**R**

**Contexto**

**Data Science:** es útil para todos. Surge en la 4ta revolución industrial

**Contexto Histórico**

Los **datos** son la *materia prima* de los científicos de datos.

**1784**. **Primera** **Revolución Industrial**: Reino unido, Escocia, con los telares, energía vapor de agua.

**1870**. **Segunda Revolución Industrial** Energía eléctrica

**1969**. **Tercera revolución Industria**: Componentes electrónicos

**Hoy.** **Cuarta revolución Industrial**, IoT, …

Generamos datos por que dejamos huellas de nuestras interacciones en, por ejemplo, redes sociales.

Componentes claves del Big Data: **Volumen**, **Velocidad**, **Variedad**.

**Volumen**: Cuando los datos ya no caben en un Excel, se habla de big data.

**Velocidad:** Interacciones muy rápidas de internet.

**Variedad**: se manejan datos estructurados y no estructurados, como fotos, sentimientos, etc.

**Small Data**: conjunto de datos en un formato y cantidad que los hace accesibles, informativos y procesables.

**Para ser un científico de Datos**: ***Matemáticas, Estadística, Programación, Visualización y Comunicación, Conocimiento del Negocio o Contexto.***

**R**, creado en **1993**, en la Universidad de Auckland en nueva Zelanda. ***Lenguaje especializado para manjar datos de manera estadística y extraer información***, realmente creado en los 70´s en los laboratorios Bell, de AT&T, evolución del lenguaje “S”. ***Más fuerte para datos estructurado***.

Nota: **Python**, más fuerte para datos no estructurados como un Word cloud.

**Proyecto de la economía naranja**: Economías de la industrio cultural y creativa combinadas con la tecnología.

Ejemplo de pregunta: *Si tienes un startup que hace software, video juegos, marketing digital, ¿en qué país abrirías una oficina?*

* A veces hay que crear los propios data set´s.

Por ejemplo, datos importantes relacionados al proyecto: el pib per capita, pib país, crecimiento pib.

* **RStudio es un IDE**

**IDE:** ***Entorno Integrado de Desarrollo***, brinda herramienta para poder escribir código de manera sencilla para administras proyectos.

R puede vivir solo en el pc, pero R Studio no puede vivir sin que R en la computadora.

* Cuando se imprime una variable no se necesita la función print(), solo la variable.
* Usar ‘ ’ y “ “ es lo mismo.
* Ctrl + L, para borrar la pantalla de la consola
* ^ para poner signo: alt+94
* [ para poner el signo alt+91
* ] para poner el signo: alt+93
* Ctrl+ enter, ejecutar la línea previa
* ~ para poner el signo: alt+126

**Jerarquía de operaciones**

* Paréntesis
* Exponentes
* Multiplicación
* División
* Suma
* Resta

**Números pequeños y grandes, notación e**

R cambia los números grandes y chicos a notación e.

>10000000

[1] 1e+07

>0.0000001

[1] 1e-07

La notación e nos dice que cualquier número *a* puede ser expresado como *a e* y, que representa al número *a × 10y*

Para representar número muy grandes o muy pequeños r usa infinitos

>10000^100

[1] Inf

>-10000^100

[1] -Inf

>1/0

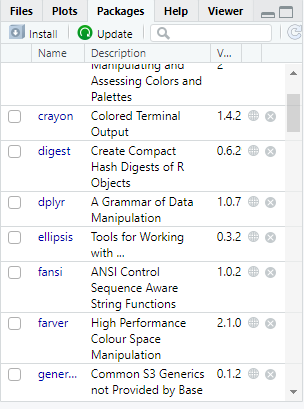
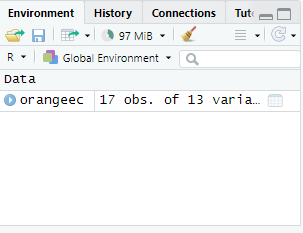
[1] Inf

> -1/0

[1] -Inf

**Importar datos del pc**

*Import Data Set > from Text (base) > seleccionar archivo> “yes” en heading*

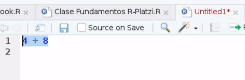
**Paquetes Cargados** **Archivos cargados**

**Dos opciones para usa r**

Desde la consola, poner directo elcódigo



Desde la parte del scrip, seleccionado lo que se quiere ejecutar



**Tipos de datos**

**Numeric:** datos numéricos, son números, esta es una clase que contiene a los enteres y flotantes

**Integer**: enteros, se presentan con una L para indicarle a R que son enteros y no decimales, 2L, 97L…

**Float**: decimales

**Complex**: Números complejos con números imaginarios 1+7i…

**Logical**: lógicos, True o False

**Character**: carácter, son letras, palabras, o frases y van entre comillas, pueden ser números pero si están entre comillas son tratados como carácter.

**Vectores**: valores que están en una definición. **Tienen que ser del mismo tipo de dato** x<-c(n, m, a, s). Entes matemáticos que se usan para guardar datos de cierto tipo, por eso se usan contenedores: c ().

**Listas**: son similares a los vectores pero pueden guardar diferentes tipos de datos.

List(n, 1, 3.5, c(2,3,4))

Se puede ver la **estructura de una columna** con la función class y el signo $,devuelve el tipo de dato de la columna.

Class(nombre\_tabla$nombre\_columna)

**Matrix**: matrices son estructuras de dos dimensiones: columnas y filas. Guardan un solo tipo de dato. Se leen filas por columnas (4\*3) 4 filas y 3 columnas.

Matrix()

**Dataframes:** tablas donde se pueden guardar diferentes tipos de datos.

x <- matrix(1 : 9, byrow = TRUE, nrow = 3)

class(data.frame$columna\_uno)

**Factores**: tipo de dato que tienen variables categóricas, labels o etiquetas, ahora son tipo char.

Nivel\_curso <-c(“Bsico”,”Intermedio”,”Avanzado”)

**Cambio de tipo de dato**

Objeto\_tabla$nombre\_columna = as.tipo\_dato(Objetotabla$nombre\_columna )

Tabla\_uno$columna\_dos = as.logical(Tabla\_uno$columna\_dos)

O

as.tipo\_de\_Dato

as.integer

**Instalar paquetes, ejemplo.**

install.packages(“ggplot2”)

Cuando se valla a usar una librería primero hay que llamarla antes de usarla, siempre:

library(ggplot2)

**Operadores R**

|  |  |
| --- | --- |
| **==** | Igual |
| **!=** | Diferente |
| **<** | Menor que |
| **<=** | Menor o igual que |
| **>** | Mayor que |
| **>=** | Mayor o igual que |
| **|** | O |
| **!** | No |
| **%in%** | Que este en el datasetplyr |

**Funciones**

Las **funciones** en R son aquellas que convierten una entrada o dato, las procesan y arrojan un resultado

Para poder crear una función, hay que conocer su estructura:

Nombre de la función

Argumentos

Cuerpo

Resultado

Nombre\_funcion <- function(argumento.1, argumento.2,...,argumento.n){

...operaciones entre argumentos...

...guardar resultado a mostrar en una variable (var.resultado)... return(var.resultado)

}

**EDA**: análisis exploratorio de datos (exploratory data analysis).

* Importancia de la visualización de los datos antes que en las formulas estadísticas, para tener pistas de los datos.

Estadísticas descriptivas:

* Correlation
* Std. Dev
* Regression

*Concepto estadístico*: **cuarteto de anscomne**: es importante visualizar los datos antes de las formulas estadísticas, cuando las formulas arrojan resultados similares, para eso se usa la visualización.

**Graficas de dispersión e histogramas**

**Histograma**: *Distribución de las frecuencias* de una variable, sus van barras pegadas, el orden en el eje x es de menor mayor, en grafica de barras sí se puede usar cualquier orden.

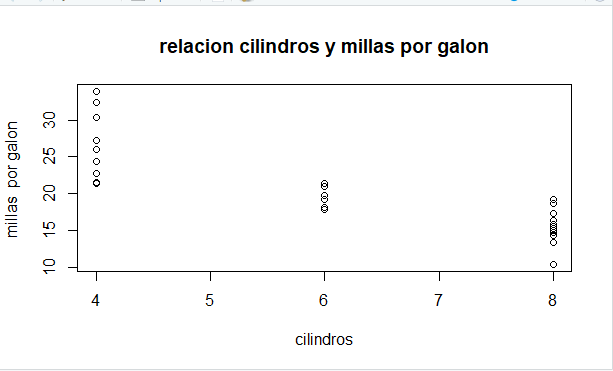
Presenta lo que hay y lo que no hay, hay huecos, en la de barras se grafica lo que hay, en histograma son números en las de barras se pueden usar etiquetas.

**Grafica de dispersión o Scatrerplot**: se mezclan o cruzan variables continuas, o datos numéricos, no etiquetas, palabras ni variables categóricas, los puntos no se pueden unir, como en las gráficas de líneas se pueden unir. En las gráficas de dispersión cada punto obedece a una coordenada (x, y).

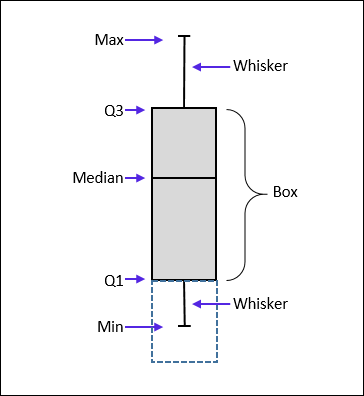
#EDA Scatter plot mtcars

plot(mtcars$mpg ~ mtcars$cyl, xlab="cilindros", ylab= "millas por galon ",

main="relacion cilindros y millas por galon")



**Boxplot**: 5 elementos claves en estadística descriptiva: mínimo, máximo, primer cuartil, mediana o segundo cuartil, y el tercer cuartil.



La media no es igual que el porcentaje de veces que sucede algo

Ejemplo:

|  |  |
| --- | --- |
| Calificación  Servicio |  |
| A | 9 |
| B | 6 |
| C | 8 |
| D | 7 |
| E | 5 |
| F | 4 |
| G | 8 |
| H | 5 |
| I | 6 |
| J | 6 |

* La media es 6 no quiere decir que el 50% tengan 6
* El porcentaje de 6 es 30%

**Para hacer un gráfico en R sin paquetes:**

Plot(F(x) ~ x, xlab=”x”, ylab= “y”)

**Funciones**

install.packages(“nombre\_del\_paquete a instalar")

library(“plyr”) Para poder usar una librería, llamarla

View(nombre\_de\_tabla) Visualizar data set o tablas

str(nombre\_tabla) Describir la tabla

dim(nombre\_tabla) Describir tabla

?nombre\_objeto Devuelve descripción de un objeto

class(nombre\_objeto) Estructura de una variable

sqrt(n) Raíz cuadrada

log(n) Logaritmo natural

exp(n) Exponencial

typeof(nombre\_variable) Tipo de dato

c() Vectores, un solo tipo de datos

list() Listas varios tipos de objetos

Matrix() tables, un solo tipo de datos

sumary() Resumen estadístico de una tabla

data\_set[data\_Set$nombre\_columna<operador>objetivo] Manipulcion de data set

orangeec[orangeeec$GDP.PC<=15000]

subset(data\_set, colmn\_n<operador> <objetivo> <operado> columna\_m operado><objetivo>)

ejeplo:

variable\_guardar\_subset <-subset(orangeec, Interner.penetrarion…population>80 & Education.invest…GDP >=4.5)

con select,se puede definar la columna a mostrar

new\_orangeec <- subset(orangeec, Internet.penetration...population >80

& Internet.penetration...population > 4.5, select = Creat.Ind...GDP )

Select =, solo se trae una columna

Head(nombre\_Dataset) Encabezados y primeras 6 columnas

Tail(nombre\_Dataset) Encabezados y ultimas 6 columnas

De la librería dplyr

Glimpse(nombre\_Dataset) Descripcioón tabla

Plot(F(x) ~ x, xlab=”x”, ylab= “y”)

Plot(mtcars$mpg ~ mtcars$cyl, xlab= “cilindros”, ylab= ”milas por galon”)

**Herramientas:**

The cia worlds factbook, página recomendada para extraer información.

ONU, página recomendada para extraer información.

Banco mundial, página recomendada para extraer información.

INEGI, página recomendada para extraer información.

<https://stackoverflow.com/>

https://support.rstudio.com/hc/en-us/articles/218611977-Importing-Data-with-RStudio