A1.Gasto Sanitario por Función

Alicia Perdices Guerra

8 de abril, 2021

Contents

1.PROCESAMIENTO DE LOS DATOS.

- En primer lugar leemos el fichero:

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

- Realicemos una breve inspección de los datos:

```
str(gasto_fun)
```

```
## '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_HC         : Factor w/ 5 levels "Curative care",..: 3 2 1 4 5 3 2 1 4 5 ...
##  $ Value             : Factor w/ 1378 levels ":","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_fun) #Nombre de las variables
```

```
## [1] "TIME"               "GEO"                "UNIT"
## [4] "ICHA11_HC"          "Value"              "Flag.and.Footnotes"
```

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

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

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

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

- Eliminamos la columna de Fal.and.footnotes.

```
gasto_fun<-gasto_fun[,-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_fun$Value<-as.character(gasto_fun$Value)
gasto_fun$Value<-as.numeric(gsub(",",".",gasto_fun$Value))
```

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

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

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

```
##
##  70.51   70.7  79.23  79.41  79.85  79.94  82.68  84.22  88.96   89.4  90.39
##      1      1      1      1      1      1      1      1      1      1      1
##  94.64 140.24 154.02 157.24 166.27 174.11  175.6 175.95 177.38 177.86 178.51
##      1      1      1      1      1      1      1      1      1      1      1
```

```
## 179.89 193.03 197.21 198.57 199.95 202.63 207.39 207.86 218.87 220.38 221.62
##      1      1      1      1      1      1      1      1      1      1      1
## 225.86 229.26 232.95 243.25 245.83  247.4 247.97 248.56 249.15 252.35 252.81
##      1      1      1      1      1      1      1      1      1      1      1
## 253.54 257.19 264.88  269.7 270.16 273.78 275.47 276.68 277.25 279.79 282.84
##      1      1      1      1      1      1      1      1      1      1      1
## 283.02 284.15 291.18  291.9 293.54 296.03 296.62 299.19 300.49 303.97 310.45
##      1      1      1      1      1      1      1      1      1      1      1
## 314.57 316.95 318.69  320.5 321.83  324.9  325.4 325.68 325.71 327.85 329.92
##      1      1      1      1      1      1      1      1      1      1      1
## 331.96 339.92  349.6 349.79    350 352.27 354.85 355.29 358.94 361.48 362.46
##      1      1      1      1      1      1      1      1      1      1      1
## 365.72  369.1 373.64 374.19 374.82 376.51 378.69 379.28 381.93 385.01 386.34
##      1      1      1      1      1      1      1      1      1      1      1
##    391 391.02  391.3 394.71 397.73 404.01 406.87  410.5 414.01 415.46 417.94
##      1      1      1      1      1      1      1      1      1      1      1
## 418.24 426.21 437.44 437.93 442.28 443.94 444.94 453.06 457.09 467.36 473.58
##      1      1      1      1      1      1      1      1      1      1      1
## 477.44 481.37 486.81  498.3 498.88 499.18 499.97 508.86 509.67 511.05 514.61
##      1      1      1      1      1      1      1      1      1      1      1
## 519.45 523.52 525.08 527.56 527.85 529.87 533.71 534.32 537.15 544.25 545.12
##      1      1      1      1      1      1      1      1      1      1      1
## 549.25 553.46 560.61 561.55 562.57 574.87 582.07 582.72 589.71 590.42 603.61
##      1      1      1      1      1      1      1      1      1      1      1
## 610.66 617.54 618.56 619.19 620.63 624.19 629.61 632.53 633.38 639.95 640.26
##      1      1      1      1      1      1      1      1      1      1      1
## 640.73  641.1 641.76 650.58 652.62 652.78 654.68 656.53  664.5 667.33 670.13
##      1      1      1      1      1      1      1      1      1      1      1
## 670.75 676.01 676.13  679.1 684.05 684.13 685.76 686.47 689.73 692.84  694.3
##      1      1      1      1      1      1      1      1      1      1      1
## 694.41 696.15 699.66 700.57 702.53 705.34 709.59 710.18  710.3 722.33 724.82
##      1      1      1      1      1      1      1      1      1      1      1
## 726.24 726.79 731.69 739.41 739.48 740.22 740.63 740.68 741.36 745.06 745.96
##      1      1      1      1      1      1      1      1      1      1      1
## 746.56 748.44 749.78 751.96 752.06 761.67 762.91 763.08 764.97 767.16  770.7
##      1      1      1      1      1      1      1      1      1      1      1
## 774.98 779.03 781.05 782.45 783.79 786.15  793.1 795.04 798.23 801.67 801.69
##      1      1      1      1      1      1      1      1      1      1      1
## 803.76 804.74 808.76 820.06 820.81  821.3  829.3 835.76 845.49 846.73 851.57
##      1      1      1      1      1      1      1      1      1      1      1
## 854.08 857.73 869.75  879.4 883.38 887.01 889.47 898.48 899.49 901.45  907.6
##      1      1      1      1      1      1      1      1      1      1      1
## 909.69 910.51  916.8 925.55 931.47 931.66  932.1 932.22 939.05 945.12  955.4
##      1      1      1      1      1      1      1      1      1      1      1
## 959.59 960.47 964.45 968.36 970.49 978.25 991.84 995.38 999.27   <NA>
##      1      1      1      1      1      1      1      1      1   1727
```

- Observamos que tenemos **1727 valores perdidos**.Guardamos en la variable **idx** los índices de los registros con valores **NA** de la variable **Value**.

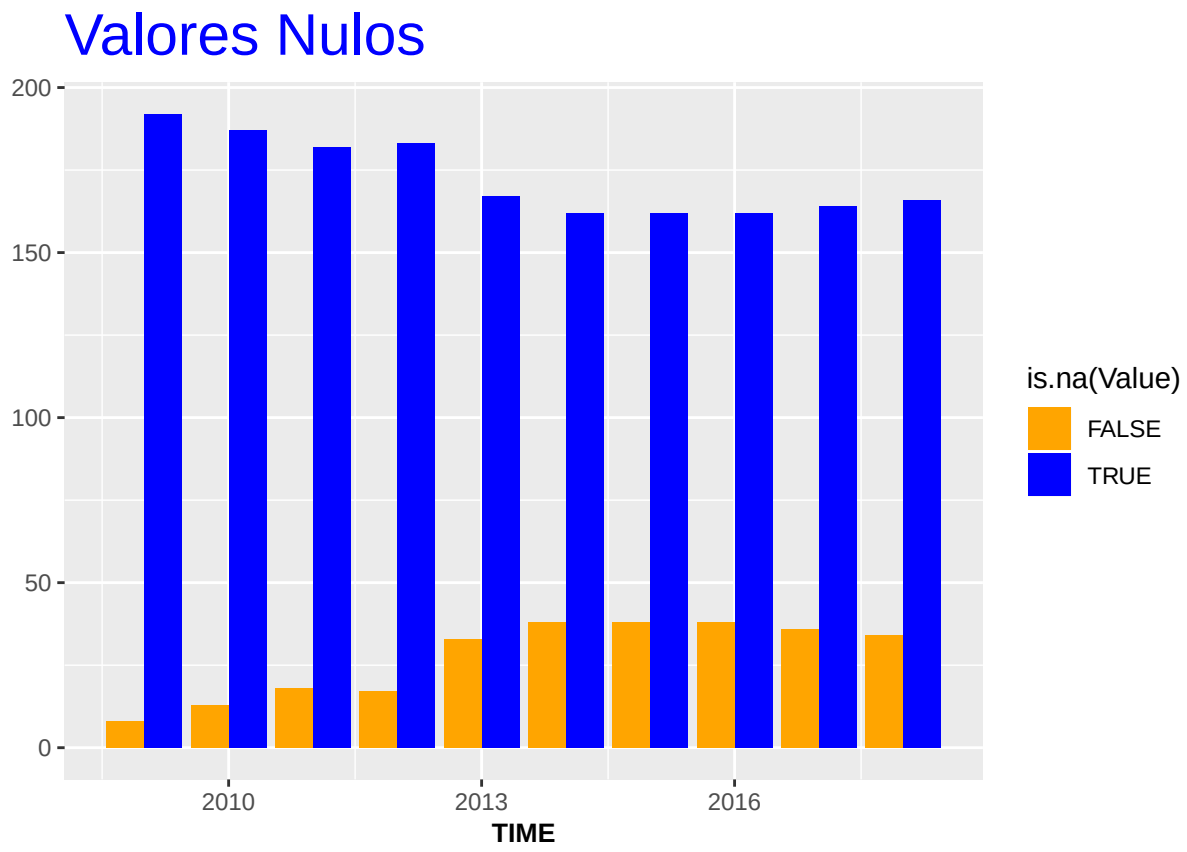
```
idx<-which(is.na(gasto_fun$Value))
length(idx)
```

```
## [1] 1727
```

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

```r
library(ggplot2)
library(scales)
g = ggplot(gasto_fun, 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.

```r
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':
##
```

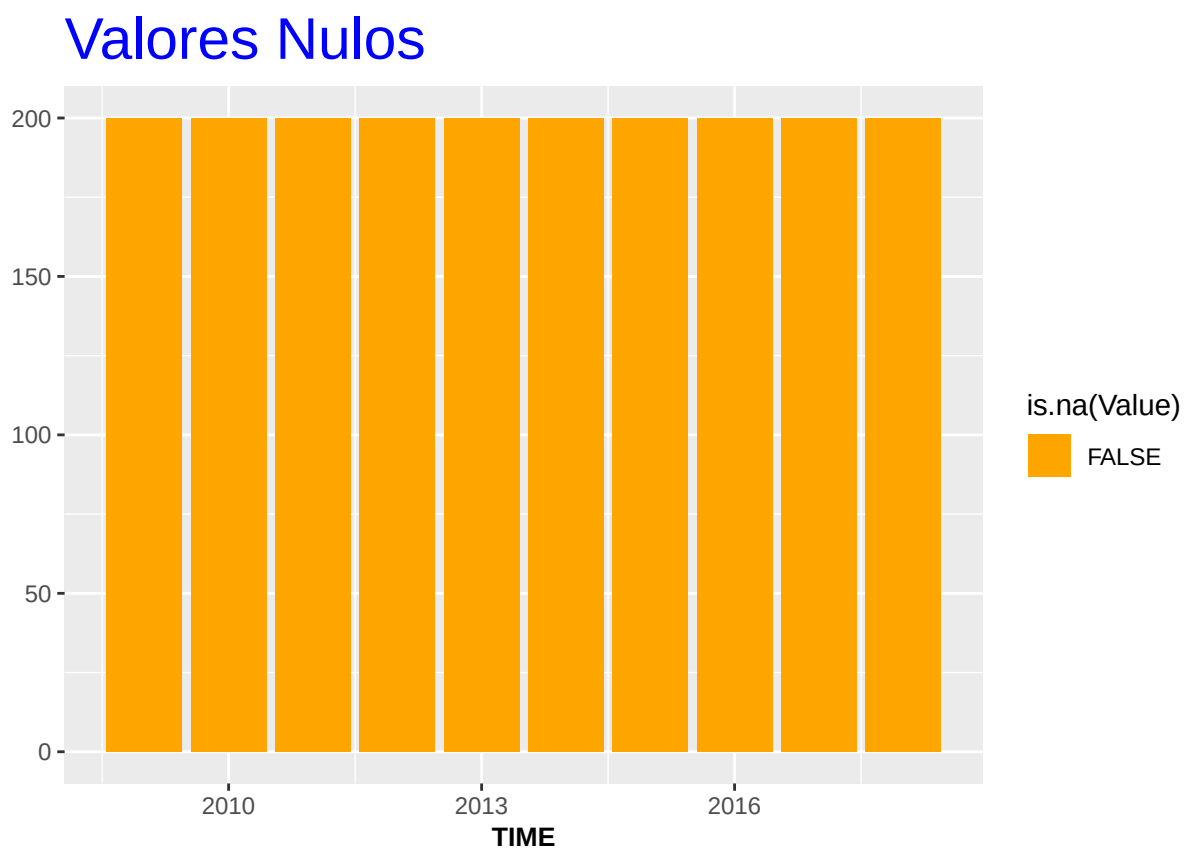
```
##     sleep
```

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

- Comprobamos que no tenemos valores nulos después de la imputación

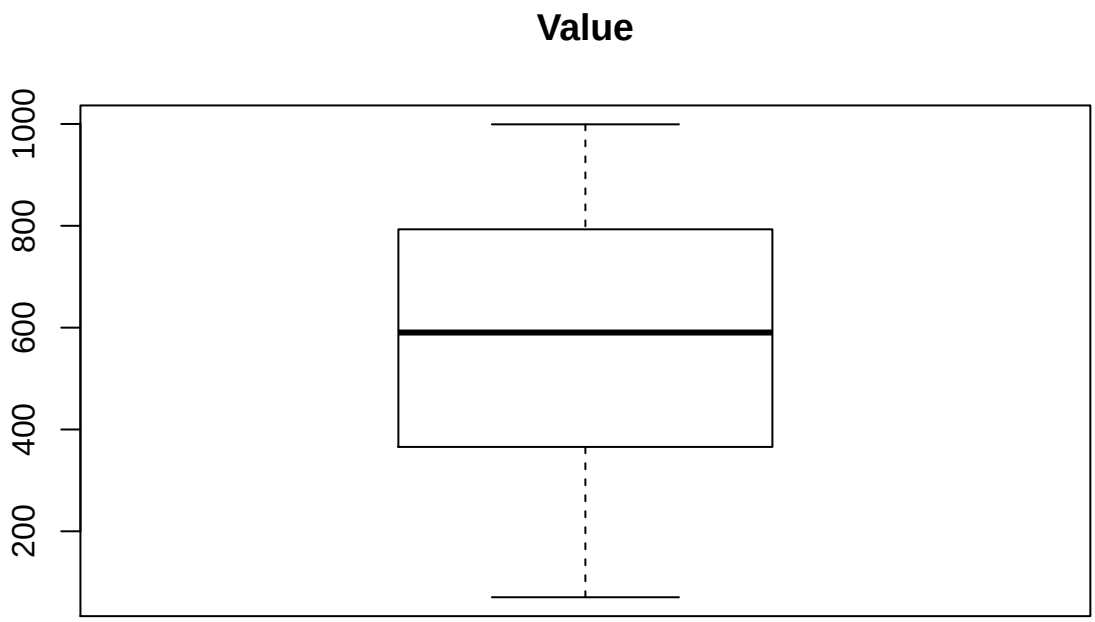
```
g = ggplot(gasto_fun, aes(TIME, fill=is.na(Value)) ) +
labs(title = "Valores Nulos")+ylab("") +
theme(plot.title = element_text(size = rel(2), colour = "blue"))

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_fun$Value, main="Value")
```



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

```
table(gasto_fun$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_fun$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_fun$UNIT, useNA = "ifany")
```

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

```
table(gasto_fun$ICHA11_HC, useNA = "ifany")
```

```
##
##                               Curative care
##                                         400
##      Curative care and rehabilitative care
##                                         400
##      Current health care expenditure (CHE)
##                                         400
## Inpatient curative and rehabilitative care
##                                         400
##                     Inpatient curative care
##                                         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_fun, file="GastoSanitario_Funcion_clean.csv", row.names = FALSE)
```