



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA CIVIL Y MECÁNICA
CARRERA DE INGENIERÍA MECÁNICA



- **Estadística**

Nombre: Paul Alexander Barragán Míguez

Nivel: 3° "A"

Docente: PhD. Federico Zertuche

```
library(dplyr, tidyr, readr)
```

```
library(readr)
```

```
clima <- read_csv("~/Estadis/clima.csv")
```

```
View(clima)
```

```
library(readr)
```

```
viajes <- read_csv("~/Estadis/viajes.csv")
```

```
View(viajes)
```

```
library(readr)
```

```
estaciones <- read_csv("~/Estadis/estaciones.csv")
```

```
View(estaciones)
```

#Ejercicio 1 Un poco de R.

Pregunta 1 Cuáles de las siguientes expresiones valen 99 para $x = 10$ en R? Analicen la sintaxis como si estuvieran programando.

```
x <- 10
```

```
10*x-1
```

```
x*x-1
```

```
abs(x*x)-abs(9-x)
```

```
11*x-x+1
```

#En todos los casos dio un valor de 99, menos en el ultimo que dio 101.

#Pregunta 2 Un vector contiene una serie de ganancias ordenadas de manera creciente. Escriban el código que genera:

```
ganancias <- c(200,500,600)
```

#la suma de todas las ganancias

```
sum(ganancias)
```

#La segunda ganancia mas grande

```
ganancias[c(2)]
```

#La diferencia mas grande entre las ganancias

```
ganancias[c(3)]-ganancias[c(1)]
```

#Un booleano que responda a la pregunta: La mas grande diferencia ente dos ganancias es mayor a 10?

```
cummin(ganancias)>10
```

#La menor diferencia positiva entre dos ganancias.

```
ganancias[c(2)]-ganancias[c(1)]
```

#El máximo número de ganancias que pueden sumar sin pasar de 1000

```
cumsum(ganancias)<1000
```

#Ejercicio 2 Dplyr en los Aeorpuertos.

```
library(nycflights13)
```

```
View(weather)
```

```
View(flights)
```

```
View(planes)
```

#Fueron de SFO(San Francisco) hasta OAK(Oakland).

```
SF_OK <- flights %>% filter(origin == 'JFK', dest == 'MIA') %>% View()
```

#Salieron en Enero.

```
enero <- flights %>% filter(month == 1) %>% View()
```

#Tienen demoras de más de una hora (las demoras están en minutos)

```
flights %>% filter(dep_delay > '60') %>% View()
```

#Salieron entre medianoche y las 5 a.m.

```
flights %>% filter(flight_hour > 00 & hour < 5) %>% View()
```

#Tuvieron una demora de llegada 2 veces mas grande que la de salida.

```
flights %>% filter(arr_delay == 2 * dep_delay) %>% View()
```

#Pregunta 4 Lean la ayuda de select(). Escriban 2 formas de seleccionar las dos variables de retraso.

```
flights %>% select(dep_delay, arr_delay) %>% View()
```

```
flights %>% select(dep_delay = 'retraso-de-salida', arr_delay = 'retraso-de-llegada') %>% View()
```

#Pregunta 5 Ordenen la tabla por fecha de salida y tiempo. Cuáles fueron los vuelos que sufrieron las mayores demoras? Cuáles recuperaron la mayor cantidad de tiempo durante el vuelo?

```
flights %>% arrange(dep_time, year, month, day) %>% View()
```

```
flights %>% group_by(origin, dest) %>% summarise(demora = max(air_time)) %>% View()
```

```
Vuelos_que_recuperaron_tiempo <- flights %>% mutate(air_time - arr_delay) %>% View()
```

#Pregunta 6 Calculen la velocidad en mph usando el tiempo (que está en minutos) y la distancia (que está en millas). Cuál fué el avión que voló mas rápido?

```
flights %>% filter(distance, air_time) %>% mutate(distancia_metros = distance * 1609.34, tiempo_horas = air_time * 60) %>% mutate(velocidad = (distancia_metros / tiempo_horas)) %>% View()
```

#Pregunta 7 En dplyr el comando pipeline %>% se lee entonces. Significa:

$x \%>\% f(y) \rightarrow f(x, y).$

Es decir pasa x como primer argumento de f. Qué significan las siguientes líneas de código:

```
flights %>% filter(!is.na(dep_delay)) %>% group_by(date, hour) %>% summarise(delay = mean(dep_delay), n = n()) %>% filter(n > 10)
```

Sale un error que nos dice que no existen columnas de dates y n.

#Pregunta 8Cuál es la destinación que tiene las demoras promedio mas grandes? Cuántos vuelos diarios hay? Cuál es la mejor hora para viajar sin retraso?

```
flights %>% group_by(origin, dest) %>%
```

```
  summarise(demoras = mean(arr_delay, na.rm=TRUE)) %>% View()
```

```
flights %>% group_by(day) %>%
```

```
  summarise(vuelos = sum(day == '1')) %>% View()
```

```
flights %>% group_by(day) %>%
```

```
  summarise(vuelos = sum(day == '2')) %>% View()
```