Manual para la extracción de datos electorales usando R

Marco Ortiz Palanques

4/9/2020

Objetivo

El objetivo de este script es modelar los datos electorales del ministerio del interior para obtener una tabla con los resultados de un municipio en particular y que pueda ser útil para otros manejos como la graficación o comparación con otros procesos electorales en espacio y tiempo

Materiales

Librerías

Aparte del R base usamos las librerías de tidyverse readr, dplyr, y tidyr:

```
#Librerias
library(readr)
library(dplyr)
library(tidyr)
```

Archivos

Los archivos de los datos electorales están en la web del ministerio del interior en el área de descargas: http://www.infoelectoral.mir.es/infoelectoral/min/areaDescarga.html?method=inicio . Allí, en la parte inferior de la página, hay que buscar la pestaña "Extracción de datos" y pedir según el tipo de elección y fecha y luego Consultar. En nuestro caso escogimos: "Congreso" y "Noviembre 2019". Luego aparecerán tres opciones de archivos disponibles y se selecciona "10 de Noviembre de 2019 (Mesa)". Aquí se descargará un archivo .zip que habrá que descomprimir. El contenido de la carpeta son 10 archivos .DAT, uno .doc y uno .rtf. Los dos últimos contienen en diferentes formatos el manual para la comprensión de los archivos .DAT. De los archivos .DAT nos interesan los llamados: 03021911.DAT, 09021911.DAT y 10021911.DAT. Procedemos a descargarlos desde el directorio donde los archivamos al descomprimirlos.

```
#Tomar ficheros
X03021911 <- read_csv("C:/directorio/03021911.DAT",
                      col_names = FALSE, locale = locale(encoding = "ASCII"))
X09021911 <- read_csv("C:/directorio/09021911.DAT",
                      col_names = FALSE)
X10021911 <- read_csv("C:/directorio/10021911.DAT",
                      col_names = FALSE)
```

data.frame con las claves de los distritos del municipio

El nivel menor de datos electorales es la mesa, sigue la sección censal, el distrito censal Las diferentes secciones son organizadas en centros electorales (los centros educativos donde se vota). Los municipios, por su parte, pueden tener una división adicional que llamaremos genéricamente el distrito municipal(trataremos de idetificarlos completamente en cada caso para que no se confundan con los distritos censales). Los límites de los distritos municipales no necesarimente coinciden con los de las secciones. Sin embargo, una parte importante del análisis es proporcionar información sobre estas unidades administrativas, pues forman unidades sociales dentro de la comunidad. Por ello, para el caso que nos ocupa hemos preparado un fichero donde hemos ubicado las secciones electorales dentro de los distritos municipales, aunque en algunos caso hemos tenido que hacer decisiones de ubicación por la superposición de límites.
En nuestro caso lo trajimos de la base de datos usando la librería RODBC.

```r
library(RODBC)
dsn_driver <- "{XXX}"
dsn_database <- "XXX"
dsn_hostname <- "XXX"
dsn_port <- "XXX"
dsn_protocol <- "XXX"
dsn_uid <- "XXX"
dsn_pwd <- "XXX"
conn_path <- paste("DRIVER=",dsn_driver,
                   ";DATABASE=",dsn_database,
                   ";HOSTNAME=",dsn_hostname,
                   ";PORT=",dsn_port,
                   ";PROTOCOL=",dsn_protocol,
                   ";UID=",dsn_uid,
                   ";PWD=",dsn_pwd,sep="")
conn <- odbcDriverConnect(conn_path)
# Traer archivo de base de datos
DISTRITOS2019 <- sqlFetch(conn,"DISTRITOS2019")
#Al venir de la base de datos, la columna de ID (que está hecha de números) es cambiada a numérica.
#Por ello toca nuevamente volverla a caracter.
DISTRITOS2019$ID2019 <- as.character(DISTRITOS2019$ID2019)
```

Table 1: Tabla de distritos municipales

DISTRITO	PKB	ID2019
CENTRO	CE	102
CENTRO	CE	103
CENTRO	CE	104
CENTRO	CE	105
CENTRO	CE	106

La mejor forma de entender la tabla es leer las columnas de derecha a izquierda. En la columna ID2019 se encuentra el ID de cada sección electoral: el primer dígito es el número del distrito censal y los dos últimos dígitos corresponden a la sección censal. El archivo tienen una fila por cada sección electoral del municipio. En la columna central está la sigla del distrito municipal (no censal) y en la columna d ela izquierda el nombre del distrito municipal al que pertenece cada sección censal.

Preparación de los ficheros

Como podemos ver al consultar los archivos de los datos X03021911,X09021911,X10021911, la forma de los ficheros ha sido simplificada para el mejor almacenamiento y extracción de datos.

data.frame X03021911. Nombres de los partidos

Table 2: El data.frame con los datos de los partidos

X1
02201911000002AHORA CANARIAS AHORA CANARIAS: Alternativa Nacionalista Canaria (ANC) y Unidad del Pueblo 000002000002000002
02201911000003ANDECHA ANDECHA ASTUR 000003000003000003
02201911000005AUNACV AUNA COMUNITAT VALENCIANA 000005000005000005

El data frame X03021911 contiene los códigos, abreviaturas y nombres completos de las organizaciones electorales participantes en el proceso electoral. A partir de su data podemos separar estas informaciones en columnas. Para ello usamos la función substr, creando una columna nueva p.e. X03021911$partidos. Veamos el proceso en detalle:

1. A la columna que hemos creado le asignamos valores de la columna original X03021911$X1 mediante el comando substr.
2. Antes de seguir debemos ir al manual ficheros.doc y buscar las instrucciones de la tabla 3. Allí se indica que el código de los partidos ocupa desde el 9º caracter hasta el 14º.
3. El primer argumento de la función es la propia columna X03021911$X1 de donde extraemos los datos.

4. El segundo argumento es donde comenzamos a extraer los caracteres de esa columna. Para nuestro ejemplo, comenzamos a extraer del caracter 9.
5. El tercer argumento es hasta cuál caracter extraemos, en este caso el 14.

6. Seguimos el mismo procedimiento para extraer la abreviatura (nombreCorto) y el nombre completo (nombreLargo) del partido.

7. Finalmente, quitamos los espacios en blanco para disponer mejor de los nombreCorto cuando el futuro los convirtamos en cabeceras de columnas.

```
#Crear columnas en el fichero 03
X03021911$partidos <- substr(X03021911$X1,9,14)
X03021911$nombreCorto <- substr(X03021911$X1,15,64)
X03021911$nombreLargo <- substr(X03021911$X1,65,214)
X03021911$nombreCorto <- gsub(" ","",X03021911$nombreCorto)
X03021911$nombreCorto <- gsub("-","",X03021911$nombreCorto)
```

Table 3: El data.frame con los nombres de partido separados

partidos	nombreCorto	nombreLargo
000002	AHORACANARIAS	AHORA CANARIAS: Alternativa Nacionalista Canaria (ANC) y Unidad del Pueblo
000003	ANDECHA	ANDECHA ASTUR
000005	AUNACV	AUNA COMUNITAT VALENCIANA

data.frame X09021911. Datos generales de la mesa

Table 4: El data.frame con los totales de las mesas

X1
022019111010400302001
B000044600004460000000000000000001410000215000000100000000000270000000000000000S
022019111010400602001
A000084200008420000000000000000002800000440000000400000080000533000000000000000S
022019111010400604001
B000045400004540000000000000000001330000216000000300000020000264000000000000000S

La idea aquí, es la misma que en el data.frame anterior: obtener unas columnas con datos claros. En estos casos hemos obtenido los siguientes:

1. localidad: es el código del municipio y está compuesto por 7 dígitos. Los dos primeros señalan la comunidad autonómica, el 3º y el 4º la provincia y los tres últimos el municipio.

2. distrito: aíslamos el distrito censal del municipio.

3. seccion (sección: sin tilde para evitar conflictos): número de la sección censal.

4. mesa: letra correspondiente a la mesa de la sección censal.

5. censoINE: es el número de electores registrados para cada mesa. Hay también el registro de los votantes en el extranjero; pero en este caso lo hemos obviado.

6. blancos: número de votos blancos.

7. nulos: número de votos nulos.

8. validos (válidos: sin tilde para evitar conflictos): número de votos válidos.

9. Todas las columnas se crearon como "character", por lo que es necesario pasar las que contienen números a "numeric" (censoINE, blancos, nulos, validos).

10. Finalmente, creamos el ID2019Mesa, identificador para cada mesa, compuesto del número del distrito censal, el de la seccion y la letra de la mesa. Cuando se extraiga un municipio en particular se convertirá en su identificador único.

```
#Crear columnas en el fichero 09
X09021911$localidad <- substr(X09021911$X1,10,16)
X09021911$distrito <- substr(X09021911$X1,17,18)
X09021911$seccion <- substr(X09021911$X1,19,22)
X09021911$mesa <- substr(X09021911$X1,23,23)
X09021911$censoINE <- substr(X09021911$X1,24,30)
X09021911$blancos <- substr(X09021911$X1,66,72)
X09021911$nulos <- substr(X09021911$X1,73,79)
X09021911$validos <- substr(X09021911$X1,80,86)
X09021911$censoINE <- as.numeric(X09021911$censoINE)
X09021911$blancos <- as.numeric(X09021911$blancos)
X09021911$nulos <- as.numeric(X09021911$nulos)
X09021911$validos <- as.numeric(X09021911$validos)
X09021911$ID2019Mesa <- paste0(substr(X09021911$distrito,2,2),substr(X09021911$seccion,2,3),X09021911$m
```

Table 5: El data.frame con los resultado por partido

localidad	distrito	seccion	mesa	censoINE	blancos	nulos	validos
0104003	02	001	B	446	1	0	270
0104006	02	001	A	842	4	8	533
0104006	04	001	B	454	3	2	264

data.frame X10021911. Resultados electorales por mesa

Table 6: El data.frame con los resultado por partido

X1
022019111010400302001
B000044600004460000000000000000001410000215000000100000000000270000000000000000S
022019111010400602001
A000084200008420000000000000000002800000440000000400000080000533000000000000000S

X1
022019111010400604001 B000045400004540000000000000000001330000216000000300000020000264000000000000000S

Para el data frame que contiene los datos por partido hacemos un proceso similar.
Lo importante de este fichero es que contiene tantas líneas como partidos han participado en cada mesa. En el data.frame anterior el número de filas es igual al de todas las mesas en España. En este data.frame X10021911 para cada mesa se desdobla en el número de partidos que allí participaron y cada fila representa el resultado de un solo partido dentro de cada mesa.

```
#Crear columnas fichero 10
X10021911$localidad <- substr(X10021911$X1,10,16)
X10021911$distrito <- substr(X10021911$X1,17,18)
X10021911$seccion <- substr(X10021911$X1,19,22)
X10021911$mesa <- substr(X10021911$X1,23,23)
X10021911$partidos <- substr(X10021911$X1,24,29)
X10021911$votos <- substr(X10021911$X1,30,36)
X10021911$votos <- as.numeric(X10021911$votos)
```

Table 7: El data.frame con el resultado de cada partido extraído

localidad	distrito	seccion	mesa	partidos	votos
0104001	01	001	B	000098	0
0104002	01	001	B	000098	0
0104003	01	001	U	000058	0

Creación de la tabla

Con los datos ya acomodados vamos a crear el data.frame correspondiente a un municipio.
Primeramente vamos a crear las columna con los nombres de los partidos en el data.frame con todos los resultados (X10021911) y seleccionaremos las columnas que necesitaremos posteriormente.

```
#Poner los nombres de los partidos y arreglar data.frame
X10021911 <- left_join(X10021911,X03021911[,2:3],by="partidos")
X10021911 <- X10021911[,c(2,3,4,5,8,7)]
```

Table 8: El data.frame con los datos de cada partido

localidad	distrito	seccion	mesa	nombreCorto	votos
0104001	01	001	B	PUM+J	0
0104002	01	001	B	PUM+J	0
0104003	01	001	U	PCPA	0

Ahora comenzamos la extracción del municipio de nuestro interés. En este caso la Comunidad Autónoma es la 12 (Madrid), la provincia la 28 (Madrid) y el municipio el 065 (Getafe). Extraemos los datos del municipio del data.frame de los totales (X09021911). Para esto creamos un nuevo data.frame (getafeTotales) y usando las funciones filter y select de dplyr y construimos unos pipes para obtener el resultado deseado. Con filter separamos nuestro municipio de interés y con select separamos las columnas que contienen los datos de las mesas y su identificador.

```r
#Extraer el municipio de interés de archivo X09021911
getafeTotales <- X09021911 %>%
  filter(localidad=="1228065") %>%
  select(6:10)
```

Table 9: El data.frame con solamente los resultados del partido

censoINE	blancos	nulos	validos	ID2019Mesa
736	2	6	538	102A
831	10	8	631	105A
763	2	4	560	111B

Ahora creamos el data.frame con los resultados electorales. Este es un paso muy importante. pues el resultado final ya será una tabla fácilmente comprensible y que se puede usar para otras aplicaciones. Creamos el data.frame getafe20191110 a partir X10021911 de la siguiente manera:

1. Filtramos la localidad como lo hicimos en el paso anterior.

2. Creamos con mutate una nueva columna ID2019 que luego usaremos para crear otras columnas y hacer unos left_join.

3. El paso más importante es pasar de la presentación larga a aquella donde se visualicen los partidos para cada mesa. Para ello usamos pivot_wider de tidyr names_from señala la columna de donde extraremos los nombres de los partidos. Cada nombre hará una columna nueva. Los datos para llenar esas columnas provendrán de values_from.

```r
#Extraer el municipio de interés de archivo X10021911
getafe20191110 <- X10021911 %>%
  filter(localidad=="1228065") %>%
  mutate(ID2019=paste0(distrito,seccion)) %>%
  pivot_wider(names_from=nombreCorto, values_from=votos) %>%
  group_by(ID2019)
```

Table 10: El data.frame con el resultado del municipio de elección

localidad	distrito	seccion	mesa	ID2019	PUM+J	VOX	MASPAMSEQ	PH
1228065	01	002	A	01002	0	87	36	1
1228065	01	002	B	01002	0	91	19	0
1228065	01	004	U	01004	1	111	25	0

Ahora vamos a crear dos columnas de ID que nos permitan hace los join con los otros data.frame:

1. El ID2019Mesa. El primer dígito corresponde al distrito censal, los dos siguientes a la sección y la letra al final a la mesa electoral. Se crea para que sea homogénea con el identificador del data.frame en getafeTotales.

2. El ID2019. Es una versión homogénea con el ID2019 que viene del data.frame DISTRITOS2019.

```r
#Homogeneizar el ID en getafe20191110 a DISTRITO2019
getafe20191110$ID2019Mesa <- paste0(substr(getafe20191110$ID2019,2,2),substr(getafe20191110$ID2019,4,5)
getafe20191110$ID2019 <- paste0(substr(getafe20191110$ID2019,2,2),substr(getafe20191110$ID2019,4,5))
```

Table 11: La tabla en el nivel de mesas electorales

localidad	distrito	seccion	mesa	ID2019	PUM+J	VOX	MASPAMSEQ
1228065	01	002	A	102	0	87	36
1228065	01	002	B	102	0	91	19
1228065	01	004	U	104	1	111	25

Table 12: La tabla anterior con los datos de identificación

localidad	distrito	seccion	mesa	ID2019	PUM+J	VOX	MASPAMSEQ	DISTRITO	PKB
1228065	01	002	A	102	0	87	36	CENTRO	CE
1228065	01	002	B	102	0	91	19	CENTRO	CE
1228065	01	004	U	104	1	111	25	CENTRO	CE

Ahora vamos a unir dos dos conjuntos de datos al data.frame getafe20191110:

1. Los correspondientes a los datos totales por mesa que vienen del data.frame X09021911.

2. La asignación de a cuál distrito municipal pertenece cada mesa.

```
#Añadir los datos totales
getafe20191110 <- left_join(getafe20191110,getafeTotales,by="ID2019Mesa")
#Añadir los distritos a getafe20191110
getafe20191110 <- left_join(getafe20191110,DISTRITOS2019,by="ID2019")
```

En este paso vamos a crear la tabla sintetizada por Distrito. En este punto ya se puede sintetizar también por sección censal usando el ID2019 (no el ID2019Mesa). Usamos nuevamente un pipe. Con él seleccionamos las columnas de datos y con los nombres de los distritos. Luego agupamos por distrito. Finalmente sumamos cada una de las columnas de los partidos, votos blanco, nulos y censo del INE y votos válidos .

```
#Agrupar por distrito
getafe20191110Distrito <- getafe20191110 %>%
  select(5:24) %>%
  group_by(DISTRITO) %>%
  summarise(
    Cs=sum(Cs),
    VOX=sum(VOX),
    PP=sum(PP),
    PSOE=sum(PSOE),
    PODEMOSIU=sum(PODEMOSIU),
    `PUM+J`=sum(`PUM+J`),
    PACMA=sum(PACMA),
    MASPAMSEQ=sum(MASPAMSEQ),
    PH=sum(PH),
    PCTE=sum(PCTE),
    RO=sum(RECORTESCEROGVPCASTC),
    PCPE=sum(PCPE),
    censoINE=sum(censoINE),
    nulos=sum(nulos),
    blancos=sum(blancos),
    validos=sum(validos)
    )
```

Table 13: Detalle de la tabla por distritos (Principio)

DISTRITO	Cs	VOX	PP	PSOE	PODEMOSIU	PUM+J	PACMA
BUENAVISTA	550	897	505	1325	1077	9	71
CENTRO	1209	2833	3824	4681	2432	24	171
EL BERCIAL	1016	1739	1774	2971	1541	18	94

Table 14: Detalle de la tabla por distritos (Final)

MASM	PH	PCTE	R0	PCPE	censoINE	nulos	blancos	validos
491	5	3	9	6	6207	67	66	4948
730	9	18	16	12	21488	131	139	15959
677	7	3	21	6	12428	106	91	9867

Graficación

Ahora haremos el gráfico correspondiente a uno de los distritos. En primer lugar hay que crear varios vectores que serán parámetros de nuestra función de graficación:

```
# El gráfico
## Construir el data frame y las variables necesarias
### Seleccionamos la fila con el distrito municipal de nuestro interés
### Nombre del distrito
xx <- "BUENAVISTA"
dataGraficante <- getafe20191110Distrito %>%
  filter(DISTRITO==xx) %>%
  select(2:13)
dataGraficante <- as.data.frame(t(dataGraficante))
dataGraficante$nomina <- rownames(dataGraficante)
dataGraficante <- dataGraficante[order(dataGraficante$V1, decreasing = TRUE),]
dataGraficanteAdd <- data.frame("V1"=sum(dataGraficante$V1[9:nrow(dataGraficante)]), "nomina"="Otros")
dataGraficante <- rbind(dataGraficante[1:8,],dataGraficanteAdd[1,])
percenta <- round((dataGraficante$V1/sum(dataGraficante$V1))*100,2)
breaks <- cumsum(porcentaje)
semiSegmenta <- breaks-(percenta/2)
nomina <- c(dataGraficante$nomina)
colores <- c("red","#551a8b","#00b200","#fa5000","#0073cf","#54EFa5","cornflowerblue","darksalmon","cora
```

Luego usamos la función gg.gauge:

```
# Gráfico para ocho partidos
gg.gauge <- function(breaks,porcentaje) {
  require(ggplot2)
  get.poly <- function(a,b,r1=0.5,r2=1.0) {
    th.start <- pi*(1-a/100)
    th.end   <- pi*(1-b/100)
    th       <- seq(th.start,th.end,length=100)
    x        <- c(r1*cos(th),rev(r2*cos(th)))
    y        <- c(r1*sin(th),rev(r2*sin(th)))
    return(data.frame(x,y))
  }
  ggplot()+
```

```
    geom_polygon(data=get.poly(0,breaks[1]),aes(x,y),fill=colores[1])+
    geom_polygon(data=get.poly(breaks[1],breaks[2]),aes(x,y),fill=colores[2])+
    geom_polygon(data=get.poly(breaks[2],breaks[3]),aes(x,y),fill=colores[3])+
    geom_polygon(data=get.poly(breaks[3],breaks[4]),aes(x,y),fill=colores[4])+
    geom_polygon(data=get.poly(breaks[4],breaks[5]),aes(x,y),fill=colores[5])+
    geom_polygon(data=get.poly(breaks[5],breaks[6]),aes(x,y),fill=colores[6])+
    geom_polygon(data=get.poly(breaks[6],breaks[7]),aes(x,y),fill=colores[7])+
    geom_polygon(data=get.poly(breaks[7],breaks[8]),aes(x,y),fill=colores[8])+
    geom_polygon(data=get.poly(breaks[8],breaks[9]),aes(x,y),fill=colores[9])+
    geom_text(data=as.data.frame(porcentaje), size=ifelse(porcentaje>10,2,1), fontface="bold", vjust=0,
              aes(x=0.75*cos(pi*(1-semiSegmenta/100)),y=0.75*sin(pi*(1-semiSegmenta/100)),label=paste0(
    coord_fixed()+
    labs(title=paste0("Porcentaje por partidos: ",xx),
         caption = "Fuente: Min.In., Cálculos propios")+
    theme_bw()+
    theme(axis.text=element_blank(),
          axis.title=element_blank(),
          axis.ticks=element_blank(),
          panel.grid=element_blank(),
          panel.border=element_blank())
  glipho<-paste0(xx,"porcentaje",".png")
  ggsave(glipho,width=10, height=6,units="cm")
}
gg.gauge(breaks,percenta)
```

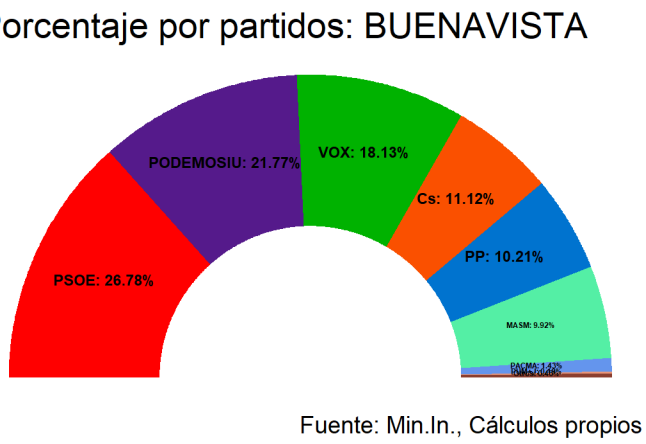

Figure 1: Gráfico de un distrito municipal

Finalmente obtenemos el gráfico: