

DEBER 1

ESTADISTICA

KEVIN CANO 3”A”

EJERCICIO 1

Pregunta 1

- $10 * X - 1$
[1] 99
- $X * X - 1$
[1] 99
- $\text{abs}(X * X) - \text{abs}(9 - X)$
[1] 99
- $11 * X - X + 1$
[1] 101

Pregunta 2

$x = c(1300, 1400, 1500, 2000, 3000)$

- $\text{sum}(x)$
[1] 9200
- $x[c(4)]$
[1] 2000
- $x[c(5)] - x[c(1)]$
[1] 1700
- $\text{cummin}(x) > 10$
[1] TRUE TRUE TRUE TRUE TRUE
- $x[c(2)] - x[c(1)]$
[1] 100
 $x[c(3)] - x[c(2)]$
[1] 100
- $\text{cumsum}(x) < 1000$
[1] FALSE FALSE FALSE FALSE FALSE

EJERCICIO 2

```
library(nycflights13), library(dplyr), library(tidyr)
View(flights)
```

Pregunta 3

- `flights %>% filter(origin=='SFO', dest=='OAK') %>% View()`
- `flights %>% filter(month=='1') %>% View()`
- `flights %>% filter(dep_delay>=1) %>% View()`
- `flights %>% filter(hour==24, hour==5) %>% View()`
- `flights %>% filter(arr_delay==2*dep_delay) %>% View()`

Pregunta 4

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

Pregunta 5

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

Pregunta 6

- `flights %>% filter(air_time, distance) %>% mutate(Velocidad=air_time*distance, air_time=air_time/60) %>% View()`

Pregunta 7

- `flights %>% filter(! is.na(dep_delay))` Filtra los 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)` En esta función apareció error. Error in n > 10 : comparison (6) is possible only for atomic and list types

Pregunta 8

- `flights %>% group_by(dest, arr_delay) %>% summarize(mean(arr_delay, na.rm=TRUE)) %>% View()`

EJERCICIO 3

Pregunta 9

- `TODO=alldata`
`View(TODO)`
- `Países=countries_of_the_world`
`View(Países)`
- `TODO$'pop'`
`PoblacionTodo=TODO %>% select(Poblacion='pop') %>%`
`mutate(Poblacion=gsub(".*$", "", Poblacion)) %>%`
`group_by(Poblacion) %>%`
`summarize(PoblacionTotal=n())`
`View(PoblacionTodo)`
- `PoblacionPaíses=Países %>% select(Poblacion=`X__2`,`
`Continente=`X__1`) %>%`
`filter(Continente=="LATIN AMER. & CARIB") %>%`
`group_by(Poblacion)`
`View(PoblacionPaíses)`
- `dim(PoblacionPaíses)`
`[1] 45 2`
`dim(PoblacionTodo)`
`[1] 2460 2`
- `PaísesTodo=inner_join(PoblacionPaíses, PoblacionTodo, by="Poblacion")`
`View(PaísesTodo)`

La tabla sale error porque no hay datos válidos para SUDAMERICA.

- Podemos calcularlo pero a mi parecer ninguna de las 2 tablas posee datos de antigüedad.
- `Países$X__6`

```
PatronMigratorio=Países %>% select(Migracion='X__6') %>%
mutate(Migracion=gsub(".*$", "", Migracion)) %>%
group_by(Migracion) %>%
summarise(mean(Migracion, na.rm=TRUE)) %>%
View()
```

Tenemos la lista de datos de migración pero no pude calcular el patron migratorio y a que el pasar de los años aumenta o disminuye dependiendo de la época.