

Laboratorio 5

Juan Diego Sique

23 de septiembre de 2020 AD

Primer inciso

Para calcular el resultado hice unas dimensiones propias. Luego las sumé.

```
## Primer inciso
# Ignorando zona horaria que no es importante en fenómenos astronómicos
# Creando la fecha base
x <- '21 de agosto del 2017 a las 18:26:40'
x <- dmy_hms(x)

# Creando las dimensiones astronómicas medievales
synodic <- ddays(29) + dhours(12) + dminutes(44) + dseconds(3)

saros <- 223*synodic

# Fecha final
pred <- x + saros

resp <- paste0("La fecha del siguiente eclipse es ", pred, sep=" ")

resp

## [1] "La fecha del siguiente eclipse es 2035-09-02 02:09:49 "
```

Segundo inciso

1. El mes en que hay más llamadas por código es: **Marzo, con 497 llamadas**

```
data <- read.csv("data.csv", sep=";")

data$`Fecha.Creación` <- dmy(data$`Fecha.Creación`)

data$`Fecha.Final` <- dmy(data$`Fecha.Final`)

primer <- data %>% group_by(month(Fecha.Creación), Cod) %>% summarise(llamadas = sum(Call)) %>% filter(

## `summarise()` regrouping output by 'month(Fecha.Creación)' (override with `.groups` argument)
primer <- primer %>% rename(mes = `month(Fecha.Creación)`)
head(primer, 1)

## # A tibble: 1 x 3
## # Groups:   mes [1]
```

```
##      mes Cod                llamadas
##      <dbl> <chr>                <int>
## 1      3 Actualización de Información    497
```

2. El día más ocupado es: **Domingo**

```
segundo <- data %>% group_by(wday(Fecha.Creación, label = TRUE)) %>% summarise(llamadas = sum(Call), correo = sum(Emisor))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
head(segundo, 1)
```

```
## # A tibble: 1 x 5
##   `wday(Fecha.Creación, label = TRUE)` llamadas correo mensaje total
##   <ord>                <int> <int> <int> <int>
## 1 "dom\\\"."                796   9124  28334 38254
```

3. El mes más ocupado es: **Marzo**

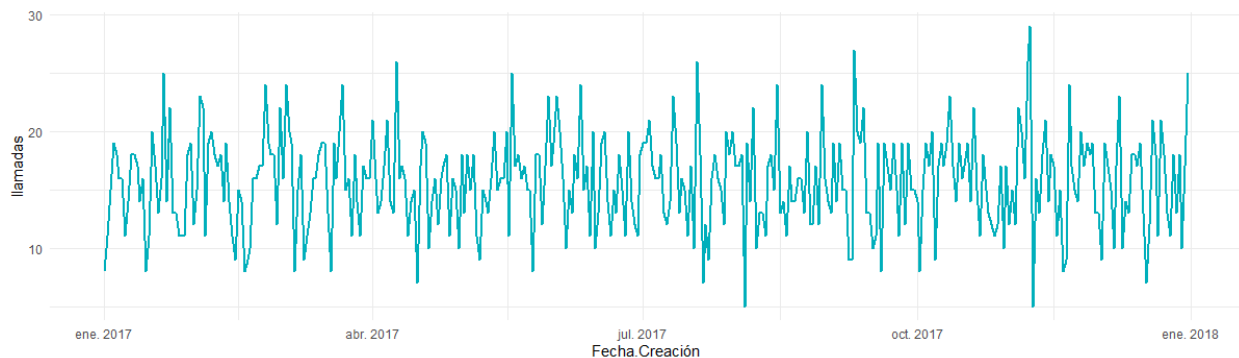
```
tercero <- data %>% group_by(month(Fecha.Creación)) %>% summarise(llamadas = sum(Call), correo = sum(Emisor))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
head(tercero, 1)
```

```
## # A tibble: 1 x 5
##   `month(Fecha.Creación)` llamadas correo mensaje total
##   <dbl> <int> <int> <int> <int>
## 1      3      497   5448  16763 22708
```

4. Sí se puede apreciar una especie de estacionalidad, donde cada dos meses hay un pico a la baja. Si se tuviera datos de más años sería genial para poder comparar estacionalidad durante el año y no sólo mes a mes.



5. Una llamada promedio dura casi 8 minutos. (7.8 aproximadamente.)

```
data$Día.Creación <- paste(data$Fecha.Creación, data$Hora.Creación)
data$Día.Creación <- parse_date_time(data$Día.Creación, c('%Y-%m-%d %I:%M %p'))
data$Día.Final <- paste(data$Fecha.Final, data$Hora.Final)
data$Día.Final <- parse_date_time(data$Día.Final, c('%Y-%m-%d %I:%M %p'))
```

```
quinto_data <- data %>% filter(Call > 0)
```

```
quinto <- mean(
  difftime(quinto_data$Día.Final, quinto_data$Día.Creación, units = 'mins')
)
```

```
quinto

## Time difference of 7.766638 mins

6. De la tabla excluí los valores negativos que eran errores de fecha probablemente.
sexto <- quinto_data
sexto$tiempo_de_llamada <- difftime(sexto$Día.Final, sexto$Día.Creación, units = 'mins')
sexto <- sexto %>% group_by(tiempo_de_llamada) %>% summarise(llamada = sum(Call)) %>% arrange(tiempo_d

## `summarise()` ungrouping output (override with `.groups` argument)
sexto

## # A tibble: 31 x 2
##   tiempo_de_llamada llamada
##   <drtn>                <int>
## 1 0 mins                221
## 2 1 mins                211
## 3 2 mins                173
## 4 3 mins                195
## 5 4 mins                193
## 6 5 mins                184
## 7 6 mins                193
## 8 7 mins                196
## 9 8 mins                209
## 10 9 mins              165
## # ... with 21 more rows
```

Tercer inciso

Usé una librería que ya me daba el resultado zodiacal a partir de una fecha. Para ello es necesario usar la siguiente función.

```
library(DescTools)

calcular_zodiaco <- function() {
  variable_entrada <- readline(prompt="Ingrese su fecha de nacimiento (Día-Mes_Año): ")
  fecha_entrada <- dmy(variable_entrada)
  zodiaco_salida <- Zodiac(fecha_entrada)
  print(zodiaco_salida)
}
```

Para ejecutarla basta con hacer el siguiente llamado.

```
calcular_zodiaco()
```

El resultado sería el siguiente:

```
Ingrese su fecha de nacimiento (Día-Mes_Año): 10 de octubre de 1997
[1] Libra
```

Cuarto inciso

Primero filtré para obtener los datos válidos. Luego la suma del resultado del retraso total lo obtuve a partir de dos columnas con las que ya contaba el dataset.

```
## Filtrando vuelos con fechas y horas válidas
vuelos <- flights %>% filter(!is.na(dep_time), !is.na(arr_time), dep_time >= 100, arr_time >= 100)
```

```
vuelos$Hora_salida <- ymd_hm(paste(vuelos$year, vuelos$month, vuelos$day, vuelos$dep_time %/% 100, ':'  
vuelos$Hora_llegada <- ymd_hm(paste(vuelos$year, vuelos$month, vuelos$day, vuelos$arr_time %/% 100, ':'  
vuelos$Hora_salida_estimada <- ymd_hm(paste(vuelos$year, vuelos$month, vuelos$day, vuelos$sched_dep_tim  
vuelos$Hora_llegada_estimada <- ymd_hm(paste(vuelos$year, vuelos$month, vuelos$day, vuelos$sched_arr_tin  
vuelos$Retraso = vuelos$dep_delay + vuelos$arr_delay
```