A1.Financiacion del Gasto Sanitario

Alicia Perdices Guerra

8 de abril, 2021

Contents

1.PROCESAMIENTO DE LOS DATOS.

- En primer lugar leemos el fichero:

```
gasto_f<-read.csv("C:/temp/GastoSanitario_Financiacion.csv",sep= ",")
```

- Realicemos una breve inspección de los datos

```
str(gasto_f)
```

```
## 'data.frame':    2000 obs. of  6 variables:
##  $ TIME              : int  2009 2009 2009 2009 2009 2009 2009 2009 2009 2009 ...
##  $ GEO               : Factor w/ 40 levels "Austria","Belgium",..: 15 15 15 15 15 16 16 16 16 16 ...
##  $ UNIT              : Factor w/ 1 level "Million euro": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ICHA11_HF         : Factor w/ 5 levels "All financing schemes",..: 1 4 3 2 5 1 4 3 2 5 ...
##  $ Value             : Factor w/ 1185 levels ":","0.00","1,001,514.67",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Flag.and.Footnotes: Factor w/ 2 levels "","b": 1 1 1 1 1 1 1 1 1 1 ...
```

```
colnames(gasto_f) #Nombre de las variables
```

```
## [1] "TIME"               "GEO"                "UNIT"
## [4] "ICHA11_HF"          "Value"              "Flag.and.Footnotes"
```

```
nrow(gasto_f) #Número de registros
```

```
## [1] 2000
```

```
ncol(gasto_f) #Número de variables
```

```
## [1] 6
```

- Eliminamos la columna de Fal.and.footnotes.

```
gasto_f<-gasto_f[,-6]
```

- Tendríamos que convertir la columna Value a numérico porque se ha cargado como factor y es erróneo. El resto de variables tienen el tipo correcto.

```
gasto_f$Value<-as.character(gasto_f$Value)
gasto_f$Value<-as.numeric(gsub(",",".",gasto_f$Value))
```

```
## Warning: NAs introducidos por coerción
```

- Comprobamos que valores tenemos en la columna **Value**:

```
table(gasto_f$Value, useNA = "ifany")
```

```
##
##     0   3.3  3.61  3.97  4.14  7.62  8.18  8.66  8.98   9.2 27.15
##    89     2     2     2     2     2     2     2     2     2     1
## 29.05 29.57 31.85 35.55 38.21 41.06 43.46 48.56 49.34  49.4 52.64
##     1     1     1     1     1     1     1     1     1     1     1
```

```
##  53.72  56.22   60.1  63.88  76.89  77.39  77.55  86.03  86.96  94.17  98.47
##      1      1      4      4      1      1      2      1      1      1      1
##  98.54 101.75 106.77 112.06 113.21  118.9 118.98 119.65 124.32 124.85 126.05
##      1      1      1      1      1      1      1      1      1      1      1
## 128.66 134.51  136.3 136.51 137.89 138.32 138.35  140.9 141.72 142.39  144.7
##      1      1      1      2      1      1      1      2      1      1      2
## 145.98 146.56 150.44 151.76 156.33 157.83 164.87 164.88 175.06 177.33 179.69
##      1      2      1      1      2      1      2      2      2      1      2
## 179.86 181.71 181.89 182.71 184.36 185.77 188.98  192.7 196.68 197.75 198.88
##      1      2      2      2      1      1      1      1      1      1      2
## 199.31 200.27 205.73 210.27 213.43 213.47 214.22 221.22    227 228.79 234.86
##      2      1      1      1      1      1      1      1      2      1      1
## 241.57 245.24  245.3 251.37    260    267 276.68 283.02    310  320.5 322.95
##      1      1      1      1      2      2      1      1      1      1      1
##  324.9 325.71    326 329.92    341    343 344.32 350.12    351 371.58  372.6
##      1      1      2      1      2      2      1      1      2      1      1
## 376.89 389.38 413.42 428.65 436.61 456.59 462.82 487.53 509.65 513.62    514
##      1      1      2      1      2      1      2      2      1      1      1
## 521.62 525.79 529.93 538.74 554.77 558.38 567.86 575.04 576.04 581.57  583.7
##      1      1      1      2      1      1      1      1      1      1      1
## 590.54 595.69 601.79 606.83 608.41  609.7 610.98 611.91 614.77 615.21 622.18
##      1      2      1      1      1      2      1      1      1      1      2
## 626.87  637.5 642.03 648.02 655.41  655.5 687.93 690.49 700.26 704.42 711.45
##      1      2      1      1      1      2      2      1      1      2      1
## 719.29 720.72 723.15 735.91  739.8 744.27 750.62 767.47 770.23 795.04 795.88
##      2      1      1      2      2      1      2      1      2      1      2
## 801.14 802.61  814.6 854.93 860.23 869.34 883.13 883.87 889.47 892.41 898.48
##      1      2      2      2      1      2      1      1      1      2      1
## 907.03 910.28 916.43 922.85 925.55 928.65  932.1 936.07 937.77 939.05 945.12
##      2      1      2      2      1      1      1      1      2      1      1
##    965 967.34 970.09 970.49 974.37 977.46 991.84 997.32 999.83   <NA>
##      1      1      2      1      2      2      1      1      1   1649
```

- Observamos que tenemos **1649 valores perdidos**.Guardamos en la variable **idx** los índices de los registros con valores **NA** de la variable **Value**.

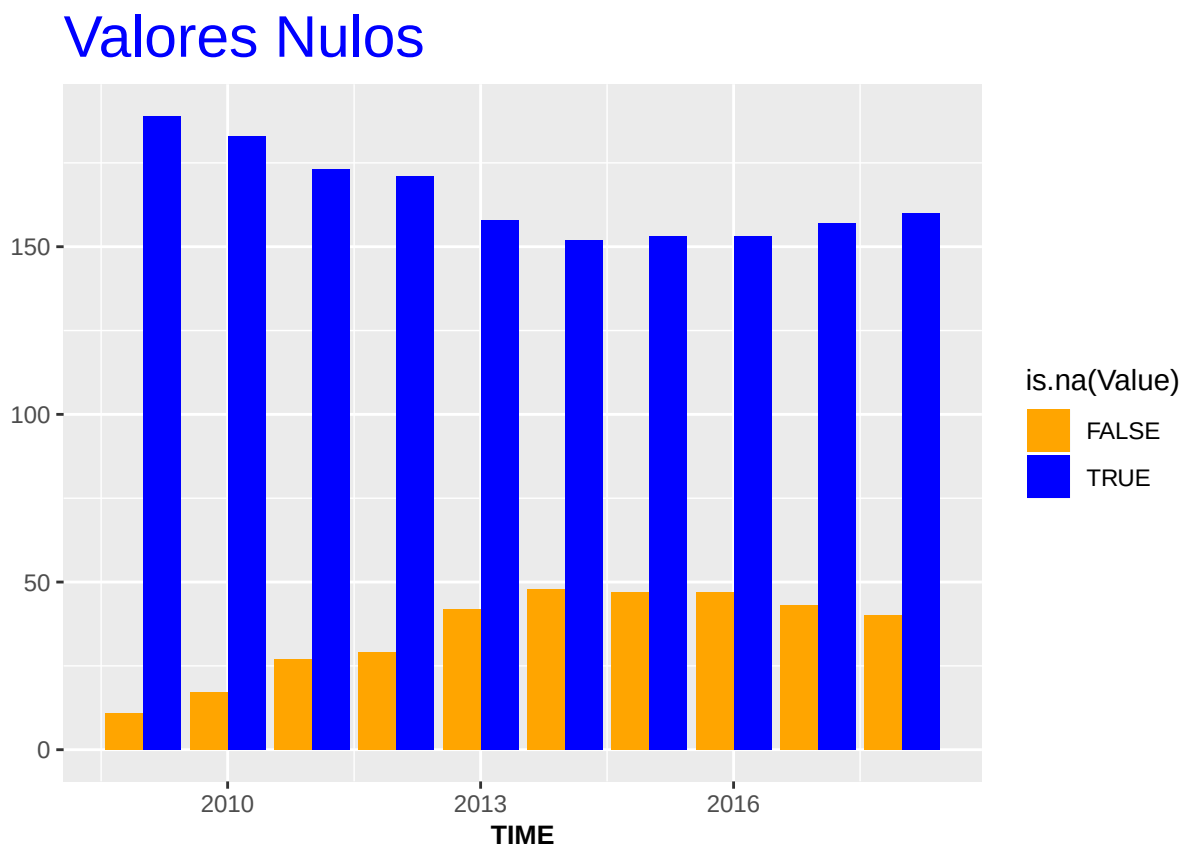
```
idx<-which(is.na(gasto_f$Value))
length(idx)
```

```
## [1] 1649
```

- Grafiquemos la información que contiene la variable **Value**

```
library(ggplot2)
library(scales)
g = ggplot(gasto_f, aes(TIME, fill=is.na(Value)) ) +
labs(title = "Valores Nulos")+ylab("") +
theme(plot.title = element_text(size = rel(2), colour = "blue"))

g+geom_bar(position="dodge") + scale_fill_manual(values = alpha(c("orange", "blue"), 1)) +
theme(axis.title.x = element_text(face="bold", size=10))
```



- En caso de detectar algún valor anómalo (en nuestro caso los NAS) en las variables tendríamos que realizar una imputación de esos valores o bien sustituyéndolos por la media o usando el algoritmo KNN (k-Nearest Neighbour) con los 3 vecinos más cercanos usando la distancia que consideremos, en este caso usaremos Gower(Mediana), por ser una medida más robusa frente a extremos.

```
library(VIM)
```

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

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

```
## VIM is ready to use.
```

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

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

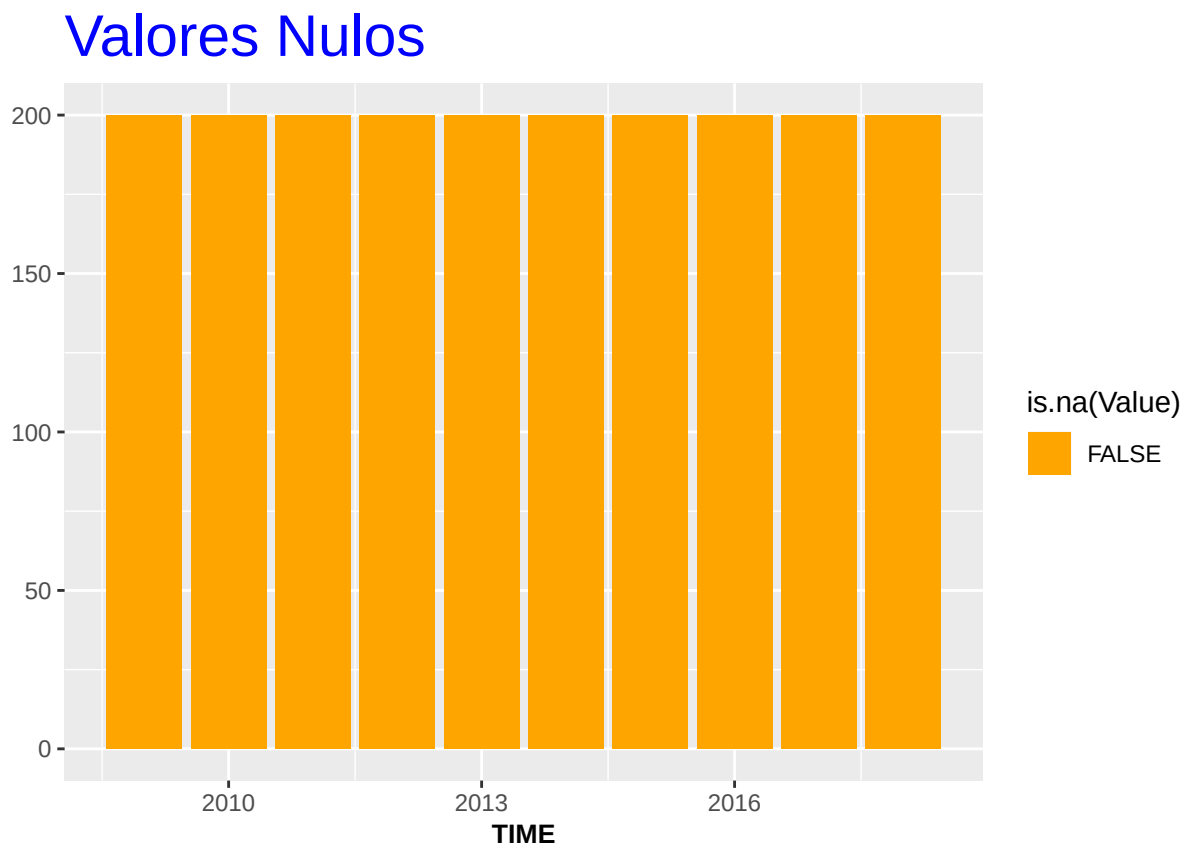
```
## The following object is masked from 'package:datasets':
##
##     sleep
```

```
output<-kNN(gasto_f, variable=c("Value"),k=3)
gasto_f<-output
```

- Comprobamos que no tenemos valores nulos después de la imputación

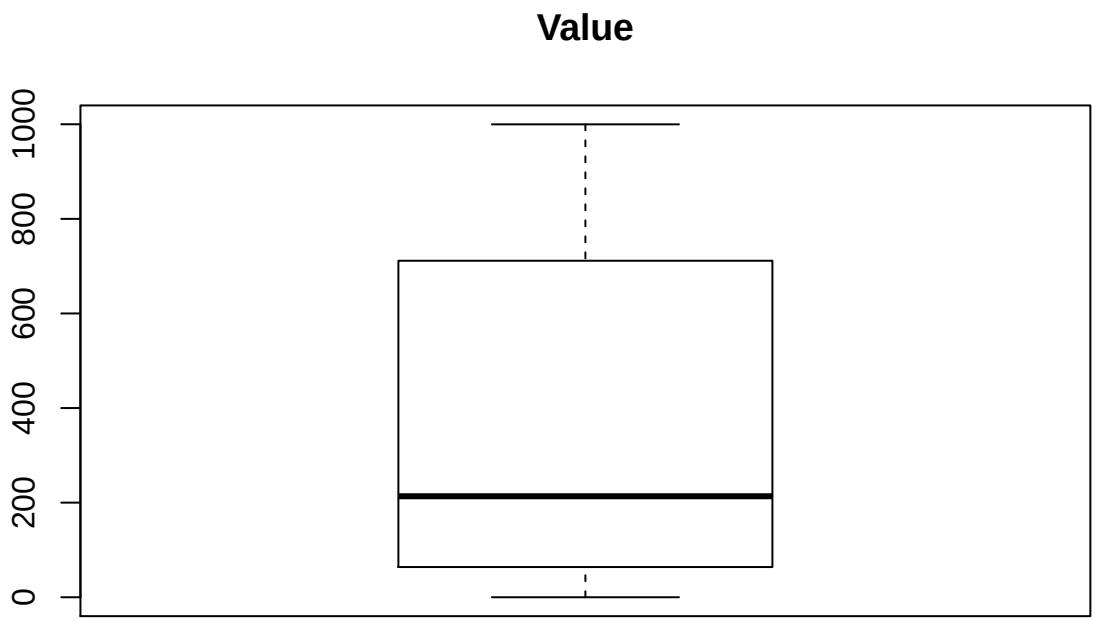
```
g = ggplot(gasto_f, aes(TIME, fill=is.na(Value)) ) +
labs(title = "Valores Nulos")+ylab("") +
theme(plot.title = element_text(size = rel(2), colour = "blue"))
```

```
g+geom_bar(position="dodge") + scale_fill_manual(values = alpha(c("orange", "blue"), 1)) +
theme(axis.title.x = element_text(face="bold", size=10))
```



- Con el siguiente gráfico, observaremos que la variable **Value** no tiene outliers o valores extremos

```
boxplot(gasto_f$Value, main="Value")
```



- Por otro lado, revisamos para el resto de columnas si tenemos valores NA.(desconocidos o perdidos)

```
table(gasto_f$TIME, useNA = "ifany")
```

```
##
## 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018
##  200  200  200  200  200  200  200  200  200  200
```

```
table(gasto_f$GEO, useNA = "ifany")
```

```
##
##                                    Austria
##                                         50
##                                    Belgium
##                                         50
##                     Bosnia and Herzegovina
##                                         50
##                                   Bulgaria
##                                         50
##                                    Croatia
##                                         50
##                                     Cyprus
##                                         50
##                                    Czechia
##                                         50
##                                    Denmark
##                                         50
##                                    Estonia
```

```
##                                             50
##             Euro area - 12 countries (2001-2006)
##                                             50
##                  Euro area - 18 countries (2014)
##                                             50
##            Euro area - 19 countries  (from 2015)
##                                             50
##        European Union - 15 countries (1995-2004)
##                                             50
##        European Union - 27 countries (2007-2013)
##                                             50
##        European Union - 27 countries (from 2020)
##                                             50
##        European Union - 28 countries (2013-2020)
##                                             50
##                                        Finland
##                                             50
##                                         France
##                                             50
## Germany (until 1990 former territory of the FRG)
##                                             50
##                                         Greece
##                                             50
##                                        Hungary
##                                             50
##                                        Iceland
##                                             50
##                                        Ireland
##                                             50
##                                          Italy
##                                             50
##                                         Latvia
##                                             50
##                                  Liechtenstein
##                                             50
##                                      Lithuania
##                                             50
##                                     Luxembourg
##                                             50
##                                          Malta
##                                             50
##                                    Netherlands
##                                             50
##                                         Norway
##                                             50
##                                         Poland
##                                             50
##                                       Portugal
##                                             50
##                                        Romania
##                                             50
##                                       Slovakia
##                                             50
##                                       Slovenia
```

```
##                                           50
##                                        Spain
##                                           50
##                                       Sweden
##                                           50
##                                  Switzerland
##                                           50
##                               United Kingdom
##                                           50
```

```
table(gasto_f$UNIT, useNA = "ifany")
```

```
##
## Million euro
##         2000
```

```
table(gasto_f$ICHA11_HF, useNA = "ifany")
```

```
##
##                                                                                All financing schemes
##                                                                                                  400
## Compulsory contributory health insurance schemes and compulsory medical saving accounts (CMSA)
##                                                                                                  400
##                                                                                   Government schemes
##                                                                                                  400
##                     Government schemes and compulsory contributory health care financing schemes
##                                                                                                  400
##                                                                          Social health insurance schemes
##                                                                                                  400
```

Observamos que no existen ahora valores perdidos después de la imputación.La suma de las cantidades de cada variable, suman el total.

- Finalmente, creamos un fichero con toda la información corregida.

```
write.csv(gasto_f, file="GastoSanitario_Financiacion_clean.csv", row.names = FALSE)
```