

Programación en R para el análisis de datos

Funciones

Nicolás Schmidt

`mail::nschmidt@cienciassociales.edu.uy`

`GitHub::@Nicolas-Schmidt`

Departamento de Ciencia Política

Facultad de Ciencias Sociales

1. Carga de datos
2. Cuatro momentos
 - Pensar en funciones
 - Usar funciones
 - Crear funciones
 - Iterar funciones
3. Estructura de una función
4. Creación de una función
5. Mejorando funciones

Carga de datos

Carga de datos

Hay muchos paquetes para cargar datos en función del formato de los mismos.

El paquete más intuitivo y conceptualmente más ordenado a mi criterio es **rio**



```
import(), import_list(), export(), convert()
```

Carga de datos

```
datos <- rio::import("elecciones_nacionales_19_montevideo.xlsx")
```

```
str(datos)
```

```
## 'data.frame': ^I2642 obs. of  19 variables:
## $ CIRCUITO          : num  1 2 3 4 5 6 7 8 9 10 ...
## $ SERIE             : chr   "AAA" "AAA" "AAA" "AAA" ...
## $ HABILITADO        : num  389 389 389 389 389 389 389 389 389 ...
## $ OBSERVADOS        : num  8 4 2 2 2 4 3 2 2 1 ...
## $ T_EMITIDOS        : num  310 348 335 319 340 330 348 318 329 293 ...
## $ EN_BLANCO         : num  4 7 8 4 1 6 6 1 0 2 ...
## $ ANULADOS          : num  3 8 10 10 5 6 4 10 7 4 ...
## $ SOLO_POR_SI       : num  4 3 1 2 3 2 2 2 3 2 ...
## $ Partido Frente Amplio : num  117 129 124 133 164 158 158 136 164 166 ...
## $ Partido Nacional    : num  87 81 67 62 78 66 87 74 70 56 ...
## $ Partido Colorado    : num  38 23 35 20 26 25 13 27 26 19 ...
## $ Partido Independiente : num  5 6 5 5 4 4 4 3 1 0 ...
## $ Partido Asamblea Popular : num  2 4 4 3 5 4 3 5 4 6 ...
## $ Partido de los Trabajadores : num  0 0 1 0 0 0 1 1 2 0 ...
## $ Partido Ecologista Radical Int.: num  5 6 3 3 2 2 1 9 2 1 ...
## $ Partido de la Gente   : num  4 2 2 6 1 4 1 3 4 5 ...
## $ Partido Verde Animalista : num  2 2 5 1 5 4 3 6 2 5 ...
## $ Partido Digital      : num  0 0 0 0 1 0 0 1 2 2 ...
## $ Partido Cabildo Abierto : num  31 73 68 68 43 45 62 38 40 24 ...
```

Cuatro momentos

1 - Pensar en funciones

```
fa <- median(datos[, "Partido Frente Amplio"], na.rm = TRUE)
pn <- median(datos[, "Partido Nacional"], na.rm = TRUE)
pc <- median(datos[, "Partido Colorado"], na.rm = TRUE)
ca <- median(datos[, "Partido Cabildo Abierto"], na.rm = TRUE)
```

fa

```
## [1] 170
```

pn

```
## [1] 79
```

pc

```
## [1] 30
```

ca

```
## [1] 26
```

2 - Usar funciones existentes

```
var <- c("Partido Frente Amplio", "Partido Nacional",  
        "Partido Colorado", "Partido Cabildo Abierto")  
  
medianas <- apply(datos[, var], 2, median, na.rm = TRUE)  
medianas
```

```
## Partido Frente Amplio      Partido Nacional      Partido Colorado  
##                170                79                30  
## Partido Cabildo Abierto  
##                26
```

```
as.data.frame(medianas)
```

```
##                medianas  
## Partido Frente Amplio    170  
## Partido Nacional         79  
## Partido Colorado         30  
## Partido Cabildo Abierto  26
```


2 - Usar funciones existentes

```
var <- c("Partido Frente Amplio", "Partido Nacional",  
        "Partido Colorado", "Partido Cabildo Abierto")
```

```
medias <- apply(datos[, var], 2, mean, na.rm = TRUE)  
medias
```

```
## Partido Frente Amplio      Partido Nacional      Partido Colorado  
##           166.10106           81.25473           37.30469  
## Partido Cabildo Abierto  
##           28.92127
```

```
as.data.frame(medias)
```

```
##           medias  
## Partido Frente Amplio 166.10106  
## Partido Nacional      81.25473  
## Partido Colorado      37.30469  
## Partido Cabildo Abierto 28.92127
```

3 - Crear funciones

```
mi_apply <- function(data, margen, funcion, name.var = "V1", ...){  
  data <- data[, sapply(data, is.numeric)]  
  app <- apply(X = data,  
               MARGIN = margen,  
               FUN = funcion,  
               ...)  
  app <- as.data.frame(app)  
  names(app) <- name.var  
  return(app)  
}
```

3 - Crear funciones

```
mi_apply(datos,2, median, na.rm = TRUE)
```

```
##                               V1
## CIRCUITO                      1321.5
## HABILITADO                    400.0
## OBSERVADOS                     2.0
## T_EMITIDOS                   359.0
## EN_BLANCO                     3.0
## ANULADOS                      7.0
## SOLO_POR_SI                   1.0
## Partido Frente Amplio        170.0
## Partido Nacional              79.0
## Partido Colorado             30.0
## Partido Independiente         4.0
## Partido Asamblea Popular      4.0
## Partido de los Trabajadores    0.0
## Partido Ecologista Radical Int. 6.0
## Partido de la Gente           4.0
## Partido Verde Animalista       3.5
## Partido Digital               1.0
## Partido Cabildo Abierto       26.0
```

3 - Crear funciones

```
cbind(mi_apply(datos,2, median, na.rm = TRUE, name.var = "Mediana"),  
      mi_apply(datos,2, mean, na.rm = TRUE, name.var = "Media"),  
      mi_apply(datos,2, max, name.var = "Máximo"))
```

##	Mediana	Media	Máximo
## CIRCUITO	1321.5	1386.7517033	9027
## HABILITADO	400.0	398.5866768	411
## OBSERVADOS	2.0	2.8523846	63
## T_EMITIDOS	359.0	353.2161241	402
## EN_BLANCO	3.0	3.6059803	12
## ANULADOS	7.0	6.9485238	19
## SOLO_POR_SI	1.0	1.6718395	10
## Partido Frente Amplio	170.0	166.1010598	269
## Partido Nacional	79.0	81.2547313	170
## Partido Colorado	30.0	37.3046934	149
## Partido Independiente	4.0	4.6990916	18
## Partido Asamblea Popular	4.0	4.1211204	20
## Partido de los Trabajadores	0.0	0.2725208	4
## Partido Ecologista Radical Int.	6.0	6.0476911	19
## Partido de la Gente	4.0	4.4341408	18
## Partido Verde Animalista	3.5	3.8156699	15
## Partido Digital	1.0	1.1654050	9
## Partido Cabildo Abierto	26.0	28.9212718	117

4 - Iterar funciones

```
resumenes <- list()
funciones <- c("median", "mean", "min", "max", "sum", "sd")
for(i in 1:length(funciones)){
  resumenes[[i]] <- round(mi_apply(data = datos,
                                  margen = 2,
                                  function = funciones[i],
                                  na.rm = TRUE,
                                  name.var = funciones[i]),
                          2)
}

tabla <- do.call(cbind, resumenes)
```

4 - Iterar funciones

```
print(tabla)
```

```
##                median    mean min  max      sum      sd
## CIRCUITO        1321.5  1386.75   1 9027 3663798 1079.41
## HABILITADO       400.0   398.59   0  411 1053066   19.16
## OBSERVADOS        2.0     2.85   0   63   7536    3.35
## T_EMITIDOS       359.0   353.22   5  402  933197   27.36
## EN_BLANCO         3.0     3.61   0   12   9527    2.11
## ANULADOS          7.0     6.95   0   19   18358    3.18
## SOLO_POR_SI        1.0     1.67   0   10   4417    1.45
## Partido Frente Amplio 170.0  166.10   0  269  438839   43.15
## Partido Nacional      79.0   81.25   0  170  214675   20.43
## Partido Colorado      30.0   37.30   0  149   98559   23.42
## Partido Independiente   4.0   4.70   0   18   12415    2.63
## Partido Asamblea Popular  4.0   4.12   0   20   10888    2.48
## Partido de los Trabajadores 0.0   0.27   0    4    720    0.56
## Partido Ecologista Radical Int. 6.0   6.05   0   19   15978    3.22
## Partido de la Gente      4.0   4.43   0   18   11715    2.87
## Partido Verde Animalista  3.5   3.82   0   15   10081    2.35
## Partido Digital          1.0   1.17   0    9    3079    1.46
## Partido Cabildo Abierto  26.0   28.92   0  117   76410   13.10
```

Volver a crear funciones

```
chucu <- function(data, margen, funcion, name.var = "V1", ...){  
  
  mi_apply <- function(data, margen, funcion, name.var = "V1", ...){  
    data <- data[, sapply(data, is.numeric)]  
    app <- apply(X = data,  
                 MARGIN = margen,  
                 FUN = funcion,  
                 ...)  
    app <- as.data.frame(app)  
    names(app) <- name.var  
    return(app)  
  }  
  funciones <- funcion  
  resmenes <- list()  
  for(i in 1:length(funciones)){  
    resmenes[[i]] <- round(mi_apply(data = data,  
                                   margen = 2,  
                                   funcion = funciones[i],  
                                   na.rm = TRUE,  
                                   name.var = funciones[i]),  
                           2)  
  }  
  tabla <- do.call(cbind, resmenes)  
  return(tabla)  
}
```

Volver a crear funciones

```
source("chucu.R")
fun <- c("median", "mean", "min", "max", "sum", "sd")
chucu(datos, 2, fun, fun, na.rm = TRUE)
```

	median	mean	min	max	sum	sd
## CIRCUITO	1321.5	1386.75	1	9027	3663798	1079.41
## HABILITADO	400.0	398.59	0	411	1053066	19.16
## OBSERVADOS	2.0	2.85	0	63	7536	3.35
## T_EMITIDOS	359.0	353.22	5	402	933197	27.36
## EN_BLANCO	3.0	3.61	0	12	9527	2.11
## ANULADOS	7.0	6.95	0	19	18358	3.18
## SOLO_POR_SI	1.0	1.67	0	10	4417	1.45
## Partido Frente Amplio	170.0	166.10	0	269	438839	43.15
## Partido Nacional	79.0	81.25	0	170	214675	20.43
## Partido Colorado	30.0	37.30	0	149	98559	23.42
## Partido Independiente	4.0	4.70	0	18	12415	2.63
## Partido Asamblea Popular	4.0	4.12	0	20	10888	2.48
## Partido de los Trabajadores	0.0	0.27	0	4	720	0.56
## Partido Ecologista Radical Int.	6.0	6.05	0	19	15978	3.22
## Partido de la Gente	4.0	4.43	0	18	11715	2.87
## Partido Verde Animalista	3.5	3.82	0	15	10081	2.35
## Partido Digital	1.0	1.17	0	9	3079	1.46
## Partido Cabildo Abierto	26.0	28.92	0	117	76410	13.10

Volver a crear funciones

```
sprod <- function(x, na.rm = FALSE){  
  if(isTRUE(na.rm)) sum(x[!is.na(x)]^2) else sum(x^2)  
}  
fun <- c("median", "mean", "sd", "sprod")  
chucu(datos, 2, fun, fun, na.rm = TRUE)
```

	median	mean	sd	sprod
## CIRCUITO	1321.5	1386.75	1079.41	8157883270
## HABILITADO	400.0	398.59	19.16	420707538
## OBSERVADOS	2.0	2.85	3.35	51088
## T_EMITIDOS	359.0	353.22	27.36	331596829
## EN_BLANCO	3.0	3.61	2.11	46085
## ANULADOS	7.0	6.95	3.18	154184
## SOLO_POR_SI	1.0	1.67	1.45	12929
## Partido Frente Amplio	170.0	166.10	43.15	77809315
## Partido Nacional	79.0	81.25	20.43	18546083
## Partido Colorado	30.0	37.30	23.42	5125321
## Partido Independiente	4.0	4.70	2.63	76575
## Partido Asamblea Popular	4.0	4.12	2.48	61152
## Partido de los Trabajadores	0.0	0.27	0.56	1016
## Partido Ecologista Radical Int.	6.0	6.05	3.22	123990
## Partido de la Gente	4.0	4.43	2.87	73639
## Partido Verde Animalista	3.5	3.82	2.35	53097
## Partido Digital	1.0	1.17	1.46	9251
## Partido Cabildo Abierto	26.0	28.92	13.10	2662758

Estructura de una función

Una función en R es un objeto. Las funciones están compuestas de tres elementos:

Función	Descripción
<code>formals()</code>	Argumentos: la lista de argumentos que contiene la función.
<code>body()</code>	Cuerpo: todo el código que está dentro de la función.
<code>environment()</code>	Entorno: determina la forma como la función encuentra los valores asociados con los nombres.

Estructura: ejemplos

```
formals(mi_apply)
```

```
## $data  
##  
##  
## $margin  
##  
##  
## $funcion  
##  
##  
## $name.var  
## [1] "V1"  
##  
## $...
```

```
environment(mi_apply)
```

```
## <environment: R_GlobalEnv>
```

Estructura: ejemplos

body(mi_apply)

```
## {  
##   data <- data[, sapply(data, is.numeric)]  
##   app <- apply(X = data, MARGIN = margen, FUN = funcion, ...)  
##   app <- as.data.frame(app)  
##   names(app) <- name.var  
##   return(app)  
## }
```

Creación de una función

En términos generales (luego veremos las excepciones) las funciones deben tener lo siguiente:

- Un nombre (Ver ayuda de: `'?Reserved'` y `'?make.names'`)
- La función `function()`
- Llaves: `'{'` y `'}'`

```
nombre <- function(argumentos){cuerpo}
```

Creación de funciones: funciones y ambientes

```
a <- function(texto){print(texto)}  
a <- 1:15  
a
```

```
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

```
a("Hola mundo")
```

```
## Error in a("Hola mundo"): could not find function "a"
```

```
ls()
```

```
## [1] "a"          "ca"         "chucu"      "datos"      "fa"  
## [6] "fun"        "funciones"  "i"          "medianas"   "medias"  
## [11] "mi_apply"   "pc"         "pn"         "resumenes"  "sprod"  
## [16] "tabla"     "var"
```


Creación de funciones: funciones y ambientes

```
mean <- 1:15  
mean
```

```
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
```

```
mean(mean)
```

```
## [1] 8
```

```
ls()
```

```
## [1] "a"          "ca"         "chucu"      "datos"      "fa"  
## [6] "fun"        "funciones" "i"          "mean"       "medianas"  
## [11] "medias"     "mi_apply"   "pc"         "pn"         "resumenes"  
## [16] "sprod"      "tabla"      "var"
```

¿Que regresa esta función?

```
foo <- function(){  
  mean <- 1  
  foo2 <- function(){  
    mean <- 3 - mean  
  }  
  c(mean, foo2(), mean(1:5))  
}
```

Mejorando funciones

Mejorando funciones

```
datos2 <- datos["SERIE"]  
head(datos2)
```

```
##      SERIE  
## 1      AAA  
## 2      AAA  
## 3      AAA  
## 4      AAA  
## 5      AAA  
## 6      AAA
```

```
mi_apply(datos2, 2, median, na.rm = TRUE, name.var = "Mediana")
```

```
## [1] Mediana  
## <0 rows> (or 0-length row.names)
```

Mejorando funciones

```
mi_apply <- function(data, margen, funcion, ..., name.var = "V1"){  
  data <- data[, sapply(data, is.numeric)]  
  
  if(ncol(data) == 0L){  
    stop("No hay ninguna variable numerica en el dataset.")  
  }  
  
  app <- apply(X = data, MARGIN = margen, FUN = funcion, ... )  
  app <- as.data.frame(app)  
  names(app) <- name.var  
  return(app)  
}
```

```
mi_apply(datos2, 2, median, na.rm = TRUE, name.var = "Mediana")
```

```
## Error in mi_apply(datos2, 2, median, na.rm = TRUE, name.var =  
"Mediana"): No hay ninguna variable numerica en el dataset.
```

```
datos3 <- datos$SERIE  
head(datos3)
```

```
## [1] "AAA" "AAA" "AAA" "AAA" "AAA" "AAA"
```

```
mi_apply(datos3, 2, median, na.rm = TRUE, name.var = "Mediana")
```

```
## Error in data[, sapply(data, is.numeric)]: incorrect number of  
dimensions
```

Mejorando funciones

```
mi_apply <- function(data, margen, funcion, ..., name.var = "V1"){  
  if(!is.data.frame(data)){  
    stop("El argumento 'data' debe ser 'data.frame'")  
  }  
  
  data <- data[, sapply(data, is.numeric)]  
  
  if(ncol(data) == 0L){  
    stop("No hay ninguna variable numerica en el dataset")  
  }  
  
  data <- data[, sapply(data, is.numeric)]  
  app <- apply(X = data, MARGIN = margen, FUN = funcion, ... )  
  app <- as.data.frame(app)  
  names(app) <- name.var  
  return(app)  
}  
mi_apply(datos3, 2, median, na.rm = TRUE, name.var = "Mediana")  
  
## Error in mi_apply(datos3, 2, median, na.rm = TRUE, name.var =  
"Mediana"): El argumento 'data' debe ser 'data.frame'
```

Funciones que van a permitir mejorar nuestras funciones:

- `stop()`
- `stopifnot()`
- `warnings()`
- `message()`
- `missing()`
- `return()`
- `invisible()`
- `on.exit()`

Juguemos un rato!

Preguntas

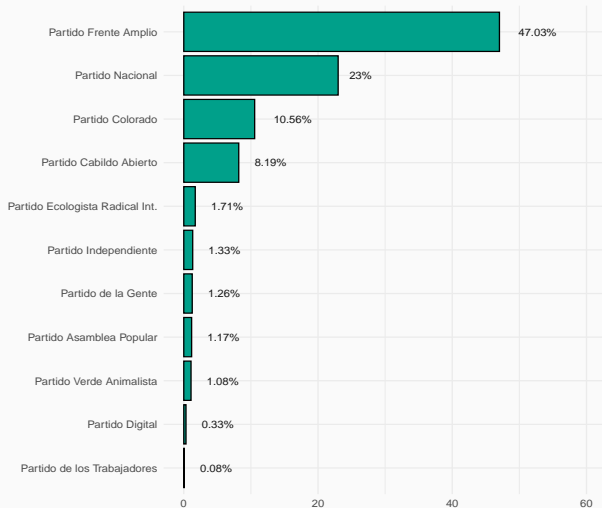
1. Quién ganó la elección en Montevideo?, por cuanto?
2. Cómo fue el resultado por municipio?
3. En que circuito votó más cada partido?
4. Cómo visualizaría el resultado por municipio?
5. Es posible georreferenciar el ganador de cada circuito?

Respuestas

```
votos <- apply(datos[,5:ncol(datos)], 2, sum, na.rm = TRUE)
votos_emitidos <- sum(datos$T_EMITIDOS)
final <- data.frame(resultados = round((votos/votos_emitidos)*100,2))
final$partido <- rownames(final)
```

```
library(ggplot2)
ggplot(data = final[5:nrow(final),], aes(x = reorder(partido, resultados), y = resultados)) +
  geom_bar(stat="identity", position="stack", fill = "#00A08A", color = "black") +
  ylim(0,60) +
  coord_flip() +
  geom_text(aes(label= paste0(resultados, "%")), hjust = -0.5, color = "gray9", size = 3) +
  labs(x = "", y = "",
       title = "Resultado de elección nacional en Montevideo",
       subtitle = "27 de octubre de 2019",
       caption = "curso mediaR") +
  theme_minimal()
```

Resultado de elección nacional en Montevideo
27 de octubre de 2019



curso mediaR

Respuestas

```
print(final[1])
```

```
##                                resultados
## T_EMITIDOS                    100.00
## EN_BLANCO                     1.02
## ANULADOS                      1.97
## SOLO_POR_SI                   0.47
## Partido Frente Amplio        47.03
## Partido Nacional             23.00
## Partido Colorado             10.56
## Partido Independiente        1.33
## Partido Asamblea Popular     1.17
## Partido de los Trabajadores  0.08
## Partido Ecologista Radical Int. 1.71
## Partido de la Gente          1.26
## Partido Verde Animalista     1.08
## Partido Digital              0.33
## Partido Cabildo Abierto      8.19
```

Respuestas

```
# MUNICIPIOS DE MONTEVIDEO

A <- c('BSA', 'BSB', 'BTA', 'BTB', 'BTC', 'BUA', 'BUB', 'BVA', 'BVB', 'BVC', 'BXA', 'BXB')
B <- c('AAA', 'ACA', 'AIA', 'AKA', 'AMA', 'APA', 'ARA', 'ASA', 'ATA', 'AVA', 'AVB', 'BEA',
      'BFA', 'BJB', 'BKB', 'AAB', 'ABA', 'ABB', 'ACB', 'ADA', 'ADB', 'AEA', 'AEB', 'AFA',
      'AFB', 'AGA', 'AGB', 'AHA', 'AHB', 'AIB', 'AJA', 'AJB', 'AKB', 'ALA', 'ALB', 'AMB',
      'ANA', 'ANB', 'AOA', 'AOB', 'APB', 'AQA', 'aqb', 'ARB', 'ASB', 'BEB', 'BFB')
C <- c('BGA', 'BIA', 'BJA', 'BKA', 'BLA', 'BLB', 'BMA', 'BMB', 'BNA', 'BPA', 'BPB', 'BQA',
      'BQB', 'BGB', 'BHA', 'BIB', 'BHB')
CH <- c('ATB', 'AUA', 'AUB', 'AXA', 'AXB', 'AZA', 'AZB', 'BAA', 'BAB')
D <- c('BBA', 'BBB', 'BDD', 'BNB', 'BNC', 'BOA', 'BOB')
E <- c('BCA', 'BCB', 'BCC', 'BCD', 'BCE', 'BCF')
F <- c('BDA', 'BDB', 'BDC', 'BDE')
G <- c('BRA', 'BRB', 'BRC', 'BZA', 'BZB', 'BZC')

series <- c(A, B, C, CH, D, E, F, G)
secuencia <- c(
  rep("A", length(A)),
  rep("B", length(B)),
  rep("C", length(C)),
  rep("CH", length(CH)),
  rep("D", length(D)),
  rep("E", length(E)),
  rep("F", length(F)),
  rep("G", length(G))
)

municipios <- data.frame(
  MUNICIPIO = secuencia,
  SERIE = series,
  stringsAsFactors = FALSE
)
```

Respuestas

```
head(municipios, 13)
```

```
##      MUNICIPIO SERIE
## 1           A   BSA
## 2           A   BSB
## 3           A   BTA
## 4           A   BTB
## 5           A   BTC
## 6           A   BUA
## 7           A   BUB
## 8           A   BVA
## 9           A   BVB
## 10          A   BVC
## 11          A   BXA
## 12          A   BXB
## 13          B   AAA
```

```
str(municipios)
```

```
## 'data.frame': ^I108 obs. of  2 variables:
## $ MUNICIPIO: chr  "A" "A" "A" "A" ...
## $ SERIE     : chr  "BSA" "BSB" "BTA" "BTB" ...
```

```
table(municipios$MUNICIPIO)
```

```
##
##  A  B  C CH D  E  F  G
## 12 47 17  9  7  6  4  6
```

Respuestas

```
dim(datos)

## [1] 2642  19

datos2 <- merge(datos, municipios, by = 'SERIE')
dim(datos2)

## [1] 2575  20

any(names(datos2) == "MUNICIPIO")

## [1] TRUE

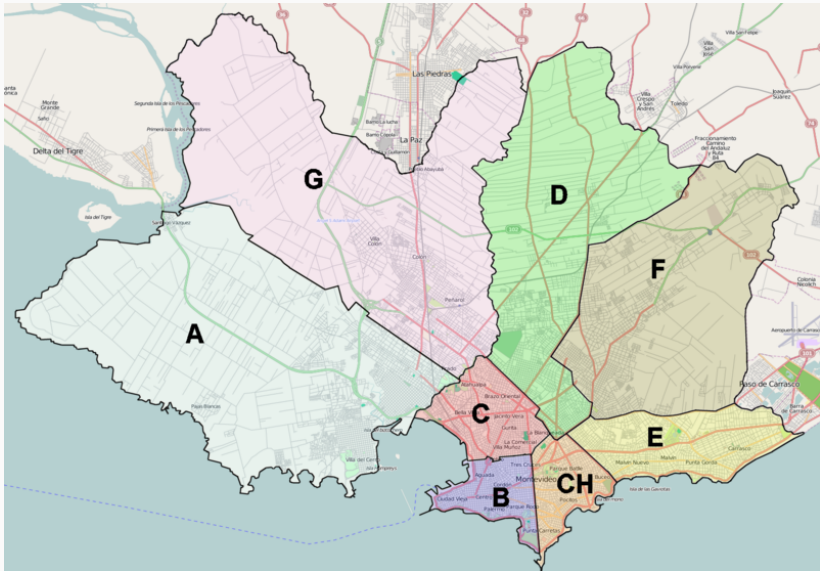
sum(is.na(datos2$MUNICIPIO))

## [1] 0
```


Respuestas

```
mi_func <- function(x){  
  df <- apply(x[,5:(ncol(x)-1)], 2, sum, na.rm = TRUE)  
  votos_emitidos <- sum(x$T_EMITIDOS)  
  final <- data.frame(resultados = round((df/votos_emitidos)*100,2))  
  final$partido <- rownames(final)  
  final$municipio <- x$MUNICIPIO[1]  
  rownames(final) <- NULL  
  final  
}  
eda <- split(datos2, datos2$MUNICIPIO)  
eda <- lapply(eda, function(x) mi_func(x))  
eda <- do.call(rbind, eda)  
eda <- eda[grepl("^Partido", eda$partido), ]
```

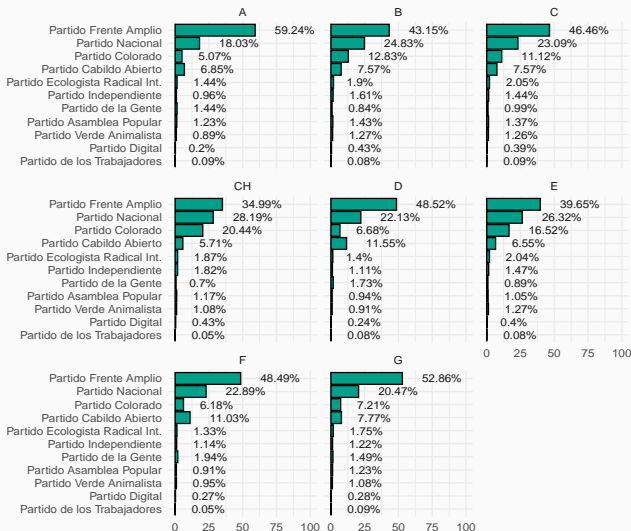
Respuestas



Respuestas

```
ggplot(data = eda, aes(x = reorder(partido, resultados), y = resultados)) +  
  geom_bar(stat="identity", position="stack", fill = "#00A08A", color = "black") +  
  ylim(0,100) +  
  coord_flip() +  
  geom_text(aes(label= paste0(resultados, "%")), hjust = -0.5, color = "gray9", size = 3) +  
    labs(x = "",  
         y = "",  
         title = "Resultado de elección nacional por municipio de Montevideo",  
         subtitle = "27 de octubre de 2019",  
         caption = "curso mediaR") +  
  theme_minimal() +  
  facet_wrap(~municipio)
```

Resultado de elección nacional por municipio de Montevideo 27 de octubre de 2019



Respuestas

```
tidyr::pivot_wider(eda, names_from = municipio, values_from = resultados)
```

##	partido	A	B	C	CH	D	E	F	G
## 1	Asamblea Popular	1.23	1.43	1.37	1.17	0.94	1.05	0.91	1.23
## 2	Cabildo Abierto	6.85	7.57	7.57	5.71	11.55	6.55	11.03	7.77
## 3	Colorado	5.07	12.83	11.12	20.44	6.68	16.52	6.18	7.21
## 4	de la Gente	1.44	0.84	0.99	0.70	1.73	0.89	1.94	1.49
## 5	de los Trabajadores	0.09	0.08	0.09	0.05	0.08	0.08	0.05	0.09
## 6	Digital	0.20	0.43	0.39	0.43	0.24	0.40	0.27	0.28
## 7	Eco. Radical Int.	1.44	1.90	2.05	1.87	1.40	2.04	1.33	1.75
## 8	Frente Amplio	59.24	43.15	46.46	34.99	48.52	39.65	48.49	52.86
## 9	Independiente	0.96	1.61	1.44	1.82	1.11	1.47	1.14	1.22
## 10	Nacional	18.03	24.83	23.09	28.19	22.13	26.32	22.89	20.47
## 11	Verde Animalista	0.89	1.27	1.26	1.08	0.91	1.27	0.95	1.08

Respuestas

```
datos <- datos[, c(1:2, 9:19)]
datos$win <- apply(datos[, -c(1:2)], 1, which.max)
vec <- character()
partidos <- names(datos)[3:13]
for(i in 1:nrow(datos)){vec[i] <- partidos[datos$win[i]]}
datos$party_win <- vec
```

```
table(datos$win)
```

```
##
##      1      2      3     11
## 2430  191   20      1
```

```
as.data.frame(table(datos$party_win))
```

```
##                               Var1 Freq
## 1 Partido Cabildo Abierto      1
## 2      Partido Colorado     20
## 3 Partido Frente Amplio  2430
## 4      Partido Nacional   191
```

Respuestas

```
geo      <- rio::import("plan_circuitual_montevideo.xlsx")
geo2     <- geo[, c("SERIE", "Circuito", "Latitud", "Longitud" )]
geo2$id  <- paste(geo2$SERIE, geo2$Circuito)
datos$id <- paste(datos$SERIE, datos$CIRCUITO)
geo2     <- merge(datos, geo2, by = "id")

geo2$color <- ifelse(geo2$win == 1, "green",
                    ifelse(geo2$win == 2, "blue",
                          ifelse(geo2$win == 3, "red", "yellow")))
geo2$Latitud <- as.numeric(geo2$Latitud)
geo2$Longitud <- as.numeric(geo2$Longitud)

geo2$titulo <- paste(geo2$Latitud, geo2$Longitud)
geo2$win2 <- ifelse(geo2$win == 1, 2,
                   ifelse(geo2$win == 2, 2,
                         ifelse(geo2$win == 3, 3, 5)))

library(leaflet)
library(magrittr)

leaflet(geo2) %>% addTiles() %>%
  addCircles(lng = ~Longitud, lat = ~Latitud,
            weight = 1, color = "black", fillColor = ~color,
            radius = ~win2 * 100, popup = ~titulo, opacity = 1, fillOpacity = 0.9)
```

Respuestas

