

Ejemplo de R Markdown: *El voto autonómico en la comarca de la Safor (VALENCIA)*

Tomàs Ferrandis Moscardó

2024-01-26

Contents

1. INTRODUCCIÓN	1
1.1 Inclusión de los paquetes (y sus librerías) que vamos a necesitar	2
2. OBTENCIÓN Y TRATAMIENTO DE LOS DATOS	2
2.1 Resultados de todas las elecciones y convocatorias a nivel de agregación comarcal. Caso de La Safor.	2
2.2 Filtramos por elecciones autonómicas.	2
Filtraremos por el campo correspondiente	3
2.3 Vemos las primeras lineas de ejemplo	3
2.4 Tratamiento de los NA	3
2.5 Creamos un vector con las candidaturas.	3
3. CÁLCULO DE VALORES DE MEDIDAS de TENDENCIA CENTRAL y DISPERSIÓN	4
3.1 Creamos un data frame. cada vector anterior será una columna	4
3.2 Añadimos al final del data frame una columna con el nombre del “cálculo”	4
3.3 Eliminamos la columna de valores de “resto de partidos”	4
3.3 Imprimimos en Markdown una tabla con los resultados	4
4. GRÁFICOS Y TABLAS	5
5. GUARDAR DATA FRAME EN FICHEROS DE DIFERENTES FORMATOS	12
5.1 Guardar datos en formato .xlsx (MS Excel)	12
5.2 Añadiendo una hoja en un ficher xlsx existente	13
5.3 Guardar datos en formato .dta / Stata.	13
5.4 Guardar datos en formato .csv	13
5.5 Guardar datos en formato .sav / SPSS	13
5.6 Guardar y abrir datos en formato RStudio.	13
5.7 Lectura de datos RData	13
5.8 RDS - guarda un solo archivo de datos	13
5.9 Lectura de datos RDS	13

1. INTRODUCCIÓN

A continuación veremos un sencillo ejemplo de uso de **R-Markdown** en el que, a partir de un fichero de tipo *csv* (*SAFOR.csv*) crearemos y manipularemos objetos *data.frame*. Obtendremos algunas medidas de *tendencia central y dispersión* que representaremos en tablas y gráficos.

Después guardaremos los datos del `data_frame` de resultados en ficheros de diversos tipos: *csv*, *xlsx*, *dta*, *RData*, *sav*, *rds*)

1.1 Inclusión de los paquetes (y sus librerías) que vamos a necesitar

Las librerías se cargan solo si previamente se ha instalado el paquete que las contiene. En nuestros *chunks* (pedazos de código R), comprobaremos si están disponibles. Solo en caso contrario instalaremos su paquete correspondiente.

Para ello, primero se intenta cargar una librería mediante `require()` . Si da error, `require` devuelve un valor `FALSE` (`!require()`), entonces se ejecuta la instalación del paquete (que carga también la librería).

```
# Requisitos previos. Paquetes y librerías
if (!require("dplyr")) {install.packages("dplyr")}
if (!require("stringr")) {install.packages("stringr")}
if (!require("curl")) {install.packages("curl")}
if (!require("rsconnect")){install.packages("rsconnect")}
if (!require("kableExtra")){install.packages("kableExtra")}
if (!require("tidyverse")){install.packages("tidyverse")}
library(data.table)
library(knitr)
library(stringr)
library(dplyr)
library(ggplot2)
```

2. OBTENCIÓN Y TRATAMIENTO DE LOS DATOS

2.1 Resultados de todas las elecciones y convocatorias a nivel de agregación comarcal. Caso de La Safor.

A partir de la base de datos de ARGOS descargamos el resumen de resultados electorales en las distintas convocatorias y procesos electorales en un fichero *CSV*.

```
df1<-read.csv("CSV/SAFOR.csv",header=TRUE, sep=",",quote="\\"", dec=",",fill=TRUE,
             comment.char = "")
# View(df1) # Para ver en consola
```

Veamos como ejemplo las primeras líneas...

```
head(df1)
```

##	Elecció	Cens	A.Cand.	PP	PSPV	COMPROMÍS	VOX	PODEMOS	Cs	EUPV	RESTA
## 1	G-2023	124383	90477	29545	31386	15855	12462	NA	NA	NA	1229
## 2	A-2023	124367	86496	28245	25559	19953	8282	2234	591	NA	1859
## 3	L-2023	127096	88398	27174	28953	15703	3126	120	364	1143	11795
## 4	G-2019	121791	86427	19634	23322	10644	14008	11334	5559	NA	1926
## 5	E-2019	123535	84970	19128	25797	15919	4818	6621	8761	NA	3926
## 6	L-2019	124552	86268	23328	26289	19803	1910	1155	4541	1197	8044

2.2 Filtramos por elecciones autonómicas.

Asumimos como **universo o población** el conjunto de todas las todas elecciones autonómicas.

Filtraremos por el campo correspondiente

```
df_autonomicas<-df1 %>% filter(str_detect(df1$Elecció,"A-"))
```

2.3 Vemos las primeras lineas de ejemplo

```
head(df_autonomicas)
```

```
## Elecció Cens A.Cand. PP PSPV COMPROMÍS VOX PODEMOS Cs EUPV RESTA
## 1 A-2023 124367 86496 28245 25559 19953 8282 2234 591 NA 1859
## 2 A-2019 121181 89534 17325 19867 23504 7666 5917 12394 NA 2861
## 3 A-2015 121362 90086 26377 18226 24789 207 6845 7668 3129 2845
## 4 A-2011 120877 92475 47413 25159 12366 NA NA NA 3323 4214
## 5 A-2007 119109 90705 46558 29321 11050 NA NA NA 3776
## 6 A-2003 116160 90337 41220 31192 10891 NA NA NA 3145 3889
```

2.4 Tratamiento de los NA

En nuestro caso los NA se deben a la ausencia de resultados por no haberse presentado la fuerza política en una convocatoria o por haberse integrado en una coalición electoral. Entendemos que el valor que debe sustituir NA para poder realizar cálculos estadísticos de forma correcta y sin errores de computación es el 0. Podríamos recorrer solo las columnas de partidos pero lo simplificamos y aplicamos la sustitución NA -> 0 en todo el data frame.

```
df_autonomicas[is.na(df_autonomicas)]<-0
```

Mostramos los datos en formato de tabla de RMarkdown

```
knitr::kable(df_autonomicas, caption = "ELECCIONES AUTONÓMICAS EN LA SAFOR")
```

Table 1: ELECCIONES AUTONÓMICAS EN LA SAFOR

Elecció	Cens	A.Cand.	PP	PSPV	COMPROMÍS	VOX	PODEMOS	Cs	EUPV	RESTA
A-2023	124367	86496	28245	25559	19953	8282	2234	591	0	1859
A-2019	121181	89534	17325	19867	23504	7666	5917	12394	0	2861
A-2015	121362	90086	26377	18226	24789	207	6845	7668	3129	2845
A-2011	120877	92475	47413	25159	12366	0	0	0	3323	4214
A-2007	119109	90705	46558	29321	11050	0	0	0	0	3776
A-2003	116160	90337	41220	31192	10891	0	0	0	3145	3889
A-1999	117654	87007	40176	27696	9066	0	0	0	3782	6287
A-1995	108888	86600	32374	29415	7600	0	0	0	7503	9708
A-1991	102923	78619	19181	30884	7998	0	0	0	5313	15243
A-1987	97022	75807	17142	28744	0	0	0	0	9042	20879
A-1983	94401	72681	22015	30781	5823	0	0	0	5815	8247

2.5 Creamos un vector con las candidaturas.

```
# Recogemos los nombres de columnas a partir de la 4ª
partits<-names(df1[4:length(df1)])

# Para nuestro caso podríamos anotarlo directamente
# partits<-c("PP", "PSPV", "COMPROMÍS", "VOX", "PODEMOS", "Cs", "EUPV", "RESTA")
```

Los visualizamos en formato de tabla de RMarkdown

```
knitr::kable(partits, caption = "CANDIDATURAS")
```

Table 2: CANDIDATURAS

x
PP
PSPV
COMPROMÍS
VOX
PODEMOS
Cs
EUPV
RESTA

3. CÁLCULO DE VALORES DE MEDIDAS de TENDENCIA CENTRAL y DISPERSIÓN

1.- Media o promedio 2.- Máximos 3.- Mínimos 4.- Rango de variación

```
mitjana<-summarise_at(df_autonomicas,partits,mean)
maxims<-summarise_at(df_autonomicas,partits,max)
minims<-summarise_at(df_autonomicas,partits,~min(. != 0)) #Descartamos los 0
rangodevariacion<-maxims - minims
```

3.1 Creamos un data frame. cada vector anterior será una columna

```
df_resultats<-rbind(mitjana)
df_resultats<-rbind(df_resultats,maxims)
df_resultats<-rbind(df_resultats,minims)
df_resultats<-rbind(df_resultats,rangodevariacion)
#Redondeamos los valores a enteros ( son votos)
df_resultats<-df_resultats %>% mutate_at(partits, as.integer)
```

3.2 Añadimos al final del data frame una columna con el nombre del “cálculo”

```
df_resultats<-cbind(df_resultats,ESTADÍSTICA=c("MEDIA","MÁXIMO","MÍNIMO","RANGO"))
# la resituamos en la posición 1 ( estática)
df_resultats<-df_resultats %>% select("ESTADÍSTICA",everything())
```

3.3 Eliminamos la columna de valores de “resto de partidos”

```
df_resultats <- df_resultats[,-length(df_resultats)]
```

3.3 Imprimimos en Markdown una tabla con los resultados

```
knitr::kable(df_resultats, caption = "ESTADÍSTICA SAFOR")
```

Table 3: ESTADÍSTICA SAFOR

ESTADÍSTICA	PP	PSPV	COMPROMÍS	VOX	PODEMOS	Cs	EUPV
MEDIA	30729	26985	12094	1468	1363	1877	3732
MÁXIMO	47413	31192	24789	8282	6845	12394	9042
MÍNIMO	17142	18226	5823	207	2234	591	3129
RANGO	30271	12966	18966	8075	4611	11803	5913

4. GRÁFICOS Y TABLAS

Vemos a continuación los gráficos y tablas de daraso a nivel de agregación de candidatura. Creamos por cada partido (columna del `df_resultats`):

- Un `data_frame`
- Un gráfico
- Una tabla

```
# Recorremos las columnas del data frame correspondientes a candidaturas ( >2)
# Creamos un data frame en cada iteración
for ( i in 2:length(df_resultats)){
  df_estPartit<- data.frame(
    ESTADÍSTICA=df_resultats$ESTADÍSTICA,
    VALORS=df_resultats[,i],
    PARTIT=colnames(df_resultats[i])
  )

  # EN UN GRAFICO
  barplot(df_estPartit$VALORS,
    main=paste("Valores estadísticos de",df_estPartit$PARTIT[1]),
    names.arg=df_estPartit$ESTADÍSTICA,
    col=c("#5fe10b","red","#48b7f7","#ede12e"),
    ylab="VOTOS")

  # EN UNA TABLA
  #Quitamos la columna que repite el nombre ( estético), lo guardamos antes
  candidatura<-df_estPartit$PARTIT[1]
  df_estPartit <- df_estPartit[, !(names(df_estPartit) %in% "PARTIT")]
  knitr::kable(df_estPartit, caption = paste( "ESTADÍSTICA SAFOR",
    candidatura )) %>% print()
}
```

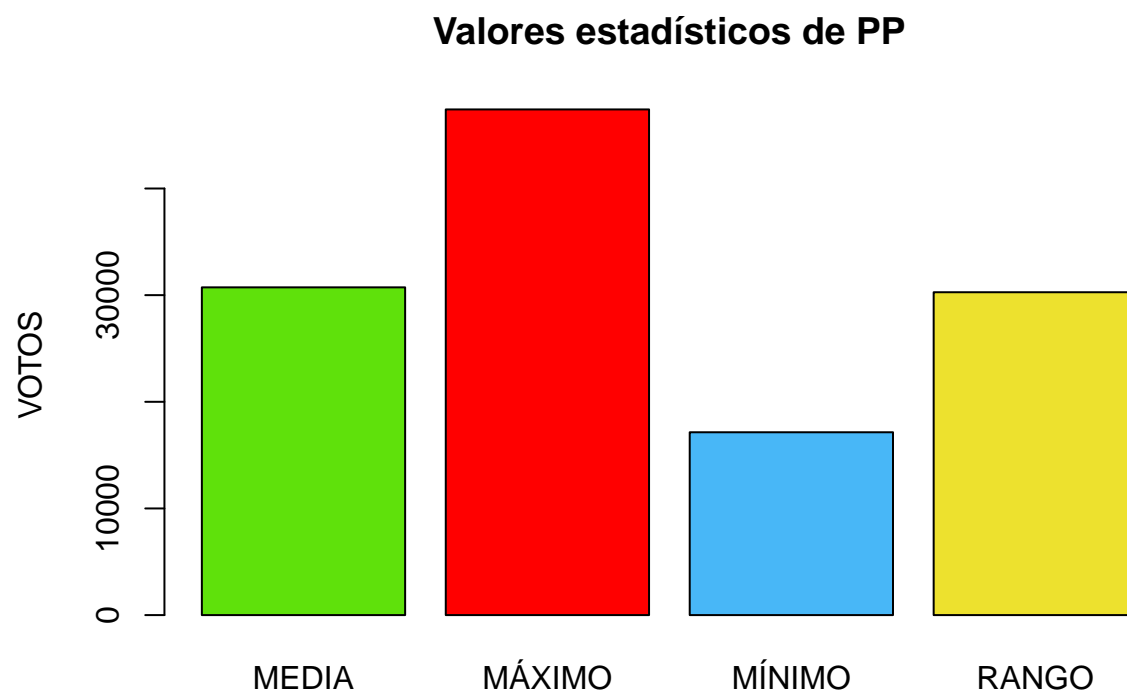


Table 4: ESTADÍSTICA SAFOR PP

ESTADÍSTICA	VALORS
MEDIA	30729
MÁXIMO	47413
MÍNIMO	17142
RANGO	30271

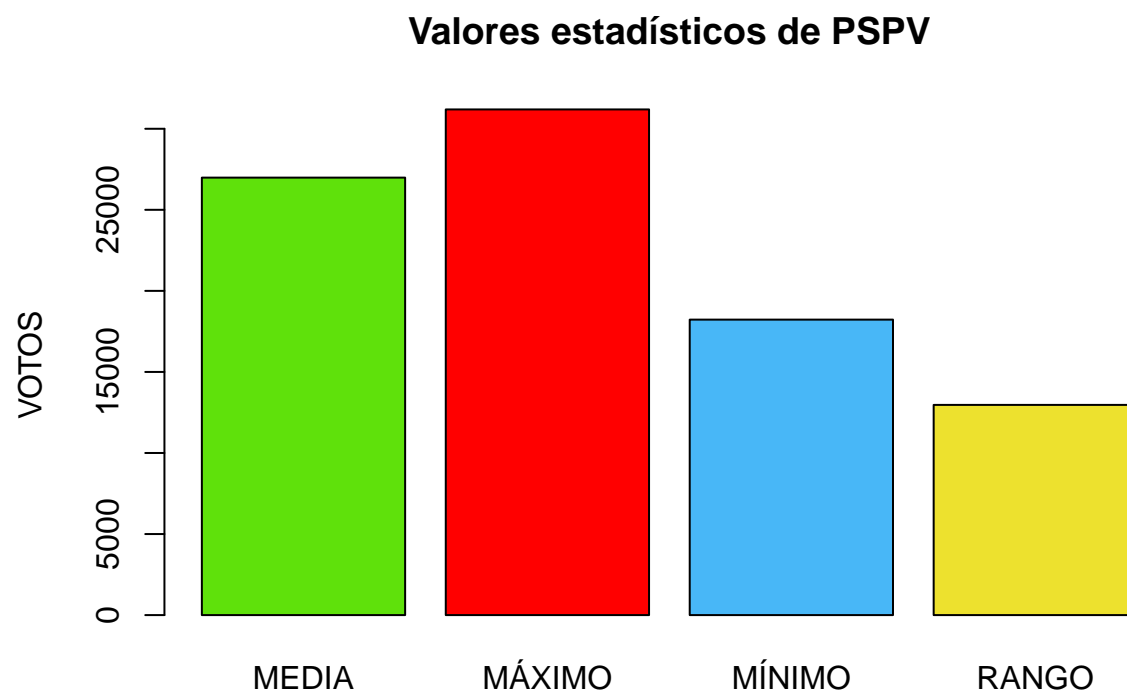


Table 5: ESTADÍSTICA SAFOR PSPV

ESTADÍSTICA	VALORS
MEDIA	26985
MÁXIMO	31192
MÍNIMO	18226
RANGO	12966

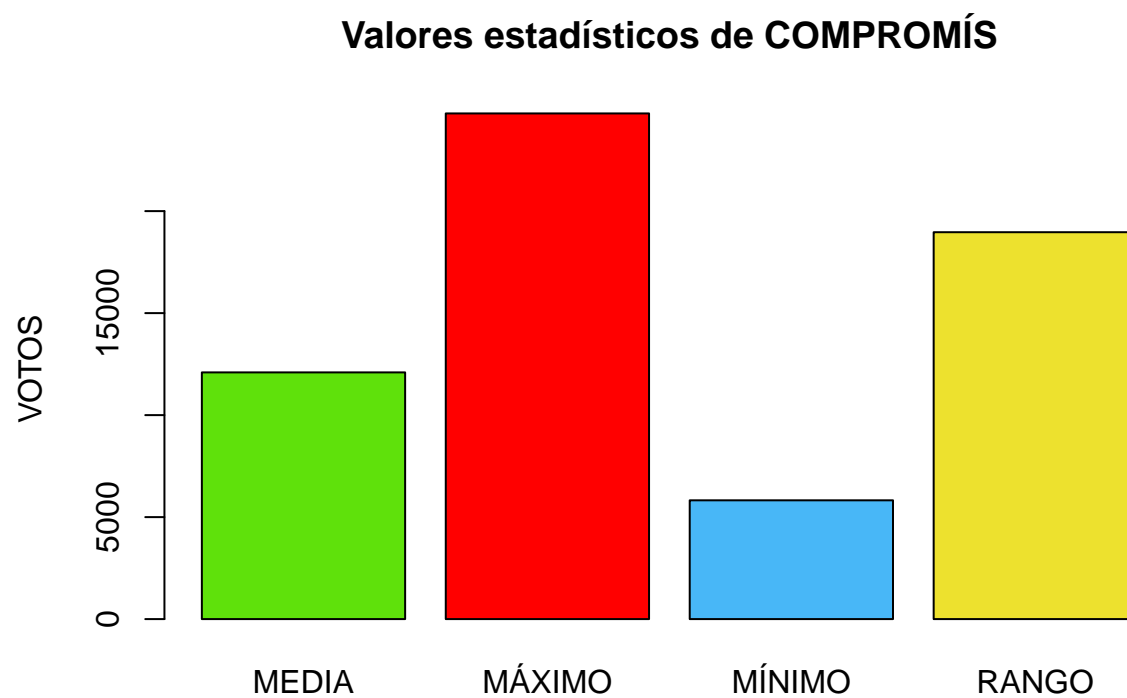


Table 6: ESTADÍSTICA SAFOR COMPROMÍS

ESTADÍSTICA	VALORS
MEDIA	12094
MÁXIMO	24789
MÍNIMO	5823
RANGO	18966

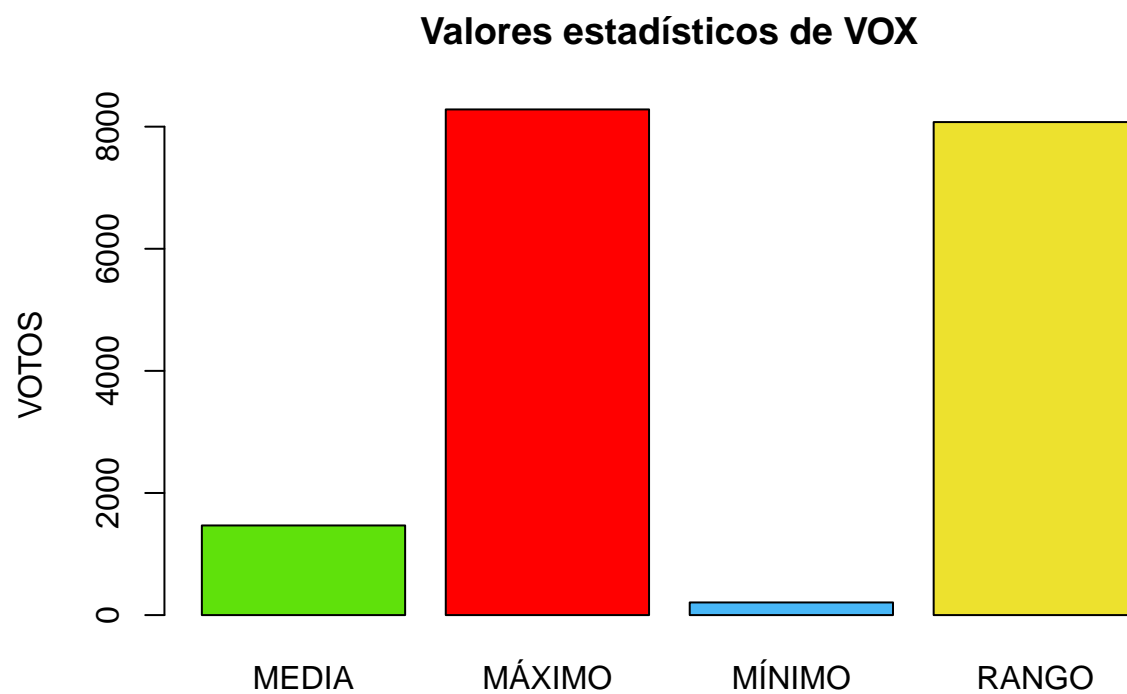


Table 7: ESTADÍSTICA SAFOR VOX

ESTADÍSTICA	VALORS
MEDIA	1468
MÁXIMO	8282
MÍNIMO	207
RANGO	8075

Valores estadísticos de PODEMOS

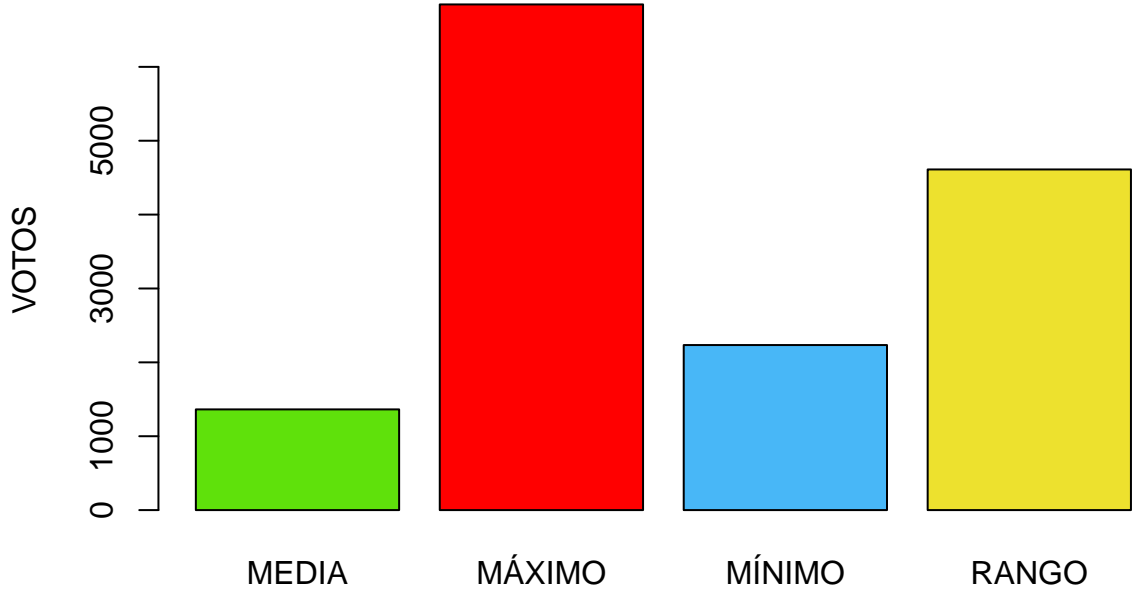


Table 8: ESTADÍSTICA SAFOR PODEMOS

ESTADÍSTICA	VALORS
MEDIA	1363
MÁXIMO	6845
MÍNIMO	2234
RANGO	4611

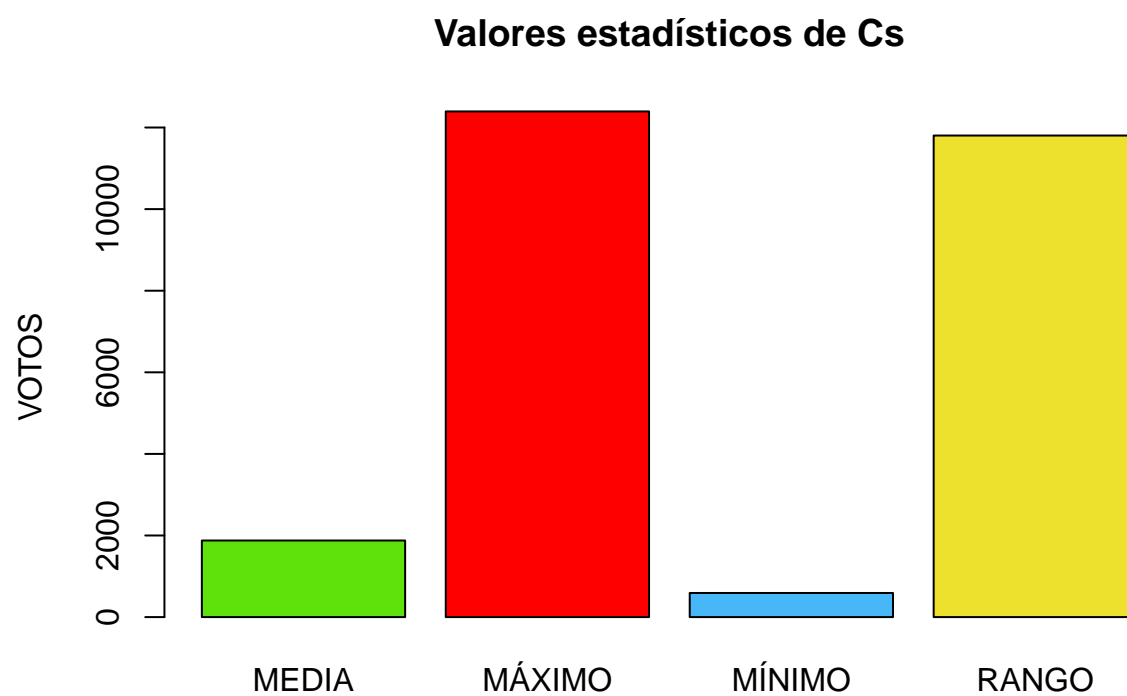


Table 9: ESTADÍSTICA SAFOR Cs

ESTADÍSTICA	VALORS
MEDIA	1877
MÁXIMO	12394
MÍNIMO	591
RANGO	11803

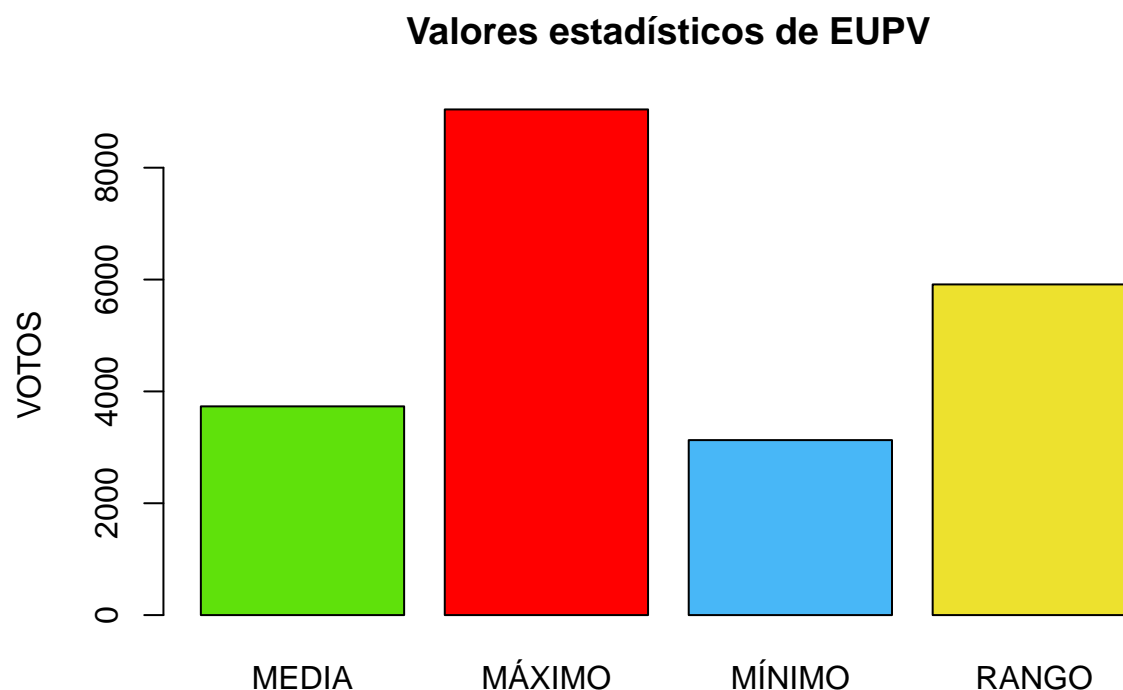


Table 10: ESTADÍSTICA SAFOR EUPV

ESTADÍSTICA	VALORS
MEDIA	3732
MÁXIMO	9042
MÍNIMO	3129
RANGO	5913

5. GUARDAR DATA FRAME EN FICHEROS DE DIFERENTES FORMATOS

5.1 Guardar datos en formato .xlsx (MS Excel)

Se instala el paquete solo si hace falta

```
if (!require("xlsx")) {install.packages("xlsx")}
```

Loading required package: xlsx

Se creará un fichero de tipo MS Excel: *resultats.xlsx*

```
### Creando un nuevo fichero xlsx
```

```
write.xlsx(df_resultats, file = "resultats.xlsx", sheetName = "resultats", append = FALSE)
```

5.2 Añadiendo una hoja en un fichero xlsx existente

Aquí añadiremos una página más a un fichero MS Excel existente (`append=TRUE`). Si *ficheroAnterior.xlsx* no existe dará error.

```
write.xlsx(df_resultats, file = "ficheroAnterior.xlsx", sheetName="resultats", append=TRUE)
```

5.3 Guardar datos en formato .dta / Stata.

```
if (!require("foreign")) {install.packages("foreign")}  
write.dta(df_resultats, "resultats.dta")
```

5.4 Guardar datos en formato .csv

```
if (!require("tidyverse")) {install.packages("tidyverse")}  
write_csv(df_resultats, "resultats.csv")
```

5.5 Guardar datos en formato .sav / SPSS

```
if (!require("haven")) {install.packages("haven")}  
write_sav(df_resultats, "resultats.sav")
```

5.6 Guardar y abrir datos en formato RStudio.

No es lo más usado, porque la costumbre es compartir los datos en formatos como .csv u otros que sean más comunes. No obstante, conviene conocer estos formatos, que se manejan muy fácilmente. No requieren librería, pues estos comandos trabajan con las funciones que trae **RStudio**.

RData - permite guardar más de una base de datos en un mismo archivo.

```
save(df_resultats, file="resultats.RData")
```

5.7 Lectura de datos RData

El parámetro *verbose=TRUE* es opcional. Sirve para que en la carga (lectura) nos muestre los datos (dataframe en este caso) existentes en el fichero de tipo *.Rdata*. en nuestro ejemplo vemos que contiene el dataframe: *df_resultats*

```
load("resultats.RData", verbose=TRUE)
```

```
## Loading objects:  
##   df_resultats
```

5.8 RDS - guarda un solo archivo de datos

```
saveRDS(df_resultats, "resultats.rds")
```

5.9 Lectura de datos RDS

```
df_resultats2 <- readRDS("resultats.rds")
```