

# ICO 187 ANÁLISIS DE DATOS

## CLASE 30: INTEGRACIÓN POWER BI Y R

Año: 2021  
Profesor: Sebastián Egaña

### 1. Integración entre Power Bi y R.

La finalidad de esta clase es lograr aplicar alguna de las funcionalidades de R y RStudio, e intentar integrarlas con el entorno en Power Bi. Esto solo es aplicable si se tiene instalado Power Bi en el computador (desktop), lo que aplica también para R. Considerar que Power Bi utiliza la instalación de R cuando se realiza dicha labor.

Replicaremos los gráficos de la última clase en Power Bi, utilizando R.

### 2. Graficar usando el set de datos “iris”

Dicho set de de datos, viene pre instalado en R y podemos ver su descripción con el siguiente comando:

```
?iris
```

1. Primero debemos declarar el set de datos utilizado. Para esto obtenemos datos desde “Script de R” en Power Bi. Considerando la lógica de R, declaramos el set de datos a utilizar de la siguiente manera:

```
data_iris <- iris
```

2. Graficamos utilizando la misma lógica de la clase pasada.

Primero, debemos generar la visualización con “Objeto visual de script de R”. Nos aparece un mensaje, para seleccionar las variables a utilizar. En este caso, lo mejor es seleccionarlas todas.

Esto nos obligará a revisar un código que aparece por defecto. Veamos la siguiente declaración:

```
# El código siguiente, que crea un dataframe y quita las filas duplicadas, siempre se ejecuta y actúa como  
  
# dataset <- data.frame(Petal.Width, Petal.Length, Sepal.Length, Sepal.Width, Species)  
# dataset <- unique(dataset)  
  
# Pegue o escriba aquí el código de script:
```

Si analizamos esto, vemos que por defecto Power Bi renombra nuestro set de datos como “dataset” considerando las variables utilizadas y aplicando un filtro para filas duplicadas.

Declaramos el set de datos haciendo las modificaciones necesarias:

```
grafico_0 <- ggplot(dataset)
```

```
grafico_0
```

Declaramos las coordenadas:

```
grafico_1 <- ggplot(dataset) +  
  aes(x = Petal.Length, y = Petal.Width, colour = Species)
```

grafico\_1

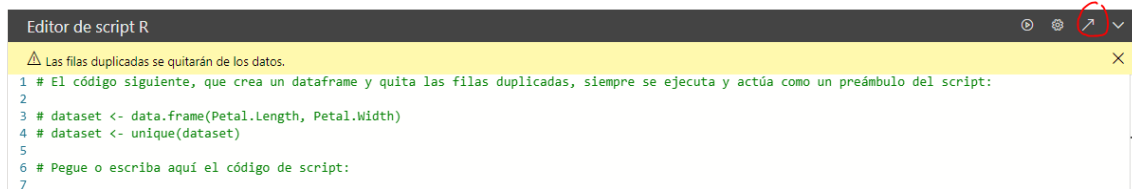
Por último, declaramos el gráfico final considerando el geom

```
library(dplyr)
library(ggplot2)

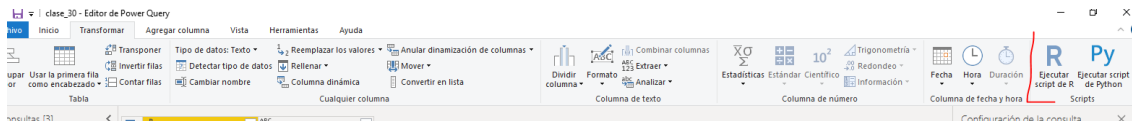
grafico_2 <- ggplot(iris) +
  aes(x = Petal.Length, y = Petal.Width, colour = Species) +
  geom_point()

grafico_2
```

Considerar que cada gráfico y su programación, es independiente. En caso de querer editar nuestro código, podemos hacer un click en la flecha que nos redirige al IDE que tengamos instalado (RStudio)



Si quisieramos modificar la data, debemos entrar a la transformación de datos considerando que lo haremos usando “Ejecutar script de R”.



Generamos una columna de 1 en el set de datos proveniente de “iris”:

```
# 'dataset' contiene los datos de entrada para este script
library(dplyr)

ds <- dataset
ds <- mutate(ds, unos = 1)
dataset <- ds
```

### 3. Graficar usando el set de datos “clase\_27”

1. Importamos el set de datos, considerando que en este caso proviene de una carpeta en nuestro computador. Aprovechamos de generar las variables de fecha de nacimiento con el formato correcto, como también el la variable relacionada con la edad:

```
library(readxl)
library(dplyr)

data_2 <- read_excel("G:/Mi unidad/Docencia 2020 - 2021/UST - 2021 01 Análisis de Datos/document/clase_27.x

# Aplicación pipe

data_2 <- data_2 %>%
  mutate(fecha_nacimiento_2 = as.POSIXct(fecha_nacimiento, format = "%d-%m-%Y"))

# Generar edad

data_2 <- data_2 %>%
```

```
mutate(hoy = as.POSIXct("2021-07-01", format = "%Y-%m-%d"),
       edad = (hoy - fecha_nacimiento_2)/365)
```

2. Insertamos el elemento gráfico, seleccionando las variables a utilizar que por general deben ser todas:

```
# El código siguiente, que crea un dataframe y quita las filas duplicadas, siempre se ejecuta y actúa como
# dataset <- data.frame(afectacion, edad, emergencia_n, fecha_nacimiento, Año, Trimestre, Mes, Día, fibe, A
# dataset <- unique(dataset)

# Pegue o escriba aquí el código de script:
```

Declaramos el gráfico terminado de la clase pasada:

```
library(dplyr)
library(ggplot2)

p_2 <- ggplot(dataset) +
  aes(y = sexo, fill = afectacion) +
  geom_bar(position=position_dodge())

p_2
```

## 4. Fechas Relevantes

Unidad	Evaluación	Ponderación	Fecha
Unidad I	Evaluación diagnóstica		25/03/2021
	Evaluación Individual Participación	(5 %)	05/04/2021
	Evaluación Grupal	(15 %)	27/04/2021 - 04/05/2021
	Evaluación Individual - Sumativa I	(30 %)	11/05/2021
Unidad II	Evaluación Grupal	(15 %)	20/06/2021
	Evaluación Grupal - Sumativa II	(15 %)	04/07/2021
Unidad III	Evaluación Formativa		-
	Evaluación Grupal Sesión I- Sumativa III	(20 %)	08/07/2021
	Evaluación grupal Sesión II- Sumativa III	(20 %)	13/07/2021