

# 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.
- **Flag.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
```

\*Países:

```
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:

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

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

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

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

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

- Comprobamos que valores tenemos en la columna Value:

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

```
##
##      0      1  1.021  1.036  1.039  1.042  1.045  1.047  1.051  1.057
##     26     2      1      2      1      1      1      1      1      1
##  1.064  1.072  1.077  1.083  1.09  1.101  1.108  1.109  1.115  1.126
##      1      1      1      1      1      1      1      1      1      1
##   1.13  1.137  1.141  1.142  1.146  1.163  1.176  1.179  1.199  1.202
##      1      1      1      1      1      1      1      1      1      1
##  1.211  1.215  1.221  1.222  1.227  1.23  1.232  1.234  1.238  1.243
##      1      1      1      1      1      1      1      1      1      1
##  1.248  1.249  1.265  1.27  1.274  1.276  1.277  1.286  1.289  1.29
##      1      1      1      1      2      1      1      1      2      1
##  1.298  1.299  1.318  1.321  1.322  1.341  1.35  1.359  1.364  1.371
##      1      1      1      1      1      1      1      1      1      1
##  1.373  1.388  1.403  1.41  1.416  1.43  1.431  1.432  1.441  1.442
##      1      1      1      3      1      2      1      1      1      1
##  1.463  1.485  1.491  1.495  1.498  1.501  1.519  1.522  1.535  1.538
##      1      1      1      1      1      1      1      1      1      1
##  1.556  1.572  1.573  1.575  1.59  1.595  1.609  1.613  1.661  1.669
##      2      1      1      1      1      1      1      1      1      1
##  1.673  1.701  1.706  1.71  1.722  1.73  1.734  1.752  1.763  1.766
##      1      1      1      1      1      1      1      1      1      1
##  1.779  1.781  1.804  1.81  1.811  1.829  1.834  1.841  1.845  1.856
##      1      1      1      1      1      2      1      1      1      1
##  1.862  1.872  1.91  1.969  1.981  1.982  1.986  1.987      2  2.007
##      1      1      1      1      1      1      1      1      6      1
##  2.008  2.023  2.029  2.031  2.051  2.062  2.073  2.078  2.09  2.096
##      1      1      1      2      1      1      1      1      1      1
##  2.099  2.119  2.127  2.146  2.172  2.179  2.196  2.207  2.212  2.222
##      1      1      1      1      1      1      1      1      1      1
##  2.227  2.265  2.275  2.294  2.326  2.366  2.392  2.399  2.406  2.423
##      1      1      2      1      1      1      1      1      1      1
##  2.443  2.46  2.463  2.485  2.492  2.501  2.515  2.57  2.581  2.632
##      1      1      1      1      1      1      1      1      1      1
##  2.638  2.652  2.708  2.716  2.72  2.732  2.734  2.751  2.782  2.793
##      1      1      1      1      1      1      1      1      1      1
##  2.825  2.85  2.855  2.866  2.882  2.907  2.913  2.946  2.972  2.987
##      1      1      1      1      1      1      1      1      1      1
##  2.994      3  3.003  3.025  3.027  3.038  3.127  3.135  3.137  3.154
##      1     10      1      1      1      1      1      1      1      1
```

##	3.174	3.176	3.183	3.185	3.193	3.199	3.251	3.255	3.261	3.301
##	1	1	1	1	1	1	1	1	1	1
##	3.305	3.309	3.31	3.322	3.327	3.386	3.397	3.417	3.428	3.466
##	1	1	1	1	1	1	1	1	1	1
##	3.52	3.524	3.636	3.648	3.768	3.785	3.797	3.897	3.898	3.903
##	1	1	1	1	1	1	1	1	1	1
##	3.92	3.94	3.95	3.954	4	4.084	4.102	4.12	4.271	4.464
##	1	1	1	1	3	1	1	1	1	1
##	4.476	4.504	4.535	4.969	4.971	5	5.018	5.033	5.139	5.256
##	1	1	1	1	1	5	1	1	1	1
##	5.286	5.37	5.372	5.403	5.418	5.429	5.436	5.459	5.548	5.55
##	1	1	1	1	1	1	1	1	1	1
##	5.563	5.564	5.583	5.614	5.62	5.649	5.659	5.666	5.669	5.721
##	1	1	1	1	1	1	1	1	1	1
##	5.857	5.92	5.991	6	6.025	6.199	6.224	6.232	6.253	6.258
##	1	1	1	7	1	1	1	1	1	1
##	6.272	6.281	6.299	6.322	6.325	6.363	6.398	6.415	6.439	6.488
##	1	1	1	1	1	1	1	1	1	1
##	6.493	6.52	6.521	6.533	6.534	6.535	6.541	6.569	6.635	6.638
##	1	1	1	1	2	1	1	1	1	1
##	6.65	6.662	6.671	6.703	6.758	6.77	6.781	6.782	6.786	6.792
##	1	1	1	1	1	1	1	1	1	1
##	6.821	6.828	6.832	6.927	6.975	7	7.037	7.093	7.139	7.145
##	1	1	1	1	1	3	1	1	1	1
##	7.174	7.18	7.22	7.29	7.393	7.396	7.423	7.428	7.431	7.467
##	1	1	1	1	1	1	1	1	1	1
##	7.488	7.522	7.568	7.597	7.603	7.611	7.642	7.73	7.842	7.849
##	1	1	1	1	1	1	1	1	1	1
##	7.922	7.975	7.999	8	8.002	8.008	8.058	8.123	8.173	8.202
##	1	1	1	1	1	1	1	1	1	1
##	8.213	8.219	8.348	8.364	8.509	8.531	8.685	8.686	8.766	8.771
##	1	1	1	1	1	1	1	1	1	1
##	8.873	8.963	8.981	9.118	9.209	9.232	9.277	9.452	9.497	9.514
##	1	1	1	1	1	1	1	1	1	1
##	9.671	9.692	9.727	9.728	9.921	10	10.101	10.109	10.11	10.185
##	1	1	1	1	1	3	1	1	1	1
##	10.235	10.448	10.487	10.52	10.599	10.612	10.662	10.714	10.738	10.837
##	1	1	1	1	1	1	1	1	1	1
##	10.913	10.919	10.95	11	11.008	11.058	11.092	11.236	11.329	11.337
##	1	1	1	4	1	1	1	1	1	1
##	11.357	11.371	11.438	11.487	11.533	11.661	11.673	11.731	11.734	11.863
##	1	1	1	1	1	1	1	1	1	1
##	11.937	11.989	11.998	12	12.015	12.057	12.06	12.155	12.168	12.179
##	1	1	1	3	1	1	1	1	1	1
##	12.202	12.314	12.323	12.379	12.501	12.609	12.713	12.747	12.815	12.875
##	1	1	2	1	1	1	1	1	1	1
##	12.935	12.982	13	13.037	13.047	13.113	13.162	13.23	13.264	13.272
##	1	1	2	1	1	1	1	1	1	1
##	13.366	13.395	13.403	13.454	13.537	13.572	13.607	13.631	13.639	13.657
##	1	1	1	1	1	1	1	1	1	1
##	13.761	13.782	13.857	13.864	13.873	13.965	13.966	13.996	14	14.005
##	1	1	1	1	1	1	1	1	3	1
##	14.024	14.147	14.175	14.21	14.23	14.251	14.331	14.337	14.354	14.498
##	1	1	1	1	1	2	1	1	1	1

##	14.799	15	15.028	15.096	15.2	15.214	15.224	15.367	15.476	15.615
##	1	6	1	1	1	1	1	1	1	1
##	15.742	15.811	15.871	16.099	16.132	16.198	16.27	16.374	16.563	16.594
##	1	1	1	1	1	1	1	1	2	1
##	16.606	16.65	16.79	16.811	16.85	17	17.2	17.332	17.415	17.565
##	1	1	1	1	1	1	1	1	1	1
##	17.668	17.679	17.732	17.933	18.261	18.281	18.292	18.505	18.541	18.591
##	1	2	1	1	1	1	1	1	2	2
##	18.69	18.692	18.714	18.85	18.941	19.231	19.271	19.303	19.415	19.468
##	1	1	1	1	2	1	1	1	1	2
##	19.721	19.855	19.865	19.923	20	20.034	20.143	20.148	20.188	20.236
##	1	2	1	2	2	1	1	1	1	1
##	20.349	20.388	20.398	20.501	20.584	20.653	20.855	20.917	20.965	21
##	1	1	1	1	1	1	1	1	1	2
##	21.04	21.116	21.216	21.259	21.508	21.552	21.843	22	22.344	22.451
##	1	1	1	1	1	1	1	6	1	1
##	22.602	22.805	23	24	24.063	24.856	25	25.126	25.136	25.166
##	1	1	4	3	1	1	6	1	1	1
##	25.167	25.422	25.508	25.681	26	26.072	26.104	26.248	26.313	27
##	1	1	1	1	5	1	2	1	1	5
##	27.032	27.28	27.756	27.921	28	28.72	29	29.454	29.597	30
##	1	1	1	1	5	1	5	1	1	2
##	30.243	30.449	30.663	31	31.093	31.493	31.501	32	32.499	33
##	1	1	1	3	1	1	1	1	1	4
##	33.316	34	34.54	34.806	35	35.129	35.132	35.22	35.318	35.692
##	1	2	1	1	1	1	1	1	1	1
##	35.765	35.879	36	36.447	36.959	36.971	37	37.02	37.032	37.084
##	1	1	2	1	1	1	3	1	1	1
##	37.154	37.162	38	38.218	38.347	38.5	39.071	39.631	39.79	39.845
##	1	1	2	1	1	1	1	1	1	1
##	39.894	39.914	40	40.031	40.192	40.574	41	41.164	41.494	41.793
##	1	1	1	1	1	1	2	1	1	1
##	42	42.073	42.652	43	43.024	43.189	43.35	43.449	44.235	44.954
##	1	1	1	3	1	1	1	1	1	1
##	45	45.327	46.166	46.406	46.596	47	47.417	48.043	48.178	49
##	1	1	1	1	1	1	1	1	1	1
##	49.18	50	50.545	51	51.296	51.775	52	52.119	53	54
##	1	1	1	4	1	1	4	1	2	1
##	55	55.183	56	56.143	57	58.808	59	61	62	62.439
##	2	1	2	1	2	1	2	2	1	1
##	64.715	64.91	65.633	65.954	66	66.174	66.554	67	67.203	67.258
##	1	1	1	1	3	1	1	4	1	1
##	67.644	68.816	69.581	69.655	69.9	70.902	70.964	71	71.046	71.192
##	1	1	1	1	1	1	1	2	1	1
##	71.236	71.389	71.64	72	72.629	73	73.665	74	74.154	74.256
##	1	1	1	1	1	1	1	1	1	1
##	75.839	76.452	77	77.202	77.922	78	78.244	79	79.394	80
##	1	1	2	1	1	3	1	1	1	1
##	81.422	81.577	82	82.328	82.76	82.777	83	84	84.315	84.547
##	1	1	2	1	1	1	1	1	1	1
##	84.7	84.816	85.009	86	86.195	86.36	86.93	87.269	88	88.241
##	1	1	1	1	1	1	1	1	2	1
##	89.646	90	90.262	91	91.629	91.69	92	92.518	93.21	93.824
##	1	1	1	1	1	1	1	1	1	1

##	93.897	93.989	94.417	95.418	96	96.783	96.922	97	97.194	97.384
##	1	1	1	1	3	1	1	1	1	1
##	97.518	97.532	97.815	98.35	98.679	98.738	99	99.715	100.001	100.473
##	1	1	1	1	1	1	1	1	1	1
##	100.742	101.081	103.625	103.899	104	105	107.023	108.109	108.694	109
##	1	1	1	1	1	1	1	1	1	1
##	110	116	117	118	120	122	124	126	128	133
##	1	2	1	1	1	1	1	1	1	1
##	141	141.526	142.676	143	144.317	146	146.613	147.963	150.697	153.085
##	1	1	1	1	1	1	1	1	1	1
##	154	155	160	173	176	183	193	205	208	209.392
##	1	1	1	1	1	1	1	1	1	1
##	218.751	224.272	227	228	229	229.998	230.575	232	232.178	236.311
##	1	1	2	1	2	1	1	1	1	1
##	238	240.259	242.123	242.3	244	246	248.958	249	252.075	253
##	1	1	1	1	1	1	1	1	1	1
##	256.954	257	258	261.567	261.667	264	265	265.763	267	272
##	1	1	1	1	1	2	1	1	1	1
##	273	274.841	276	278	282	283	284.568	286	287	290.266
##	1	1	1	1	1	1	1	2	1	1
##	292	294	295	297	297.784	298	309.02	310	320	322.481
##	1	1	1	1	1	2	1	1	2	1
##	324	325	326	328	329	331	332	333	338	338.267
##	1	2	1	2	1	2	3	2	1	1
##	339	340	342	344	350	351	352	352.045	355	358.609
##	1	1	1	2	2	2	1	1	1	1
##	360	361.966	364	367	369.091	372	372.249	373	376	377.162
##	1	1	2	1	1	1	1	1	3	1
##	377.926	378	383.636	383.906	384	385	387	389.134	389.966	391
##	1	1	1	1	1	1	1	1	1	1
##	393.675	395	396	398	399.054	399.962	402.148	405	407	407.731
##	1	1	1	1	1	1	1	1	1	1
##	408	408.679	409	411.422	416	417.509	418	418.567	420	423
##	1	1	1	1	1	1	1	1	4	2
##	424	427	430.796	433	434	436	439	440	441	442
##	1	2	1	1	1	1	1	1	1	1
##	443	444	444.547	452	454	456.184	457	458	459	461
##	1	1	1	1	2	1	1	1	1	1
##	463	465	469.913	471	473	475	476.558	477	481	483.524
##	1	2	1	1	1	2	1	1	1	1
##	484	489	490	492	494	496.066	497	499	500	505
##	1	3	1	1	1	1	1	2	1	1
##	509	511	512	516	520	522	522.72	523.922	524	524.006
##	2	1	2	1	1	1	1	1	1	1
##	525.859	529	532	532.38	537	540	543.353	546	551.868	553.241
##	1	1	1	1	1	1	1	1	1	1
##	554.754	556	556.185	557	559	561	562	565.569	567.108	572
##	1	1	1	1	1	1	1	1	1	1
##	576	577	579	580	580.535	582.204	585	587	592	593
##	1	1	1	1	1	1	1	2	1	1
##	598	605	609	617	622	624	636	638	641	656
##	1	1	1	1	1	1	1	1	1	1
##	658	668	679	692	693	698	704	705	706	708
##	1	1	1	1	1	1	2	1	1	2

##	709	716	722	724	727	728	737	739	741	742
##	1	1	1	1	1	1	2	1	1	1
##	745	751	752	757	759	762	764	765	766	774
##	1	1	1	2	3	1	1	1	1	2
##	786	788	795	800	804	808	810	813	824	831
##	1	1	1	1	1	1	1	1	1	1
##	834	835	849	851	852	854	858	864	873	887
##	1	2	1	1	1	1	1	1	1	1
##	889	890	898	901	908	922	925	932	938	939
##	1	2	1	1	1	1	1	1	2	1
##	945	948	949	961	966	969	970	975	981	986.082
##	1	2	1	1	1	1	1	1	1	1
##	991	999	<NA>							
##	1	1	731							

- Observamos que tenemos **731 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)
```

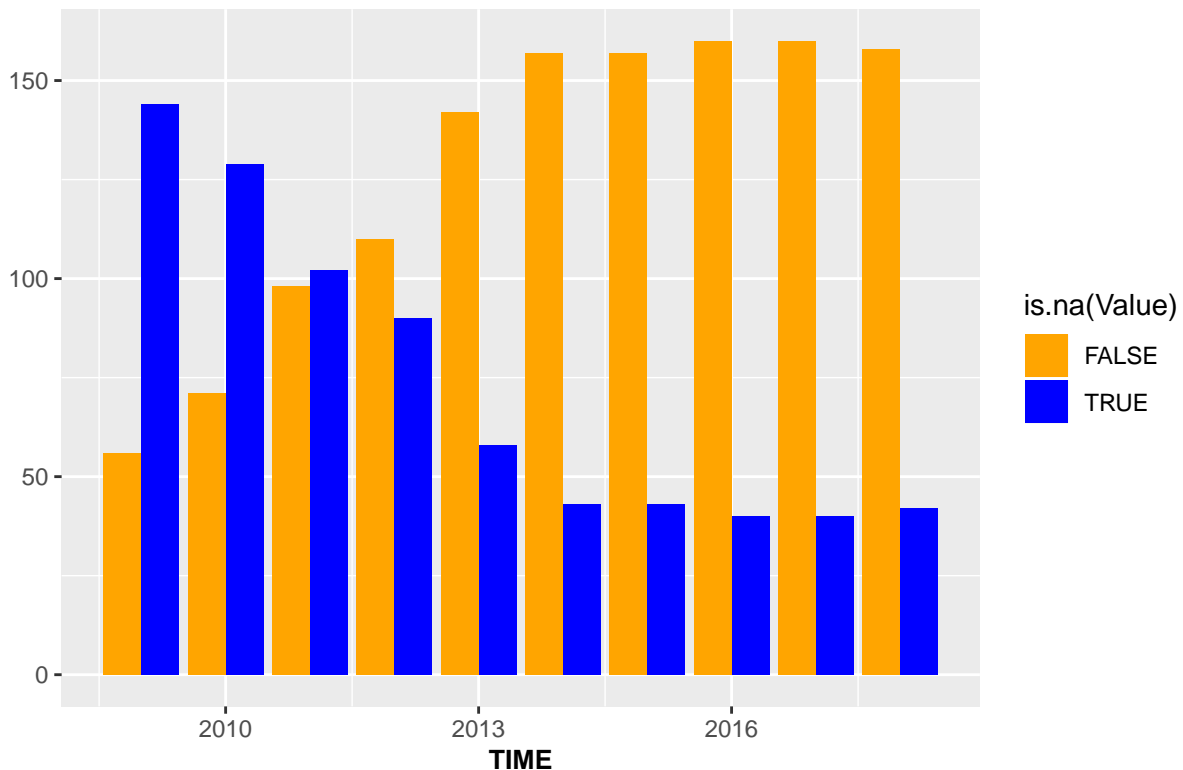
```
## [1] 731
```

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

```
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))
```

## 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(gasto_pro, variable=c("Value"),k=3)
```

```
gasto_pro<-output
```

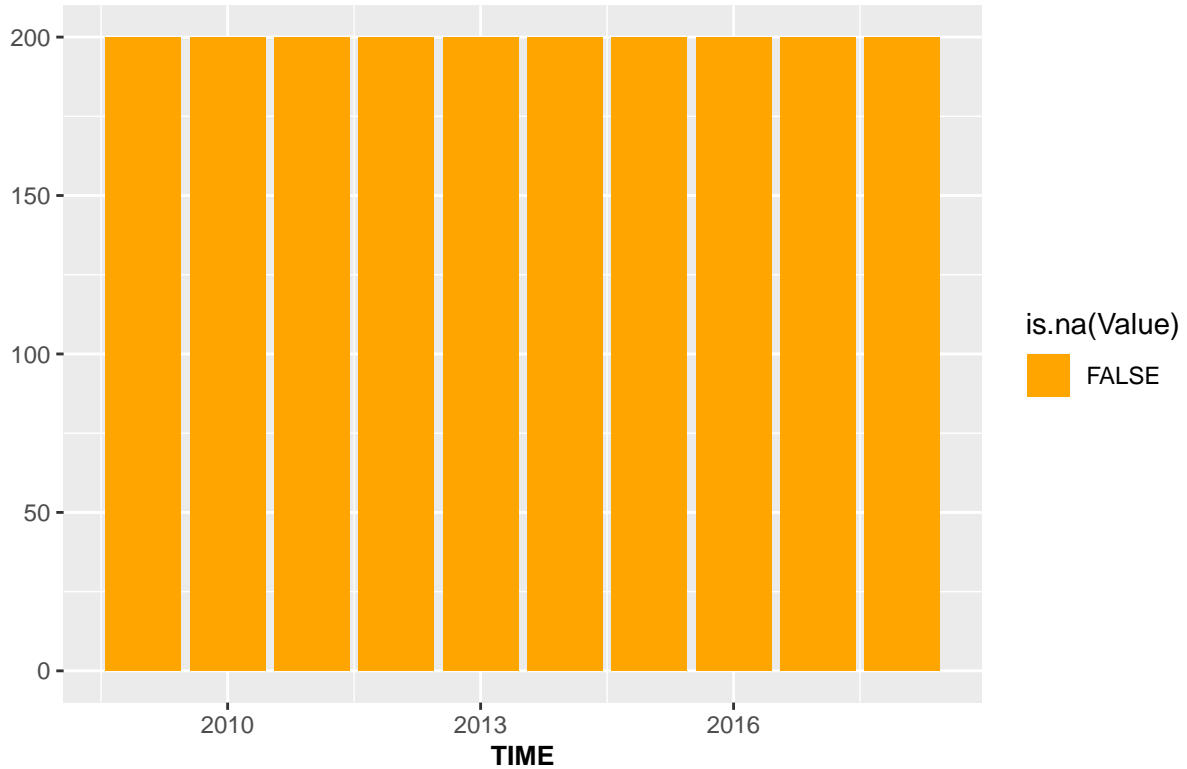
- Comprobamos que no tenemos valores nulos después de la imputación

```
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"))
```



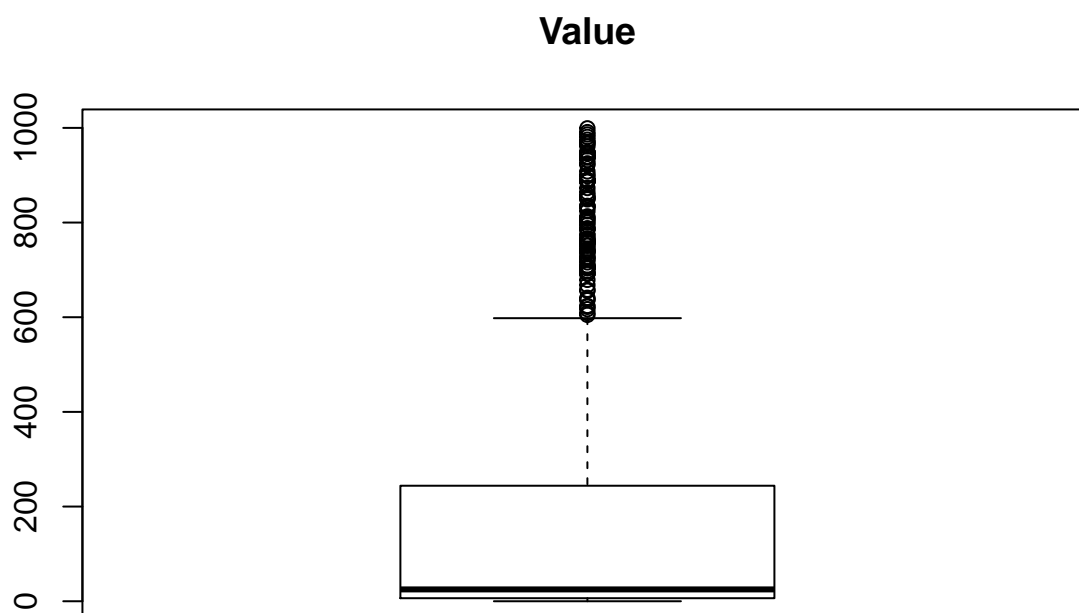
```
g+geom_bar(position="dodge") + scale_fill_manual(values = alpha(c("orange", "blue"), 1)) +  
theme(axis.title.x = element_text(face="bold", size=10))
```

## Valores Nulos



- Con el siguiente gráfico, observaremos que la variable **Value** 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)
```