A1.Ingresos Sanitarios por Paises

Alicia Perdices Guerra

12 de abril, 2021

Contents

1.PROCESAMIENTO DE LOS DATOS.

- En primer lugar leemos el fichero:

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

- Realicemos una breve inspección de los datos

```
str(ingreso)
```

```
## 'data.frame':    220 obs. of  6 variables:
##  $ TIME              : int  2009 2009 2009 2009 2009 2009 2009 2009 2009 2009 ...
##  $ GEO               : Factor w/ 22 levels "Belgium","Croatia",..: 1 4 5 8 6 11 19 2 3 12 ...
##  $ UNIT              : Factor w/ 1 level "Million euro": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ICHA11_FS         : Factor w/ 1 level "All revenues of financing schemes": 1 1 1 1 1 1 1 1 1 1 ..
##  $ Value             : Factor w/ 159 levels ":","1,042.18",..: 101 1 1 76 147 1 153 1 1 1 ...
##  $ Flag.and.Footnotes: Factor w/ 2 levels "","b": 1 1 1 1 1 1 1 1 1 1 ...
```

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

```
## [1] "TIME"               "GEO"                "UNIT"
## [4] "ICHA11_FS"          "Value"              "Flag.and.Footnotes"
```

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

```
## [1] 220
```

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

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

*Observamos las siguientes variables:

- **TIME**: variable cuantitativa. Indica el año en el que se ha realizado la medida, en este caso el valor de la variable "Value". Se ha cargado bien como número entero.
- **GEO**: variable cualitativa. Indica el país o región en el que se ha realizado la medida.Se ha cargado bien como factor.
- **UNIT**: variable cualitativa. Indica la medida de la variable valor.Se ha cargado bien como factor.
- **ICHA11_FS**: variable cualitativa. Indica que la variable "Value" corresponde a todo tipo de ingresos por paises.
- **Value**: Variable cuantitativa. Indica el valor en Millones de Euros de esta financiación.Se ha cargado mal como factor. Haremos la transformación a valor numérico.
- **Fal.and.footnotes**. Notas sobre etiquetas. Eliminamos esta columna.

*Años de las mediciones:

```
unique(ingreso$TIME)
```

```
##  [1] 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018
```

*Paises:

```r
unique(ingreso$GEO)
```

```
##  [1] Belgium
##  [2] Czechia
##  [3] Denmark
##  [4] Germany (until 1990 former territory of the FRG)
##  [5] Estonia
##  [6] Ireland
##  [7] Spain
##  [8] Croatia
##  [9] Cyprus
## [10] Latvia
## [11] Lithuania
## [12] Luxembourg
## [13] Hungary
## [14] Malta
## [15] Poland
## [16] Slovenia
## [17] Finland
## [18] Sweden
## [19] Iceland
## [20] Norway
## [21] Switzerland
## [22] United Kingdom
## 22 Levels: Belgium Croatia Cyprus Czechia Denmark Estonia ... United Kingdom
```

*Unidad de las mediciones:

```r
unique(ingreso$UNIT)
```

```
## [1] Million euro
## Levels: Million euro
```

*Variable que indica que la variable value corresonde a todo tipo de ingresos por paises.

```r
unique(ingreso$ICHA11_FS)
```

```
## [1] All revenues of financing schemes
## Levels: All revenues of financing schemes
```

- Eliminamos la columna Fal.and.footnotes.

```r
ingreso<-ingreso[,-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.

```r
ingreso$Value<-as.character(ingreso$Value )
ingreso$Value<-(gsub(',','.',ingreso$Value) )
ingreso$Value<-substr(ingreso$Value,1,nchar(ingreso$Value)-3)
ingreso$Value<-as.numeric(ingreso$Value)
```

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

```r
table(ingreso$Value, useNA = "ifany")
```

```
##
##   1.042   1.045   1.108   1.109   1.137   1.211   1.227   1.234   1.249   1.265
##       1       1       1       1       1       1       1       1       1       1
```

```
##   1.274   1.277   1.289   1.318    1.35    1.41    1.43   1.522   1.572   1.609
##       2       1       1       1       1       1       1       1       1       1
##   1.734   1.804    1.81   1.862   2.265   2.423   2.463    2.57   2.581   2.638
##       1       1       1       1       1       1       1       1       1       1
##   2.708   2.732   2.751    2.85   2.907   2.972   2.987   3.027   3.174   3.183
##       1       1       1       1       1       1       1       1       1       1
##   3.199   3.309   3.327   3.428    3.52   3.524   3.797   6.832   7.396   7.428
##       1       1       1       1       1       1       1       1       1       1
##   7.431   7.488   7.642    7.73   8.123   8.531   8.963  15.871   16.65    17.2
##       1       1       1       1       1       1       1       1       1       1
##  18.261  18.505   18.85  19.231  19.271  20.034  20.143  20.236  20.388  20.398
##       1       1       1       1       1       1       1       1       1       1
##  20.653  21.116  21.259  22.451  25.126  25.166  25.167  25.681  26.072  26.313
##       1       1       1       1       1       1       1       1       1       1
##  27.032   27.28  27.603  27.756  27.921   28.72  29.597  30.449  30.663  31.202
##       1       1       1       1       1       1       1       1       1       1
##  31.501   35.22  35.318  35.879  36.447  36.971  37.162  39.071  40.574  41.494
##       1       1       1       1       1       1       1       1       1       1
##  42.073  43.024  43.449  44.235  45.327  46.166  46.406  47.417  48.043  48.178
##       1       1       1       1       1       1       1       1       1       1
##   49.18  50.545  51.296  51.775  52.119  55.183  56.143  58.808  69.655  70.902
##       1       1       1       1       1       1       1       1       1       1
##  71.046   71.64  92.518  93.824  94.417  97.384  97.532  97.815   98.35  99.715
##       1       1       1       1       1       1       1       1       1       1
## 103.899 108.109 209.392 229.998 232.178 240.259   242.3 261.567 274.841 284.568
##       1       1       1       1       1       1       1       1       1       1
## 290.266 297.784  309.02 322.481 338.267 352.045 369.091 383.636     795     889
##       1       1       1       1       1       1       1       1       1       1
##     898     925     932     939     945     970     991    <NA>
##       1       1       1       1       1       1       1      62
```

- Observamos que tenemos **62 valores perdidos**.Guardamos en la variable **idx** los índices de los registros con valores **NA** de la variable **Value**.

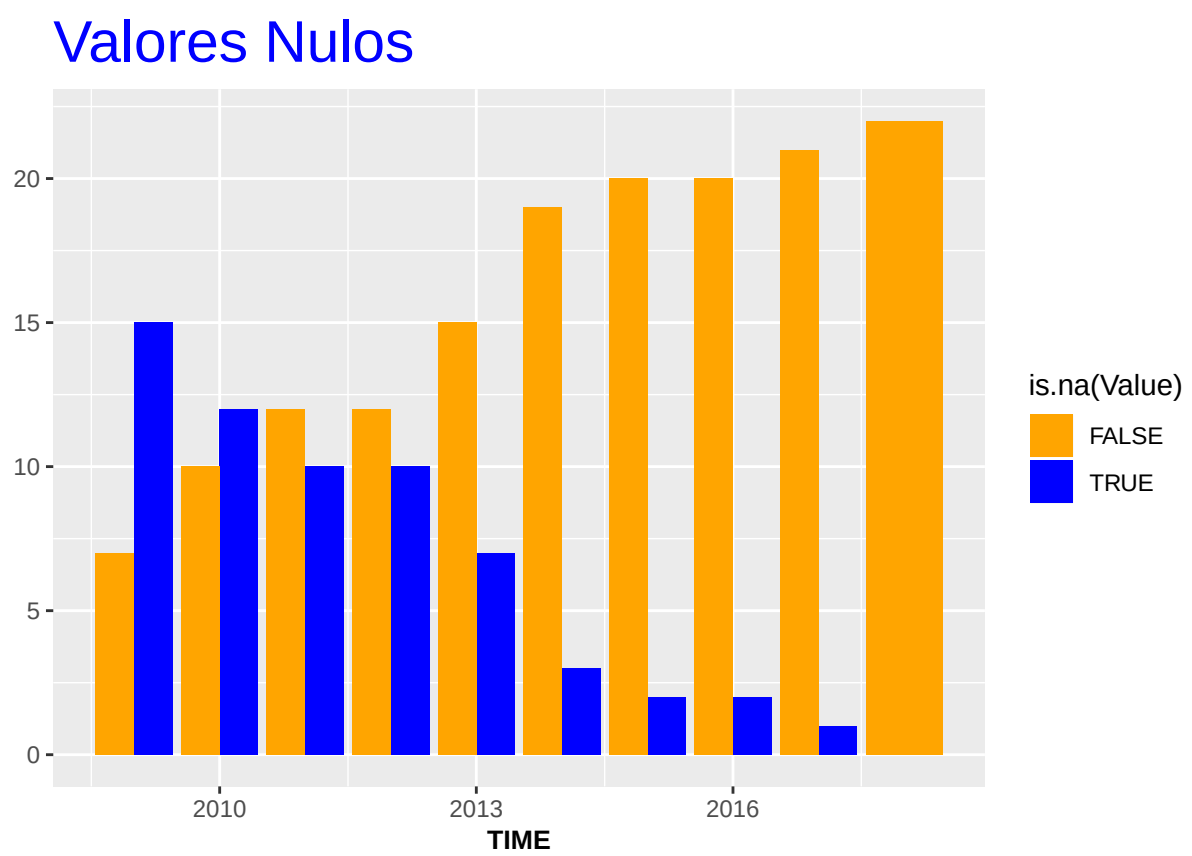
```
idx<-which(is.na(ingreso$Value))
length(idx)
```

```
## [1] 62
```

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

```
library(ggplot2)
library(scales)
g = ggplot(ingreso, 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)
```

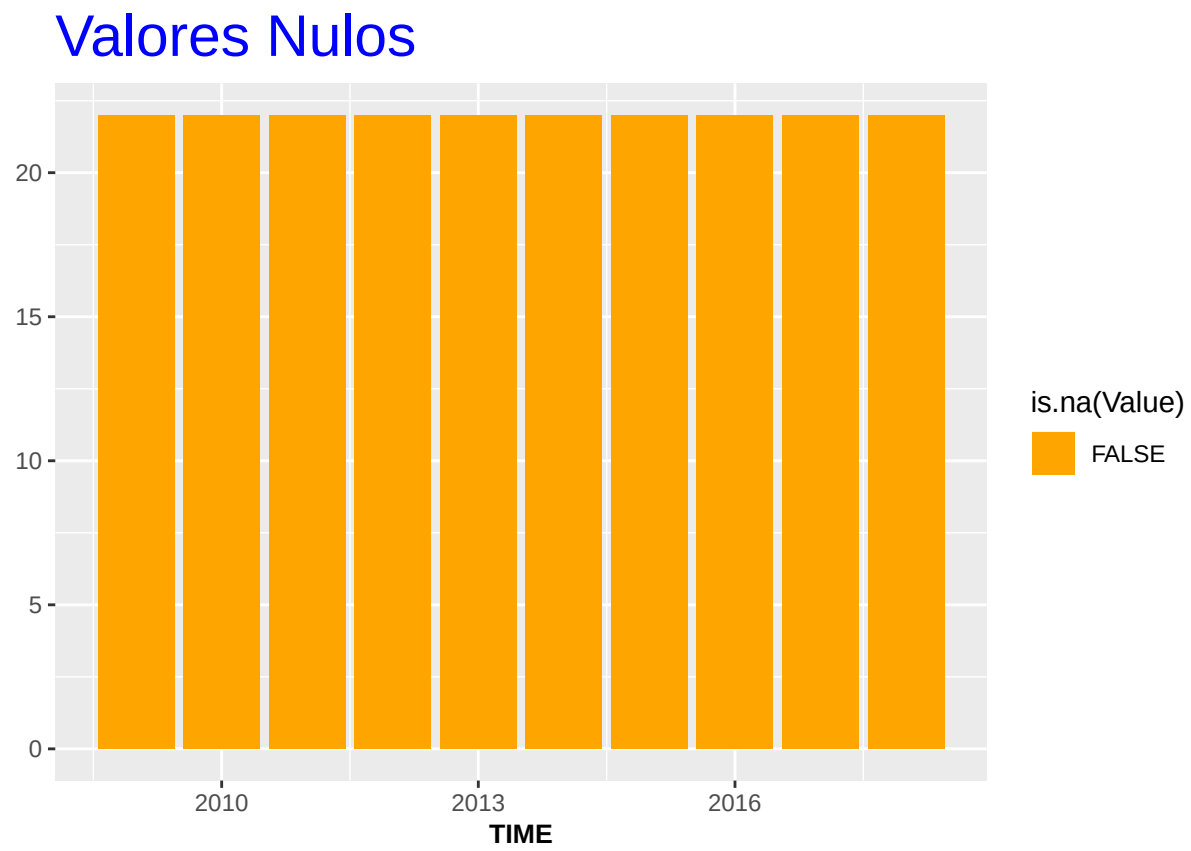
```
## Loading required package: colorspace
## Loading required package: grid
## VIM is ready to use.
## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues
##
## Attaching package: 'VIM'
## The following object is masked from 'package:datasets':
##
##     sleep
```

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

- Comprobamos que no tenemos valores nulos después de la imputación

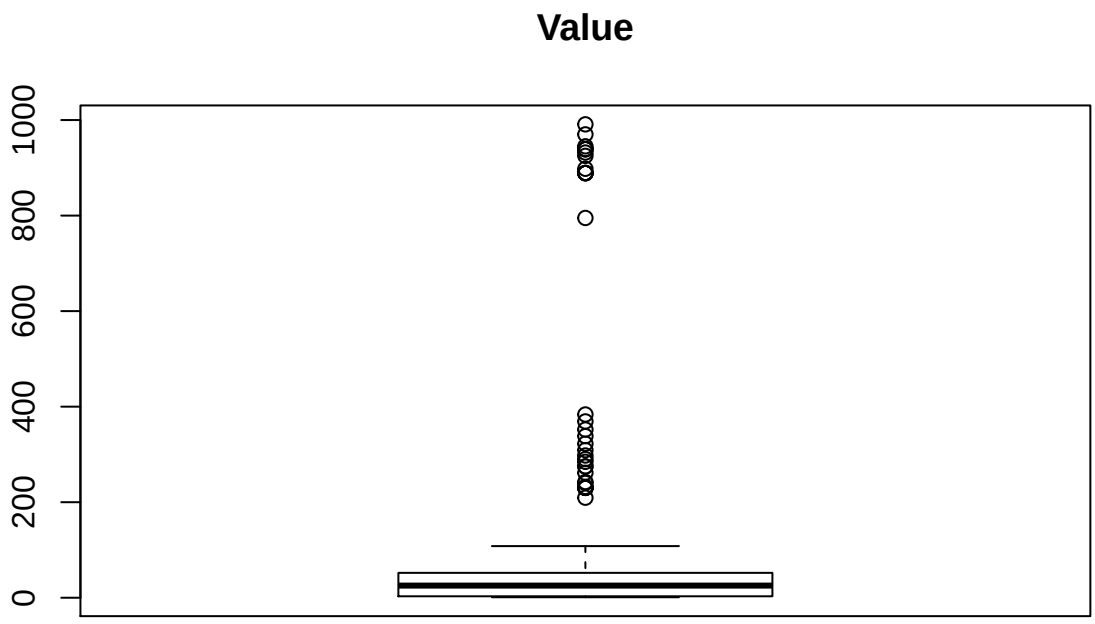
```
g = ggplot(ingreso, aes(TIME, fill=is.na(Value)) ) +
labs(title = "Valores Nulos")+ylab("") +
theme(plot.title = element_text(size = rel(2), colour = "blue"))
```

```
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** tiene outliers o valores extremos

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



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

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

```
##
## 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018
##   22   22   22   22   22   22   22   22   22   22
```

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

```
##
##                                          Belgium
##                                               10
##                                          Croatia
##                                               10
##                                           Cyprus
##                                               10
##                                          Czechia
##                                               10
##                                          Denmark
##                                               10
##                                          Estonia
##                                               10
##                                          Finland
##                                               10
## Germany (until 1990 former territory of the FRG)
##                                               10
##                                          Hungary
```

```
##                                         10
##                                    Iceland
##                                         10
##                                    Ireland
##                                         10
##                                     Latvia
##                                         10
##                                  Lithuania
##                                         10
##                                 Luxembourg
##                                         10
##                                      Malta
##                                         10
##                                     Norway
##                                         10
##                                     Poland
##                                         10
##                                   Slovenia
##                                         10
##                                      Spain
##                                         10
##                                     Sweden
##                                         10
##                                Switzerland
##                                         10
##                             United Kingdom
##                                         10
```

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

```
##
## Million euro
##          220
```

```
table(ingreso$ICHA11_FS, useNA = "ifany")
```

```
##
## All revenues of financing schemes
##                               220
```

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(ingreso, file="IngresosSanitarios_Financiacion_clean.csv", row.names = FALSE)
```