A1.Gasto Sanitario por Proveedor

Alicia Perdices Guerra

12 de abril, 2021

Contents

1.PROCESAMIENTO DE LOS DATOS.

- En primer lugar leemos el fichero:

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

- Realicemos una breve inspección de los datos:

```
str(gasto_pro)
```

```
## '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_HP         : Factor w/ 5 levels "All providers of health care",..: 1 3 2 4 5 1 3 2 4 5 ...
##  $ Value             : Factor w/ 1259 levels ":","0.00","1,001,514.67",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Flag.and.Footnotes: Factor w/ 3 levels "","b","d": 1 1 1 1 1 1 1 1 1 1 ...
```

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

```
## [1] "TIME"               "GEO"                "UNIT"
## [4] "ICHA11_HP"          "Value"              "Flag.and.Footnotes"
```

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

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

```
ncol(gasto_pro) #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_HP**: variable cualitativa. Entidad a la que se destina el gasto sanitario
- **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(gasto_pro$TIME)
```

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

*Paises:

```r
unique(gasto_pro$GEO)
```

```
##  [1] European Union - 27 countries (from 2020)
##  [2] European Union - 28 countries (2013-2020)
##  [3] European Union - 27 countries (2007-2013)
##  [4] European Union - 15 countries (1995-2004)
##  [5] Euro area - 19 countries  (from 2015)
##  [6] Euro area - 18 countries (2014)
##  [7] Euro area - 12 countries (2001-2006)
##  [8] Belgium
##  [9] Bulgaria
## [10] Czechia
## [11] Denmark
## [12] Germany (until 1990 former territory of the FRG)
## [13] Estonia
## [14] Ireland
## [15] Greece
## [16] Spain
## [17] France
## [18] Croatia
## [19] Italy
## [20] Cyprus
## [21] Latvia
## [22] Lithuania
## [23] Luxembourg
## [24] Hungary
## [25] Malta
## [26] Netherlands
## [27] Austria
## [28] Poland
## [29] Portugal
## [30] Romania
## [31] Slovenia
## [32] Slovakia
## [33] Finland
## [34] Sweden
## [35] Iceland
## [36] Liechtenstein
## [37] Norway
## [38] Switzerland
## [39] United Kingdom
## [40] Bosnia and Herzegovina
## 40 Levels: Austria Belgium Bosnia and Herzegovina Bulgaria Croatia ... United Kingdom
```

*Unidad de las mediciones:

```r
unique(gasto_pro$UNIT)
```

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

*Variable que indica la entidad a la que se destina el gasto sanitario:

```r
unique(gasto_pro$ICHA11_HC)
```

```
## NULL
```

- Eliminamos la columna Fal.and.footnotes.

```
gasto_pro<-gasto_pro[,-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_pro$Value<-as.character(gasto_pro$Value)
gasto_pro$Value<-as.numeric(gsub(",",".",gasto_pro$Value))
```

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

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

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

```
##
##      0   1.01   1.04   2.44   2.61    2.8   2.86   2.97    3.2   3.32   3.54
##     26      1      1      1      1      1      1      1      1      1      1
##   3.56   3.64   3.67   3.71    3.8   3.92   4.31   4.36   4.51    5.3   5.46
##      1      1      2      1      1      1      1      1      1      1      1
##   5.57   5.78    5.9   6.23   6.35   6.49    6.5   6.55   6.74   6.83   7.25
##      1      1      1      1      1      1      1      1      1      1      1
##   7.48   7.85   8.98  10.28  10.39  10.65  11.29  11.51  11.67  11.92  12.02
##      1      1      1      1      1      1      1      1      1      1      1
##  12.21  12.58  13.16  13.42   14.5  14.64  14.68  15.18  15.23  15.45  15.64
##      1      1      1      1      1      1      1      1      1      1      1
##  15.84  15.89  17.56  20.14  20.84  21.01  21.78  22.02   22.1  22.27  22.65
##      1      1      1      1      1      1      1      1      1      1      1
##  22.86  22.98  23.26  23.49  23.64  23.65  24.15  24.77  25.03  25.38  25.49
##      1      1      1      1      1      1      2      1      2      1      1
##  25.57  25.93  26.13  26.59  26.81  26.85   26.9  27.44  27.85   27.9  28.14
##      1      1      1      1      1      1      1      1      2      2      1
##  28.25  28.53  28.66  29.69  29.75  29.79  29.95  29.96  30.29  30.95  31.07
##      1      2      1      1      1      1      1      1      1      1      1
##  31.25  31.53  32.46     33  33.15  33.29  33.58  34.31  34.76   35.8  36.47
##      1      1      1      1      1      1      1      1      1      1      1
##  36.48  37.39  37.43  37.89  38.24  38.36  40.98  41.34  41.37  42.09   43.3
##      1      1      1      1      1      1      1      1      1      1      1
##  43.57   43.9  45.74  47.46  49.69  50.15  51.01  51.38  51.71  51.79  52.07
##      1      1      1      1      1      1      1      1      1      1      1
##  52.11  52.19  52.61  53.39  53.51  54.55  55.14  55.26  56.22  56.43  57.08
##      1      1      1      1      1      1      1      1      1      1      1
##  57.42  59.79  59.84  61.45  61.49  62.82  66.22  66.39  66.65  67.34  67.49
##      1      1      1      1      1      1      1      1      1      1      1
##   67.8  67.96  71.18  71.24  72.11  73.58  74.94  77.17  77.65  78.27  78.54
##      1      1      1      1      1      1      1      1      1      1      1
##  78.55  79.63  80.27  82.13  82.76  83.47  84.47  86.05  88.56  88.99  90.68
##      1      1      1      1      1      1      1      1      1      1      1
##  91.16  92.58  96.01  96.88  96.92  97.05  99.13 104.78  105.2 109.95 110.37
##      1      1      1      1      1      1      1      1      1      1      1
## 116.26 116.88 117.58 118.99 120.17 122.78 124.66 126.57 128.45 133.73  141.5
##      1      1      1      1      1      1      1      1      1      1      1
## 143.95 146.83 154.65 155.89 160.95 173.28 176.87  183.5 193.96  205.3    208
##      1      1      1      1      1      1      1      1      1      1      1
## 227.03  227.2 228.21 229.39 229.95 232.81 238.84 244.51  246.8 249.81 253.54
##      1      1      1      1      1      1      1      1      1      1      1
```

```
## 257.28 258.99 264.26 264.32 265.16 267.18 272.77 273.15 276.68 278.47 282.27
##      1      1      1      1      1      1      1      1      1      1      1
## 283.02 286.35 286.53 287.89 292.73 294.31 295.54 297.32 298.16  298.3 310.91
##      1      1      1      1      1      1      1      1      1      1      1
## 320.34  320.5  324.9 325.15 325.71 326.25 328.21 328.44 329.92 331.86 331.99
##      1      1      1      1      1      1      1      1      1      1      1
## 332.03 332.36 333.01 333.75 338.87 339.47 340.77 342.76 344.05 344.77 350.02
##      2      1      1      1      1      1      1      1      1      1      1
## 350.97 351.73 352.66 355.55 360.71 364.18 364.78 367.47 372.05 373.66 376.49
##      1      2      1      1      1      1      1      1      1      1      3
## 378.53  384.9 385.21  387.5 391.59 395.11 396.93 398.87 405.28 407.76 408.48
##      1      1      1      1      1      1      1      1      1      1      1
##  409.4 416.38 418.67 420.03 420.23 420.61 420.95 423.01 424.78 427.48 427.51
##      1      1      1      1      1      1      1      2      1      1      1
## 433.26 434.22 436.46 439.97 440.97 441.95 442.33 443.65 444.73 452.04 454.49
##      1      1      1      1      1      1      1      1      1      1      1
## 454.94    457 458.39 459.91 461.49 463.49 465.17 465.46 471.01 473.66  475.1
##      1      1      1      1      1      1      1      1      1      1      1
##  475.8 477.12 481.84 484.43 489.11 489.73  490.9 492.36 494.23 497.15  499.5
##      1      1      1      1      2      1      1      1      1      1      1
##  499.7 500.74 505.05  509.3 509.61 511.84 512.02 512.63 516.65 520.36 522.11
##      1      1      1      1      1      1      1      1      1      1      1
## 524.43 529.04  532.4 537.39 540.41 546.19 556.35 557.22 559.01 561.19 562.58
##      1      1      1      1      1      1      1      1      1      1      1
## 572.99 576.23 577.07 579.82 580.36  585.6 587.37 592.22 593.13 598.05 605.99
##      1      1      1      1      1      1      2      1      1      1      1
## 609.26 617.73 622.76 624.48 636.29 638.76 641.92 656.43 658.97 668.87 679.42
##      1      1      1      1      1      1      1      1      1      1      1
## 692.88 693.26 698.46    704 704.26 705.11 706.57 708.45 709.28 716.39 722.94
##      1      1      1      1      1      1      1      2      1      1      1
## 724.79 727.21 728.24 737.65 739.41 741.17 742.86 745.19 751.72  752.1 757.26
##      1      1      1      2      1      1      1      1      1      1      1
## 757.31 759.25 759.42 759.54 762.85 764.54 765.16 766.83 774.08  774.2 786.48
##      1      1      1      1      1      1      1      1      1      1      1
## 788.18 795.04 800.23    804 808.73 810.01 813.84 824.81 831.93 834.18  835.3
##      1      1      1      1      1      1      1      1      1      1      1
##  835.9 849.25  851.1 852.35 854.65 858.79 864.47  873.9 887.08 889.47 890.15
##      1      1      1      1      1      1      1      1      1      1      1
##  890.2 898.48  901.8 908.02 922.15 925.55  932.1 938.06 938.09 939.05 945.12
##      1      1      1      1      1      1      1      1      1      1      1
## 948.67 948.77 949.63 961.38 966.14 969.18 970.49  975.4 981.36 991.84  999.5
##      1      1      1      1      1      1      1      1      1      1      1
##   <NA>
##   1509
```

- Observamos que tenemos **1509 valores perdidos**.Guardamos en la variable **idx** los índices de los registros con valores **NA** de la variable **Value**.

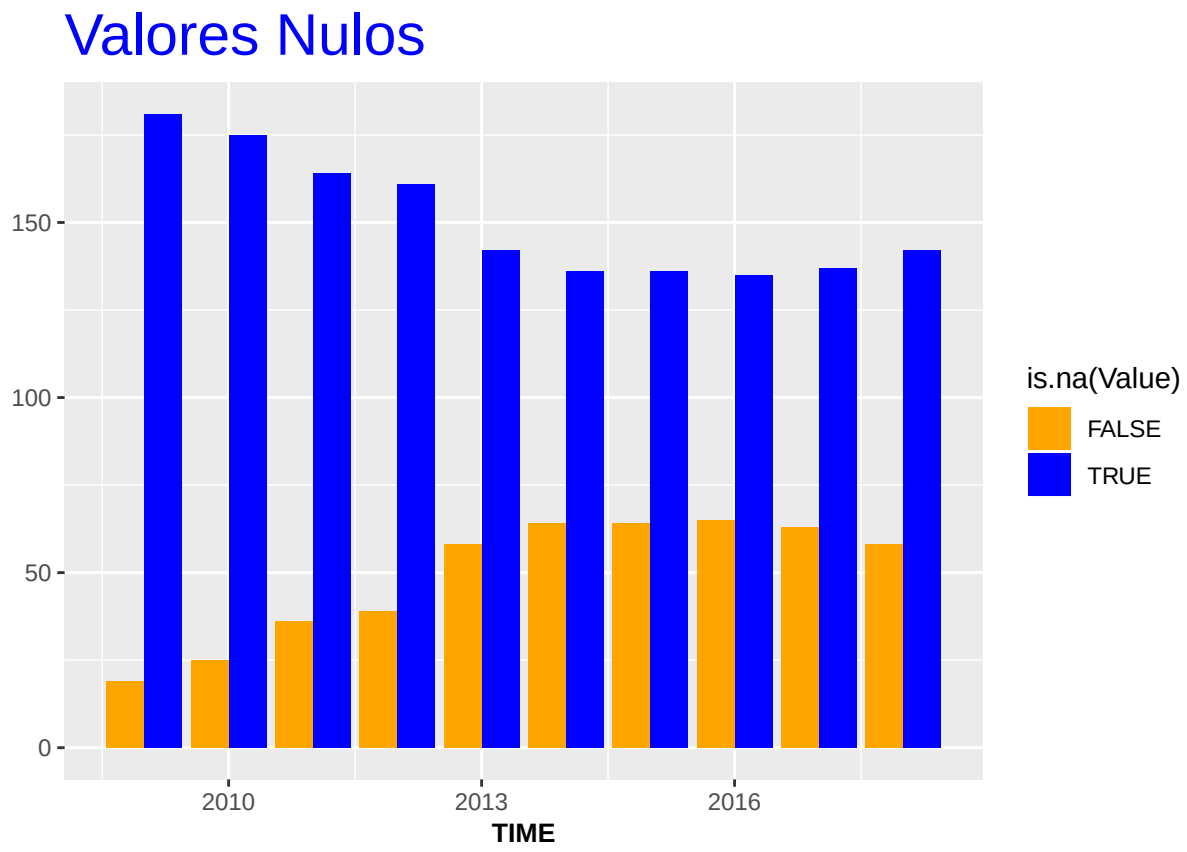
```
idx<-which(is.na(gasto_pro$Value))
length(idx)
```

```
## [1] 1509
```

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

```r
library(ggplot2)
library(scales)
g = ggplot(gasto_pro, 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':
##
```

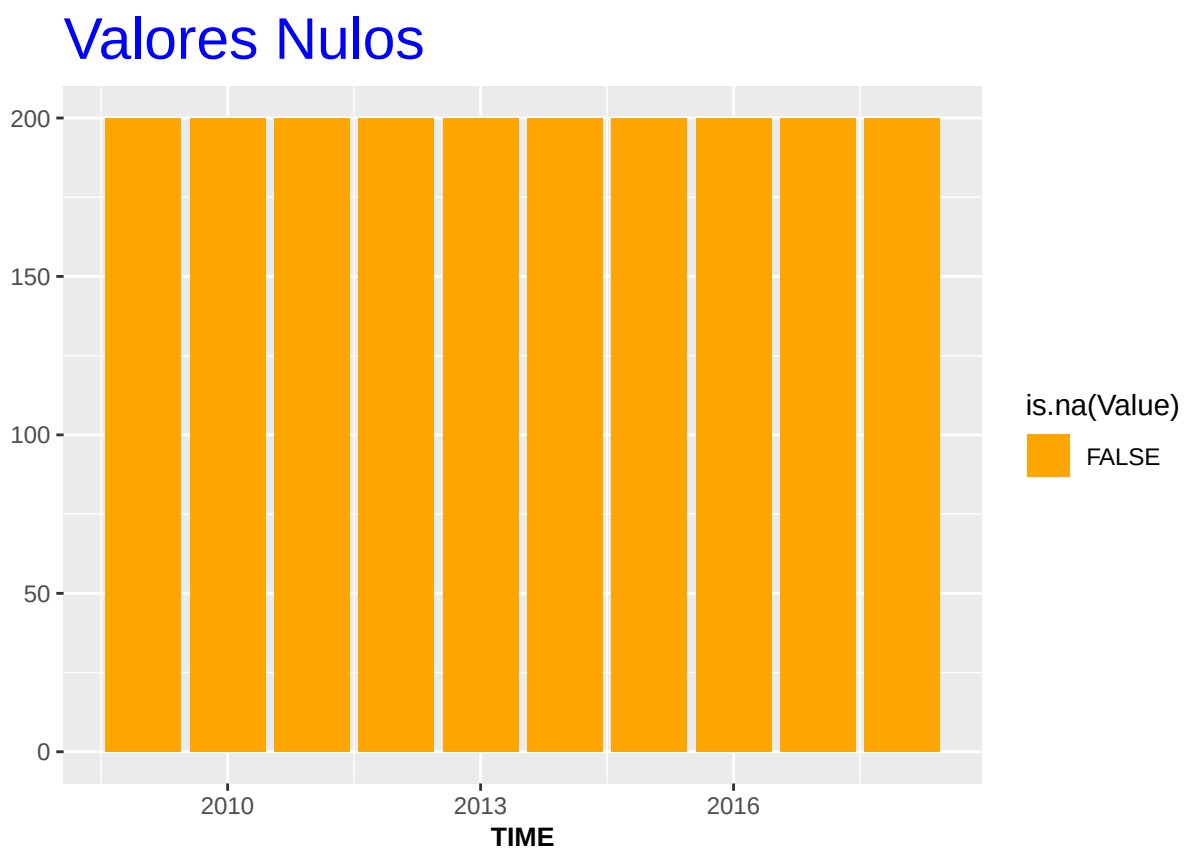
```
##     sleep
```

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

- Comprobamos que no tenemos valores nulos después de la imputación

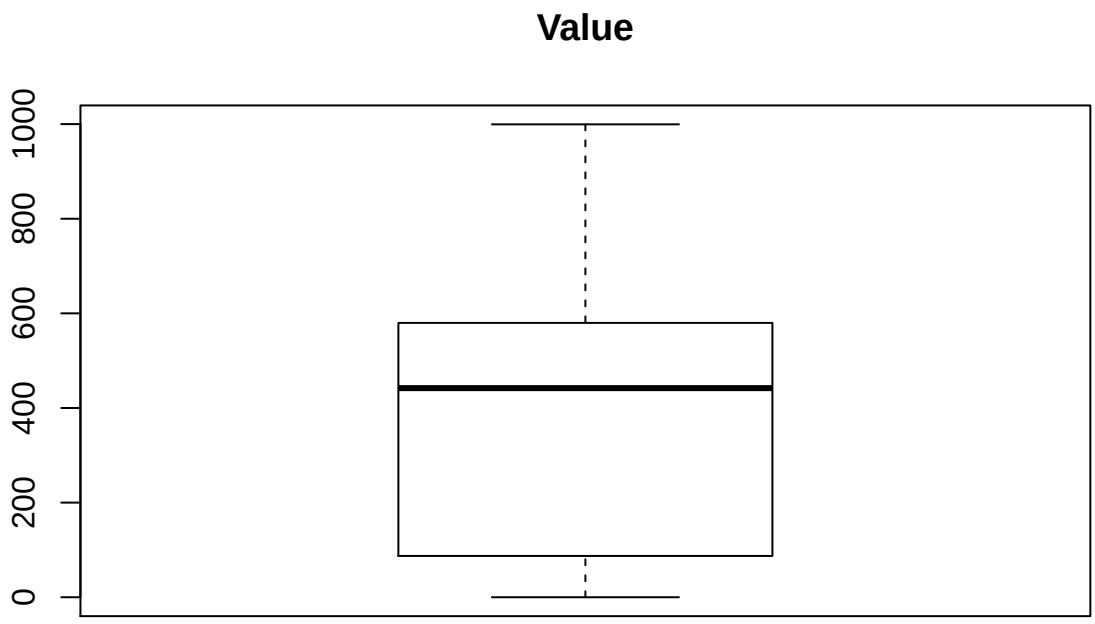
```
g = ggplot(gasto_pro, aes(TIME, fill=is.na(Value)) ) +
labs(title = "Valores Nulos")+ylab("") +
theme(plot.title = element_text(size = rel(2), colour = "blue"))

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



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

```
table(gasto_pro$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_pro$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_pro$UNIT, useNA = "ifany")
```

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

```
table(gasto_pro$ICHA11_HP, useNA = "ifany")
```

```
##
##                              All providers of health care
##                                                       400
##                                        General hospitals
##                                                       400
##                                                Hospitals
##                                                       400
##                                     Mental health hospitals
##                                                       400
## Specialised hospitals (other than mental health hospitals)
##                                                       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_pro, file="GastoSanitario_Proveedor_clean.csv", row.names = FALSE)
```