---
title: "Entrega_1_Visualización_de_Datos"
format: html
editor: visual
author: Rubén Garrido Hidalgo
asignatura: Visualización de Datos
fecha: 21/01/2025
posit Cloud: Rstudio
---

Vamos a recopilar las cinco funciones con sus respectivas gráficas realizadas de la actividad 1

## 1) Primera Representación (Comparisons)

He escogido el waffle como modelo para diseñar nuestra gráfica.

Diseño de la visualización:

#### Importamos las librerias que vamos a emplear

```{r}
#install.packages("ggplot2")
library(ggplot2)
#install.packages("waffle")
library(waffle)

```

#### Importamos los datos que vamos a usar

```{r}

ruta <- "alumnos_calificaciones.csv"
datos_waffle <- read.csv(ruta)

head(datos_waffle)
```

```{r}
datos_waffle$grupo <- cut(datos_waffle$Calificacion,
                   breaks = c(-1, 59, 69, 79, 89, 100),
                   labels = c("F", "D", "C", "B", "A"))
#Contamos cuantos alumnos hay por grupo:
cantidad_grupo <- table(datos_waffle$grupo)

```

#### Por último relizamos la representación gráfica

```{r}
waffle(cantidad_grupo, rows = 10, size = 0.5) +
  labs(
    title = "Distribución de Calificaciones de Alumnos",
    x = "1 cuadrado = 1 alumno",
    y = NULL
  ) +
  theme_void() +
```

```
  theme(
    plot.title = element_text(hjust = 0.5, size = 16, face = "bold"),
    axis.title.x = element_text(size = 12)
  )
```

#### Conclusión:

Podemos observar que 57 alumnos suspendieron, obteniendo una F, ,nueve alumnos obtuvieron una D, doce alumnos obtuvieron una C y una B, y solo nueve alumnos obtuvieron un sobresaliente.

## 2) Segunda Representación (Distributions)

He seleccionado el tema de physical, voy a realizar un histograma de las muertes diarias desde que empezó el COVID-19 en 2020 hasta el 2023.Además, vamos a ver la evolución de las muertes mediante los datos recopilados de distintos paises a lo largo del mundo en paises como Afghanistán, Australia, Jamaica....

Diseño de la visualización:

#### Vamos a cargara los datos

```{r}
library(ggplot2)
library(dplyr)
```

#### Leemos el archivo CSV y creamos un primer histograma de los datos

```{r}
data <- read.csv("data_global_data.csv")

# Inspeccionar las primeras filas para verificar los datos
head(data)

# Creamos un histograma de las muertes diarias (daily_deaths)
ggplot(data, aes(x = daily_deaths)) +
  geom_histogram(binwidth = 10, fill = "blue", color = "black", alpha = 0.
7) +
  labs(title = "Histograma de Muertes Diarias",
       x = "Muertes Diarias",
       y = "Frecuencia") +
  scale_x_continuous(limits = c(100, 3000)) +  # Ajusta los límites del eje
X según el rango deseado
  scale_y_continuous(limits = c(0, 500)) +  # Ajusta los límites del eje Y
según las frecuencias
  theme_minimal()
```

#### Filtramos los datos para eliminar aquellos que poseen valores NA

```{r}
filtered_data <- data %>%
  filter(!is.na(daily_deaths))
minimo_muertes <- min(data$daily_deaths)
minimo_muertes
```

```

#### Creamos el histograma tras limpiar los datos

```{r}
ggplot(filtered_data, aes(x = daily_deaths)) +
  geom_histogram(binwidth = 100, fill = "skyblue", color = "black", alpha =
0.8) +  # Mejorar las barras
  labs(
    title = "Distribución de Muertes Diarias",
    subtitle = "Frecuencia de las muertes diarias por COVID-19",
    x = "Muertes Diarias",
    y = "Frecuencia"
  ) +
  scale_x_continuous(
    breaks = seq(0, max(filtered_data$daily_deaths, na.rm = TRUE), by =
1000),
    labels = scales::comma_format()
  ) +
  scale_y_continuous(limits = c(0, 400)) +   # Limitar los valores del eje
Y
  theme_minimal() +
  theme(
    plot.title = element_text(size = 16, face = "bold", hjust = 0.5),
    plot.subtitle = element_text(size = 12, hjust = 0.5),
    axis.title = element_text(size = 14, face = "bold"),
    axis.text = element_text(size = 12),
    axis.text.x = element_text(angle = 90, hjust = 1),
    panel.grid.major = element_line(color = "gray", size = 0.5),  # Líneas
de cuadrícula más finas
    panel.grid.minor = element_blank(),  # Eliminar cuadrículas menores
    panel.background = element_rect(fill = "white")
  )
```

#### Conclusión:

Durante la pandemia de COVID-19 la mayor freceuncia de muertes por día que se alcanzó fue de más de 4000 muertes. Además podemos deducir que de media se produjeron diariamente mas de 500 muertes. Esa fue la cifra de muertes más repetida durante la pandemia que asoló al mundo.

## 3) Tercera Representación(Relationships)

En este caso hemos seleccinado el mapa de calor como gráfico para modelar nuestros datos. Vamos a realizar un mapa de calor sobre el la longevidad de los cohes en función de las distintas marcas que encontramos en nuestros datos. No son todas las marcas de coches que existen pero tenemos algunas de las más relevantes del mundo.

Diseño de la visualización

#### Importamos las librerias que vamos a usar

```{r}
library(ggplot2)
library(dplyr)
```
```

#### Importamos los datos extraidos de Kaggle, y los cargamos en R

```{r}
cars <-read.csv("used_car_dataset.csv")
head(cars)
#Tomamos 200 valores de los datos que hemos cargado
data_cars <- cars %>%
  slice(1:200) %>%
  select(used_ages = Age, car_model = Brand, modelo = model, FuelType,
Owner)

#Seleccionamos aquellos con tipo de fuel petrol
data_cars_petrol <- data_cars %>% filter(FuelType == 'Petrol') %>%
filter(Owner == 'first')
head(data_cars_petrol)
```

#### Filtramos los datos y agrupamos los coches por su respectiva marca

```{r}
#Filtamos los datos
data_cars <- data_cars_petrol %>%
  filter(!is.na(used_ages), !is.na(car_model)) %>%
  arrange(used_ages)

# Agrupar los coches por la marca (compañía)
data_cars_grouped <- data_cars %>%
  group_by(car_model) %>%  # Agrupamos por la columna 'Brand'
  summarize(
    avg_used_ages = mean(used_ages, na.rm = TRUE),  # Promedio de la edad
de los coches por marca
    count = n()  # Número de coches por marca
  ) %>%
  arrange(desc(avg_used_ages))  # Ordenar por edad promedio (opcional)

# Ver las primeras filas del nuevo dataframe agrupado
head(data_cars_grouped)
```

#### Realizamos la representación gráfica de los datos

```{r}
ggplot(data_cars_grouped, aes(x = reorder(car_model, avg_used_ages), y = 1,
fill = avg_used_ages)) +
  geom_tile(aes(width = 1, height = 1)) +  # Usamos 'geom_tile' para crear
cuadrados
  scale_fill_gradient(low = "blue", high = "red") +  # Gradiente de colores
más claros
  labs(
    title = "Promedio Coches Gasolina con un único Propietario",
    x = "Marca del Coche",
    y = NULL,  # No necesitamos eje Y para este gráfico
    fill = "Edad Promedio"
  ) +
  theme_minimal() +  # Tema minimalista para un gráfico limpio
  theme(
```

```
    axis.text.x = element_text(angle = 45, hjust = 1),  # Rotar las
etiquetas del eje X para mejorar la legibilidad
    plot.title = element_text(size = 12, face = "bold"),  # Aumentar tamaño
y poner el título en negrita
    legend.title = element_text(size = 10),  # Tamaño del título de la
leyenda
    legend.text = element_text(size = 8),  # Tamaño del texto de la leyenda
    panel.background = element_rect(fill = "lightgray", color =
"lightgray"),  # Fondo del gráfico
    plot.background = element_rect(fill = "white")  # Fondo fuera del panel
  )
```

#### Conclusión:

Los ccoches que tiene una duración mas grande, que han tenído solo un propietario y cuyo combustible es la gasolina son aquellos de la marca Honda, con una duración de 12 años de media.

Es un factor a tener en cuenta si queremos comprar un coche, este mapa de calor tan especifico nos puede guiar a la hora de adquirir un coche el cual queremos tener durante un largo periodo de tiempo.

## 4) Cuarta Representación (Time Series)

En esta ocasión hemos escogido el data day, es decir nuestros datos están basados en ILO Region for Africa.

El diseño de nuestra serie temporal nos permitirá observar la evolución del gasto en trasnporte desde enero del año 2020 hasta septiembre de 2024, en dos regiones de África, Angola y Algeria. Una situada al norte y otra al sur de África.

Diseño de la visualización:

#### Importamos las librerías que vamos a usar

```{r}
library(ggplot2)
#install.packages('extrafont')
library(extrafont) #Paquetes de letras y diseños personalizados
#install.packages("lubridate")
library(lubridate)
library(dplyr)

```

#### Cargamos los datos

```{r}
datos_ilo <- "ILO_Region_Africa.csv"

dt_africa <- read.csv(datos_ilo)
head(dt_africa)
```

#### Filtramos los datos y obtenemos todo los datos de dos unicos paises: Algeria y Angola

```{r}
dt_filtrados <- dt_africa %>%
  filter( ref_area.label == "Algeria" | ref_area.label == "Angola") %>%
  filter(classif1.label == "COICOP2012: 7. Transport")


head(dt_filtrados)
```

#### Vamos a separar los datos en los de Angola y Algeria para crear dos series temporales distintas y ponerlas en la msima gráfica

```{r}
dt_filtrados_angola <- dt_africa %>%
  filter( ref_area.label == "Angola") %>%
  filter(classif1.label == "COICOP2012: 7. Transport")

datos_1<- dt_filtrados_angola$obs_value

dt_filtrados_algeria <- dt_africa %>%
  filter( ref_area.label == "Algeria") %>%
  filter(classif1.label == "COICOP2012: 7. Transport")

datos_2<- dt_filtrados_algeria$obs_value
```

#### Vamos a convertir mis fechas en tipo Date,

```{r}
dt_filtrados_angola$time <- gsub("M", "-", dt_filtrados_angola$time)
# Reemplazar "M" por "-"
dt_filtrados_angola$time <- as.Date(paste0(dt_filtrados_angola$time,
"-01")) # Agregar el día 01 y convertir a Date
print(dt_filtrados_angola$time)

dt_filtrados_algeria$time <- gsub("M", "-", dt_filtrados_algeria$time)
# Reemplazar "M" por "-"
dt_filtrados_algeria$time <- as.Date(paste0(dt_filtrados_algeria$time,
"-01")) # Agregar el día 01 y convertir a Date
print(dt_filtrados_algeria$time)

#datos totales
dt_filtrados$time <- gsub("M", "-", dt_filtrados$time)       # Reemplazar
"M" por "-"
dt_filtrados$time <- as.Date(paste0(dt_filtrados$time, "-01")) # Agregar el
día 01 y convertir a Date
print(dt_filtrados$time)
```

#### Creamos la serie temporal con los datos de Angola

```{r}
ggplot(data = dt_filtrados_angola, aes(x = time, y = obs_value)) +
  geom_line(color = "black",linetype = "dashed", size = 1) +
  scale_y_continuous(breaks = seq(100, 300, 10)) +
  geom_point(colour = "#a50f15", size = .9)  +
```

```
  scale_x_date(date_labels = "%Y", date_breaks = "year") + # Formato de
fechas
  labs(
    title = "Uso del transporte en Angola 2020-2024",
    x = "Años",
    y = "Valor medio de la población"
  )
  #theme_minimal()
```

#### Creamos la serie temporal con los datos de Algeria

```{r}
ggplot(data = dt_filtrados_algeria, aes(x = time, y = obs_value)) +
  geom_line(color = "black", linetype = "longdash",size = 1) +
  #scale_y_continuous(breaks = seq(90, 130, 5)) +
  geom_point(colour = "#a50f15", size = .9)  +
  scale_x_date(date_labels = "%Y", date_breaks = "year") + # Formato de
fechas
  labs(
    title = "Uso del transporte en Algeria 2020-2024",
    x = "Años",
    y = "Valor medio de la población"
  )

```

#### Creamos la gráfica con ambas series

```{r}
ggplot(data = dt_filtrados, aes(x = time, y = obs_value, color =
ref_area.label, linetype = ref_area.label )) +
  geom_line(size = .5) + # Graficar las líneas
  scale_y_continuous(breaks = seq(90, 300, 20)) +
  #geom_point(size = .9, shape = 17) + # Graficar los punto s, con shape 17
convertimos los puntos en rombos
  scale_x_date(date_labels = "%Y", date_breaks = "year") + # Formato de
fechas
  labs(
    title = "Uso del transporte en Angola y Algeria (2020-2024)",
    x = "Años",
    y = "Valor medio de la población",
    color = "País",
    linetype = "País"
  ) +
  scale_color_manual(values = c("Angola" = "red", "Algeria" = "black")) + #
Colores personalizados
  scale_linetype_manual(values = c("Angola" = "solid", "Algeria" =
"solid")) +# Estilos de línea
  theme_minimal()+
  theme(
    legend.key = element_rect(fill = "white", color = "black", size = 1), #
Cuadrado blanco con borde negro para la leyenda
    legend.key.size = unit(2, "lines"), # Ajustar el tamaño del cuadrado en
la leyenda
    legend.title = element_text(face = "bold", size = 12), # Personalizar
título de la leyenda
```

```
    legend.text = element_text(size = 10),# Personalizar el texto de la
leyenda
    panel.grid = element_blank(),  # Eliminar las líneas de cuadrícula
    panel.background = element_rect(fill = "white"),  # Fondo blanco para
el panel
    plot.title = element_text(family = "Georgia", face = "bold", size = 10,
hjust = 0.5)  # Personalización del título
  )
```

#### Análisis del resultado:

Como resultado inmediato de la visualización representada podemos ver que Angola ha invertido mucho más capital por habitante en promover el uso de los diferentes medios de transporte, soblando la inversión. A diferencia de Algeria cuyo crecimiento es casi imperceptible durante el periodo de tiempo estudiado.

## 5) Quinta Representación (Uncertainties)

Para esta última representación hemos elegido el theme day,por lo tanto hemos extraido los datos de una encuesta realizada en Estados Unidos de la página FiveThirtyEight.

Diseño de la visualización: \#### Cargamos las librerias que vamos a usar

```{r}
library(dplyr)
library(ggplot2)
```

#### Leemos los datos

```{r}
polls <- "president_approval_polls.csv"
datos_encuestas <- read.csv(polls)
```

#### Filtramos y agrupamos los datos más relevantes

```{r}
data_summary <- datos_encuestas %>%
  summarise(
    promedio_si = mean(yes, na.rm = TRUE),
    promedio_no = mean(no, na.rm = TRUE),
    error_si = sd(yes, na.rm = TRUE) / sqrt(sum(!is.na(yes))),  # Error
estándar
    error_no = sd(no, na.rm = TRUE) / sqrt(sum(!is.na(no)))
  )
```

#### Creamos un data frame para el gráfico

```{r}
grafico_datos <- data.frame(
  respuesta = c("Sí", "No"),
  porcentaje = c(data_summary$promedio_si, data_summary$promedio_no),
  error = c(data_summary$error_si, data_summary$error_no)
```

```
)
```

#### Creamos un gráfico de barras con barras de error

```{r}
ggplot(grafico_datos, aes(x = respuesta, y = porcentaje, fill = respuesta))
+
  geom_bar(stat = "identity", width = 0.6) +
  geom_errorbar(aes(ymin = porcentaje - error, ymax = porcentaje + error),
width = 0.4,size =1.2) +
  labs(
    title = "Aprobación Presidencial con Barras de Error",
    x = "Respuesta",
    y = "Porcentaje",
    fill = "Respuesta"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
    axis.title = element_text(size = 12),
    legend.position = "none"
  )
```

Obteniendo así un gráfico sobre la aprobación de Joe Biden con una cionfianza casi del 100% sin margen de error.

#### Conclusión:

La aprobación de Biden no es buena según los datos de la encuesta que hemos empleado para diseñar nuestra visualización.