Universitat Oberta de Catalunya

Estudios de Informática, Multimedia y Telecomunicaciones

Práctica 2: Limpieza y análisis de datos

Cristina Merino García de la Reina, Isabel González Valle

8 de junio, 2020

Descripción

El objetivo de esta actividad será el tratamiento de un dataset, que puede ser el creado en la práctica 1 o bien cualquier dataset libre disponible en Kaggle (https://www.kaggle.com (https://www.kaggle.com)).

Siguiendo las principales etapas de un proyecto analítico, las diferentes tareas a realizar (y justificar) son las siguientes:

1. Descripción del dataset

¿Por qué es importante y qué pregunta/problema pretende responder?

Para esta práctica hemos buscado un dataset de los vuelos que fueron cancelados o que sufrieron retrasos en durante el año 2015 comunicados a través del Departamento de Estadísticas de Transportes de los Estados Unidos. Este conjunto de datos contiene la información correspondiente

a los vuelos operados por las grandes compañías aéreas.
Poder conocer y tener una referencia del motivo de los vuelos cancelados o retrasados es interesante para las personas que deben usar este medio de transporte. Se pueden analizar diferentes problemáticas, como la relación entre los vuelos retrasados y los días de la semana, así como identificar cuál puede ser el mejor mes para viajar, que aeropuerto debemos evitar y por último, buscaremos identificar cuál es la mejor compañía para viajar. Intentaremos dar respuesta a algunas de estas preguntas durante esta práctica.
https://www.kaggle.com/usdot/flight-delays (https://www.kaggle.com/usdot/flight-delays)
El tipo de licencia de este dataset es: *CC0 1.0 Universal (CC0 1.0) Public Domain Dedication* . Por lo tanto es público y puede ser utilizado libremente para el trabajo que vamos a realizar.
El conjunto de datos elegido contiene 31 variables y casi 6 millones de observaciones, por lo que de cara a la práctica reduciremos la cantidad de datos, intentando que la muestra a utilizar represente el conjunto de datos original lo más fielmente posible, para ello utilizaremos las técnicas de muestreo que se estudiaron en el módulo anterior.
Del mismo modo, eliminaremos aquellas variables que no aporten valor al estudio que vamos a realizar, reduciendo así la dimensionalidad del conjunto de datos.
Las variables que componen este dataset, son las siguientes:

- YEAR: Año del vuelo (2015)
- MONTH: Mes del vuelo
- DAY: Día del vuelo
- DAY_OF_WEEK: Día de la semana, donde el día 1=lunes y el 7=Domingo
- AIRLINE: Código de la aerolínea
- FLIGHT_NUMBER: Número de vuelo
- TAIL_NUMBER: Número de identificación de la aeronave
- ORIGIN_AIRPORT: Aeropuerto Origen
- DESTINATION_AIRPORT: Aeropuerto Destino
- SCHEDULED_DEPARTURE: Hora programada de salida en formato hhmm (55 -> 00:55)
- DEPARTURE_TIME: Hora de salida del vuelo en formato hhmm
- DEPARTURE_DELAY: Diferencia en minutos entre la salida programada y la real (valores negativos identifican salidas del vuelo con antelaión)
- TAXI_OUT: Tiempo de rodaje del avión desde que deja la puerta de embarque hasta despegue.
- WHEELS_OFF: Hora en la que el avión despega, momento en el que las ruedas del avión dejan de tocar el suelo
- SCHEDULED_TIME: Tiempo programado de vuelo.
- ELAPSED_TIME: Tiempo total de vuelo contado desde el momento que el avión se pone en marcha hasta que para completamente en destino, es decir contando el rodaje en el aeropuerto.
- AIR_TIME: Tiempo desde despegue hasta aterrizaje
- DISTANCE: Distancia en millas
- WHEELS_ON: Hora en la que el avión toca tierra.
- TAXI_IN: Tiempo de rodaje en el aeropuerto destino hasta que el avión para completamente.
- SCHEDULED_ARRIVAL: Hora programada de llegada en formato hhmm
- ARRIVAL_TIME: Hora de llegada real en formato hhmm
- ARRIVAL_DELAY: Diferencia en minutos entre la salida programada y la real
- DIVERTED: Vuelo desviado (0-No, 1-Sí)
- CANCELLED: Vuelo Cancelado (0-No, 1-Sí)
- CANCELLATION_REASON: Motivo de cancelación (A-Carrier, B-Weather, C-National Air System, D-Security)
- AIR_SYSTEM_DELAY: Tiempo de retraso por el motivo indicado
- SECURITY_DELAY: Tiempo de retraso por el motivo indicado

- AIRLINE_DELAY: Tiempo de retraso por el motivo indicado
- LATE_AIRCRAFT_DELAY: Tiempo de retraso por el motivo indicado
- WEATHER_DELAY: Tiempo de retraso por el motivo indicado

```
#Cargamos el dataset
vuelos <- read.csv("flights.csv", sep=c(","), header = TRUE)

head(vuelos)
```

```
##   YEAR MONTH DAY DAY_OF_WEEK AIRLINE FLIGHT_NUMBER TAIL_NUMBER ORIGIN_AIRPOR
T
## 1 2015     1   1           4      AS            98      N407AS            AN
C
## 2 2015     1   1           4      AA          2336      N3KUAA            LA
X
## 3 2015     1   1           4      US           840      N171US            SF
O
## 4 2015     1   1           4      AA           258      N3HYAA            LA
X
## 5 2015     1   1           4      AS           135      N527AS            SE
A
## 6 2015     1   1           4      DL           806      N3730B            SF
O
##   DESTINATION_AIRPORT SCHEDULED_DEPARTURE DEPARTURE_TIME DEPARTURE_DELAY
## 1                 SEA                   5           2354             -11
## 2                 PBI                  10              2              -8
## 3                 CLT                  20             18              -2
## 4                 MIA                  20             15              -5
## 5                 ANC                  25             24              -1
## 6                 MSP                  25             20              -5
##   TAXI_OUT WHEELS_OFF SCHEDULED_TIME ELAPSED_TIME AIR_TIME DISTANCE WHEELS_O
N
## 1       21         15            205          194      169     1448       40
4
## 2       12         14            280          279      263     2330       73
7
## 3       16         34            286          293      266     2296       80
0
## 4       15         30            285          281      258     2342       74
8
## 5       11         35            235          215      199     1448       25
4
## 6       18         38            217          230      206     1589       60
4
##   TAXI_IN SCHEDULED_ARRIVAL ARRIVAL_TIME ARRIVAL_DELAY DIVERTED CANCELLED
## 1       4              430          408           -22        0         0
## 2       4              750          741            -9        0         0
## 3      11              806          811             5        0         0
## 4       8              805          756            -9        0         0
## 5       5              320          259           -21        0         0
## 6       6              602          610             8        0         0
##   CANCELLATION_REASON AIR_SYSTEM_DELAY SECURITY_DELAY AIRLINE_DELAY
## 1                                  NA             NA            NA
## 2                                  NA             NA            NA
## 3                                  NA             NA            NA
## 4                                  NA             NA            NA
## 5                                  NA             NA            NA
## 6                                  NA             NA            NA
##   LATE_AIRCRAFT_DELAY WEATHER_DELAY
## 1                  NA            NA
## 2                  NA            NA
```

```
## 3                 NA              NA
## 4                 NA              NA
## 5                 NA              NA
## 6                 NA              NA
```

Este conjunto de datos tiene un tamaño demasiado grande para algunas de las operaciones que necesitamos realizar y por este motivo hemos decidido realizar la práctica con un subconjunto del mismo. En el caso necesario, todos los cálculos se podrían repetir con el conjunto completo.

```
#Reducción de la cantidad
set.seed(222)
index <- sample(1:nrow(vuelos), size=0.07*nrow(vuelos))
vuelos_reduc <- vuelos[index,]
str(vuelos_reduc)
```

```
## 'data.frame':    407335 obs. of  31 variables:
##  $ YEAR               : int  2015 2015 2015 2015 2015 2015 2015 2015 2015 20
15 ...
##  $ MONTH              : int  11 4 4 7 2 7 12 2 3 2 ...
##  $ DAY                : int  16 7 1 28 9 23 24 22 8 8 ...
##  $ DAY_OF_WEEK        : int  1 2 3 2 1 4 4 7 7 7 ...
##  $ AIRLINE            : Factor w/ 14 levels "AA","AS","B6",..: 5 14 4 5 14 1
14 10 8 4 ...
##  $ FLIGHT_NUMBER      : int  5084 1023 2182 4330 1963 2148 1915 4636 2950 21
04 ...
##  $ TAIL_NUMBER        : Factor w/ 4898 levels "","7819A","7820L",..: 3370 35
05 4730 149 3706 4748 1306 1795 4640 4580 ...
##  $ ORIGIN_AIRPORT     : Factor w/ 628 levels "10135","10136",..: 327 358 439
504 483 346 344 593 535 523 ...
##  $ DESTINATION_AIRPORT: Factor w/ 629 levels "10135","10136",..: 614 577 328
459 500 490 368 432 573 393 ...
##  $ SCHEDULED_DEPARTURE: int  825 930 540 1545 1055 600 1125 1956 1145 1935
...
##  $ DEPARTURE_TIME     : int  819 943 538 1559 1105 553 1124 2002 1242 1932
...
##  $ DEPARTURE_DELAY    : int  -6 13 -2 14 10 -7 -1 6 57 -3 ...
##  $ TAXI_OUT           : int  14 10 12 21 7 15 18 18 15 37 ...
##  $ WHEELS_OFF         : int  833 953 550 1620 1112 608 1142 2020 1257 2009
...
##  $ SCHEDULED_TIME     : int  128 240 57 108 190 74 80 63 68 137 ...
##  $ ELAPSED_TIME       : int  129 214 47 105 166 68 86 57 68 140 ...
##  $ AIR_TIME           : int  111 201 28 75 153 47 64 35 51 95 ...
##  $ DISTANCE           : int  674 1407 153 468 1363 184 439 216 268 680 ...
##  $ WHEELS_ON          : int  924 1214 618 1735 1545 655 1346 2055 1348 2044
...
##  $ TAXI_IN            : int  4 3 7 9 6 6 4 4 2 8 ...
##  $ SCHEDULED_ARRIVAL  : int  933 1230 637 1733 1605 714 1345 2059 1253 2052
...
##  $ ARRIVAL_TIME       : int  928 1217 625 1744 1551 701 1350 2059 1350 2052
...
##  $ ARRIVAL_DELAY      : int  -5 -13 -12 11 -14 -13 5 0 57 0 ...
##  $ DIVERTED           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CANCELLED          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CANCELLATION_REASON: Factor w/ 5 levels "","A","B","C",..: 1 1 1 1 1 1 1
1 1 1 ...
##  $ AIR_SYSTEM_DELAY   : int  NA NA NA NA NA NA NA NA 13 NA ...
##  $ SECURITY_DELAY     : int  NA NA NA NA NA NA NA NA 0 NA ...
##  $ AIRLINE_DELAY      : int  NA NA NA NA NA NA NA NA 0 NA ...
##  $ LATE_AIRCRAFT_DELAY: int  NA NA NA NA NA NA NA NA 44 NA ...
##  $ WEATHER_DELAY      : int  NA NA NA NA NA NA NA NA 0 NA ...
```

Hemos visto los primeros valores que toman las variables del conjunto con los datos reducidos. Ahora veremos un pequeño resumen estadístico de dichas variables.

```
summary(vuelos_reduc)
```

```
##       YEAR          MONTH             DAY         DAY_OF_WEEK
##  Min.   :2015   Min.   : 1.000   Min.   : 1.00   Min.   :1.000
##  1st Qu.:2015   1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.:2.000
##  Median :2015   Median : 7.000   Median :16.00   Median :4.000
##  Mean   :2015   Mean   : 6.527   Mean   :15.71   Mean   :3.927
##  3rd Qu.:2015   3rd Qu.: 9.000   3rd Qu.:23.00   3rd Qu.:6.000
##  Max.   :2015   Max.   :12.000   Max.   :31.00   Max.   :7.000
##
##     AIRLINE      FLIGHT_NUMBER   TAIL_NUMBER     ORIGIN_AIRPORT
##  WN     :88427   Min.   :   1          :   998   ATL    : 24384
##  DL     :61409   1st Qu.: 733   N480HA :   269   ORD    : 19887
##  AA     :50761   Median :1690   N486HA :   269   DFW    : 16725
##  OO     :41209   Mean   :2172   N488HA :   267   DEN    : 13607
##  EV     :39945   3rd Qu.:3229   N475HA :   263   LAX    : 13476
##  UA     :36077   Max.   :7438   N478HA :   253   IAH    : 10383
##  (Other):89507                  (Other):405016   (Other):308873
##  DESTINATION_AIRPORT SCHEDULED_DEPARTURE DEPARTURE_TIME DEPARTURE_DELAY
##  ATL    : 24154      Min.   :   3        Min.   :   1   Min.   : -68.000
##  ORD    : 19577      1st Qu.: 916        1st Qu.: 920   1st Qu.:  -5.000
##  DFW    : 16994      Median :1325        Median :1329   Median :  -2.000
##  LAX    : 13750      Mean   :1329        Mean   :1334   Mean   :   9.399

##  DEN    : 13545      3rd Qu.:1730        3rd Qu.:1739   3rd Qu.:   7.000
##  PHX    : 10426      Max.   :2359        Max.   :2400   Max.   :1380.000
##  (Other):308889                          NA's   :6027   NA's   :6027
##     TAXI_OUT        WHEELS_OFF   SCHEDULED_TIME   ELAPSED_TIME     AIR_TIME
##  Min.   :  1.00   Min.   :   1   Min.   : 20.0   Min.   : 16    Min.   :  8.
0
##  1st Qu.: 11.00   1st Qu.: 935   1st Qu.: 85.0   1st Qu.: 82    1st Qu.: 60.
0
##  Median : 14.00   Median :1342   Median :123.0   Median :118    Median : 94.
0
##  Mean   : 16.09   Mean   :1356   Mean   :141.6   Mean   :137    Mean   :113.
5
##  3rd Qu.: 19.00   3rd Qu.:1753   3rd Qu.:173.0   3rd Qu.:168    3rd Qu.:144.
0
##  Max.   :167.00   Max.   :2400   Max.   :705.0   Max.   :733    Max.   :687.
0
##  NA's   :6234     NA's   :6234   NA's   :2       NA's   :7382   NA's   :7382
##     DISTANCE        WHEELS_ON       TAXI_IN        SCHEDULED_ARRIVAL
##  Min.   :  31.0   Min.   :   1   Min.   :  1.000   Min.   :   1
##  1st Qu.: 373.0   1st Qu.:1054   1st Qu.:  4.000   1st Qu.:1110
##  Median : 647.0   Median :1507   Median :  6.000   Median :1519
##  Mean   : 821.7   Mean   :1470   Mean   :  7.424   Mean   :1493
##  3rd Qu.:1062.0   3rd Qu.:1911   3rd Qu.:  9.000   3rd Qu.:1917
##  Max.   :4983.0   Max.   :2400   Max.   :184.000   Max.   :2359
##                   NA's   :6491   NA's   :6491
##   ARRIVAL_TIME  ARRIVAL_DELAY         DIVERTED          CANCELLED
##  Min.   :   1   Min.   : -87.000   Min.   :0.000000   Min.   :0.00000
##  1st Qu.:1058   1st Qu.: -13.000   1st Qu.:0.000000   1st Qu.:0.00000
##  Median :1511   Median :  -5.000   Median :0.000000   Median :0.00000
##  Mean   :1475   Mean   :   4.462   Mean   :0.002664   Mean   :0.01546
##  3rd Qu.:1916   3rd Qu.:   8.000   3rd Qu.:0.000000   3rd Qu.:0.00000
```

```
##  Max.   :2400   Max.   :1384.000   Max.   :1.000000   Max.   :1.00000
##  NA's   :6491   NA's   :7382
##  CANCELLATION_REASON AIR_SYSTEM_DELAY SECURITY_DELAY   AIRLINE_DELAY
##   :401038            Min.   :  0.0    Min.   :  0.0    Min.   :   0.0
##  A:  1817            1st Qu.:  0.0    1st Qu.:  0.0    1st Qu.:   0.0
##  B:  3444            Median :  2.0    Median :  0.0    Median :   2.0
##  C:  1034            Mean   : 13.7    Mean   :  0.1    Mean   :  18.9
##  D:     2            3rd Qu.: 18.0    3rd Qu.:  0.0    3rd Qu.:  19.0
##                      Max.   :916.0    Max.   :221.0    Max.   :1380.0
##                      NA's   :332838   NA's   :332838   NA's   :332838
##  LATE_AIRCRAFT_DELAY WEATHER_DELAY
##  Min.   :   0.0      Min.   :  0.0
##  1st Qu.:   0.0      1st Qu.:  0.0
##  Median :   3.0      Median :  0.0
##  Mean   :  23.6      Mean   :  2.9
##  3rd Qu.:  29.0      3rd Qu.:  0.0
##  Max.   :1294.0      Max.   :896.0
##  NA's   :332838      NA's   :332838
```

Nuestro conjunto de datos ahora tiene una dimensión de 407335 valores, como podemos apreciar en el siguiente output de R:

```
length(vuelos_reduc$YEAR)
```

```
## [1] 407335
```

2. Integración y selección de los datos de interés a analizar

Vamos a cargar los datos de localización de los aeropuertos y haremos un merge de los datos de los aeropuertos con el dataset que tenemos para, entre otras cosas, hacer una visualización en un mapa. Renombramos las columnas para dejar la misma nomenclatura en aquellas que queremos unir

```
airports <- read.csv("datasets_810_1496_airports.csv", header=TRUE)
head(airports)
```

```
##   IATA_CODE                              AIRPORT        CITY STATE COUNTRY
## 1       ABE Lehigh Valley International Airport   Allentown    PA     USA
## 2       ABI            Abilene Regional Airport     Abilene    TX     USA
## 3       ABQ   Albuquerque International Sunport Albuquerque    NM     USA
## 4       ABR           Aberdeen Regional Airport    Aberdeen    SD     USA
## 5       ABY  Southwest Georgia Regional Airport      Albany    GA     USA
## 6       ACK          Nantucket Memorial Airport   Nantucket    MA     USA
##   LATITUDE  LONGITUDE
## 1 40.65236  -75.44040
## 2 32.41132  -99.68190
## 3 35.04022 -106.60919
## 4 45.44906  -98.42183
## 5 31.53552  -84.19447
## 6 41.25305  -70.06018
```

```
colnames(vuelos_reduc)[8]  <- "ORIGIN_CODE"
colnames(airports) <- c("ORIGIN_CODE","ORIGIN_AIRPORT", "ORIGIN_CITY", "ORIGIN_
STATE", "ORIGIN_COUNTRY", "ORIGIN_LATITUDE", "ORIGIN_LONGITUDE" )

flight_airports <- left_join(vuelos_reduc,airports,by="ORIGIN_CODE")
```

```
## Warning: Column `ORIGIN_CODE` joining factors with different levels, coercin
g to
## character vector
```

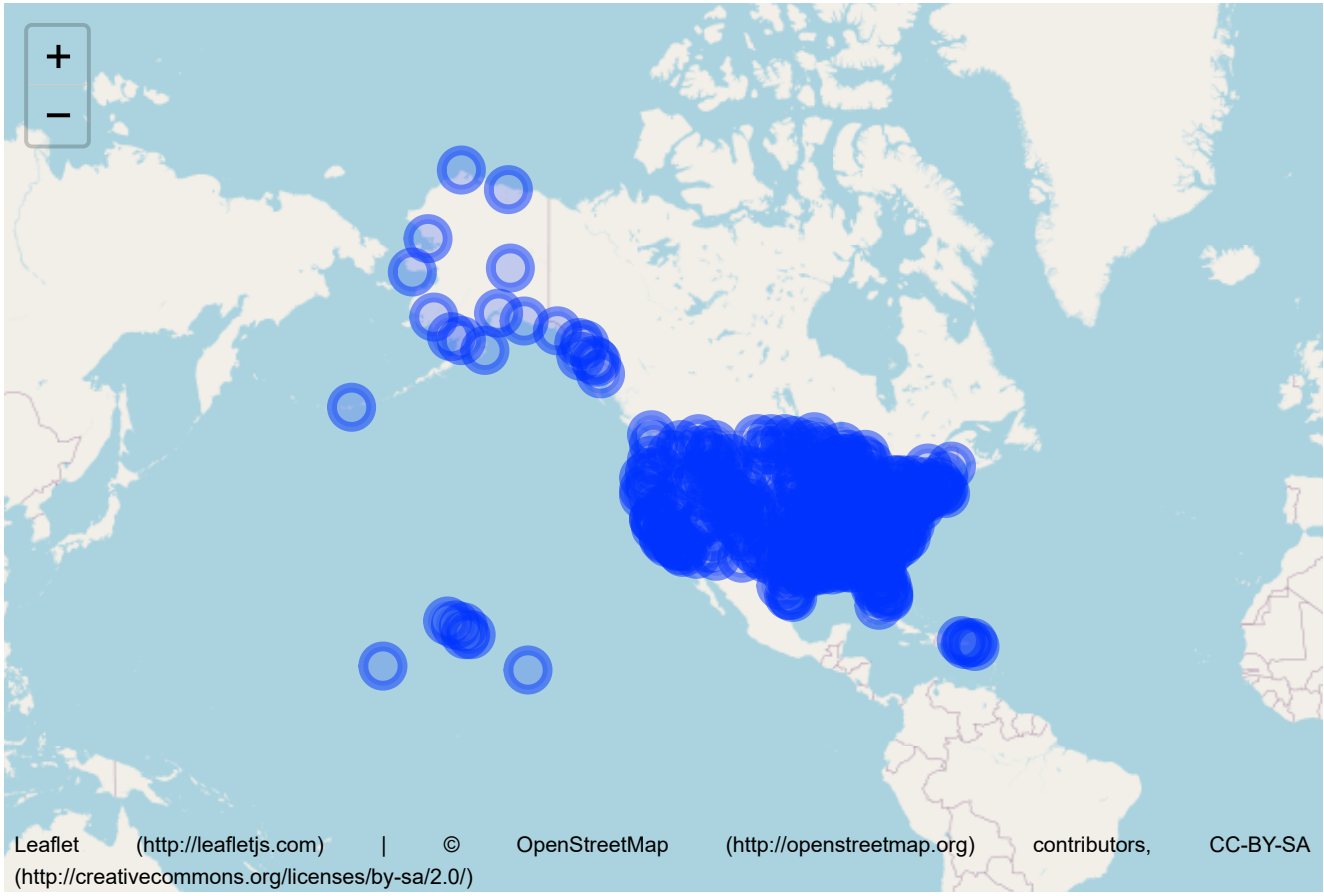
Nos da error, porque como nuestro dataset es un subconjunto de los datos iniciales, puede que no estén todos los aeropuertos, por lo que tenemos que igualar los niveles de los dos campos tipo factor.

```
combined <- sort(union(levels(vuelos_reduc$ORIGIN_CODE), levels(airports$ORIGIN
_CODE)))
flight_airports <- left_join(mutate(vuelos_reduc, ORIGIN_CODE=factor(ORIGIN_COD
E, levels=combined)),
                             mutate(airports, ORIGIN_CODE=factor(ORIGIN_CODE, l
evels=combined)))
```

```
## Joining, by = "ORIGIN_CODE"
```

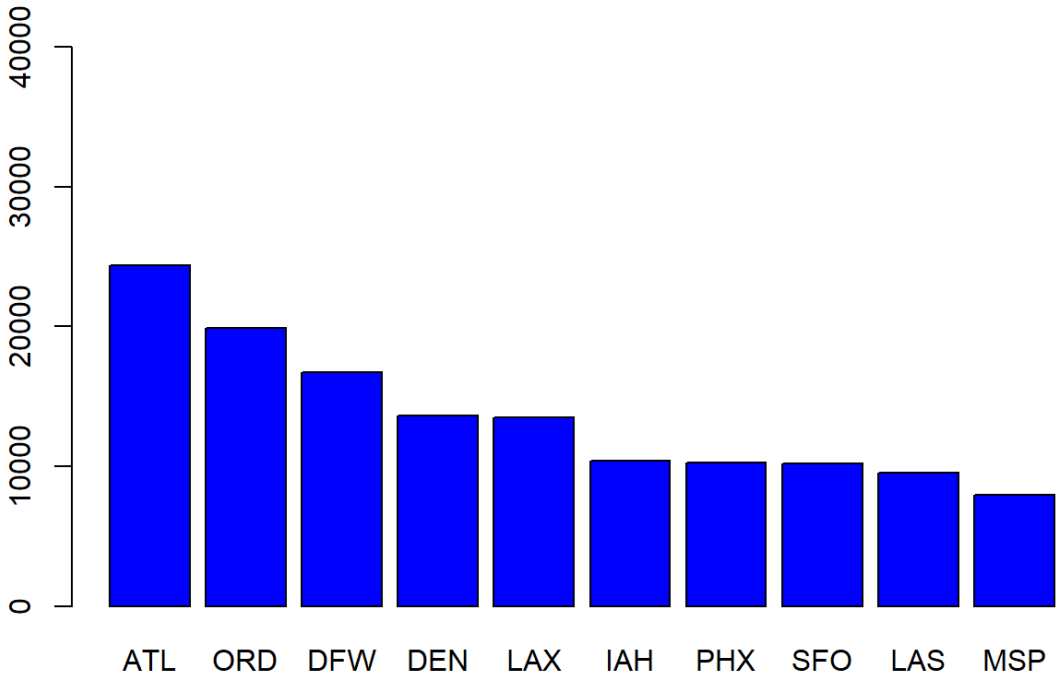
```
#visualización del volumen de vuelos de cada aeropuerto
longitude <- unique(flight_airports$ORIGIN_LONGITUDE)
latitude <- unique(flight_airports$ORIGIN_LATITUDE)
df = data.frame(Lat = latitude, Long = longitude)
leaflet(df) %>% addTiles() %>% addCircleMarkers() #map visualization
```

```
## Warning in validateCoords(lng, lat, funcName): Data contains 1 rows with eit
her
## missing or invalid lat/lon values and will be ignored
```



En el mapa podemos visualizar donde están localizados los aeropuertos analizados en este dataset. A continuación, lo que haremos será mostrar un gráfico de barras donde veremos el volumen de vuelos por cada aeropuerto. Debemos tener en cuenta que estamos trabajando con un conjunto reducido de los datos, por lo que este gráfico no nos aporta la información real del número de vuelos pero sí nos sirve para ver qué aeropuertos tienen mayor volumen de tráfico aereo.

```
popular_airports <- sort(table(flight_airports$ORIGIN_CODE), decreasing = TRUE
 )
barplot(popular_airports[1:10], col = "blue", ylim = c(0,40000))
```



También vemos que el jueves es el día de la semana con mayor número de vuelos. Al igual que antes, estamos trabajando con el conjunto de datos reducido, por lo que la información mostrada nos ayuda a ver el día de la semana con mayor número de vuelos pero el número que obtenemos no representa al total del año 2015, si no a nuestra muestra reducida.

```
sort(table(flight_airports$DAY_OF_WEEK), decreasing = TRUE )
```

```
##
##     4     5     1     3     2     7     6
## 61369 60515 60391 60050 59025 56984 49001
```

A partir de aquí, vamos a identificar los tipos de variables y a quedarnos con los datos que nos interesarán para realizar nuestro estudio.

```
res <- sapply(vuelos_reduc,class)
kable(data.frame(variables=names(res),clase=as.vector(res)),
      caption = "asignación de clase de objeto R a cada variable")
```

asignación de clase de objeto R a cada variable

variables	clase
YEAR	integer
MONTH	integer
DAY	integer
DAY_OF_WEEK	integer
AIRLINE	factor
FLIGHT_NUMBER	integer

variables	clase
TAIL_NUMBER	factor
ORIGIN_CODE	factor
DESTINATION_AIRPORT	factor
SCHEDULED_DEPARTURE	integer
DEPARTURE_TIME	integer
DEPARTURE_DELAY	integer
TAXI_OUT	integer
WHEELS_OFF	integer
SCHEDULED_TIME	integer
ELAPSED_TIME	integer
AIR_TIME	integer
DISTANCE	integer
WHEELS_ON	integer
TAXI_IN	integer
SCHEDULED_ARRIVAL	integer
ARRIVAL_TIME	integer
ARRIVAL_DELAY	integer
DIVERTED	integer
CANCELLED	integer
CANCELLATION_REASON	factor
AIR_SYSTEM_DELAY	integer
SECURITY_DELAY	integer
AIRLINE_DELAY	integer
LATE_AIRCRAFT_DELAY	integer
WEATHER_DELAY	integer

Vemos la asignación de clase que R le da a cada variable. Sin embargo, tenemos que reasignarle la clase a algunas variables como vemos a continuación.

```
vuelos[1:4] <- lapply(vuelos[1:4], as.numeric)
vuelos[6] <- lapply(vuelos[6], as.numeric)
vuelos[10:25] <- lapply(vuelos[10:25], as.numeric)
vuelos[27:31] <- lapply(vuelos[27:31], as.numeric)
res <- sapply(vuelos,class)
tabla_datos <- data.frame(variables=names(res),clase=as.vector(res))
kable(tabla_datos,
      caption = "asignación de clase de objeto R a cada variable")
```

asignación de clase de objeto R a cada variable

variables	clase
YEAR	numeric
MONTH	numeric
DAY	numeric
DAY_OF_WEEK	numeric
AIRLINE	factor
FLIGHT_NUMBER	numeric
TAIL_NUMBER	factor
ORIGIN_AIRPORT	factor
DESTINATION_AIRPORT	factor

variables	clase
SCHEDULED_DEPARTURE	numeric
DEPARTURE_TIME	numeric
DEPARTURE_DELAY	numeric
TAXI_OUT	numeric
WHEELS_OFF	numeric
SCHEDULED_TIME	numeric
ELAPSED_TIME	numeric
AIR_TIME	numeric
DISTANCE	numeric
WHEELS_ON	numeric
TAXI_IN	numeric
SCHEDULED_ARRIVAL	numeric
ARRIVAL_TIME	numeric
ARRIVAL_DELAY	numeric
DIVERTED	numeric
CANCELLED	numeric
CANCELLATION_REASON	factor
AIR_SYSTEM_DELAY	numeric
SECURITY_DELAY	numeric
AIRLINE_DELAY	numeric
LATE_AIRCRAFT_DELAY	numeric
WEATHER_DELAY	numeric

Volvemos a ver un resumen con los primeros datos de cada variable:

```
str(vuelos)
```

```
## 'data.frame':    5819079 obs. of  31 variables:
##  $ YEAR               : num  2015 2015 2015 2015 2015 ...
##  $ MONTH              : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ DAY                : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ DAY_OF_WEEK        : num  4 4 4 4 4 4 4 4 4 4 ...
##  $ AIRLINE            : Factor w/ 14 levels "AA","AS","B6",..: 2 1 12 1 2 4
9 12 1 4 ...
##  $ FLIGHT_NUMBER      : num  98 2336 840 258 135 ...
##  $ TAIL_NUMBER        : Factor w/ 4898 levels "","7819A","7820L",..: 1624 15
58 423 1518 2133 1143 2767 2412 1563 3936 ...
##  $ ORIGIN_AIRPORT     : Factor w/ 628 levels "10135","10136",..: 324 483 585
483 584 585 481 483 585 481 ...
##  $ DESTINATION_AIRPORT: Factor w/ 629 levels "10135","10136",..: 585 543 374
511 325 524 524 374 394 328 ...
##  $ SCHEDULED_DEPARTURE: num  5 10 20 20 25 25 25 30 30 30 ...
##  $ DEPARTURE_TIME     : num  2354 2 18 15 24 ...
##  $ DEPARTURE_DELAY    : num  -11 -8 -2 -5 -1 -5 -6 14 -11 3 ...
##  $ TAXI_OUT           : num  21 12 16 15 11 18 11 13 17 12 ...
##  $ WHEELS_OFF         : num  15 14 34 30 35 38 30 57 36 45 ...
##  $ SCHEDULED_TIME     : num  205 280 286 285 235 217 181 273 195 221 ...
##  $ ELAPSED_TIME       : num  194 279 293 281 215 230 170 249 193 203 ...
##  $ AIR_TIME           : num  169 263 266 258 199 206 154 228 173 186 ...
##  $ DISTANCE           : num  1448 2330 2296 2342 1448 ...
##  $ WHEELS_ON          : num  404 737 800 748 254 604 504 745 529 651 ...
##  $ TAXI_IN            : num  4 4 11 8 5 6 5 8 3 5 ...
##  $ SCHEDULED_ARRIVAL  : num  430 750 806 805 320 602 526 803 545 711 ...
##  $ ARRIVAL_TIME       : num  408 741 811 756 259 610 509 753 532 656 ...
##  $ ARRIVAL_DELAY      : num  -22 -9 5 -9 -21 8 -17 -10 -13 -15 ...
##  $ DIVERTED           : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ CANCELLED          : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ CANCELLATION_REASON: Factor w/ 5 levels "","A","B","C",..: 1 1 1 1 1 1 1
1 1 1 ...
##  $ AIR_SYSTEM_DELAY   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ SECURITY_DELAY     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ AIRLINE_DELAY      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ LATE_AIRCRAFT_DELAY: num  NA NA NA NA NA NA NA NA NA NA ...
##  $ WEATHER_DELAY      : num  NA NA NA NA NA NA NA NA NA NA ...
```

De todas las variables cargadas, de momento nos vamos a quedar con las siguientes:
MONTH
DAY
DAY_OF_WEEK
AIRLINE
ORIGIN_CODE
DESTINATION_AIRPORT
SCHEDULED_DEPARTURE
DEPARTURE_TIME
DEPARTURE_DELAY
SCHEDULED_TIME
ELAPSED_TIME
AIR_TIME
DISTANCE

SCHEDULED_ARRIVAL
ARRIVAL_TIME
ARRIVAL_DELAY

```
vuelos_reduc <- dplyr::select(vuelos_reduc, -"YEAR",-"TAIL_NUMBER",-"AIR_SYSTEM
_DELAY",-"SECURITY_DELAY",
                              -"AIRLINE_DELAY",-"LATE_AIRCRAFT_DELAY",-"WEATHER
_DELAY",
                               -"TAXI_OUT",-"TAXI_IN",-"WHEELS_OFF", -"WHEELS_O
N",-"DIVERTED",
                              -"CANCELLED",-"CANCELLATION_REASON")

str(vuelos_reduc)
```

```
## 'data.frame':    407335 obs. of  17 variables:
##  $ MONTH              : int  11 4 4 7 2 7 12 2 3 2 ...
##  $ DAY                : int  16 7 1 28 9 23 24 22 8 8 ...
##  $ DAY_OF_WEEK        : int  1 2 3 2 1 4 4 7 7 7 ...
##  $ AIRLINE            : Factor w/ 14 levels "AA","AS","B6",..: 5 14 4 5 14 1
14 10 8 4 ...
##  $ FLIGHT_NUMBER      : int  5084 1023 2182 4330 1963 2148 1915 4636 2950 21
04 ...
##  $ ORIGIN_CODE        : Factor w/ 628 levels "10135","10136",..: 327 358 439
504 483 346 344 593 535 523 ...
##  $ DESTINATION_AIRPORT: Factor w/ 629 levels "10135","10136",..: 614 577 328
459 500 490 368 432 573 393 ...
##  $ SCHEDULED_DEPARTURE: int  825 930 540 1545 1055 600 1125 1956 1145 1935
...
##  $ DEPARTURE_TIME     : int  819 943 538 1559 1105 553 1124 2002 1242 1932
...
##  $ DEPARTURE_DELAY    : int  -6 13 -2 14 10 -7 -1 6 57 -3 ...
##  $ SCHEDULED_TIME     : int  128 240 57 108 190 74 80 63 68 137 ...
##  $ ELAPSED_TIME       : int  129 214 47 105 166 68 86 57 68 140 ...
##  $ AIR_TIME           : int  111 201 28 75 153 47 64 35 51 95 ...
##  $ DISTANCE           : int  674 1407 153 468 1363 184 439 216 268 680 ...
##  $ SCHEDULED_ARRIVAL  : int  933 1230 637 1733 1605 714 1345 2059 1253 2052
...
##  $ ARRIVAL_TIME       : int  928 1217 625 1744 1551 701 1350 2059 1350 2052
...
##  $ ARRIVAL_DELAY      : int  -5 -13 -12 11 -14 -13 5 0 57 0 ...
```

3. Limpieza de los datos

3.1 Elementos vacíos

¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?

En este apartado, vamos a comprobar los valores que contienen nuestras variables para detectar si hay errores en los mismo, si tenemos elementos vacios o ceros o si hay datos fuera de los valores esperados, por ejemplo en las horas, días o meses.

```
#Comprobamos valore nulos o valores perdidos
sapply(vuelos_reduc, function(x) sum(is.na(x)))
```

```
##               MONTH                 DAY         DAY_OF_WEEK             AIRL
INE
##                   0                   0                   0
0
##       FLIGHT_NUMBER         ORIGIN_CODE DESTINATION_AIRPORT SCHEDULED_DEPART
URE
##                   0                   0                   0
0
##      DEPARTURE_TIME     DEPARTURE_DELAY      SCHEDULED_TIME        ELAPSED_T
IME
##                6027                6027                   2                7
382
##            AIR_TIME            DISTANCE   SCHEDULED_ARRIVAL        ARRIVAL_T
IME
##                7382                   0                   0                6
491
##       ARRIVAL_DELAY
##                7382
```

```
#Otra forma de sacar los valores nulos
colSums(is.na(vuelos_reduc))
```

```
##               MONTH                 DAY         DAY_OF_WEEK             AIRL
INE
##                   0                   0                   0
0
##       FLIGHT_NUMBER         ORIGIN_CODE DESTINATION_AIRPORT SCHEDULED_DEPART
URE
##                   0                   0                   0
0
##      DEPARTURE_TIME     DEPARTURE_DELAY      SCHEDULED_TIME        ELAPSED_T
IME
##                6027                6027                   2                7
382
##            AIR_TIME            DISTANCE   SCHEDULED_ARRIVAL        ARRIVAL_T
IME
##                7382                   0                   0                6
491
##       ARRIVAL_DELAY
##                7382
```

De los valores nulos que hemos identificado, hacemos una revisión para conocer el motivo de esos valores. Comprobamos que en el caso de los valores nulos en las variables DEPARTURE_DELAY y ARRIVAL_DELAY, se trata de aquellos vuelos que han sido cancelados o desviados, por lo que, como

nuestro estudio va a estar basado en los vuelos completados, eliminaremos todos estos valores nulos.

Comprobamos los valores nulos de la columna DEPARTURE_DELAY y ARRIVAL_DELAY:

```
head(vuelos %>% filter(is.na(vuelos$DEPARTURE_DELAY)))
```

```
##   YEAR MONTH DAY DAY_OF_WEEK AIRLINE FLIGHT_NUMBER TAIL_NUMBER ORIGIN_AIRPOR
T
## 1 2015     1   1           4      AS           136      N431AS            AN
C
## 2 2015     1   1           4      AA          2459      N3BDAA            PH
X
## 3 2015     1   1           4      OO          5254      N746SK            MA
F
## 4 2015     1   1           4      MQ          2859      N660MQ            SG
F
## 5 2015     1   1           4      OO          5460      N583SW            RD
D
## 6 2015     1   1           4      MQ          2926      N932MQ            CH
S
##   DESTINATION_AIRPORT SCHEDULED_DEPARTURE DEPARTURE_TIME DEPARTURE_DELAY
## 1                 SEA                 135             NA              NA
## 2                 DFW                 200             NA              NA
## 3                 IAH                 510             NA              NA
## 4                 DFW                 525             NA              NA
## 5                 SFO                 530             NA              NA
## 6                 DFW                 545             NA              NA
##   TAXI_OUT WHEELS_OFF SCHEDULED_TIME ELAPSED_TIME AIR_TIME DISTANCE WHEELS_O
N
## 1       NA         NA            205           NA       NA     1448        N
A
## 2       NA         NA            120           NA       NA      868        N
A
## 3       NA         NA             87           NA       NA      429        N
A
## 4       NA         NA             95           NA       NA      364        N
A
## 5       NA         NA             90           NA       NA      199        N
A
## 6       NA         NA            190           NA       NA      987        N
A
##   TAXI_IN SCHEDULED_ARRIVAL ARRIVAL_TIME ARRIVAL_DELAY DIVERTED CANCELLED
## 1      NA              600           NA            NA        0         1
## 2      NA              500           NA            NA        0         1
## 3      NA              637           NA            NA        0         1
## 4      NA              700           NA            NA        0         1
## 5      NA              700           NA            NA        0         1
## 6      NA              755           NA            NA        0         1
##   CANCELLATION_REASON AIR_SYSTEM_DELAY SECURITY_DELAY AIRLINE_DELAY
## 1                   A              NA             NA            NA
## 2                   B              NA             NA            NA
## 3                   B              NA             NA            NA
## 4                   B              NA             NA            NA
## 5                   A              NA             NA            NA
## 6                   B              NA             NA            NA
##   LATE_AIRCRAFT_DELAY WEATHER_DELAY
## 1                  NA            NA
## 2                  NA            NA
```

```
## 3                NA              NA
## 4                NA              NA
## 5                NA              NA
## 6                NA              NA
```

```
head(vuelos %>% filter(is.na(vuelos$ARRIVAL_DELAY)))
```

```
##   YEAR MONTH DAY DAY_OF_WEEK AIRLINE FLIGHT_NUMBER TAIL_NUMBER ORIGIN_AIRPOR
T
## 1 2015     1   1           4      AS           136      N431AS            AN
C
## 2 2015     1   1           4      AA          2459      N3BDAA            PH
X
## 3 2015     1   1           4      OO          5254      N746SK            MA
F
## 4 2015     1   1           4      MQ          2859      N660MQ            SG
F
## 5 2015     1   1           4      OO          5460      N583SW            RD
D
## 6 2015     1   1           4      MQ          2926      N932MQ            CH
S
##   DESTINATION_AIRPORT SCHEDULED_DEPARTURE DEPARTURE_TIME DEPARTURE_DELAY
## 1                 SEA                 135             NA              NA
## 2                 DFW                 200             NA              NA
## 3                 IAH                 510             NA              NA
## 4                 DFW                 525             NA              NA
## 5                 SFO                 530             NA              NA
## 6                 DFW                 545             NA              NA
##   TAXI_OUT WHEELS_OFF SCHEDULED_TIME ELAPSED_TIME AIR_TIME DISTANCE WHEELS_O
N
## 1       NA         NA            205           NA       NA     1448        N
A
## 2       NA         NA            120           NA       NA      868        N
A
## 3       NA         NA             87           NA       NA      429        N
A
## 4       NA         NA             95           NA       NA      364        N
A
## 5       NA         NA             90           NA       NA      199        N
A
## 6       NA         NA            190           NA       NA      987        N
A
##   TAXI_IN SCHEDULED_ARRIVAL ARRIVAL_TIME ARRIVAL_DELAY DIVERTED CANCELLED
## 1      NA              600           NA            NA        0         1
## 2      NA              500           NA            NA        0         1
## 3      NA              637           NA            NA        0         1
## 4      NA              700           NA            NA        0         1
## 5      NA              700           NA            NA        0         1
## 6      NA              755           NA            NA        0         1
##   CANCELLATION_REASON AIR_SYSTEM_DELAY SECURITY_DELAY AIRLINE_DELAY
## 1                   A               NA             NA            NA
## 2                   B               NA             NA            NA
## 3                   B               NA             NA            NA
## 4                   B               NA             NA            NA
## 5                   A               NA             NA            NA
## 6                   B               NA             NA            NA
##   LATE_AIRCRAFT_DELAY WEATHER_DELAY
## 1                  NA            NA
## 2                  NA            NA
```

```
## 3                   NA             NA
## 4                   NA             NA
## 5                   NA             NA
## 6                   NA             NA
```

De momento como lo que queremos es trabajar con los vuelos retrasados vamos a eliminar los valores nulos de estas variables

```
vuelos_reduc <- vuelos_reduc[!is.na(vuelos_reduc$DEPARTURE_DELAY),]
vuelos_reduc <- vuelos_reduc[!is.na(vuelos_reduc$ARRIVAL_DELAY),]
```

Comprobamos que ya no quedan valores nulos:

```
colSums(is.na(vuelos_reduc))
```

```
##               MONTH                 DAY         DAY_OF_WEEK             AIRL
INE
##                   0                   0                   0
0
##       FLIGHT_NUMBER         ORIGIN_CODE DESTINATION_AIRPORT SCHEDULED_DEPART
URE
##                   0                   0                   0
0
##      DEPARTURE_TIME     DEPARTURE_DELAY      SCHEDULED_TIME        ELAPSED_T
IME
##                   0                   0                   0
0
##            AIR_TIME            DISTANCE   SCHEDULED_ARRIVAL        ARRIVAL_T
IME
##                   0                   0                   0
0
##       ARRIVAL_DELAY
##                   0
```

```
summary(vuelos_reduc)
```

```
##      MONTH             DAY          DAY_OF_WEEK       AIRLINE
##  Min.   : 1.000   Min.   : 1.00   Min.   :1.000   WN     :87041
##  1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.:2.000   DL     :61000
##  Median : 7.000   Median :16.00   Median :4.000   AA     :49798
##  Mean   : 6.551   Mean   :15.72   Mean   :3.933   OO     :40415
##  3rd Qu.: 9.000   3rd Qu.:23.00   3rd Qu.:6.000   EV     :38768
##  Max.   :12.000   Max.   :31.00   Max.   :7.000   UA     :35531
##                                                   (Other):87400
##  FLIGHT_NUMBER   ORIGIN_CODE     DESTINATION_AIRPORT SCHEDULED_DEPARTURE
##  Min.   :   1   ATL    : 24118   ATL    : 23896      Min.   :   3
##  1st Qu.: 730   ORD    : 19251   ORD    : 18883      1st Qu.: 916
##  Median :1683   DFW    : 16229   DFW    : 16426      Median :1324
##  Mean   :2164   DEN    : 13410   LAX    : 13583      Mean   :1328
##  3rd Qu.:3211   LAX    : 13275   DEN    : 13338      3rd Qu.:1730
##  Max.   :7438   IAH    : 10202   PHX    : 10321      Max.   :2359
##                 (Other):303468   (Other):303506
##  DEPARTURE_TIME DEPARTURE_DELAY    SCHEDULED_TIME   ELAPSED_TIME
##  Min.   :   1   Min.   : -68.000   Min.   : 20.0   Min.   : 16
##  1st Qu.: 920   1st Qu.:  -5.000   1st Qu.: 85.0   1st Qu.: 82
##  Median :1329   Median :  -2.000   Median :123.0   Median :118
##  Mean   :1334   Mean   :   9.318   Mean   :141.8   Mean   :137
##  3rd Qu.:1739   3rd Qu.:   7.000   3rd Qu.:174.0   3rd Qu.:168
##  Max.   :2400   Max.   :1380.000   Max.   :705.0   Max.   :733
##
##     AIR_TIME        DISTANCE      SCHEDULED_ARRIVAL  ARRIVAL_TIME
##  Min.   :  8.0   Min.   :  31.0   Min.   :   1      Min.   :   1
##  1st Qu.: 60.0   1st Qu.: 373.0   1st Qu.:1110      1st Qu.:1058
##  Median : 94.0   Median : 650.0   Median :1518      Median :1511
##  Mean   :113.5   Mean   : 823.9   Mean   :1492      Mean   :1475
##  3rd Qu.:144.0   3rd Qu.:1065.0   3rd Qu.:1916      3rd Qu.:1916
##  Max.   :687.0   Max.   :4983.0   Max.   :2359      Max.   :2400
##
##  ARRIVAL_DELAY
##  Min.   : -87.000
##  1st Qu.: -13.000
##  Median :  -5.000
##  Mean   :   4.462
##  3rd Qu.:   8.000
##  Max.   :1384.000
##
```

Comprobamos los datos de meses y días, por ver que no hay valores extraños. También revisaremos que no hay distancias ni tiempos horarios negativos.

```
#Comprobamos si hay valores extraños en las variables día, día de la semana y m
es
month_wrong <- which(vuelos_reduc$MONTH > 12 | vuelos_reduc$MONTH < 1)
month_wrong
```

```
## integer(0)
```

```
day_wrong <- which(vuelos_reduc$DAY > 31 | vuelos_reduc$DAY < 1)
day_wrong
```

```
## integer(0)
```

```
day_week_wrong <- which(vuelos_reduc$DAY_OF_WEEK > 7 | vuelos_reduc$DAY_OF_WEEK
<1)
day_week_wrong
```

```
## integer(0)
```

Comprobamos que los valores de distancias y horas en base a los valores mínimos y máximos son correctos. En el caso de los aeropuertos, conocemos que hay aeropuertos que tienen diferente nomenclatura para un mismo aeropuerto, de momento no vamos a realizar ninguna modificación sobre este dato, supondremos que son aeropuertos diferentes.

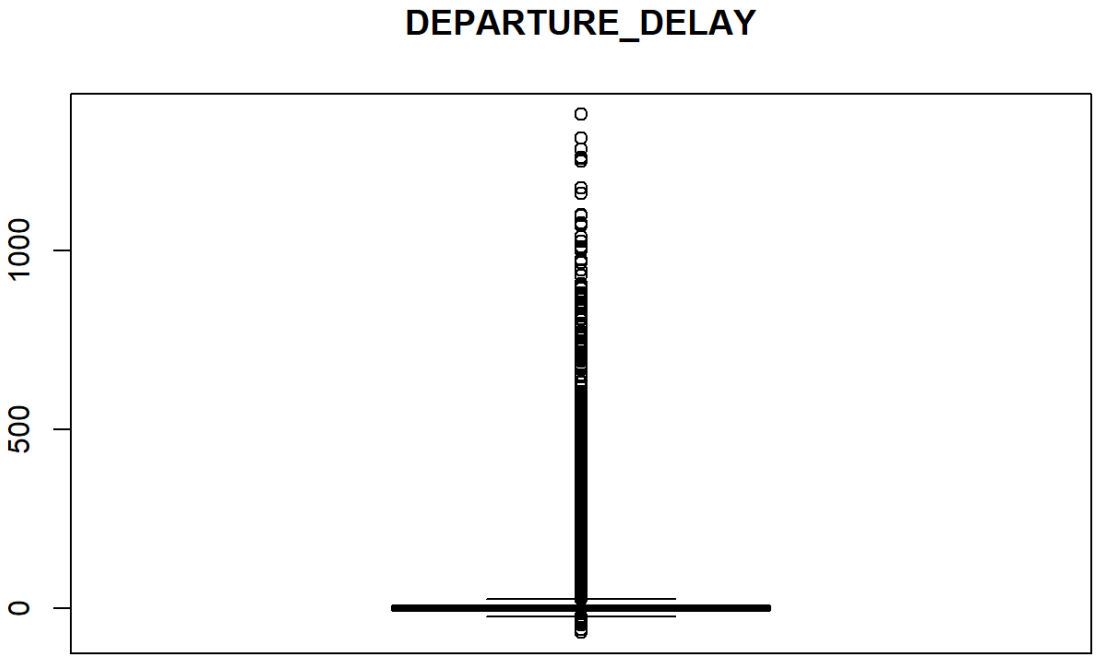
3.2 Identificación y tratamiento de valores extremos

Vamos a identificar los posibles valores extermos que tenemos en las variables de tiempo y distancia. En base a los resultados decidiremos que acciones tomar con estas variables.

Para ello vamos representar las variables con un diagrama de cajas y bigotes que nos mostrará visualmente dichas variables a través de sus cuartiles.

Las lineas que salen de las cajas se conocen como “bigotes” e indican la variabilidad fuera de los cuartiles superior e inferior. Los valores extremos (outliers) se representarán como puntos individuales. Dentro de la caja, la línea que vemos nos muestra la mediana.

```
boxplot(vuelos_reduc$DEPARTURE_DELAY, main="DEPARTURE_DELAY")
```




```
table(vuelos_reduc$DEPARTURE_DELAY [1:200])
```

```
##
## -11 -10  -9  -8  -7  -6  -5  -4  -3  -2  -1   0   1   2   3   5   6   8   9
10
##   1   1   4   7  15   8  14  14  24  12   9  15   5   2   3   4   2   7   2
4
##  11  12  13  14  15  16  17  18  21  22  25  28  31  32  33  35  38  40  41
46
##   1   1   1   4   1   1   1   1   1   4   2   1   1   1   1   1   1   2   1
1
##  47  53  57  64  70  72  76  77  83  87  90  95  96 104 117 126 140 195
##   2   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
```

```
head(table(boxplot.stats(vuelos_reduc$DEPARTURE_DELAY)$out))
```

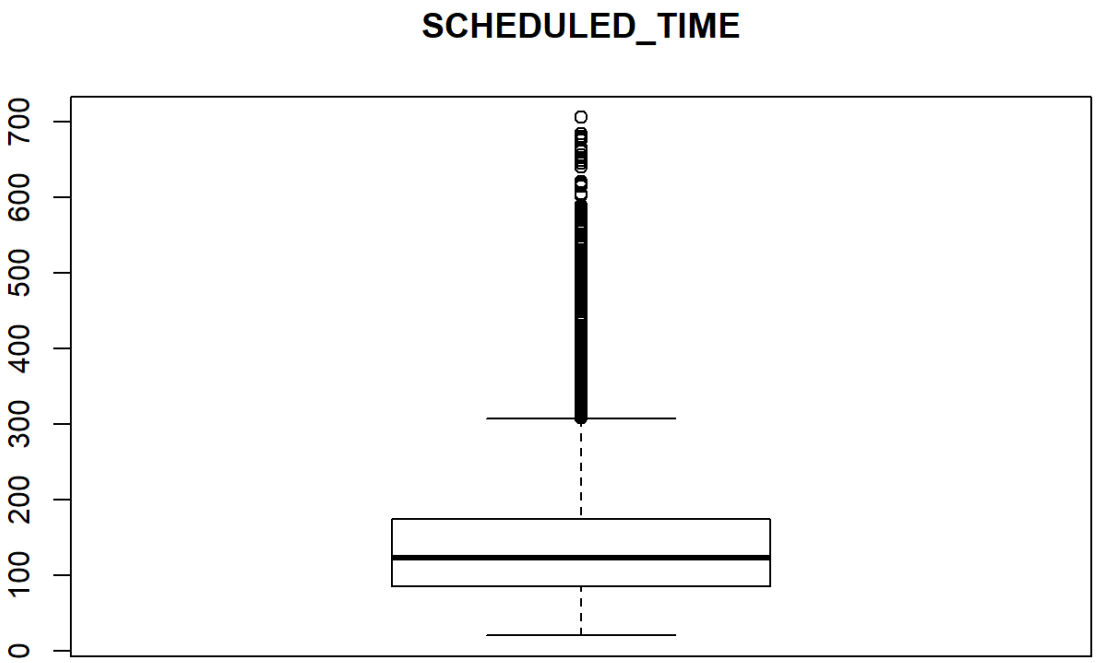
```
##
## -68 -61 -48 -44 -40 -39
##   1   1   1   1   1   1
```

En el diagrama de cajas para la variable DEPARTURE_DELAY podemos apreciar como la mayoría de los vuelos no se retrasan y por eso la caja y los bigotes toman valores de 0 (aprox). Por este motivo, todos los retrasos se consideran valores outliers y, dado que el propósito de esta variable es mostrar los retrasos en las salidas, no podemos eliminar estos valores ya que son los que le dan sentido a la variable. Por otro lado, observamos el amplio rango de valores outliers (retrasos) que hay, llegando a superar valores de 1500 minutos, aunque sí que es cierto que la mayoría se concentran entre los valores de <0 a 1000 minutos.

En las tablas podemos observar con más detalle los cinco primeros valores que toma esta variable y, comprobamos la disparidad de los retrasos.

Los valores negativos nos indican que los vuelos salieron antes de su hora programada en un primer momento.

```
boxplot(vuelos_reduc$SCHEDULED_TIME, main="SCHEDULED_TIME")
```



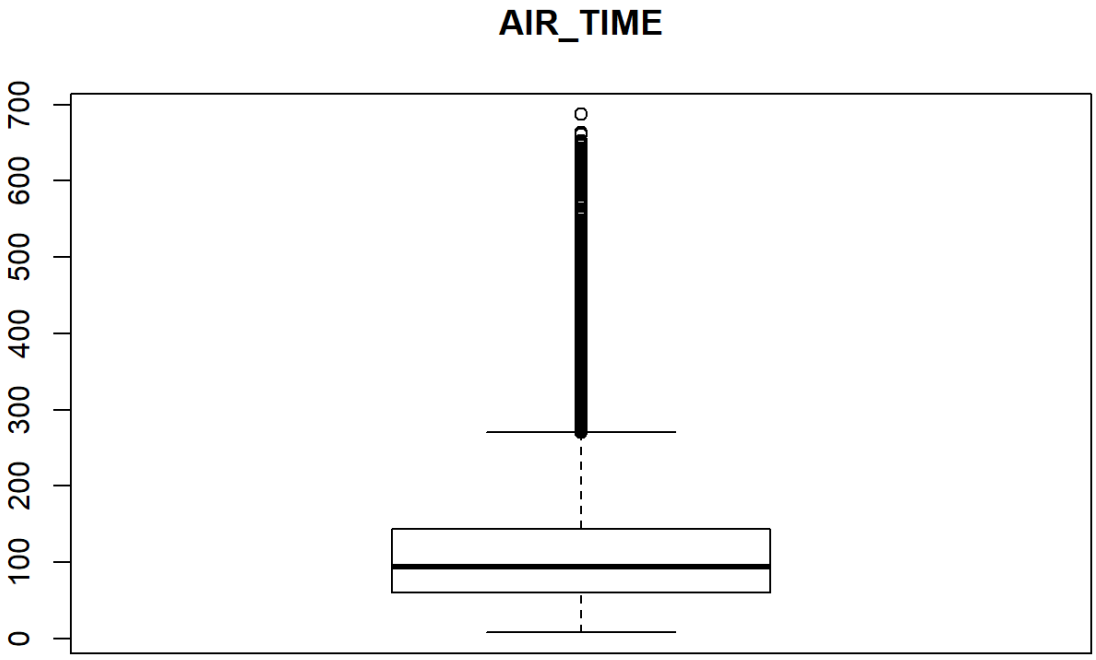

```
head(table(vuelos_reduc$SCHEDULED_TIME))
```

```
##
## 20 21 22 23 24 25
##  3  6  5 16  9 11
```

En el caso de la variable SCHEDULED_TIME (tiempo programado de vuelo) sí podemos apreciar mejor la caja y los bigotes del diagrama. Vemos que todos los valores extremos se encuentran por encima del extremo superior por lo que todos nos indican que el tiempo programado de vuelo ha sido excedido. Al igual que ocurría con la variable anteriormente analizada, de esta variable nos interesan todos sus valores por lo que no podemos eliminar sus valores extremos ya que estaríamos desperdiciando información importante.

En las tablas vemos con más detalles sus valores.

```
boxplot(vuelos_reduc$AIR_TIME, main="AIR_TIME")
```




```
head(table(boxplot.stats(vuelos_reduc$AIR_TIME)$out))
```

```
##
## 271 272 273 274 275 276
## 289 286 321 342 345 306
```

Para la vairable AIR_TIME (tiempo desde el despegue hasta el aterrizaje) vemos que la mayoría de vuelos duran entre 50 y 150 minutos aproximadamente, llegando el extremo superior a prácticamente 300 minutos. A partir de este valor nos encontramos con los datos extremos llegando hasta prácticamente los 700 minutos en su valor máximo. Dado que los valores extremos representan valores reales de duraciones de vuelo no nos interesa quitar dichos valores de la muestra ya que no se deben a posibles errores y nos aportan información valiosa. En la tabla vemos de manera más detallada la información de dicha variable.

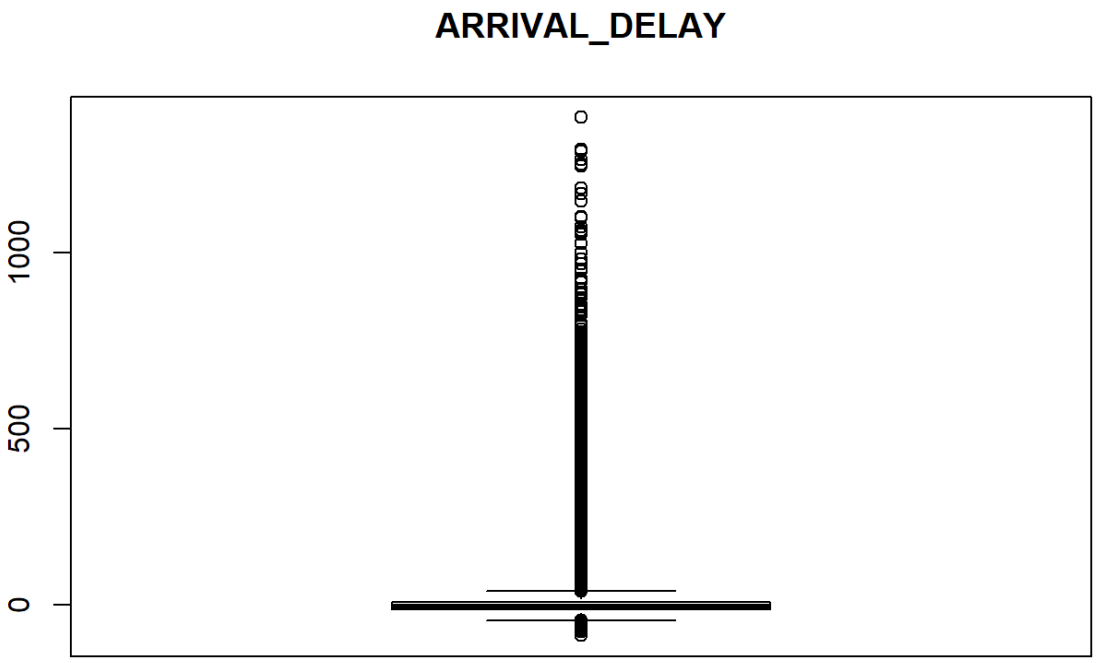
```
boxplot(vuelos_reduc$DISTANCE, main="DISTANCE")
```


```
head(table(boxplot.stats(vuelos_reduc$DISTANCE)$out))
```

```
##
## 2105 2106 2110 2116 2125 2130
##  101  255   15   44  318    9
```

Para la varialbe DISTANCE (distancia en millas) observamos que la mayor parte de los vuelos transcurren en distancias de hasta 1000 millas aproximadamente, llegando el extremo superior a las 2000 millas aproximadamente. A partir de las 2000 millas nos encontramos los valores outliers, concentrandose la mayor parte de ellos hasta las 3000 millas. Vuelos de entre 3000 millas hasta 5000 millas nos encontramos muy pocos como observamos tanto en el diagrama como en la tabla.

```
boxplot(vuelos_reduc$ARRIVAL_DELAY, main="ARRIVAL_DELAY")
```




```
head(table(vuelos_reduc$ARRIVAL_DELAY))
```

```
##
## -87 -77 -75 -72 -71 -69
##   1   1   1   1   2   1
```

Como era de esperar, el análisis de la variable ARRIVAL_DELAY (diferencia en minutos entre la llegada programada y la real) es muy similar al que vimos en la primera variable analizada (DEPARTURE_DELAY). Como ya explicamos anterioremente, la mayoría de los vuelos no sufren retrasos por lo que tanto la caja como los bigotes toman valores cercanos a 0. A partir de ahí, todos los valores ouliers que podemos observar nos muestran los retrasos en la llegada de los vuelos (vemos que el vuelo con mayor retraso superó los 1500 minutos). No podemos eliminar dichos valores ya que son la esencia de esta variable.

En resumen, el conjunto de datos que estamos tratando tiene muchos valores muy dispersos, de momento no vamos a hacer nada con ellos, los mantendremos y en algunos casos concretos, si fuese necesario, realizaremos los filtrados correspondientes.

4. Análisis de los datos

4.1 Selección de los grupos de datos

Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar)

Para poder trabajar mejor con las variables de tiempo, y dado que no nos van a interesar en principio

los minutos vamos a dejar solamente las horas para poder realizar ciertos análisis de este dato.

```
vuelos_reduc$SCHEDULED_DEPARTURE_HOUR=format(round(trunc(vuelos_reduc$SCHEDULED
_DEPARTURE/100),digits=0), nsmall=0)
vuelos_reduc$SCHEDULED_DEPARTURE_HOUR <- as.numeric(vuelos_reduc$SCHEDULED_DEPA
RTURE_HOUR)
head(vuelos_reduc$SCHEDULED_DEPARTURE_HOUR)
```

```
## [1]  8  9  5 15 10  6
```

```
table(vuelos_reduc$SCHEDULED_DEPARTURE_HOUR)
```

```
##
##     0     1     2     3     4     5     6     7     8     9    10    11    1
2
##   998   418    91    50    45  8107 27956 27092 26300 24289 25705 24847 2441
9
##    13    14    15    16    17    18    19    20    21    22    23
## 24974 22698 25344 22797 26682 22705 22823 17798 12794  7999  3022
```

```
vuelos_reduc$DEPARTURE_HOUR=format(round(trunc(vuelos_reduc$DEPARTURE_TIME/100
),digits=0), nsmall=0)
vuelos_reduc$DEPARTURE_HOUR <- as.factor(vuelos_reduc$DEPARTURE_HOUR)
head(vuelos_reduc$DEPARTURE_HOUR)
```

```
## [1]  8  9  5 15 11  5
## 25 Levels:  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 ... 2
4
```

```
table(vuelos_reduc$DEPARTURE_HOUR)
```

```
##
##     0     1     2     3     4     5     6     7     8     9    10    11    1
2
##  1768   685   176    62   714 13674 25504 24373 24836 23631 25143 24396 2410
7
##    13    14    15    16    17    18    19    20    21    22    23    24
## 23996 22876 24528 23027 25315 22492 22947 18645 13910  9200  3921    27
```

```
vuelos_reduc$ARRIVAL_HOUR=format(round(trunc(vuelos_reduc$ARRIVAL_TIME/100),dig
its=0), nsmall=0)
vuelos_reduc$ARRIVAL_HOUR <- as.factor(vuelos_reduc$ARRIVAL_HOUR)
head(vuelos_reduc$ARRIVAL_HOUR)
```

```
## [1]  9 12  6 17 15  7
## 25 Levels:  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 ... 2
4
```

```
table(vuelos_reduc$ARRIVAL_HOUR)%>% knitr::kable("html")  %>% kable_styling(pos
ition='center', font_size=12, fixed_thead=list(enabled=T))
```

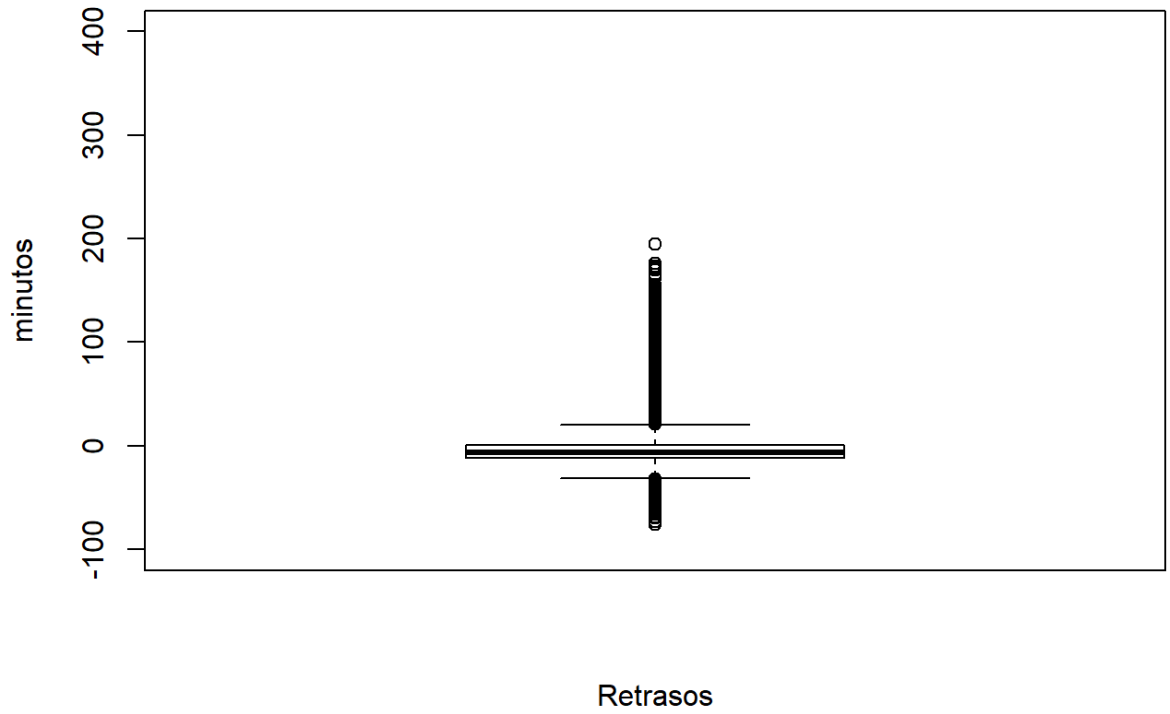
0	6941
1	2164
2	668
3	258
4	815
5	2889
6	6842
7	14086
8	19354
9	23152
10	23499
11	23813
12	23641
13	23452
14	24174
15	22845
16	25709
17	24172
18	24511
19	23947
20	24470
21	22907
22	19929
23	15512
Var1	**Freq**

Creamos una nueva variables que identifique el retraso total, en base a la perdida o ganancia en el tiempo de llegada frente al tiempo de adelanto o retraso en la salida

```
vuelos_reduc <- mutate(vuelos_reduc, RETRASO_TOTAL=ARRIVAL_DELAY - DEPARTURE_DE
LAY)

str(vuelos_reduc)
```

```
## 'data.frame':    399953 obs. of  21 variables:
##  $ MONTH                   : int  11 4 4 7 2 7 12 2 3 2 ...
##  $ DAY                     : int  16 7 1 28 9 23 24 22 8 8 ...
##  $ DAY_OF_WEEK             : int  1 2 3 2 1 4 4 7 7 7 ...
##  $ AIRLINE                 : Factor w/ 14 levels "AA","AS","B6",..: 5 14 4 5
14 1 14 10 8 4 ...
##  $ FLIGHT_NUMBER           : int  5084 1023 2182 4330 1963 2148 1915 4636 29
50 2104 ...
##  $ ORIGIN_CODE             : Factor w/ 628 levels "10135","10136",..: 327 35
8 439 504 483 346 344 593 535 523 ...
##  $ DESTINATION_AIRPORT     : Factor w/ 629 levels "10135","10136",..: 614 57
7 328 459 500 490 368 432 573 393 ...
##  $ SCHEDULED_DEPARTURE     : int  825 930 540 1545 1055 600 1125 1956 1145 1
935 ...
##  $ DEPARTURE_TIME          : int  819 943 538 1559 1105 553 1124 2002 1242 1
932 ...
##  $ DEPARTURE_DELAY         : int  -6 13 -2 14 10 -7 -1 6 57 -3 ...
##  $ SCHEDULED_TIME          : int  128 240 57 108 190 74 80 63 68 137 ...
##  $ ELAPSED_TIME            : int  129 214 47 105 166 68 86 57 68 140 ...
##  $ AIR_TIME                : int  111 201 28 75 153 47 64 35 51 95 ...
##  $ DISTANCE                : int  674 1407 153 468 1363 184 439 216 268 680
...
##  $ SCHEDULED_ARRIVAL       : int  933 1230 637 1733 1605 714 1345 2059 1253
2052 ...
##  $ ARRIVAL_TIME            : int  928 1217 625 1744 1551 701 1350 2059 1350
2052 ...
##  $ ARRIVAL_DELAY           : int  -5 -13 -12 11 -14 -13 5 0 57 0 ...
##  $ SCHEDULED_DEPARTURE_HOUR: num  8 9 5 15 10 6 11 19 11 19 ...
##  $ DEPARTURE_HOUR          : Factor w/ 25 levels " 0"," 1"," 2",..: 9 10 6 1
6 12 6 12 21 13 20 ...
##  $ ARRIVAL_HOUR            : Factor w/ 25 levels " 0"," 1"," 2",..: 10 13 7
18 16 8 14 21 14 21 ...
##  $ RETRASO_TOTAL           : int  1 -26 -10 -3 -24 -6 6 -6 0 3 ...
```

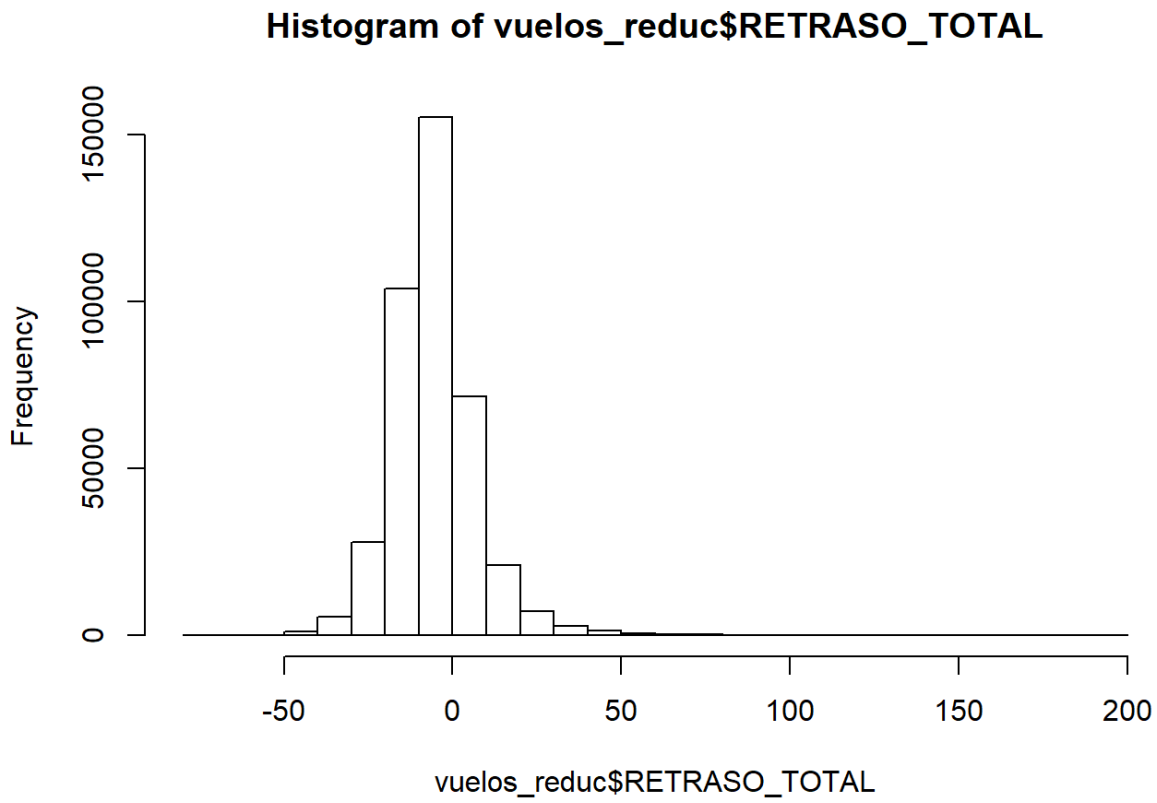
```
boxplot(vuelos_reduc$RETRASO_TOTAL , xlab="Retrasos", ylab="minutos", ylim=c(-1
00, 400) )
```



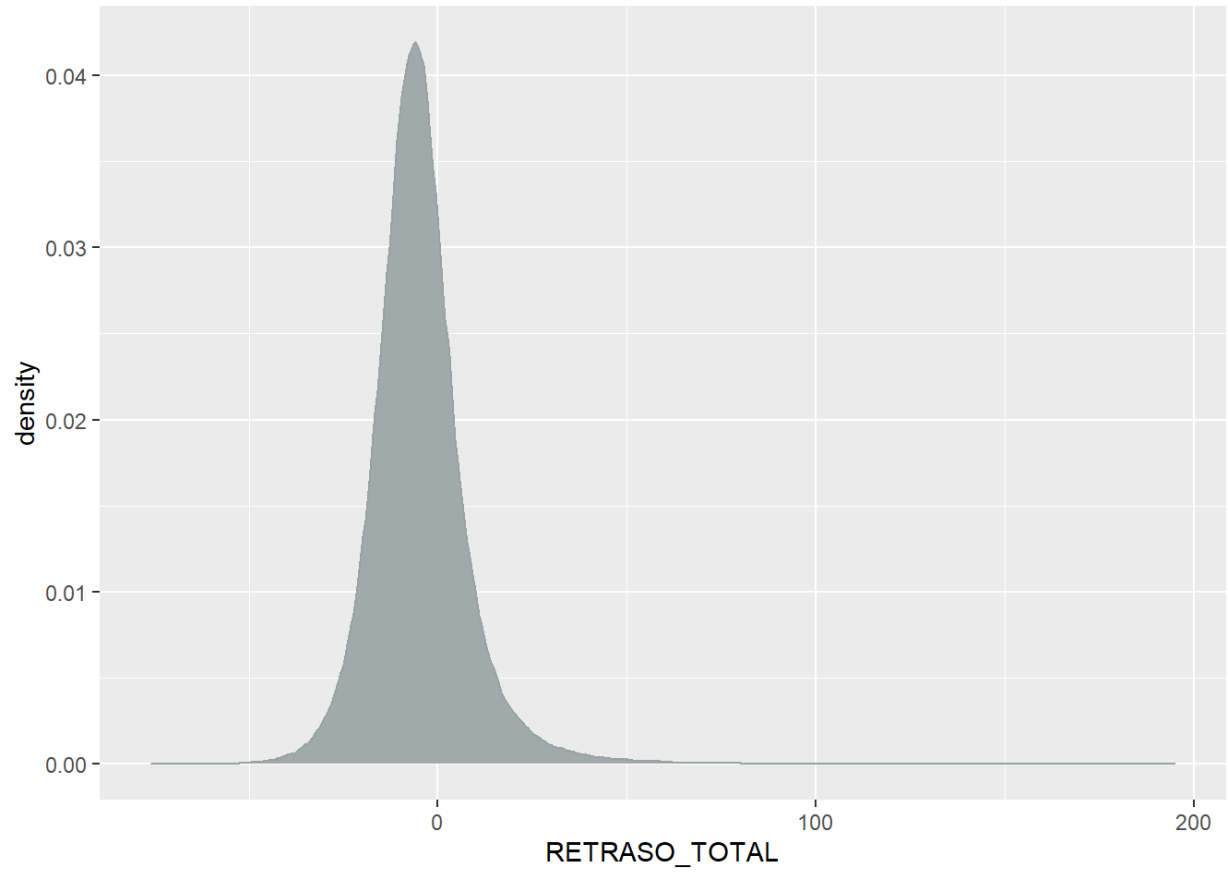

```
table(boxplot.stats(vuelos_reduc$RETRASO_TOTAL)$out)
```

```
##
##  -76  -73  -70  -69  -66  -65  -64  -63  -62  -61  -60  -59  -58  -57  -56
-55
##    1    1    1    1    1    1    2    1    4    3    3    3    2    9    9
12
##  -54  -53  -52  -51  -50  -49  -48  -47  -46  -45  -44  -43  -42  -41  -40
-39
##   15   16   32   17   40   47   52   52   76   84   98  108  151  170  215
252
##  -38  -37  -36  -35  -34  -33  -32   21   22   23   24   25   26   27   28
29
##  245  330  392  457  539  630  795 1064 1015  883  810  700  690  595  541
484
##   30   31   32   33   34   35   36   37   38   39   40   41   42   43   44
45
##  433  372  389  356  325  304  262  262  214  232  187  196  144  153  143
152
##   46   47   48   49   50   51   52   53   54   55   56   57   58   59   60
61
##  123   97  119  110  134   84   78   86   75   76   64   71   61   66   56
48
##   62   63   64   65   66   67   68   69   70   71   72   73   74   75   76
77
##   40   37   43   36   30   38   27   23   31   32   20   25   35   28   30
21
##   78   79   80   81   82   83   84   85   86   87   88   89   90   91   92
93
##   25   20   21   16   13   14   16   20   13   17   16    3    8    9    6
10
##   94   95   96   97   98   99  100  101  102  103  104  105  106  107  108
109
##    5    8    7    7    9    6    6    7    5    4   11    8    7    7    6
8
##  110  111  112  113  114  115  116  117  118  119  120  121  122  123  125
126
##    6    4    9    3    6    3    6    3    5    4    4    2    2    1    3
2
##  127  128  129  130  131  132  133  134  136  137  138  139  140  141  142
143
##    4    4    4    3    6    2    1    3    1    1    2    1    3    2    2
1
##  144  145  146  147  148  149  150  152  153  156  162  165  170  172  173
176
##    2    1    2    1    3    1    1    1    1    1    1    1    1    1    2
1
##  195
##    1
```

```
hist(vuelos_reduc$RETRASO_TOTAL)
```




```
vuelos_reduc %>%
  ggplot( aes(x=RETRASO_TOTAL)) +
  geom_density(fill="#99A3A4", color="#99A3A4", alpha=0.9)
```



En los gráficos podemos ver como, del mismo modo que se tenía en el gráfico de cajas y bigotes del apartado 3 para las variables DEPARTURE_DELAY y ARRIVAL_DELAY, para esta nueva variable, se tiene un comportamiento similar, como era de esperar. La gran mayoría de los vuelos se concentran alrededor del 0, es decir que los vuelos salen y llegan en los tiempos establecidos. Sin embargo el gráfico se alarga a ambos lados, sobre todo hacia la derecha, debido a la gran dispersión de tiempos de vuelos retrasados (diferencia positiva).

4.2 Comprobación de la normalidad y homogeneidad de la varianza

Utilizaremos la prueba Shapiro-Wilk test para comprobar la normalidad de la muestra para los valores DEPARTURE_DELAY, ARRIVAL_DELAY y DISTANCE

```
shapiro.test(vuelos_reduc$DEPARTURE_DELAY[1:5000])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  vuelos_reduc$DEPARTURE_DELAY[1:5000]
## W = 0.45789, p-value < 2.2e-16
```

```
shapiro.test(vuelos_reduc$ARRIVAL_DELAY[1:5000])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  vuelos_reduc$ARRIVAL_DELAY[1:5000]
## W = 0.58009, p-value < 2.2e-16
```

```
shapiro.test(vuelos_reduc$DISTANCE[1:5000])
```

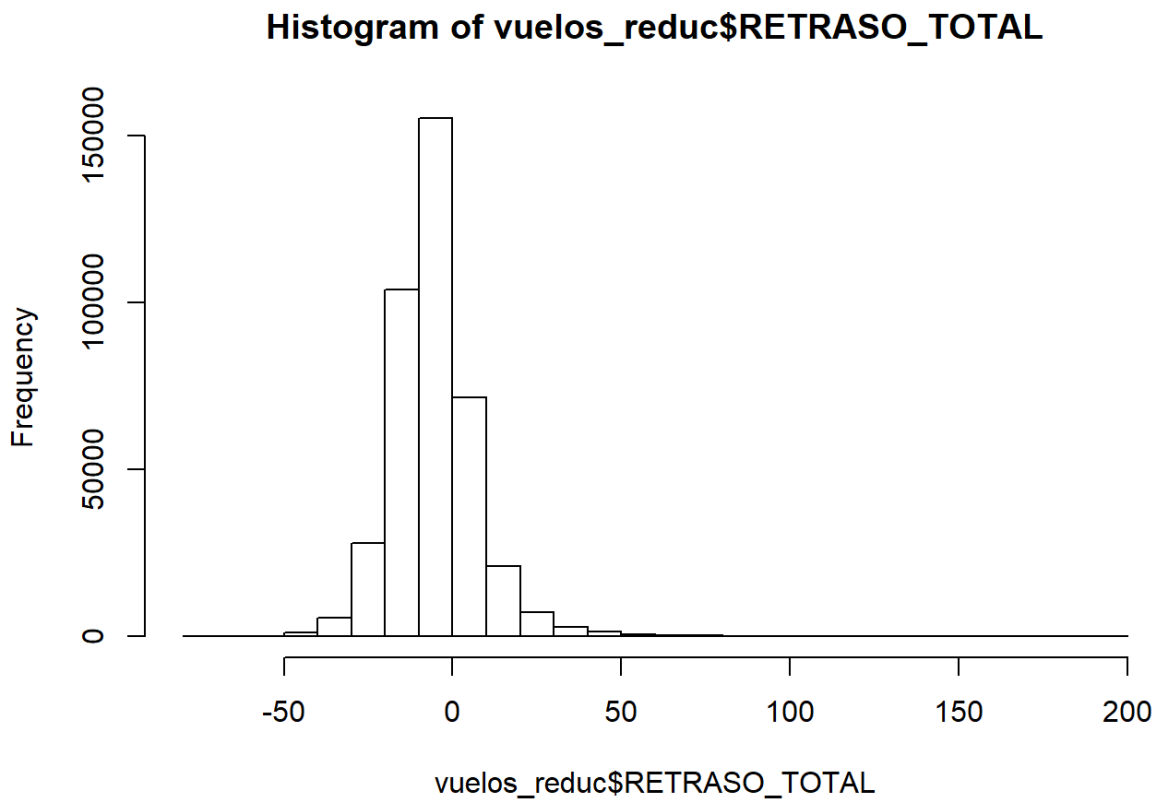
```
##
##  Shapiro-Wilk normality test
##
## data:  vuelos_reduc$DISTANCE[1:5000]
## W = 0.88101, p-value < 2.2e-16
```

```
shapiro.test(vuelos_reduc$RETRASO_TOTAL[1:5000])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  vuelos_reduc$RETRASO_TOTAL[1:5000]
## W = 0.91736, p-value < 2.2e-16
```

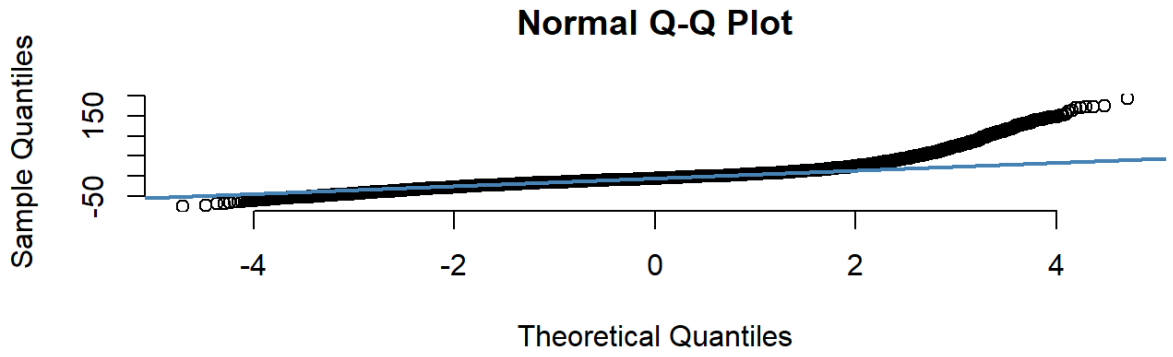
La prueba de Saphiro-Wilk, solo es posible para un máximo de 5000 registros, por lo que hemos realizado la prueba con un subconjunto con esa cantidad, y nos da como resultado que debemos rechazar la hipótesis nula, es decir, nos indicaría que las variables no siguen una distribución normal. Sin embargo, por el teorema central del límite, la distribución de la media de cualquier muestra de datos se considera cada vez más normal según aumenta el tamaño de la misma y para muestras superiores con N>30 se puede suponer normalidad, dado que podría aproximarse a una disribución normal.

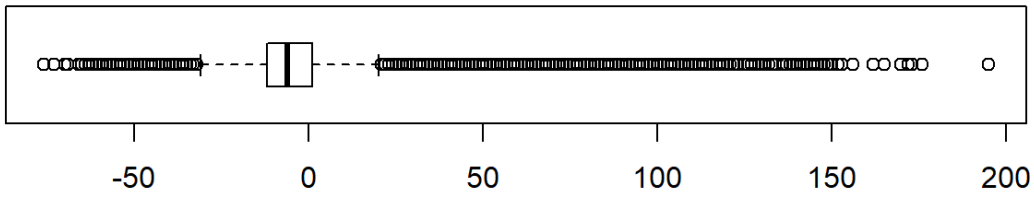
```
hist(vuelos_reduc$RETRASO_TOTAL)
```



Aparentemenete, para esta variable, según el gráfico podríamos decir que la variable sigue una distribución normal.
Vamos a intentar generar Q-QPlot para mostrar gráficamente si la muestra sigue una distribución normal.

```
par(mfrow=c(2,1))
qqnorm(vuelos_reduc$RETRASO_TOTAL, pch = 1, frame = FALSE)
qqline(vuelos_reduc$RETRASO_TOTAL, col = "steelblue", lwd = 2)
boxplot(vuelos_reduc$RETRASO_TOTAL, horizontal=T)
```





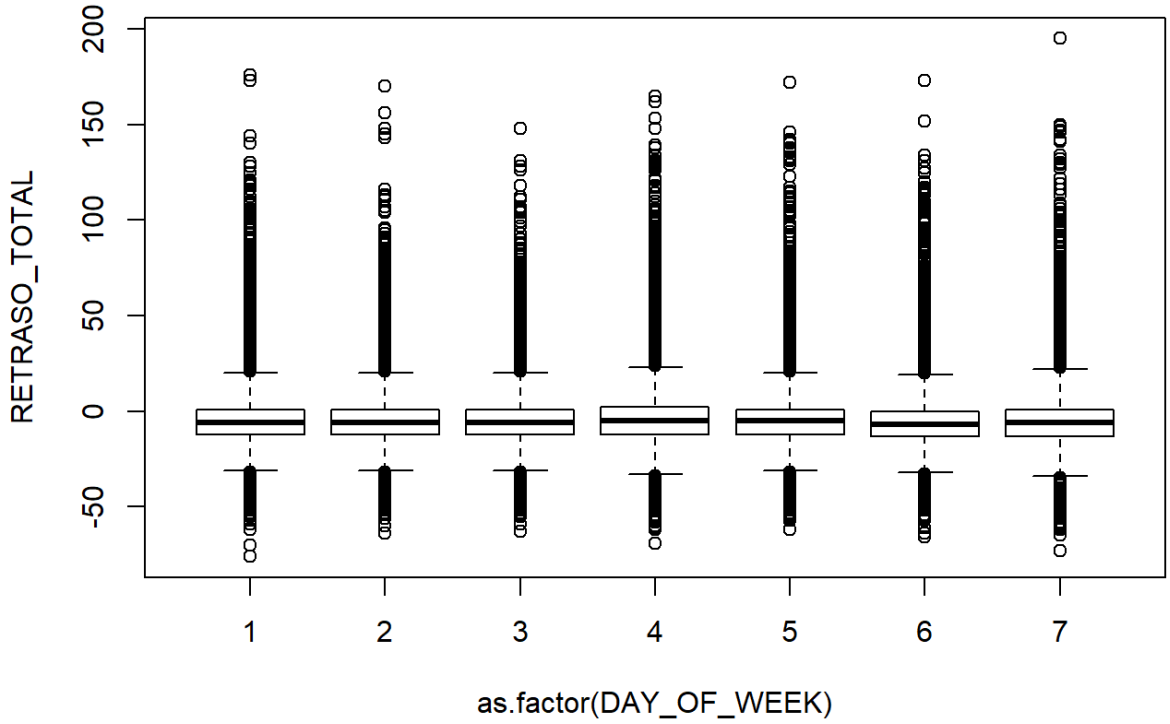
Contrastamos el gráfico Q-QPlot contra el gráfico de cajas, para ver el comportamiento. En el gráfico de arriba, podemos ver que los valores de la variable no se pegan a la recta, de hecho, se levanta hacia la parte de la derecha despegándose de la recta y solo unos cuantos valores centrales se tienen alrededor de la recta. Esto nos está identificando que esta variable tiene una gran cantidad de valores extremos que hacen que tengamos que se genere esa cola a la derecha que se separa de la línea. La dispersión de valores pone en duda ldistribución normal de la variable.

Como no tenemos seguridad de si la población que estamos estudiando sigue una distribución normal, para comprobar la homogeneidad de las varianzas u homocedasticidad, utilizaremos el test de Levene. Compararemos nuestra variable con la variable DAY_OF_WEEK (realizamos una agrupación de los datos por día de la semana). Tomaremos como hipótesis nula la semajanza de variables y como hipótesis alternativa la no igualdad en la varianza de las varibales con un índice de significación α=0.05

```
str(vuelos_reduc)
```

```
## 'data.frame':    399953 obs. of  21 variables:
##  $ MONTH                   : int  11 4 4 7 2 7 12 2 3 2 ...
##  $ DAY                     : int  16 7 1 28 9 23 24 22 8 8 ...
##  $ DAY_OF_WEEK             : int  1 2 3 2 1 4 4 7 7 7 ...
##  $ AIRLINE                 : Factor w/ 14 levels "AA","AS","B6",..: 5 14 4 5
14 1 14 10 8 4 ...
##  $ FLIGHT_NUMBER           : int  5084 1023 2182 4330 1963 2148 1915 4636 29
50 2104 ...
##  $ ORIGIN_CODE             : Factor w/ 628 levels "10135","10136",..: 327 35
8 439 504 483 346 344 593 535 523 ...
##  $ DESTINATION_AIRPORT     : Factor w/ 629 levels "10135","10136",..: 614 57
7 328 459 500 490 368 432 573 393 ...
##  $ SCHEDULED_DEPARTURE     : int  825 930 540 1545 1055 600 1125 1956 1145 1
935 ...
##  $ DEPARTURE_TIME          : int  819 943 538 1559 1105 553 1124 2002 1242 1
932 ...
##  $ DEPARTURE_DELAY         : int  -6 13 -2 14 10 -7 -1 6 57 -3 ...
##  $ SCHEDULED_TIME          : int  128 240 57 108 190 74 80 63 68 137 ...
##  $ ELAPSED_TIME            : int  129 214 47 105 166 68 86 57 68 140 ...
##  $ AIR_TIME                : int  111 201 28 75 153 47 64 35 51 95 ...
##  $ DISTANCE                : int  674 1407 153 468 1363 184 439 216 268 680
...
##  $ SCHEDULED_ARRIVAL       : int  933 1230 637 1733 1605 714 1345 2059 1253
2052 ...
##  $ ARRIVAL_TIME            : int  928 1217 625 1744 1551 701 1350 2059 1350
2052 ...
##  $ ARRIVAL_DELAY           : int  -5 -13 -12 11 -14 -13 5 0 57 0 ...
##  $ SCHEDULED_DEPARTURE_HOUR: num  8 9 5 15 10 6 11 19 11 19 ...
##  $ DEPARTURE_HOUR          : Factor w/ 25 levels " 0"," 1"," 2",..: 9 10 6 1
6 12 6 12 21 13 20 ...
##  $ ARRIVAL_HOUR            : Factor w/ 25 levels " 0"," 1"," 2",..: 10 13 7
18 16 8 14 21 14 21 ...
##  $ RETRASO_TOTAL           : int  1 -26 -10 -3 -24 -6 6 -6 0 3 ...
```

```
plot(RETRASO_TOTAL ~ as.factor(DAY_OF_WEEK), data=vuelos_reduc)
```



Según el gráfico parece que podemos tener homogeneidad en las varianzas

```
LeveneTest(RETRASO_TOTAL ~ as.factor(DAY_OF_WEEK), vuelos_reduc, center=median)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##           Df F value    Pr(>F)
## group      6  9.1256 5.206e-10 ***
##       399946
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Sin embargo, el Test de Levene nos da un p-valor cercano a 0, menor que 0.05 (95% de nivel de significación), con lo que rechazaríamos la hipotesis nula de varianzas iguales.

Comprobamos con el test de Filgner-Killeen, que habitualmente se utiliza si desconocemos si la distribución es normal o en el caso de que no se tenga una distribución normal de la varible.

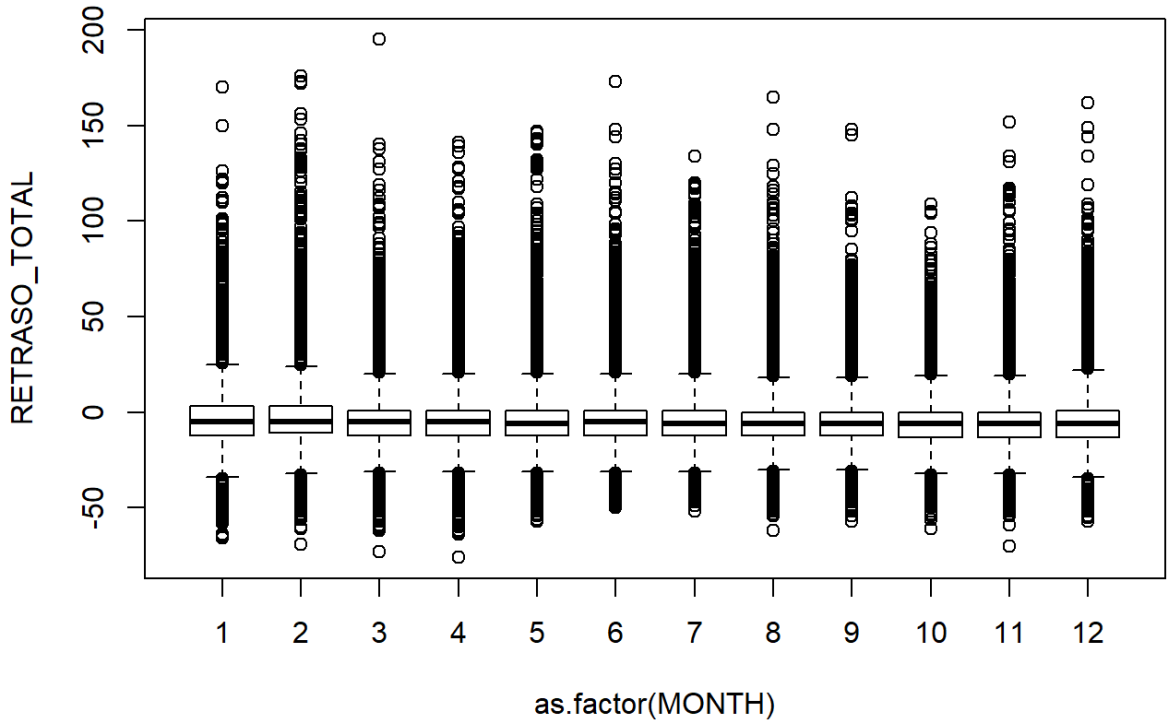
```
fligner.test(RETRASO_TOTAL ~ as.factor(DAY_OF_WEEK), data=vuelos_reduc)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  RETRASO_TOTAL by as.factor(DAY_OF_WEEK)
## Fligner-Killeen:med chi-squared = 49.164, df = 6, p-value = 6.913e-09
```

Comprobamos que para este caso, el resultado sigue siendo un p-valor por debajo de 0.05, por lo que también rechazaríamos la hipótesis nula. Es decir, parece que se tienen diferencias significativas entre las varianzas de ambos grupos.

Comprobamos con otras variables como el mes o la hora de salida.

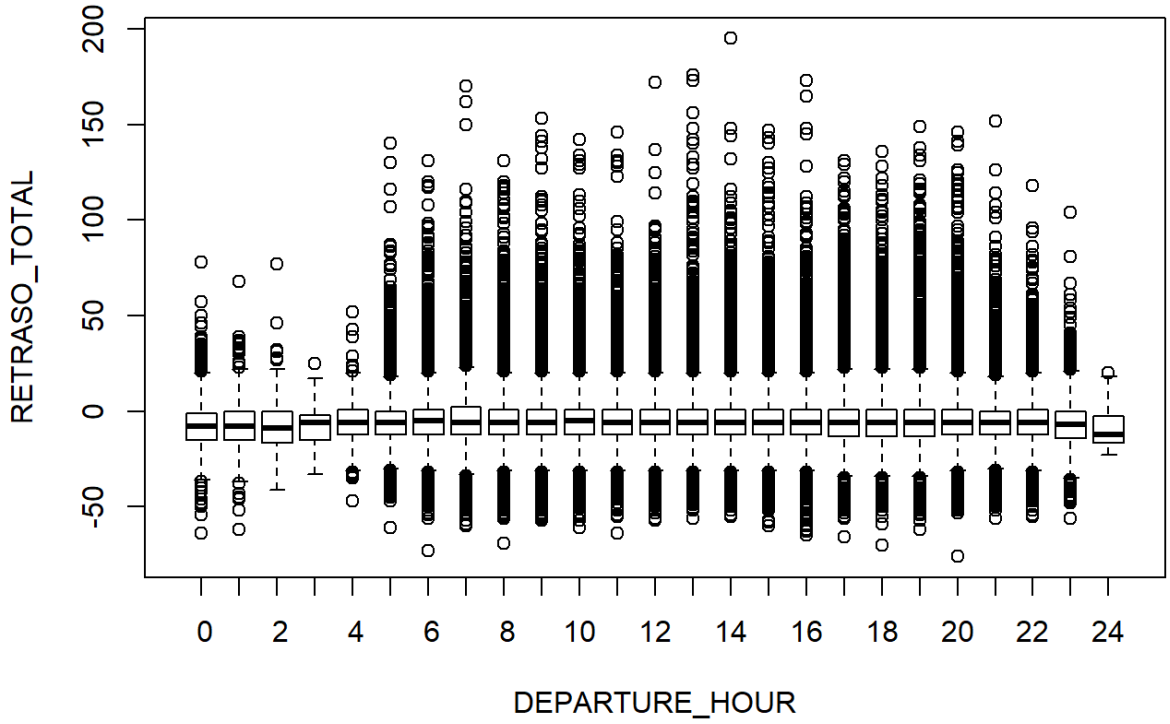
```
plot(RETRASO_TOTAL ~ as.factor(MONTH), data=vuelos_reduc)
```




```
LeveneTest(RETRASO_TOTAL ~ as.factor(MONTH), vuelos_reduc, center=mean)
```

```
## Levene's Test for Homogeneity of Variance (center = mean)
##           Df F value    Pr(>F)
## group     11  106.51 < 2.2e-16 ***
##       399941
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(RETRASO_TOTAL ~ DEPARTURE_HOUR, data=vuelos_reduc)
```



En el caso de la hora de salida si que se aprecian ciertas diferencias los tramos horarios de madrugada, donde no hay apenas vuelos programados, por lo que el resultado esperado en este caso podría decirse que será la no homogeneidad de la varianza.

```
LeveneTest(RETRASO_TOTAL ~ DEPARTURE_HOUR, vuelos_reduc, center=mean)
```

```
## Levene's Test for Homogeneity of Variance (center = mean)
##           Df F value    Pr(>F)
## group     24    36.8 < 2.2e-16 ***
##       399928
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

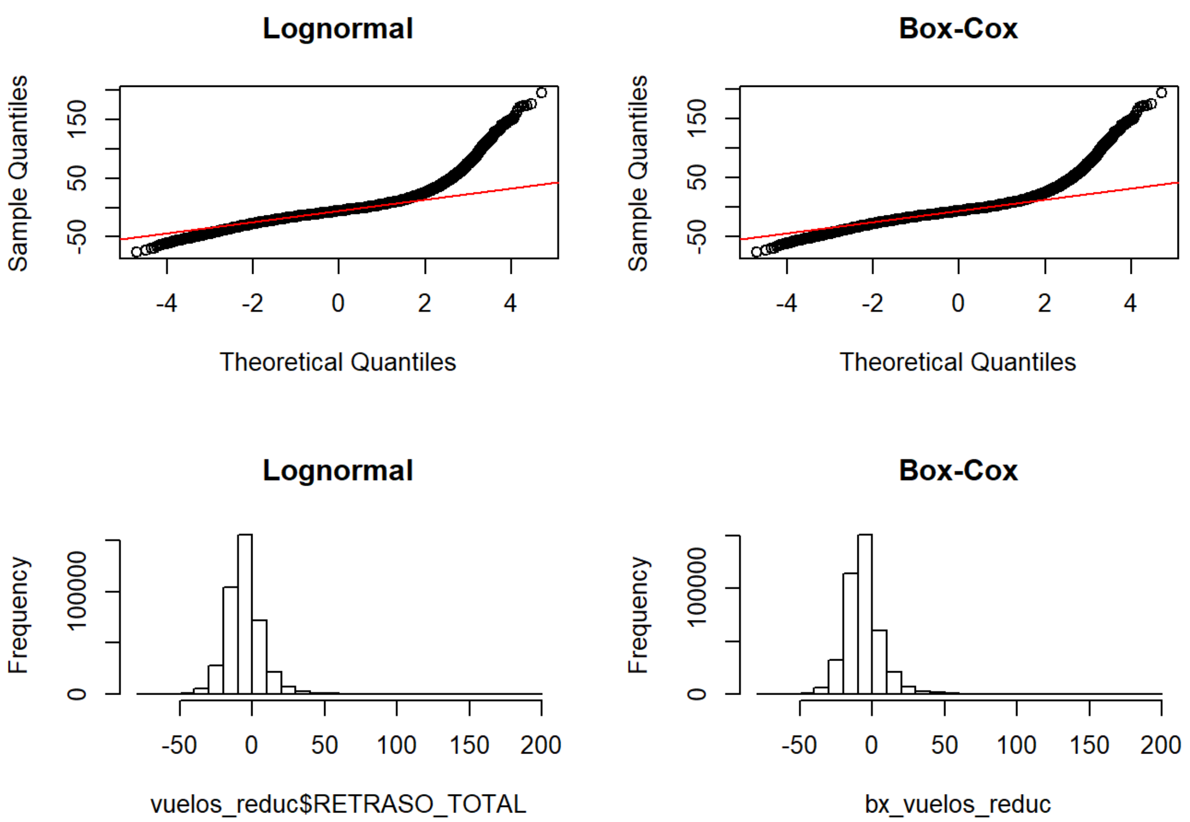
Comprobamos que el p-valor obtenido, como en los casos anteriores, nos hace rechazar la hipótesis nula de varianzas similares. Por lo que se determina que no hay homogeneidad en las variables.

Vemos si podemos conseguir una mejora de la normalidad y la homocedasticidad aplicando la transformación de Box-Cox

```
bx_vuelos_reduc <- BoxCox(vuelos_reduc$RETRASO_TOTAL, lambda = BoxCoxLambda(vue
los_reduc$RETRASO_TOTAL))
```

```
## Warning in optimize(guer.cv, c(lower, upper), x = x, nonseasonal.length =
## nonseasonal.length): NA/Inf replaced by maximum positive value

## Warning in optimize(guer.cv, c(lower, upper), x = x, nonseasonal.length =
## nonseasonal.length): NA/Inf replaced by maximum positive value

## Warning in optimize(guer.cv, c(lower, upper), x = x, nonseasonal.length =
## nonseasonal.length): NA/Inf replaced by maximum positive value

## Warning in optimize(guer.cv, c(lower, upper), x = x, nonseasonal.length =
## nonseasonal.length): NA/Inf replaced by maximum positive value

## Warning in optimize(guer.cv, c(lower, upper), x = x, nonseasonal.length =
## nonseasonal.length): NA/Inf replaced by maximum positive value

## Warning in optimize(guer.cv, c(lower, upper), x = x, nonseasonal.length =
## nonseasonal.length): NA/Inf replaced by maximum positive value

## Warning in optimize(guer.cv, c(lower, upper), x = x, nonseasonal.length =
## nonseasonal.length): NA/Inf replaced by maximum positive value
```

```
par(mfrow=c(2,2))
qqnorm(vuelos_reduc$RETRASO_TOTAL, main="Lognormal")
qqline(vuelos_reduc$RETRASO_TOTAL,col=2)
qqnorm(bx_vuelos_reduc, main="Box-Cox")
qqline(bx_vuelos_reduc,col=2)
hist(vuelos_reduc$RETRASO_TOTAL,main="Lognormal")
hist(bx_vuelos_reduc, main="Box-Cox")
```



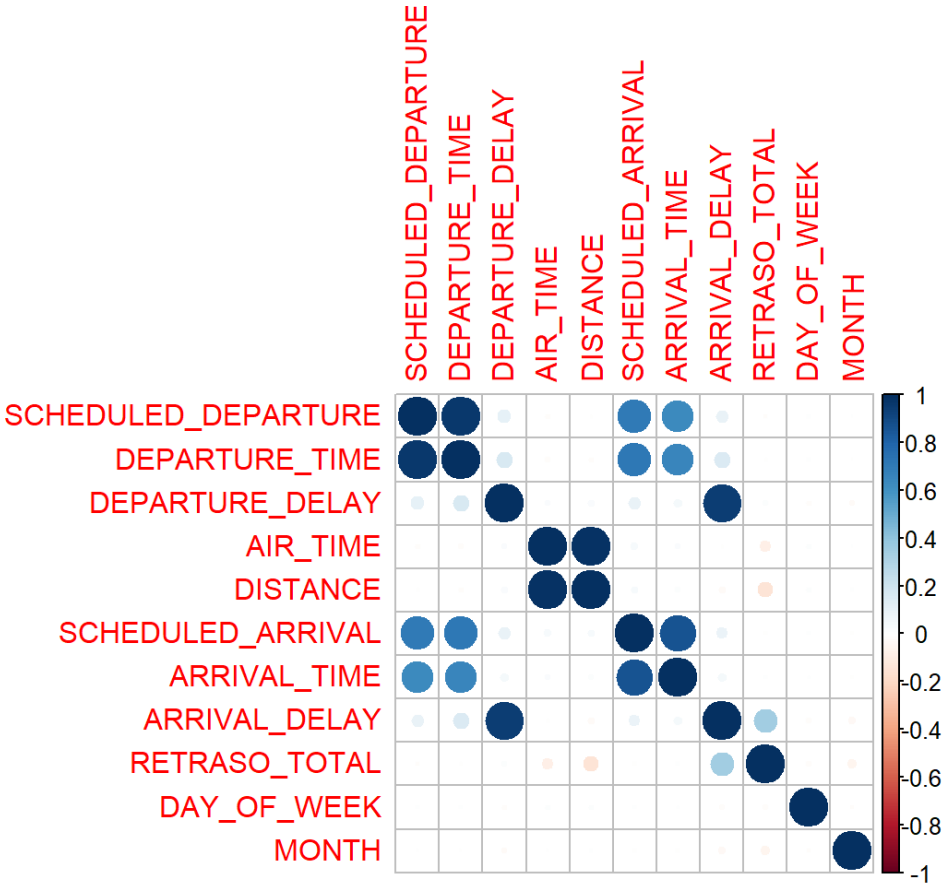
No parece que encontremos una mejora de los datos con la transformación de Box-Cox.

4.3 Aplicación de pruebas estadísticas

Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

Lo primero que vamos a hacer es comprobar la correlación entre algunas de las variables:

```
correlacion <- dplyr::select(vuelos_reduc, "SCHEDULED_DEPARTURE", "DEPARTURE_TI
ME", "DEPARTURE_DELAY",
                      "AIR_TIME", "DISTANCE", "SCHEDULED_ARRIVAL", "ARRIVAL_TIM
E","ARRIVAL_DELAY",
                      "RETRASO_TOTAL","DAY_OF_WEEK","MONTH")

corr.res<-cor(correlacion)
corrplot(corr.res,method="circle")
```



Comprobamos una fuerte relación entre el retraso en la salida y el retraso en la llegada, que puee deberse a los vuelos no retrasados o quizá pueda ser, que al contrario de lo que podríamos pensar, el retraso en la salida no es recuperado en la llegada.
También vemos una fuerte relación entre la distancia y el tiempo de vuelo, algo que era de esperar. Se tiene también una relación, aunque no tan fuerte entre el tiempo estimado de llegada y el tiempo real de llegada.

Nos preguntamos ahora por la relación entre el retraso/adelanto de un vuelo con la hora del día en la que se realiza Para ello usaremos un modelo de regresión, dado que la variable día de la semana es de tipo factor y además crearemos una nueva variable que nos indique si el vuelo se ha retrasado o no.

Primero vamos a crear la tabla de contingencia y calcularemos la estimación Odds Ratio para ver la relación entre las variables Veremos si existe relación entre la variable dependiente, en nuestro caso si hay retraso o no y las variables explicativas.
Dividimos la muestra:

```
retraso <- data.frame (RETRASO=vuelos_reduc$RETRASO_TOTAL)
retraso$RETRASO <- ifelse (retraso$RETRASO>0, "SI", "NO")
retraso <- data.frame (retraso, WEEKEND=vuelos_reduc$DAY_OF_WEEK)
#Contamos como fin de semana los viernes, sábados y domingos
retraso$WEEKEND <- ifelse ((retraso$WEEKEND=="5" | retraso$WEEKEND=="6" | retra
so$WEEKEND=="7"), "WEEKEND", "WEEKDAY")
str(retraso)
```

```
## 'data.frame':    399953 obs. of  2 variables:
##  $ RETRASO: chr  "SI" "NO" "NO" "NO" ...
##  $ WEEKEND: chr  "WEEKDAY" "WEEKDAY" "WEEKDAY" "WEEKDAY" ...
```

```
table (retraso$RETRASO)
```

```
##
##     NO     SI
## 293846 106107
```

```
tabla_retraso_dias <- with(retraso, table(retraso$RETRASO,retraso$WEEKEND))
tabla_retraso_dias %>% knitr::kable("html")  %>% kable_styling(position='cente
r', font_size=12, fixed_thead=list(enabled=T))
```

NO	172200	121646
	WEEKDAY	**WEEKEND**

Aplicamos la función chi-cuadrado de Pearson a las variables para conocer si podemos aceptar la hipótesis nula y por lo tanto las variables no están relacionadas.

```
chisq.test(tabla_retraso_dias, correct=FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tabla_retraso_dias
## X-squared = 75.036, df = 1, p-value < 2.2e-16
```

El p-value encontrado con el test chi-cuadrado es p-value < 2.2e-16 que se encuentra muy por debajo del nivel de significación marcado de 0.05, por lo que rechazamos la hipótesis nula y por lo tanto podemos concluir en este caso que existe relación entre retraso y el día de la semana.

```
library(epitools)
```

```
##
## Attaching package: 'epitools'
```

```
## The following object is masked from 'package:survival':
##
##     ratetable
```

```
oddsratio(tabla_retraso_dias, verbose = TRUE)
```

```
## $x
##
##      WEEKDAY WEEKEND
##   NO  172200  121646
##   SI   63800   42307
##
## $data
##
##         WEEKDAY WEEKEND  Total
##   NO     172200  121646 293846
##   SI      63800   42307 106107
##   Total  236000  163953 399953
##
## $p.exposed
##
##          WEEKDAY   WEEKEND     Total
##   NO    0.729661 0.7419565 0.7347013
##   SI    0.270339 0.2580435 0.2652987
##   Total 1.000000 1.0000000 1.0000000
##
## $p.outcome
##
##           WEEKDAY   WEEKEND Total
##   NO    0.5860212 0.4139788     1
##   SI    0.6012798 0.3987202     1
##   Total 0.5900693 0.4099307     1
##
## $measure
##     odds ratio with 95% C.I.
##       estimate     lower     upper
##   NO 1.0000000        NA        NA
##   SI 0.9386763 0.9253757 0.9522588
##
## $conf.level
## [1] 0.95
##
## $p.value
##     two-sided
##      midp.exact fisher.exact   chi.square
##   NO         NA           NA           NA
##   SI          0 4.329552e-18 4.621849e-18
##
## $correction
## [1] FALSE
##
## attr(,"method")
## [1] "median-unbiased estimate & mid-p exact CI"
```

```
oddsratio(tabla_retraso_dias, rev="columns", verbose = TRUE)
```

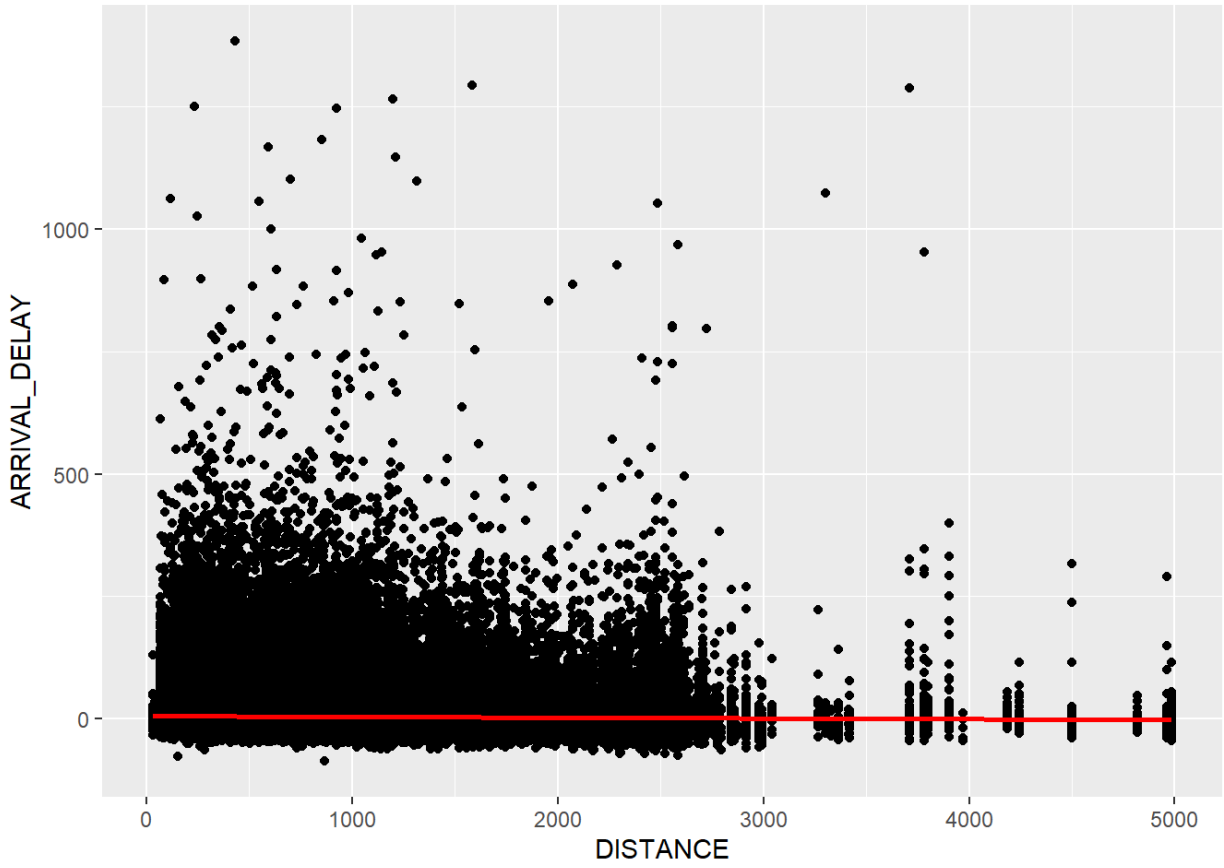
```
## $x
##
##      WEEKEND WEEKDAY
##   NO  121646  172200
##   SI   42307   63800
##
## $data
##
##         WEEKEND WEEKDAY  Total
##   NO     121646  172200 293846
##   SI      42307   63800 106107
##   Total  163953  236000 399953
##
## $p.exposed
##
##           WEEKEND  WEEKDAY     Total
##   NO    0.7419565 0.729661 0.7347013
##   SI    0.2580435 0.270339 0.2652987
##   Total 1.0000000 1.000000 1.0000000
##
## $p.outcome
##
##           WEEKEND   WEEKDAY Total
##   NO    0.4139788 0.5860212     1
##   SI    0.3987202 0.6012798     1
##   Total 0.4099307 0.5900693     1
##
## $measure
##     odds ratio with 95% C.I.
##      estimate    lower    upper
##   NO 1.000000       NA       NA
##   SI 1.065275 1.050135 1.080642
##
## $conf.level
## [1] 0.95
##
## $p.value
##     two-sided
##      midp.exact fisher.exact   chi.square
##   NO         NA           NA           NA
##   SI          0 4.329552e-18 4.621849e-18
##
## $correction
## [1] FALSE
##
## attr(,"method")
## [1] "median-unbiased estimate & mid-p exact CI"
```

El Odds Ratio nos indica que la razón entre la ocurrencia de retraso del vuelo frente a no retraso es 0,94 veces mayor en día laborable y de 1.065 veces superior en fin de semana. Identificamos que no es una diferencia muy pronunciada.

Realizamos el modelo de regresión lineal simple en el que estudiaremos la relación entre el retraso del vuelo en la llegada junto con la distancia

```
ggplot(vuelos_reduc, aes(x=DISTANCE, y=ARRIVAL_DELAY)) +
  geom_point() +
  geom_smooth(method=lm , color="red", se=FALSE)
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
retraso_modelo1 <- lm(ARRIVAL_DELAY~DISTANCE, vuelos_reduc)
retraso_modelo1
```

```
##
## Call:
## lm(formula = ARRIVAL_DELAY ~ DISTANCE, data = vuelos_reduc)
##
## Coefficients:
## (Intercept)     DISTANCE
##    5.761644    -0.001578
```

```
summary(retraso_modelo1)
```

```
##
## Call:
## lm(formula = ARRIVAL_DELAY ~ DISTANCE, data = vuelos_reduc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -91.39  -17.81   -9.43    3.27 1378.92
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.7616436  0.1047835   54.99   <2e-16 ***
## DISTANCE    -0.0015778  0.0001023  -15.42   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.36 on 399951 degrees of freedom
## Multiple R-squared:  0.0005941,  Adjusted R-squared:  0.0005916
## F-statistic: 237.7 on 1 and 399951 DF,  p-value: < 2.2e-16
```

No conseguimos un buen modelo, seguramente porque no tenemos una dependencia estrictamente lineal entre los valores. El gráfico de dispersión no presenta un ajuste de los puntos a la recta, lo que nos indica que no se tiene una fuerte relación lineal entre las variables. Según nos indica el modelo obtenido, la relación, qunque muy débil, es negativa, es decir, parecería que el retraso en la llegada disminuye con la distancia.

Probamos a incluir en el modelo la variable Scheduled_arrival y posteriormente scheduled_departure, por si podemos mejorar el modelo identificando las horas programadas de salida y de llegada.

```
retraso_modelo2 <- lm(ARRIVAL_DELAY~SCHEDULED_ARRIVAL + DISTANCE, vuelos_reduc)
retraso_modelo2
```

```
##
## Call:
## lm(formula = ARRIVAL_DELAY ~ SCHEDULED_ARRIVAL + DISTANCE, data = vuelos_red
uc)
##
## Coefficients:
##       (Intercept)  SCHEDULED_ARRIVAL           DISTANCE
##         -4.148202           0.006735          -0.001749
```

```
summary(retraso_modelo2)
```

```
##
## Call:
## lm(formula = ARRIVAL_DELAY ~ SCHEDULED_ARRIVAL + DISTANCE, data = vuelos_red
uc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -82.73  -17.72   -8.98    3.55 1381.16
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -4.1482023  0.2080543  -19.94   <2e-16 ***
## SCHEDULED_ARRIVAL  0.0067353  0.0001223   55.06   <2e-16 ***
## DISTANCE          -0.0017493  0.0001020  -17.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.21 on 399950 degrees of freedom
## Multiple R-squared:  0.008114,   Adjusted R-squared:  0.008109
## F-statistic:  1636 on 2 and 399950 DF,  p-value: < 2.2e-16
```

Seguimos sin tener un buen modelo lineal, no tenemos diferencia con el obtenido anteriormente.

```
retraso_modelo3 <- lm(ARRIVAL_DELAY~SCHEDULED_DEPARTURE + DISTANCE + SCHEDULED_
ARRIVAL, vuelos_reduc)
retraso_modelo3
```

```
##
## Call:
## lm(formula = ARRIVAL_DELAY ~ SCHEDULED_DEPARTURE + DISTANCE +
##     SCHEDULED_ARRIVAL, data = vuelos_reduc)
##
## Coefficients:
##         (Intercept)  SCHEDULED_DEPARTURE             DISTANCE
##           -6.221822             0.006112            -0.001614
##   SCHEDULED_ARRIVAL
##            0.002611
```

```
summary(retraso_modelo3)
```

```
##
## Call:
## lm(formula = ARRIVAL_DELAY ~ SCHEDULED_DEPARTURE + DISTANCE +
##     SCHEDULED_ARRIVAL, data = vuelos_reduc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -93.06  -17.73   -8.77    3.68 1381.74
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -6.2218222  0.2166590  -28.72   <2e-16 ***
## SCHEDULED_DEPARTURE  0.0061124  0.0001812   33.74   <2e-16 ***
## DISTANCE            -0.0016137  0.0001019  -15.83   <2e-16 ***
## SCHEDULED_ARRIVAL    0.0026106  0.0001728   15.11   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.15 on 399949 degrees of freedom
## Multiple R-squared:  0.01093,    Adjusted R-squared:  0.01092
## F-statistic:  1473 on 3 and 399949 DF,  p-value: < 2.2e-16
```

La bondad del ajuste para este modelo es casi 0, como en el resto de los casos. La proporción de variabilidad que queda explicada por este modelo es mínima, por lo que lo descartamos.

Modelos de regresión logística:
Utilizaremos el dataframe creada anteriormente para incluir las variables en el formato correcto para la regresión. ¿Está el retraso asociado a los vuelos de fin de semana y a la hora de salida?

```
retraso1 <- data.frame(retraso, DEPARTURE_HOUR=vuelos_reduc$DEPARTURE_HOUR)
retraso$WEEKEND <- ifelse(retraso1$RETRASO=="NO", 0, 1)
retraso1[1:2] <- lapply(retraso1[1:2], as.factor)
str(retraso1)
```

```
## 'data.frame':    399953 obs. of  3 variables:
##  $ RETRASO       : Factor w/ 2 levels "NO","SI": 2 1 1 1 1 1 2 1 1 2 ...
##  $ WEEKEND       : Factor w/ 2 levels "WEEKDAY","WEEKEND": 1 1 1 1 1 1 1 2 2
2 ...
##  $ DEPARTURE_HOUR: Factor w/ 25 levels " 0"," 1"," 2",..: 9 10 6 16 12 6 12
21 13 20 ...
```

```
retraso_glm1 <- glm(RETRASO~WEEKEND , data=retraso1, family=binomial)
retraso_glm1
```

```
##
## Call:  glm(formula = RETRASO ~ WEEKEND, family = binomial, data = retraso1)
##
## Coefficients:
##    (Intercept)  WEEKENDWEEKEND
##       -0.99290        -0.06326
##
## Degrees of Freedom: 399952 Total (i.e. Null);  399951 Residual
## Null Deviance:       462800
## Residual Deviance: 462700    AIC: 462700
```

```
summary(retraso_glm1)
```

```
##
## Call:
## glm(formula = RETRASO ~ WEEKEND, family = binomial, data = retraso1)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.7940  -0.7940  -0.7726   1.6175   1.6460
##
## Coefficients:
##                 Estimate Std. Error  z value Pr(>|z|)
## (Intercept)    -0.992903   0.004635 -214.229   <2e-16 ***
## WEEKENDWEEKEND -0.063259   0.007303   -8.662   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 462767  on 399952  degrees of freedom
## Residual deviance: 462692  on 399951  degrees of freedom
## AIC: 462696
##
## Number of Fisher Scoring iterations: 4
```

El modelo nos indica que el retraso en fin de semana es menor que en días laborables, tenemos un valor negativo de la varibale din de semana cuando esta toma el valor “SI” (=1)

```
retraso2 <- data.frame(retraso1, DISTANCE=vuelos_reduc$DISTANCE)
str(retraso2)
```

```
## 'data.frame':    399953 obs. of  4 variables:
##  $ RETRASO       : Factor w/ 2 levels "NO","SI": 2 1 1 1 1 1 2 1 1 2 ...
##  $ WEEKEND       : Factor w/ 2 levels "WEEKDAY","WEEKEND": 1 1 1 1 1 1 1 2 2
2 ...
##  $ DEPARTURE_HOUR: Factor w/ 25 levels " 0"," 1"," 2",..: 9 10 6 16 12 6 12
21 13 20 ...
##  $ DISTANCE      : int  674 1407 153 468 1363 184 439 216 268 680 ...
```

```
retraso_glm2 <- glm(RETRASO~DEPARTURE_HOUR + DISTANCE, data=retraso2, family=bi
nomial(logit))
retraso_glm2
```

```
##
## Call:  glm(formula = RETRASO ~ DEPARTURE_HOUR + DISTANCE, family = binomial
(logit),
##     data = retraso2)
##
## Coefficients:
##      (Intercept)  DEPARTURE_HOUR 1  DEPARTURE_HOUR 2  DEPARTURE_HOUR 3
##       -1.0978532         0.1651042         0.0686657        -0.1531218
## DEPARTURE_HOUR 4  DEPARTURE_HOUR 5  DEPARTURE_HOUR 6  DEPARTURE_HOUR 7
##        0.1387693         0.0930797         0.2361956         0.2814174
## DEPARTURE_HOUR 8  DEPARTURE_HOUR 9  DEPARTURE_HOUR10  DEPARTURE_HOUR11
##        0.2304691         0.2449998         0.2181709         0.1934779
## DEPARTURE_HOUR12  DEPARTURE_HOUR13  DEPARTURE_HOUR14  DEPARTURE_HOUR15
##        0.1399409         0.1793045         0.1728571         0.1697311
## DEPARTURE_HOUR16  DEPARTURE_HOUR17  DEPARTURE_HOUR18  DEPARTURE_HOUR19
##        0.1985642         0.2454503         0.2411858         0.1860060
## DEPARTURE_HOUR20  DEPARTURE_HOUR21  DEPARTURE_HOUR22  DEPARTURE_HOUR23
##        0.1451163         0.0517050         0.1276795         0.0726738
## DEPARTURE_HOUR24          DISTANCE
##       -0.2043180        -0.0001412
##
## Degrees of Freedom: 399952 Total (i.e. Null);  399927 Residual
## Null Deviance:       462800
## Residual Deviance: 462000    AIC: 462000
```

```
summary(retraso_glm2)
```

```
##
## Call:
## glm(formula = RETRASO ~ DEPARTURE_HOUR + DISTANCE, family = binomial(logit),
##     data = retraso2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8522  -0.8013  -0.7738   1.5701   1.9124
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.098e+00  5.871e-02 -18.700  < 2e-16 ***
## DEPARTURE_HOUR 1  1.651e-01  1.064e-01   1.551 0.120899
## DEPARTURE_HOUR 2  6.867e-02  1.891e-01   0.363 0.716482
## DEPARTURE_HOUR 3 -1.531e-01  3.268e-01  -0.469 0.639342
## DEPARTURE_HOUR 4  1.388e-01  1.036e-01   1.339 0.180621
## DEPARTURE_HOUR 5  9.308e-02  6.149e-02   1.514 0.130122
## DEPARTURE_HOUR 6  2.362e-01  5.985e-02   3.946 7.94e-05 ***
## DEPARTURE_HOUR 7  2.814e-01  5.987e-02   4.700 2.60e-06 ***
## DEPARTURE_HOUR 8  2.305e-01  5.988e-02   3.849 0.000119 ***
## DEPARTURE_HOUR 9  2.450e-01  5.996e-02   4.086 4.39e-05 ***

## DEPARTURE_HOUR10  2.182e-01  5.989e-02   3.643 0.000270 ***
## DEPARTURE_HOUR11  1.935e-01  5.996e-02   3.227 0.001252 **
## DEPARTURE_HOUR12  1.399e-01  6.004e-02   2.331 0.019759 *
## DEPARTURE_HOUR13  1.793e-01  6.001e-02   2.988 0.002808 **
## DEPARTURE_HOUR14  1.729e-01  6.010e-02   2.876 0.004027 **
## DEPARTURE_HOUR15  1.697e-01  5.998e-02   2.830 0.004661 **
## DEPARTURE_HOUR16  1.986e-01  6.007e-02   3.306 0.000948 ***
## DEPARTURE_HOUR17  2.455e-01  5.986e-02   4.101 4.12e-05 ***
## DEPARTURE_HOUR18  2.412e-01  6.007e-02   4.015 5.94e-05 ***
## DEPARTURE_HOUR19  1.860e-01  6.009e-02   3.096 0.001964 **
## DEPARTURE_HOUR20  1.451e-01  6.056e-02   2.396 0.016571 *
## DEPARTURE_HOUR21  5.171e-02  6.150e-02   0.841 0.400467
## DEPARTURE_HOUR22  1.277e-01  6.294e-02   2.029 0.042507 *
## DEPARTURE_HOUR23  7.267e-02  6.941e-02   1.047 0.295068
## DEPARTURE_HOUR24 -2.043e-01  4.992e-01  -0.409 0.682329
## DISTANCE         -1.412e-04  6.141e-06 -22.995  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 462767  on 399952  degrees of freedom
## Residual deviance: 461988  on 399927  degrees of freedom
## AIC: 462040
##
## Number of Fisher Scoring iterations: 4
```

El modelo no es muy bueno, hay algunas variables que no aportan mucho al modelo, seguramente porque no tienen datos suficientes. Además, algunas variables no tienen relación entre si, como ya se había visto en el gráfico de correlación (hora de salida, retraso en la llegada). Vamos a intentar crear

el modelo teniendo como variables explicativas la distancia y la hora de salida del vuelo por ver si puede haber una posible relación

```
retraso3 <- data.frame(retraso2, MONTH=vuelos_reduc$MONTH,
                       SCHEDULED_DEPARTURE=vuelos_reduc$SCHEDULED_DEPARTURE)
str(retraso3)
```

```
## 'data.frame':    399953 obs. of  6 variables:
##  $ RETRASO             : Factor w/ 2 levels "NO","SI": 2 1 1 1 1 1 2 1 1 2
...
##  $ WEEKEND            : Factor w/ 2 levels "WEEKDAY","WEEKEND": 1 1 1 1 1 1
1 2 2 2 ...
##  $ DEPARTURE_HOUR     : Factor w/ 25 levels " 0"," 1"," 2",..: 9 10 6 16 12
6 12 21 13 20 ...
##  $ DISTANCE           : int  674 1407 153 468 1363 184 439 216 268 680 ...
##  $ MONTH              : int  11 4 4 7 2 7 12 2 3 2 ...
##  $ SCHEDULED_DEPARTURE: int  825 930 540 1545 1055 600 1125 1956 1145 1935
...
```

```
retraso_glm3 <- glm(RETRASO~SCHEDULED_DEPARTURE + DISTANCE + MONTH, data=retras
o3, family=binomial)
retraso_glm3
```

```
##
## Call:  glm(formula = RETRASO ~ SCHEDULED_DEPARTURE + DISTANCE + MONTH,
##     family = binomial, data = retraso3)
##
## Coefficients:
##         (Intercept)  SCHEDULED_DEPARTURE             DISTANCE
##           -0.6250959           -0.0000559           -0.0001397
##               MONTH
##           -0.0318811
##
## Degrees of Freedom: 399952 Total (i.e. Null);  399949 Residual
## Null Deviance:       462800
## Residual Deviance: 461200    AIC: 461300
```

```
summary(retraso_glm3)
```

```
##
## Call:
## glm(formula = RETRASO ~ SCHEDULED_DEPARTURE + DISTANCE + MONTH,
##     family = binomial, data = retraso3)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9066  -0.8047  -0.7597   1.5303   1.9547
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -6.251e-01  1.339e-02 -46.673  < 2e-16 ***
## SCHEDULED_DEPARTURE -5.590e-05  7.439e-06  -7.514 5.75e-14 ***
## DISTANCE            -1.397e-04  6.090e-06 -22.943  < 2e-16 ***
## MONTH               -3.188e-02  1.057e-03 -30.159  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 462767  on 399952  degrees of freedom
## Residual deviance: 461250  on 399949  degrees of freedom
## AIC: 461258
##
## Number of Fisher Scoring iterations: 4
```

Vamos a realizar una predicción de retraso con el modelo que hemos creado, para el caso de un vuelo que salga a las 8horas y que tenga una distancia de 1000 milas en el mes de junio y otro que salga a las 19 horas

```
newdata = data.frame(SCHEDULED_DEPARTURE = 0800 , DISTANCE=1000, MONTH=6)
predict(retraso_glm3, newdata , type="response")
```

```
##         1
## 0.2687817
```

```
newdata2 = data.frame(SCHEDULED_DEPARTURE =1600, DISTANCE=1000, MONTH=6)
predict(retraso_glm3, newdata2 , type="response")
```

```
##         1
## 0.2600845
```

La probabilidad de retraso en el primer caso será de 0.27 veces superior que no sufrir retraso. En el caso de que el vuelo tenga su salida a las 16 horas será de 0.26 veces más.

Incluimos ahora en el modelo el atributo de Aeropuerto de salida.

```
retraso4 <- data.frame(retraso3, ORIGIN_AIRPORT=vuelos_reduc$ORIGIN_CODE)
```

PAra este modelo, nos Vamos a centrar únicamente en los 10 aeropuertos con más tráfico.

```
# Para que los resultados estén acotados, elegimos los 10 aeropuertos con más t
ráfico:
retraso$ORIGIN_AIRPORT <- ifelse (retraso4$ORIGIN_AIRPORT %in%
                             c("ATL","ORD","DFW","DEN","LAX","SFO","PHX","IAH",
"LAS","MSP"),
                            retraso4$ORIGIN_AIRPORT,
                            "OTROS")
str(retraso4)
```

```
## 'data.frame':    399953 obs. of  7 variables:
##  $ RETRASO             : Factor w/ 2 levels "NO","SI": 2 1 1 1 1 1 2 1 1 2
...
##  $ WEEKEND             : Factor w/ 2 levels "WEEKDAY","WEEKEND": 1 1 1 1 1 1
1 2 2 2 ...
##  $ DEPARTURE_HOUR      : Factor w/ 25 levels " 0"," 1"," 2",..: 9 10 6 16 12
6 12 21 13 20 ...
##  $ DISTANCE            : int  674 1407 153 468 1363 184 439 216 268 680 ...
##  $ MONTH               : int  11 4 4 7 2 7 12 2 3 2 ...
##  $ SCHEDULED_DEPARTURE: int  825 930 540 1545 1055 600 1125 1956 1145 1935
...
##  $ ORIGIN_AIRPORT      : Factor w/ 628 levels "10135","10136",..: 327 358 439
504 483 346 344 593 535 523 ...
```

Dejamos comentado el código, porque no nos ha sido posible realizar este cálculo con nuestros equipos.

```
#retraso_glm4 <- glm(RETRASO~DEPARTURE_HOUR + DISTANCE + MONTH + ORIGIN_AIRPOR
T, data=retraso4,
#                    family=binomial)
#retraso_glm4
#summary(retraso_glm4)
```

Vamos a realizar un contraste de hipótesis, nos preguntamos si la proporción de vuelos retrasados es inferior a la de vuelos en los tiempos establecidos.Para ello realizaremos un contraste sobre la proporción para muestras grandes.
Consideramos que tenemos una población que toma el valor 1 cuando el vuelo está retrasado y el 0 cuando no lo está. Tenemos por tanto una distribución de Bernoulli con parámetro *p* desconocido. p_0 será un valor prefijado, en nuestro caso 0.5, es decir el 50% que marcaría la igualdad de proporción de vuelos retrasados y no retrasados. Nuestra hipótesis alternativa será que la proporción de vuelos retrasados es inferior al 50%.

$$\begin{cases} Hipótesis\ Nula & H_0: \quad p = p_0 \\ Hipótesis\ Alternativa & H_1: \quad p < p_0 \end{cases}$$

Separamos las muestras de los vuelos retrasados RETRASO_TOTAL>0 y los no retrasados RETRASO_TOTAL<=0

```
head(vuelos_reduc)
```

```
##   MONTH DAY DAY_OF_WEEK AIRLINE FLIGHT_NUMBER ORIGIN_CODE DESTINATION_AIRPOR
T
## 1    11  16           1      EV          5084         ATL                 TU
L
## 2     4   7           2      WN          1023         BWI                 SA
T
## 3     4   1           3      DL          2182         GSP                 AT
L
## 4     7  28           2      EV          4330         MEM                 IA
H
## 5     2   9           1      WN          1963         LAX                 MC
I
## 6     7  23           4      AA          2148         BOS                 LG
A
##   SCHEDULED_DEPARTURE DEPARTURE_TIME DEPARTURE_DELAY SCHEDULED_TIME
## 1                 825            819              -6            128
## 2                 930            943              13            240
## 3                 540            538              -2             57
## 4                1545           1559              14            108
## 5                1055           1105              10            190
## 6                 600            553              -7             74
##   ELAPSED_TIME AIR_TIME DISTANCE SCHEDULED_ARRIVAL ARRIVAL_TIME ARRIVAL_DELA
Y
## 1          129      111      674               933          928            -
5
## 2          214      201     1407              1230         1217           -1
3
## 3           47       28      153               637          625           -1
2
## 4          105       75      468              1733         1744            1
1
## 5          166      153     1363              1605         1551           -1
4
## 6           68       47      184               714          701           -1
3
##   SCHEDULED_DEPARTURE_HOUR DEPARTURE_HOUR ARRIVAL_HOUR RETRASO_TOTAL
## 1                        8              8            9             1
## 2                        9              9           12           -26
## 3                        5              5            6           -10
## 4                       15             15           17            -3
## 5                       10             11           15           -24
## 6                        6              5            7            -6
```

```
vuelos_reduc$DELAYED <- ifelse(vuelos_reduc$RETRASO_TOTAL>0, 1, 0)

vuelos_reduc %>%
  group_by(DELAYED) %>%
  summarize(num_obs = n(),
            mean_delayed = round(mean(RETRASO_TOTAL),0),
            sd_delayed = round(sd(RETRASO_TOTAL),0))
```

```
## # A tibble: 2 x 4
##   DELAYED num_obs mean_delayed sd_delayed
##     <dbl>   <int>        <dbl>      <dbl>
## 1       0  293846          -10          8
## 2       1  106107           10         12
```

```
DELAYED_GROUPED <- vuelos_reduc %>% group_by(DELAYED) %>% summarize(num_obs = n
(),
                                                                    obs_totales
=length(vuelos_reduc$DAY),
                                                                     p = num_ob
s/obs_totales)
DELAYED_GROUPED %>% knitr::kable("html")  %>% kable_styling(position='center',
 font_size=12, fixed_thead=list(enabled=T))
```

DELAYED	num_obs	obs_totales	p
0	293846	399953	0.7347013

```
n1=DELAYED_GROUPED[1,2]
n2=DELAYED_GROUPED[2,2]

prop.test(x=106107, n=399953, alternative="less", conf.level=0.95, p=0.5)
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  106107 out of 399953, null probability 0.5
## X-squared = 88124, df = 1, p-value < 2.2e-16
## alternative hypothesis: true p is less than 0.5
## 95 percent confidence interval:
##  0.0000000 0.2664498
## sample estimates:
##         p
## 0.2652987
```

Con estos datos se obtiene un p-valor muy cercano a 0, es decir menor que el nivel de confianza establecido del 0.05, por lo que podemos concluir que los vuelos retrasados son menos probables que los vuelos en los tiempos establecidos, al contrario de lo que podríamos esperar.

Predecir si un vuelo va a retrasarse en función del mes, día, hora y aeropuerto. Vamos a aplicar un modelo Ramdon Forest para intentar dar respuesta a esta pregunta.

```
str(retraso4)
```

```
## 'data.frame':    399953 obs. of  7 variables:
##  $ RETRASO             : Factor w/ 2 levels "NO","SI": 2 1 1 1 1 1 2 1 1 2
...
##  $ WEEKEND             : Factor w/ 2 levels "WEEKDAY","WEEKEND": 1 1 1 1 1 1
1 2 2 2 ...
##  $ DEPARTURE_HOUR      : Factor w/ 25 levels " 0"," 1"," 2",..: 9 10 6 16 12
6 12 21 13 20 ...
##  $ DISTANCE            : int  674 1407 153 468 1363 184 439 216 268 680 ...
##  $ MONTH               : int  11 4 4 7 2 7 12 2 3 2 ...
##  $ SCHEDULED_DEPARTURE: int  825 930 540 1545 1055 600 1125 1956 1145 1935
...
##  $ ORIGIN_AIRPORT      : Factor w/ 628 levels "10135","10136",..: 327 358 439
504 483 346 344 593 535 523 ...
```

```
#Seleccionamos solo las variables que vamos a utilizar para aplicar el modelo.
 Dejaremos fuera SCHEDULED_DEPARTURE.
retraso_rf <- select (retraso4, -"SCHEDULED_DEPARTURE")
```

Para poder crear el modelo, factorizamos la variable distancia, creando rangos de 1000 Millas.

```
RANGO_DISTANCIA <- vector()
RANGO_DISTANCIA[retraso_rf$DISTANCE<=500] <- 1
RANGO_DISTANCIA[retraso_rf$DISTANCE>500 & retraso_rf$DISTANCE<=1000] <- 2
RANGO_DISTANCIA[retraso_rf$DISTANCE>1000 & retraso_rf$DISTANCE<=1500] <- 3
RANGO_DISTANCIA[retraso_rf$DISTANCE>1500 & retraso_rf$DISTANCE<=2000] <- 4
RANGO_DISTANCIA[retraso_rf$DISTANCE>2000 & retraso_rf$DISTANCE<=2500] <- 5
RANGO_DISTANCIA[retraso_rf$DISTANCE>2500 & retraso_rf$DISTANCE<=3000] <- 6
RANGO_DISTANCIA[retraso_rf$DISTANCE>3000] <- 7

retraso_rf$RANGO_DISTANCIA <- as.factor(RANGO_DISTANCIA)

levels(retraso_rf$RANGO_DISTANCIA) <- c("Menor 500","Entre 500 y 1000","Entre 1
000 y 1500",
                                        "Entre 1500 y 2000","Entre 2000 y 2500"
,"Entre 2500 y 3000",
                                        "Mayor 3000")
table(retraso_rf$RANGO_DISTANCIA)
```

```
##
##         Menor 500  Entre 500 y 1000 Entre 1000 y 1500 Entre 1500 y 2000
##            146537            139715             59401             28342
## Entre 2000 y 2500 Entre 2500 y 3000        Mayor 3000
##             17668              7647               643
```

```
retraso_rf <- select (retraso_rf, -"DISTANCE")

retraso_rf$MONTH <- as.factor(retraso_rf$MONTH)
```

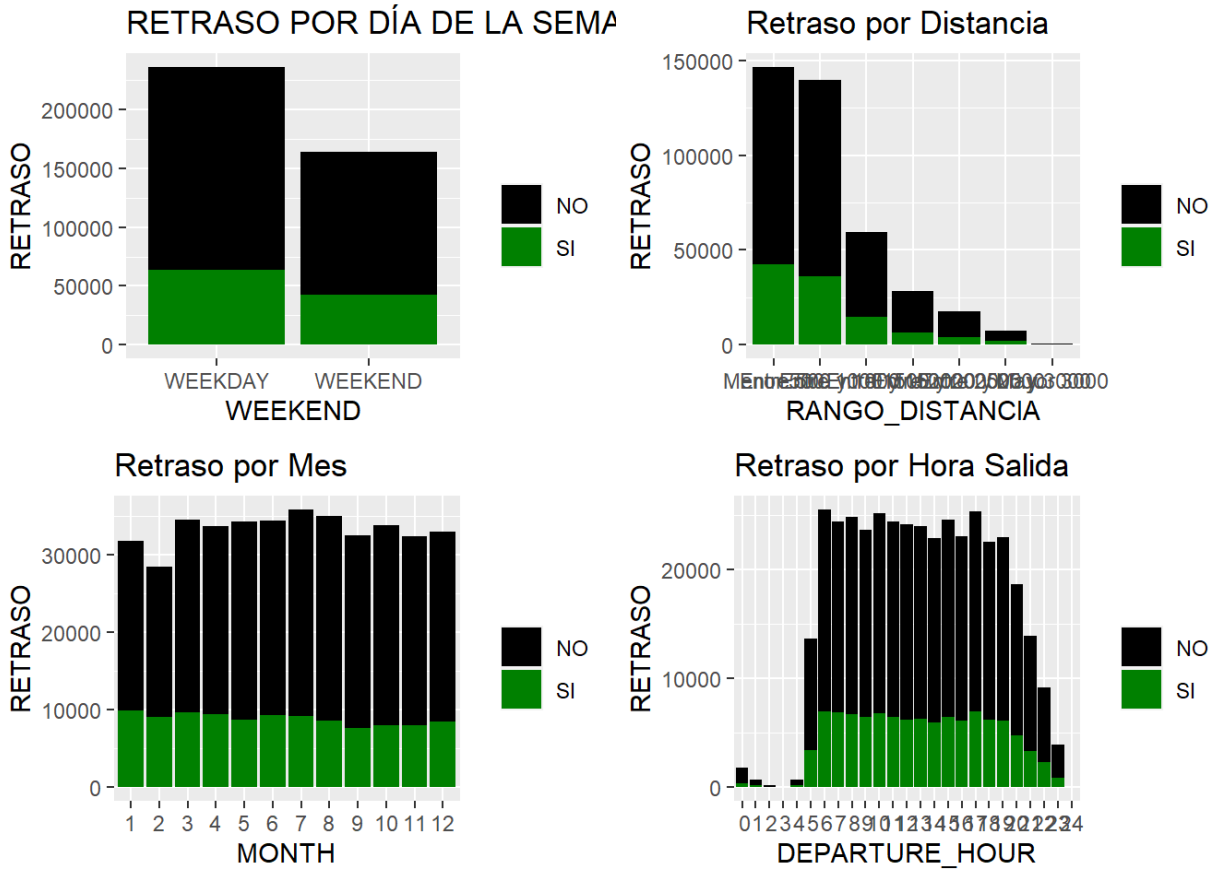
Factorizamos también la variable aeropuerto origen para incluirla a posteriori en el modelo.

```
retraso_rf$ORIGIN_AIRPORT <- ifelse (retraso_rf$ORIGIN_AIRPORT %in%
                              c("ATL","ORD","DFW","DEN","LAX","SFO","PHX","IAH",
"LAS","MSP"),
                             retraso_rf$ORIGIN_AIRPORT,
                             "OTROS")
retraso_rf$ORIGIN_AIRPORT <- as.factor(retraso_rf$ORIGIN_AIRPORT)
head(retraso_rf)
```

```
##   RETRASO WEEKEND DEPARTURE_HOUR MONTH ORIGIN_AIRPORT   RANGO_DISTANCIA
## 1      SI WEEKDAY              8    11            327  Entre 500 y 1000
## 2      NO WEEKDAY              9     4          OTROS Entre 1000 y 1500
## 3      NO WEEKDAY              5     4          OTROS         Menor 500
## 4      NO WEEKDAY             15     7          OTROS         Menor 500
## 5      NO WEEKDAY             11     2            483 Entre 1000 y 1500
## 6      NO WEEKDAY              5     7          OTROS         Menor 500
```

Contrastamos algunos de los atributos con la variable retraso:

```
grid.newpage()
plotbyweekend<-ggplot(retraso_rf,aes(WEEKEND,fill=RETRASO))+geom_bar() +labs(x=
"WEEKEND", y="RETRASO")+ guides(fill=guide_legend(title=""))+ scale_fill_manual
(values=c("black","#008000"))+ggtitle("RETRASO POR DÍA DE LA SEMANA")
plotbydistance<-ggplot(retraso_rf,aes(RANGO_DISTANCIA,fill=RETRASO))+geom_bar()
+labs(x="RANGO_DISTANCIA", y="RETRASO")+ guides(fill=guide_legend(title=""))+ s
cale_fill_manual(values=c("black","#008000"))+ggtitle("Retraso por Distancia")
plotbyMeses<-ggplot(retraso_rf,aes(MONTH,fill=RETRASO))+geom_bar() +labs(x="MON
TH", y="RETRASO")+ guides(fill=guide_legend(title=""))+ scale_fill_manual(value
s=c("black","#008000"))+ggtitle("Retraso por Mes")
plotbySalida<-ggplot(retraso_rf,aes(DEPARTURE_HOUR,fill=RETRASO))+geom_bar() +l
abs(x="DEPARTURE_HOUR", y="RETRASO")+ guides(fill=guide_legend(title=""))+ scal
e_fill_manual(values=c("black","#008000"))+ggtitle("Retraso por Hora Salida")

grid.arrange(plotbyweekend,plotbydistance,plotbyMeses,plotbySalida,ncol=2)
```



Dividimos el conjunto de datos en 2 conjuntos, uno que utilizaremos de entrenamiento y otro para test (60/40)
Como variable de clasificación será “RETRASO” y la incluiremos en el valor de las “y”, el resto de atributos se incluirán en el valor de “x”
Comprobamos si tenemos que desordenar el data set para seleccionar el conjunto de entrenamiento.

```
head(retraso_rf,10)
```

```
##    RETRASO WEEKEND DEPARTURE_HOUR MONTH ORIGIN_AIRPORT   RANGO_DISTANCIA
## 1       SI WEEKDAY              8    11            327  Entre 500 y 1000
## 2       NO WEEKDAY              9     4          OTROS Entre 1000 y 1500
## 3       NO WEEKDAY              5     4          OTROS          Menor 500
## 4       NO WEEKDAY             15     7          OTROS          Menor 500
## 5       NO WEEKDAY             11     2            483 Entre 1000 y 1500
## 6       NO WEEKDAY              5     7          OTROS          Menor 500
## 7       SI WEEKDAY             11    12          OTROS          Menor 500
## 8       NO WEEKEND             20     2          OTROS          Menor 500
## 9       NO WEEKEND             12     3            535          Menor 500
## 10      SI WEEKEND             19     2            523  Entre 500 y 1000
```

No se encuentra ordenado por la variable de clasificación, así que no vamos a desordenar el conjunto.

```
set.seed(300)
indexes = sample(1:nrow(retraso_rf), size=floor((0.6)*nrow(retraso_rf)))
train <- retraso_rf[indexes,]
test <- retraso_rf[-indexes,]

str(train)
```

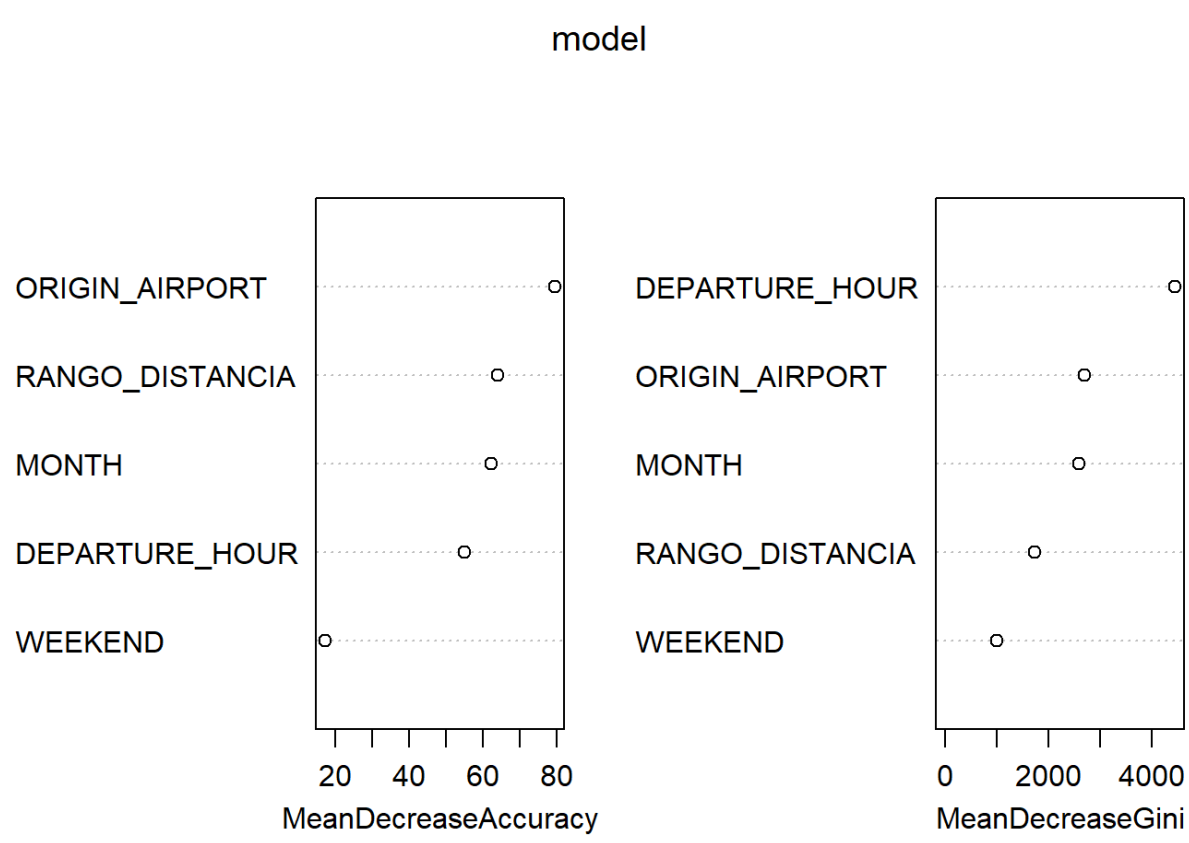
```
## 'data.frame':    239971 obs. of  6 variables:
##  $ RETRASO        : Factor w/ 2 levels "NO","SI": 1 1 1 1 1 1 1 1 1 1 ...
##  $ WEEKEND        : Factor w/ 2 levels "WEEKDAY","WEEKEND": 1 2 1 1 1 2 2 2
1 2 ...
##  $ DEPARTURE_HOUR : Factor w/ 25 levels " 0"," 1"," 2",..: 11 16 21 8 10 9 6
6 6 20 ...
##  $ MONTH          : Factor w/ 12 levels "1","2","3","4",..: 12 2 1 3 9 1 10
6 3 10 ...
##  $ ORIGIN_AIRPORT : Factor w/ 11 levels "327","392","393",..: 11 11 3 10 11
11 11 11 11 11 ...
##  $ RANGO_DISTANCIA: Factor w/ 7 levels "Menor 500","Entre 500 y 1000",..: 1
2 1 4 2 2 2 2 1 ...
```

Creamos el modelo de árbol de decisión

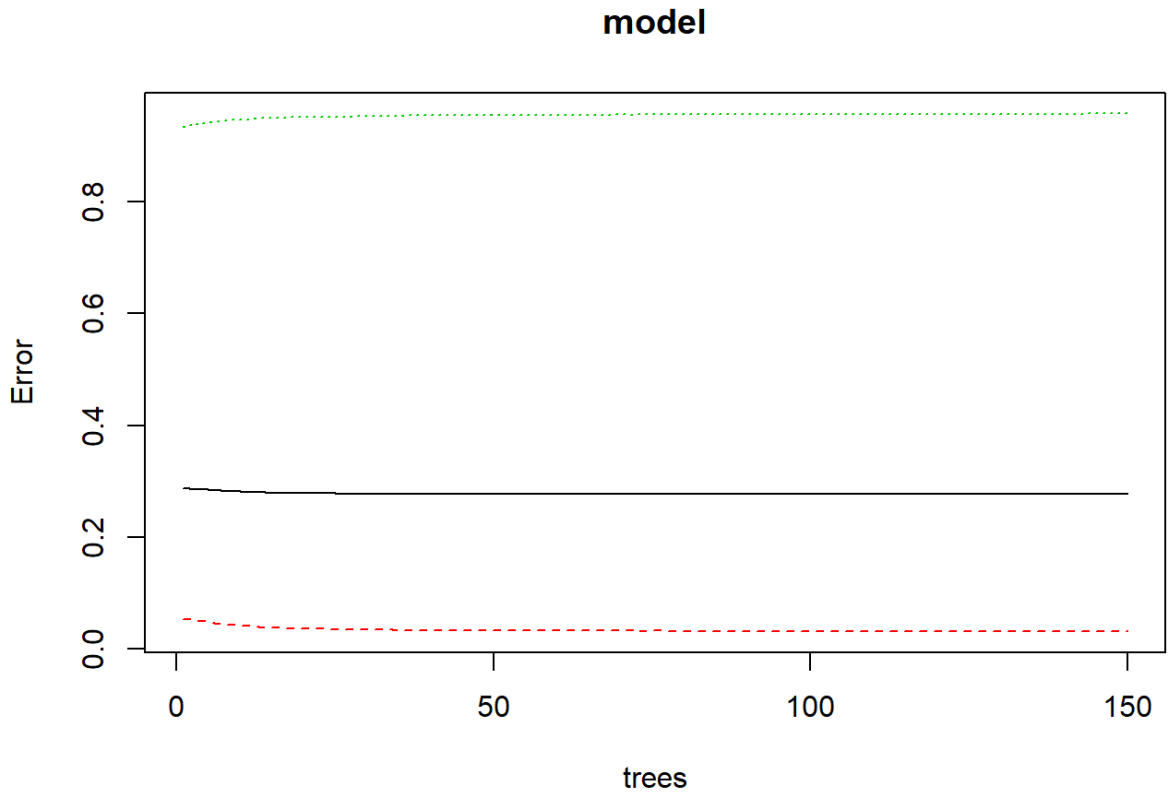
```
#Creamos el modelo
model <- randomForest(RETRASO ~.,data=train, importance=T, ntree=150, mtry=4)
```

Comprobamos la importancia de las variables en el modelo.

```
varImpPlot(model)
```



```
plot(model)
```



Vemos a continuación como funciona este modelo, generando hasta 300 posibles árboles de decisión con las disitntas variables.

```
print(model)
```

```
##
## Call:
##  randomForest(formula = RETRASO ~ ., data = train, importance = T,      ntre
e = 150, mtry = 4)
##                Type of random forest: classification
##                      Number of trees: 150
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 27.63%
## Confusion matrix:
##        NO   SI class.error
## NO 171031 5403  0.03062335
## SI  60903 2634  0.95854384
```

Este modelo nos da una tasa de error del 27.8%, esta es la proporción de observaciones que no han sido bien clasificadas por el modelo. Comprobamos que la clasificación de vuelos retrasados no es buena. Comprobamos la matriz de confusión.

```
mat_confusion<-table(train$RETRASO,model$predicted)
mat_confusion
```

```
##
##          NO     SI
##   NO 171031   5403
##   SI  60903   2634
```

Vamos a ver el porcentaje de registros clasificados correctamente.

```
porcentaje_correct<-100 * sum(diag(mat_confusion)) / sum(mat_confusion)
print(sprintf("El %% de registros correctamente clasificados es: %.4f %%",porce
ntaje_correct))
```

```
## [1] "El % de registros correctamente clasificados es: 72.3692 %"
```

Validamos el modelo con los datos de test.

```
resultado<- predict( model, test, type="class" )

confusionMatrix(model$predicted, train$RETRASO)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     NO     SI
##         NO 171031  60903
##         SI   5403   2634
##
##                Accuracy : 0.7237
##                  95% CI : (0.7219, 0.7255)
##     No Information Rate : 0.7352
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.015
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.96938
##             Specificity : 0.04146
##          Pos Pred Value : 0.73741
##          Neg Pred Value : 0.32773
##              Prevalence : 0.73523
##          Detection Rate : 0.71272
##    Detection Prevalence : 0.96651
##       Balanced Accuracy : 0.50542
##
##        'Positive' Class : NO
##
```

Realizamos la predicción sobre el resultado de test.

```
confusionMatrix(resultado, test$RETRASO)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     NO     SI
##         NO 113920  40866
##         SI   3492   1704
##
##                Accuracy : 0.7227
##                  95% CI : (0.7205, 0.7249)
##     No Information Rate : 0.7339
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.0143
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.97026
##             Specificity : 0.04003
##          Pos Pred Value : 0.73598
##          Neg Pred Value : 0.32794
##              Prevalence : 0.73391
##          Detection Rate : 0.71208
##    Detection Prevalence : 0.96752
##       Balanced Accuracy : 0.50514
##
##        'Positive' Class : NO
##
```

Obtenemos un grado de precisión muy similar sobre el conjunto de entrenamiento y sobre el conjunto de test. No es un modelo demasiado bueno, en ambos casos la precisión de predicción del modelo se encuentra alrededor del 70%.

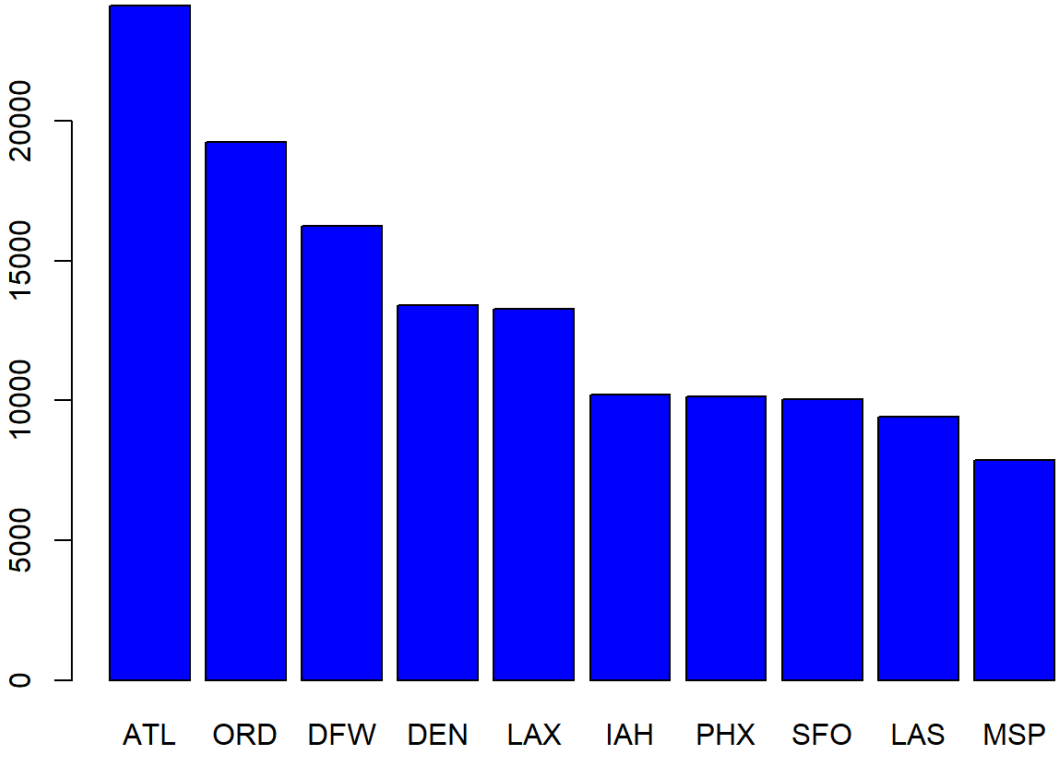
---

5. Representación de los resultados

Representación de los resultados a partir de tablas y gráficas.

En este apartado intentaremos dar respuesta a algunas de nuestras dudas mediante la representación gráfica de los datos elegidos.
Comenzamos las representaciones viendo qué aeropuertos son los que mayor volumen de vuelos acogen:

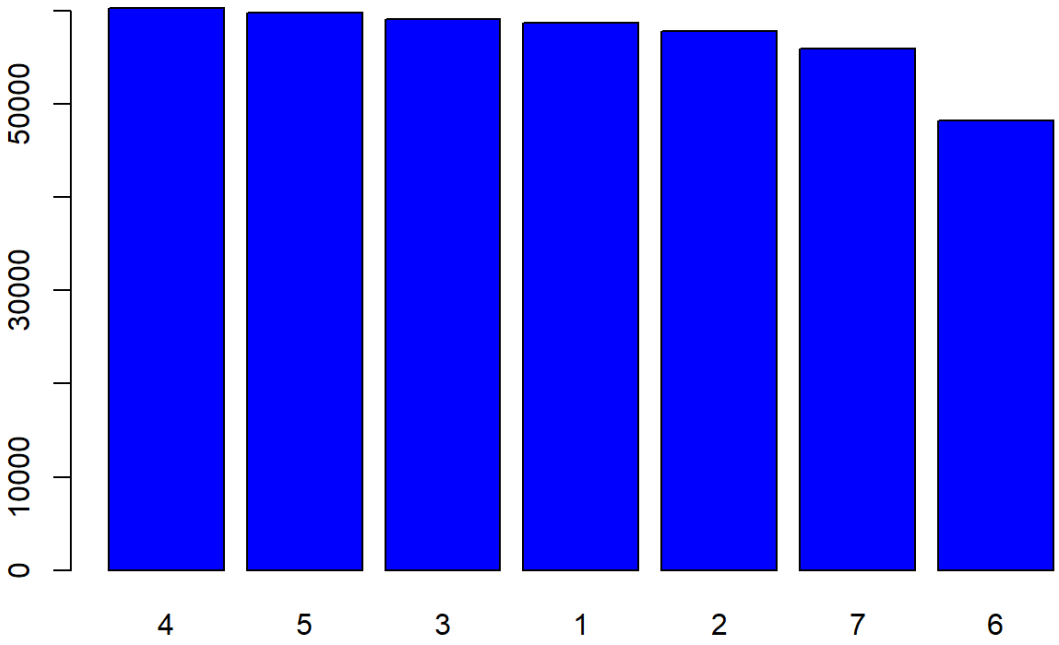
```
#Comprobamos en un gráfico de barras los aeropuertos más populares.
popular_airports <- sort(table(vuelos_reduc$ORIGIN_CODE), decreasing = TRUE )
barplot(popular_airports[1:10], col = "blue", ylim = c(0,20000))
```



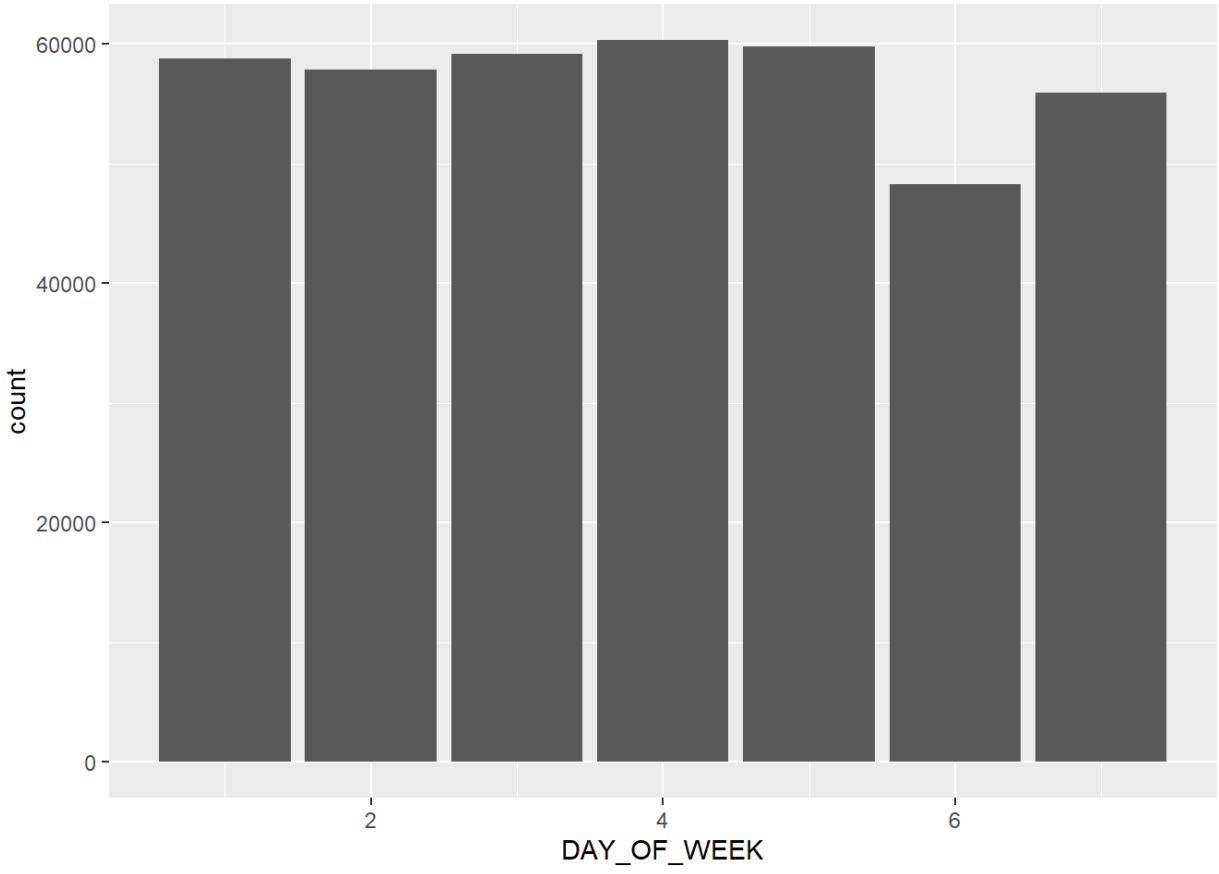
En la gráfica de barras podemos observar que fue el aeropuerto de Atlanta el más popular de Estados Unidos en el año 2015. En realidad, era de esperar este resultado, dado que el aeropuerto de Atlanta es el aeropuerto con más tráfico aéreo de Estados Unidos y del mundo.

Una vez que tenemos el aeropuerto, ahora vamos a ver qué dia de la semana es en el que se producen más vuelos.

```
#visualización del volumen de vuelos de cada día de la semana

dias_semana <- sort(table(vuelos_reduc$DAY_OF_WEEK), decreasing=TRUE)


barplot(dias_semana, col = "blue", ylim = c(0,60000))
```



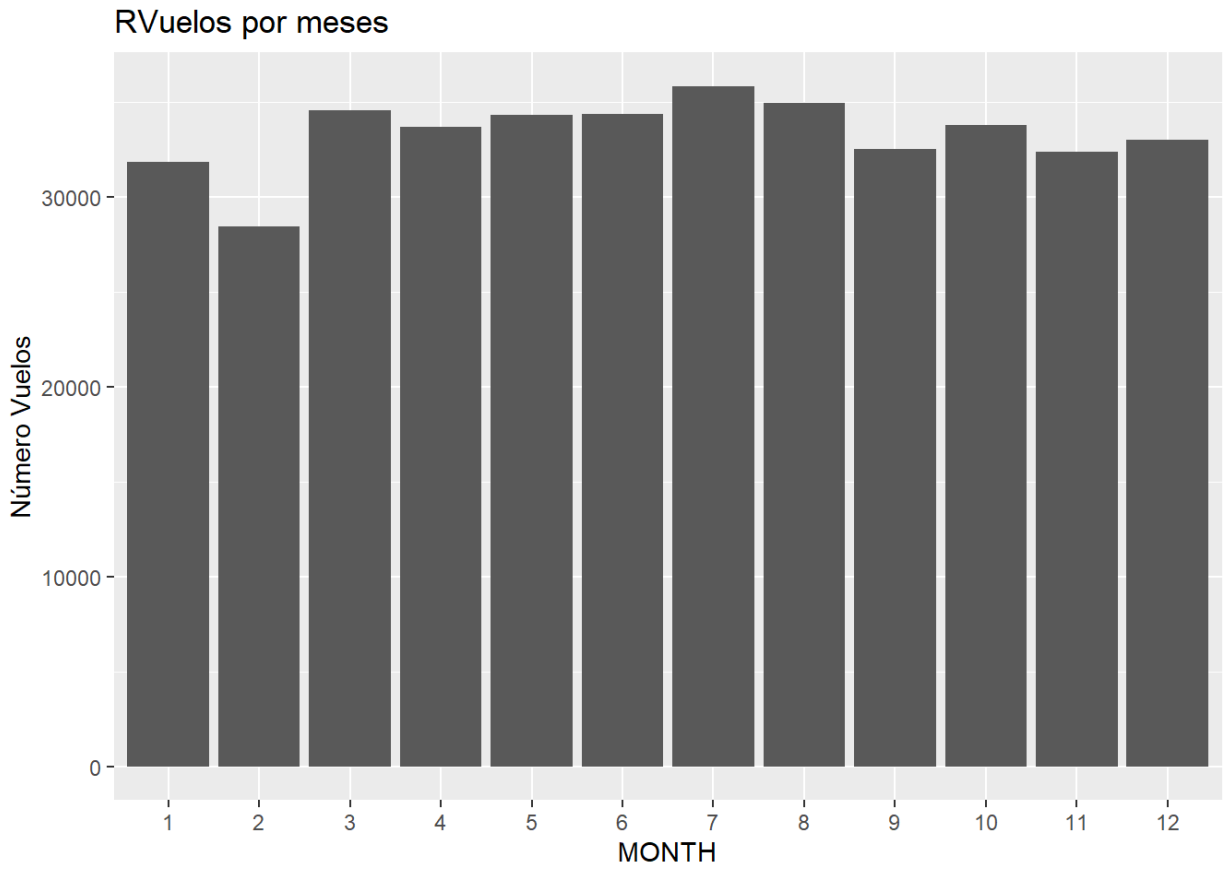

```
ggplot(vuelos_reduc, aes(x=DAY_OF_WEEK, fill=DAY_OF_WEEK )) +
  geom_bar( ) +
  scale_fill_hue(c = 40)
```



Se comprueba que el tráfico diario en los días laborables es muy similar, viendose un cambio de tendencia en el fin de semana, en concreto en el sábado.

En el siguiente gráfico, veremos la relación del número de vuelos frente al mes del año:

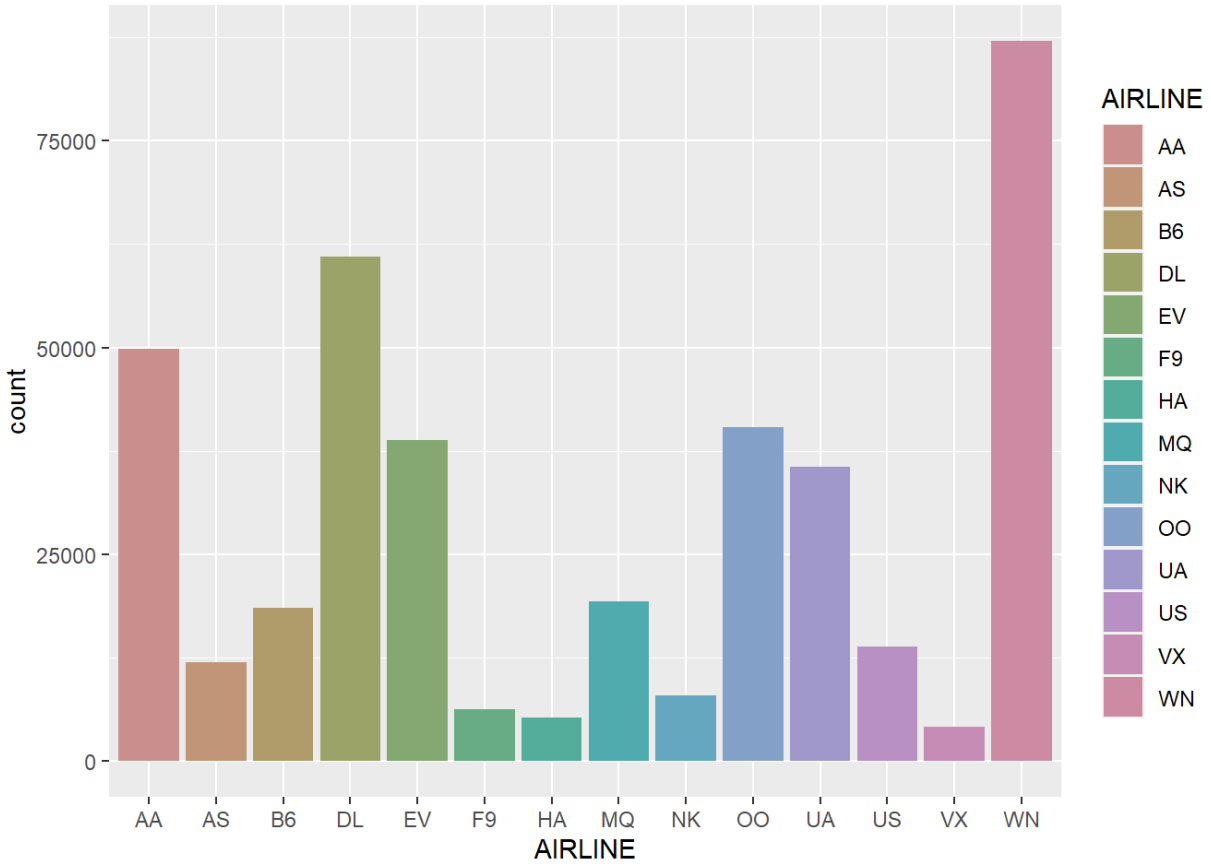
```
ggplot(vuelos_reduc,aes(factor(MONTH),fill=RETRASO_TOTAL))+geom_bar()  +labs(x="MONTH", y="Número Vuelos")+ guides(fill=guide_legend(title=""))+ scale_fill_manual(values=c("black","#008000"))+ggtitle("RVuelos por meses")
```



De este gráfico comprobamos que febrero es el mes que menos vuelos se producen (uno de los factores que pueden influir en este hecho es que es el mes del año más corto).
Por otro lado vemos, que los meses de verano son los que mayor afluencia de vuelos hay, siendo julio el que alberga el mayor número de vuelos.

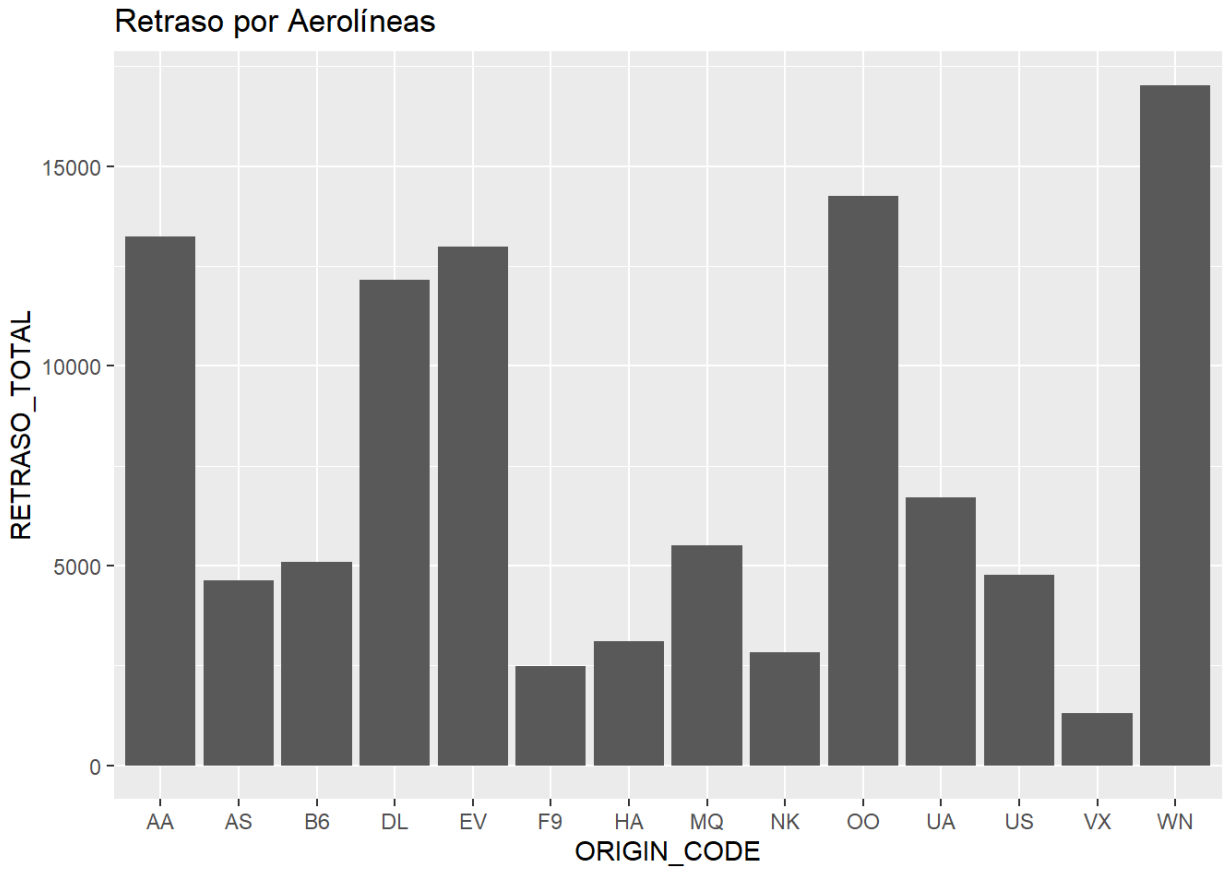
Otra conclusión que podemos sacar, es que quitando los meses de verano, en el resto del año se aprecia la tencencia en el que los meses con 30 días tienen menor volumen de vuelos que los de 31 días.

```
ggplot(vuelos_reduc, aes(x=AIRLINE, fill=AIRLINE )) +
  geom_bar( ) +
  scale_fill_hue(c = 40)
```



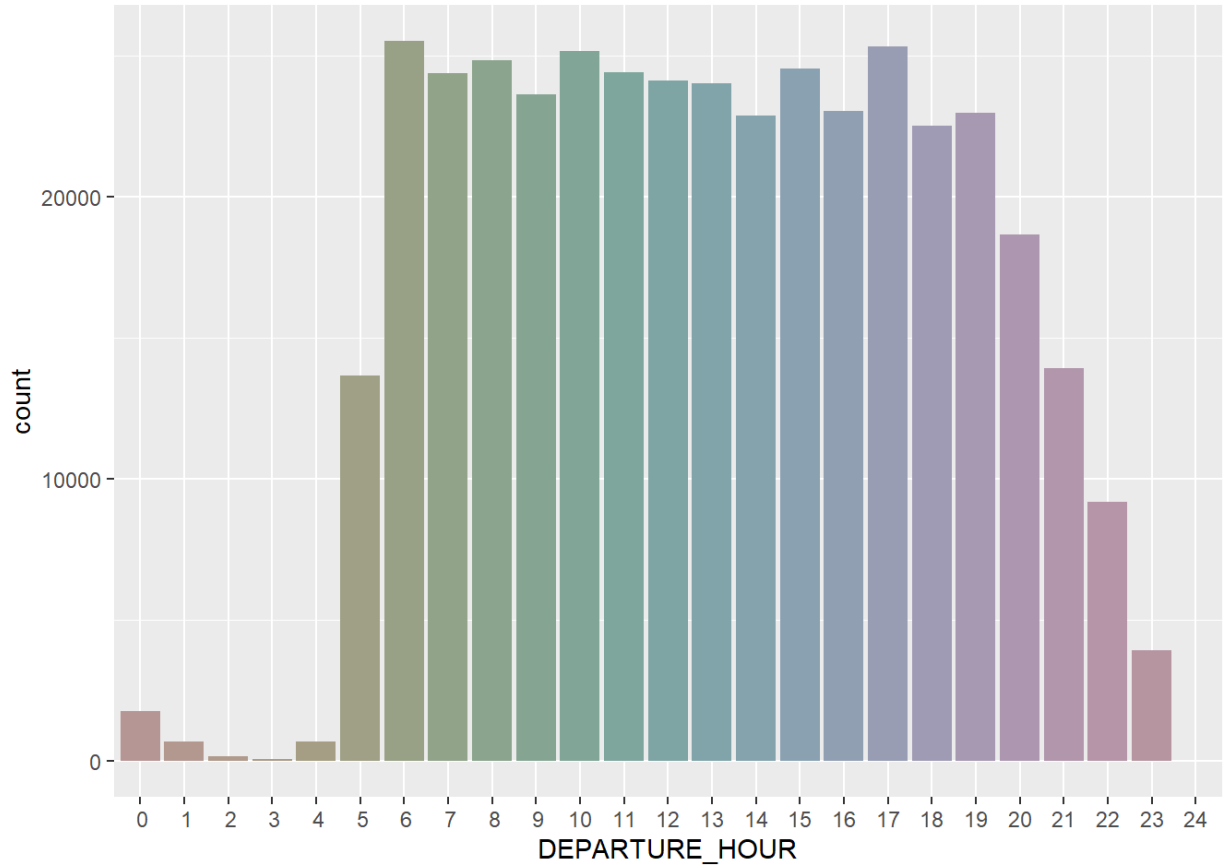
En el grafico que muestran el número de vuelos de cada aerolínea podemos ver como una de ellas prevalece sobre todas las demás.
La aerolínea WN, cuyas siglas corresponden a Southwest Airlines Co. operó más de 60000 vuelos en el año 2015 en Estados Unidos, mientras que la segunda que le sigue en el ranking operó 45000 vuelos. Vemos que la diferencia es muy significativa. Southwest Airlines es la mayor compañía aeronáutica de Estados Unidos y el mayor operador de bajo coste del mundo.

```
vuelos_reduc %>%
  filter( RETRASO_TOTAL>0) %>%
   ggplot(aes(AIRLINE,fill=RETRASO_TOTAL))+geom_bar() +labs(x="ORIGIN_CODE", y=
"RETRASO_TOTAL")+ guides(fill=guide_legend(title=""))+ scale_fill_manual(values
=c("black","#008000"))+ggtitle("Retraso por Aerolíneas")
```



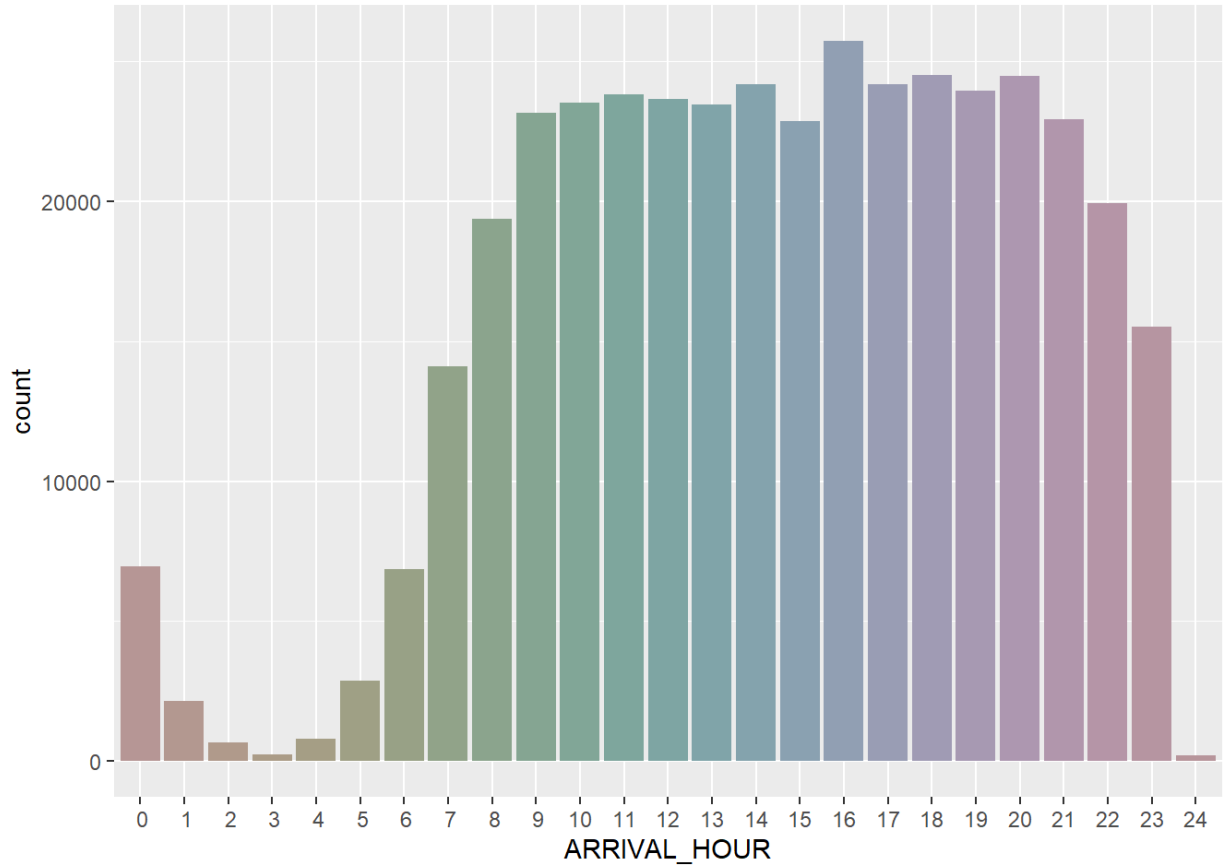
Si comprobamos el retraso por aerolínea, nos aparece Southwest como la aerolínea con más vuelos retrasados con bastante diferencia.

```
#Horas de salida de los vuelos
ggplot(vuelos_reduc, aes(x=DEPARTURE_HOUR, fill=DEPARTURE_HOUR )) +
  geom_bar( ) +
  scale_fill_hue(c = 20) +
  theme(legend.position="none")
```



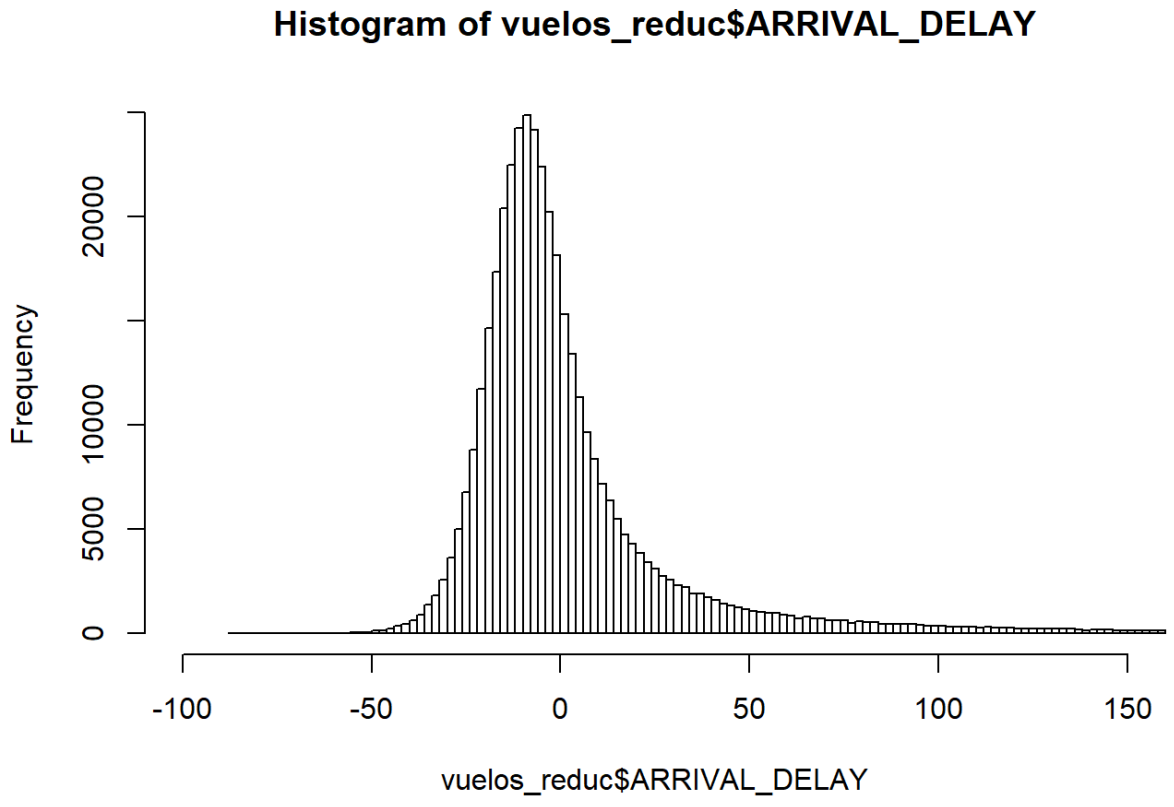
También queremos comprobar las horas del día con mayor número de vuelos y, como era de esperar la mayoría de los vuelos se producen de 6 de la mañana a 7 de la tarde. Cayendo significativamente a valores prácticamente nulos en la madrugada.

```
#Horas de llegada
ggplot(vuelos_reduc, aes(x=ARRIVAL_HOUR, fill=ARRIVAL_HOUR )) +
  geom_bar( ) +
  scale_fill_hue(c = 20) +
  theme(legend.position="none")
```

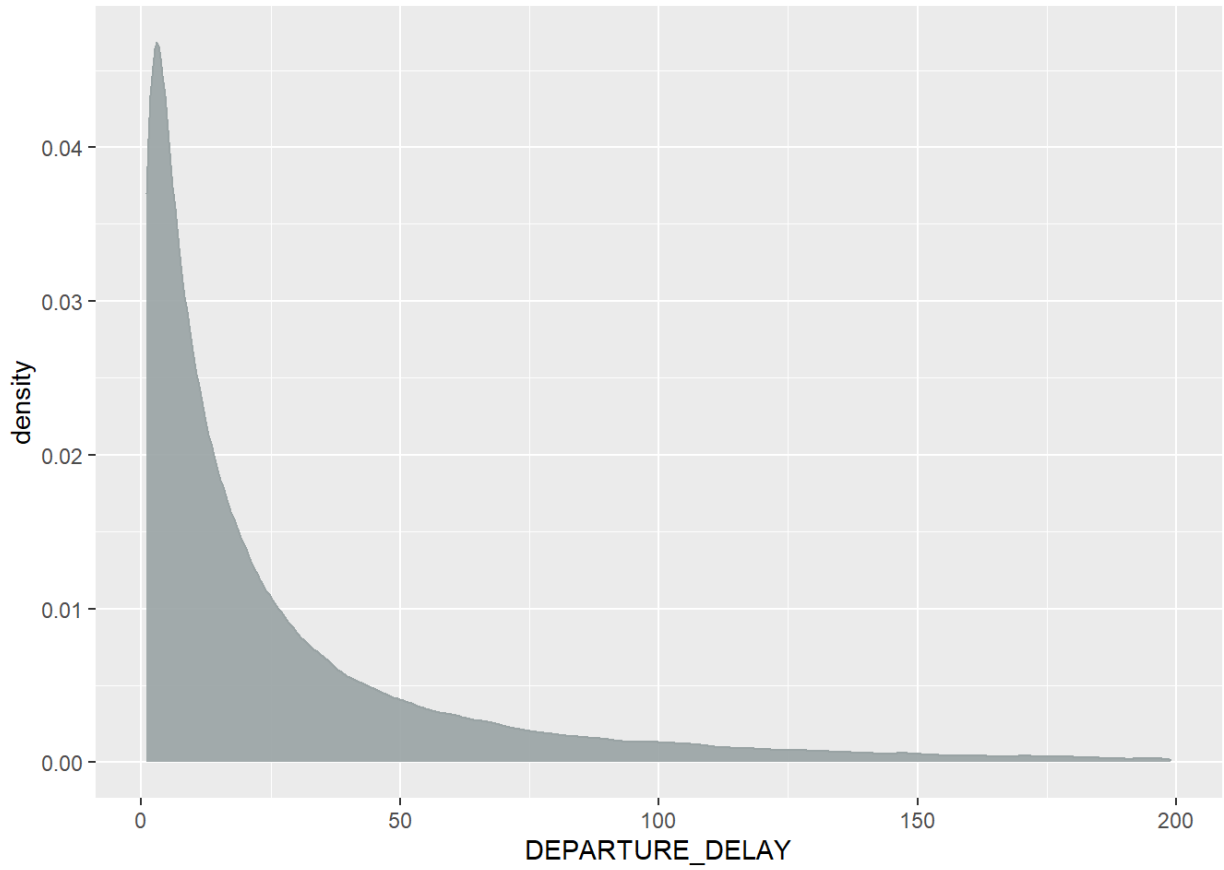


Por otro lado, las horas de llegadas más populares fueron de 9 de la mañana a 9 de la noche. Vemos que hay un desplazamiento de horario de 2-3 horas con respecto a la gráfica de las horas de salida, lo cual es lógico si tenemos en cuenta la duración de los vuelos.

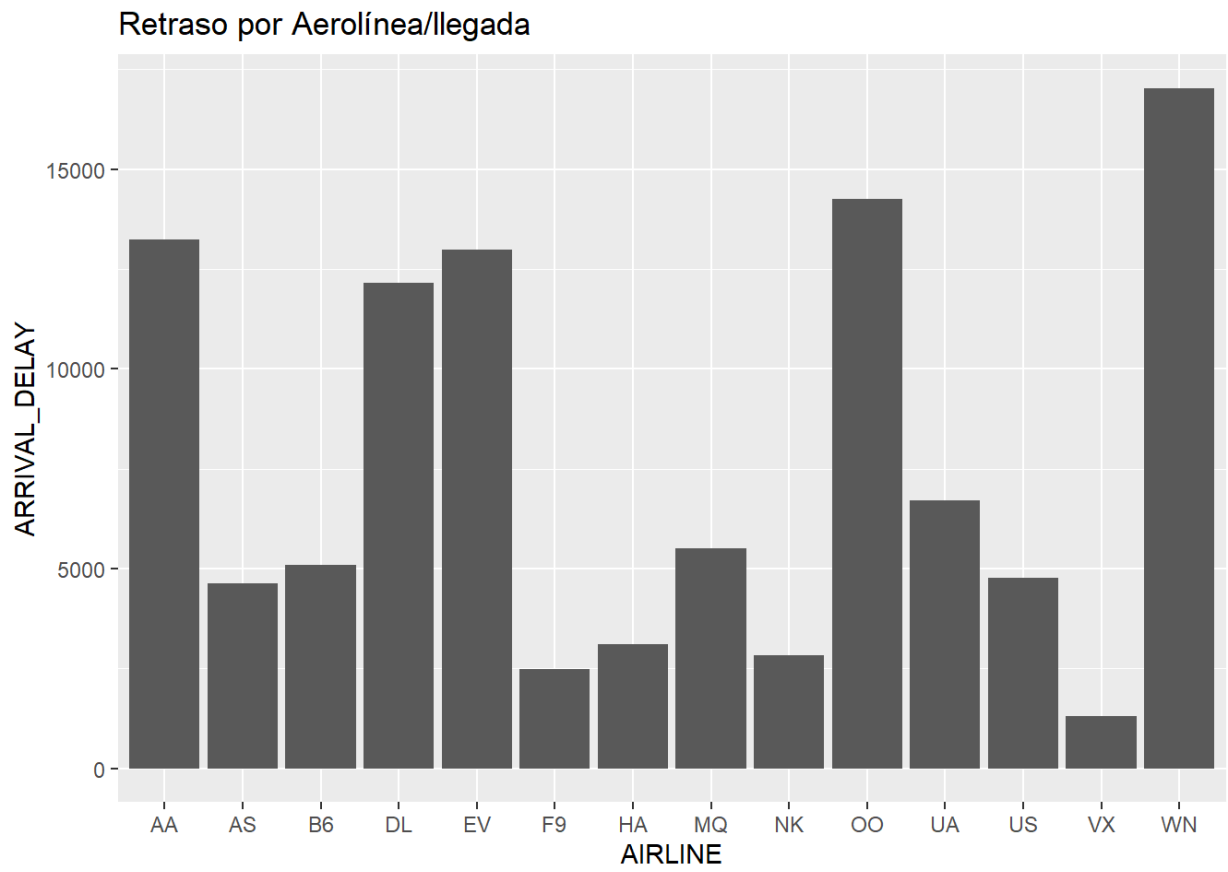
```
hist(vuelos_reduc$ARRIVAL_DELAY,breaks = 1000, xlim = c(-100,150))
```



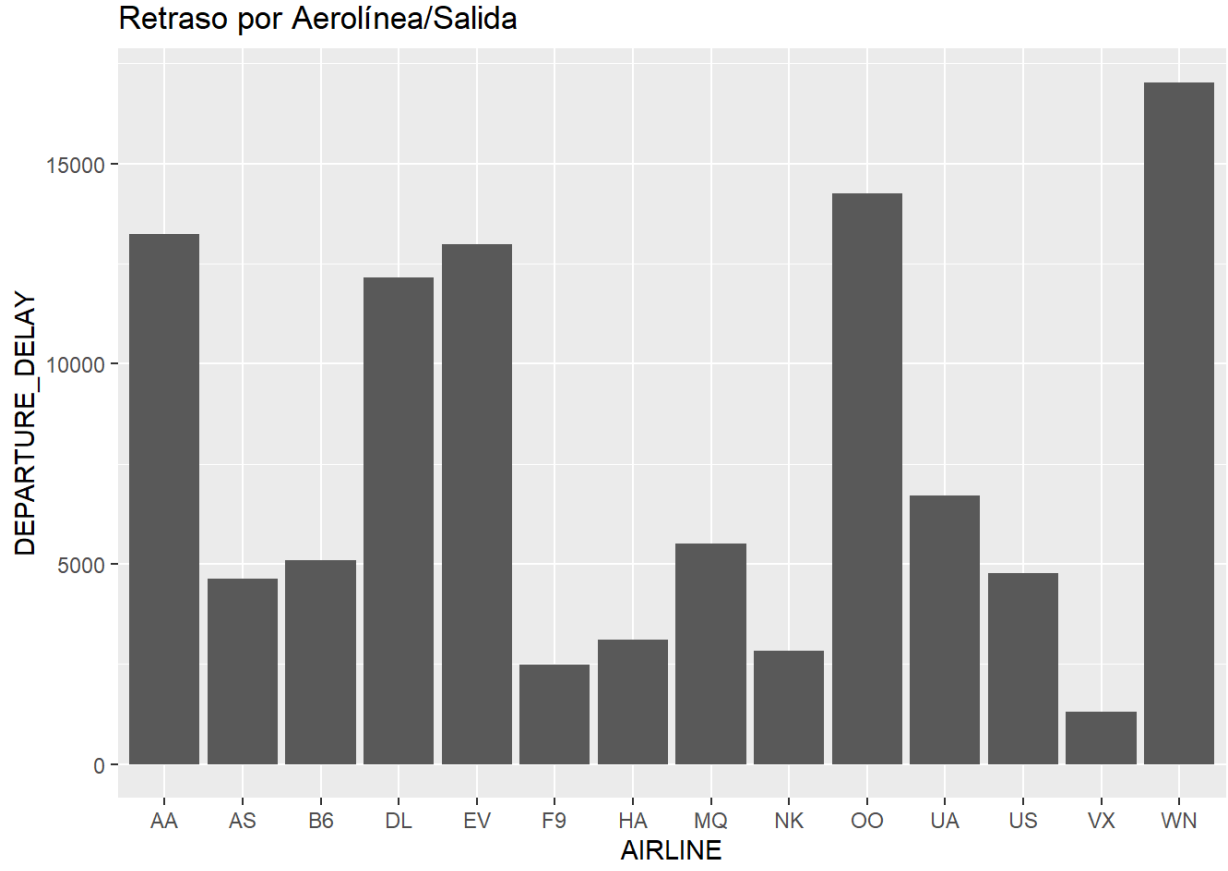
```
#Analizamos los vuelos retrasados cuando el retraso es menor de 200 minutos
vuelos_reduc %>%
  filter( DEPARTURE_DELAY<200 & DEPARTURE_DELAY>0) %>%
  ggplot( aes(x=DEPARTURE_DELAY)) +
  geom_density(fill="#99A3A4", color="#99A3A4", alpha=0.9)
```



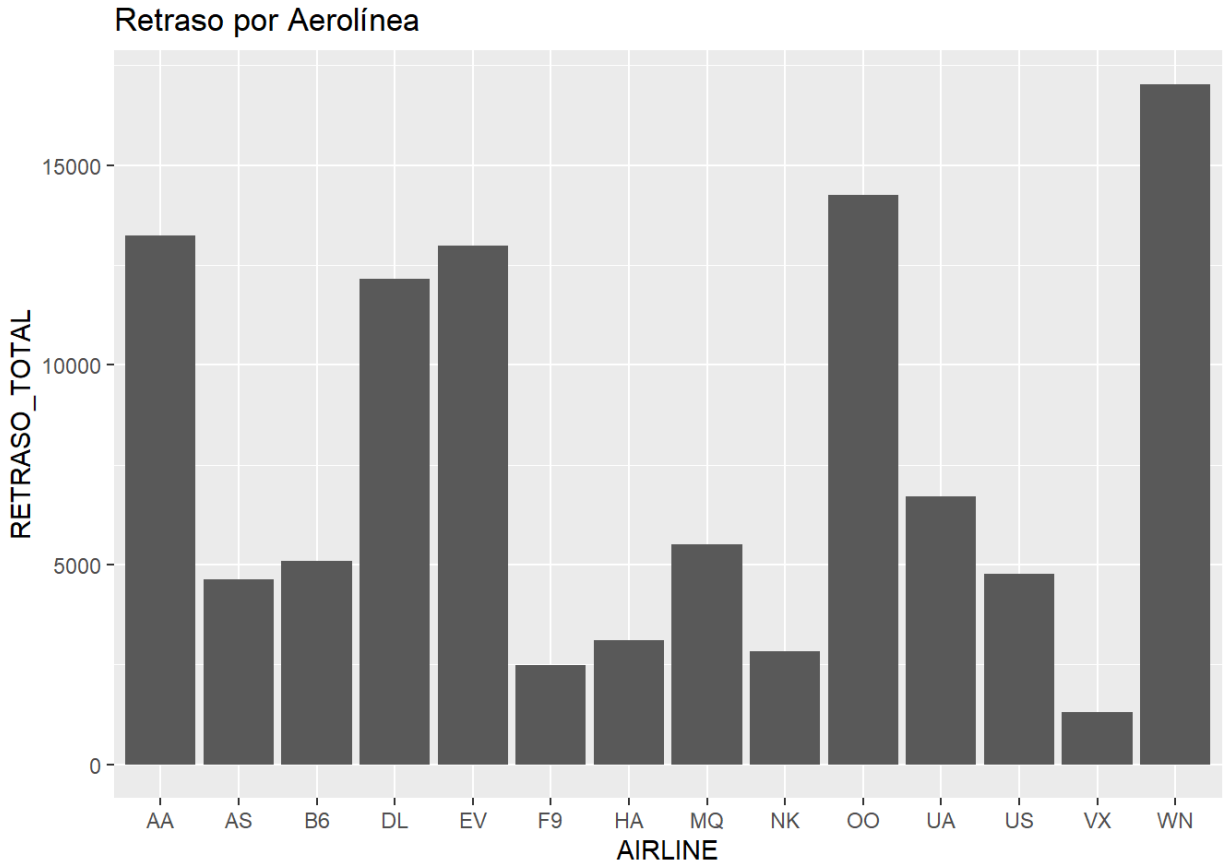

```
par(mfrow=c(2,2))

vuelos_reduc %>%
  filter( RETRASO_TOTAL>0) %>%
  ggplot(aes(AIRLINE,fill=ARRIVAL_DELAY))+geom_bar() +labs(x="AIRLINE", y="ARRI
VAL_DELAY")+ guides(fill=guide_legend(title=""))+ scale_fill_manual(values=c("b
lack","#008000"))+ggtitle("Retraso por Aerolínea/llegada")
```




```
vuelos_reduc %>%
  filter( RETRASO_TOTAL>0) %>%
  ggplot(aes(AIRLINE,fill=DEPARTURE_DELAY))+geom_bar() +labs(x="AIRLINE", y="DE
PARTURE_DELAY")+ guides(fill=guide_legend(title=""))+ scale_fill_manual(values=
c("black","#008000"))+ggtitle("Retraso por Aerolínea/Salida")
```

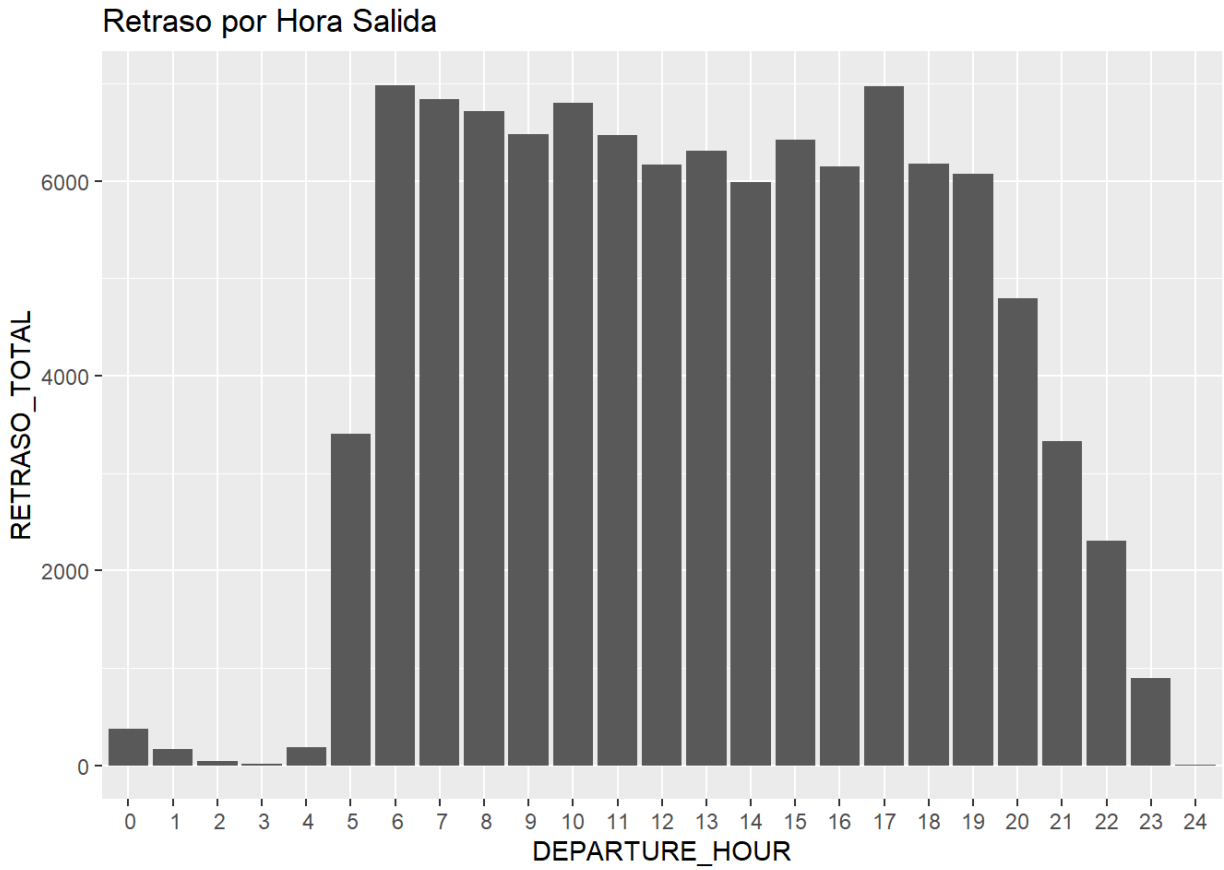


```
vuelos_reduc %>%
  filter( RETRASO_TOTAL>0) %>%
  ggplot(aes(AIRLINE,fill=RETRASO_TOTAL))+geom_bar() +labs(x="AIRLINE", y="RETR
ASO_TOTAL")+ guides(fill=guide_legend(title=""))+ scale_fill_manual(values=c("b
lack","#008000"))+ggtitle("Retraso por Aerolínea")
```



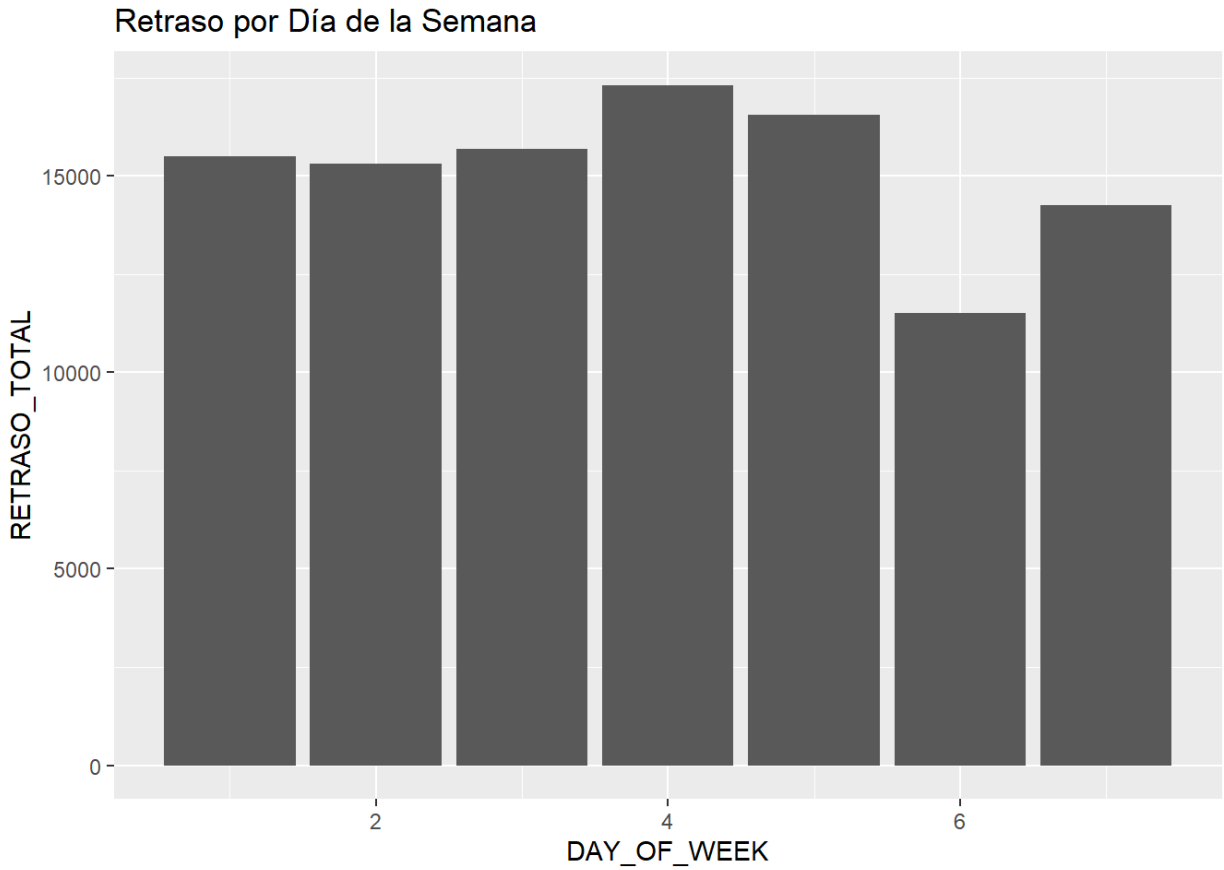
Comprobamos que los resultados entre los retrasos tanto de llegada como de salida por aerolínea son prácticamente iguales, por lo que Southwest Airlines fue la que más retrasos tuvo en las llegadas. Por otro lado, es lógico que si un vuelo se retrasa en la salida, se retrase también en la llegada.

```
vuelos_reduc %>%
  filter( RETRASO_TOTAL>0) %>%
  ggplot(aes(DEPARTURE_HOUR,fill=RETRASO_TOTAL))+geom_bar() +labs(x="DEPARTURE_
HOUR", y="RETRASO_TOTAL")+ guides(fill=guide_legend(title=""))+ scale_fill_manu
al(values=c("black","#008000"))+ggtitle("Retraso por Hora Salida")
```



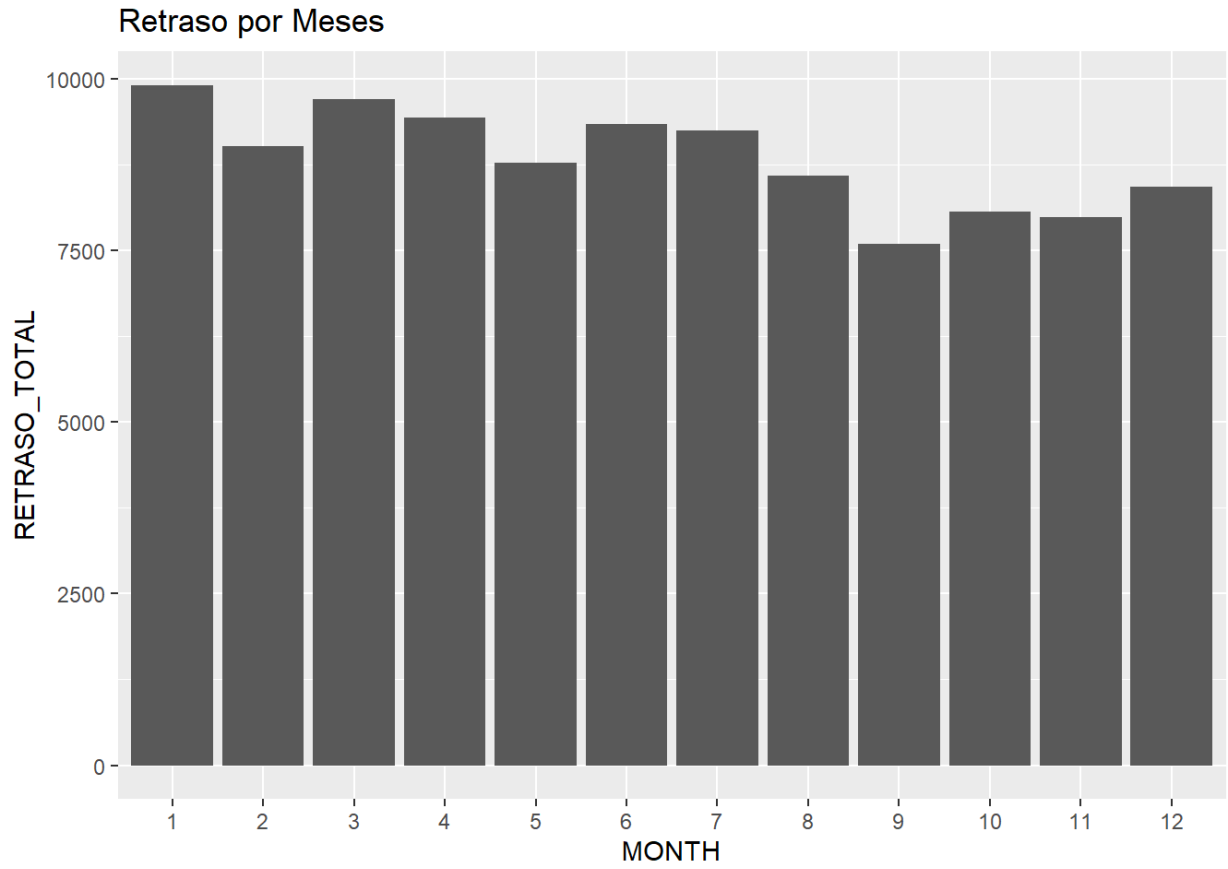
En cuanto al retraso por hora de salida, podemos identificar que las horas más problemáticas se tienen en la franja entre las 6am y las 7pm, siendo las 6am, las 10am y las 5pm las horas que acumulan mayor retraso.En esta gráfica vemos totales, no porcentajes, y es lógico que en las horas de más afluencia haya más vuelos retrasados.

```
vuelos_reduc %>%
  filter( RETRASO_TOTAL>0) %>%
  ggplot(aes(DAY_OF_WEEK,fill=RETRASO_TOTAL))+geom_bar() +labs(x="DAY_OF_WEEK",
y="RETRASO_TOTAL")+ guides(fill=guide_legend(title=""))+ scale_fill_manual(valu
es=c("black","#008000"))+ggtitle("Retraso por Día de la Semana")
```



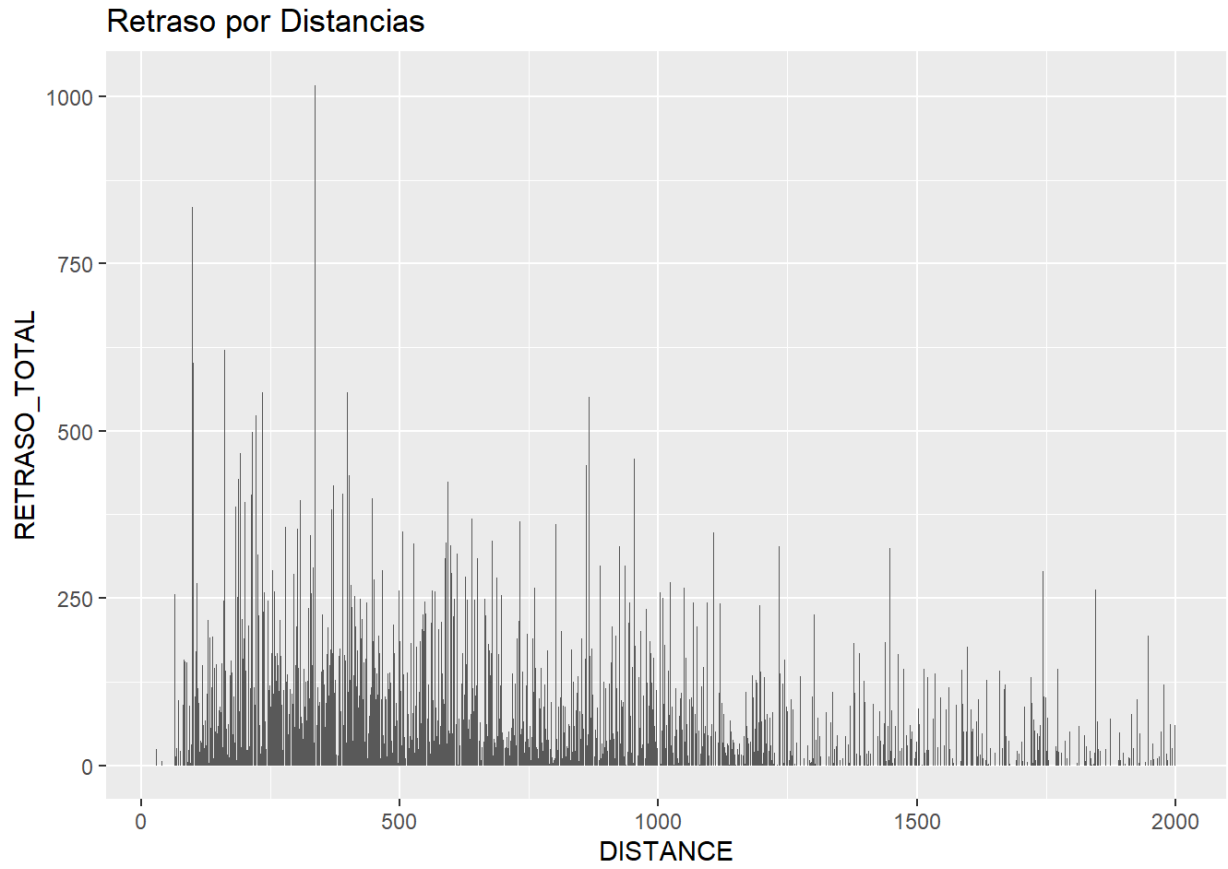
Según el gráfico podemos decir que los días donde se concentra el mayor número de retrasos son el jueves y el viernes. Según esto, el mejor día para viajar sería el sábado, también coincidiendo con el día de menos afluencia de vuelos.

```
vuelos_reduc %>%
  filter( RETRASO_TOTAL>0) %>%
   ggplot(aes(factor(MONTH),fill=RETRASO_TOTAL))+geom_bar() +labs(x="MONTH", y=
"RETRASO_TOTAL")+ guides(fill=guide_legend(title=""))+ scale_fill_manual(values
=c("black","#008000"))+ggtitle("Retraso por Meses")
```



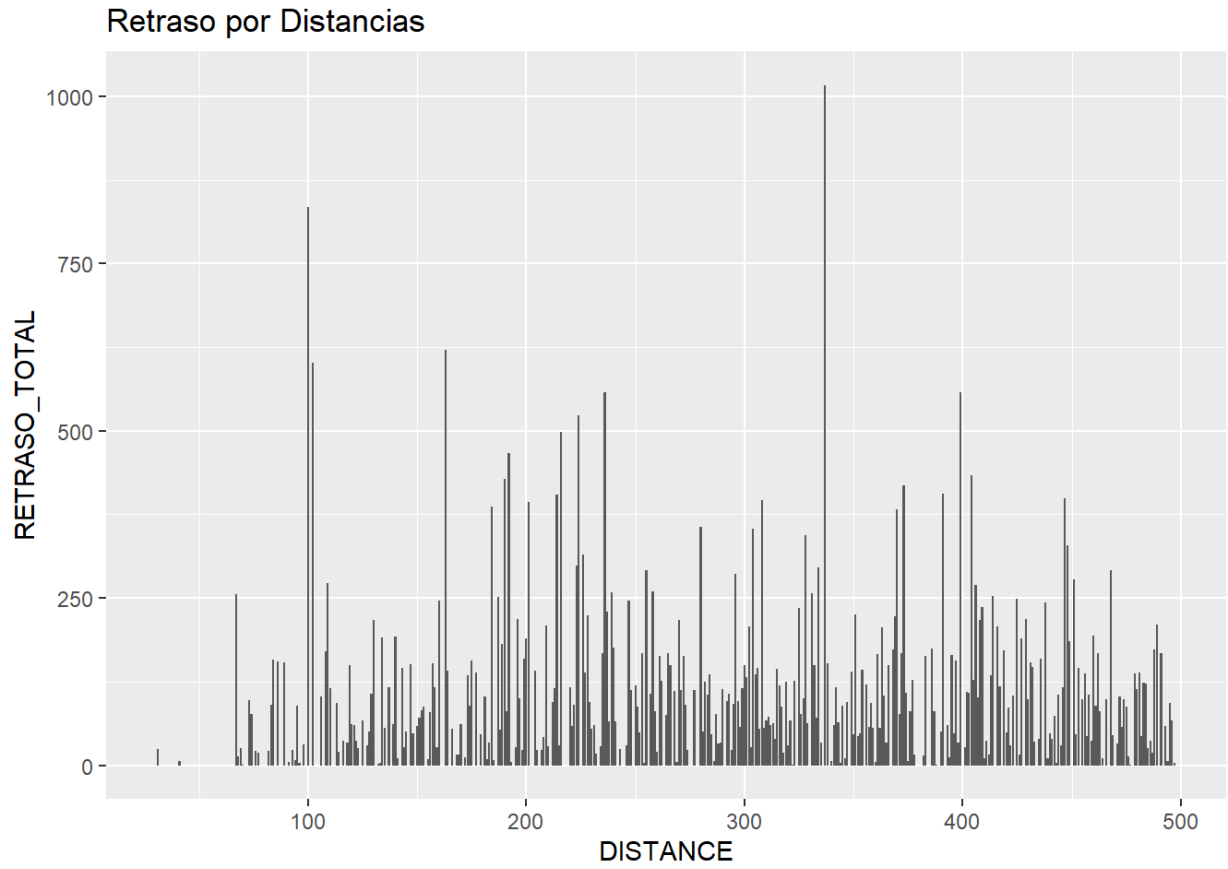
Según el gráfico, parece que el mes con más retrasos es enero, algo que no es lo que podríamos esperar en un inicio, tendríamos que hacer un estudio de las causas de los retrasos para poder valorar bien este dato. Los mejores meses para viajar parecen ser los meses de septiembre, octubre y noviembre. Al ver la gráfica en porcentaje, observaremos que la distribución de vuelos retrasados por porcentaje es algo diferente y nos decuelve otros resultados.

```
vuelos_reduc %>%
  filter( RETRASO_TOTAL>0 & DISTANCE<2000) %>%
  ggplot(aes(DISTANCE,fill=RETRASO_TOTAL))+geom_bar() +labs(x="DISTANCE", y="RE
TRASO_TOTAL")+ guides(fill=guide_legend(title=""))+ scale_fill_manual(values=c(
"black","#008000"))+ggtitle("Retraso por Distancias")
```



Comprobamos que los mayores retrasos se tienen en distancias más cortas, distancias menores de 1000 millas, y se tienen picos de retraso en distancias menores de 500 millas. Vamos a centrar un poco más el gráfico para comprobar que distancias dan esos picos de retraso.

```
vuelos_reduc %>%
  filter( RETRASO_TOTAL>0 & DISTANCE<500) %>%
  ggplot(aes(DISTANCE,fill=RETRASO_TOTAL))+geom_bar() +labs(x="DISTANCE", y="RE
TRASO_TOTAL")+ guides(fill=guide_legend(title=""))+ scale_fill_manual(values=c(
"black","#008000"))+ggtitle("Retraso por Distancias")
```



A continuación, vamos a ver varias gráficas que nos aportan información en % con lo que podemos sacar mejores conclusiones sobre los retrasos y las aerolíneas, aeropuertos, etc. Para ello vamos a trabajar con el conjunto original de datos, por lo que los resultados obtenidos serán los reales el año 2015.

Comenzaremos obteniendo un conjunto de datos que no contenga datos desconocidos para poder estimar los porcentajes:

```
indice <- which(is.na(vuelos$AIR_SYSTEM_DELAY)) #locations of obs with missing
 values
vuelos_delay <- vuelos[-indice,]
dim(vuelos_delay)
```

```
## [1] 1063439      31
```

Lo que vamos a hacer ahora es ver los vuelos que no sufren retrasos de nuestro dataset.

```
vuelos_no_delay <- vuelos[indice,]
dim(vuelos_no_delay)
```

```
## [1] 4755640      31
```

```
head(vuelos_no_delay)
```

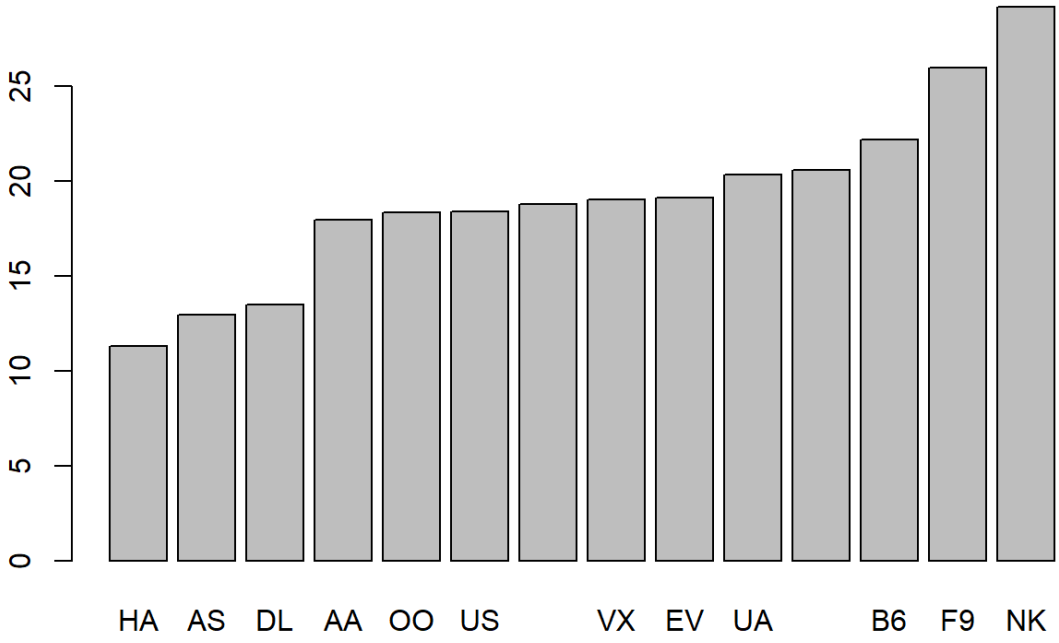
```
##   YEAR MONTH DAY DAY_OF_WEEK AIRLINE FLIGHT_NUMBER TAIL_NUMBER ORIGIN_AIRPOR
T
## 1 2015     1   1           4      AS            98      N407AS            AN
C
## 2 2015     1   1           4      AA          2336      N3KUAA            LA
X
## 3 2015     1   1           4      US           840      N171US            SF
O
## 4 2015     1   1           4      AA           258      N3HYAA            LA
X
## 5 2015     1   1           4      AS           135      N527AS            SE
A
## 6 2015     1   1           4      DL           806      N3730B            SF
O
##   DESTINATION_AIRPORT SCHEDULED_DEPARTURE DEPARTURE_TIME DEPARTURE_DELAY
## 1                 SEA                   5           2354             -11
## 2                 PBI                  10              2              -8
## 3                 CLT                  20             18              -2
## 4                 MIA                  20             15              -5
## 5                 ANC                  25             24              -1
## 6                 MSP                  25             20              -5
##   TAXI_OUT WHEELS_OFF SCHEDULED_TIME ELAPSED_TIME AIR_TIME DISTANCE WHEELS_O
N
## 1       21         15            205          194      169     1448       40
4
## 2       12         14            280          279      263     2330       73
7
## 3       16         34            286          293      266     2296       80
0
## 4       15         30            285          281      258     2342       74
8
## 5       11         35            235          215      199     1448       25
4
## 6       18         38            217          230      206     1589       60
4
##   TAXI_IN SCHEDULED_ARRIVAL ARRIVAL_TIME ARRIVAL_DELAY DIVERTED CANCELLED
## 1       4              430          408           -22        0         0
## 2       4              750          741            -9        0         0
## 3      11              806          811             5        0         0
## 4       8              805          756            -9        0         0
## 5       5              320          259           -21        0         0
## 6       6              602          610             8        0         0
##   CANCELLATION_REASON AIR_SYSTEM_DELAY SECURITY_DELAY AIRLINE_DELAY
## 1                                  NA             NA            NA
## 2                                  NA             NA            NA
## 3                                  NA             NA            NA
## 4                                  NA             NA            NA
## 5                                  NA             NA            NA
## 6                                  NA             NA            NA
##   LATE_AIRCRAFT_DELAY WEATHER_DELAY
## 1                  NA            NA
## 2                  NA            NA
```

```
## 3                NA              NA
## 4                NA              NA
## 5                NA              NA
## 6                NA              NA
```

Tenemos 1063439 vuelos que sufrieron retrasos frente a los 4755640 que no sufrieron retrasos, por lo que la probabilidad de que un vuelo se retrasara en el 2015 fue del 18.27%

Ahora, mostraremos una gráfica de barras donde podemos ver el porcentaje de retrasos de cada aerolínea:

```
barplot(sort(table(vuelos_delay$AIRLINE) / table(vuelos$AIRLINE) * 100))
```



Vemos que la que mayor porcentaje de vuelos retrasados tiene es NK. Esta gráfica nos aporta una información más valiosa que la anterior donde veíamos los retrasos totales por compañías, ya que en la anterior obteníamos que la que más retrasos registraba era WN, pero este dato se debía a que es la compañía que más opera en Estados Unidos.

Los porcentajes exactos los vemos en la siguiente tabla:

```
sort((table(vuelos_delay$AIRLINE) / table(vuelos$AIRLINE) * 100)) %>% knitr::ka
ble("html")  %>% kable_styling(position='center', font_size=12, fixed_thead=lis
t(enabled=T))
```

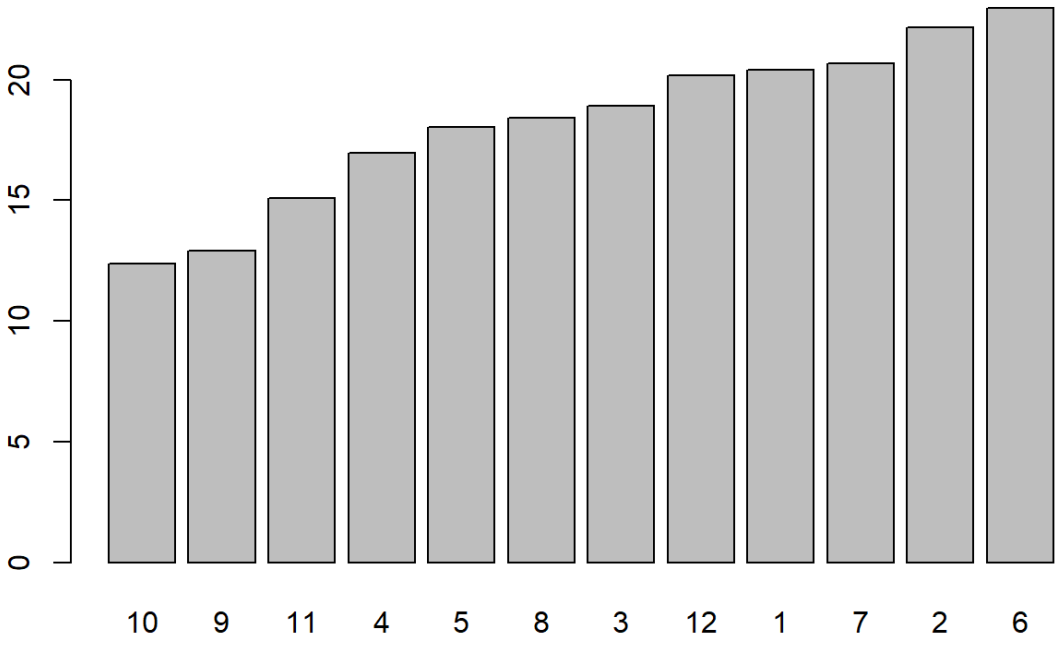
Var1	Freq
HA	11.29903
AS	12.95610

Var1	Freq
DL	13.47478
AA	17.94516
OO	18.32148
US	18.39267
WN	18.75223
VX	19.02654
EV	19.08888
UA	20.30586
MQ	20.55004
B6	22.15894
F9	25.94786
NK	29.15428

Apreciamos que NK tiene un 29.15 % de vuelos retrasados, frente al 11.29 de HA que es la que menor porcentaje presenta.

Ahora lo que analizaremos será el porcentaje de vuelos retrasados según los meses:

```
barplot(sort(table(vuelos_delay$MONTH) / table(vuelos$MONTH) * 100))
```



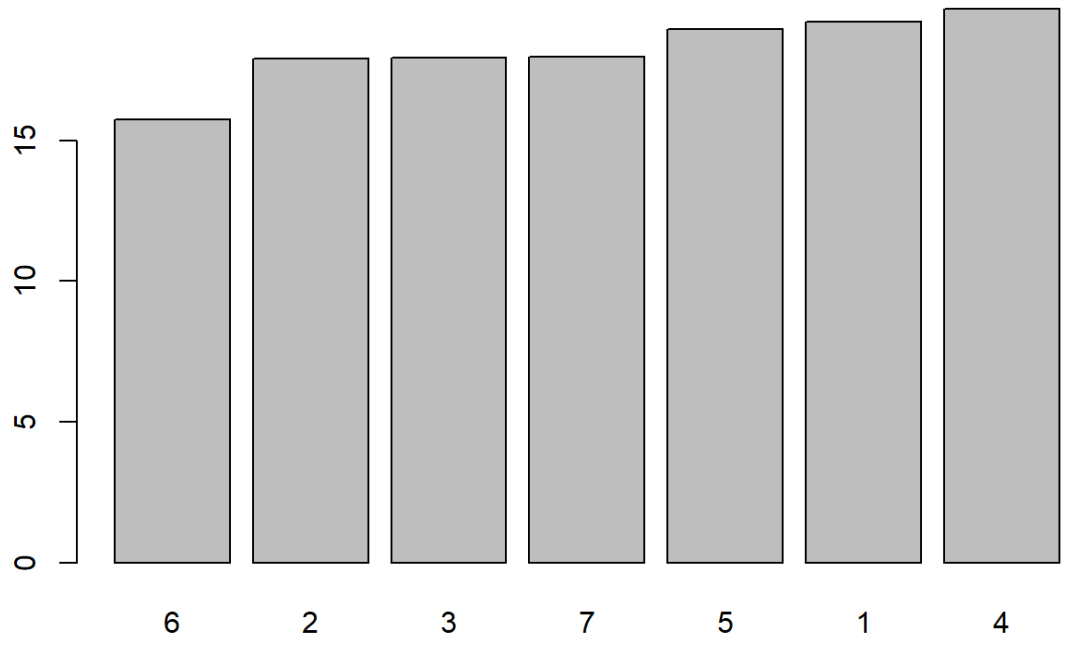
```
table(vuelos_delay$MONTH) / table(vuelos$MONTH) * 100
```

```
##
##        1        2        3        4        5        6        7        8
## 20.41650 22.17637 18.92717 16.95287 18.03748 22.96938 20.66896 18.43416
##        9       10       11       12
## 12.91784 12.35774 15.08018 20.19323
```

Vemos como el mes de junio es el que presenta más retrasos. Un dato curioso que obtenemos de esta gráfica es que observamos que el mes con mayor % de retrasos no se corresponde con el mes con mayor número de vuelos. En una gráfica anterior podíamos ver como julio fue el mes con mayor número de vuelos en el año 2015 en Estados Unidos.

Ya que con los meses hemos obtenido resultados que no esperábamos, vamos a ver si con los días de la semana ocurre lo mismo. Recordemos que era el jueves el día de la semana con mayor número de vuelos:

```
barplot(sort(table(vuelos_delay$DAY_OF_WEEK) / table(vuelos$DAY_OF_WEEK) * 100
))
```



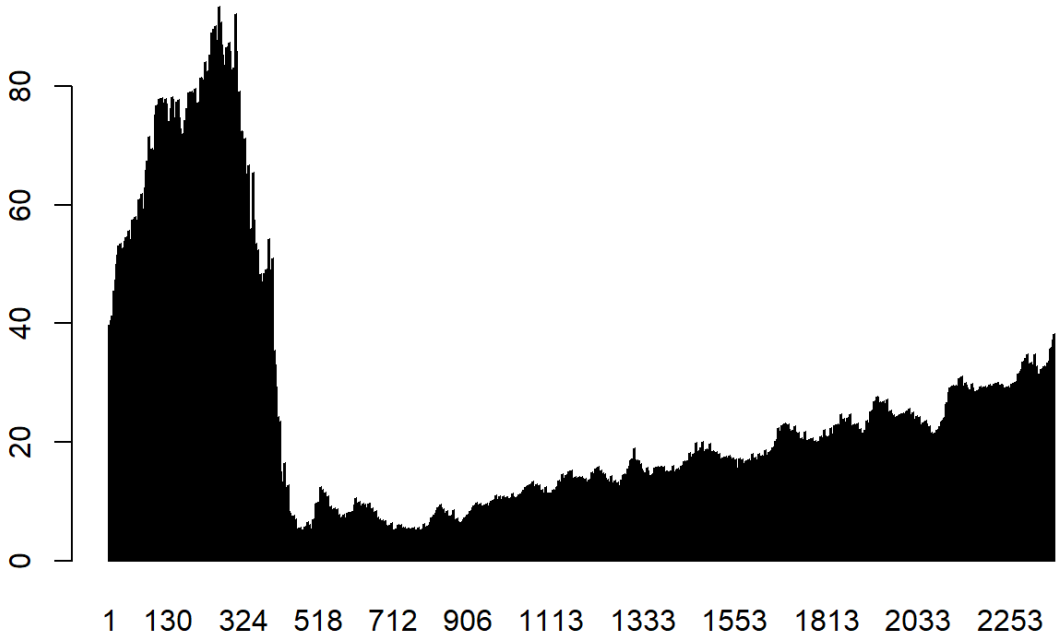

```
table(vuelos_delay$DAY_OF_WEEK) / table(vuelos$DAY_OF_WEEK) * 100
```

```
##
##        1        2        3        4        5        6        7
## 19.20944 17.89995 17.94655 19.67769 18.95608 15.74703 17.96826
```

En esta ocasión obtenemos el resultado esperado: el jueves es el día con mayor % de vuelos retrasados, lo que es lógico al ser el día de la semana con mayor número de vuelos.

Lo que vamos a mostrar en la siguiente gráfica son las franjas horarias donde se producen los retrasos:

```
barplot(table(vuelos_delay$ARRIVAL_TIME) / table(vuelos$ARRIVAL_TIME) * 100)
```



Vemos que las franjas horarias donde llegan un mayor porcentaje de vuelos retrasados es de 1 a 5 de la madrugada, coincidiendo con el menor número de vuelos. Mientras que cuando más retrasos se producen es de 7 a 8, que coincide con el mayor número de vuelos. Este resultado es lógico, ya que a mayor volumen de tráfico aéreo, más puntuales deben de ser porque un retraso podría tener efecto cadena, por lo que en las horas puntas debe haber una mayor puntualidad.

Otro dato curioso que podemos obtener es ver el porcentaje de vuelos retrasados según el aeropuerto del que despegan:

```
t <- sort(table(vuelos_delay$ORIGIN_AIRPORT) / table(vuelos$ORIGIN_AIRPORT) * 1
00, decreasing = TRUE)

head(t)
```

```
##
##    10165    14222      GST    10154    13964      ADK
## 55.55556 44.44444 44.15584 39.28571 38.88889 38.54167
```

Mostramos los 5 valores mayores y vemos como el aeropuerto con mayor porcentaje de vuelos retrasado es el de GST con un 44.15%

Ahora haremos lo mismo pero según el aeropuerto de destino:

```
t1 <- sort(table(vuelos_delay$DESTINATION_AIRPORT) / table(vuelos$DESTINATION_A
IRPORT) * 100, decreasing = TRUE)

head(t1)
```

```
##
##    13964      STC      PBG      GUM    14025    15070
## 38.88889 36.58537 32.26950 30.83832 30.76923 30.64516
```

En esta ocasión es el aeropuerto de STC el que presenta un mayor porcentaje con un 36.53%.
Por lo tanto, las conclusiones que podemos sacar de estas últimas gráficas es que para reducir la posibilidad de que nuestro vuelo sufriera un retraso en el año 2015 deberíamos haber evitado volar desde el aeropuerto de GST (Gustavus Airport), que el aeropuerto de destino no fuera el de STC (St. Cloud Regional Airport), que la aerolínea no fuera NK (Spirit Air Lines) y, no volar un jueves de junio.

---

6. Resolución del problema

A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

Aunque durante el desarrollo de la práctica hemos ido analizando los resultados obtenidos, en este apartado vamos a responder a una serie de preguntas a modo resumen: - ¿Son más los vuelos retrasados que los que llegan en hora? ¿Probabilidades? Al contrario de lo que podríamos pensar, la probabilidad de vuelos retrasados es menor que la de vuelos que llegan a su hora. De hecho, los vuelos retrasados en el 2015 fueron 1063439, frente a los 4755640 que llegaron a su hora, por lo que la probabilidad de que un vuelo se retrasara es del 18.27%

- ¿Qué aeropuerto evitar? El aeropuerto que debemos evitar a la hora de coger un vuelo es el de GST (Gustavus Airport), mientras que el aeropuerto de llegada a evitar es el STC (St. Cloud Regional Airport).
- ¿Qué aerolínea evitar? La aerolínea NK (Spirit Air Lines) es la que mayor porcentaje de vuelos retrasados presenta.
- ¿Qué meses, días, horas son los mejores y los peores para viajar para evitar retrasos? El peor mes es junio, mientras que el mejor es octubre El peor día es el jueves, mientras que el mejor es el sábado. A pesar de ese hecho, hemos comprobado que la razón entre la ocurrencia de retraso del vuelo frente a no retraso de 1.068 veces superior en fin de semana.
  Las peores horas de llegada del vuelo son de 1 a 5 de la madrugada, mientras que las mejores son de 7 a 8 de la mañana. Es decir los vuelos que llegan muy tarde llegan más retrasados, mientras que los que lo hacen a primera hora parece que no acumulan tanto retraso.
- ¿Es mayor el retraso de los vuelos en distancias largas?
  Según las gráficas de retraso por distancia, los mayores retrasos se dan en vuelos por debajo de las 1000 millas.

---

7. Código

Adjuntar el código con el que se ha realizado la limpieza, análisis y representación de los datos.

https://github.com/igonzalezvalle/PRA2---Limpieza-Analisis-Datos/blob/master/src/PRA2-Limpieza-Analisis.R (https://github.com/igonzalezvalle/PRA2---Limpieza-Analisis-Datos/blob/master/src/PRA2-Limpieza-Analisis.R)

8. Contribuciones al trabajo

Contribuciones	Firma
Investigación previa	IGV, CMGR
Redacción de las respuestas	IGV, CMGR
Desarrollo código	IGZ, CMGR

Referencias:

https://rpubs.com (https://rpubs.com) https://www.kaggle.com/usdot/flight-delays (https://www.kaggle.com/usdot/flight-delays) https://rstudio-pubs-static.s3.amazonaws.com/326515_e25772acbbba47d98052df5520eed1a4.html (https://rstudio-pubs-static.s3.amazonaws.com/326515_e25772acbbba47d98052df5520eed1a4.html) https://www.r-graph-gallery.com/ (https://www.r-graph-gallery.com/)

- Laia Subirats Maté, Diego Oswaldo Pérez Trenard, Mireia Calvo González. (2019). Introducción a la limpieza de datos.
- Squire, Megan (2015). Clean Data. Packt Publishing Ltd.
- Jiawei Han, Micheine Kamber, Jian Pei (2012). Data mining: concepts and techniques. Morgan Kaufmann.
- Jason W. Osborne (2010). Data Cleaning Basics: Best Practices in Dealing with Extreme Scores. Newborn and Infant Nursing Reviews; 10 (1): pp. 1527-3369.
- Peter Dalgaard (2008). Introductory statistics with R. Springer Science & Business Media. Wes McKinney (2012). Python for Data Analysis. O'Reilley Media, Inc.