



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



- **ESTADÍSTICA**

Nombre: Edison Caguana

Nivel: Tercero “A”

Tema: “Programación en R”

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.

- $x <- 10$
- $10x-1$
- $xx-1$
- $\text{abs}(x*x)-\text{abs}(9-x)$
- $11*x-x+1$

Respuesta: En las primeras expresiones el signo (*) se omite por lo que el programa da error, en la tercera y cuarta los signos correspondientes ya se encuentran y se logra realizar las operaciones.

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

```
p <- c(150, 250, 300, 400, 670)
```

- **La suma de todas las ganancias.**

```
sum(p)
```

- **La segunda ganancia mas grande.**

```
g <- length(p)
```

```
p[g-1]
```

- **La diferencia mas grande entre las ganancias.**

```
h <- diff(p)
```

```
max(h)
```

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

```
t <- max(h)
```

```
nchar(t) < 10
```

Respuesta: El valor es verdadero puesto que el booleano cuenta la cantidad de dígitos que tiene el valor máximo.

- **La menor diferencia positiva entre dos ganancias.**

```
j <- diff(p)
```

```
min(j)
```

- **El máximo número de ganancias que pueden sumar sin pasar de 10000.**

```
cumsum(p) < 10000
```

Pregunta 3.- Instalen la librería `nycflights13`. Escriban el código para encontrar todos los vuelos que:

- **Fueron de SFO(San Francisco) hasta OAK(Oakland).**

```
vuelos <- flights %>%
```

```
  select(origin, dest)
```

```
vuelos2 <- mutate(vuelos,
```

```
  dest = factor(origin, labels = c('OAK')),
```

```
  origin = factor(dest, labels = c('SFO')))
```

```
View(vuelos2)
```

- **Salieron en Enero.**

```
vuelos_enero <- flights %>%
```

```
  select(month)
```

```
vuelos_enero2 <- mutate(vuelos_enero,
```

```
  month = factor(1))
```

```
View(vuelos_enero2)
```

- **Tienen demoras de mas de una hora (las demoras están en minutos).**

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

- **Salieron entre medianoche y las 5 a.m.**

```
vuelos_madrugada2 <- filter(vuelos_madrugada, hour <= 5)
```

```
View(vuelos_madrugada2)
```

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

```
flights %>% select(retraso_salida = dep_time, retraso_llegada = arr_time) %>%
```

```
  mutate(diferencia_retrasos = (retraso_llegada - retraso_salida)) %>%
```

```
  filter(retraso_salida == diferencia_retrasos) %>% View()
```

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

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

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

Pregunta 5.

- **Ordenen la tabla por fecha de salida y tiempo:**

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

- **¿Cuáles fueron los vuelos que sufrieron las mayores demoras?**

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

- **¿Cuáles recuperaron la mayor cantidad de tiempo durante el vuelo?**

Vuelos_rt<- 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?**

velocidad <- flights %>% select(air_time, distance) %>%

mutate(tiempo.horas = (air_time/60),

distancia.metros = distance*1609.34)

velocidad %>% filter(tiempo.horas, distancia.metros) %>%

select(tiempo.horas, distancia.metros) %>%

mutate(velocidad.mph = (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:**

a flights %>% filter(! is.na(dep_delay))

b #%>% group_by(date, hour)

c #%>% summarise(delay = mean(dep_delay), n = n())

d #%>% filter(n > 10)

Respuesta. – En los códigos que se muestran en las líneas `b` no corre debido a que no se conoce la columna de `date` en la tabla que tenemos “flights”. En el siguiente el código `d` no se conoce a la columna `n` es decir que no existe.

- **Pregunta 8**

a. ¿Cuál es la destinación que tiene las demoras promedio más grandes?

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

b. ¿Cuántos vuelos diarios hay?

```
flights %>% group_by(day) %>% summarise(vuelos = sum(day == '1')) %>% View()  
flights %>% group_by(day) %>% summarise(vuelos = sum(day == '2')) %>% View()
```