

```
library(tidyverse)
```

```
library(tidyr)
```

```
library(dplyr)
```

```
#ejercicio 1
```

```
#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
```

```
(x)(x) - 1
```

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

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

```
# la primera ecuacion tiene un error falta de un simbolo "10x"
```

```
# la segunda funcion tiene error no reconoce r la funcion
```

```
# la tercera funcion es la solucion su operacion suma 99.
```

```
# La cuarta no es solucion suma 101.
```

## #Pregunta 2

#Un vector contiene una serie de ganancias ordenadas de manera

#creciente. Escriban el código que genera:

```
g <-c(1,12,43,54,69,77,85,94)
```

#suma total

```
sum(g)
```

#La segunda ganacia mas grande

```
x <- length(g)
```

```
g[x-1]
```

#la diferencia mas grande entre lasganacias

```
g <-c(1,12,43,55,69,77,85,98)
```

```
max(diff(g))
```

#Un booleano que responda a la pregunta:

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

```
y <- max(g)
```

```
nchar(y) < 10
```

```
#es verdadero porque un booleano lo que hace es contar el numero de digitos  
# en una cifra en nuestro caso no supera los 10 digitos por lo cual es verdadero
```

```
#La menor diferencia positiva entre dos ganancias.
```

```
min(diff(g))
```

```
#El máximo número de ganancias que pueden sumar sin pasar de 10000.
```

```
cumsum(g) < 10000
```

```
# ejercicio 2
```

```
#Vamos a estudiar los datos de los vuelos locales en Estados Unidos durante el 2011.
```

```
#Usen los verbos:
```

```
library(nycflights13)
```

```
View(planes)
```

```
View(weather)
```

```
View(flights)
```

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

```
viajes <- select(flights, c(origin , dest))
```

```
View(viajes)
```

```
viajes_completos <- mutate(viajes, origin = factor('SFO'), dest = factor('OAK') )
```

```
View(viajes_completos)
```

```
#Salieron en Enero
```

```
viajes_enero <- select(flights,c(month))
```

```
View(viajes_enero)
```

```
viajes_mes1 <- mutate(viajes_enero, month = factor(1))
```

```
View(viajes_mes1)
```

```
#Tienen demoras de mas de una hora (las demoras están en minutos).
```

```
demora <- select(flights, flight, dep_delay, arr_delay) %>%
```

```
mutate(demora = dep_delay + arr_delay) %>%
```

```
filter(demora > 60)
```

```
View(demora)
```

```
# Salieron entre medianoche y las 5 a.m
```

```
vuelos_nocturnos <- select(flights, c('hour'))
```

```
View(vuelos_nocturnos)
```

```
vuelos_nocturnos2 <- filter(vuelos_nocturnos, hour <= 5 )
```

```
View(vuelos_nocturnos2)
```

```
# Existe un solo vuelo a la 1 de la mañana se puede deducir que fue
```

```
#un evento unico ya que no hay vuelos a partir de las 12 de la noche ya no existen
```

```
# en la tabla de datos empiezan a las 5 de la mañana.
```

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

```
VUelos_salida_llegada <- select(flights, c('dep_time','arr_time'))
```

```
View(VUelos_salida_llegada)
```

```
#Pregunta 4 Lean la ayuda de select().
```

```
#Escriban 2 formas de seleccionar las dos variables de retraso.
```

```
help("select")
```

```
#Primera forma de seleccionar con select()
```

```
variables <- select(flights, c('dep_time','arr_time'))
```

```
flights %>% select(dep_time, arr_time) %>% 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?

#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:  $x \rightarrow f(y)$  o  $f(x,y)$ . Es decir pasa x como primer argumento de f.

# Qué significa 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)
```

# selecciona la tabla flights luego filtra filas todas menos dep\_delay

# agrupa entre date y hour como primer argumento del filter posteriormente realiza un summarise

# el cual saca el promedio como argumento de group\_by y por ultimo selecciona los numeros mayores a 10

# W como argumento de summarize