Pacientes en Diálisis y Trasplantados

Alicia Perdices Guerra

3 de mayo, 2021

Contents

1.PROCESAMIENTO DE LOS DATOS.

- En primer lugar leemos el fichero:

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

- Realicemos una breve inspección de los datos

```
str(dt)
```

```
## 'data.frame':    1740 obs. of  6 variables:
##  $ TIME              : int  2005 2005 2005 2005 2005 2005 2005 2005 2005 2005 ...
##  $ GEO               : Factor w/ 29 levels "Austria","Belgium",..: 2 2 2 2 2 2 3 3 3 3 ...
##  $ UNIT              : Factor w/ 2 levels "Number","Per hundred thousand inhabitants": 1 1 1 2 2 2 1
##  $ ICD9CM            : Factor w/ 3 levels "Haemodialysis",..: 3 1 2 3 1 2 3 1 2 3 ...
##  $ Value             : Factor w/ 923 levels ":","0","0.00",..: 43 665 432 920 658 419 1 1 1 1 ...
##  $ Flag.and.Footnotes: Factor w/ 5 levels "","b","be","d",..: 1 1 1 1 1 1 1 1 1 1 ...
```

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

```
## [1] "TIME"              "GEO"               "UNIT"
## [4] "ICD9CM"            "Value"             "Flag.and.Footnotes"
```

```
nrow(dt) #Número de registros
```

```
## [1] 1740
```

```
ncol(dt) #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. Total y ratio
- **ICD9CM**: Variable cualitativa. Hace referencia a si el paciente está recibiendo diálisis o está trasplantado.
- **Value**: Variable cuantitativa. Indica el número y ratio de pacientes en diálisis y trasplantados.
- **Fal.and.footnotes**. Notas sobre etiquetas. Eliminamos esta columna.

*Años de las mediciones:

```
unique(dt$TIME)
```

```
##  [1] 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014
```

*Paises:

```r
unique(dt$GEO)
```

```
##  [1] Belgium
##  [2] Bulgaria
##  [3] Czechia
##  [4] Denmark
##  [5] Germany (until 1990 former territory of the FRG)
##  [6] Estonia
##  [7] Ireland
##  [8] Greece
##  [9] Spain
## [10] France
## [11] Croatia
## [12] Italy
## [13] Lithuania
## [14] Luxembourg
## [15] Hungary
## [16] Malta
## [17] Netherlands
## [18] Austria
## [19] Poland
## [20] Portugal
## [21] Romania
## [22] Slovakia
## [23] Finland
## [24] Iceland
## [25] Liechtenstein
## [26] Switzerland
## [27] United Kingdom
## [28] Serbia
## [29] Turkey
## 29 Levels: Austria Belgium Bulgaria Croatia Czechia Denmark Estonia ... United Kingdom
```

*Unidad de las mediciones:

```r
unique(dt$UNIT)
```

```
## [1] Number                            Per hundred thousand inhabitants
## Levels: Number Per hundred thousand inhabitants
```

- En relación a si el paciente está en diálisis o trasplantado.

```r
unique(dt$ICD9CM)
```

```
## [1] Transplantation of kidney and haemodialysis
## [2] Haemodialysis
## [3] Transplantation of kidney
## 3 Levels: Haemodialysis ... Transplantation of kidney and haemodialysis
```

- Eliminamos la columna Fal.and.footnotes ya que no nos aporta información relevante.

```r
dt<-dt[,-6]
```

- Tendríamos que resolver las posibles inconsistencias en relación al formato del valor numérico de la variable **Value** y convertirla a valor numérico.

```r
dt$Value<-as.character(dt$Value)
dt$Value<- (gsub(',','.',dt$Value) )
```

```
dt$Value<- (gsub(' ','',dt$Value) )
dt$Value<- as.numeric(dt$Value)
```

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

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

```
tail(table(dt$Value, useNA = "ifany"))
```

```
##
## 67270 70322 73491 87151 91718  <NA>
##     1     1     1     1     1   759
```

- Observamos que tenemos **759 valores perdidos**.Guardamos en la variable **idx** los índices de los registros con valores **NA** de la variable **Value**.

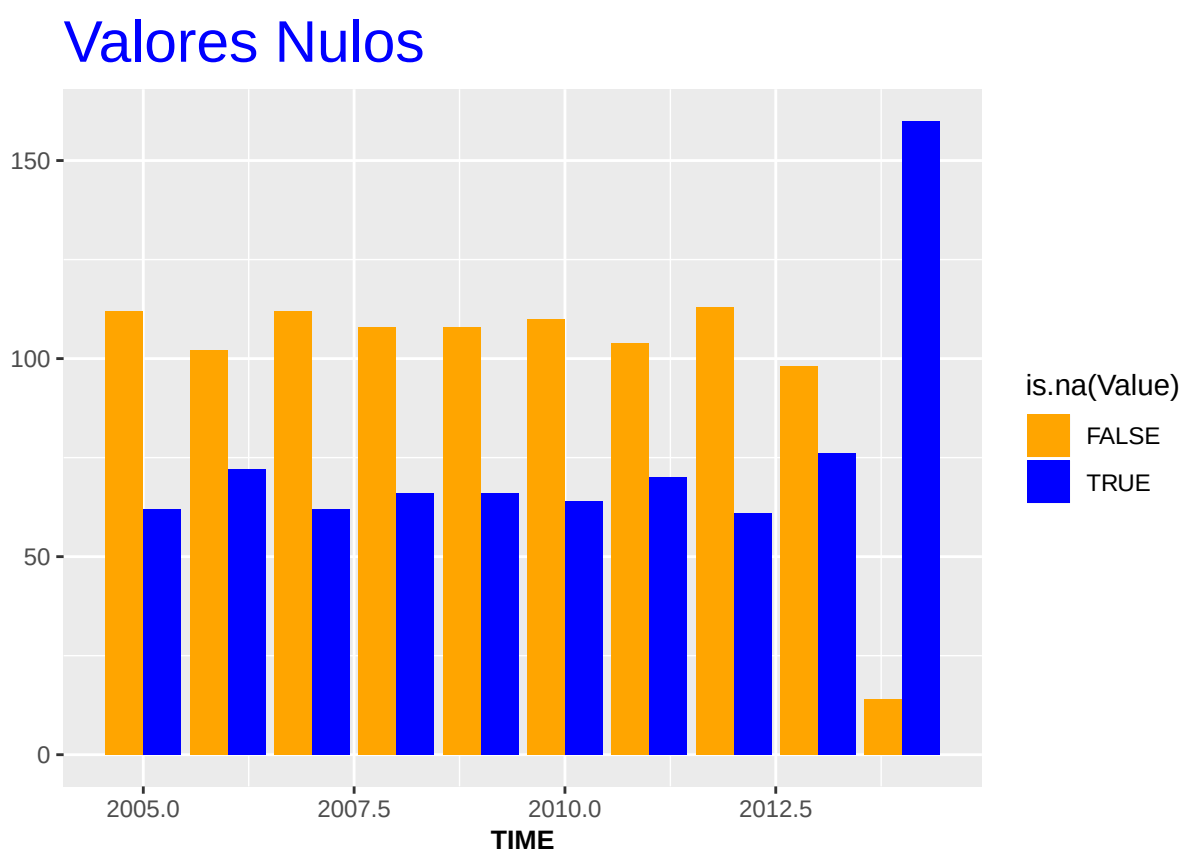
```
idx<-which(is.na(dt$Value))
length(idx)
```

```
## [1] 759
```

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

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

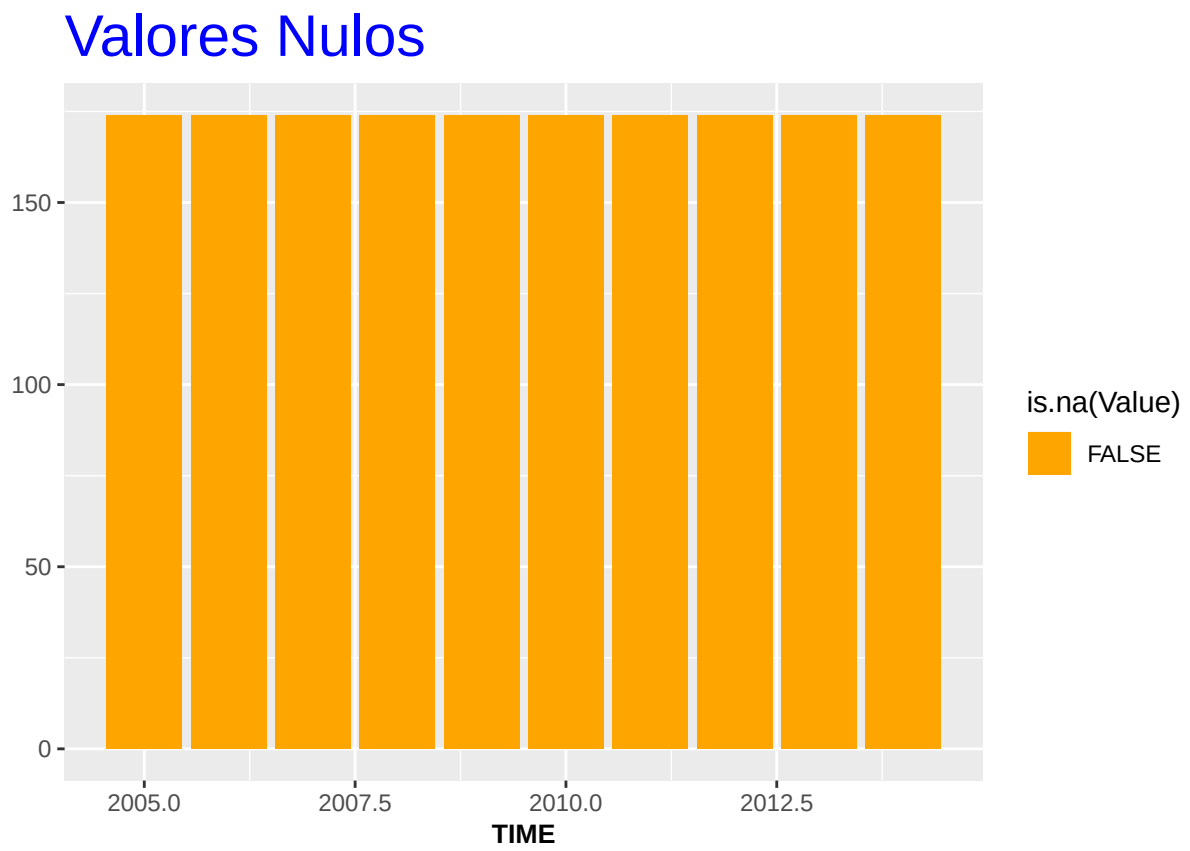
```
## Loading required package: colorspace
## Loading required package: grid
## VIM is ready to use.
## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues
##
## Attaching package: 'VIM'
## The following object is masked from 'package:datasets':
##
##     sleep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

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

- Comprobamos que no tenemos valores nulos después de la imputación

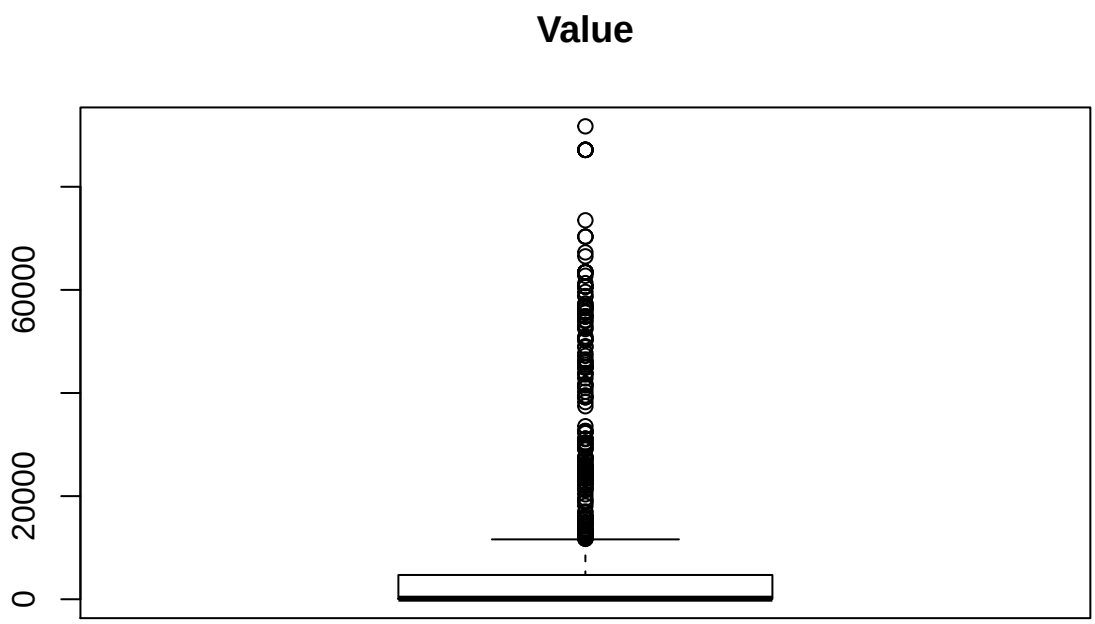
```
g = ggplot(dt, aes(TIME, fill=is.na(Value)) ) +
labs(title = "Valores Nulos")+ylab("") +
theme(plot.title = element_text(size = rel(2), colour = "blue"))
```

```
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(dt$Value, main="Value")
```



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

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

```
##
## 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014
##  174  174  174  174  174  174  174  174  174  174
```

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

```
##
##                                        Austria
##                                             60
##                                        Belgium
##                                             60
##                                       Bulgaria
##                                             60
##                                        Croatia
##                                             60
##                                        Czechia
##                                             60
##                                        Denmark
##                                             60
##                                        Estonia
##                                             60
##                                        Finland
##                                             60
##                                         France
```

```
##                                             60
## Germany (until 1990 former territory of the FRG)
##                                             60
##                                         Greece
##                                             60
##                                        Hungary
##                                             60
##                                        Iceland
##                                             60
##                                        Ireland
##                                             60
##                                          Italy
##                                             60
##                                  Liechtenstein
##                                             60
##                                      Lithuania
##                                             60
##                                     Luxembourg
##                                             60
##                                          Malta
##                                             60
##                                    Netherlands
##                                             60
##                                         Poland
##                                             60
##                                       Portugal
##                                             60
##                                        Romania
##                                             60
##                                         Serbia
##                                             60
##                                       Slovakia
##                                             60
##                                          Spain
##                                             60
##                                    Switzerland
##                                             60
##                                         Turkey
##                                             60
##                                 United Kingdom
##                                             60
```

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

```
##
##                           Number Per hundred thousand inhabitants
##                              870                               870
```

```
table(dt$ICD9CM, useNA = "ifany")
```

```
##
##                               Haemodialysis
##                                         580
##                  Transplantation of kidney
##                                         580
```

```
## Transplantation of kidney and haemodialysis
##                                         580
```

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

La estructura de los datos quedaría:

```
str(dt)
```

```
## 'data.frame':    1740 obs. of  6 variables:
##  $ TIME     : int  2005 2005 2005 2005 2005 2005 2005 2005 2005 2005 ...
##  $ GEO      : Factor w/ 29 levels "Austria","Belgium",..: 2 2 2 2 2 2 3 3 3 3 ...
##  $ UNIT     : Factor w/ 2 levels "Number","Per hundred thousand inhabitants": 1 1 1 2 2 2 1 1 1 2 ..
##  $ ICD9CM   : Factor w/ 3 levels "Haemodialysis",..: 3 1 2 3 1 2 3 1 2 3 ...
##  $ Value    : num  10371 6192 4179 99 59.1 ...
##  $ Value_imp: logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
```

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

```
write.csv(dt, file="Pacientes_Dialisis_Trasplantes_clean.csv", row.names = FALSE)
```