

# A1.Financiacion del Gasto Sanitario

*Alicia Perdices Guerra*

*12 de abril, 2021*

## Contents

### 1.PROCESAMIENTO DE LOS DATOS.

- En primer lugar leemos el fichero:

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

- Realicemos una breve inspección de los datos

```
str(gasto_f)
```

```
## '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_HF     : Factor w/ 5 levels "All financing schemes",...: 1 4 3 2 5 1 4 3 2 5 ...
## $ Value         : Factor w/ 1185 levels ":", "0.00", "1,001,514.67",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ Flag.and.Footnotes: Factor w/ 2 levels "","b": 1 1 1 1 1 1 1 1 1 1 ...
```

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

```
## [1] "TIME"          "GEO"           "UNIT"
## [4] "ICHA11_HF"     "Value"         "Flag.and.Footnotes"
```

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

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

```
ncol(gasto_f) #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\_HF**: variable cualitativa. Indica el organismo que financia la sanidad, ya sea gubernamental o por seguros privados etc..
- **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_f$TIME)
```

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

\*Países:

```
unique(gasto_f$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_f$UNIT)
```

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

\*Variable que indica cómo se financia el gasto sanitario

```
unique(gasto_f$ICHA11_HF)
```

```
## [1] All financing schemes
```

```
## [2] Government schemes and compulsory contributory health care financing schemes
## [3] Government schemes
## [4] Compulsory contributory health insurance schemes and compulsory medical saving accounts (CMSA)
## [5] Social health insurance schemes
## 5 Levels: All financing schemes ...
```

- Eliminamos la columna Fal.and.footnotes.

```
gasto_f<-gasto_f[,-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_f$Value<-as.character(gasto_f$Value)
gasto_f$Value<-(gsub(',', '.',gasto_f$Value) )
gasto_f$Value<-substr(gasto_f$Value,1,nchar(gasto_f$Value)-3)
gasto_f$Value<-as.numeric(gasto_f$Value)
```

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

- Comprobamos que valores tenemos en la columna Value:

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

```
##
##      0  1.001  1.009  1.029  1.03  1.042  1.045  1.066  1.067  1.069
##    89      1      1      2      2      1      1      1      1      1
##   1.08  1.108  1.109  1.114  1.12  1.137  1.144  1.157  1.172  1.211
##      2      1      1      1      2      1      2      1      2      1
##  1.218  1.222  1.227  1.232  1.233  1.234  1.24  1.245  1.249  1.265
##      1      2      1      1      1      1      2      1      1      1
##  1.274  1.277  1.289  1.29  1.301  1.318  1.322  1.35  1.364  1.369
##      4      2      2      1      2      1      1      1      1      1
##  1.385  1.388  1.395  1.402  1.41  1.411  1.422  1.43  1.432  1.435
##      2      1      2      1      2      1      1      1      1      1
##  1.442  1.466  1.473  1.482  1.497  1.519  1.522  1.53  1.533  1.553
##      1      1      2      2      1      1      1      1      2      1
##  1.556  1.562  1.572  1.577  1.601  1.609  1.61  1.624  1.627  1.64
##      3      2      1      1      1      1      2      1      1      2
##  1.705  1.719  1.734  1.741  1.783  1.791  1.804  1.807  1.81  1.812
##      1      1      1      2      1      2      1      1      1      1
##  1.823  1.862  1.9  1.91  1.933  1.946  1.981  1.993  2.005  2.013
##      1      1      1      1      1      2      1      1      2      1
##  2.016  2.027  2.031  2.035  2.037  2.04  2.059  2.067  2.096  2.108
##      1      1      1      1      2      1      1      1      1      1
##  2.135  2.141  2.146  2.156  2.176  2.18  2.193  2.248  2.261  2.264
##      1      2      1      1      2      1      1      2      1      1
##  2.265  2.275  2.301  2.325  2.345  2.357  2.366  2.376  2.383  2.391
##      1      1      1      2      2      1      2      1      1      1
##  2.392  2.393  2.411  2.414  2.422  2.423  2.434  2.436  2.441  2.463
##      1      2      1      2      1      2      1      1      1      1
##  2.484  2.493  2.5  2.501  2.507  2.509  2.522  2.525  2.53  2.541
##      1      1      1      1      1      1      1      2      2      1
##  2.55  2.57  2.581  2.589  2.591  2.594  2.605  2.614  2.627  2.636
##      2      1      1      1      1      1      1      1      1      2
##  2.638  2.645  2.653  2.668  2.708  2.718  2.732  2.746  2.751  2.765
##      1      1      1      1      2      2      1      1      1      1
```

##	2.773	2.782	2.822	2.823	2.841	2.85	2.855	2.873	2.881	2.907
##	2	1	1	1	1	1	1	1	1	1
##	2.919	2.942	2.946	2.949	2.972	2.987	3	3.003	3.027	3.14
##	1	1	1	1	1	1	6	1	1	1
##	3.174	3.183	3.185	3.199	3.305	3.309	3.327	3.336	3.346	3.377
##	2	1	1	1	1	1	1	1	1	1
##	3.386	3.428	3.479	3.482	3.52	3.524	3.564	3.628	3.636	3.771
##	1	1	1	1	1	1	1	1	1	1
##	3.797	3.81	3.877	3.898	3.916	3.918	3.973	3.984	4	4.005
##	1	1	2	1	1	1	1	1	4	1
##	4.006	4.007	4.021	4.025	4.054	4.056	4.059	4.063	4.084	4.087
##	1	2	2	2	1	2	2	2	1	1
##	4.095	4.118	4.12	4.143	4.202	4.205	4.21	4.217	4.22	4.226
##	2	2	1	1	1	2	1	1	2	2
##	4.27	4.274	4.286	4.293	4.301	4.319	4.365	4.38	4.387	4.401
##	2	2	2	2	2	1	1	2	2	2
##	4.435	4.437	4.439	4.456	4.469	4.518	4.52	4.553	4.573	4.599
##	2	1	1	2	2	1	2	1	1	1
##	4.602	4.634	4.638	4.663	4.665	4.669	4.693	4.801	4.832	4.841
##	1	2	1	2	1	1	2	1	1	2
##	4.868	4.929	4.952	4.959	4.978	4.987	5.024	5.082	5.084	5.111
##	1	1	1	2	2	1	1	1	1	2
##	5.198	5.256	5.271	5.418	5.457	5.532	5.55	5.583	5.666	5.721
##	2	1	1	1	2	1	3	1	1	1
##	5.898	5.924	5.978	5.991	6.031	6.109	6.115	6.181	6.199	6.225
##	1	1	1	1	2	2	1	1	1	1
##	6.281	6.475	6.664	6.832	7	7.242	7.396	7.428	7.431	7.467
##	1	1	1	1	2	2	1	1	1	1
##	7.488	7.568	7.607	7.642	7.73	7.765	7.922	8	8.078	8.088
##	1	1	1	1	1	1	1	6	2	1
##	8.123	8.182	8.345	8.372	8.509	8.511	8.531	8.584	8.678	8.718
##	1	1	2	1	1	1	1	2	1	2
##	8.783	8.805	8.871	8.963	9	9.037	9.066	9.15	9.272	9.383
##	2	1	1	1	2	1	1	2	1	1
##	9.552	9.569	9.601	9.667	9.671	9.913	10.002	10.05	10.118	10.12
##	1	2	1	1	1	1	1	1	1	1
##	10.139	10.144	10.175	10.296	10.317	10.319	10.322	10.337	10.357	10.386
##	1	1	1	1	1	1	1	1	1	1
##	10.433	10.474	10.664	10.674	10.712	10.734	10.8	10.826	11.026	11.07
##	1	1	1	1	1	1	1	1	1	2
##	11.074	11.113	11.171	11.186	11.242	11.366	11.371	11.381	11.411	11.417
##	1	1	1	1	1	1	1	1	1	1
##	11.464	11.788	11.831	11.88	11.937	11.989	12.031	12.118	12.151	12.202
##	1	1	1	1	1	1	1	1	1	1
##	12.245	12.257	12.28	12.314	12.326	12.609	12.667	12.737	12.739	12.793
##	1	1	1	1	1	1	1	1	1	1
##	12.887	12.906	12.914	12.941	13.082	13.113	13.119	13.173	13.177	13.178
##	1	1	1	1	2	1	1	1	1	1
##	13.192	13.202	13.233	13.258	13.269	13.299	13.322	13.386	13.435	13.488
##	2	1	1	1	1	1	1	1	1	1
##	13.51	13.599	13.794	13.842	13.854	13.864	14.024	14.093	14.152	14.173
##	1	2	1	1	1	1	1	1	2	1
##	14.21	14.251	14.274	14.338	14.354	14.498	14.512	14.528	14.592	14.666
##	1	1	1	1	1	1	2	1	1	1

##	14.82	15.028	15.067	15.13	15.177	15.265	15.272	15.396	15.473	15.476
##	1	1	1	2	1	1	1	1	1	1
##	15.504	15.549	15.615	15.616	15.631	15.652	15.687	15.72	15.742	15.752
##	1	1	1	1	1	2	1	2	1	1
##	15.775	15.783	15.788	15.821	15.871	16.127	16.132	16.242	16.306	16.442
##	1	1	1	1	1	1	1	1	2	1
##	16.453	16.459	16.471	16.57	16.589	16.605	16.65	16.79	16.811	16.957
##	1	1	1	1	1	1	1	1	1	2
##	17.03	17.2	17.332	17.565	17.668	17.692	17.781	17.93	18.071	18.146
##	1	1	1	1	1	2	1	1	1	1
##	18.261	18.281	18.292	18.505	18.533	18.597	18.69	18.714	18.793	18.85
##	1	1	1	1	1	1	1	1	1	1
##	18.874	19.092	19.137	19.231	19.241	19.271	19.296	19.303	19.381	19.387
##	1	1	1	1	1	1	1	1	1	1
##	19.629	20.034	20.143	20.169	20.236	20.388	20.398	20.653	20.999	21.043
##	1	1	1	1	1	1	1	1	1	2
##	21.076	21.116	21.246	21.259	21.508	21.895	21.945	21.963	22.12	22.171
##	2	1	1	1	1	2	1	1	1	2
##	22.344	22.451	22.521	22.576	22.758	22.769	23.027	23.2	23.245	23.474
##	1	1	1	1	2	2	1	1	1	1
##	23.494	23.505	24.141	24.16	24.27	24.335	24.354	24.475	24.498	24.656
##	1	2	1	2	1	1	1	1	1	1
##	24.756	24.77	24.79	24.869	25.126	25.138	25.157	25.166	25.167	25.355
##	1	1	1	2	1	1	1	1	1	1
##	25.541	25.563	25.633	25.65	25.675	25.681	25.695	25.714	26.072	26.26
##	2	1	1	1	1	1	1	2	1	1
##	26.28	26.313	26.433	26.587	26.644	26.663	27	27.032	27.08	27.099
##	1	1	1	2	1	1	1	1	1	1
##	27.28	27.304	27.38	27.603	27.756	27.92	27.921	27.938	28.272	28.368
##	1	1	1	1	1	1	1	1	1	1
##	28.72	29	29.139	29.454	29.5	29.597	29.705	29.724	29.868	29.963
##	1	2	1	1	2	1	1	1	2	2
##	29.999	30.118	30.155	30.243	30.312	30.449	30.663	30.88	31	31.024
##	1	2	2	1	1	1	1	1	1	2
##	31.028	31.062	31.093	31.202	31.493	31.501	31.539	31.544	31.912	32.499
##	1	1	1	1	1	1	1	2	1	1
##	32.923	33.316	33.662	33.776	34.54	34.806	35	35.129	35.132	35.22
##	1	1	1	1	1	1	1	1	1	1
##	35.248	35.318	35.692	35.879	35.947	36.33	36.447	36.514	36.971	37.02
##	1	1	1	1	1	2	1	1	1	1
##	37.162	37.595	38	38.347	38.865	38.878	39.071	39.79	40.369	40.483
##	1	1	1	1	1	2	1	1	2	2
##	40.574	41	41.349	41.494	42.073	42.669	43	43.024	43.449	43.649
##	1	1	2	1	1	2	1	1	1	2
##	43.883	44.235	45.327	45.651	46.133	46.166	46.406	46.511	47.189	47.417
##	2	1	1	1	1	1	1	1	1	1
##	48	48.043	48.178	48.59	49	49.18	50.545	50.782	51.296	51.638
##	1	1	1	1	2	1	1	1	1	1
##	51.775	51.936	52	52.119	53	53.023	53.242	53.535	53.973	54.119
##	1	1	1	1	1	1	1	1	1	1
##	54.333	55.016	55.183	56	56.029	56.143	56.506	56.722	57.537	57.972
##	1	1	1	1	1	1	1	1	1	1
##	58.404	58.808	58.931	60	60.631	61.636	61.808	62.439	63	63.356
##	1	1	1	4	1	1	1	1	4	1

##	64.077	64.91	65.695	65.92	65.926	66.554	67.057	67.646	68.085	68.754
##	1	1	1	1	1	1	1	1	1	1
##	68.816	69.106	69.137	69.655	69.9	70.14	70.902	70.964	71.046	71.236
##	1	1	1	1	1	1	1	1	1	1
##	71.35	71.592	71.64	71.672	72.629	72.817	73.106	73.408	74.256	76
##	1	1	1	1	1	1	1	1	1	1
##	76.112	77	77.202	86	92.518	93.824	94	94.417	97.384	97.532
##	1	3	1	2	1	1	1	1	1	1
##	97.815	98	98.35	99.715	101	103.899	106	106.877	107.218	107.932
##	1	2	1	1	1	1	1	1	1	1
##	108.109	108.258	108.5	108.789	108.851	109.132	109.819	110.086	110.857	111.117
##	1	1	1	1	1	1	1	1	1	1
##	112	112.885	113	113.112	118	119	124	126	128	134
##	1	1	1	1	2	1	2	1	1	1
##	136	137	138	140	141	141.526	142	142.676	144	144.317
##	3	1	2	2	1	1	1	1	2	1
##	145	146	146.613	147.963	150	150.697	151	153.085	156	156.594
##	1	2	1	1	1	1	1	1	2	2
##	157	160.07	164	164.087	165.915	167.421	168.094	171.096	172.422	175
##	1	2	4	2	2	1	2	1	2	2
##	175.33	177	177.818	179	179.882	180.297	181	182	182.709	182.777
##	1	1	2	3	1	2	4	2	2	2
##	184	184.1	184.679	185	187.631	188	188.255	188.458	190.413	190.558
##	1	1	1	1	1	1	1	2	1	1
##	191.383	192	193.184	194.905	196	197	198	198.169	199	200
##	2	1	1	1	1	1	2	1	2	1
##	200.238	203.38	204.044	205	207.927	207.954	209.392	210	210.249	213
##	1	1	1	1	1	2	1	1	1	2
##	213.44	213.987	214	217.591	217.886	218.751	221	221.609	222.266	224.272
##	1	1	1	1	1	1	1	1	1	1
##	225.308	226.896	227	228	229.043	229.998	230.575	232.178	234	236.213
##	1	1	2	1	1	1	1	1	1	1
##	236.311	236.887	238.061	240.259	241	241.238	242.123	242.3	245	246.387
##	1	1	1	1	1	1	1	1	2	1
##	247.064	248.958	249.583	251	252.075	256.954	259.06	260	260.651	261.421
##	1	1	1	1	1	1	1	2	1	1
##	261.567	261.667	261.705	262.928	263.67	265.763	265.897	266.126	267	271.546
##	1	1	1	1	1	1	1	1	2	1
##	271.97	272.309	274.757	274.841	276	277.11	277.351	281.357	283	283.787
##	1	1	1	1	1	1	1	1	1	1
##	284.032	284.568	284.731	286.743	287.82	289.314	289.559	290.266	295.447	296.807
##	1	1	1	1	1	1	1	1	1	1
##	297.784	298.268	298.52	299.647	309.02	310	311.962	320	322	322.481
##	1	1	1	1	1	1	1	1	1	1
##	324	324.402	325	326	329	335.183	338.267	341	343	344
##	1	1	1	2	1	1	1	2	2	1
##	348.58	350	351	352.045	357.452	366.108	369.091	371	372	376
##	1	1	2	1	1	1	1	1	1	1
##	376.125	383.636	389	413	428	436	456	460.441	462	478.171
##	1	1	1	2	1	2	1	2	2	2
##	485.457	487	489.222	499.883	504.199	509	509.576	513	514	517.874
##	2	2	1	2	1	1	1	1	1	1
##	517.96	519.475	521	521.069	525	529	536.892	538	538.068	538.206
##	1	2	1	1	1	1	2	2	2	1

```
## 539.571 545.315 547.057 547.568 548.597 548.759 548.818 548.836 550.459 553.096
##      1      1      1      1      1      1      1      1      2      1
##    554 556.448 556.535 557.667    558 559.053 564.437 564.583    567    575
##      1      1      1      1      1      1      1      1      1      1
##    576    581 582.432    583    590 590.111 591.585 592.075 594.849    595
##      1      1      2      1      1      1      1      1      2      2
## 595.264 597.679    601 605.491    606    608    609    610    611 613.275
##      1      2      1      2      1      1      2      1      1      1
##    614 614.837    615    622 625.228    626 629.007 631.557 633.651 635.393
##      1      1      1      2      2      1      1      2      1      1
##    637    642    648    655 655.577 658.103 680.552 683.325    687    690
##      2      1      1      3      1      2      1      2      2      1
##    700    704    711    719    720    723    735    739    744    750
##      1      2      1      2      1      1      2      2      1      2
##    767 767.126    770 780.126    795    801 801.739    802 810.939 812.47
##      1      1      2      1      3      1      1      2      1      1
##    814 825.216 834.776 836.403    854    860 863.789    869 873.898 875.617
##      2      1      1      1      2      1      1      2      1      1
##    883    889    892 892.236    898 902.645 904.452    907    910    916
##      2      1      2      1      1      1      1      2      1      2
## 919.005 920.675    922    925    928 931.919    932 933.912    936    937
##      1      1      2      1      1      1      1      1      1      2
##    939    945 946.258    965    967    970    974    977 986.082 989.009
##      1      1      1      1      1      3      2      2      1      1
##    991    997    999    <NA>
##      1      1      1    568
```

- Observamos que tenemos **568 valores perdidos**. Guardamos en la variable **idx** los índices de los registros con valores **NA** de la variable **Value**.

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

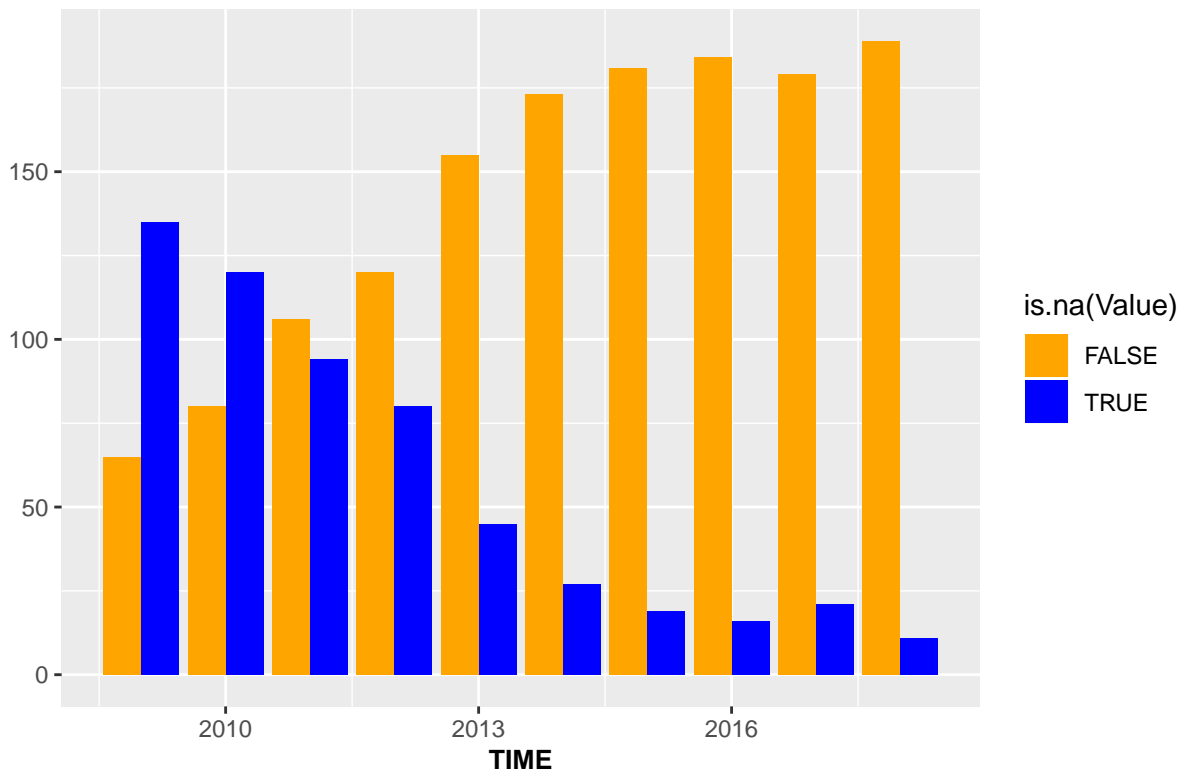
```
## [1] 568
```

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

```
library(ggplot2)
library(scales)
g = ggplot(gasto_f, 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_f, variable=c("Value"),k=3)
gasto_f<-output
```

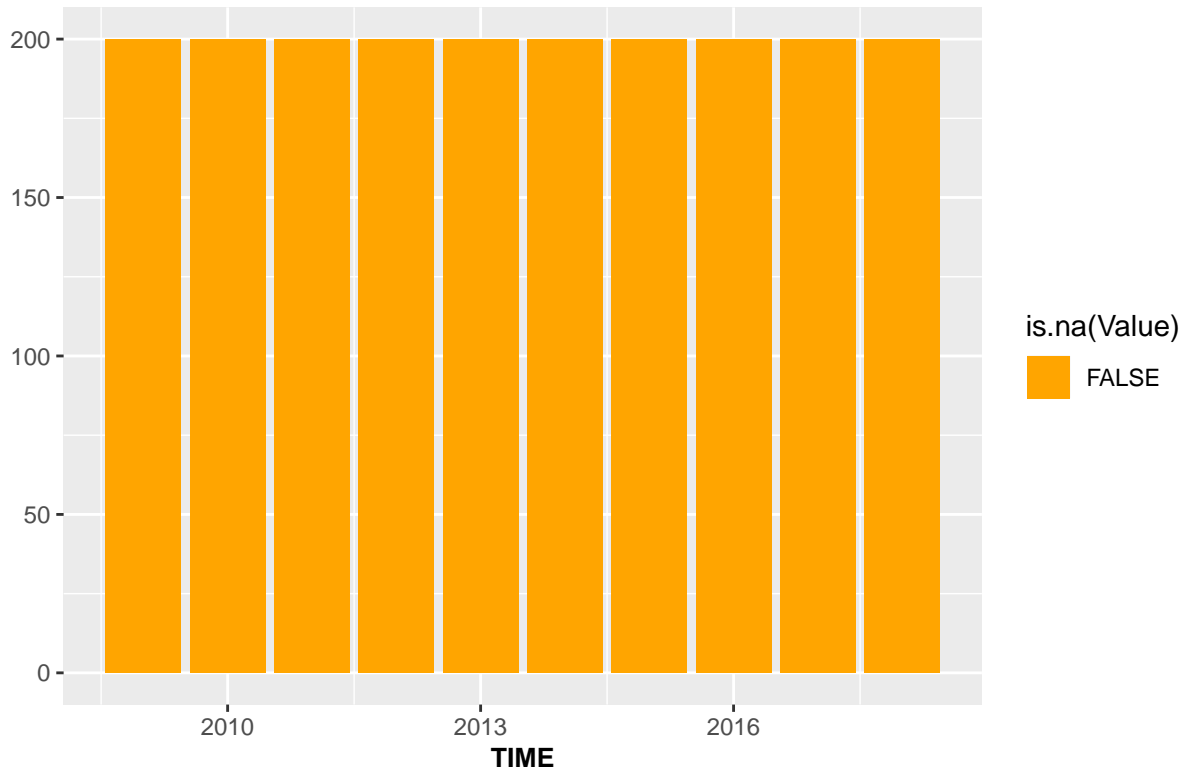
- Comprobamos que no tenemos valores nulos después de la imputación

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



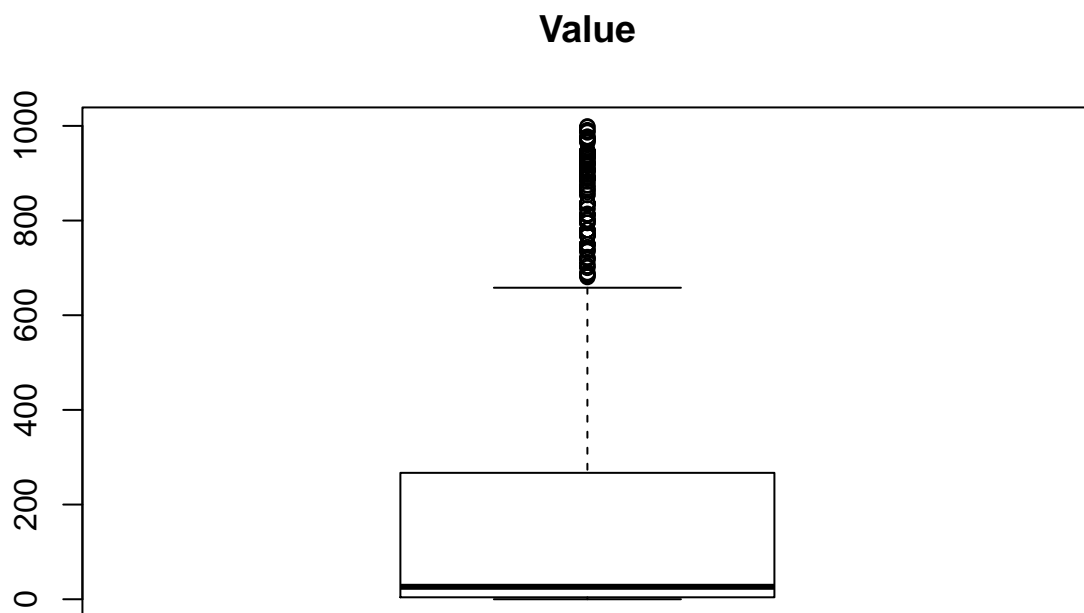
```
g+geom_bar(position="dodge") + scale_fill_manual(values = alpha(c("orange", "blue"), 1)) +  
theme(axis.title.x = element_text(face="bold", size=10))
```

## Valores Nulos



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

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



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

```
table(gasto_f$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_f$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_f$UNIT, useNA = "ifany")
```

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

```
table(gasto_f$ICHA11_HF, useNA = "ifany")
```

```
##
##                                     All financing schemes
##                                     400
## Compulsory contributory health insurance schemes and compulsory medical saving accounts (CMSA)
##                                     400
##                                     Government schemes
##                                     400
##          Government schemes and compulsory contributory health care financing schemes
##                                     400
##                                     Social health insurance schemes
##                                     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_f, file="GastoSanitario_Financiacion_clean.csv", row.names = FALSE)
```