A1 - Preproceso de datos

Solución

Semestre 2021.2

Índice

Introducción

En esta actividad realizaremos el preprocesado de un fichero de datos que contiene información de una muestra extraída a partir de un censo, en el que para cada persona, se registran los salarios aparte de información personal adicional. El conjunto de datos contiene 32,560 registros y 14 variables. Los datos se han extraído y modificado parcialmente (por motivos académicos) de la base de datos disponible en la web Machine Learning Repository: https://archive.ics.uci.edu/ml/datasets/Adult.

Las variables son:

- *CS_ID:* Identificador del individuo.
- *age:* Edad del individuo.
- *workclass*: Categorización del individuo en base al perfil laboral.
- *fnlwgt*: Numero de unidades de la población objetivo que representa la unidad respondiente.
- *education_num*: Numero de años de formación educativa del individuo.
- *marital_status*: Estado civil del individuo.
- *occupation*: Categorización del individuo en base a la tipologia del trabajo.
- *relationship*: Parentesco de la unidad que responde de la familia.
- *race*: Grupo racial al que pertenece el individuo.
- *sex*: Genero del individuo.
- *capital_gain*: Dineros ganados por inversión del individuo.
- *capital_loss*: Dineros perdidos por inversión del individuo.
- *hours_per_week*: Horas por semana trabajadas por el individuo.
- *income*: Salario (anual) del individuo.

El objetivo de esta actividad es preparar el fichero para su posterior análisis. Para ello, se examinará el fichero para detectar y corregir posibles errores, inconsistencias y valores perdidos. Además, se presentará una breve estadística descriptiva.

Criterios de verificación y de normalización de las variables:

A continuación se muestran los criterios con los que deben limpiarse los datos del conjunto:

1. Selección del dataset de estudio a partir del dataset original.
   a) Eliminar las variables ‘fnlwgt‘, ‘capital_gain‘ y ‘capital_loss‘ y los registros con más de 5 valores NAs.
   b) Crear la variable ‘education_cat‘ como categorización de la variable numérica ‘education_num‘ (más detalles en la sección correspondiente)
   c) Cambiar el nombre de la variable ‘sex‘ por ‘gender‘.
2. Verificar si hay registros con el valor de ‘CS_ID‘ duplicado. En caso de duplicación, se han de asignar nuevos códigos no repetidos.
3. Eliminar espacios en blanco en el inicio de los valores en las variables cualitativas.
4. Los valores posibles en la variable ‘marital_status‘ son: M (Married), S (Single), X (Separated), D (Divorced), W (Widowed).
5. Los valores posibles en la variable ‘gender‘ son: f (femeníno) y m (masculíno).

6. En los datos numéricos, el símbolo de separador decimal es el punto y no la coma. Además, si se presenta la unidad de la variable, por ejemplo horas en el caso de hours_per_week, se debe eliminar para convertir la variable a tipo numérico.
7. En la variable 'hours_per_week' se consideran valores atípicos aquellos valores superiores a 80 horas/semana.
8. La variable 'income' se ha de expresar en miles de euros (k€)
9. Las variables 'age' y 'education_num' han de ser de tipos entero, sin decimales.

Nota importante a tener en cuenta para entregar la actividad:

- Es necesario entregar el archivo Rmd y el archivo de salida (PDF o html). El archivo de salida debe incluir: el código y el resultado de la ejecución del código (paso a paso).
- Se respetará la misma numeración de los apartados que el enunciado.
- No se pueden realizar listas completas del conjunto de datos en la solución. Esto generaría un documento con cientos de páginas y dificulta la revisión del texto. Para comprobar las funcionalidades del código sobre los datos, se pueden utilizar las funciones **head** y **tail** que sólo muestran unas líneas del archivo de datos.
- Se valora la precisión de los términos utilizados (es necesario utilizar de forma precisa la terminología de la estadística).
- Se valora también la concisión en la respuesta. No se trata de realizar explicaciones muy largas o documentos muy extensos. Hay que explicar el resultado y argumentar la respuesta a partir de los resultados obtenidos de forma clara y concisa.

Para realizar el preproceso del fichero, seguir los pasos que se indican a continuación.

---

1. Carga del archivo

Cargar el archivo de datos y examinar el tipo de datos con los que R ha interpretado cada variable. Examinar también los valores resumen de cada tipo de variable.

```
ds0<-read.csv2("CensusIncomedataset.csv",stringsAsFactors=TRUE)

# Obtenim les dimensions del conjunt de dades, l'estructura i contingut.
dim(ds0)
```

```
## [1] 32560    14
```

```
str(ds0)
```

```
## 'data.frame':    32560 obs. of  14 variables:
##  $ CS_ID         : Factor w/ 32553 levels "CS1","CS10","CS100",..: 1 11112 22222 25894 27005 28116 29
##  $ age           : int  50 38 53 28 37 49 52 31 42 37 ...
##  $ workclass     : Factor w/ 4 levels " Government",..: 4 3 3 3 3 3 4 3 3 3 ...
##  $ fnlwgt        : int  77516 83311 215646 234721 338409 284582 160187 209642 45781 159449 ...
##  $ education_num : int  13 9 7 13 14 5 9 14 13 10 ...
##  $ marital_status: Factor w/ 5 levels " Divorced"," Married",..: 2 1 2 2 2 2 2 4 2 2 ...
##  $ relationship  : Factor w/ 6 levels "Husband","Not-in-family",..: 2 1 2 1 6 6 2 1 2 1 ...
##  $ occupation    : Factor w/ 6 levels " Blue-Collar",..: 6 1 1 3 6 5 6 3 6 6 ...
##  $ race          : Factor w/ 5 levels " Amer-Indian-Eskimo",..: 5 5 3 3 5 3 5 5 5 3 ...
##  $ sex           : Factor w/ 8 levels " F"," Fem"," female",..: 6 6 6 2 2 2 6 2 6 6 ...
##  $ capital_gain  : int  2174 0 0 0 0 0 0 0 14084 5178 ...
##  $ capital_loss  : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ hours_per_week: Factor w/ 181 levels "1 h","1,5 h",..: 10 69 70 69 70 15 79 92 69 149 ...
##  $ income        : Factor w/ 4103 levels "0,1 Milers d'euros",..: 2 2509 2308 1517 1 1463 3 749 3291
```

```
head(ds0)
```

```
##   CS_ID age      workclass fnlwgt education_num marital_status  relationship
## 1   CS1  50  Self-Employed  77516            13        Married Not-in-family
## 2   CS2  38        Private  83311             9       Divorced       Husband
## 3   CS3  53        Private 215646             7        Married Not-in-family
## 4   CS4  28        Private 234721            13        Married       Husband
## 5   CS5  37        Private 338409            14        Married          Wife
## 6   CS6  49        Private 284582             5        Married          Wife
##      occupation   race  sex capital_gain capital_loss hours_per_week
## 1  White-Collar  White    M         2174            0         13,5 h
## 2   Blue-Collar  White    M            0            0           40 h
## 3   Blue-Collar  Black    M            0            0         40,5 h
## 4  Professional  Black  Fem            0            0           40 h
## 5  White-Collar  White  Fem            0            0         40,5 h
## 6       Service  Black  Fem            0            0           16 h
##                  income
## 1 2000,3 Milers d'euros
## 2  51,67 Milers d'euros
## 3  50,08 Milers d'euros
## 4  44,21 Milers d'euros
## 5    0,1 Milers d'euros
## 6  43,93 Milers d'euros
```

```
summary(ds0)
```

```
##      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
```

2. Obtención del dataset para realizar el estudio

En esta fase se seleccionan los registros y/o variables que se utilizarán para realizar el estudio. En muchas ocasiones, no se trabaja con todo el dataset disponible ya que existen variables y/o registros de control o que no es necesario usar para el estudio concreto que se quiere realizar.

En este caso, se quieren eliminar las variables `fnlwgt`, `capital_gain` y `capital_loss` además de los registros con más de 5 valores NAs.

Por otra parte, se pueden crear nuevas variables en función de las disponibles. En este caso, se creará la variable `education_cat` que categoriza la formación académica en formación `primaria` si `education_num` es menor de 7 años, `secundaria` si `education_num` está entre 7 y 9 años, `universitaria` si `education_num` esta entre 10 y 13 años y `postuniversitaria` si `education_num` es mayor de 13 años.

Por último, se quiere cambiar el nombre de la variable `sex` por `gender`.

```
# eliminar atributos
elim.names<- c( "fnlwgt", "capital_gain", "capital_loss")
elim.pos <- which(names(ds0) %in% elim.names)
ds <- ds0[,-elim.pos]
dim(ds)
```

```
## [1] 32560    11
```

```
# eliminar registros
NA.count <- apply(is.na(ds),1,sum)
NA.reg <- which(NA.count>5)
ds <- ds[-NA.reg,]
dim(ds)
```

```
## [1] 32553    11
```

```
# Creación nueva variable
ds$education_cat <- NULL
ds$education_cat[ ds$education_num < 7 ] <- "Primaria"
ds$education_cat[ ds$education_num >= 7  & ds$education_num < 10] <- "Secundaria"
ds$education_cat[ ds$education_num >= 10  & ds$education_num < 14] <- "Universitaria"
ds$education_cat[ ds$education_num >= 14] <- "Postuniversitaria"

table(ds$education_num,ds$education_cat)
```

```
##
##      Postuniversitaria Primaria Secundaria Universitaria
##   1                  0       50          0             0
##   2                  0      168          0             0
##   3                  0      333          0             0
##   4                  0      646          0             0
##   5                  0      514          0             0
##   6                  0      933          0             0
##   7                  0        0       1174             0
##   8                  0        0        433             0
```

```
##   9                0        0      10499            0
##   10               0        0          0         7290
##   11               0        0          0         1382
##   12               0        0          0         1067
##   13               0        0          0         5352
##   14            1723        0          0            0
##   15             576        0          0            0
##   16             413        0          0            0
```

```
# Cambiar el nombre de una variable

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

3. Duplicación de códigos

Verificad la consistencia en la variable `CS_ID`. Si existen registros duplicados, asignad un nuevo código para evitar códigos duplicados. El nuevo código debe ser un valor no usado (valores superiores al máximo valor numérico contenido en `CS_ID`). Conservad el mismo formato que el resto de códigos, con "CS" delante de la secuencia numérica. Podéis usar la función **duplicated** de R para detectar los duplicados.

```
#detectar si hay codigos duplicados
sum( table(ds$CS_ID)>1 )
```

```
## [1] 7
```

```
#Códigos
codes <- as.character( ds$CS_ID )
#posiciones que contienen codigos duplicados
idx <- which( duplicated(codes) == TRUE ); idx
```

```
## [1]   624  7162  7451  8014  8083  9192 23648
```

```
#N es el número de registros duplicados
N <- length(idx); N
```

```
## [1] 7
```

```
#Generemos una secuencia correlativa desde max+1
max.code <- max( as.numeric( str_sub(codes, 3) )) + 1
new.codes <- seq( max.code, max.code + N-1 )
#Asignamos los nuevos codigos, siguiendo el formato del codigo con "CS"
codes[idx] <- paste("CS", new.codes, sep="")
#check
codes[idx]
```

```
## [1] "CS32561" "CS32562" "CS32563" "CS32564" "CS32565" "CS32566" "CS32567"
```

```
sum( table(codes)>1 )
```

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

```
ds$CS_ID <- codes
```

4. Normalización de los datos cualitativos

4.1. Eliminación de espacios en blanco

Se ha observado que existen espacios en blanco al inicio de los valores en las variables cualitativas. Por tanto, es necesario eliminar estos espacios en blancos.

```
# Espacios en blanco
ds$workclass <- gsub(" ", "", ds$workclass)
ds$marital_status <- gsub(" ", "", ds$marital_status)
ds$occupation <- gsub(" ", "", ds$occupation)
ds$race <- gsub(" ", "", ds$race)
ds$gender <- gsub(" ", "", ds$gender)

# Convertir a factor

ds$workclass <- as.factor(ds$workclass)
ds$marital_status <- as.factor(ds$marital_status)
ds$occupation <- as.factor(ds$occupation)
ds$race <- as.factor(ds$race)
ds$gender <- as.factor(ds$gender)
```

4.2. Marital-Status

Cambiar las categorías de la variable marital status actuales por otras que ocupen un carácter. Los valores que se asignaron a la variable `marital_status` son: M por Married, S por Single, X por Separated, D por Divorced, W por Widowed. Representad gráficamente la distribución de los valores de la variable.

```
# Valores posibles
unique(ds$marital_status)
```

```
## [1] Married   Divorced  Single    Separated Widowed
## Levels: Divorced Married Separated Single Widowed
```

```
# convertir a character para poder hacer el cambio de niveles
  ds$marital_status <- as.character(ds$marital_status)
  unique(ds$marital_status)
```

```
## [1] "Married"   "Divorced"  "Single"    "Separated" "Widowed"
```

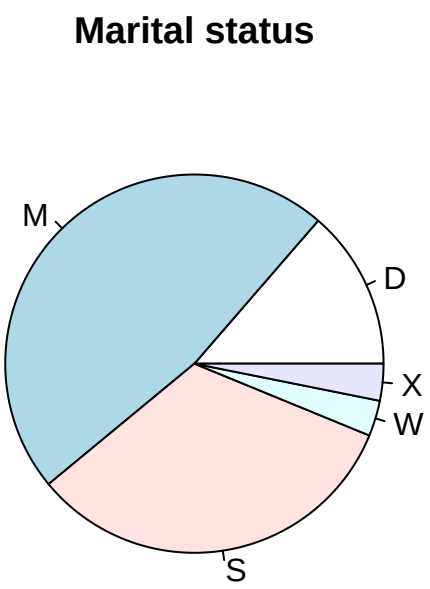
```
# cambios
  ds$marital_status[ grep( "Married", ds$marital_status )] <- "M"
  ds$marital_status[ grep( "Divorced", ds$marital_status )] <- "D"
  ds$marital_status[ grep( "Separated", ds$marital_status )] <- "X"
  ds$marital_status[ grep( "Single", ds$marital_status )] <- "S"
  ds$marital_status[ grep( "Widowed", ds$marital_status )] <- "W"

# Volver a factor
  ds$marital_status <- as.factor(ds$marital_status)
  levels(ds$marital_status)
```

```
## [1] "D" "M" "S" "W" "X"
```

```
    #Verificación si existen valores NA en MaritalStatus
  sum( complete.cases(ds$marital_status) )
```

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

```
# Representación gráfica
  pie(table(ds$marital_status), main="Marital status")
```



4.3. Género

Revisad la consistencia de los valores de la variable `gender` y realice las modificaciones oportunas para indicar las categorías finales como `f` y `m` que corresponde a `femeníno` y `masculíno`, respectivamente. Representad gráficamente la distribución de los valores de la variable.

```
  table( ds$gender )
```

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

```
  str( ds$gender )
```

```
##  Factor w/ 8 levels "F","Fem","female",..: 6 6 6 2 2 2 6 2 6 6 ...
```

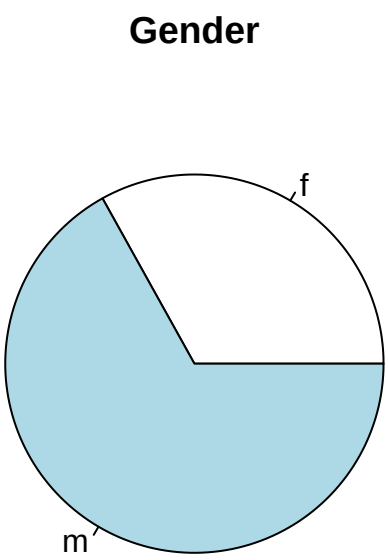
```
  levels( ds$gender )
```

```
## [1] "F"      "Fem"    "female" "Female" "m"      "M"      "male"   "Male"
```

```
  levels( ds$gender )<-c( "f", "f", "f", "f", "m","m","m","m")
  table( ds$gender )
```

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

```
# Representación gráfica
pie(table(ds$gender), main="Gender")
```



5. Normalización de los datos cuantitativos

Inspeccionar los valores de los datos cuantitativos y realizar las normalizaciones oportunas siguiendo los criterios especificados anteriormente. Estas normalizaciones tienen como objetivo uniformizar los formatos. Si hay valores perdidos o valores extremos, se tratarán más adelante.

Al realizar estas normalizaciones, se debe demostrar que la normalización sobre cada variable ha dado el resultado esperado. Por lo tanto, se recomienda mostrar un fragmento del archivo de datos resultante. Para evitar mostrar todo el conjunto de datos, se puede mostrar una parte del mismo, con las funciones **head** y/o **tail**.

Seguid el orden de los apartados.

5.1. Edad

Revisad el formato de la variable `age` y realizad las transformaciones oportunas según los criterios especificados anteriormente. Si existen valores atípicos, se tratarán más adelante.

```
class(ds$age)
```

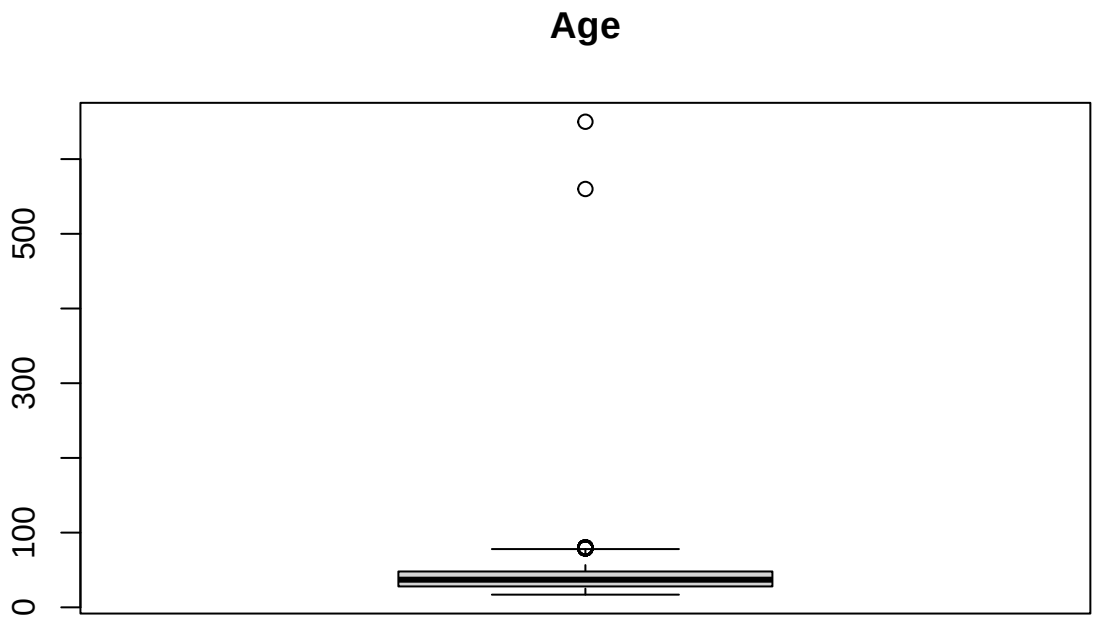
```
## [1] "integer"
```

```
head(ds$age, 30)
```

```
##  [1] 50 38 53 28 37 49 52 31 42 37 30 23 32 40 34 25 32 38 43 40 54 35 43 59 56
```

```
## [26] 19 54 39 49 23
```

```
   boxplot(ds$age, main="Age")
```




```
   #No se hacen transformaciones. Age es un número entero y lor valores atípicos se revisan más adelant
```

5.2. Educación

Revisad el formato de la variable `education_num` y realizad las transformaciones oportunas según los criterios especificados anteriormente. Si existen valores atípicos, se tratarán más adelante.

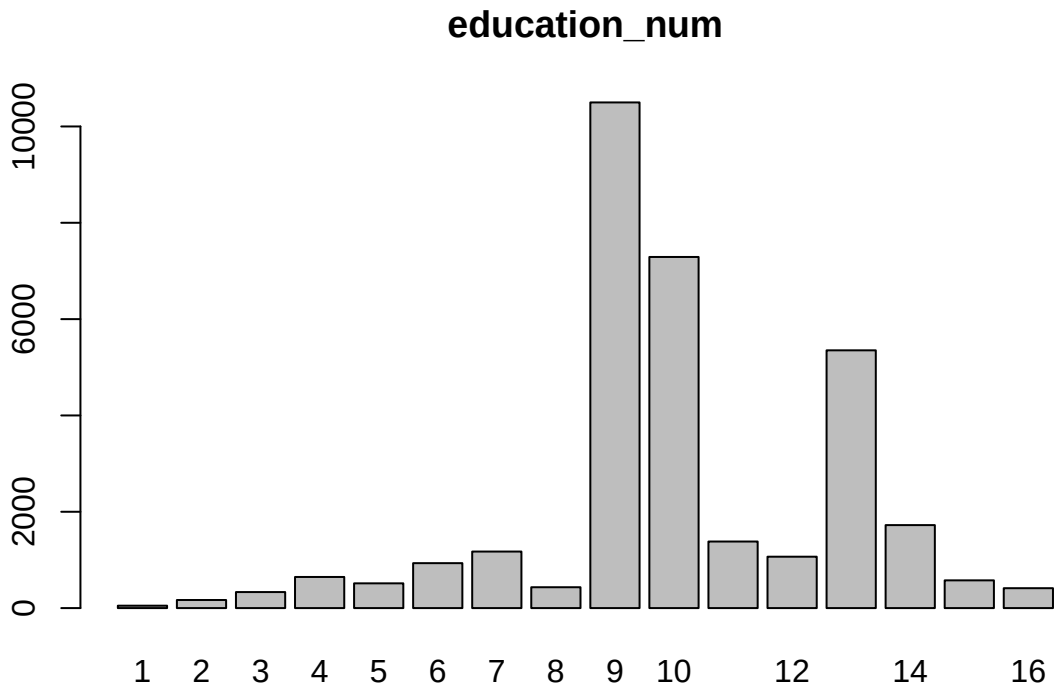
```
   class(ds$education_num)
```

```
## [1] "integer"
```

```
   head(ds$education_num)
```

```
## [1] 13  9  7 13 14  5
```

```
   barplot( table(ds$education_num), main="education_num")
```




```
   #No se hacen  transformaciones.
```

5.3. Horas por semana

Revisad el formato de la variable `hours_per_week` y realizad las transformaciones oportunas según los criterios especificados anteriormente. Si existen valores atípicos, se tratarán más adelante.

```
   class(ds$hours_per_week)
```

```
## [1] "factor"
```

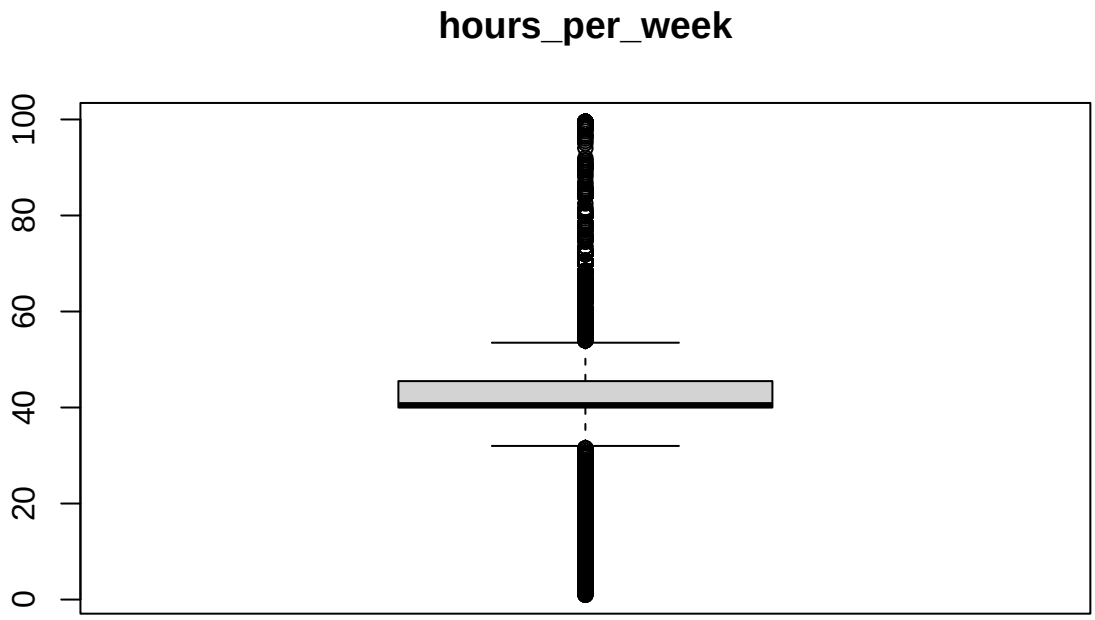
```
   head( ds$hours_per_week, 30)
```

```
##  [1] 13,5 h 40 h   40,5 h 40 h   40,5 h 16 h   45 h   50,5 h 40 h   80,5 h
## [11] 40 h   30,5 h 50 h   40,5 h 45 h   35,5 h 40,5 h 50,5 h 45,5 h 60,5 h
## [21] 20 h   40,5 h 40,5 h 40,5 h 40,5 h 40,5 h 60 h   80 h   40 h   52,5 h
## 181 Levels: 1 h 1,5 h 10 h 10,5 h 11 h 11,5 h 12 h 12,5 h 13 h 13,5 h ... 99,5 h
```

```
   ds$hours_per_week <- as.character( ds$hours_per_week )
   # seleccionar los valores numéricos
   ds$hours_per_week <- word(ds$hours_per_week)
   #corregir la coma por el punto decimal
   ds$hours_per_week <- gsub("\\,","\\.", ds$hours_per_week)
   ds$hours_per_week <- as.numeric( ds$hours_per_week )
   head( ds$hours_per_week, 30)
```

```
##  [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
```

```
boxplot( ds$hours_per_week, main="hours_per_week" )
```




```
      #No se hacen transformaciones.
```

5.4. Income

Revisad el formato de la variable `income` y realizad las transformaciones oportunas según los criterios especificados anteriormente. Si existen valores atípicos, se tratarán más adelante.

```
class(ds$income)
```

```
## [1] "factor"
```

```
head(ds$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
## 4103 Levels: 0,1 Milers d'euros 2000,3 Milers d'euros ... NA Milers d'euros
```

```r
#Cambiar la coma decimal por el punto; pasar a unidades
ds$income<-gsub("\\,", "\\.", as.character(ds$income) )

# Verificación
idx.coma<-grep("\\,", ds$income)
idx.coma
```

```
## integer(0)
```

```r
# función para convertir euros a miles de euros
euros_to_keuros <-function(income){

  income <- as.character(income)
  # indice para los income en euros
  idxm <- grep( "\\ euros", income )
  # seleccionar los valores numéricos
  num_val <- word(income)
  # convertir a numérico
  num_val <- as.numeric(num_val)
  # pasar los valores en euros a miles de euros (kiloeuros)
  num_val[idxm] <- num_val[idxm]/1000

  return (num_val)
}

income <- euros_to_keuros( ds$income )
head( paste( income, ds$income), 30)
```

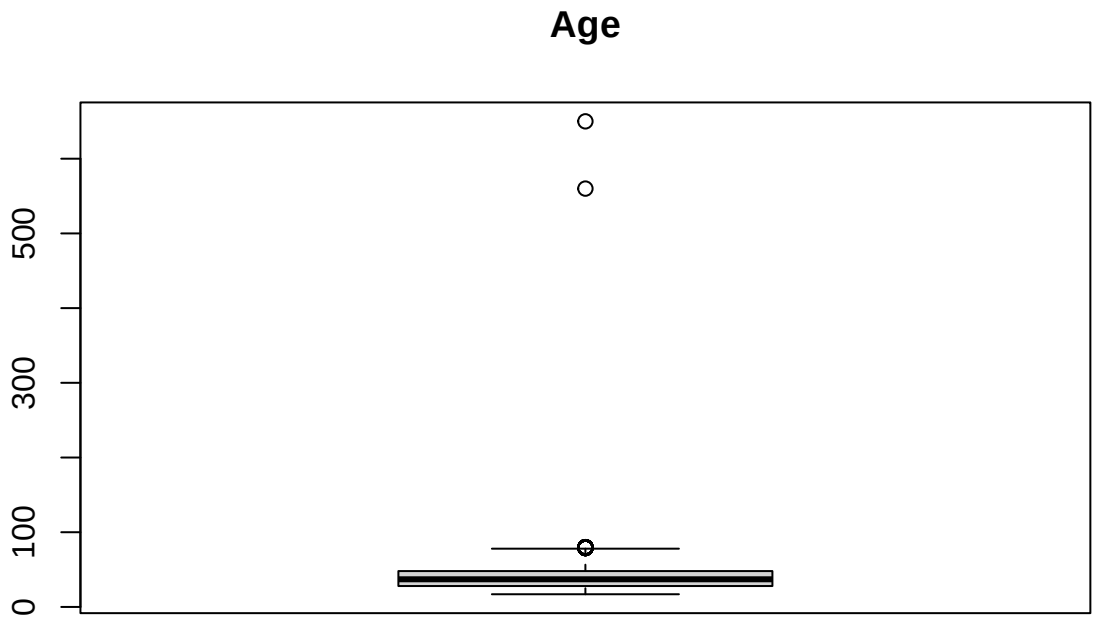
```
##  [1] "2000.3 2000.3 Milers d'euros" "51.67 51.67 Milers d'euros"
##  [3] "50.08 50.08 Milers d'euros"   "44.21 44.21 Milers d'euros"
##  [5] "0.1 0.1 Milers d'euros"       "43.93 43.93 Milers d'euros"
##  [7] "2134.6 2134.6 Milers d'euros" "38.24 38.24 Milers d'euros"
##  [9] "57.12 57.12 Milers d'euros"   "49.72 49.72 Milers d'euros"
## [11] "51.08 51.08 Milers d'euros"   "44.34 44.34 Milers d'euros"
## [13] "41.98 41.98 Milers d'euros"   "46.53 46.53 Milers d'euros"
## [15] "42.5 42.5 Milers d'euros"     "51.16 51.16 Milers d'euros"
## [17] "51.95 51.95 Milers d'euros"   "47.88 47.88 Milers d'euros"
## [19] "44.38 44.38 Milers d'euros"   "55.01 55.01 Milers d'euros"
## [21] "35.87 35.87 Milers d'euros"   "49.36 49.36 Milers d'euros"
## [23] "52.61 52610 euros"            "38.3 38.3 Milers d'euros"
## [25] "56.66 56.66 Milers d'euros"   "47.19 47.19 Milers d'euros"
## [27] "42.7 42.7 Milers d'euros"     "55.67 55.67 Milers d'euros"
## [29] "53.18 53.18 Milers d'euros"   "55.48 55.48 Milers d'euros"
```

```r
ds$income <- income
```

6. Valores atípicos

Revisad si hay valores atípicos en las variables `age`, `education_num`, `hours_per_week` y `income`. Si se trata de un valor anómalo, es decir anormalmente alto o bajo, substituir su valor por NA y posteriormente, se imputará.

```r
  #Age
  boxplot(ds$age, main="Age")
```



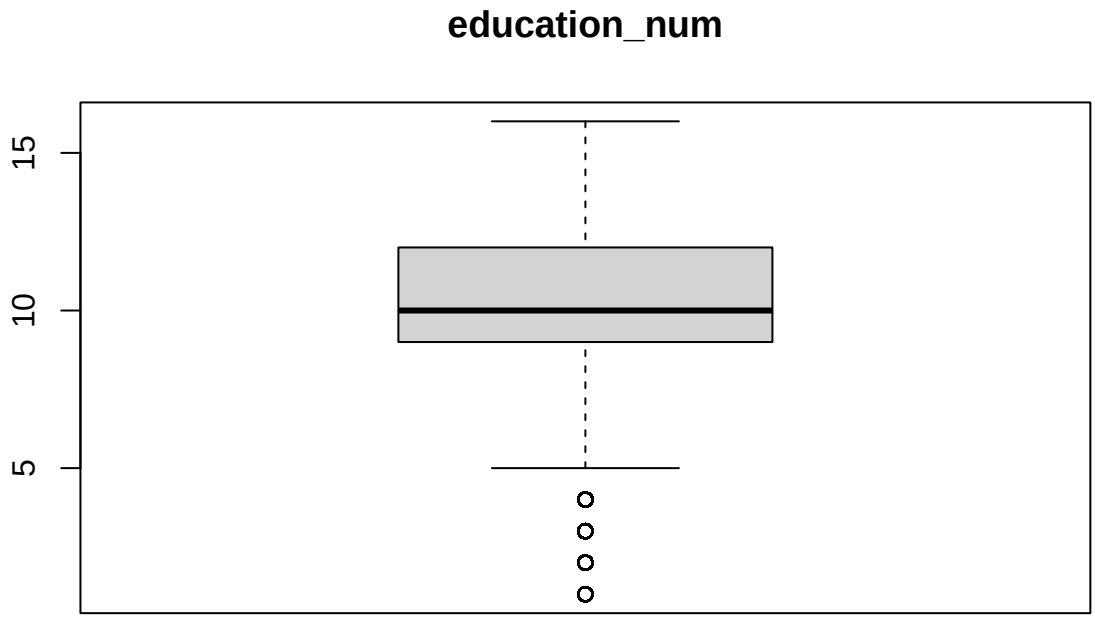

```
  x<-boxplot.stats(ds$age)$out
  idx <- which( ds$age %in% x)
  sort(ds$age[idx])
```

```
##  [1]  79  79  79  79  79  79  79  79  79  79  79  79  79  79  79  79  79  79
## [20]  79  79  79  80  80  80  80  80  80  80  80  80  80  80  80  80  80  80
## [39]  80  80  80  80 560 650
```

```
  #se asigna a NA els valors >120. El resto se deja igual.
  ds$age[ ds$age > 120 ] <- NA
     #Check
  sum(is.na(ds$age))
```

```
## [1] 2
```

```
  #education_num
  boxplot(ds$education_num, main="education_num")
```



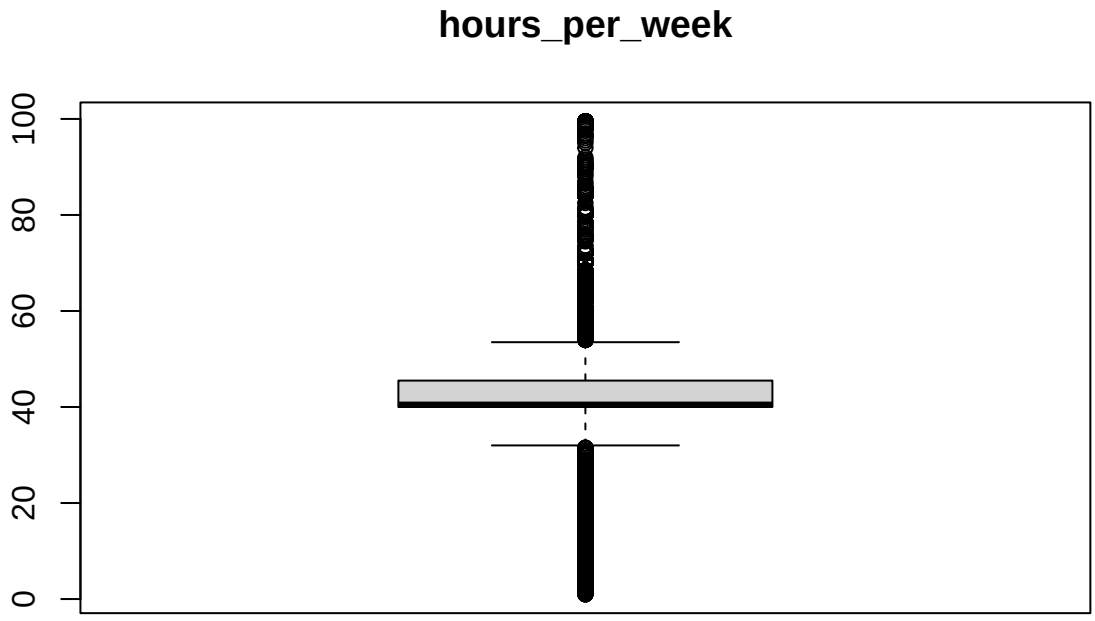

```
  x<-boxplot.stats(ds$education_num)$out
  length(x)
```

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

```
  sort(x)[1:100]
```

```
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [75] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

```
  #No se cambian los valores porqué aunque halla muchos valores
  #atípics, no se consideran valores anómalos.


  #hours_per_week
  boxplot(ds$hours_per_week,main="hours_per_week")
```




```
  x<-boxplot.stats(ds$hours_per_week)$out
  idx <- which( ds$hours_per_week %in% x)
  head( sort(ds$hours_per_week[idx],decreasing=TRUE) , 30)
```

```
##  [1] 99.5 99.5 99.5 99.5 99.5 99.5 99.5 99.5 99.5 99.5 99.5 99.5 99.5 99.5 99.5
## [16] 99.5 99.5 99.5 99.5 99.5 99.5 99.5 99.5 99.5 99.5 99.5 99.5 99.5 99.5 99.5
```

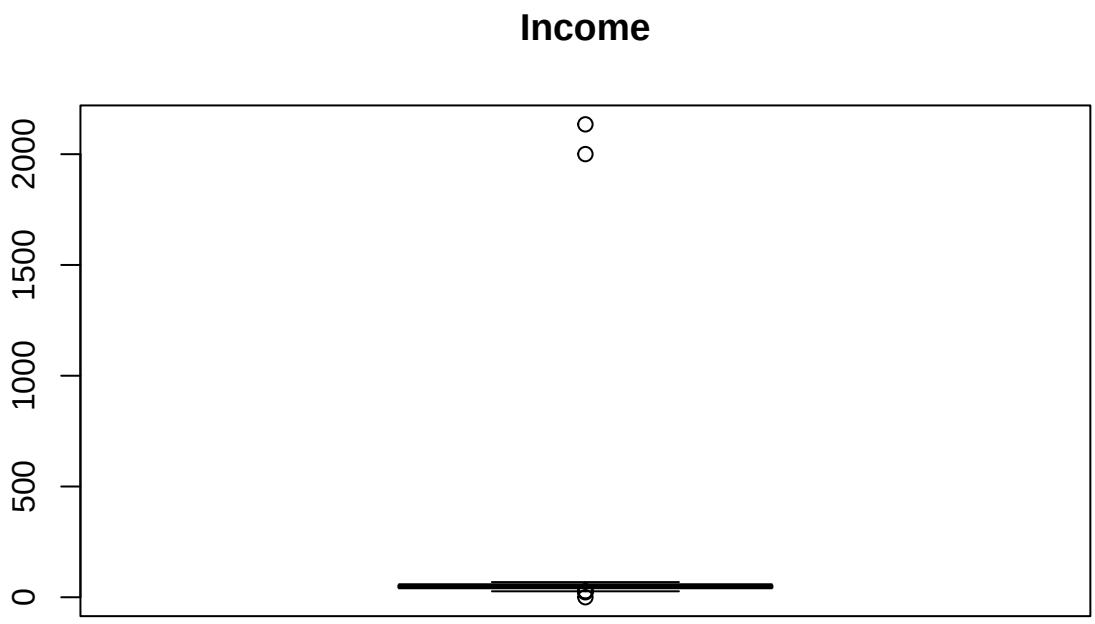
```
  sum(ds$hours_per_week>80)
```

```
## [1] 277
```

```
  #se asigna NA a los valores mayores de 80. Si se usa un criterio menos restrictivo, también  és corre
  ds$hours_per_week[ ds$hours_per_week > 80 ] <- NA
    #Check
  sum(is.na(ds$hours_per_week))
```

```
## [1] 277
```

```
  #income
  boxplot(ds$income,main="Income")
```




```
  x<-boxplot.stats(ds$income)$out
  idx <- which( ds$income %in% x)
  sort(ds$income[idx],decreasing=TRUE)
```

```
## [1] 2134.60 2000.30   26.04   25.71   25.69   25.06   22.54    0.10
```

```
  #se asigna a NA los valores >100. El resto se deja igual.
  ds$income[ ds$income > 100 ] <- NA
   #Check
  sum(is.na(ds$income))
```

```
## [1] 2
```

```
  #Los valores son correctos. No se modifican
```

7. Imputación de valores

Buscad si existen valores perdidos en las variables cuantitativas `age`, `education_num`, `hours_per_week` y `income`.

En caso de valores perdidos, aplicad el proceso siguiente:

- Para 'age', aplicad imputación por la media aritmética.
- Para 'income', aplicar imputación por la media aritmética de los registros del mismo género, es decir, separado por género.
- En el resto de variables, aplicad imputación por vecinos más cercanos, usando la distancia de Gower, considerando en el cómputo de los vecinos más cercanos el resto de variables cuantitativas mencionadas

en este apartado. Además, considerad que la imputación debe hacerse con registros del mismo género. Por ejemplo, si un registro a imputar es de género "M", se debe realizar la imputación usando las variables cuantitativas de los registros de género "M". Para realizar esta imputación, podéis usar la función "kNN" de la librería VIM con un número de vecinos igual a 11.

Mostrad que la imputación se ha realizado correctamente, mostrando el resultado de los datos afectados por la imputación.

```
# total registres
nrow(ds)
```

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

```
# Total sense valors NAs
sum(complete.cases(ds$age))
```

```
## [1] 32551
```

```
sum(complete.cases(ds$education_num))
```

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

```
sum(complete.cases(ds$hours_per_week))
```

```
## [1] 32276
```

```
sum(complete.cases(ds$income))
```

```
## [1] 32551
```

```
#Hay valores perdidos en age, hours_per_week y income
#Age
idx <- which(is.na(ds$age))
ds$age[idx]
```

```
## [1] NA NA
```

```
ds$age[idx] <- round( mean( ds$age, na.rm=TRUE ))
ds$age[idx]
```

```
## [1] 39 39
```

```
#Income
idx <- which(is.na(ds$income))
ds$income[idx]
```

```
## [1] NA NA
```

```
mean.m <- round( mean( ds$income[ds$gender=="m"], na.rm=TRUE ))
mean.f <- round( mean( ds$income[ds$gender=="f"], na.rm=TRUE ))
ds$income[idx] <- ifelse(ds$gender[idx]=="m",mean.m,mean.f)
ds$income[idx]
```

```
## [1] 52 52
```

```
#Valores perdidos en hours_per_week
variables.num <- which( colnames(ds) %in% c("age", "education_num", "hours_per_week", "income"))
idx <- which( is.na(ds$hours_per_week))

#Identificar por separado los NAs de género femení y los de género masculíno
fem.idx <- which( is.na(ds$hours_per_week) & (ds$gender=="f") ); fem.idx
```

```
##  [1]  1271  2014  4293  4347  5431  5488  8068  8775 10722 10844 10948 12782
## [13] 14362 14587 14761 15010 15528 16030 17434 17605 18597 19135 19725 19991
## [25] 20030 20872 22184 22231 23220 23742 23823 23918 25667 26138 28105 28618
## [37] 29180 30747 31590 31747 32462 32524
```

```
mas.idx <- which( is.na(ds$hours_per_week) & ds$gender=="m"); mas.idx
```

```
##   [1]    10   272   934  1065  1171  1199  1416  1729  1823  1886  2332  2428
##  [13]  2760  2919  2958  2962  3091  3224  3342  3577  3748  3908  4085  4090
##  [25]  4307  4311  4336  4440  4735  4777  4861  5084  5241  5375  5467  5506
##  [37]  5681  5876  6069  6379  6389  6474  6523  6617  6692  6742  6897  7099
##  [49]  7141  7675  7806  7860  8034  8041  8065  8144  8155  8219  8388  8623
##  [61]  8649  8664  8791  8818  9119  9712  9805  9825  9878 10054 10120 10137
##  [73] 10254 10260 10371 10462 10908 10980 11151 11497 11827 11829 12203 12324
##  [85] 12514 12611 12619 12918 12967 12999 13330 13552 13574 13611 13701 13852
##  [97] 13926 14503 14505 14636 14980 15002 15174 15350 15454 15502 15703 15789
## [109] 15848 15893 16288 16312 16598 16862 16925 16954 16986 17247 17280 17327
## [121] 17607 17668 17809 18390 18651 18699 18819 18875 18969 19047 19154 19393
## [133] 19440 19523 19919 19989 20277 20423 20572 20587 21050 21141 21243 21825
## [145] 21846 21853 21994 22210 22307 22385 22551 22714 22949 23100 23173 23390
## [157] 23392 23461 23470 23513 23643 23661 23896 23998 24371 24476 24569 24725
## [169] 24862 25123 25139 25168 25321 25324 25347 25357 25706 25710 25799 25824
## [181] 25960 25979 26148 26243 26393 26422 26589 26631 26749 26773 26851 26880
## [193] 27173 27215 27587 27604 27720 27757 27802 27832 28061 28356 28404 28471
## [205] 28472 28891 29256 29439 29485 29676 29744 29768 29982 30004 30029 30198
## [217] 30515 30548 30603 30759 30800 30829 30985 31101 31156 31495 31644 31674
## [229] 31692 31842 31994 32150 32384 32395 32469
```

```
#Imputar en los registros "f"
new.ds.fem<- kNN( ds[ ds$gender=="f", variables.num], variable="hours_per_week", k=11)
ds[fem.idx, variables.num]
```

```
##       age education_num hours_per_week income
## 1272   28            12             NA  41.18
## 2015   31            15             NA  41.42
## 4294   28            15             NA  38.55
## 4348   40            10             NA  41.34
## 5432   44            15             NA  44.44
## 5489   41            10             NA  43.92
## 8072   44            12             NA  38.97
## 8780   39            16             NA  43.27
## 10728  61             9             NA  45.72
## 10850  35             4             NA  39.03
## 10954  68             6             NA  42.84
## 12788  24            16             NA  41.87
## 14368  48             9             NA  40.46
## 14593  27            13             NA  46.07
## 14767  39             9             NA  43.36
## 15016  62             9             NA  40.87
## 15534  29             9             NA  40.45
## 16036  28            14             NA  46.60
## 17440  35             7             NA  43.15
## 17611  28             4             NA  39.02
## 18603  33            11             NA  40.54
## 19141  59             4             NA  44.64
```

```
## 19731  34            16             NA  46.91
## 19997  72             4             NA  39.73
## 20036  61            14             NA  44.72
## 20878  61            12             NA  49.28
## 22190  31             9             NA  41.96
## 22237  62             9             NA  40.20
## 23226  51            12             NA  46.81
## 23749  31            16             NA  45.34
## 23830  45            10             NA  40.39
## 23925  41            13             NA  48.67
## 25674  28            14             NA  41.26
## 26145  25            12             NA  37.14
## 28112  61             9             NA  38.61
## 28625  20             9             NA  35.58
## 29187  31             9             NA  35.88
## 30754  31            15             NA  42.74
## 31597  49            14             NA  46.39
## 31754  26            13             NA  52.03
## 32469  58            16             NA  41.94
## 32531  30            13             NA  36.70
```

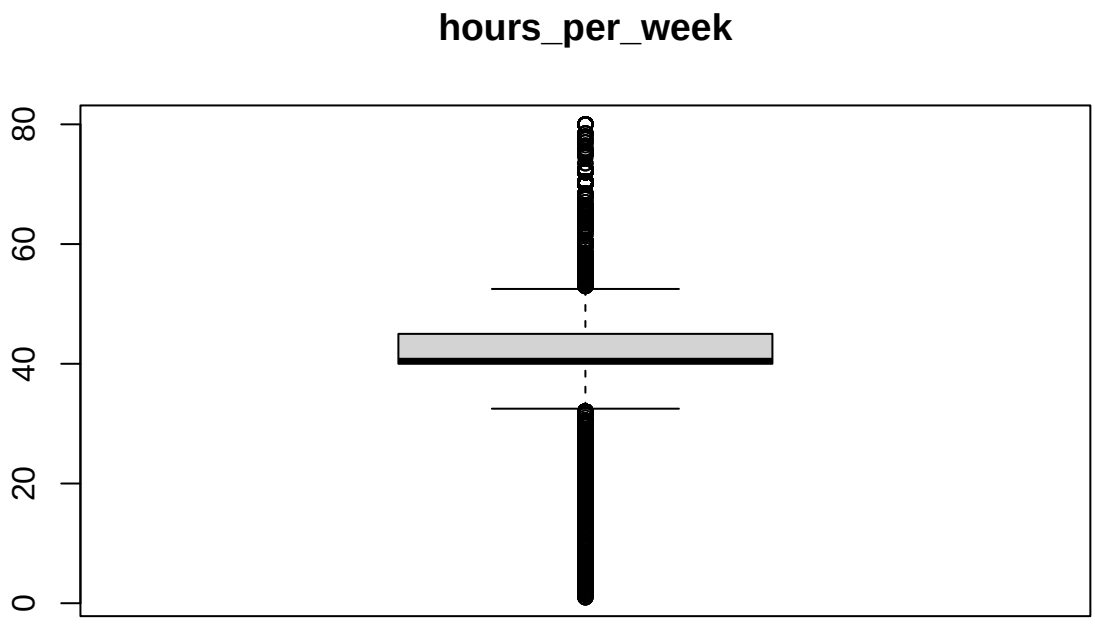
```
new.ds.fem[new.ds.fem$hours_per_week==TRUE,]
```

```
##      age education_num hours_per_week income hours_per_week_imp
## 412   69            11              1  37.52              FALSE
## 1826  78             9              1  33.75              FALSE
## 2792  67            11              1  36.67              FALSE
```

```
ds[fem.idx,]$hours_per_week <- new.ds.fem[new.ds.fem$hours_per_week_imp==TRUE,]$hours_per_week
#Imputar en los registros "m"
new.ds.mas <- kNN( ds[ ds$gender=="m", variables.num], variable="hours_per_week", k=11)
ds[mas.idx,]$hours_per_week <- new.ds.mas[new.ds.mas$hours_per_week_imp==TRUE,]$hours_per_week

sum( complete.cases(ds$hours_per_week) )
```

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

```
boxplot( ds$hours_per_week, main="hours_per_week")
```



8. Estudio descriptivo

8.1. Funciones de media robustas

Implementad una función en R que, dado un vector con datos numéricos, calcule la media recortada y la media Winsor. Estas funciones se deben definir como sigue:

```
media.recortada <- function( x, perc=0.05){}

media.winsor( x, perc=0.05){}
```

donde x es el vector de datos y perc la fracción de los datos a recortar (por defecto, 0.05). Implementad estas funciones en R y comprobad que funcionan correctamente.

```
perc<-0.05
mitjana.retallada <- function( x, perc=0.05 ){
  x.sorted <- sort(x)
  n <- length(x.sorted)
  low <- trunc(n*perc)+1
  high <- n-low+1
  x.sorted <-x.sorted[ low:high ]
  return (mean(x.sorted,na.rm=TRUE))
}
mitjana.winsor <- function( x, perc=0.05 ){
  x.sorted <- sort(x)
  n <- length(x.sorted)
  low <- trunc(n*perc)+1
```

Cuadro 1: Estimacions de Tendència Central

variables	Media	Mediana	Media.recort.0.05	Media.winsor.0.05
age	38.55	37.00	37.99	38.29
education_num	10.08	10.00	10.17	10.10
hours_per_week	40.31	40.50	40.52	40.39
income	48.75	49.71	48.86	48.76

```
    high <- n-low+1
    x.sorted[1:(low-1)] <- x.sorted[low]
    x.sorted[(high+1):n]<- x.sorted[high]
    return (mean(x.sorted,na.rm=TRUE))
  }

  #test
  a<-ds$age
  mitjana.retallada(a, 0.05); mean(a, trim=0.05, na.rm=TRUE); mean(a,na.rm=TRUE);
```

```
## [1] 37.99116
## [1] 37.99116
## [1] 38.54987
```

```
  mitjana.winsor(a); winsor.mean(a,0.05)
```

```
## [1] 38.29192
## [1] 38.29192
```

8.2. Estudio descriptivo de las variables cuantitativas

Realizad un estudio descriptivo de las variables cuantitativas `age`, `education_num`, `hours_per_week` y `income`.

Para ello, preparad una tabla con varias medidas de tendencia central y dispersión, robustas y no robustas. Usad, entre otras, las funciones del apartado anterior. Presentad, asimismo gráficos donde se visualice la distribución de los valores de estas variables cuantitativas.

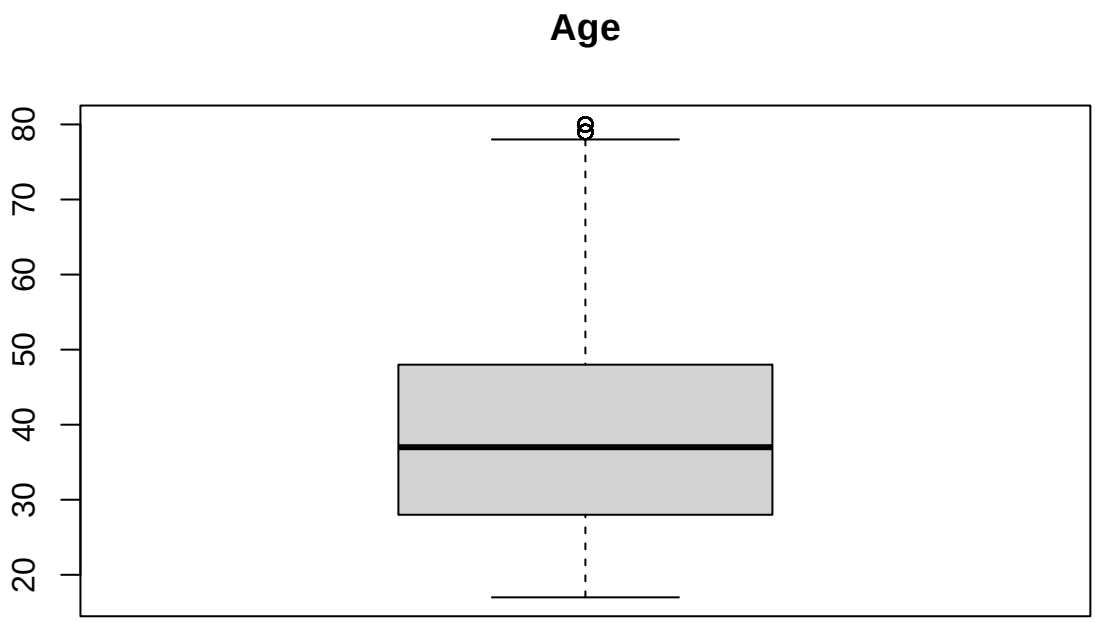
```
idx.numeric <- which( colnames(ds) %in% c("age", "education_num", "hours_per_week", "income") )
mean.n   <- as.vector(sapply( ds[,idx.numeric ],mean,na.rm=TRUE ) )
std.n    <- as.vector(sapply(ds[,idx.numeric ],sd, na.rm=TRUE))
median.n <- as.vector(sapply(ds[,idx.numeric], median, na.rm=TRUE))
mean.trim.0.05     <- as.vector(sapply(ds[,idx.numeric],  mitjana.retallada, perc=0.05))
mean.winsor.0.05   <- as.vector(sapply(ds[,idx.numeric], mitjana.winsor, perc=0.05))
IQR.n <- as.vector(sapply(ds[,idx.numeric],IQR, na.rm=TRUE))
mad.n <- as.vector(sapply(ds[,idx.numeric],mad, na.rm=TRUE))

kable(data.frame(variables= names(ds)[idx.numeric],
                 Media = mean.n,
                 Mediana = median.n,
                 Media.recort.0.05= mean.trim.0.05,
                 Media.winsor.0.05= mean.winsor.0.05
                 ),
      digits=2, caption="Estimacions de Tendència Central")
```

Cuadro 2: Estimacions de Dispersió

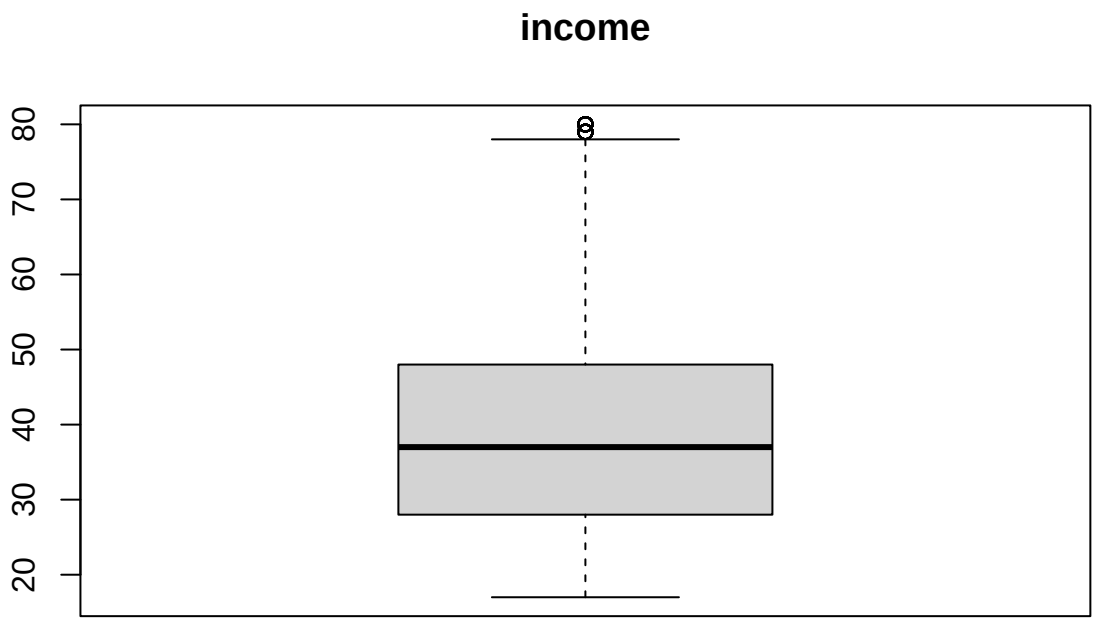
variables	Desv.Standard	IQR	MAD
age	13.54	20.0	14.83
education_num	2.57	3.0	1.48
hours_per_week	11.47	5.0	4.45
income	7.10	11.1	7.90

```
kable(data.frame(variables= names(ds)[idx.numeric],
                 Desv.Standard = std.n,
                 IQR = IQR.n,
                 MAD = mad.n
                 ),
      digits=2, caption="Estimacions de Dispersió")
```

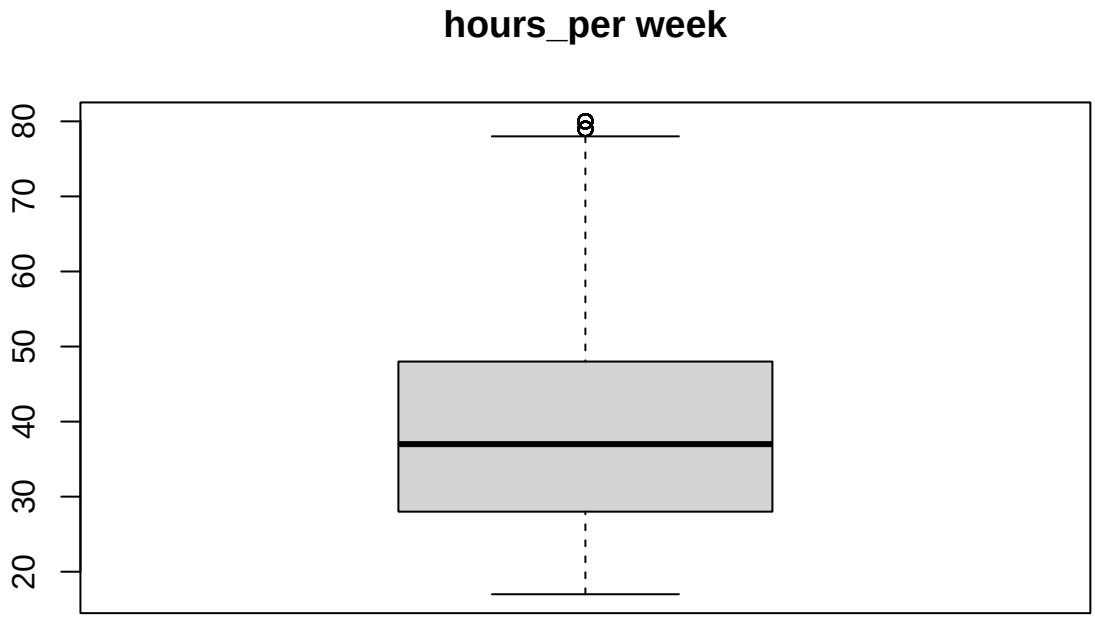
```
#Gráficos
boxplot(ds$age, main="Age")
```




```
boxplot(ds$age, main="income")
```



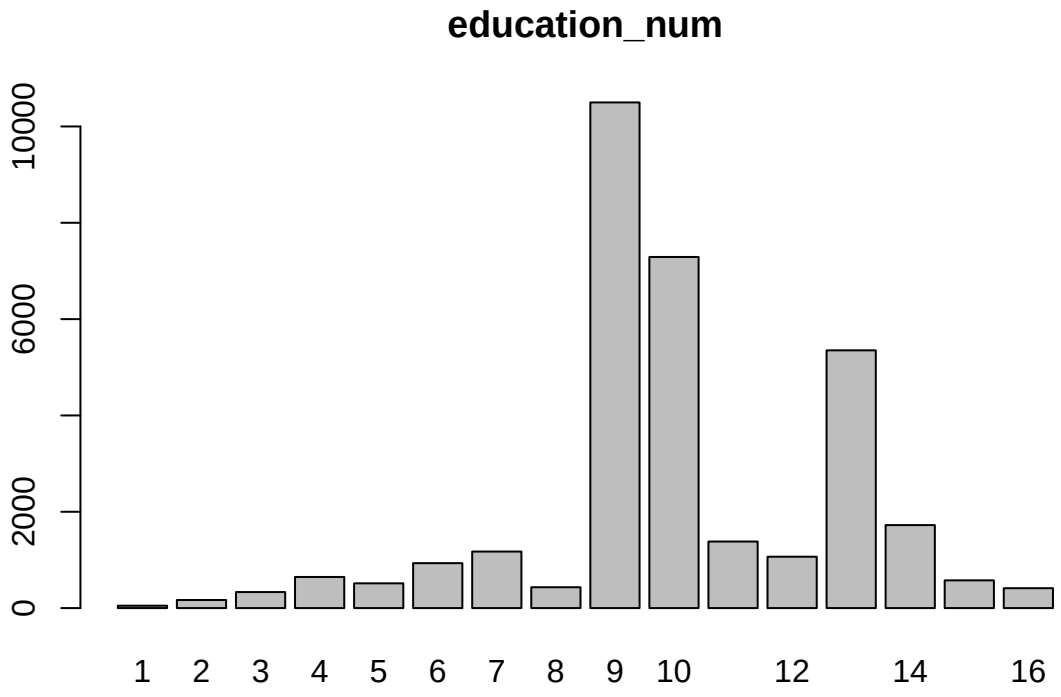

```
boxplot(ds$age, main="hours_per week")
```




```
table(ds$education_num) %>% kable()
```

Var1	Freq
1	50
2	168
3	333
4	646
5	514
6	933
7	1174
8	433
9	10499
10	7290
11	1382
12	1067
13	5352
14	1723
15	576
16	413

```
barplot( table(ds$education_num), main="education_num")
```



9. Archivo final

Una vez realizado el preprocesamiento sobre el archivo, copiad el resultado de los datos en un archivo llamado **CensusIncome_clean.csv**.

```
write.csv(ds, "CensusIncome_clean.csv", row.names = FALSE)
```

10. Evaluación de la actividad

- Secciones 1, 2 (20 %)
- Secciones 3, 4 (10 %)
- Sección 5 (10 %)
- Sección 6 (10 %)
- Sección 7 (10 %)
- Sección 8 (20 %)
- Seccion 9 (10 %)
- Calidad del informe dinámico (calidad del código, formato y estructura del documento, concisión y precisión en las respuestas) (10 %)