

UNIVERSIDAD TÉCNICA DE AMBATO

INGENIERÍA MECÁNICA

NOMBRE: BRYAN GUATO

SEMESTRE: TERCERO "B"

DEBER 1

#EJERCICIO 1 UN POCO DE R

#1.-¿Cuales de las expresiones valen 99 para $x=10$ en R?

#Analice la sintaxis como si estuvieran programando

```
> x <- 10
```

```
> 10*x-1
```

```
[1] 99
```

#Esta función si funciona

```
> x*x-1
```

```
[1] 99
```

#Esta función si funciona

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

```
[1] 99
```

#Esta función si funciona

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

```
[1] 101
```

#Esta función no funciona

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

#Escriba el código que genere:

#La suma de todas las ganancias

```
b<-c(2, 453, 232, 2732, 3023)
```

```
sum (b[seq (1, length (b))])
```

#La segunda ganancia mas grande

```
b[l-1]
```

#La diferencia mas grande entre todas las ganancias

```
mas<-masgrande<-max(b)
```

```
menos<-menosgrande<-min(b)
```

```
d<-sum(mas-menos)
```

```
print(d)
```

#Un booleano que responda a la pregunta:

#La mas grande diferencia entre dos ganancias es mayor a 10?

```
mas<-masgrande<-max(b)
```

```
menos<-menosgrande<-min(b)
```

```
d<-sum(mas-menos)
```

```
print(d)
```

```
d>10
```

#La menor diferencia positiva entre dos ganancias

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

#El máximo numero de ganancias que pueden sumar sin pasar 10000

```
cumsum(b)<10000
```

#EJERCICIO2 DPLYR EN LOS AEROPUERTOS

#Vamos a estudiar los datos de los vuelos locales en Estados Unidos en el 211

#Instalen la libreria nycflights y escriba los cogios para encontrar:

#Fueron de Sfo(San francisco) hasta OAK(Oakland)

```
flights %>% filter(origin=='SFO', dest=='OAK') %>%
```

```
View()
```

#Salieron en enero

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

```
View()
```

#Tienen demoras de mas de 1 hora

```
flights %>% filter(dep_delay>=1) %>%
```

```
View()
```

#Salieron a media noche y 5 am

```
flights %>% filter(hour==24, hour==5) %>%
```

```
View()
```

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

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

```
View()
```

#4.-Escriba dos formas de seleccionar dos variables de retraso

```
flights %>% select(dep_delay)
```

```
flights %>% select(arr_delay, -starts_with("-"))
```

#5.- Ordenen la tabla por fecha de salida y tiempo

#Cuales fueron los vuelos que sufrieron mayores demoras

#Cuales recuperaron la mayor cantidad de tiempo durante el vuelo

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

```
View()
```

#6.- Calculen la velocidad en mph usando el tiempo y la distancia

#Cual fue el avión que voló mas rápido

```
flights %>% filter(air_time, distance) %>%
```

```
mutate(Velocidad=air_time*distance, air_time=air_time/60) %>%
```

```
View()
```

#7.- En dplyr el comando pipeline se lee “entonces”

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

#Filtrar valores de elementos en los retrasos de salida que se están perdiendo

```
group_by(date, hour)
```

#Indica los valores agrupados entre la fecha y la hora

```
summarise(delay=mean(dep_delay, n=n()))
```

#Hace un resumen del promedio de los retrasos

```
filter(n > 10)  Error in n > 10 : comparison (6) is possible only for  
atomic and list types
```

#En esta función apareció error

#8.- Cual es la destinación que tiene las demoras promedio mas grandes

#Cuantos vuelos diarios hay

#cual es la mejor hora para viajes sin retrasos

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

```
summarize(mean(arr_delay, na.rm=TRUE)) %>%
```

```
View()
```