

# Lab 5

Hans Walter

9/27/2020

#Librerías

```
library(tidyverse)
library(dplyr)
library(readxl)
library(lubridate)
library(ggplot2)
```

## Ejercicio 1

En tiempo de Norte América, el eclipse total inició el 21 de agosto del 2017 a las 18:26:40. Este mismo evento, sucederá un Saros después. Un Saros equivale a 223 Synodic Months. Un Synodic Month equivale a 29 días con 12 horas, con 44 minutos y 3 segundos.

```
initial_se <- dmy_hms("21 Aug 2017 18:26:40")
synodic_month <- days(29) + hours(12) + minutes(44) + seconds(3)
saros <- synodic_month * 223
new_se <- initial_se + saros
new_se
```

```
## [1] "2035-09-02 02:09:49 UTC"
```

## Ejercicio 2

### Limpieza de Dataset

#### Hora Creación

```
df <- readxl::read_excel("C:/Users/Hans/Documents/UFM/Año 3 Semestre 6/Data Wrangling/data_lab5.xlsx")
vector1 <- hour(df$`Hora Creación`)
vector2 <- minute(df$`Hora Creación`)
vector3 <- second(df$`Hora Creación`)
vector_f <- paste(vector1,vector2, sep=":")
vector_f2 <- paste(vector_f, vector3, sep=":")
df$`Hora Creación` <- hms::as_hms(vector_f2)
```

## Hora Final

```
vector1 <- hour(df$'Hora Final')
vector2 <- minute(df$'Hora Final')
vector3 <- second(df$'Hora Final')
vector_f <- paste(vector1,vector2, sep=":")
vector_f2 <- paste(vector_f, vector3, sep=":")
df$'Hora Final' <- hms::as_hms(vector_f2)
df2 <- df
```

## Fecha Creación

```
df2modified <- df2[grepl("-",df2$'Fecha Creación', invert = TRUE),]
df2modified$'Fecha Creación' <- as.numeric(df2modified$'Fecha Creación')
df2modified$'Fecha Creación' <- format(as.Date(df2modified$'Fecha Creación', origin = "1899-12-30"), '%d-%m-%Y')
df2modified$'Fecha Final' <- as.numeric(df2modified$'Fecha Final')
df2modified$'Fecha Final' <- format(as.Date(df2modified$'Fecha Final', origin = "1899-12-30"), '%d-%m-%Y')
```

## Dataset Limpio

```
df <- df[grepl("-",df2$'Fecha Creación'),]
df_final <- rbind(df, df2modified)
df_final$'Fecha Creación' <- as_date(dmy(df_final$'Fecha Creación'))
df_final$'Fecha Final' <- as_date(dmy(df_final$'Fecha Final'))
```

## Inciso 1

¿En qué meses existe una mayor cantidad de llamadas por código?

```
llamadas <- df_final %>%
  .[grepl(1, df_final$Call),] %>%
  mutate(mes = format('Fecha Creación', "%m")) %>%
  group_by(mes) %>%
  summarise(total = n()) %>%
  arrange(-total)
```

En el mes de marzo

## Inciso 2

¿Qué día de la semana es el más ocupado?

```
dia_ocupado <- df_final %>%
  mutate(dia= wday('Fecha Creación')) %>%
  group_by(dia) %>%
  summarise(total = n()) %>%
  arrange(-total)
```

El día lunes es el día más ocupado.

### Inciso 3

¿Qué mes es el más ocupado?

```
mes_ocupado <- df_final %>%  
  mutate(mes = format('Fecha Creación', "%m")) %>%  
  group_by(mes) %>%  
  summarise(total = n()) %>%  
  arrange(-total)
```

El mes más ocupado es el mes de marzo

### Inciso 4

¿Existe una concentración o estacionalidad en la cantidad de llamadas?

```
mes_ocupado <- df_final %>%  
  mutate(mes = format('Fecha Creación', "%m")) %>%  
  group_by(mes) %>%  
  summarise(total = n()) %>%  
  arrange(-total)
```

Vemos que el período entre los meses de Marzo y Julio existe una estacionalidad. Se puede decir que es entre primavera y verano.

### Inciso 5

¿Cuántos minutos dura la llamada promedio?

```
calls <- df_final[grepl(1, df_final$Call),]  
calls$duracion <- interval(calls$'Hora Creación', calls$'Hora Final', ) %/% minutes(1)  
calls2 <- calls[calls$duracion<0,]  
calls <- calls[calls$duracion>=0,]  
calls2$duracion <- calls2$duracion + 1440  
calls_final <- rbind(calls, calls2)  
promedio_llamadas <- mean(calls_final$duracion)  
promedio_llamadas
```

```
## [1] 14.5579
```

Las llamadas duran en promedio 14 minutos.

### Inciso 6

Tabla de Frecuencias

```
tabla_frecuencia <- as.data.frame(table(calls_final$duracion))  
colnames(tabla_frecuencia) <- c("Duración min", "Frecuencia")  
tabla_frecuencia
```

##	Duración min	Frecuencia
## 1	0	221
## 2	1	211
## 3	2	173
## 4	3	195
## 5	4	193
## 6	5	184
## 7	6	194
## 8	7	197
## 9	8	212
## 10	9	166
## 11	10	190
## 12	11	197
## 13	12	169
## 14	13	163
## 15	14	203
## 16	15	188
## 17	16	181
## 18	17	178
## 19	18	186
## 20	19	190
## 21	20	179
## 22	21	205
## 23	22	175
## 24	23	192
## 25	24	186
## 26	25	174
## 27	26	157
## 28	27	173
## 29	28	158
## 30	29	171
## 31	30	164

### Por Intervalos de 5 minutos

```
breaks <- seq(0, 30, by = 5)
duracion.cut = cut(calls_final$duracion, breaks, right=TRUE)
intervalos <- as.data.frame(table(duracion.cut))
colnames(intervalos) <- c("Duración min", "Frecuencia")
intervalos
```

##	Duración min	Frecuencia
## 1	(0,5]	956
## 2	(5,10]	959
## 3	(10,15]	920
## 4	(15,20]	914
## 5	(20,25]	932
## 6	(25,30]	823

## Ejercicio 3

```
zodiac <- function(){  
  print("Porfavor, la fecha que naciste. Empezando con el año, seguido por el mes y el día.  
  Por ejemplo  
    si naciste el 18 de diciembre de 1998, se escribiría '19981218'")  
  fecha = scan(what = character(), nmax=1)  
  fecha = ymd(fecha)  
  dp <- yday(fecha)  
  if (dp >= yday(ymd(20200120)) & dp <= yday(ymd(20200218))){  
    print("Aquarius")  
  }  
  else if (dp >= yday(ymd(20200219)) & dp <= yday(ymd(20200320))){  
    print("Pisces")  
  }  
  else if (dp >= yday(ymd(20200321)) & dp <= yday(ymd(20200419))){  
    print("Aries")  
  }  
  else if (dp >= yday(ymd(20200420)) & dp <= yday(ymd(20200520))){  
    print("Taurus")  
  }  
  else if (dp >= yday(ymd(20200521)) & dp <= yday(ymd(20200620))){  
    print("Gemini")  
  }  
  else if (dp >= yday(ymd(20200621)) & dp <= yday(ymd(20200722))){  
    print("Cancer")  
  }  
  else if (dp >= yday(ymd(20200723)) & dp <= yday(ymd(20200822))){  
    print("Leo")  
  }  
  else if (dp >= yday(ymd(20200823)) & dp <= yday(ymd(20200922))){  
    print("Virgo")  
  }  
  else if (dp >= yday(ymd(20200923)) & dp <= yday(ymd(20201022))){  
    print("Libra")  
  }  
  else if (dp >= yday(ymd(20201023)) & dp <= yday(ymd(20201121))){  
    print("Scorpio")  
  }  
  else if (dp >= yday(ymd(20201122)) & dp <= yday(ymd(20201221))){  
    print("Saggitarius")  
  }  
  else (  
    print("Capricorn")  
  )  
}
```

## Ejercicio 4

```

library(nycflights13)
flights <- nycflights13::flights
flights$dep_time <- hms::parse_hm(format(strptime(sprintf('%04d',
                                                         flights$dep_time), format='%H%M'), '%H:%M'))
flights$arr_time <- hms::parse_hm(format(strptime(sprintf('%04d',
                                                         flights$arr_time), format='%H%M'), '%H:%M'))
flights$sched_dep_time <- hms::parse_hm(format(strptime(sprintf('%04d',
                                                         flights$sched_dep_time), format='%H%M'))
flights$sched_arr_time<- hms::parse_hm(format(strptime(sprintf('%04d',
                                                         flights$sched_arr_time), format='%H%M'),
flights_final <- flights

flights_final$dep_time_FH <- make_datetime(year = flights_final$year,
                                           month = flights_final$month,
                                           day = flights_final$day,
                                           hour = hour(flights$dep_time),
                                           min = minute(flights$dep_time), sec = second(flights$dep_time))

flights_final$arr_time_FH <- make_datetime(year = flights_final$year,
                                           month = flights_final$month,
                                           day = flights_final$day,
                                           hour = hour(flights$arr_time),
                                           min = minute(flights$arr_time), sec = second(flights$arr_time))

flights_final$sched_dep_time_FH <- make_datetime(year = flights_final$year,
                                                  month = flights_final$month,
                                                  day = flights_final$day,
                                                  hour = hour(flights$sched_dep_time),
                                                  min = minute(flights$sched_dep_time),
                                                  sec = second(flights$sched_dep_time))

flights_final$sched_arr_time_FH <- make_datetime(year = flights_final$year,
                                                  month = flights_final$month,
                                                  day = flights_final$day,
                                                  hour = hour(flights$sched_arr_time),
                                                  min = minute(flights$sched_arr_time),
                                                  sec = second(flights$sched_arr_time))

flights_final <- subset(flights_final, select = c(6,9,19:23))
flights_final$total_delay <- flights_final$dep_delay + flights_final$arr_delay
names(flights_final)[names(flights_final) == 'total_delay'] <- 'Total Delay en Min'
head(flights_final)

```

```

## # A tibble: 6 x 8
##   dep_delay arr_delay time_hour      dep_time_FH
##   <dbl>      <dbl> <dtm>      <dtm>
## 1         2        11 2013-01-01 05:00:00 2013-01-01 05:17:00
## 2         4        20 2013-01-01 05:00:00 2013-01-01 05:33:00
## 3         2        33 2013-01-01 05:00:00 2013-01-01 05:42:00
## 4        -1       -18 2013-01-01 05:00:00 2013-01-01 05:44:00
## 5        -6       -25 2013-01-01 06:00:00 2013-01-01 05:54:00
## 6        -4        12 2013-01-01 05:00:00 2013-01-01 05:54:00
## # ... with 4 more variables: arr_time_FH <dtm>, sched_dep_time_FH <dtm>,

```

```
## # sched_arr_time_FH <dtm>, 'Total Delay en Min' <dbl>
```