

Médicos Por Países

Alicia Perdices Guerra

3 de mayo, 2021

Contents

1.PROCESAMIENTO DE LOS DATOS.

- En primer lugar leemos el fichero:

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

- Realicemos una breve inspección de los datos

```
str(medicos_p)
```

```
## 'data.frame': 1900 obs. of 7 variables:
## $ TIME : int 2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...
## $ GEO : Factor w/ 38 levels "Austria","Belgium",...: 9 9 9 9 9 10 10 10 10 10 ...
## $ UNIT : Factor w/ 1 level "Number": 1 1 1 1 1 1 1 1 1 1 ...
## $ AGE : Factor w/ 5 levels "From 35 to 44 years",...: 5 4 1 2 3 5 4 1 2 3 ...
## $ SEX : Factor w/ 1 level "Total": 1 1 1 1 1 1 1 1 1 1 ...
## $ Value : Factor w/ 1411 levels ":", "1 008", "1 037",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ Flag.and.Footnotes: Factor w/ 6 levels "","b","be","d",...: 1 1 1 1 1 1 1 1 1 1 ...
```

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

```
## [1] "TIME" "GEO" "UNIT"
## [4] "AGE" "SEX" "Value"
## [7] "Flag.and.Footnotes"
```

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

```
## [1] 1900
```

```
ncol(medicos_p) #Número de variables
```

```
## [1] 7
```

*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.
- **AGE**: variable cualitativa. Indica el rango de edad de los médicos disponibles. Se carga bien como factor.
- **SEX**: variable cualitativa. Único valor “Todos”. Se carga bien como Factor.
- **Value**: Variable cuantitativa. Indica el valor en número de de médicos por países.
- **Fal.and.footnotes**. Notas sobre etiquetas. Eliminamos esta columna.

*Años de las mediciones:

```
unique(medicos_p$TIME)
```

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

*Países:

```
unique(medicos_p$GEO)
```

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

*Unidad de las mediciones:

```
unique(medicos_p$UNIT)
```

```
## [1] Number
## Levels: Number
```

*Variable que indica el rango de edad de los médicos disponibles por países:

```
unique(medicos_p$AGE)
```

```
## [1] Total          Less than 35 years  From 35 to 44 years
## [4] From 45 to 54 years  From 55 to 64 years
```

```
## 5 Levels: From 35 to 44 years From 45 to 54 years ... Total
```

- Eliminamos la columna **Fal.and.footnotes** y **SEX** ya que no proporcionan información relevante.

```
medicos_p<-medicos_p[,-5]
medicos_p<-medicos_p[,-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.

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

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

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

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

```
##
##      2      4      5      9     11     25     26     27     28     31
##      5      1      1      1      2      1      2      4      3      1
##     33     34     37     38     39     40     41     42     43     45
##      1      1      3      1      2      4      1      3      1      2
##     63     69     71     72     73     76     85    104    111    114
##      1      1      1      1      2      1      1      1      1      1
##    116    119    120    124    135    136    138    143    144    152
##      1      1      2      2      1      1      1      1      1      1
##    166    174    175    177    186    187    189    195    198    205
##      1      1      2      1      1      1      2      2      1      1
##    213    227    229    230    233    235    236    237    240    242
##      1      2      2      1      1      1      1      1      1      1
##    243    244    247    249    251    259    261    270    280    283
##      1      1      2      2      1      1      1      1      1      1
##    285    287    288    292    293    295    297    300    303    306
##      1      1      1      1      1      1      1      1      1      1
##    311    312    313    317    321    323    325    327    328    330
##      1      1      3      1      1      1      1      1      1      1
##    331    332    333    334    335    342    345    347    348    351
##      1      2      1      2      1      1      1      1      1      1
##    353    354    355    356    357    359    360    362    364    365
##      2      1      2      1      1      1      3      2      2      1
##    367    370    372    373    379    382    385    386    388    389
##      1      1      1      1      1      1      1      1      1      1
##    390    398    400    406    411    414    425    429    434    436
##      1      1      1      1      1      1      1      1      1      1
##    452    454    457    478    491    494    498    499    509    511
##      1      1      2      1      1      1      1      1      1      1
##    517    519    522    527    532    534    550    553    559    562
##      1      1      1      1      1      1      1      1      1      1
##    566    573    609    619    624    632    633    638    639    652
##      1      1      1      1      1      1      1      1      1      1
##    653    654    661    667    687    690    704    705    706    717
##      1      2      1      1      2      1      1      1      1      1
##    731    746    749    757    760    767    782    804    809    810
```

##	1	1	1	1	1	1	1	1	1	1
##	813	830	834	835	839	861	865	879	892	899
##	1	1	1	1	1	1	1	1	1	1
##	902	903	904	905	906	930	943	954	959	977
##	1	1	1	1	1	1	1	1	1	1
##	996	1008	1037	1040	1047	1055	1060	1070	1079	1089
##	1	1	1	1	1	1	1	1	1	1
##	1100	1106	1110	1118	1142	1159	1163	1166	1168	1169
##	1	1	1	1	2	1	1	1	2	1
##	1172	1178	1180	1193	1200	1204	1208	1214	1216	1220
##	1	1	1	1	1	1	1	1	2	1
##	1248	1249	1259	1273	1276	1279	1280	1284	1290	1292
##	1	1	1	1	2	2	1	1	1	1
##	1299	1302	1306	1319	1323	1330	1338	1339	1342	1346
##	1	1	1	1	1	1	1	1	1	1
##	1347	1348	1354	1357	1370	1373	1374	1376	1379	1381
##	1	1	1	1	1	1	1	1	1	1
##	1389	1391	1397	1404	1407	1409	1413	1416	1427	1429
##	1	1	1	1	1	1	1	1	1	1
##	1439	1441	1450	1466	1473	1476	1489	1490	1495	1496
##	1	1	1	1	1	1	1	1	1	1
##	1505	1516	1526	1531	1537	1566	1571	1573	1583	1586
##	1	1	1	1	1	2	1	1	1	1
##	1603	1604	1616	1626	1632	1636	1657	1659	1664	1676
##	1	1	1	1	1	1	1	1	1	1
##	1680	1683	1689	1692	1705	1730	1740	1741	1743	1781
##	1	2	1	1	1	1	1	1	1	1
##	1785	1797	1826	1842	1854	1855	1870	1891	1895	1901
##	1	1	1	1	1	1	1	1	1	1
##	1904	1912	1925	1972	1982	2007	2035	2037	2040	2051
##	1	1	1	1	1	1	1	1	1	1
##	2068	2111	2130	2136	2223	2228	2232	2238	2248	2310
##	1	1	1	1	1	1	1	1	1	1
##	2320	2351	2356	2396	2416	2420	2448	2457	2489	2516
##	1	1	1	1	1	1	2	1	1	1
##	2550	2551	2595	2615	2616	2640	2661	2704	2773	2778
##	1	1	1	1	1	1	1	1	1	1
##	2780	2827	2836	2837	2847	2861	2864	2879	2880	2881
##	1	1	1	1	1	1	1	1	1	1
##	2886	2897	2904	2926	2943	2962	2963	2973	2983	3009
##	1	1	1	2	1	1	1	1	1	2
##	3020	3040	3046	3056	3058	3060	3068	3104	3112	3168
##	1	1	1	1	1	1	1	1	1	1
##	3169	3198	3209	3216	3219	3231	3234	3260	3265	3275
##	1	1	1	1	1	1	1	1	1	1
##	3279	3283	3285	3295	3304	3325	3330	3354	3368	3384
##	1	1	1	1	1	1	1	1	1	1
##	3394	3397	3399	3415	3421	3424	3436	3439	3446	3453
##	1	1	1	2	3	1	1	1	1	1
##	3458	3464	3468	3482	3488	3498	3507	3511	3528	3529
##	1	1	1	1	1	1	1	1	1	1
##	3538	3544	3549	3565	3571	3575	3576	3586	3594	3609
##	1	1	1	1	1	1	1	1	1	1
##	3611	3633	3654	3678	3700	3709	3710	3712	3718	3724

##	1	2	1	1	1	1	1	1	1	1
##	3725	3736	3744	3749	3755	3780	3793	3797	3862	3903
##	1	1	1	1	1	1	1	1	1	1
##	3917	3930	3936	3945	3958	3969	3970	3988	3989	3990
##	1	1	2	1	1	1	1	1	1	1
##	4000	4005	4010	4012	4022	4026	4028	4037	4051	4056
##	1	1	1	1	1	1	1	1	1	1
##	4058	4063	4070	4093	4095	4111	4118	4125	4132	4136
##	2	1	1	1	1	1	1	1	1	1
##	4142	4149	4159	4170	4187	4208	4246	4260	4264	4268
##	1	1	1	1	1	1	1	1	1	1
##	4272	4302	4306	4317	4318	4319	4327	4328	4339	4343
##	1	1	1	1	1	1	1	1	1	2
##	4350	4352	4356	4369	4372	4378	4388	4394	4395	4399
##	1	1	1	1	1	1	1	1	1	2
##	4418	4430	4436	4444	4452	4457	4463	4473	4492	4501
##	1	1	1	1	1	1	1	1	1	1
##	4510	4530	4532	4533	4548	4558	4569	4584	4603	4605
##	2	1	1	1	1	1	1	1	2	1
##	4621	4655	4670	4682	4690	4694	4696	4703	4785	4798
##	1	1	1	1	1	1	1	1	1	1
##	4801	4804	4805	4871	4878	4883	4909	4932	4937	4939
##	1	1	1	1	1	1	1	1	1	1
##	4966	4979	5022	5034	5038	5039	5042	5073	5074	5081
##	1	2	1	1	1	1	1	1	1	1
##	5084	5102	5110	5114	5115	5121	5126	5127	5206	5216
##	1	1	1	1	1	2	1	1	1	1
##	5225	5228	5258	5265	5274	5276	5284	5308	5310	5328
##	1	1	1	1	1	1	1	1	1	1
##	5373	5416	5467	5474	5502	5507	5514	5515	5518	5541
##	1	1	1	1	1	1	1	1	1	1
##	5547	5573	5591	5644	5649	5674	5698	5699	5712	5725
##	1	1	1	1	1	1	1	1	1	1
##	5741	5753	5755	5756	5759	5761	5762	5784	5804	5830
##	1	1	1	1	1	1	1	1	1	1
##	5865	5867	5869	5912	5964	5975	5982	5992	6034	6035
##	1	1	1	1	1	1	1	1	1	1
##	6051	6074	6114	6177	6196	6197	6203	6210	6219	6224
##	1	1	1	1	1	1	1	1	1	1
##	6225	6291	6295	6296	6315	6320	6324	6332	6367	6381
##	1	1	1	2	1	1	1	1	1	1
##	6396	6400	6406	6408	6412	6417	6423	6442	6445	6456
##	1	1	1	1	1	1	1	1	1	1
##	6457	6459	6468	6492	6517	6591	6618	6628	6652	6672
##	1	1	1	1	1	1	1	1	1	1
##	6716	6719	6730	6751	6765	6945	6972	6976	7021	7047
##	1	1	1	1	1	1	1	1	1	1
##	7101	7192	7206	7223	7229	7232	7262	7280	7283	7287
##	1	1	1	1	1	1	1	1	1	1
##	7292	7296	7298	7304	7329	7333	7361	7403	7412	7430
##	1	1	1	1	1	1	1	1	1	1
##	7441	7464	7506	7524	7541	7566	7571	7590	7618	7642
##	1	1	1	1	2	1	1	1	1	1
##	7664	7675	7781	7804	7821	7835	7845	7849	7867	7871

##	1	1	1	1	1	1	1	1	1	1
##	7873	7878	7884	7898	7946	7961	7966	8068	8103	8105
##	1	1	1	1	1	1	1	1	1	1
##	8129	8143	8164	8174	8180	8192	8195	8220	8232	8258
##	1	1	1	1	1	1	1	1	1	1
##	8273	8293	8298	8320	8337	8343	8370	8427	8440	8452
##	1	1	1	1	1	1	1	1	1	1
##	8508	8555	8573	8589	8598	8610	8631	8632	8641	8656
##	1	1	1	1	1	1	1	2	1	1
##	8665	8670	8691	8698	8703	8707	8759	8814	8821	8850
##	1	1	1	1	1	1	1	1	1	1
##	8853	8882	8886	8911	8942	8943	8961	8974	9004	9058
##	1	1	1	1	1	1	1	1	1	1
##	9061	9108	9109	9132	9135	9161	9183	9195	9240	9250
##	1	1	1	2	1	1	1	1	1	1
##	9265	9303	9354	9357	9394	9431	9446	9456	9457	9461
##	1	1	1	1	1	1	1	1	1	1
##	9479	9487	9488	9502	9508	9555	9576	9586	9660	9675
##	1	1	1	1	1	1	1	1	1	1
##	9755	9779	9797	9834	9841	9847	9910	9947	9955	9961
##	1	1	1	1	1	1	1	1	1	1
##	9968	10037	10088	10207	10249	10261	10264	10294	10310	10385
##	1	1	1	1	1	1	1	1	1	1
##	10449	10461	10477	10492	10549	10628	10630	10718	10831	10859
##	1	1	1	1	1	1	1	1	1	1
##	10866	10954	11000	11003	11067	11092	11100	11115	11117	11133
##	1	1	1	1	1	1	1	1	1	1
##	11337	11451	11472	11504	11543	11568	11636	11639	11752	11763
##	1	1	1	1	1	1	1	1	1	1
##	11772	11813	11831	11832	11883	11887	11898	11912	12036	12085
##	1	1	1	1	1	1	1	1	1	1
##	12098	12103	12145	12157	12188	12200	12244	12255	12259	12262
##	1	1	1	1	1	1	1	1	1	1
##	12269	12274	12304	12312	12324	12336	12348	12401	12404	12414
##	1	1	1	1	1	1	1	1	1	1
##	12461	12490	12511	12535	12570	12599	12663	12739	12772	12807
##	1	1	1	1	1	1	1	1	1	1
##	12881	12902	12906	12915	12975	12991	13048	13152	13173	13193
##	1	1	1	1	1	1	1	1	1	1
##	13203	13244	13289	13302	13327	13423	13430	13488	13490	13504
##	1	1	1	1	1	1	1	1	1	1
##	13510	13552	13570	13582	13601	13669	13675	13677	13681	13731
##	1	1	1	1	1	1	1	1	1	1
##	13797	13885	13912	14029	14075	14077	14133	14255	14314	14356
##	1	1	1	1	1	1	1	1	1	1
##	14358	14377	14451	14490	14621	14666	14688	14793	14890	14968
##	1	1	1	1	1	1	1	1	1	1
##	15178	15201	15240	15293	15295	15353	15361	15416	15626	15660
##	1	1	1	1	1	1	1	1	1	1
##	15719	15721	15763	15841	15962	16155	16366	16538	16680	16729
##	1	1	1	1	1	1	1	1	1	1
##	17386	17423	17502	17520	17589	17778	17790	17849	17865	17931
##	1	1	1	1	1	1	1	1	1	1
##	17945	18110	18147	18160	18167	18193	18258	18355	18526	18572

##	1	1	1	1	1	1	1	1	1	1
##	18574	18608	18719	18795	18821	18851	18864	18921	18938	18945
##	1	1	1	1	1	1	1	1	1	1
##	19013	19081	19178	19253	19416	19679	19779	19839	19918	20087
##	1	1	1	1	1	1	1	1	1	1
##	20114	20188	20331	20377	20406	20545	20652	20670	20708	20744
##	1	1	1	1	1	1	1	1	1	1
##	20773	20810	20824	20853	21108	21238	21345	21399	21457	21458
##	1	1	1	1	1	1	1	1	1	1
##	21591	21614	21708	21724	21840	21874	21882	21900	21962	22159
##	1	1	1	1	1	1	1	1	1	1
##	22248	22315	22316	22332	22338	22412	22754	22848	22902	23619
##	1	1	1	1	1	1	1	1	1	1
##	23688	23736	23743	24226	24232	24301	24583	25538	26276	26322
##	1	1	1	1	1	1	1	1	1	1
##	26668	28686	28801	29027	29038	29170	29492	29500	29625	29803
##	1	1	1	1	1	1	1	1	1	1
##	30037	30327	30486	30618	30641	31223	31313	31515	31748	31815
##	1	1	1	1	1	1	1	1	1	1
##	32182	32543	32583	32601	32681	32791	32999	33078	33233	33353
##	1	1	1	1	1	1	1	1	1	1
##	33785	33841	34020	34733	34762	34833	34834	34933	35049	35069
##	1	1	1	1	1	1	1	1	1	1
##	35260	35366	35592	35708	35762	35780	36100	36267	36324	36566
##	1	1	1	1	1	1	1	1	1	1
##	36743	36940	37010	37134	37274	37661	37769	37825	37929	38008
##	1	1	1	1	1	1	1	1	1	1
##	38032	38171	38261	38500	38584	38624	38700	38725	38766	38776
##	1	1	1	1	1	1	1	1	1	1
##	38817	38966	39090	39464	39501	39777	39822	39839	39967	40008
##	1	1	1	1	1	1	1	1	1	1
##	40065	40105	40124	40133	40233	40298	40315	40333	40445	40457
##	1	1	1	1	2	1	1	1	1	1
##	40496	40634	40638	40667	40672	40674	40899	40910	40947	41051
##	1	1	1	1	1	1	1	1	1	1
##	41161	41197	41268	41420	41758	41789	41963	41993	42054	42057
##	1	1	1	1	1	1	1	1	1	1
##	42233	42251	42302	42314	42386	42629	42732	42898	42919	42934
##	1	1	1	1	1	1	1	1	1	1
##	43123	43126	43133	43166	43200	43325	43700	43729	44000	44002
##	1	1	1	1	1	1	1	1	1	1
##	44077	44142	44167	44333	44347	44513	44555	44816	44867	45013
##	1	1	1	1	1	1	1	1	1	1
##	45065	45166	45200	45232	45596	45708	46036	46123	46167	46337
##	1	1	1	1	1	1	1	1	1	1
##	46500	46847	47167	47300	47792	47825	48068	48271	48566	48800
##	1	1	1	1	1	1	1	1	1	2
##	49100	49242	49333	49541	49892	50183	50260	50778	50887	51097
##	1	1	1	1	1	1	1	1	1	1
##	51133	51153	51241	51666	52033	52228	52295	52362	52400	52448
##	1	1	1	1	1	1	1	1	1	1
##	52617	52646	52828	52966	53100	53720	53821	54414	54511	54760
##	1	1	1	1	1	1	1	1	1	1
##	54807	55085	55110	55586	55681	55725	55975	56194	56471	56518

```
##      1      1      1      1      1      1      1      1      1      1
## 57049 57329 57455 57748 57762 58108 58326 58379 58595 58657
##      1      1      1      1      1      1      1      1      1      1
## 59144 59333 60009 60233 61033 61415 61730 61853 62703 63233
##      1      1      1      1      1      1      1      1      1      1
## 63736 63838 63866 63906 64032 64145 64989 64992 65240 65513
##      1      1      1      1      1      1      1      1      1      1
## 65566 65972 66281 66887 67536 69019 69169 69506 69758 70682
##      1      1      1      1      1      1      1      1      1      1
## 70762 71004 71403 71459 71792 75838 80045 80318 80976 81804
##      1      1      1      1      1      1      1      1      1      1
## 83201 83991 84221 85025 85246 85586 86349 87687 88334 88437
##      1      1      1      1      1      1      1      1      1      1
## 90284 90636 91537 91591 91730 93445 95419 96582 98969 99188
##      1      1      1      1      1      1      1      1      1      1
## 99546 104373 108915 113069 114308 114365 120468 123447 123815 124023
##      1      1      1      1      1      1      1      1      1      1
## 126029 128499 129772 130999 133558 133775 135616 135664 137536 141259
##      1      1      1      1      1      1      1      1      1      1
## 144827 149997 153128 165060 168751 170580 172792 175033 176135 176665
##      1      1      1      1      1      1      1      1      1      1
## 177665 177731 178600 178833 179162 179267 180633 181249 184384 188166
##      1      1      1      1      1      1      1      1      1      1
## 188783 195669 199920 201784 204129 206159 207789 209367 211162 212132
##      1      1      1      1      1      1      1      1      1      1
## 212337 230621 233102 234918 235889 239642 240301 241512 242595 303645
##      1      1      1      1      1      1      1      1      1      1
## 311223 317390 325407 332695 338129 344755 351195 357401 371450 379930
##      1      1      1      1      1      1      1      1      1      1
## 1599619 1612767 1642461 1663238 1689621 1775754 1791929 1823710 1847622 1878404
##      1      1      1      1      1      1      1      1      1      1
##      <NA>
##      419
```

- Observamos que tenemos **419 valores perdidos**. Guardamos en la variable **idx** los índices de los registros con valores **NA** de la variable **Value**.

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

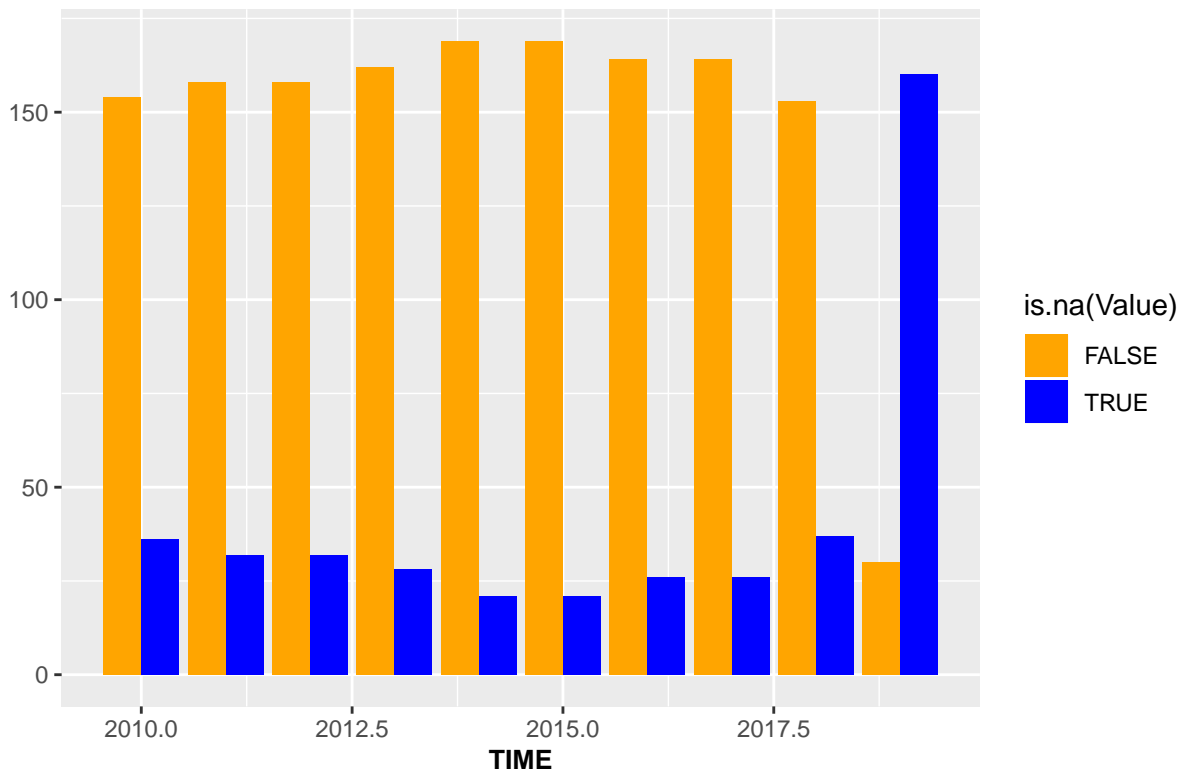
```
## [1] 419
```

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

```
library(ggplot2)
library(scales)
g = ggplot(medicos_p, 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(medicos_p, variable=c("Value"),k=3)
```

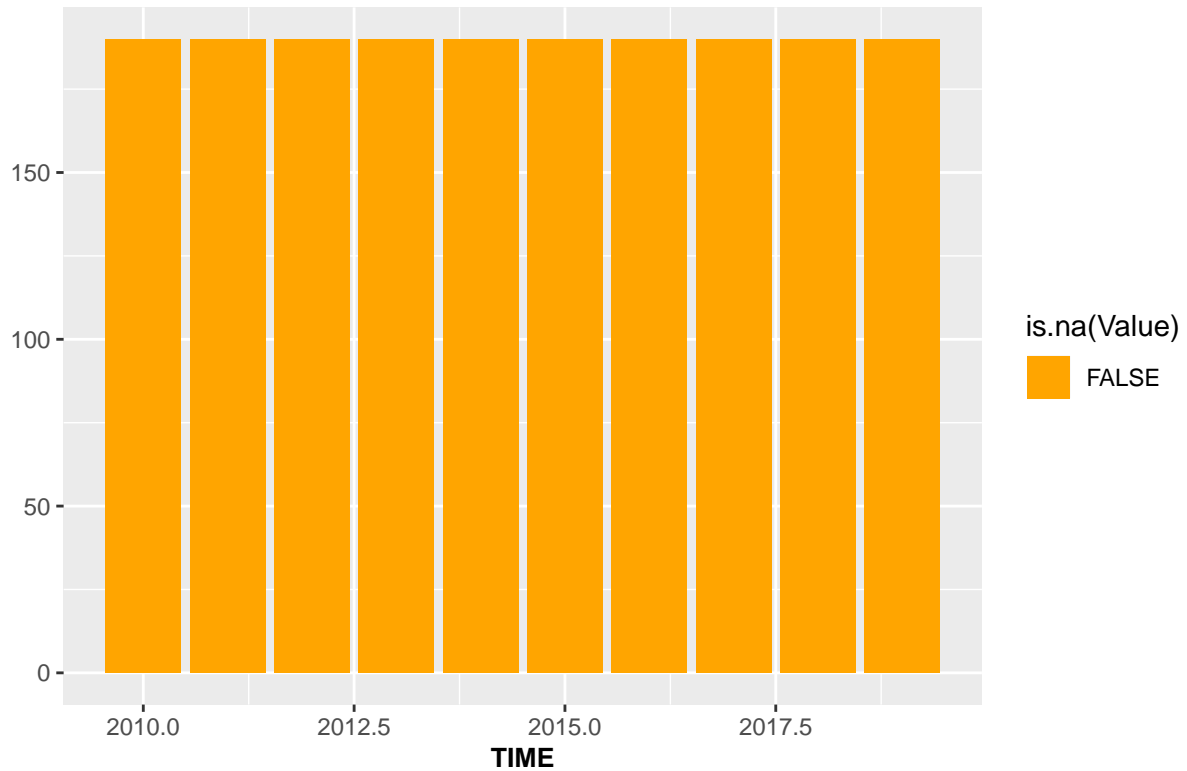
```
medicos_p<-output
```

- Comprobamos que no tenemos valores nulos después de la imputación

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

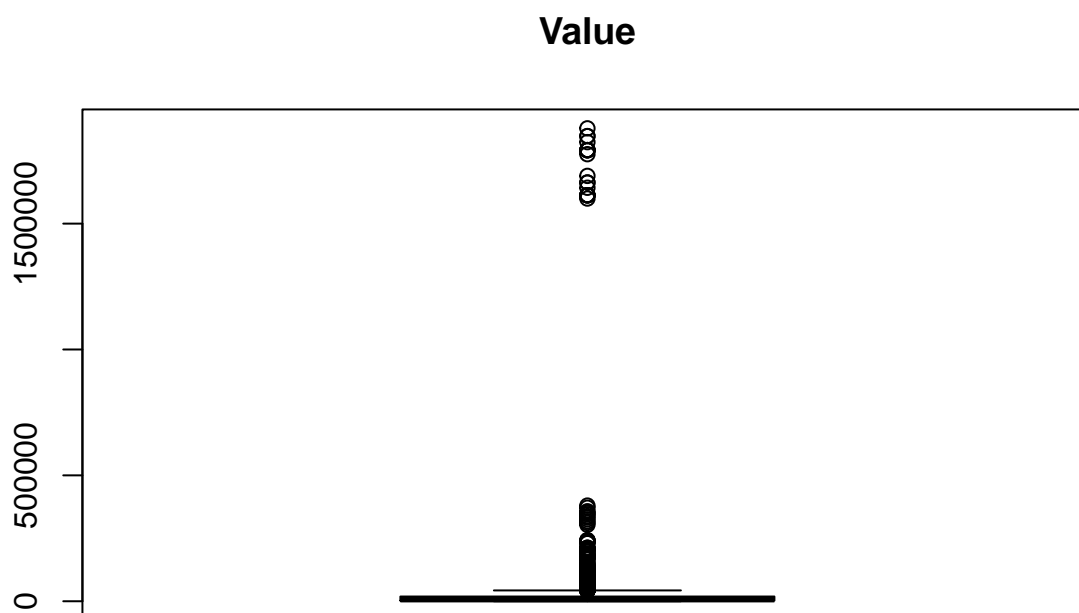
```
g+geom_bar(position="dodge") + scale_fill_manual(values = alpha(c("orange", "blue"), 1)) +  
theme(axis.title.x = element_text(face="bold", size=10))
```

Valores Nulos



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

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



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

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

```
##
## 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019
## 190 190 190 190 190 190 190 190 190 190
```

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

```
##
## Austria
## 50
## Belgium
## 50
## Bulgaria
## 50
## Croatia
## 50
## Cyprus
## 50
## Czechia
## 50
## Denmark
## 50
## Estonia
## 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
##	Montenegro
##	50
##	Netherlands
##	50
##	North Macedonia
##	50
##	Norway
##	50
##	Poland
##	50
##	Portugal
##	50
##	Romania
##	50
##	Serbia
##	50
##	Slovakia
##	50
##	Slovenia
##	50
##	Spain
##	50
##	Sweden
##	50
##	Switzerland

```
##
##
##
##
##
```

	50
Turkey	
	50
United Kingdom	
	50

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

```
##
## Number
## 1900
```

```
table(medicos_p$AGE, useNA = "ifany")
```

```
##
## From 35 to 44 years From 45 to 54 years From 55 to 64 years Less than 35 years
## 380 380 380 380
## Total
## 380
```

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(medicos_p, file="Medicos_Por_Paises_clean.csv", row.names = FALSE)
```