#Deber de estadistica.

library(dplyr)

library(tidyr)

library(readr)

#Ejercicio 1

#pregunta 1

#Cuáles de las siguientes expresiones valen 99 para x=10.

#en esta expresion el codigo es invalido ya que falta el simbolo de multiplicacion.

x<-10

De la misma forma el simbolo de multiplicacion falta en esta expresion.

10x-1

Nuevamente el simbolo de multiplicacion es inexistente.

(x)(x)-1

En vale 99 ya que el codigo consta de todos los simbolos necesarios.

abs(x*x)- abs(x*x)

Esta expresion no cumple ya que no es igual al valor deseado.

11*x-x+1

#Pregunta 2

Un vector contiene una serie de ganancias ordenadas de manera creciente.

a<-c(12,34,65,78,98,456,4567,2659,6437,9753,12897)

#la suma de todas las ganancias.

sum(a)

#la segunda ganancia mas grande.

length(a)

a[10]

#la diferencia mas grande entre las ganancias.

diferencia <- a[11]-a[1]

#un booleano que responda a la pregunta.

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

diferencia >10

#La menor diferencia positiva entre dos ganancias.

diferencia<- a[2]-a[1]

#El maximo numero de ganancias que puedan sumar sin pasar de 10000.

cumsum(a)<10000

#EJERCICIO 2

library(nycflights13)

#Pregunta3. Escriban el codigo para encontrar todos los vuelos que:

#fueron de San Francisco hasta Oakland.

View(flights)

vuelos <- select(flights, origin, dest)

viajes<-mutate(vuelos,

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

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

View(viajes)

#salieron en enero.

salida <- select(flights, flight, month)%>%

filter(month == 1)

View(salida)

 # Tienen demora de mas de una hora (en minutos)

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

mutate(retraso = dep_delay+arr_delay)%>%

```
  filter(retraso >60)



View(retraso)

 # Salieron entre medianoche y las 5 a.m.

medianoche <- select(flights, flight, horadesalida =hour)%>%
  filter(horadesalida <=5)%>%
filter(horadesalida >=0)

View(medianoche)

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

demora <- select(flights,flight, dep_delay, arr_delay)%>%
mutate(eldoble_de_la_demora_de_llegada_mayora_la_de_salida = arr_delay/2>dep_delay)
View(demora)

# Pregunta 4.Lean la ayuda de select().
#Escriban 2 formas de selecionar dos variables de retraso.
?select()
#primera forma
retraso1<- select(flights,dep_delay, arr_delay)
View(retraso1)
# segunda forma
retraso2 <-select(flights,  dep_delay, "arr_delay")
View(retraso2)

# Pregunta 5. Ordene la tabla por fecha de salida y tiempo.
length(flights)
```

```
orden <- arrange(flights, time_hour, hour, minute, dep_time, sched_dep_time,
        dep_delay, arr_time, sched_arr_time, arr_delay, carrier, flight, tailnum,
        origin, dest, air_time, distance, year, month, day)
View(orden)
```

 # Cuáles fueron los vuelos que sufrieron las mayores demoras?

```
mayordemora <- select(flights, flight, dep_delay, arr_delay)%>%
  mutate(demora = dep_delay+arr_delay )
  mutate(programado = sched_dep_time+sched_arr_time)%>%
  filter("demora" < "programado")
View(mayordemora)
```

 # Cuáles recuperaron la mayor cantidad de tiempo durante el vuelo?

```
tiempodevuelo<- select(flights, flight, dep_delay, arr_delay)%>%
  mutate(llego_con_ventaja_antes_de_lo_programado = dep_delay+arr_delay)%>%
  filter(llego_con_ventaja_antes_de_lo_programado < 0)
View(tiempodevuelo)
```

#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?

```
conversion <- flights %>% select("air_time", "distance") %>%
   mutate(tiempo_en_horas = (air_time/60), distancia_en_metros = distance*1609.34)

velocidad<- conversion %>% filter(tiempo_en_horas, distancia_en_metros) %>%
  select(tiempo_en_horas, distancia_en_metros) %>%
  mutate(velocidad_en_mph = (distancia_en_metros / tiempo_en_horas))
```

View(velocidad)

 # 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))
    %>% group_by(date, hour)
    %>%summarise(delay = mean(dep_delay), n = n())
    %>% filter(n > 10)
```

De la tabla flights va a filtrar los datos na en la columnna dep delay

#con la función '!is.na ()', que devolvería False si el valor es NA y True de lo contrario.

#despues este filtardo pasa como primer argumento y agrupa las columnnas en comin como date y hour

#despues al agrupacion pasa como primer argumento y se realiza una resumen de todas las filas en la columnna dep_delay

#el resumen pasa como primer argumento y se reliza otro filtrado solo de los valores mayores a 10

Lo que hace el codigo es filtrar el retraso de los vuelos con la fecha y hora en que salio y que tienen que sr mayores de 10

Pero no funciona ya que el codigo no incluye la seleccion de la columnna dep_delay de la tabla flights y en el inicio se intenta filtrar los valores na de dep_delay los cuales no exixten en la tabla.

 # Pregunta 8

Cuál es la destinación que tiene las demoras promedio mas grandes?

 # Cuántos vuelos diarios hay?

```
dia<-select(flights, flight, day)%>%
      filter(day == 1)
```

```
View(dia)
```

 # Cuál es la mejor hora para viajar sin retraso?

```
horaadecuada<- select(flights, flight, dep_delay, arr_delay, hour, minute)%>%
  mutate(llegada_con_una_anticipacion = dep_delay+arr_delay)%>%
  filter(llegada_con_una_anticipacion < 0)

View(horaadecuada)
```