminar les variables fnlwgt, capital_gain i capital_loss a més de, els registres amb més de 5 valors NAs. Per altra part, es poden crear noves variables en funció de les disponibles. En aquest cas, es crearà la variable education_cat que categoritza la formació acadèmica en formació primaria si education_num és menor de 7 anys, secundaria si education_num esta entre 7 i 9 anys, universitària si education_num esta entre 10 i 13 anys i postuniversitaria si education_num és major de 13 anys. Per últim, es vol canviar el nom de la variable sex per gender.

Eliminar variables

Inicialment el *dataset* té 14 variables (columnes).

```
ncol(CIds)
```

```
## [1] 14
```

Eliminem les variables sol · licitades (*fnlwgt, capital_gain* i *capital_loss*), re-assignant al *dataset* totes les variables excepte les mencionades.

Per fer-ho considerem novament un vector amb el nom de les variables a eliminar. En aquest cas, però, no utilitzem un *for* per fer les modificacions si no que ho fem a traves de l'assignació del complementari (operació més eficient). Finalment es comprova novament el nombre de columnes del *dataset* per validar els canvis.

```
#Vector de variables a modificar
t_vector<-c("fnlwgt","capital_gain","capital_loss")

#Reasignació del complementari
CIds<-CIds[ ,!(names(CIds) %in% t_vector)]

#Reportem numero de variables
ncol(CIds)
```

```
## [1] 11
```

Després de la modificació, el *dataset* té 11 variables.

Eliminar registres

Per a tenir un punt de partida, amb la següent comanda s'informa que inicialment el *dataset* té 32560 registres (files).

```
nrow(CIds)
```

```
## [1] 32560
```

L'eliminació dels registres amb més de 5 valors = NA, es realitza novament a través de l'operació d'assignació. Així doncs re-assignem només amb aquells registres que no tenen més de 5 NA al *dataset* i mostrem el nombre de registres després de la modificació.

```
CIds<-CIds[!rowSums(is.na(CIds))>5,]
nrow(CIds)
```

```
## [1] 32553
```

Ara el *dataset* té 32553 registres.

Variable categòrica

Per tal de categoritzar la formació acadèmica en funció de la variable *education_num*, creem una nova variable i assignem els valors d'acord als *levels* proposats en l'anunciat a través de la combinació de múltiples funcions *ifelse()*:

```
CIds$education_cat<-as.factor(ifelse(CIds$education_num<7,"primaria",
                              ifelse(CIds$education_num<=9,"secundaria",
                              ifelse(CIds$education_num<=13,"universitaria",
                                                            "postuniversitaria"))))
ncol(CIds)
```

```
## [1] 12
```

Ara el *dataset* té 12 variables.

```
summary(CIds$education_cat)
```

```
## postuniversitaria          primaria        secundaria     universitaria
##              2712              2644             12106             15091
```

Canvi nom variable

Finalment, canviem el nom de la variable *sex* a *gender* a través de la funció *names()* identificant la variable on es vol canviar el nom i assignant el nou nom.

```
names(CIds)[names(CIds) == "sex"] <- "gender"
colnames(CIds)
```

```
##  [1] "CS_ID"          "age"            "workclass"      "education_num"
##  [5] "marital_status" "relationship"   "occupation"     "race"
##  [9] "gender"         "hours_per_week" "income"         "education_cat"
```

3. Duplicació de codis

Verifiqueu la consistència en la variable CS_ID. Si hi ha registres duplicats, assigneu un nou codi per evitarcodis duplicats. El nou codi ha de ser un valor no usat (valors superiors al màxim valor numèric contingut en CS_ID). Conserveu el mateix format que la resta de codis, amb "CS" davant de la seqüència numèrica. Podeu utilitzar la funció duplicated d'R per detectar els duplicats.

El *dataset* té 7 registres on la variable *CS_ID* està duplicada.

Per a resoldre la duplicació de codis d'acord als requeriments de l'anunciat, comencem per analitzar el valor màxim de la part numèrica de la variable *CS_ID* utilitzant les funcions *substring()*, *as.integer()* i *max()*.

Continuem amb la creació de dos sub-*datasets* temporals, un amb els registres on la variable *CS_ID* no està repetida i l'altre amb el complementari, és a dir amb els registres on la variable *CS_ID* si està repetida, a través de la funció *duplicated()*.

Calculem els valors de *CS_ID* inicial i final que s'han d'assignar als registres del *dataset* de repetits partint del valor màxim de *CS_ID* inicial i el número de registres repetits trobats. Amb aquests valors generem

un vector que posteriorment, després de donar-li el format de la variable *CS_ID* fent ús de les funcions *as.character()* i *paste()* s'imputa en la variable *CS_ID* del sub-*dataset* de repetits.

Finalment ajuntem per files els dos sub-*datasets* amb la funció *rbind()* i assignem el resultat al identificador de *dataset* que s'ha utilitzat fins ara **CIds**. Comprovem que no tenim duplicats en la variable *CS_ID* i mostrem alguns registres del final del *dataset* per validar el procés.

```
#Obtenim el valor màxim del codi
t_maxcodis<-max(as.integer(substring(CIds$CS_ID,3,)))

#Separem el dataset en 2 sub-datasets
t_repetits<-CIds[duplicated(CIds$CS_ID),]
t_norepetits<-CIds[!duplicated(CIds$CS_ID),]

#Extraiem el numero de registres repetits i els valor numèrics dels nous codis inicial i final
t_numrep<-nrow(t_repetits)
t_inici<-t_maxcodis+1
t_final<-t_maxcodis+t_numrep

#Generem un vector amb els nous codis per als registres repetits
t_ID_vector<-t_inici:t_final

#Formatem el vector per a complir amb el format de ID del dataset
t_ID_vector<-as.character(t_ID_vector)
t_ID_vector<-paste("CS",t_ID_vector, sep="")

#Asignem els valors del ID al sub-dataset de repetits i combinem els 2 sub-datasets
t_repetits$CS_ID<-t_ID_vector
CIds<-rbind(t_norepetits,t_repetits)

#Comprovem duplicats
sum(duplicated(CIds$CS_ID))
```

```
## [1] 0
```

```
#Comprovem registres finals del dataset
tail(CIds[,c(1,2,3)],15)
```

```
##         CS_ID age      workclass
## 32553 CS32553  32        Private
## 32554 CS32554  53        Private
## 32555 CS32555  22        Private
## 32556 CS32556  27        Private
## 32557 CS32557  40        Private
## 32558 CS32558  58        Private
## 32559 CS32559  22        Private
## 32560 CS32560  52  Self-Employed
## 624   CS32561  47        Private
## 7163  CS32562  39        Private
## 7453  CS32563  31        Private
## 8017  CS32564  33        Private
## 8087  CS32565  30        Private
## 9197  CS32566  47  Other/Unknown
## 23654 CS32567  34  Self-Employed
```

4. Normalització de les dades qualitatives

4.1. Eliminació d'espais en blanc

S'ha observat que hi espais en blanc l'inici dels valors a les variables qualitatives. Per tant, cal eliminar aquest espais en blancs.

De manera anàloga als canvis addicionals del apartat 1, generem un vector amb les variables on aplicar les modificacions i les recorrem amb un *loop for*. Per tal d'eliminar els espais en blanc fem ús de la funció *trimws()*. Cal destacar que s'ha de garantir que les dades d'entrada a la funció són de tipus *char*, així doncs es converteixen amb la funció *as.character()*.

El resultat de la eliminació d'espais en blanc es torna a convertir a *factor* amb la funció *as.factor()* i s'assigna a la variable corresponent del *dataset*.

```
#Vector de variables a modificar
t_vector<-c("workclass","marital_status","relationship","occupation","race","gender")

#Loop
for (i in t_vector){
  #Conversio a char, eliminació de blancs, reconversió a factor i assignació
  CIds[,i]<-as.factor(trimws(as.character(CIds[,i])))
}
```

4.2. Marital-status

Canviar les categories de la variable marital status actuals per altres que ocupin una caràcter. Els valors que s'assignaren a la variable marital_status són: M per Married, S per Single, X per Separated, D per Divorced, W per Widowed. Representeu gràficament la distribució dels valors de la variable.

Inicialment es comprova l'ordre de les categories de la variable *marital_status* amb la funció *summary()*. Posteriorment s'assignen les noves categories utilitzant la funció *levels()*. Finalment es comprova novament que les modificacions s'han realitzat correctament amb la funció *summary()* comparant els resultats finals amb els inicials.

```
summary(CIds$marital_status)
```

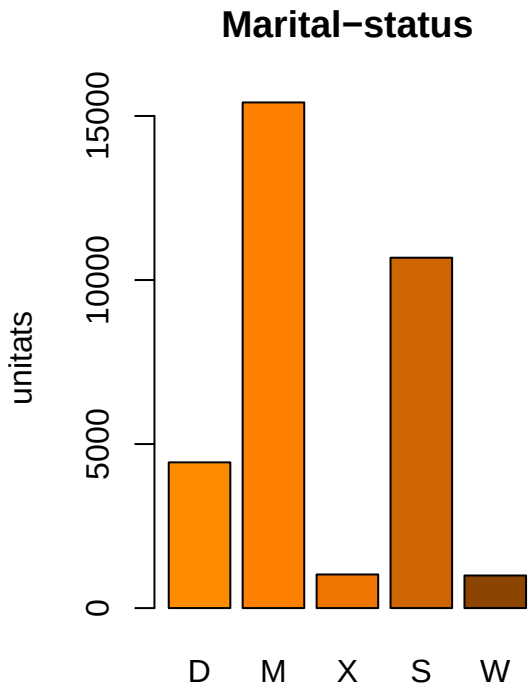
```
##  Divorced   Married Separated    Single   Widowed
##      4442     15413      1025     10680       993
```

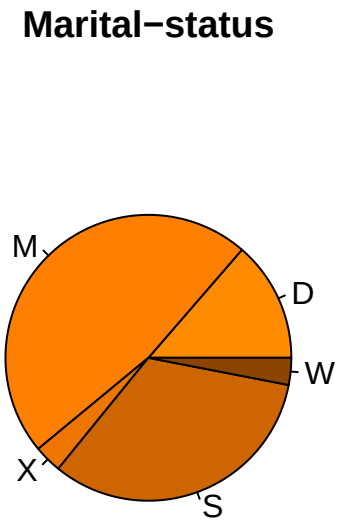
```
levels(CIds$marital_status)<-c("D","M","X","S","W")
summary(CIds$marital_status)
```

```
##     D     M     X     S     W
##  4442 15413  1025 10680   993
```

Els gràfics de la següent pàgina mostren, a l'esquerra, en gràfic de barres, el nombre de registres per cada categoria de la variable *marital_status* i a la dreta en diagrama de sectors la freqüència relativa de cada categoria respecte la mostra. Els gràfics es generen amb les funcions *plot()* i *pie()* respectivament.

```
par(mfrow=c(1,2))
plot(CIds$marital_status, main="Marital-status", col = mypalette,ylab="unitats")
pie(summary(CIds$marital_status),main = "Marital-status", col = mypalette)
```





4.3. Gènere

Reviseu la consistència dels valors de la variable gender i feu les modificacions oportunes per indicar les categories finals com f i m que correspon a femeni i masculi, respectivament. Representeu gràficament la distribució dels valors de la variable.

Inicialment es comprova la quantitat de categories de la variable *gender* amb la funció *summary()*. S'observa que existeixen errors sintàctics que introdueixen un nombre més elevat de categories. Es procedeix a reparar-los amb l're-assignació de les categories a valors de base "f" i "m" utilitzant la funció *levels()*. Finalment es comprova novament que les modificacions s'han realitzat correctament amb la funció *summary()* comparant els resultats finals amb els inicials.

```
summary(CIds$gender)
```

```
##      F    Fem female Female      m      M   male   Male
##    500    400   1100   8767    499    400   1100  19787
```

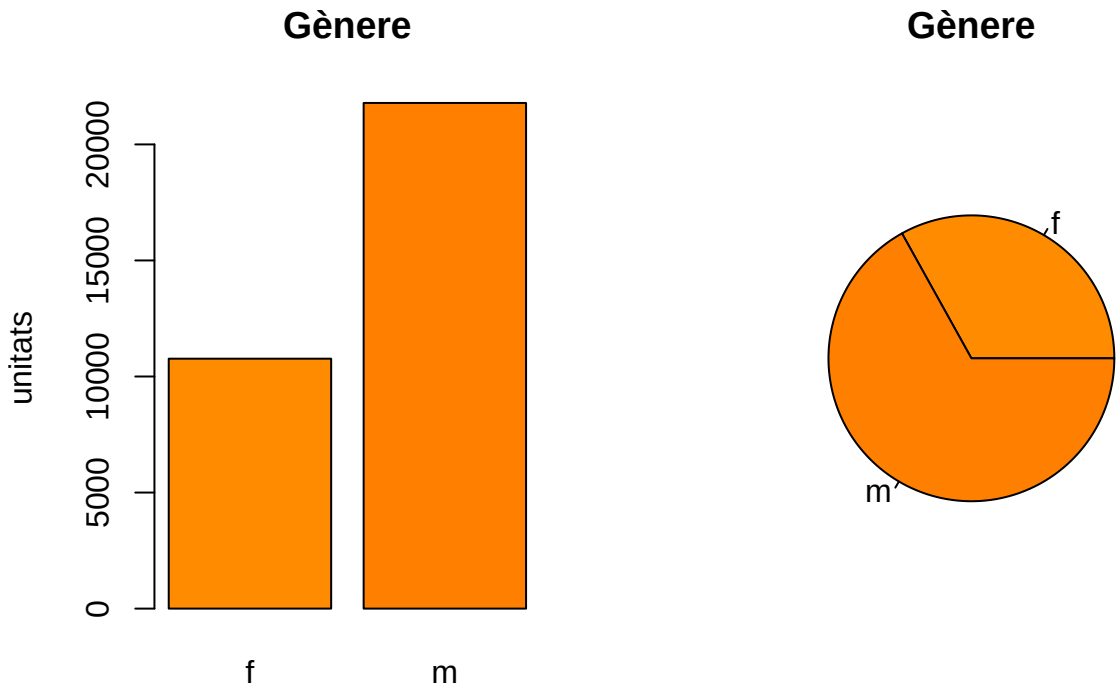
```
levels(CIds$gender)<-c("f","f","f","f","m","m","m","m")
summary(CIds$gender)
```

```
##     f     m
## 10767 21786
```

Els gràfics de la pàgina següent mostren, a l'esquerra, en gràfic de barres, el nombre de registres per cada categoria de la variable *gender* i a la dreta, en diagrama de sectors la freqüència relativa de cada categoria respecte la mostra.

```
par(mfrow=c(1,2))
plot(CIds$gender, main="Gènere", col = mypalette, ylab="unitats")
```

```
pie(summary(CIds$gender),main = "Gènere", col = mypalette)
```



5. Normalització de les dades quantitatives

5.1. Edat

Reviseu el format de la variable age i feu les transformacions oportunes segons els criteris especificats anteriorment: han de ser de tipus sencer, sense decimals.

Es comprova que la variable *age* sigui de tipus *integer* amb la funció *is.integer()*. I es valida que no es necessari realitzar modificacions.

```
is.integer(CIds$age)
```

```
## [1] TRUE
```

5.2. Educació

Reviseu el format de la variable education_num i feu les transformacions oportunes segons els criteris especificats anteriorment: han de ser de tipus sencer, sense decimals.

Es comprova que la variable *education_num* sigui de tipus *integer* amb la comanda *is.integer()*. I es valida que no es necessari realitzar modificacions.

```
is.integer(CIds$education_num)
```

```
## [1] TRUE
```

5.3. Hores per setmana

Reviseu el format de la variable hours_per_week i feu les transformacions oportunes segons els criteris especificats anteriorment: En les dades numèriques, el símbol de separador decimal és el punt i no la coma. A més, si es presenta la unitat de la variable, per exemple hores en el cas de hours_per_week, cal eliminar per convertir la variable a tipus numèric.

Per tal de complir amb els requisits de l'anunciat s'ha de treure la unitat "h" de cada un dels registres. També canviar la coma (,) pel punt (.) i convertir el tipus de variable a numèrica. Així doncs es canvia el *substring* " h" per " " (conjunt buit) en els registres de la variable *hours_per_week* amb la funció *sub()*. A continuació, també amb la funció *sub()* es canvia la coma "," pel punt ".", també en tots els registres de la mateixa variable. Per acabar, es converteix el tipus de variable a numèric amb decimals amb la funció *as.double()*. Es mostren els 100 primers valors de la variable amb la funció *head()* i se'n fa un resum amb la funció *summary()*. Cal destacar que totes les conversions es re-assignen a la mateixa variable del *dataset*, *hours_per_week*.

```
#Modificacions
CIds$hours_per_week<-sub(" h","",CIds$hours_per_week)
CIds$hours_per_week<-sub(",",".",CIds$hours_per_week)
CIds$hours_per_week<-as.double(CIds$hours_per_week)

#Comprovacions
head(CIds$hours_per_week,100)
```

```
##   [1] 13.5 40.0 40.5 40.0 40.5 16.0 45.0 50.5 40.0 80.5 40.0 30.5 50.0 40.5 45.0
##  [16] 35.5 40.5 50.5 45.5 60.5 20.0 40.5 40.5 40.5 40.5 40.5 60.0 80.0 40.0 52.5
##  [31] 44.0 40.0 40.5 15.0 40.0 40.0 25.5 38.5 40.5 43.5 40.5 50.0 40.5 35.0 40.5
##  [46] 38.5 40.5 40.0 43.5 40.5 30.5 60.5 55.0 60.5 40.0 40.0 40.5 48.0 40.5 40.0
##  [61] 40.5 40.0 45.5 58.5 40.5 40.0 40.5 50.5 40.5 32.5 40.0 70.5 40.5 20.5 40.5
##  [76] 40.5  2.0 22.0 40.0 30.5 40.5 40.5 48.5 40.0 35.5 40.0 50.0 40.5 50.0 40.0
##  [91] 40.0 25.5 35.0 40.0 50.5 60.5 48.0 40.5 40.0 40.5
```

```
summary(CIds$hours_per_week)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00   40.00   40.50   40.71   45.50   99.50
```

5.3. Income

Reviseu el format de la variable income segons el criteri indicat més amunt: La variable 'income' s'ha d'expressar en milers d'euros (k€)

D'acord amb l'anunciat, la variable *income* ha de ser de tipus numèric i s'ha d'expressar en milers d'euros (k€). Procedim a visualitzar alguns registres de la variable *income* amb la funció *head()* per a conèixer les possibles modificacions a realitzar.

```
head(CIds$income,30)
```

```
##  [1] "2000,3 Milers d'euros" "51,67 Milers d'euros"  "50,08 Milers d'euros"
##  [4] "44,21 Milers d'euros"  "0,1 Milers d'euros"    "43,93 Milers d'euros"
##  [7] "2134,6 Milers d'euros" "38,24 Milers d'euros"  "57,12 Milers d'euros"
## [10] "49,72 Milers d'euros"  "51,08 Milers d'euros"  "44,34 Milers d'euros"
## [13] "41,98 Milers d'euros"  "46,53 Milers d'euros"  "42,5 Milers d'euros"
## [16] "51,16 Milers d'euros"  "51,95 Milers d'euros"  "47,88 Milers d'euros"
## [19] "44,38 Milers d'euros"  "55,01 Milers d'euros"  "35,87 Milers d'euros"
## [22] "49,36 Milers d'euros"  "52610 euros"           "38,3 Milers d'euros"
## [25] "56,66 Milers d'euros"  "47,19 Milers d'euros"  "42,7 Milers d'euros"
## [28] "55,67 Milers d'euros"  "53,18 Milers d'euros"  "55,48 Milers d'euros"
```

Observem que alguns registres ja estan expressats en "Milers d'euros" mentre que d'altres estan expressats en "euros". Per tant s'ha de diferenciar entre dos grups de dades per a realitzar les modificacions.

Primer grup, enfocat en els registres expressats en "Milers d'euros" en els que es substitueix els *substring* " Milers d'euros" per " " a través de la funció *sub()*.

Segon grup, enfocat en els registres expressats en "euros" . La modificació dels registres es realitza a través d'una *REGEX* que defineix un primer grup [Grup 1] de 0 o més dígits (\\d*), concatenat a un segon grup [Grup 2] que defineix exactament 3 dígits (\\d{3}) i concatenat al *substring* " euros". Aquesta estructura es substitueix pel [Grup 1] i el [Grup 2] separats per una coma "," amb la funció *sub()*. Amb aquesta modificació s'obté el resultat de passar "euros" a "k€" sense deixar de treballar amb cadenes de caràcters (*chr*).

Finalment, es canvia la coma "," pel punt "." de tots els registres novament amb la funció *sub()*, es canvia el tipus de variable a numèrica amb la funció *as.double()* i es validen les modificacions mostrant els 100 primers valors de la variable amb la funció *head()* i fent un resum amb la funció *summary()*

```
#Modificacions primer group en la variable income
CIds$income<-sub(" Milers d'euros","",CIds$income)

#Modificacions segon grup en la variable income
CIds$income<-sub("(\\d*)(\\d{3}) euros","\\1,\\2",CIds$income, fixed=FALSE)

#Modificacions tots els registres de la variable income
CIds$income<-sub(",",".",CIds$income)
CIds$income<-as.double(CIds$income)

#Comprovacions
head(CIds$income,100)
```

```
##   [1] 2000.30   51.67   50.08   44.21    0.10   43.93 2134.60   38.24   57.12
##  [10]   49.72   51.08   44.34   41.98   46.53   42.50   51.16   51.95   47.88
##  [19]   44.38   55.01   35.87   49.36   52.61   38.30   56.66   47.19   42.70
##  [28]   55.67   53.18   55.48   40.87   54.99   62.99   48.97   52.45   52.31
##  [37]   47.22   52.05   50.75   53.01   50.43   52.83   40.93   54.93   53.36
##  [46]   53.27   45.05   58.91   51.13   38.88   37.74   45.80   60.09   54.74
##  [55]   52.14   56.54   49.56   58.63   55.14   53.56   48.16   54.43   54.90
##  [64]   47.67   52.93   44.75   48.48   55.18   46.89   52.73   38.18   46.38
##  [73]   53.00   52.65   46.42   60.17   47.36   33.93   57.73   46.95   53.80
##  [82]   43.14   52.26   43.60   38.99   57.12   57.02   47.52   45.10   56.62
##  [91]   44.40   42.79   40.72   55.99   58.19   52.00   57.02   48.43   43.48
## [100]   57.77
```

```
summary(CIds$income)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.10   43.22   49.71   48.88   54.32 2134.60
```

6. Valors atípics

Reviseu si hi ha valors atípics en les variables age, education_num, hours_per_week i income. Si es tracta d'un valor anòmal, és a dir anormalment alt o baix, substituir el seu valor per NA, que posteriorment s'ha d'imputar.

Age

Per analitzar els valors atípics es mostren gràficament els registres de la variable *age* en un diagrama de caixa (esquerra), les estadístiques principals i el numero de registres que queden fora dels "bigotis". Aquestes operacions es realitzen a través de les funcions *boxplot*, el mètode *stats* de *boxplot* i la funció *length*.

Amb aquest anàlisi i aplicant sentit comú, es defineixen com a valors atípics aquells amb un valor en la variable *age* superior a 120 i conseqüentment se'ls assigna NA.

Finalment es genera un nou diagrama de caixa (dreta) i s'extreuen estadístiques de la variable *age* ja modificada amb les assignacions de NA per a valors atípics.

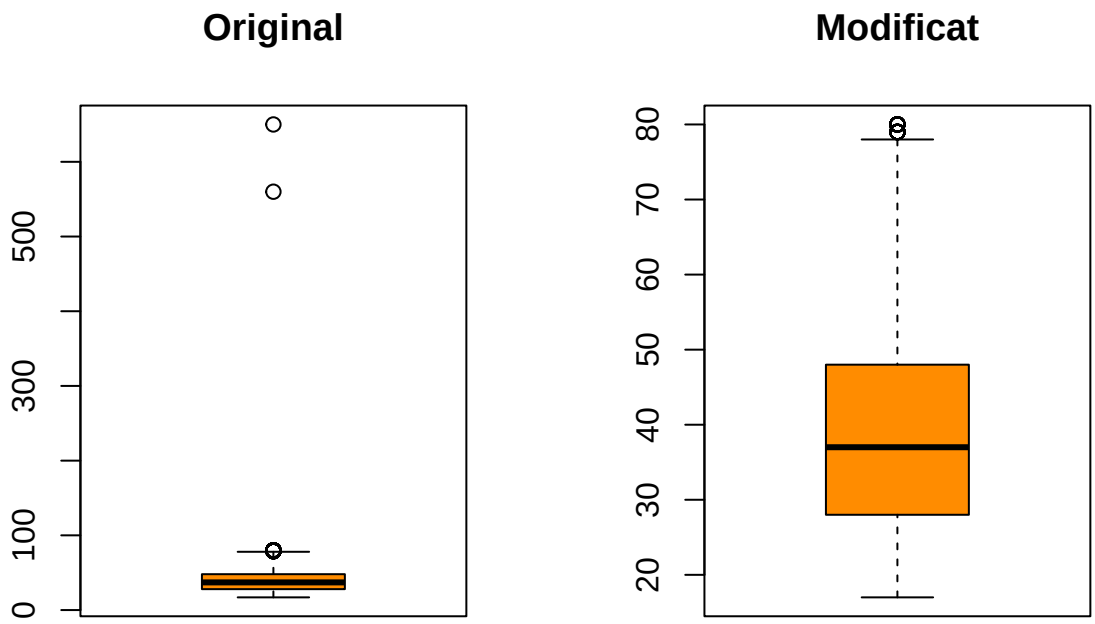
```
par(mfrow=c(1,2))

#Boxplot amb dades originals
boxplot(CIds$age, main="Original", col = mypalette)
boxplot.stats(CIds$age)$stats
```

```
## [1] 17 28 37 48 78
```

```
length(boxplot.stats(CIds$age)$out)
```

```
## [1] 44
```

```
#Considerem valors atípics valors d'edats > 120
CIds$age[CIds$age>120]<-NA

#Boxplot amb dades modificades
boxplot(CIds$age, main="Modificat", col = mypalette)
```




```
summary(CIds$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   17.00   28.00   37.00   38.55   48.00   80.00       2
```

Education num

Amb la variable *education_num* es procedeix de la mateixa manera que en la variable anterior. En aquest cas, observant el diagrama de caixa inicial s'interpreta que no hi ha valors atípics i per tant no es realitzen modificacions.

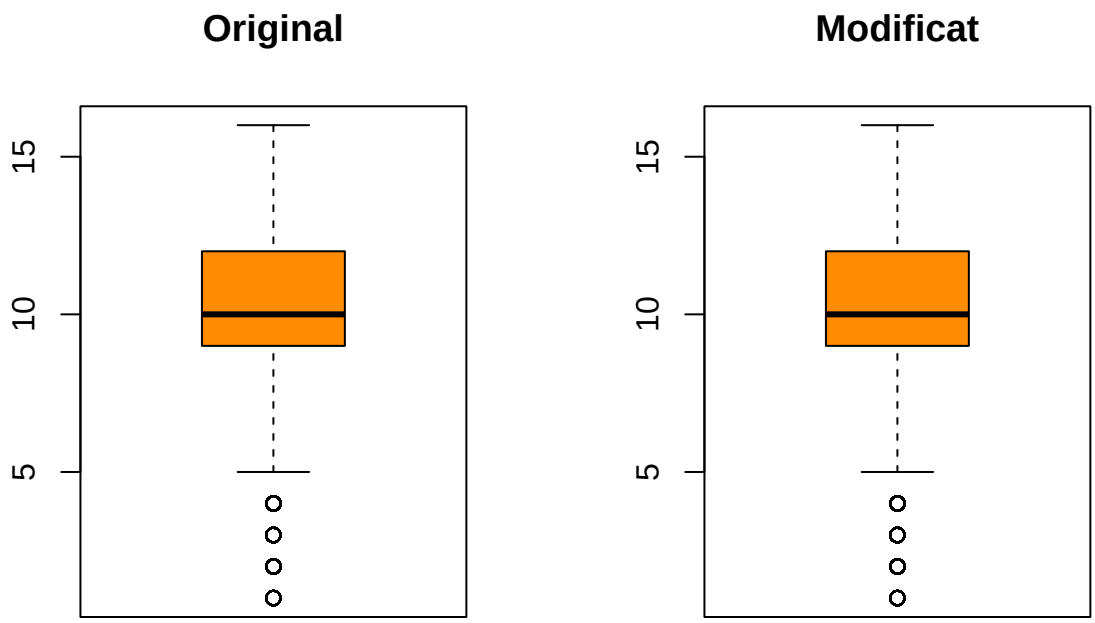
```
par(mfrow=c(1,2))

#Boxplot amb dades originals
boxplot(CIds$education_num, main="Original", col = mypalette)
boxplot.stats(CIds$education_num)$stats
```

```
## [1]  5  9 10 12 16
```

```
length(boxplot.stats(CIds$education_num)$out)
```

```
## [1] 1197
```

```
#Considerarem que en aquesta variable no hi ha valors atípics

boxplot(CIds$education_num, main="Modificat", col = mypalette)
```




```
summary(CIds$education_num)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    9.00   10.00   10.08   12.00   16.00
```

Hours per week

Per la variable *hours_per_week* es procedeix de la mateixa manera que amb les dues anteriors variables. En aquest cas, segons l'anunciat de l'activitat es defineixen com a valors atípics aquells amb un nombre *hours_per_week* superior a 80.

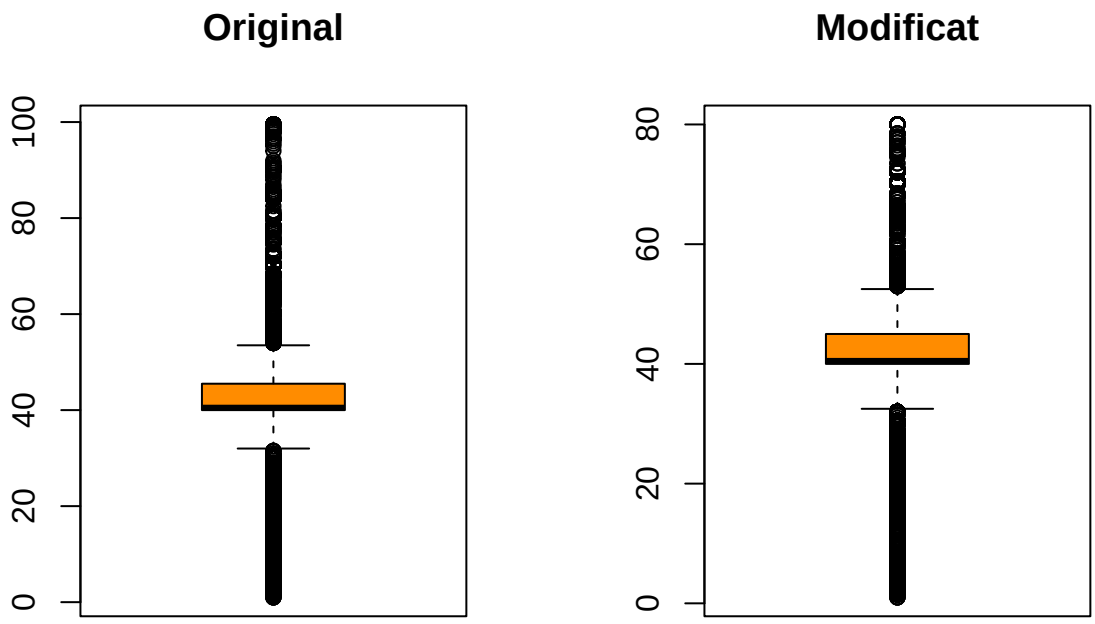
```
par(mfrow=c(1,2))

#Boxplot amb dades originals
boxplot(CIds$hours_per_week, main="Original", col = mypalette)
boxplot.stats(CIds$hours_per_week)$stats
```

```
## [1] 32.0 40.0 40.5 45.5 53.5
```

```
length(boxplot.stats(CIds$hours_per_week)$out)
```

```
## [1] 8715
```

```
#Es considera atípic valors d'hours_per_week > 80
CIds$hours_per_week[CIds$hours_per_week>80]<-NA

#Boxplot amb dades modificades
boxplot(CIds$hours_per_week, main="Modificat", col = mypalette)
```




```
summary(CIds$hours_per_week)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    1.00   40.00   40.50   40.29   45.00   80.00     277
```

Income

Finalment, per la variable *income* es procedeix de la mateixa manera que les variables anteriors. En aquest cas, a partir de l'observació del diagrama de caixa inicial es defineixen com a valors atípics aquells valors de la variable *income* que son superiors a 200 o inferiors a 5.

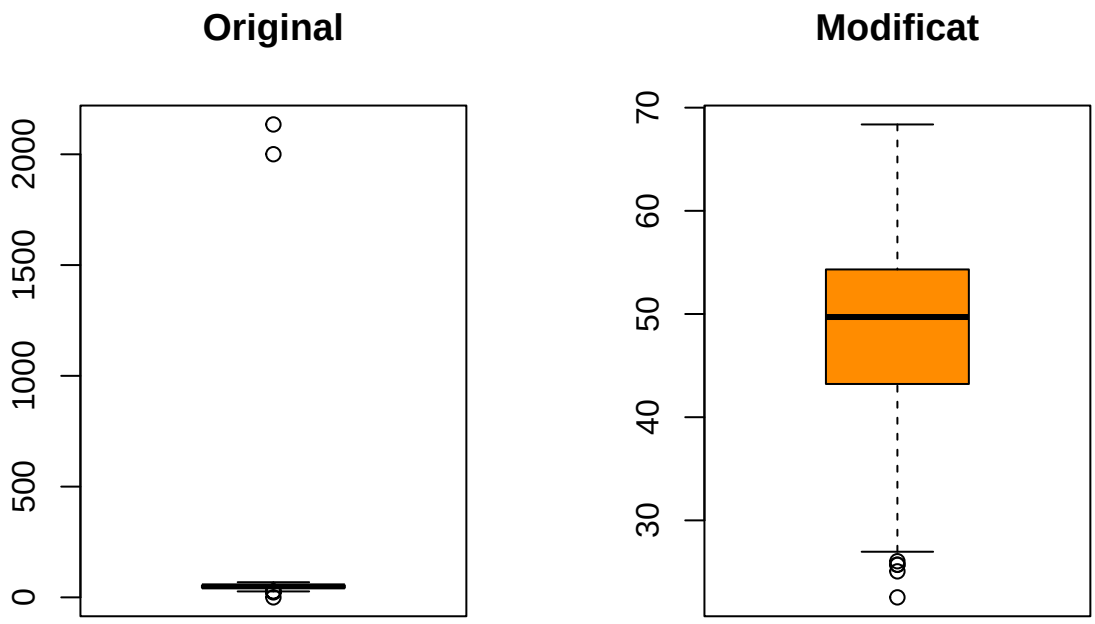
```
par(mfrow=c(1,2))

#Boxplot amb dades originals
boxplot(CIds$income, main="Original", col = mypalette)
boxplot.stats(CIds$income)$stats
```

```
## [1] 26.96 43.22 49.71 54.32 68.37
```

```
length(boxplot.stats(CIds$income)$out)
```

```
## [1] 8
```

```
#Es considera atípic valors d'income > 200 k€ o < 5k€
CIds$income[CIds$income>200 | CIds$income<5]<-NA

#Boxplot amb dades modificades
boxplot(CIds$income, main="Modificat", col = mypalette)
```




```
summary(CIds$income)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   22.54   43.22   49.71   48.75   54.32   68.37       3
```

7. Imputació de valors

Busqueu si hi ha valors perduts en les variables quantitatives age, education_num, hours_per_week i income. En cas de valors perduts, apliqueu el procés:

Age

En la variable 'age', apliqueu imputació per la mitjana aritmètica.

Inicialment hi ha 2 NA en la variable *age*.

Per al càlcul de la mitjana s'utilitza la funció *mean()* amb l'opció *na.rm=TRUE* per a no considerar els valors NA en el càlcul. El resultat s'assigna als registres NA de la variable *age* utilitzant la funció *is.na()*

```
#Imputació
CIds$age[is.na(CIds$age)]<-mean(CIds$age,na.rm=TRUE)

#Comprovació NA
sum(is.na(CIds$age))
```

```
## [1] 0
```

Després de la imputació, tenim 0 NA en la variable *age*.

Income

En la variable 'income', apliqueu imputació per la mitjana aritmètica dels registres del mateix gènere, és a dir, separat per gènere.

Inicialment hi ha 3 NA en la variable *income*.

L'operativa per a imputar valors es equivalent a l'explicada anteriorment amb la diferencia que s'ha de considerar la mitjana aritmètica dels registres del mateix gènere. Llavors l'operació es realitza en dos passos:

En el primer pas, es realitza la imputació en els registres de la variable *income* amb gènere masculí; fet que s'aconsegueix utilitzant dues expressions condicionals unides amb l'operador "&", que obliga que siguin certes les dues.

```
#Registres NA
CIds[is.na(CIds$income),c("CS_ID","income")]
```

```
##   CS_ID income
## 1   CS1     NA
## 5   CS5     NA
## 7   CS7     NA
```

```
#Imputació
CIds$income[is.na(CIds$income) & CIds$gender=="m"]<-
  mean(CIds[CIds$gender=="m","income"],na.rm = TRUE)
```

Després de la imputació de NAs en el pas anterior, hi ha 1 NA en la variable *income*. El segon pas es per a imputar en els registres de la variable *income* amb gènere femení. Aquesta imputació es anàloga a l'anterior canviant en *level* objectiu de la variable *gender* de "m" a "f".

```
#Registres NA
CIds[is.na(CIds$income),c("CS_ID","income")]
```

```
##   CS_ID income
## 5   CS5     NA
```

```
#Imputació
CIds$income[is.na(CIds$income) & CIds$gender=="f"]<-
  mean(CIds[CIds$gender=="f","income"],na.rm = TRUE)
```

```
#Comprovació NA
sum(is.na(CIds$income))
```

```
## [1] 0
```

Després de totes les imputacions, hi ha 0 NA en la variable *income*.

Hours per week

A la resta de variables, apliqueu imputació per veïns més propers, utilitzant la distància de Gower, considerant en el còmput dels veïns més propers la resta de variables quantitatives esmentades en aquest apartat. A més, considereu que la imputació s'ha de fer amb registres del mateix gènere. Per exemple, si un registre a imputar és de gènere "M", s'ha de realitzar la imputació usant les variables quantitatives dels registres de gènere "M". Per realitzar aquesta imputació, podeu fer servir la funció "kNN" de la llibreria VIM amb un nombre de veïns igual a 11.

Inicialment hi ha 277 NA en la variable *hours_per_week*. De la mateixa manera que en l'apartat anterior, es realitza la imputació de valors en dos passos. Primer en els registres de gènere "m" i posteriorment en els registres de gènere "f".

A continuació es mostren els 10 primers registres NA en la variable *hours_per_week* de genere "m" i s'assignen els seus codis o valors de la variable *CS_ID* en un vector temporal que es recuperarà posteriorment per a comprovar que les imputacions s'han realitzat correctament.

```
#Per a gènere == "m"
#10 primers valors a imputar
head(CIds[is.na(CIds$hours_per_week) & CIds$gender=="m",c("CS_ID","hours_per_week")],10)
```

```
##       CS_ID hours_per_week
## 10     CS10             NA
## 272   CS272             NA
## 935   CS935             NA
## 1066 CS1066             NA
## 1172 CS1172             NA
## 1200 CS1200             NA
## 1417 CS1417             NA
## 1730 CS1730             NA
## 1824 CS1824             NA
## 1887 CS1887             NA
```

```
#Vector codis CS_ID dels 10 primers valors a imputar de gènere "m"
mID<-head(CIds[is.na(CIds$hours_per_week) & CIds$gender=="m","CS_ID"],10)

#Per a gènere == "f"
#10 primers valors a imputar
head(CIds[is.na(CIds$hours_per_week) & CIds$gender=="f",c("CS_ID","hours_per_week")],10)
```

```
##         CS_ID hours_per_week
## 1272   CS1272             NA
## 2015   CS2015             NA
## 4294   CS4294             NA
## 4348   CS4348             NA
## 5432   CS5432             NA
## 5489   CS5489             NA
## 8072   CS8072             NA
## 8780   CS8780             NA
## 10728 CS10728             NA
```

```
## 10850 CS10850             NA
```

```
#Vector codis CS_ID dels 10 primers valors a imputar de gènere "f"
fID<-head(CIds[is.na(CIds$hours_per_week) & CIds$gender=="f","CS_ID"],10)
```

S'imputem NA per kNN en dos passos utilitzant la funció *kNN()* de la llibreria *VIM*; separant per gènere "m" i gènere "f". s'escull l'opció *imp_var = FALSE* per evitar que es generi una variable adicional i que es tingui un *Warning* informant que s'està realitzant una imputació de N+1 variables en un *dataset* de N variables.

```
#Imputació
CIds[CIds$gender=="m",]<-kNN(CIds[CIds$gender=="m",],
                             variable="hours_per_week",k=11, imp_var = F)
CIds[CIds$gender=="f",]<-kNN(CIds[CIds$gender=="f",],
                             variable="hours_per_week",k=11, imp_var = F)
```

Comprovem les imputacions recuperant el vector de codis per a gènere "m"

```
CIds[CIds$CS_ID %in% mID,c("CS_ID","hours_per_week")]
```

```
##       CS_ID hours_per_week
## 10     CS10           40.5
## 272   CS272           50.5
## 935   CS935           40.5
## 1066 CS1066           45.5
## 1172 CS1172           40.5
## 1200 CS1200           40.5
## 1417 CS1417           40.5
## 1730 CS1730           40.5
## 1824 CS1824           42.5
## 1887 CS1887           40.5
```

Comprovem les imputacions recuperant el vector de codis per a gènere "f"

```
CIds[CIds$CS_ID %in% fID,c("CS_ID","hours_per_week")]
```

```
##         CS_ID hours_per_week
## 1272   CS1272           38.0
## 2015   CS2015           40.5
## 4294   CS4294           40.5
## 4348   CS4348           40.0
## 5432   CS5432           40.5
## 5489   CS5489           40.5
## 8072   CS8072           40.0
## 8780   CS8780           40.0
## 10728 CS10728           40.0
## 10850 CS10850           31.5
```

Després de la imputació, tenim 0 NA en la variable *hours_per_week*.

8. Estudi descriptiu

8.1. Funcions de mitjana robustes

Implementeu una funció en R que, donat un vector amb dades numèriques, calculi la mitjana retallada i la mitjana Winsor.

La implementació de la mitjana retallada es pot observar a continuació.

```
mitjana.retallada<-function(x, perc=0.05){
  lowval<-quantile(x,perc)
  highval<-quantile(x,1-perc)
  x<-x[x>lowval & x<highval]
  output<-mean(x)
  output
}
```

Es busquen els valors extrems (*lowval* i *highval*) que deixen el percentatge indicat de la mostra fora amb la funció *quantile()*. Posteriorment es re-assignen els valors que compleixen els dos condicionals units per un operador "&", ser més gran que *lowval* i més petit que *highval*. En el vector resultant es realitza la mitjana aritmètica amb la funció *mean()* i se'n retorna els resultat.

La implementació de la mitjana winsor es pot observar a continuació.

```
mitjana.winsor<-function(x, perc=0.05){
  lowval<-quantile(x,perc)
  highval<-quantile(x,1-perc)
  x<-replace(x,x<lowval,lowval)
  x<-replace(x,x>highval,highval)
  output<-mean(x)
  output
}
```

De manera similar a la mitjana retallada, per la mitjana winsor també es defineixen els valors extrems amb la funció *quantile()*. En aquest cas, però enlloc de re-assignar els valors deixant fora del vector els que no estan en l'interval (*lowval*,*highval*), fet que redueix el nombre d'elements; el que es fa es substituir els valors de fora l'interval, pel valor extrem. Aquesta operació es realitza en dos passos, primer pels valors més petits que *lowval* als que se'ls assigna *lowval* i després pels valors més grans que *highval* als que se'ls assigna *highval*. Aquestes operacions de substitució es realitzen amb la funció *replace()*.Finalment, es calcula la mitja aritmètica i es retorna el resultat.

8.2. Estudi descriptiu de les variables quantitatives

Feu un estudi descriptiu de les variables quantitatives age, education_num, hours_per_week i income. Per a això, prepareu una taula amb diverses mesures de tendència central i dispersió, robustes i no robustes. Feu servir, entre d'altres, les funcions de l'apartat anterior. Presenteu, així mateix gràfics on es visualitzi la distribució dels valors d'aquestes variables quantitatives.

La següent taula mostra les principals mesures de tendència central i dispersió, robustes i no robustes per a les variables *age*, *education_num*, *hours_per_week* i *income*.

```
t_row<-c("age","education_num","hours_per_week","income")
t_col<-c("mean","median","trim_mean","winsor_mean","sd","IQR","mad")
M<-matrix(1,length(t_row),length(t_col))
rownames(M)<-t_row
colnames(M)<-t_col

for (i in t_row){
  M[i,"mean"]<-round(mean(CIds[,i]),2)
```

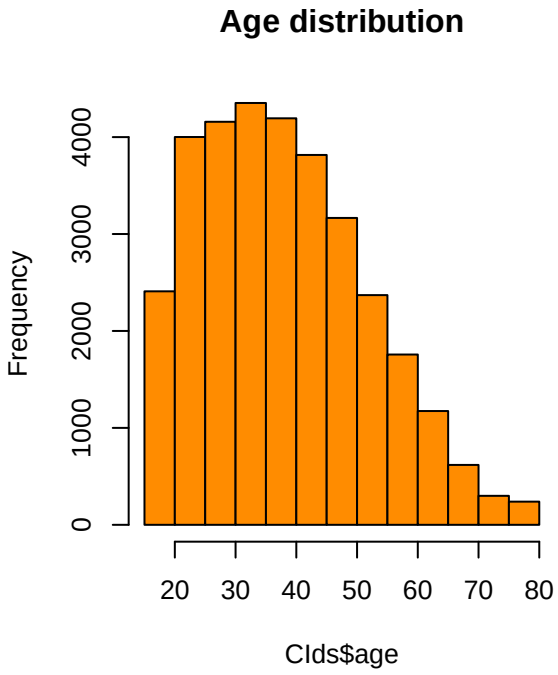
```
  M[i,"median"]<-round(median(CIds[,i]),2)
  M[i,"trim_mean"]<-round(mitjana.retallada(CIds[,i]),2)
  M[i,"winsor_mean"]<-round(mitjana.winsor(CIds[,i]),2)
  M[i,"sd"]<-round(sd(CIds[,i]),2)
  M[i,"IQR"]<-round(IQR(CIds[,i]),2)
  M[i,"mad"]<-round(mad(CIds[,i]),2)
}

kable(M)
```

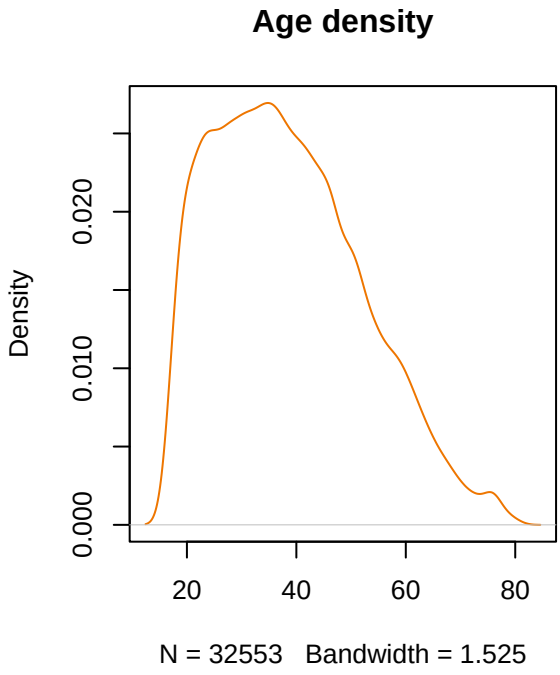
	mean	median	trim_mean	winsor_mean	sd	IQR	mad
age	38.55	37.00	37.89	38.29	13.54	20.0	14.83
education_num	10.08	10.00	10.03	10.10	2.57	3.0	1.48
hours_per_week	40.32	40.50	40.52	40.40	11.48	5.0	4.45
income	48.75	49.71	48.86	48.76	7.10	11.1	7.90

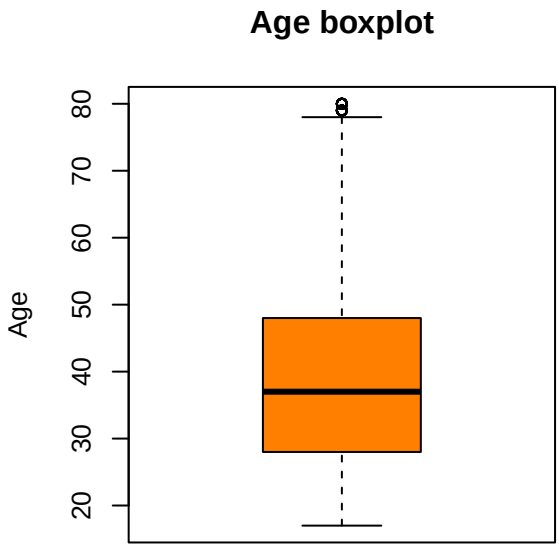
En les properes pàgines es mostren diversos gràfics de la distribució de valors (histograma, diagrama de densitat i diagrama de caixa) per cada una de les variables considerades.

Age

```
par(mfrow=c(2,2))
hist(CIds$age, col = mypalette[1], main = "Age distribution")
plot(density(CIds$age), col = mypalette[3], main = "Age density")
boxplot(CIds$age, col = mypalette[2], ylab="Age", main = "Age boxplot")
```

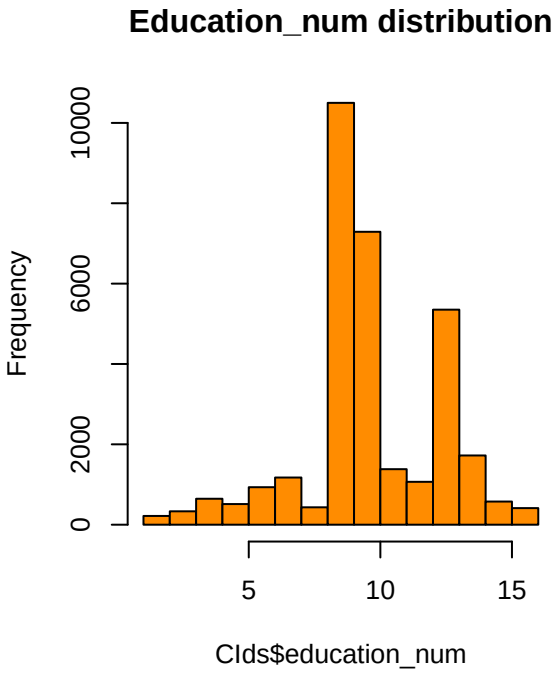


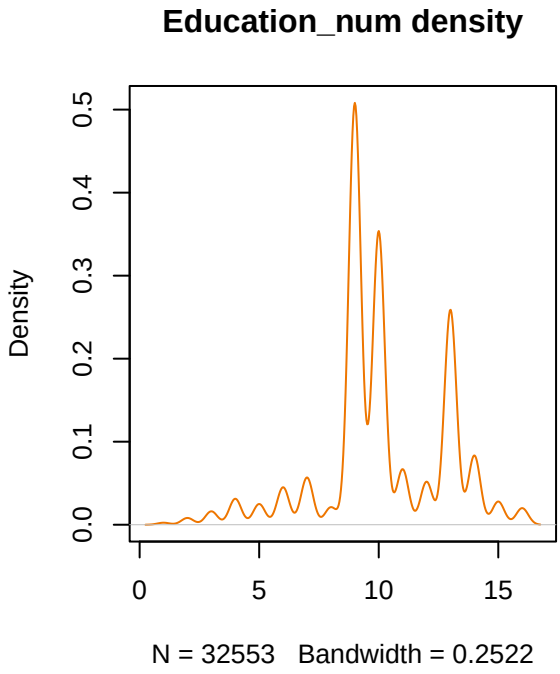


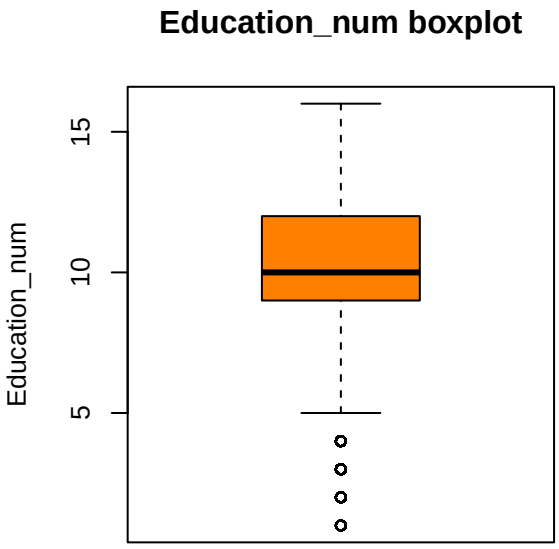


Education_num

```
par(mfrow=c(2,2))
hist(CIds$education_num, col = mypalette[1], main = "Education_num distribution")
plot(density(CIds$education_num), col = mypalette[3], main = "Education_num density")
boxplot(CIds$education_num, col = mypalette[2], ylab="Education_num",
        main = "Education_num boxplot")
```

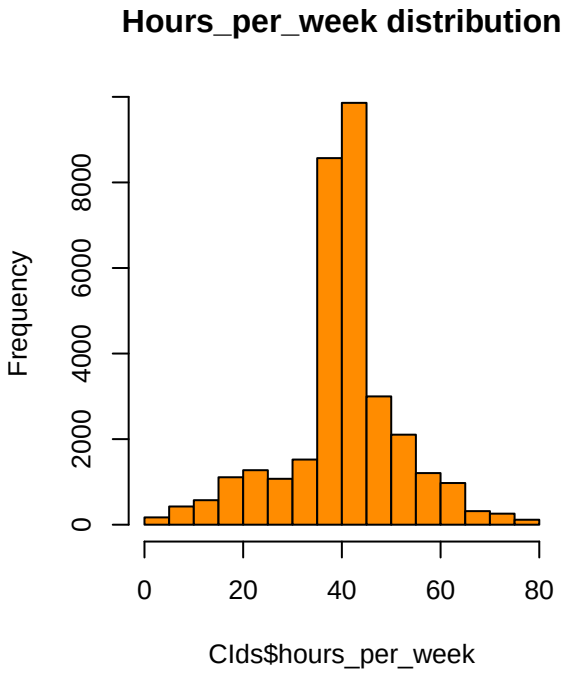


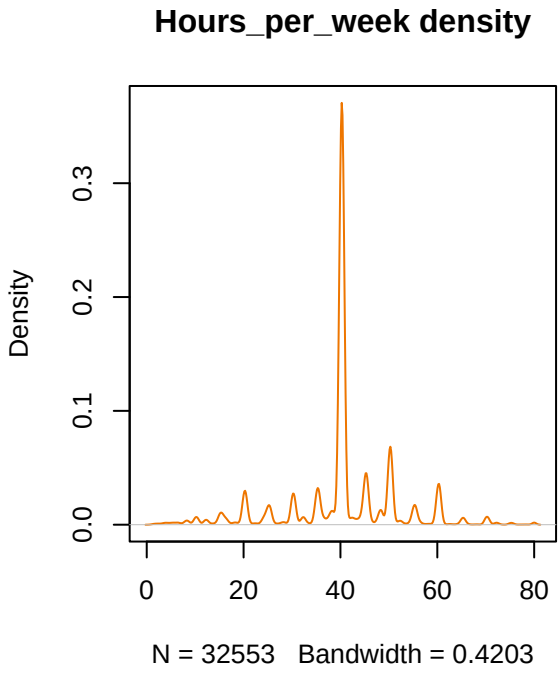


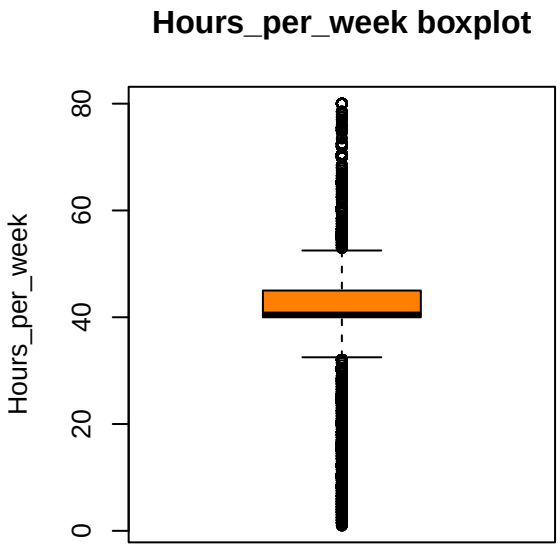


Hours_per_week

```
par(mfrow=c(2,2))
hist(CIds$hours_per_week, col = mypalette[1], main = "Hours_per_week distribution")
plot(density(CIds$hours_per_week), col = mypalette[3], main = "Hours_per_week density")
boxplot(CIds$hours_per_week, col = mypalette[2], ylab="Hours_per_week",
        main = "Hours_per_week boxplot")
```

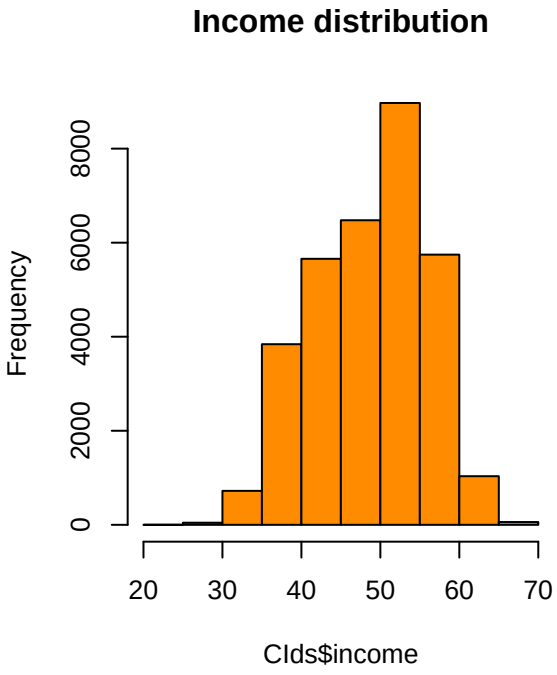


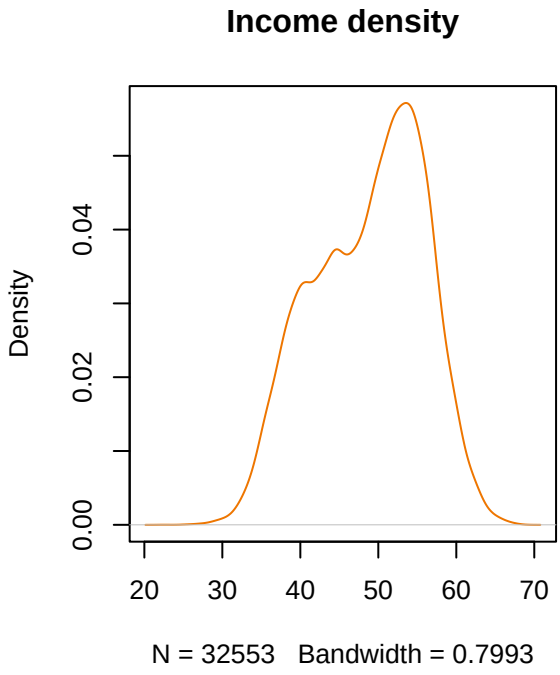


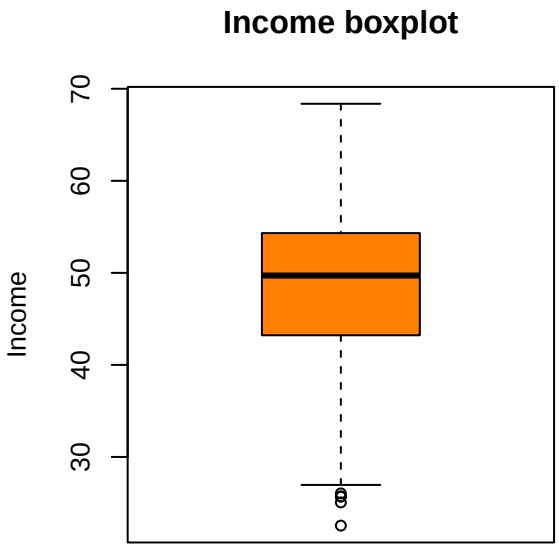


Income

```
par(mfrow=c(2,2))
hist(CIds$income, col = mypalette[1], main = "Income distribution")
plot(density(CIds$income), col = mypalette[3], main = "Income density")
boxplot(CIds$income, col = mypalette[2], ylab="Income", main = "Income boxplot")
```



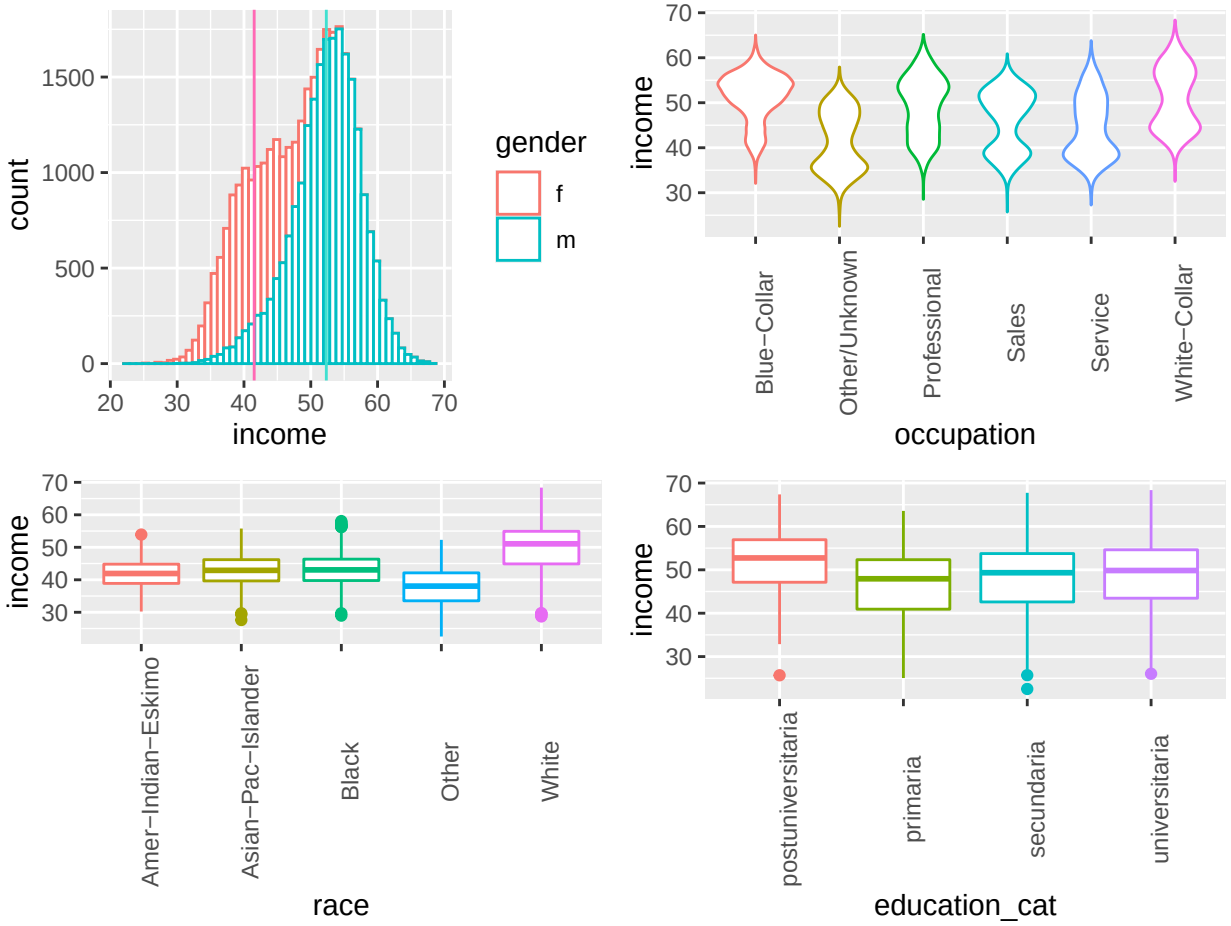




Gràfics addicionals

Finalment es mostren alguns diagrames addicionals generats amb la llibreria *ggplot*, per analitzar a simple vista si existeix biaix de gènere (*gender*) o raça (*race*) en la variable *income*. També es mostra la distribució de la variable *income* per cada nivell d'ocupació (*occupation*) i per cada categoria d'educació (*educacion_cat*).

```
#Creació de gràfics
fmean<-mean(CIds[CIds$gender=="f",]$income)
mmean<-mean(CIds[CIds$gender=="m",]$income)
g1<-ggplot(CIds, aes(x=income, color=gender)) +
  geom_histogram(fill="white", bins = 50) +
  geom_vline(aes(xintercept=fmean), color="hotpink") +
  geom_vline(aes(xintercept=mmean), color="turquoise")
g2<-ggplot(CIds, aes(x=race, y=income, color=race, order)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 90), legend.position = 'none')
g3<-ggplot(CIds, aes(x=education_cat, y=income, color=education_cat, order)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 90), legend.position = 'none')
g4<-ggplot(CIds, aes(x=occupation, y=income, color=occupation)) +
  geom_violin()+
  theme(axis.text.x = element_text(angle = 90), legend.position = 'none')

#Alineament de gràfics
grid.arrange(g1,g4,g2,g3,ncol=2)
```



9. Arxiu final

Un cop realitzat el preprocessament sobre l'arxiu, copieu el resultat de les dades a un fitxer anomenat CensusIncome_clean.csv.

Per a la creació del l'arxiu final es proposa el codi que es mostra a continuació, considerant el format espanyol i per tant utilitzant la funció *write.csv2*:

```
write.csv2(CIds, file="CensusIncome_clean.csv", row.names = FALSE)
```