



# Inferencia Estadística

## Ejercicio: EDA2

DSLab

junio, 2024

---

### Ejercicio 1

Crear el siguiente *dataframe* mediante estas instrucciones

```
set.seed(1234)

stocks = data.frame(time = as.Date("2009-01-01") + 0:9,
                    Walmart = rnorm(10,20,1),
                    Target = rnorm(10,20,2),
                    Walgreens = rnorm (10,20,4)
                    )

stocks
```

##	time	Walmart	Target	Walgreens
## 1	2009-01-01	18.79293	19.04561	20.53635
## 2	2009-01-02	20.27743	18.00323	18.03726
## 3	2009-01-03	21.08444	18.44749	18.23781
## 4	2009-01-04	17.65430	20.12892	21.83836
## 5	2009-01-05	20.42912	21.91899	17.22512
## 6	2009-01-06	20.50606	19.77943	14.20718
## 7	2009-01-07	19.42526	18.97798	22.29902
## 8	2009-01-08	19.45337	18.17761	15.90538
## 9	2009-01-09	19.43555	18.32566	19.93945

```
## 10 2009-01-10 19.10996 24.83167 16.25621
```

A continuación, realizar las siguientes operaciones de limpieza de datos:

- Como se puede observar, hay un problema de clave-valor en las compañías con sus observaciones. Por lo tanto, se pide transformar los datos para que tengan una clave "stock" y un valor "precio". Utilizar la instrucción "gather".
- Devolver el dataframe al estado original empleando la instrucción *spread*.
- Utilizando el operador tubería %>% se desea realizar las siguientes operaciones anidadas:
  - Transformar los datos para que tengan una clave "stock" y el valor sea el "precio". Utilizar la instrucción "gather".
  - Agrupar los datos por la clave "stock" mediante la instrucción "group\_by".
  - Obtener el precio mínimo y el máximo utilizando la instrucción "summarise".

## Ejercicio 2

En este ejercicio vamos a manejar datos contenidos en distintos *dataframes* y operar sobre ellos con *dplyr*.

1. Descargar el paquete *nycflights13*.
2. Evaluar el contenido de los dataframes proporcionados por el paquete. Utilizar *head* y *summary*.
3. Simplificar los dataframes originales a 100 observaciones mediante el comando *head*. Asignarlos a una variable que indique el tipo de dataframe añadiendo la co-letilla "\*\_simple". Ejemplo: "flights\_simple".
4. Selecciona los tipos de aerolínea ("carrier") mediante la instrucción *select* y el operador *unique* concatenados con el operador tubería %>%. (Utilizar "airlines\_simple").
5. Obtener la media y el número máximo de asientos ("seats") que tienen los aviones. Utilizar el operador tubería %>% y la instrucción *summarise*.
6. Ordenar los aviones por su número de motores ("engines") y número de asientos ("seats"). Utilizar la instrucción *arrange*.
7. Averigua qué número de cola ("tailnum") comparten los dataframes "flights\_simple" y "planes\_simple" que has creado anteriormente. Obten su aerolínea ("carrier"). Utilizar la instrucción *inner\_join*.
8. Cruzar los datos de vuelos ("flights") con los aviones ("planes") por el número de cola ("tailnum") que no coincidan (usar la instrucción *anti\_join*). De esos obtener aquellos con 2 o más motores(usar la instrucción *filter*). Finalmente obtener los distintos modelos de avión que satisfacen las premisas anteriores (usar la instrucción *unique*).
9. Crea una nueva variable ("total\_delay") que calcule el retraso total sumando los de-

- lays acumulados (*dep\_delay*) y (*arr\_delay*). Utilizar la instrucción *mutate*. Almacena el dataframe resultante en *flights\_total*.
10. En base a la variable anteriormente obtenida (*total\_delay*), devuelve los aviones que han llegado con antelación a su destino, es decir aquellos tal que la variable *total\_delay* tiene valores negativos.