# Lab5

Marianna Flores

September 24, 2020

# PARTE 1: prediccion eclipse solar

```
eclipse <- ymd_hms("2017/08/21 18:26:40")
SynodicMonth <- days(29)+hours(12)+minutes(44)+seconds(3)
Saros <- SynodicMonth*223
SiguienteEclipse <- eclipse+Saros
SiguienteEclipse
```

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

# PARTE 2: agrupaciones y operaciones con fechas

## 1

Estamos suponiendo que todos son llamadas pues si se pone el filtro para solo Call solo saca el Cod - Actualizacion de informacion

```
data <- read_excel("~/data.xlsx")
a <- convertToDateTime(data$`Fecha Creacion`, origin = "1900-01-01")
b <- dmy(data$`Fecha Creacion`)
a[is.na(a)] <- b[!is.na(b)]
data$`Fecha Creacion` <- a

c <- convertToDateTime(data$`Fecha Final`, origin = "1900-01-01")
d <- dmy(data$`Fecha Final`)
c[is.na(c)] <- d[!is.na(d)]
data$`Fecha Final` <- c
```

```
data$mes <- months(data$`Fecha Creacion`)
data$dia <- day(data$`Fecha Creacion`)
data$weekday <- weekdays(data$`Fecha Creacion`)
data$year <- year(data$`Fecha Creacion`)
data$duracion_en_secs <- (data$`Hora Final`-data$`Hora Creacion`)

#1
lpcodigo <- data %>% select(Cod, Call, mes) %>% group_by(Cod, mes) %>%
  filter(Call == 1) %>%  summarise(n = n(), .groups = 'drop')
lpcodigo2 <- data %>% select(Cod, Call, mes) %>% group_by(Cod, mes) %>%
  summarise(n = n(), .groups = 'drop')
lpcodigo2
```

```
## # A tibble: 84 x 3
##    Cod   mes          n
##    <chr> <chr>    <int>
##  1 0     April     1362
##  2 0     August    1442
##  3 0     December  1367
##  4 0     February  1236
##  5 0     January   1361
##  6 0     July      1463
##  7 0     June      1330
##  8 0     March     1419
##  9 0     May       1404
## 10 0     November  1337
## # ... with 74 more rows
```

0 - Julio (1463) Actualizacion de informacion - Mayo (1691) Cancelaciones - Marzo (4092) Cobros - Enero (688) Consultas - Octubre (10790) Empresarial - Octubre (3136) Otros/varios - Enero (1129)

# 2

El dia mas ocupado de la semana es Domingo

```
pDiaSemana <- data %>% select(weekday,Cod) %>% group_by(weekday) %>%
  summarise(n = n(), .groups = 'drop')
pDiaSemana2 <- data %>% select(weekday) %>% group_by(weekday) %>%
  summarise(n = n(), .groups = 'drop')
pDiaSemana2
```

```
## # A tibble: 7 x 2
##   weekday       n
##   <chr>     <int>
## 1 Friday    36804
## 2 Monday    37438
## 3 Saturday  37390
## 4 Sunday    39106
## 5 Thursday  37766
## 6 Tuesday   36945
## 7 Wednesday 38276
```

# 3

El mes mas ocupado es marzo

```
pmes <- data %>% select(mes,Cod) %>% group_by(mes) %>%
  summarise(n = n(), .groups = 'drop')
pmes2 <- data %>% select(mes) %>% group_by(mes) %>%
  summarise(n = n(), .groups = 'drop')
pmes2
```

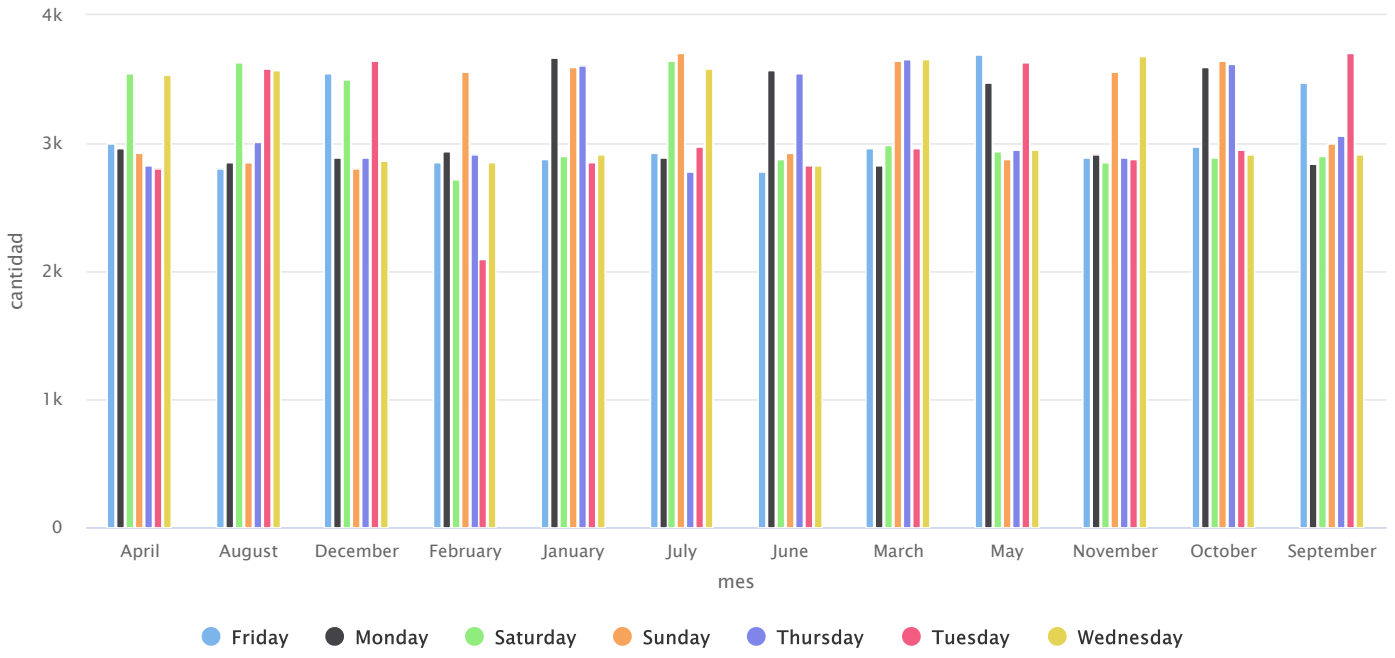
```
## # A tibble: 12 x 2
##    mes           n
##    <chr>     <int>
##  1 April     21611
##  2 August    22316
##  3 December  22151
##  4 February  19932
##  5 January   22425
##  6 July      22514
##  7 June      21370
##  8 March     22708
##  9 May       22525
## 10 November  21681
## 11 October   22601
## 12 September 21891
```

# 4

Los miercoles y domingos se hacen mas entregas en la mayoria de meses. Aun asi no se ve una temporalidad consistentes y solo se tienen datos de 1 year, podria depender de otros factores.

```
temp <- data %>% select(mes,weekday) %>% group_by(mes, weekday) %>%
  summarise(cantidad = n(), .groups = 'drop')

temp %>%
  hchart('column', hcaes(x = 'mes', y = 'cantidad', group = 'weekday')) %>%
  hc_title(text = "Viajes por ubicacion",
           align = "center",
           style = list(fontWeight = "bold", fontSize = "30px"))
```




## 5

```
lpromedio <- data %>% select(duracion_en_secs) %>% summarise(llamada_por_segundos = mean(duracion_en_secs), por_minutos = ll
amada_por_segundos/60)
lpromedio
```

```
## # A tibble: 1 x 2
##   llamada_por_segundos por_minutos
##   <drtn>               <drtn>
## 1 893.377 secs         14.88962 secs
```

## 6

```
data$duracion_en_mins <- (data$`Hora Final`-data$`Hora Creacion`)/60
df <- seq(from = 0, to = 30, by = 5)
fecuencia <- table(cut(x = as.numeric(data$duracion_en_mins), breaks = df))
frecuencia <- t(fecuencia)
df2 <- as.data.frame(frecuencia)
resultados <- df2 %>% select(Var2,Freq) %>% summarise(`Tiempo llamada minutos` = Var2, `cantidad llamadas` = Freq)
resultados
```

```
##   Tiempo llamada minutos cantidad llamadas
## 1                  (0,5]             42620
## 2                 (5,10]             42482
## 3                (10,15]             42371
## 4                (15,20]             42400
## 5                (20,25]             42143
## 6                (25,30]             42003
```

# PARTE 3: signo zodiacal

La fecha en la funcion se debe ingresar como "mes/dia/year"

```
zodiaco <- function(fecha){
fecha <- mdy(fecha)
mes <- month(fecha)
dia <- day(fecha)

if ((mes==2 && dia>=20) || (mes==3 && dia<= 20)){
  print("Piscis")
}else if ((mes==3 && dia>=21) || (mes==4 && dia<= 20)){
  print("Aries")
}else if ((mes==4 && dia>=21) || (mes==5 && dia<= 20)){
  print("Tauro")
}else if ((mes==5 && dia>=21) || (mes==6 && dia<= 20)){
  print("Geminis")
}else if ((mes==6 && dia>=21) || (mes==7 && dia<= 22)){
  print("Cancer")
}else if ((mes==7 && dia>=23) || (mes==8 && dia<= 22)){
  print("Leo")
}else if ((mes==8 && dia>=23) || (mes==9 && dia<= 22)){
  print("Virgo")
}else if ((mes==9 && dia>=23) || (mes==10 && dia<= 22)){
  print("Libra")
}else if ((mes==10 && dia>=23) || (mes==11 && dia<= 22)){
  print("Scorpio")
}else if ((mes==11 && dia>=23) || (mes==12 && dia<= 21)){
  print("Sagitario")
}else if ((mes==12 && dia>=22) || (mes==1 && dia<= 19)){
  print("Capricornio")
}else if ((mes==1 && dia>=20) || (mes==2 && dia<= 19)){
  print("Aquario")
}
}

##formato %m/%d/%Y
zodiaco("3/12/1999")
```

```
## [1] "Piscis"
```

# PARTE 4: Flights

```
flights$departure_time <- sub('(\\d{2})$', ':\\1', flights$dep_time)
flights$arrival_time <- sub('(\\d{2})$', ':\\1', flights$arr_time)
flights$scheduled_arrival <- sub('(\\d{2})$', ':\\1', flights$sched_arr_time)
flights$scheduled_departure <- sub('(\\d{2})$', ':\\1', flights$sched_dep_time)

flights$departure_time <- hm(flights$departure_time)
flights$arrival_time <- hm(flights$arrival_time)
flights$scheduled_arrival <- hm(flights$scheduled_arrival)
flights$scheduled_departure <- hm(flights$scheduled_departure)

flights$total_delay <- (flights$departure_time - flights$scheduled_departure)+(flights$arrival_time - flights$scheduled_arri
val)
delays <- flights %>% select(scheduled_departure, departure_time, scheduled_arrival, arrival_time, total_delay)

flights$departure <- ISOdatetime(flights$year,flights$month,flights$day,hour(flights$departure_time),minute(flights$departur
e_time),second((flights$departure_time)))
flights$arrival <- flights$departure+flights$arrival_time
flights$sched_departure <- ISOdatetime(flights$year,flights$month,flights$day,hour(flights$scheduled_departure),minute(fligh
ts$scheduled_departure),second((flights$scheduled_departure)))
flights$sched_arrival <- flights$sched_departure+flights$scheduled_arrival

delays <- flights %>% select(departure,sched_departure,arrival,sched_arrival,total_delay)
head(delays)
```

```
## # A tibble: 6 x 5
##   departure           sched_departure     arrival
##   <dttm>              <dttm>              <dttm>
## 1 2013-01-01 05:17:00 2013-01-01 05:15:00 2013-01-01 13:47:00
## 2 2013-01-01 05:33:00 2013-01-01 05:29:00 2013-01-01 14:23:00
## 3 2013-01-01 05:42:00 2013-01-01 05:40:00 2013-01-01 15:05:00
## 4 2013-01-01 05:44:00 2013-01-01 05:45:00 2013-01-01 15:48:00
## 5 2013-01-01 05:54:00 2013-01-01 06:00:00 2013-01-01 14:06:00
## 6 2013-01-01 05:54:00 2013-01-01 05:58:00 2013-01-01 13:34:00
## # ... with 2 more variables: sched_arrival <dttm>, total_delay <Period>
```