A1

Marc Garrido Casas

25 de març, 2020

Contents

1 Càrrega de l'arxiu de dades i breu descripció

Obrir el fitxer BWn.csv i examinar el tipus de dades amb què R ha interpretat cada variable. Avaluar també els valors resum de cada tipus de variable.

```
setwd("C:/Users/garridom/Documents/UOC/Estadistica/A1")
df <- read.csv("BWn.csv", header = TRUE, sep = ';', row.names=1)
head(df)
```

```
dfOriginal<-data.frame(df)
#sapply(df, class) # El head() ja mostra el tipus de variables.
```

2. Normalització de les variables qualitatives

2.1 Hospital, Ciutat

En primer lloc, s'examinaran les possibles inconsistències en les variables HP i City. Detecteu si hi ha diversos noms per a un mateix hospital i/o per a una mateixa ciutat i si és així, normalitzar els noms, segons la llista de noms proporcionada anteriorment. Comproveu així mateix la possible inconsistència entre l'hospital i la seva ciutat. En cas de incosistencia, identificar-la, reportar-la i introduir els canvis necessaris. S'han de seguir els criteris especificats pel preprocessat.

```
# Tots els hospitals que existeixen dins el dataset.
summary(df[1], maxsum = 20)
```

```
##                              HP
##  C.U.Navarra                  :27
##  H.Clínic                     :25
##  H.G.U.Gregorio Marañón       :27
##  H.M.I. Virgen de las Nieves:30
```

```
##  H.U.12 de octubre          :26
##  H.U.de Bellvitge           :40
##  H.U.La Paz                 :18
##  H.U.Politècnic La Fe       :31
##  H.U.Quirón Dexeus          :36
##  H.U.Vall d'Hebron          :26
##  H.U.Virgen del Rocío       :14
```

```
# Els hospitals són correctament escrits.

# Totes les ciutats que existeixen dins el dataset.
summary(df[2], maxsum = 20)
```

```
##         City
##  Barcelona:124
##  Granada  : 28
##  Madrid   : 71
##  Pamplona : 27
##  sevilla  :  1
##  Sevilla  : 14
##  valencia :  4
##  Valencia : 31
```

```
# Veig que Valencia i Sevilla, pot estar escrit sense la primera majúscula.

# Modificació
df$City <- gsub('valencia', 'Valencia', df$City)
df$City <- gsub('sevilla', 'Sevilla', df$City)
```

```
# Miro el nom dels hospitals per a cada ciutat.
by(df[1], df$City, summary)
```

```
## df$City: Barcelona
##                         HP
##  H.U.de Bellvitge      :38
##  H.U.Quirón Dexeus     :36
##  H.U.Vall d'Hebron     :26
##  H.Clínic              :24
##  C.U.Navarra           : 0
##  H.G.U.Gregorio Marañón: 0
##  (Other)               : 0
## ------------------------------------------------------------
## df$City: Granada
##                              HP
##  H.M.I. Virgen de las Nieves:28
##  C.U.Navarra                 : 0
##  H.Clínic                    : 0
##  H.G.U.Gregorio Marañón      : 0
##  H.U.12 de octubre           : 0
##  H.U.de Bellvitge            : 0
##  (Other)                     : 0
## ------------------------------------------------------------
## df$City: Madrid
##                              HP
##  H.G.U.Gregorio Marañón      :26
```

```
##  H.U.12 de octubre          :26
##  H.U.La Paz                 :17
##  H.M.I. Virgen de las Nieves: 1
##  H.U.de Bellvitge           : 1
##  C.U.Navarra                : 0
##  (Other)                    : 0
## ------------------------------------------------------------
## df$City: Pamplona
##                            HP
##  C.U.Navarra                :27
##  H.Clínic                   : 0
##  H.G.U.Gregorio Marañón     : 0
##  H.M.I. Virgen de las Nieves: 0
##  H.U.12 de octubre          : 0
##  H.U.de Bellvitge           : 0
##  (Other)                    : 0
## ------------------------------------------------------------
## df$City: Sevilla
##                            HP
##  H.U.Virgen del Rocío       :14
##  H.M.I. Virgen de las Nieves: 1
##  C.U.Navarra                : 0
##  H.Clínic                   : 0
##  H.G.U.Gregorio Marañón     : 0
##  H.U.12 de octubre          : 0
##  (Other)                    : 0
## ------------------------------------------------------------
## df$City: Valencia
##                         HP
##  H.U.Politècnic La Fe  :31
##  H.Clínic              : 1
##  H.G.U.Gregorio Marañón: 1
##  H.U.de Bellvitge      : 1
##  H.U.La Paz            : 1
##  C.U.Navarra           : 0
##  (Other)               : 0
```

```r
# En efecte, alguns hospitals no concorden amb al ciutat.

# Modificació
df[which(df[,1]=="H.U.de Bellvitge"),2] <- "Barcelona"
df[which(df[,1]=="H.Clínic"),2] <- "Barcelona"
df[which(df[,1]=="H.G.U.Gregorio Marañón"),2] <- "Madrid"
df[which(df[,1]=="H.U.La Paz"),2] <- "Madrid"
df[which(df[,1]=="H.M.I. Virgen de las Nieves"),2] <- "Granada"

# Per a comprobar-ho, miro la ciutat de Valencia, que tenia uns quants hospitals mal indicats.
by(df[1], df$City == "Valencia", summary)
```

```
## df$City == "Valencia": FALSE
##                            HP
##  H.U.de Bellvitge           :40
##  H.U.Quirón Dexeus          :36
##  H.M.I. Virgen de las Nieves:30
##  C.U.Navarra                :27
##  H.G.U.Gregorio Marañón     :27
```

```
##  H.U.12 de octubre          :26
##  (Other)                    :83
## ------------------------------------------------------------
## df$City == "Valencia": TRUE
##                              HP
##  H.U.Politècnic La Fe       :31
##  C.U.Navarra                : 0
##  H.Clínic                   : 0
##  H.G.U.Gregorio Marañón     : 0
##  H.M.I. Virgen de las Nieves: 0
##  H.U.12 de octubre          : 0
##  (Other)                    : 0
```

2.2 Dia de la setmana

Verificar que els valors de la variable dia són correctes i normalitzar la variable dia segons les indicacions proporcionades. La variable ha de ser de tipus categòric (en R, factor).

```
# He de substituïr el número pel nom del día de la setmana.
head(df$Day)
```

```
## [1] 5 1 2 4 3 6
```

```
# Modificació
df$Day <- gsub('1', 'Dilluns', df$Day)
df$Day <- gsub('2', 'Dimarts', df$Day)
df$Day <- gsub('3', 'Dimecres', df$Day)
df$Day <- gsub('4', 'Dijous', df$Day)
df$Day <- gsub('5', 'Divendres', df$Day)
df$Day <- gsub('6', 'Dissabte', df$Day)
df$Day <- gsub('7', 'Diumenge', df$Day)

head(df$Day)
```

```
## [1] "Divendres" "Dilluns"   "Dimarts"   "Dijous"    "Dimecres"  "Dissabte"
```

2.3 Sexe

Normalitzar la variable sexe (Sex) segons les indicacions proporcionades.

```
# He de substituïr el número pel nom del día de la setmana.
summary(df[8], maxsum = 20)
```

```
##    Sex
##  boy : 14
##  f   :  5
##  F   :149
##  fem :  8
##  girl:  6
##  M   :118
```

```r
# Modificació
df$Sex <- gsub('boy', 'M', df$Sex)
df$Sex <- gsub('fem', 'F', df$Sex)
df$Sex <- gsub('girl', 'F', df$Sex)

# Vec que la forma més ràpida és mostrar tots els possibles Strings que hi apareixen. Després substituïrlo

# Convertir en majúscules.
df$Sex <- toupper(df$Sex)
```

2.4 Mare fumadora

Normalitzar els valors de la variable mare fumadora (Sm) segons les indicacions proporcionades.

```r
# He de substituïr el número pel nom del día de la setmana.
summary(df[10], maxsum = 20)
```

```
##  Sm
##  N:238
##  S: 62
```

```r
# Ja és correcte.
```

3 Normalització de les variables quantitatives

Revisar el tipus de dada i el format de les variables que han de ser quantitatives. Convertiu a tipus numèric si les variables no s'han carregat amb aquest tipus. Abans, però, cal corregir les possibles inconsistències en el punt decimal. Reviseu, per a cada variable quantitativa el format especificat en els criteris de preprocés.

3.1 Pes

Tranformar la variable pes per convertir-la en un format numèric en grams. Per extreure el valor numèric d'un string en R, podeu fer servir expressions regulars i la funció str_extract. Podeu trobar informació a: https://stringr.tidyverse.org/articles/regular-expressions.html

```r
head(df[5])
```

```r
#Poden haver-hi valors en quilos, valors amb coma, amb punt. No se sap quants decimals hi haurà després de

# Quedar-se amb els números només. Mantenint punts i comes.
df$BW <- gsub("[^[:digit:].,]", "", df$BW)

# Passar-ho a numeric
df[5] <- lapply(df[5], as.numeric)
```

```
## Warning in lapply(df[5], as.numeric): NAs introduced by coercion
```

```r
# Per a cada cel·la de la columna on el pes sigui més petit que 10. Multipliquem el valor per a 1000, pase
#És evident que la mesura és pressa en quilos, ningún nadó pesa més de 10 quilos o menys de 10 grams.
df[which(df[,5]<10),5] <- df[which(df[,5]<10),5]*1000

head(df[5])
```

3.2 Diàmetre biparietal i Diàmetre abdominal

Normalitzar les variables diàmetre biparietal i diàmetre abdominal segons les indicacions proporcionades.

```
head(df[6])
```

```
# Sembla que només m'he de quedar amb el número i pasar la columna a numeric.

# Quedar-se amb els números només. Mantenint punts i comes.
df$BPD <- gsub("[^[:digit:].,]", "", df$BPD)

# Passar-ho a numeric
df[6] <- lapply(df[6], as.numeric)

head(df[6])
```

3.3 Setmanes de gestació

Normalitzar la variable setmanes de gestació segons els criteris establerts.

```
head(df[7])
```

```
df$AD <- gsub("[^[:digit:].,]", "", df$AD)
df[7] <- lapply(df[7], as.numeric)
head(df[7])
```

```
head(df[9])
```

```
# Substituir comas per punts.
df$Ge <- gsub(',', '.', df$Ge)

#Convertir a numeric.
df[9] <- lapply(df[9], as.numeric)

#Arrodonir al valor més pròxim.
df[9] <-round(df[9],0)

head(df[9])
```

3.4 Hora

Transformar la variable hora a format HH: MM. Podeu fer servir llibreries o realitzar la transformació a partir del vostre propi codi.

```
head(df[3])
```

```
library(stringr)
str_split_fixed(df$time, ":", 2)
```

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

```
for(time in df[3]) {
  print(time)
  str_split(time, ",")

}
```

```
##   [1] 13,34 11    24    18,64 9,89  5,68  9,55  6,86  8,43  12,87 24    14,72
##  [13] 9,65  11,39 10,68 24    10,46 17,82 16,15 16,04 13,18 8,67  16,35 10,56
##  [25] 9,58  10,25 18,06 24    10,66 10,61 10,99 9,85  8,77  10,9  23,15 13,79
##  [37] 7,4   9,29  11,94 10,46 4     7,45  7,4   8,96  8,63  7,65  14,28 14,82
##  [49] 8,82  24    10,15 3,43  18,48 15,46 8,41  9,66  10,77 5,6   19,34 7,4
##  [61] 10,94 11,01 15,54 18,46 9,53  8,95  7,73  10,09 6,56  19,8  9,69  7,14
##  [73] 3,73  11,49 11,94 13,22 9,44  9,17  11,05 14,8  9,63  14,12 8,74  12,06
##  [85] 12,81 11,51 11,83 10,83 11,78 14,69 12,74 7,58  24    9,74  24    9,36
##  [97] 10,5  13,75 8,91  5,58  1,87  9,83  8,01  10,56 9,19  4,21  8,93  8,36
## [109] 13,39 13,21 8,51  23,56 10,04 13,27 7,94  12,65 6,66  13,78 16,96 16,38
## [121] 12,13 9,05  9,8   12,29 12,22 5,65  21,4  14,01 11,27 13,06 4,81  11,99
## [133] 13,58 6,72  17,17 10,4  16,1  8,8   10,4  13,54 14,71 4,06  16,43 12,58
## [145] 24    8,26  17,84 13,73 20,57 9,98  4,6   13,8  8,81  13,53 16,31 7,27
## [157] 2,64  14,81 3,66  9,84  6,84  13,26 18,89 3,26  15,29 8,69  12,27 8,7
## [169] 13,61 9,8   12,97 10,75 7,32  5,69  12,11 15,4  20,85 7,75  16,05 13,92
## [181] 17,77 6,27  9,54  7,69  10,88 11,07 7,69  7,24  7,27  11,13 15,28 15,07
## [193] 13,18 12,67 15    14    8,53  7,34  13,11 18,31 3,99  13,14 9,57  6,84
## [205] 12,72 6,63  24    10,54 14,33 7,11  6,92  8,02  12,95 13,29 10,32 8,98
## [217] 13,43 11,92 4,76  12,23 13,12 6,26  10,9  10,63 7,82  9,16  12,49 13,3
## [229] 17,32 21,85 11,34 13,17 15,97 8,48  9,19  12,02 9,42  12,43 19,54 8,38
## [241] 14,26 9,35  23    24    12,32 19,71 14,34 8,07  12,72 12,94 6,42  13,21
## [253] 5,39  11,73 9,87  9,94  23,72 8,86  9,58  12,98 15,93 9,35  9,35  11,56
## [265] 7,27  7,98  11,8  16,2  8,42  13,71 13,57 7,02  15,56 8,24  7,32  22,16
## [277] 16,17 13,08 21,29 14,11 7,77  6,84  10,94 24    10,74 0     13,78 2,6
## [289] 8,94  8     4,26  21,43 16,53 7,39  11,38 18,1  16,59 12,63 11,59 5,85
## 267 Levels: 0 1,87 10,04 10,09 10,15 10,25 10,32 10,4 10,46 10,5 10,54 ... 9,98
```

```
#sapply(strsplit(dat,":"),
#       function(x) {
#         x <- as.numeric(x)
#         x[1]+x[2]*60/100
#       }
#)

df$Sex <- toupper(df$Sex)


# Substituir comas per punts.
df$Ge <- gsub(',', '.', df$Ge)

#Convertir a numeric.
df[9] <- lapply(df[9], as.numeric)

#Arrodonir al valor més pròxim.
df[9] <-round(df[9],0)

df[3]
```

4 Valors perduts

Analitzar la presència de valors perduts. En el cas de detectar algun valor perdut en les variables quantitatives, realitzar una imputació de valors en aquestes variables. La imputació s'ha de fer amb els 3 veïns més propers usant la distància de Gower, usant només la informació de les variables quantitatives i dins d'aquestes, aquelles que tinguin sentit en la imputació de la variable. Després de realitzar la imputació cal verificar que els valors assignats s'han copiat sobre el conjunt de dades originals. Visualitzar el resultat de les imputacions realitzades (per evitar mostrar tot el conjunt de dades, només s'han de mostrar els registres del conjunt de dades que contenen la imputació realitzada).

```
sum(is.na(df))
```

```
## [1] 18
```

```
# Hi ha 18 valors nuls en tot el dataset.
# Amb la següent comanda, vec en quina fila i columna es troben aquests
which(is.na(df), arr.ind=TRUE)
```

```
##     row col
## 36   36   5
## 43   43   5
## 45   45   5
## 24   24   6
## 65   65   6
## 101 101   6
## 159 159   6
## 183 183   6
## 186 186   6
## 192 192   6
## 218 218   6
## 220 220   6
## 225 225   6
## 53   53   7
## 114 114   7
## 127 127   7
## 202 202   7
## 242 242   7
```

```
# Veure d'aprop la transforimació d'un valor nul.
head(df[33:39,5], 7)
```

```
## [1] 2600 1722 4100   NA 2132 2950 1700
```

```
#install.packages("VIM")
#aplicar KNN (Gower distnace)
library(VIM)
```

```
## Loading required package: colorspace
```

```
## Loading required package: grid
```

```
## Loading required package: data.table
```

```
## VIM is ready to use.
##  Since version 4.0.0 the GUI is in its own package VIMGUI.
##
##           Please use the package to use the new (and old) GUI.
```

```
## Suggestions and bug-reports can be submitted at: https://github.com/alexkowa/VIM/issues
```

```
##
## Attaching package: 'VIM'
```

```
## The following object is masked from 'package:datasets':
##
##     sleep
```

```
df <- kNN(df, k=3)
```

```
## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx, numericalX, :
## NAs introduced by coercion

## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx, numericalX, :
## NAs introduced by coercion

## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx, numericalX, :
## NAs introduced by coercion

## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx, numericalX, :
## NAs introduced by coercion

## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx, numericalX, :
## NAs introduced by coercion

## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx, numericalX, :
## NAs introduced by coercion

## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx, numericalX, :
## NAs introduced by coercion

## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx, numericalX, :
## NAs introduced by coercion

## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx, numericalX, :
## NAs introduced by coercion

## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx, numericalX, :
## NAs introduced by coercion

## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx, numericalX, :
## NAs introduced by coercion

## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx, numericalX, :
## NAs introduced by coercion

## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx, numericalX, :
## NAs introduced by coercion
```

```
## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx, numericalX, :
## NAs introduced by coercion

## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx, numericalX, :
## NAs introduced by coercion

## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx, numericalX, :
## NAs introduced by coercion

## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx, numericalX, :
## NAs introduced by coercion

## Warning in gowerD(don_dist_var, imp_dist_var, weights = weightsx, numericalX, :
## NAs introduced by coercion
```

```
head(df[33:39,5], 7)
```

```
## [1] 2600 1722 4100 3200 2132 2950 1700
```

```
sum(is.na(df))
```

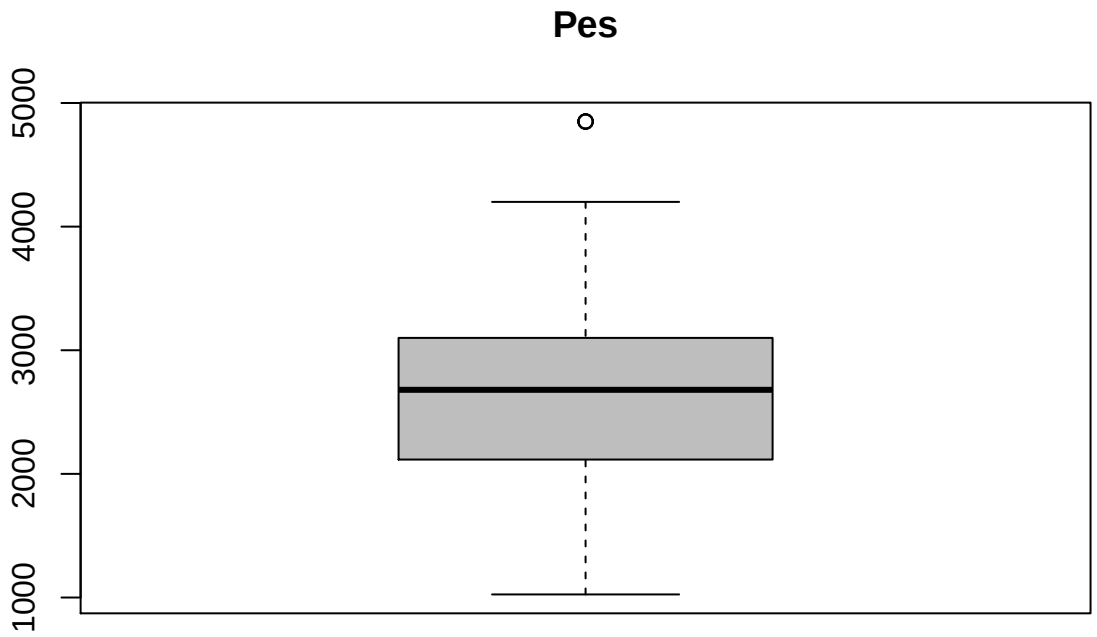
```
## [1] 0
```

```
# Ja no queden valors nuls
```

5 Valors extrems

Analitzar la presència de possibles valors extrems (outliers) en les variables pes, diàmetre biparietal i diàmetre abdominal. Per això, dibuixeu diagrames de caixa i també feu servir els resultats de la funció boxplot.stats. Un cop identificats, investigar les possibles causes d'aquests valors extrems i decidiu una estratègia apropiada, en funció d'aquesta anàlisi. L'estratègia pot ser eliminar els valors extrems, realitzar imputacions sobre els valors extrems o simplement mantenir els valors extrems pel seu valor explicatiu en el conjunt de dades. Justifiqueu les vostres eleccions.

```
b <- boxplot(df[5],main="Pes", col="gray")
```




```
# En efecte, amb el boxplot vec un outlayer.

# Veiem aquest outlier.
b$out
```

```
## [1] 4850 4850 4850 4850
```

```
# És una mica sospitós que quatre nens pesin exactament 4.850kg.

# On són localitzats aquest outliers.
which(df[5] == 4850, arr.ind=TRUE)
```

```
##      row col
## [1,] 133   1
## [2,] 151   1
## [3,] 199   1
## [4,] 264   1
```

```
df[133,5]
```

```
## [1] 4850
```

```
dfOriginal[133,5]
```

```
## [1] 4850  gr
## 91 Levels: 1.45  kg 1.65  kg 1.7  kg 1.804  kg 1025  gr 1150  gr ... 4850  gr
```

```r
df[151,5]
```

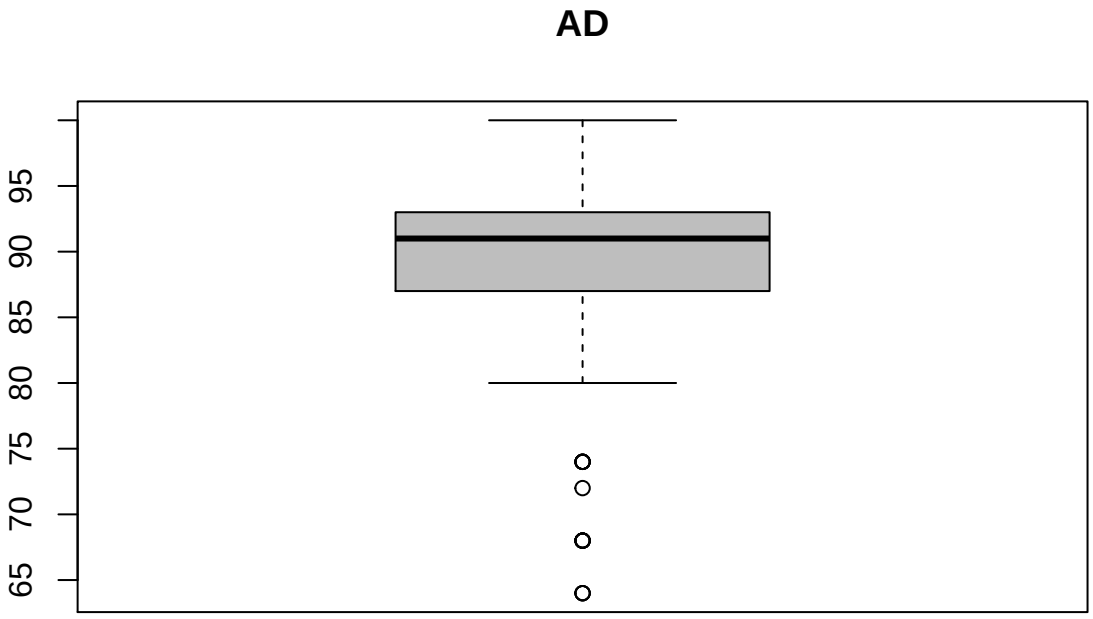
```
## [1] 4850
```

```r
dfOriginal[151,5]
```

```
## [1] 4.85  kg
## 91 Levels: 1.45  kg 1.65  kg 1.7  kg 1.804  kg 1025  gr 1150  gr ... 4850  gr
```

```r
# En l'original són 4.85 kg, no m'he equivocat en la conversió.


# Si miro aquestes quatre columnes
df[c(133,151,199,264),]
```

```r
# Vec que són de quatre hospitals de ciutats diferents, en hores i dies diferetns. Les setmanes de gestaci

# Per aquesta raó opto per deixar el outlayer, ja que he vist que no sembla una errata de preprocessament


# Pel que fa als diàmetres


b <- boxplot(df[6],main="AD", col="gray")
```



```r
b$out
```

```
##  [1] 74 68 74 64 68 64 68 68 74 68 64 74 72 68 74
```

```r
# On són localitzats aquest outliers.
which(df[6] < 75, arr.ind=TRUE)
```

```
##       row col
##  [1,]   4   1
##  [2,]   5   1
##  [3,]  16   1
##  [4,]  63   1
##  [5,]  78   1
##  [6,]  95   1
##  [7,] 127   1
##  [8,] 134   1
##  [9,] 140   1
## [10,] 155   1
## [11,] 194   1
## [12,] 210   1
## [13,] 240   1
## [14,] 248   1
## [15,] 286   1
```

```r
df[c(4,5,16,63,78,95,127,134,140,155,194,210,240,248,286),]
```