

Guillermo Robles

```
%%capture
!pip install rpy2==3.5.1

%load_ext rpy2.ipython

from google.colab import drive
drive.mount('/content/drive', force_remount=True)

Mounted at /content/drive

import rpy2.robj as robj

# Establecer la configuración regional en R
robj.r('Sys.setlocale(category = "LC_ALL", locale = "en_US.UTF-8")')

StrVector with 1 elements.
'LC_CTYPE=en_US.UTF-8;LC_NUMERIC=C;LC_TIME=en_US....

%%R
library(readr)
library(magrittr)
library(dplyr)
library(ggplot2)

%%R
datos <- read.csv("drive/MyDrive/Resultados.csv", fileEncoding = "UTF-8")

%%R
colnames(datos)

      [1] "No."           "Mes"           "Delegaci.n"
      [4] "Localidad"     "Esquema"       "N.mero"
      [7] "Nombre"        "X2.1"          "X2.2"
     [10] "X2.3"          "X2.4"          "X2.5"
     [13] "X2.6"          "X2.7"          "X2.8"
     [16] "X2.9"          "X2.1.1"        "X2.11"
     [19] "X2.12"         "X3.1"          "X3.2"
     [22] "X5.1"          "X5.2"          "X5.3"
     [25] "X5.4"          "X5.5"          "X5.6"
     [28] "X5.7"          "X5.8"          "X6.1"
     [31] "X6.2"          "X6.3"          "X6.4"
     [34] "X6.5"          "X7.1"          "X7.2"
     [37] "X7.3"          "X7.4"          "X8.1"
     [40] "X8.2"          "X9.1"          "X9.2"
     [43] "X9.3"          "X9.4"          "X10.1"
     [46] "X10.2"         "X10.3"         "X10.4"
     [49] "X10.5"         "X10.6"         "X10.7"
     [52] "X10.8"         "X10.9"         "X11.1"
     [55] "X11.2"         "X11.3"         "X11.4"
     [58] "X11.5"         "X11.6"         "X11.7"
     [61] "X11.8"         "X11.9"         "X11.1.1"
     [64] "X11.11"        "X11.12"        "X11.13"
     [67] "X11.14"        "X11.15"        "X11.16"
     [70] "X12.1"         "X12.2"         "X12.3"
     [73] "X12.4"         "X12.5"         "X12.6"
     [76] "X12.7"         "X12.8"         "X14.1"
     [79] "X14.2"         "Puntos.m.ximos" "Puntos.alcanzados"
     [82] "X."           "No..de.Padres" "Horas"
```

Una que contabilice la delegación con más localidades participantes en la encuesta.

```
%%R
```

```

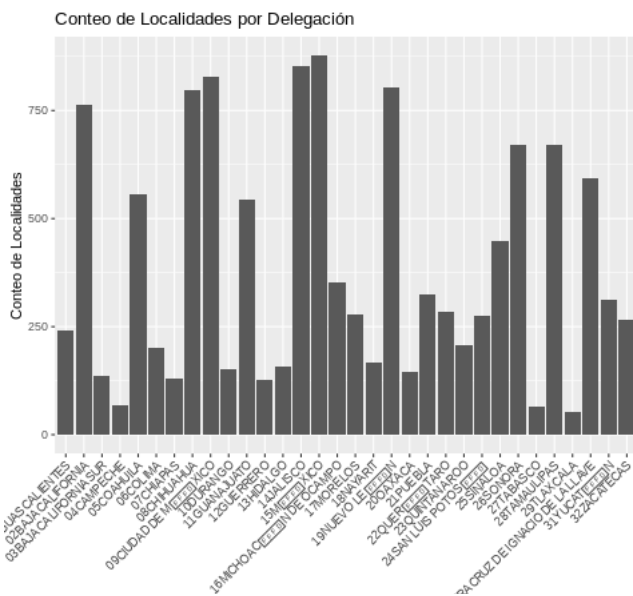
get_delegacion_con_mas_localidades <- function(df) {
  conteo_localidades <- table(df$Delegaci.n)
  max_localidades <- max(conteo_localidades)
  delegacion_mas_localidades <- names(conteo_localidades)[conteo_localidades == max_localidades]
  return(delegacion_mas_localidades)
}

delegacion_con_mas_localidades <- get_delegacion_con_mas_localidades(datos)

# Crear un data frame con el conteo de localidades por delegación
conteo_df <- data.frame(Delegaci.n = names(table(datos$Delegaci.n)),
  Conteo = as.numeric(table(datos$Delegaci.n)))

# Crear la gráfica de barras
ggplot(conteo_df, aes(x = Delegaci.n, y = Conteo)) +
  geom_bar(stat = "identity") +
  labs(x = "Delegación", y = "Conteo de Localidades") +
  ggtitle("Conteo de Localidades por Delegación") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

```



```

%%R
get_delegacion_con_mas_localidades <- function(df) {
  conteo_localidades <- table(df$Delegaci.n)
  max_localidades <- max(conteo_localidades)
  delegacion_mas_localidades <- names(conteo_localidades)[conteo_localidades == max_localidades]
  return(delegacion_mas_localidades)
}

delegacion_con_mas_localidades <- get_delegacion_con_mas_localidades(datos)

delegacion_con_mas_localidades

[1] "15MEXICO"

```

b) Función que indique el porcentaje de participación por delegación.

```

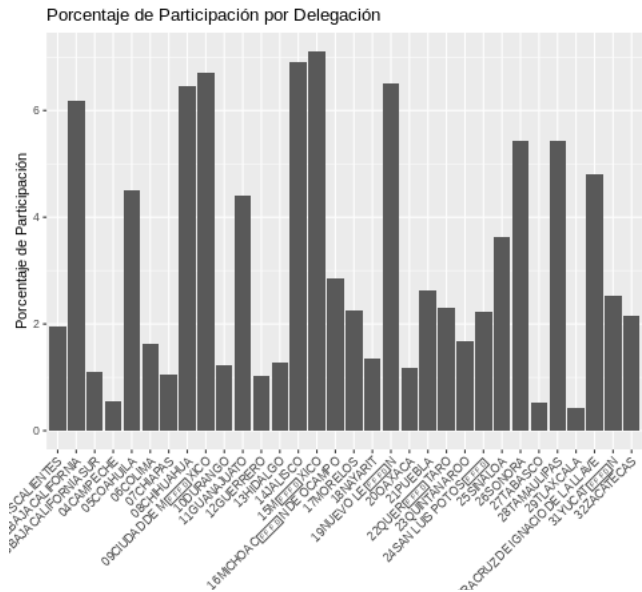
%%R
calcular_porcentaje_participacion <- function(df) {
  conteo_localidades <- table(df$Delegaci.n)
  porcentaje_participacion <- prop.table(conteo_localidades) * 100
  return(porcentaje_participacion)
}

porcentaje_participacion_delegacion <- calcular_porcentaje_participacion(datos)

# Crear un data frame con el porcentaje de participación por delegación
porcentaje_df <- data.frame(Delegaci.n = names(porcentaje_participacion_delegacion),
  Porcentaje = as.numeric(porcentaje_participacion_delegacion))

```

```
# Crear la gráfica de barras
ggplot(porcentaje_df, aes(x = Delegaci.n, y = Porcentaje)) +
  geom_bar(stat = "identity") +
  labs(x = "Delegación", y = "Porcentaje de Participación") +
  ggtitle("Porcentaje de Participación por Delegación") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
%%R
calcular_porcentaje_participacion <- function(df) {
  conteo_localidades <- table(df$Delegaci.n)
  porcentaje_participacion <- prop.table(conteo_localidades) * 100
  return(porcentaje_participacion)
}

porcentaje_participacion_delegacion <- calcular_porcentaje_participacion(datos)
porcentaje_participacion_delegacion
```

01AGUASCALIENTES	02BAJA CALIFORNIA
1.9518911	6.1877379
03BAJA CALIFORNIA SUR	04CAMPECHE
1.1014821	0.5588402
05COAHUILA	06COLIMA
4.5031182	1.6279258
07CHIAPAS	08CHIHUAHUA
1.0447882	6.4550093
09CIUDAD DE MEXICO	10DURANGO
6.6979833	1.2391674
11GUANAJUATO	12GUERRERO
4.3978294	1.0204908
13HIDALGO	14JALISCO
1.2796631	6.9085608
15MICHOACÁN	16MICHOACÁN DE OCAMPO
7.1029400	2.8508950
17MORELOS	18NAYARIT
2.2434599	1.3444561
19NUEVO LEÓN	20OAXACA
6.5117032	1.1743743
21PUEBLA	22QUERÉTARO
2.6322184	2.3001539
23QUINTANA ROO	24SAN LUIS POTOSÍ
1.6684215	2.2272617
25SINALOA	26SONORA
3.6365109	5.4426176
27TABASCO	28TAMAULIPAS
0.5345428	5.4426176
29TLAXCALA	30VERACRUZ DE IGNACIO DE LA LLAVE
0.4292541	4.8027861
31YUCATÁN	32ZACATECAS
2.5188305	2.1624686

Una que indique las preguntas menos respondidas en relación con el resto.

```
##R
# Convertir columna de respuestas a numérico
datos[, 8:ncol(datos)] <- lapply(datos[, 8:ncol(datos)], function(x) as.numeric(as.character(x)))

# Calcular el número de respuestas para cada pregunta
respuestas <- colSums(datos[, 8:ncol(datos)], na.rm = TRUE)

# Combinar las respuestas con el nombre de la pregunta correspondiente
preguntas <- names(respuestas)
preguntas_respuestas <- data.frame(pregunta = preguntas, respuestas = respuestas)

# Ordenar las preguntas por cantidad de respuestas en orden ascendente
preguntas_menos_respondidas <- preguntas_respuestas[order(preguntas_respuestas$respuestas, na.last = NA), ]
preguntas_menos_respondidas
```

X11.6	X11.6	12193
X5.1	X5.1	12208
X12.5	X12.5	12226
X3.2	X3.2	12231
X5.7	X5.7	12232
X11.15	X11.15	12244
X11.9	X11.9	12246
X11.8	X11.8	12248
X10.2	X10.2	12261
X14.1	X14.1	12271
X6.2	X6.2	12272
X12.6	X12.6	12273
X10.8	X10.8	12274
X11.1.1	X11.1.1	12275
X7.3	X7.3	12281
X10.7	X10.7	12281
X11.11	X11.11	12282
X5.8	X5.8	12283
X9.3	X9.3	12286
X10.5	X10.5	12290
X10.9	X10.9	12297
X2.2	X2.2	12299
X9.2	X9.2	12299
X2.1.1	X2.1.1	12302
X2.9	X2.9	12304
X7.4	X7.4	12306
X2.5	X2.5	12308
X2.7	X2.7	12309
X10.3	X10.3	12313
X11.12	X11.12	12314
X12.4	X12.4	12314
X2.12	X2.12	12317
X6.4	X6.4	12318
X7.1	X7.1	12321
X2.4	X2.4	12322
X2.11	X2.11	12322
X5.4	X5.4	12324
X6.1	X6.1	12324
X10.4	X10.4	12325
X2.1	X2.1	12326
X8.1	X8.1	12327
X8.2	X8.2	12330
X11.1	X11.1	12330
X9.4	X9.4	12333
X2.3	X2.3	12334
X5.3	X5.3	12334
X10.6	X10.6	12334
X12.1	X12.1	12334
X6.3	X6.3	12335
X9.1	X9.1	12335
X11.3	X11.3	12335
X3.1	X3.1	12338
X2.8	X2.8	12340
X11.2	X11.2	12340
X6.5	X6.5	12341
No..de.Padres	No..de.Padres	65872
Horas	Horas	131744
Puntos.alcanzados	Puntos.alcanzados	859992
Puntos.m.vimos	Puntos.m.vimos	868267

```
##R
# Convertir columna de respuestas a numérico
datos[, 8:ncol(datos)] <- lapply(datos[, 8:ncol(datos)], function(x) as.numeric(as.character(x)))
```

```

datos[, 8:ncol(datos)] <- apply(datos[, 8:ncol(datos)], MARGIN=2, FUN=function(x) {summarize(orden=order(x))})

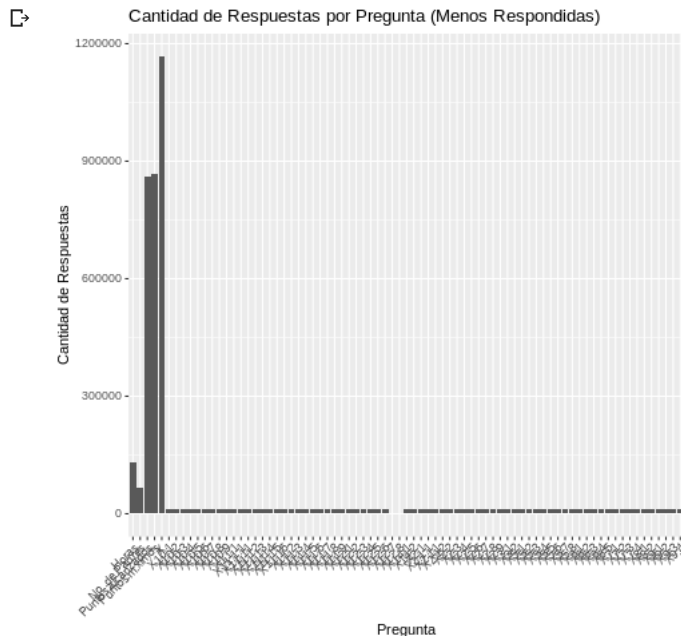
# Calcular el número de respuestas para cada pregunta
respuestas <- colSums(datos[, 8:ncol(datos)], na.rm = TRUE)

# Combinar las respuestas con el nombre de la pregunta correspondiente
preguntas <- names(respuestas)
preguntas_respuestas <- data.frame(pregunta = preguntas, respuestas = respuestas)

# Ordenar las preguntas por cantidad de respuestas en orden ascendente
preguntas_menos_respondidas <- preguntas_respuestas[order(preguntas_respuestas$respuestas, na.last = NA), ]

# Crear la gráfica de barras
ggplot(preguntas_menos_respondidas, aes(x = pregunta, y = respuestas)) +
  geom_bar(stat = "identity") +
  labs(x = "Pregunta", y = "Cantidad de Respuestas") +
  ggtitle("Cantidad de Respuestas por Pregunta (Menos Respondidas)") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

```



Una función que muestre los porcentajes de preguntas contestadas por localidad.

```

%%R
calcularPorcentajes <- function(datos) {
  # Agrupar los datos por localidad y calcular el porcentaje de preguntas contestadas
  porcentajes <- datos %>%
    group_by(Localidad) %>%
    summarize(Porcentaje = mean(!is.na(`Puntos.alcanzados`)) * 100)

  # Mostrar los resultados
  print(porcentajes)
}

```

```

# Llamar a la función
calcularPorcentajes(datos)

```

```

# A tibble: 309 × 2
  Localidad          Porcentaje
  <chr>              <dbl>
1 0001CIUDAD CONSTITUCION      100
2 01001AGUASCALIENTES         100
3 01003CALVILLO               100
4 01005JESUS MARQUEZ          100
5 01006PABELLON ARTEAGA        100
6 01007RINCON DE ROMOS         100
7 01011SAN FRANCISCO DE LOS ROMO 100
8 02001ENSENADA               100
9 02002MEXICALI               100
10 02003TECATE                 100
# i 299 more rows
# i Use `print(n = ...)` to see more rows

```

```
calcularPorcentajes <- function(datos) {
  # Agrupar los datos por localidad y calcular el porcentaje de preguntas contestadas
  porcentajes <- datos %>%
    group_by(Localidad) %>%
    summarize(Porcentaje = mean(!is.na(`Puntos.alcanzados`)) * 100)

  # Crear la gráfica de barras
  ggplot(porcentajes, aes(x = Localidad, y = Porcentaje)) +
    geom_bar(stat = "identity") +
    labs(x = "Localidad", y = "Porcentaje de preguntas contestadas") +
    ggtitle("Porcentaje de preguntas contestadas por Localidad") +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
}

# Llamar a la función
calcularPorcentajes(datos)
```

