



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA CIVIL Y MECÁNICA
CARRERA DE INGENIERÍA MECÁNICA
“ESTADÍSTICA”



NOMBRE: CHRISTOPHER PUNGUIL

FECHA: 19/10/17

NIVEL: 3^{ro} “A”

TEMA: “PROGRAMACIÓN R”

Pregunta 1.- ¿Cuáles de las siguientes expresiones valen 99 para $x = 10$ en R?

- **$10x - 1$:** Si cumple con la condición.
- **$(x)(x) - 1$:** Si cumple con la condición.
- **$\text{abs}(x*x) - \text{abs}(9-x)$:** Si cumple con la condición.
- **$11 * x - x + 1$:** No cumple con la condición.

Pregunta 2.- Un vector contiene una serie de ganancias ordenadas de manera creciente.

Escriban el código que genera:

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

```
w = c(10, 20, 30, 40, 50, 60, 70, 80, 90, 100)
sum(w[seq(1,length(w),2)])
```
- **La segunda ganancia más grande.**

```
w[c(9)]
```
- **La diferencia más grande entre las ganancias.**

```
w[c(10) - c(1)]
```
- **Un booleano que responda a la pregunta: La más grande diferencia entre dos ganancias es mayor a 10?**

```
cummin(w)>10
```
- **La menor diferencia positiva entre dos ganancias.**

```
w[c(2) - c(1)]
```
- **El máximo número de ganancias que pueden sumar sin pasar de 10000.**

cumsum(w)<1000

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

- **Fueron de JFK hasta OAK(Oakland).**

```
flights %>% filter(origin == 'JFK', dest == "MIA") %>%  
  View()
```

- **Salieron en Enero.**

```
flights %>% filter(month == '1') %>%  
  View()
```

-Código que muestra los vuelos que salieron en Enero.

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

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

- Código que permite comparar todos los vuelos retrasados en minutos que sean mayores a 60 min.

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

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

-Código que permite identificar los vuelos que están en ese rango de horas.

- **Tuvieron una demora de llegada 2 veces más 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(arr_delay, dep_delay, -starts_with("-")) %>%  
  View()
```

-Aquí se pueden observar dos códigos que permiten seleccionar de manera diferente las variables de retraso.

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?**

```
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).**

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

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

$x \%>\% f(y) - f(x; y):$

Es decir, pasa x como primer argumento de f.

Qué significan las siguientes líneas de código:

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

-Este código permite que se filtren los valores de las demoras de salida sin tomar en cuenta los datos que no existen.

flights %>% group_by(date, hour)

-Esta línea de código no corre porque aparece existe un error donde dice que “date” es desconocido es decir no existe en la tabla.

flights %>% summarise(delay = mean(dep_delay), n = n())

-Esta linea de codigo realiza un resumen de los retrasos de despegue.

flights %>% filter(n > 10)

Esta linea de codigo no corre por un error: la comparación (6) es posible solo para tipos atómicos y de lista.

Pregunta 8.-

- **¿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()
```

-La destinación que tiene más demoras es EWR.

- **¿Cuántos vuelos diarios hay?**

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

-Este código permite ver la cantidad de vuelos diarios, pero, hay que cambiar el día para observar la cantidad de vuelos.