

Personal Sanitario Hospitalario

Alicia Perdices Guerra

3 de mayo, 2021

Contents

1.PROCESAMIENTO DE LOS DATOS.

- En primer lugar leemos el fichero:

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

- Realicemos una breve inspección de los datos

```
str(sanitarios_h)
```

```
## 'data.frame': 1800 obs. of 6 variables:
## $ TIME : int 2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
## $ GEO : Factor w/ 36 levels "Albania","Austria",...: 3 3 3 3 3 4 4 4 4 4 ...
## $ ISCO08 : Factor w/ 5 levels "Health care assistants",...: 3 5 4 1 2 3 5 4 1 2 ...
## $ UNIT : Factor w/ 1 level "Full-time equivalent (FTE)": 1 1 1 1 1 1 1 1 1 1 ...
## $ Value : Factor w/ 819 levels ":", "0.00", "1 003 000.00",...: 625 618 22 780 170 1 1 1 1 ...
## $ Flag.and.Footnotes: Factor w/ 4 levels "","b","be","e": 4 4 4 4 4 1 1 1 1 1 ...
```

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

```
## [1] "TIME" "GEO" "ISCO08"
## [4] "UNIT" "Value" "Flag.and.Footnotes"
```

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

```
## [1] 1800
```

```
ncol(sanitarios_h) #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.
- **WSTATUS**: variable cualitativa. Indica el estatus laboral del personal sanitario hospitalario.
- **ISCO08**: Variable cualitativa. Indica si la variable “Value” se refiere a Médicos, Enfermeras y Matronas, Enfermeras adjuntas, Asistentes de cuidados de salud, o Personal Hospitalario.
- **Value**: Variable cuantitativa. Indica el número de profesionales sanitarios hospitalarios por países.
- **Flag.and.footnotes**. Notas sobre etiquetas. Eliminamos esta columna.

*Años de las mediciones:

```
unique(sanitarios_h$TIME)
```

```
## [1] 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
```

*Países:

```
unique(sanitarios_h$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] Cyprus
## [14] Latvia
## [15] Lithuania
## [16] Luxembourg
## [17] Hungary
## [18] Malta
## [19] Netherlands
## [20] Austria
## [21] Poland
## [22] Portugal
## [23] Romania
## [24] Slovenia
## [25] Slovakia
## [26] Finland
## [27] Iceland
## [28] Liechtenstein
## [29] Norway
## [30] Switzerland
## [31] United Kingdom
## [32] Montenegro
## [33] North Macedonia
## [34] Albania
## [35] Serbia
## [36] Turkey
## 36 Levels: Albania Austria Belgium Bulgaria Croatia Cyprus Czechia ... United Kingdom
```

*Unidad de las mediciones:

```
unique(sanitarios_h$UNIT)
```

```
## [1] Full-time equivalent (FTE)
## Levels: Full-time equivalent (FTE)
```

*Variable que indica cómo agrupamos el personal de sanitario para las mediciones.

```
unique(sanitarios_h$ISCO08)
```

```
## [1] Medical doctors           Nursing professionals and midwives
## [3] Nursing associate professionals Health care assistants
## [5] Hospital employment
## 5 Levels: Health care assistants Hospital employment ... Nursing professionals and midwives
```

- Eliminamos la columna Fal.and.footnotes.

```
sanitarios_h<-sanitarios_h[,-7]
```

- 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.

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

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

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

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

```
##
##      0      10.46      10.51      11.7      12.1      12.4      13.15
##    99         1         1         1         1         1         1
##   15.2      15.6      18.6      19.1      19.2      22      22.52
##     1         1         1         1         1         1         1
##   22.6      23.2      23.25      23.7      25.4      25.7      28.79
##     1         1         1         1         1         1         1
##  30.01      36.47      38.89      39.6      42.85      46.9      48.96
##     1         1         1         1         1         1         1
##  49.73      52.2      53.75      57.3      60      60.2      81
##     1         1         1         1         1         1         1
##    93        95       102       119       134       139      141.47
##     1         1         1         1         1         1         1
## 144.31     147.12     151.05     153.8     167     169.05     170
##     1         1         1         1         1         1         1
##  171.1      196      197.8     203.92     490     540.82     541.54
##     1         1         1         1         1         1         1
##  541.86     544.79     546.32     546.42     554.9     555.83     557.96
##     1         1         1         1         1         1         1
##  560.26     561.4     564.69     565.37     581.43     602.86     616
##     1         1         1         1         1         1         1
##  616.81     630.68       634     647.21     667.22     685.24     692.83
##     1         1         1         1         1         1         1
##  695.59     741.28     741.36       747       762     773.8       787
##     1         1         1         1         1         1         1
##  788.02      790      796      801      811      842     886.65
##     1         1         1         1         1         1         1
##  908.04     934.87     1005.72     1049.01     1084.03     1102.78     1122
##     1         1         1         1         1         1         1
## 1155.1     1177.94     1217.85     1250.36     1277.35     1360.83     1378.08
##     1         1         1         1         1         1         1
## 1382.78     1401.43     1408.25     1432.5     1434.24     1468.98     1481.84
##     1         1         1         1         1         1         1
## 1496.55     1515.62     1515.83     1549.74     1566.84     1588.23     1640.4
##     1         1         1         1         1         1         1
##   2096     2221.07     2234.14     2304.38     2502     2531.9     2547.7
##     1         1         1         1         1         1         1
##  2556.5     2567.1     2580.8     2652.47     2660.9     2663.2     2679.5
##     1         1         1         1         1         1         1
```

##	2727.5	2741.2	2766.9	2773.33	2774.47	2790.7	2791.48
##	1	1	1	1	1	1	1
##	2813.2	2824.92	2847.01	2852.8	2859.09	2908.31	2908.37
##	1	1	1	1	1	1	1
##	2909.7	2941.09	2958.69	3055.3	3110.9	3157.08	3237
##	1	1	1	1	1	1	1
##	3262	3293.64	3318	3363	3430.58	3448	3630
##	1	1	1	1	1	1	1
##	3663	3711	3718	3773	3985	4023	4141
##	1	1	1	1	1	1	1
##	4158	4217	4228	4385	4569	4615.07	4629.52
##	1	1	1	1	1	1	1
##	4673.12	4720.56	4723.43	4811.35	4824.4	4858.9	4871
##	1	1	1	1	1	1	1
##	4871.12	4890.9	4901.3	4902.1	4963.1	5038	5046.73
##	1	1	1	1	1	1	1
##	5070	5087	5153.58	5212	5279	5343.04	5348
##	1	1	1	1	1	1	1
##	5406	5424	5433	5476.86	5490.63	5504	5530.59
##	1	1	1	1	1	1	1
##	5555	5607	5697	5704	5710	5725.36	5759.22
##	1	1	1	1	1	1	1
##	5794	5801	5855.4	5878	5884	5919.73	5930.1
##	1	1	1	1	1	1	1
##	5943	5969.7	6072	6083.74	6142.5	6167	6212
##	1	1	1	1	1	1	1
##	6269	6274	6293.97	6302	6308.2	6321.21	6359
##	1	1	1	1	1	1	1
##	6381.84	6447.43	6504.65	6522.08	6535.19	6627.94	6661
##	1	1	1	1	1	1	1
##	6697.27	6700.02	6736.48	6743.55	6774	6806.88	6903.42
##	1	1	1	1	1	1	1
##	6920	7006.63	7022.41	7028.78	7055	7064.67	7087.5
##	1	1	1	1	1	1	1
##	7146.81	7181	7214.87	7220	7273.15	7366.72	7394.95
##	1	1	1	1	1	1	1
##	7444.63	7448.83	7485.26	7490.92	7491.89	7580	7677
##	1	1	1	1	1	1	1
##	7698.74	7751.04	7752	7766.25	7824	7830	7855.51
##	1	1	1	1	1	1	1
##	7909	7921	7937	8055	8058.85	8108	8169.54
##	1	1	1	1	1	1	1
##	8206	8219	8270.81	8278.26	8356	8356.17	8420.45
##	1	1	1	1	1	1	1
##	8469.13	8527.49	8532.88	8547	8554.44	8621.02	8628.31
##	1	1	1	1	1	1	1
##	8640.66	8647	8672.96	8748.94	8760.16	8763.41	8840.38
##	1	1	1	1	1	1	1
##	8883	8914.7	8936.08	8959	8969.75	9020.17	9027.29
##	1	1	1	1	1	1	1
##	9027.73	9058	9083.46	9086.69	9229.93	9282.42	9309.16
##	1	1	1	1	1	1	1
##	9369.48	9455.2	9523.36	9558	9573	9582	9726
##	1	1	1	1	1	1	1

##	9747	9753	9761	9762	9772	9818	10007
##	1	1	1	1	1	1	1
##	10067	10112	10191.45	10196	10213	10546	10734.23
##	1	1	1	1	1	1	1
##	10737	10756.23	10824	10901.35	10919.81	10921.61	10933.74
##	1	1	1	1	1	1	1
##	10980	11010	11048	11070.45	11101	11216	11256
##	1	1	1	1	1	1	1
##	11268	11293.28	11321	11324	11345	11379	11419
##	1	1	1	1	1	1	1
##	11473	11483	11563	11631	11778	11794	11909
##	1	1	1	1	1	1	1
##	11975	11994	12360	12747.72	12756.6	12790.2	12806.64
##	1	1	1	1	1	1	1
##	12854.4	12875.9	12886	12985.4	13055.2	13208	13539.81
##	1	1	1	1	1	1	1
##	13543	13667	13882.05	13922.49	13931.29	13988.31	14049
##	1	1	1	1	1	1	1
##	14102.33	14114.23	14335	14437	14490	14796	15120
##	1	1	1	1	1	1	1
##	15346.41	15385	15465	15478	15632	15674	15675
##	1	1	1	1	1	1	1
##	15676.92	15770	15962	15971	16021	16091	16106
##	1	1	1	1	1	1	1
##	16588.5	16614	16970.61	17094	17142.19	17209.44	17320.63
##	1	1	1	1	1	1	1
##	17342.23	17364.05	17406.65	17435.9	17582.3	18059.48	18256.1
##	1	1	1	1	1	1	1
##	18426.31	19026.13	19221	19369	19562	19596.72	19650
##	1	1	1	1	1	1	1
##	19741	19775.97	19820	19824	20057.43	20314	20377.24
##	1	1	1	1	1	1	1
##	20386.79	20503.96	20661.84	20707	20846.07	21104.31	21134.89
##	1	1	1	1	1	1	1
##	21154	21178.38	21352.19	21361	21381.23	21394.42	21453.48
##	1	1	1	1	1	1	1
##	21454.68	21491.75	21501	21514	21536	21605	21632.52
##	1	1	1	1	1	1	1
##	21682	21757	21761.62	22011.05	22043	22193	22285
##	1	1	1	1	1	1	1
##	22371.82	22376	22449.57	22470.96	22716.84	22731.09	22807.83
##	1	1	1	1	1	1	1
##	22972	23029	23110.78	23148	23216	23222	23271
##	1	1	1	1	1	1	1
##	23364	23422	23617	23671	23905	23920	25327
##	1	1	1	1	1	1	1
##	25536	25906	25992.19	26011	26107.91	26368	27211
##	1	1	1	1	1	1	1
##	27509	27667	27803.23	28116.62	28191.02	28465	28684.52
##	1	1	1	1	1	1	1
##	28701.12	28764.81	29000	29342	30000	31000	32000
##	1	1	1	1	1	1	2
##	33000	34000	35000	35523	35652	35996	36792
##	1	1	1	1	1	1	1

##	36802	37092	37127	37155	37172	37231	37243
##	1	1	1	1	1	1	1
##	37350	37637	37702	37711	37747	37894	38459
##	1	1	1	1	1	1	1
##	39065	39957	40181.17	40256.78	40723.64	41262.81	41345.26
##	1	1	1	1	1	1	1
##	41349	41442.12	41458.33	41474	41479.97	41795	42077
##	1	1	1	1	1	1	1
##	42287.43	42599	42900.62	43005.48	43337	43530.5	43644.56
##	1	1	1	1	1	1	1
##	43893	45343.32	45637	45742	45940	45988	46090
##	1	1	1	1	1	1	1
##	46144	46387.12	46819	46963	47396	47458.1	47935
##	1	1	1	1	1	1	1
##	48019.39	48068.77	48247.66	48289	48833.7	48908	49194
##	1	1	1	1	1	1	1
##	49318	49747	49764	49775	50307	50823	50875
##	1	1	1	1	1	1	1
##	50952	50998.08	51186	51203.43	51629.17	52293.24	52511.26
##	1	1	1	1	1	1	1
##	52542	52598	52690	52749.84	52864.88	52904.51	53207.71
##	1	1	1	1	1	1	1
##	53455.95	53562.02	53645.08	54123.78	54144	54679	54868
##	1	1	1	1	1	1	1
##	55103.74	56260	56655.33	56801	56899.5	57243.5	57423
##	1	1	1	1	1	1	1
##	58010	58474	58481.59	58536	59112	59479.45	60080
##	1	1	1	1	1	1	1
##	60708.79	60773	61278	61564	61609.04	61782.11	61820.93
##	1	1	1	1	1	1	1
##	62080	62181.95	62606	62822	62971.53	62978	63269
##	1	1	1	1	1	1	1
##	63656	63695.29	63839.09	63965	64155	64423	64940
##	1	1	1	1	1	1	1
##	65556.97	67049.02	67429.96	69212.21	69371.38	69897.31	70366.21
##	1	1	1	1	1	1	1
##	71135.73	71293.24	73213.39	76567.22	76719.99	79031.65	79365.7
##	1	1	1	1	1	1	1
##	82398.34	83760.13	91981	92626	92822	94745.35	94827.99
##	1	1	1	1	1	1	1
##	95973	96331	97281	98240	99187	99556.29	99873.85
##	1	1	1	1	1	1	1
##	99930.78	100064.07	100372.68	100705	100795.43	102178	102494.73
##	1	1	1	1	1	1	1
##	102578	103270	103805	105861	106246	106525	106529
##	1	1	1	1	1	1	1
##	106676	107078	110956.52	111848.94	114599.04	114770.7	115989.96
##	1	1	1	1	1	1	1
##	116291.96	118086.09	120607.07	121137.02	122743.29	124057.19	124555.65
##	1	1	1	1	1	1	1
##	126023.95	126619.32	129312.18	129366.47	131059.27	132737.08	133123.05
##	1	1	1	1	1	1	1
##	133988.71	134002.9	134130	134386.54	134949.51	135056.52	135061
##	1	1	1	1	1	1	1

```
## 135729.02      135829      136931 137558.57      138958 139251.93 139702.6
##          1          1          1          1          1          1          1
## 140507.19 140592.65 141311.37 141423.39 141704.22      143000 143458.63
##          1          1          1          1          1          1          1
## 144065.66 144741.52 144836.12 146152.98 146171.23 146475.33 146598.44
##          1          1          1          1          1          1          1
##      147000 147380.48 148362.81 149036.47 149044.33      150489      150695
##          1          1          1          1          1          1          1
##      151000      152126      152435.2      154315      155000      156818 157664.36
##          1          1          1          1          1          1          1
##      159000      160424 161945.41      163000      165463.1      166000      166297
##          1          1          1          1          1          1          1
## 168372.17      170000      174423.5      176297      198647      201280      202528
##          1          1          1          1          1          1          1
##      204480      204854      211903      215281      215396      216350 225408.54
##          1          1          1          1          1          1          1
## 225947.78 226711.78 227044.21 227119.01 227468.49 227829.65 228124.77
##          1          1          1          1          1          1          1
## 261687.47      312000      319000      322000      332000      335000 336299.53
##          1          1          1          1          1          1          1
##      337000      341000 345196.95 345798.06 348535.05 349585.59      350000
##          1          1          1          1          1          1          1
## 350218.46 350464.87 352461.05 354397.24 354504.78 356733.74 359840.92
##          1          1          1          1          1          1          1
## 360464.18 362951.19 364229.35 365055.56 365458.22 367609.62 374550.58
##          1          1          1          1          1          1          1
##      926000      934000      946000      959000      968000      978000      988000
##          1          1          1          1          1          1          1
##      1003000 1130915.69 1147732.18 1150444.67 1155583.2 1158639.44 1164679.41
##          1          1          1          1          1          1          1
## 1169711.72 1176945.99 1190503.45 1204434.87 1216111.85 1223996.34 1229041.62
##          1          1          1          1          1          1          1
## 1253394.85 1255606.48 1283240.71 1290047.12      1317664 1363598.98      <NA>
##          1          1          1          1          1          1          883
```

- Observamos que tenemos **883 valores perdidos**. Guardamos en la variable **idx** los índices de los registros con valores **NA** de la variable **Value**.

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

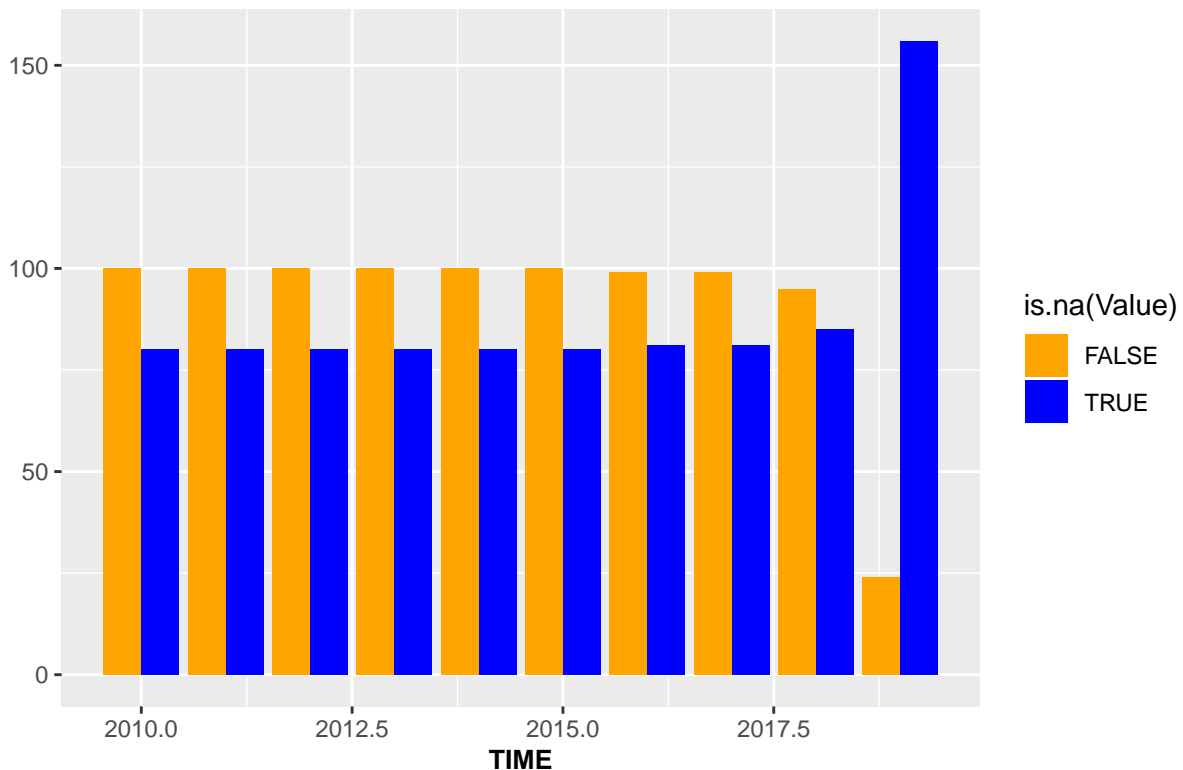
```
## [1] 883
```

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

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

Valores Nulos



- 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 robusta 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(sanitarios_h, variable=c("Value"),k=3)
```

```
sanitarios_h<-output
```

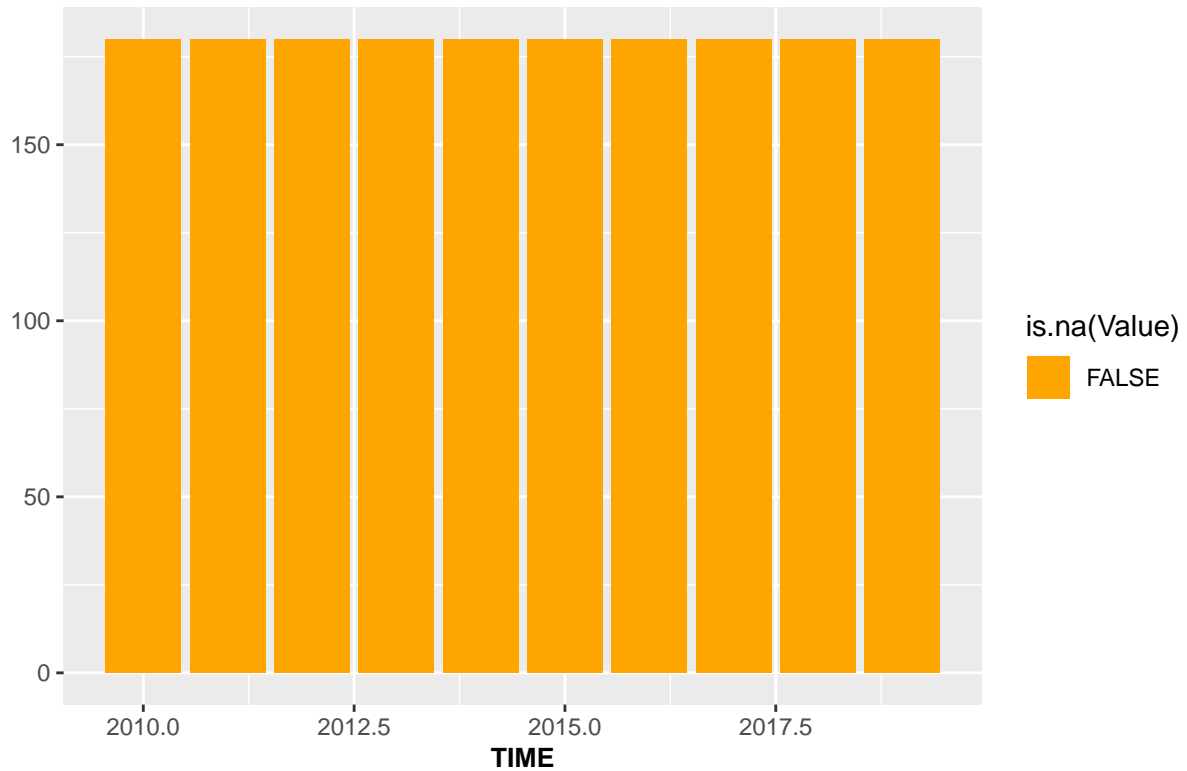
- Comprobamos que no tenemos valores nulos después de la imputación

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



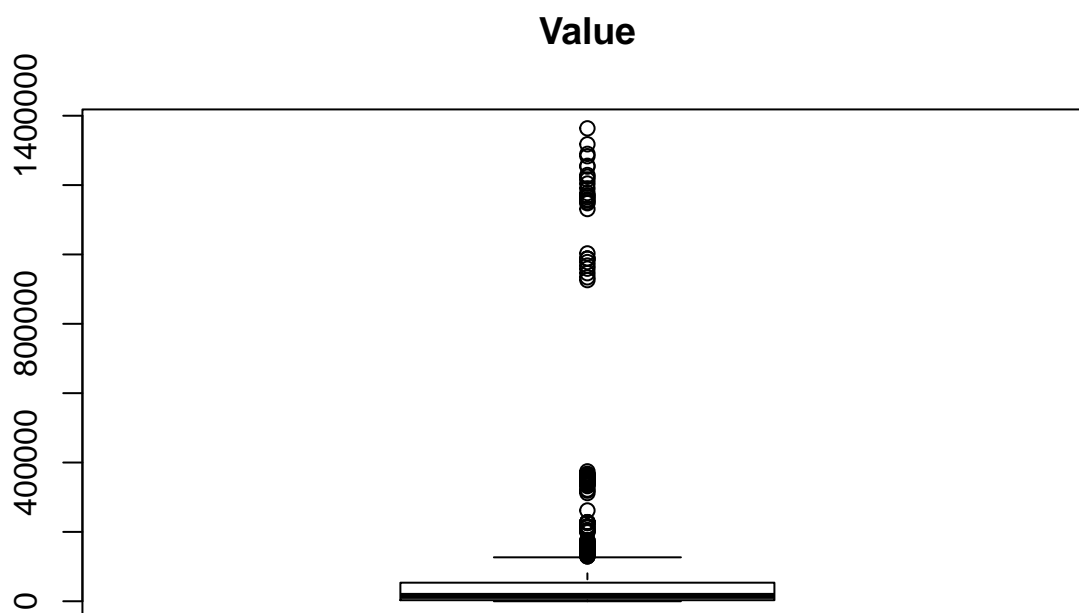
```
g+geom_bar(position="dodge") + scale_fill_manual(values = alpha(c("orange", "blue"), 1)) +  
theme(axis.title.x = element_text(face="bold", size=10))
```

Valores Nulos



- Con el siguiente gráfico, observaremos que la variable **Value** tiene outliers o valores extremos

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



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

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

```
##
## 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
## 180 180 180 180 180 180 180 180 180 180
```

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

```
##
##
## Albania
## 50
## Austria
## 50
## Belgium
## 50
## Bulgaria
## 50
## Croatia
## 50
## Cyprus
## 50
## Czechia
## 50
## Denmark
## 50
## Estonia
```

##	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
##	Montenegro
##	50
##	Netherlands
##	50
##	North Macedonia
##	50
##	Norway
##	50
##	Poland
##	50
##	Portugal
##	50
##	Romania
##	50
##	Serbia
##	50
##	Slovakia
##	50
##	Slovenia
##	50
##	Spain
##	50
##	Switzerland
##	50
##	Turkey
##	50
##	United Kingdom

```
##
```

50

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

```
##
```

```
## Full-time equivalent (FTE)
```

```
## 1800
```

```
table(sanitarios_h$ISCO08, useNA = "ifany")
```

```
##
```

```
## Health care assistants Hospital employment
```

```
## 360 360
```

```
## Medical doctors Nursing associate professionals
```

```
## 360 360
```

```
## Nursing professionals and midwives
```

```
## 360
```

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(sanitarios_h, file="PersonalSanitarioHospitalario_clean.csv", row.names = FALSE)
```