Máster en Data Science. URJC
Técnicas y Métodos de Ciencia de Datos
Práctica Final R

2017

Índice

0.1 Introducción

El paquete `nycflights13`, disponible en CRAN, contiene datos sobre 336.776 vuelos que despegaron de alguno de los tres aeropuertos que dan servicio a la ciudad de Nueva York (EE.UU.) en 2013, procedentes del Bureau of Transport Statistics:

- Aeropuerto Internacional Libertad de Newark (EWR).
- Aeropuerto Internacional John. F. Kennedy (JFK).
- Aeropuerto Internacional de La Guardia (LGA).

El conjunto principal de datos sobre los vuelos está disponible en el `data.frame flights`, dentro de este paquete. Adicionalmente, su autor (Hadley Wickham) también ha incluido datos sobre los propios aeropuertos, condiciones meteorológicas, etc. Para más detalles, ver archivo de descripción del paquete con el comando `?nycflights13`.

0.2 Preparación del ejercicio

Durante el ejercicio, se utilizarán las bibliotecas `ggplot2` y `dplyr`, ya introducidas en clase.

Nota importante 1: Se recomienda revisar y practicar con los ejemplos del documento de introducción a `dplyr` antes de realizar este ejercicio, así como los ejemplos incluidos en el seminario de H. Wickham sobre "Tidy Data", enlazado en la sección referencias del Tema 2 en Aula Virtual.

Nota importante 2: intente utilizar el operador %>% (*forward pipe*) para el código de resolución de todos los ejercicios.

```
# Importamos bibliotecas y datos
library(ggplot2)
library(dplyr)
library(nycflights13)
library(cowplot)
```

0.3 Ejercicio 1 (*30 puntos*)

Utiliza las funciones incluidas en el paquete `dplyr`, para responder a las siguientes preguntas:

a) ¿Cuantos vuelos se realizan en total cada mes?
b) ¿Qué aeropuerto acumula el mayor nÃºmero de salidas de vuelos en todo el año?
c) ¿Qué compañía acumula el mayor nÃºmero de salida de vuelos en los meses de verano (jun-sep.)?
d) ¿Qué compañia acumula más tiempo de vuelo en todo el año?
e) ¿Qué compañía registra los mayores retrasos de salida de sus vuelos? ¿Tienen los retrasos alguna correlación con la duración de los vuelos?

0.3.1 Estudio de los datos

Antes de enfrentarnos a la resolución del ejercicio realizamos un análisis exploratorio sobre los datos con el fin de observar el tipo de datos que conforman la muestra a estudio y realizar una correcta conversión hacia *tidy data*.

```
summary(flights)
```

```
##       year          month             day           dep_time
##  Min.   :2013   Min.   : 1.000   Min.   : 1.00   Min.   :   1
##  1st Qu.:2013   1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.: 907
##  Median :2013   Median : 7.000   Median :16.00   Median :1401
##  Mean   :2013   Mean   : 6.549   Mean   :15.71   Mean   :1349
##  3rd Qu.:2013   3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:1744
##  Max.   :2013   Max.   :12.000   Max.   :31.00   Max.   :2400
##                                                  NA's   :8255
##  sched_dep_time   dep_delay          arr_time    sched_arr_time
##  Min.   : 106   Min.   : -43.00   Min.   :   1   Min.   :   1
##  1st Qu.: 906   1st Qu.:  -5.00   1st Qu.:1104   1st Qu.:1124
##  Median :1359   Median :  -2.00   Median :1535   Median :1556
##  Mean   :1344   Mean   :  12.64   Mean   :1502   Mean   :1536
##  3rd Qu.:1729   3rd Qu.:  11.00   3rd Qu.:1940   3rd Qu.:1945
```

```
##  Max.   :2359   Max.   :1301.00   Max.   :2400   Max.   :2359
##                 NA's   :8255      NA's   :8713
##    arr_delay          carrier              flight       tailnum
##  Min.   : -86.000   Length:336776      Min.   :   1   Length:336776
##  1st Qu.: -17.000   Class :character   1st Qu.: 553   Class :character
##  Median :  -5.000   Mode  :character   Median :1496   Mode  :character
##  Mean   :   6.895                      Mean   :1972
##  3rd Qu.:  14.000                      3rd Qu.:3465
##  Max.   :1272.000                      Max.   :8500
##  NA's   :9430
##     origin              dest              air_time        distance
##  Length:336776      Length:336776      Min.   : 20.0   Min.   :  17
##  Class :character   Class :character   1st Qu.: 82.0   1st Qu.: 502
##  Mode  :character   Mode  :character   Median :129.0   Median : 872
##                                        Mean   :150.7   Mean   :1040
##                                        3rd Qu.:192.0   3rd Qu.:1389
##                                        Max.   :695.0   Max.   :4983
##                                        NA's   :9430
##       hour           minute        time_hour
##  Min.   : 1.00   Min.   : 0.00   Min.   :2013-01-01 05:00:00
##  1st Qu.: 9.00   1st Qu.: 8.00   1st Qu.:2013-04-04 13:00:00
##  Median :13.00   Median :29.00   Median :2013-07-03 10:00:00
##  Mean   :13.18   Mean   :26.23   Mean   :2013-07-03 05:02:36
##  3rd Qu.:17.00   3rd Qu.:44.00   3rd Qu.:2013-10-01 07:00:00
##  Max.   :23.00   Max.   :59.00   Max.   :2013-12-31 23:00:00
##
```

El data frame *flights* esta compuesto por quince columnas, en las cuales los datos son de tipo numérico, carácter y fechas. Encontramos diversos *NA* repartidos entre las variables que indican tiempo respecto al avión. Siendo minutos la unidad de medida de tiempo utilizada en todos las variables de observaciones temporales relacionadas con el avión y millas la utilizada para las observaciones de distancia.

0.3.2 Limpieza de datos

Con el estudio preliminar ya podemos ver donde se encontraron los problemas de *missing values*, por tanto, podemos entrar directamente a resolverlos. Debido a la distribución del ejercicio, se aplicará en los distintos apartados la correspondiente limpieza de los datos.

0.3.3 Caso A: ¿Cuantos vuelos se realizan cada mes en total?

0.3.3.1 Tratamiento de datos

Utilizamos para la resolución las funciones que nos brinda el paquete *dplyr* y el operador *pipe* para obtener un código más ordenado.

```
flights %>%
  group_by(year, month) %>%
   summarise(total = n()) %>%
    select(year, month, total)
```

```
## # A tibble: 12 x 3
## # Groups:   year [1]
##     year month total
##    <int> <int> <int>
##  1  2013     1 27004
##  2  2013     2 24951
##  3  2013     3 28834
##  4  2013     4 28330
##  5  2013     5 28796
##  6  2013     6 28243
##  7  2013     7 29425
##  8  2013     8 29327
##  9  2013     9 27574
## 10  2013    10 28889
## 11  2013    11 27268
## 12  2013    12 28135
```

Se presentan los datos en formato *tibble* en el cual se encuentra el resultado en la columna *total*, con máximo en el mes de julio y el mínimo en febrero.

0.3.4 Caso B: ¿Que aeropuerto acumula el mayor número de salidas de vuelos en todo el año?

0.3.4.1 Tratamiento de datos

De nuevo como a lo largo de todo el informe utilizamos *dplyr* y *pipe* para obtener la solución. El formato de salida de la respuesta también es de tipo *tibble* y la solución se muestra la columna *faa*.

```
airport_max_flight <- flights %>%
                        group_by(year, origin) %>%
```

```
                        summarise(total = n()) %>%
                         filter(total == max(total)) %>%
                          rename(faa=origin)

inner_join(airports, airport_max_flight, by="faa")%>%
  select(faa,name, year, total)
```

```
## # A tibble: 1 x 4
##     faa                 name  year  total
##   <chr>               <chr> <int>  <int>
## 1   EWR Newark Liberty Intl  2013 120835
```

0.3.5 Caso C: ¿Que compañia acumula el mayor número de salidas de vuelos en los meses de verano (jun-sep)?

0.3.5.1 Tratamiento de datos

El resultado se muestra en formato *tibble* con solución en la columna *carrier*.

```
airline_max_summer <-flights %>%
                      filter(month %in% (6:8)) %>%
                       group_by(carrier) %>%
                        summarise(total = n()) %>%
                         filter(total == max(total))

inner_join(airlines,airline_max_summer, by = "carrier")
```

```
## # A tibble: 1 x 3
##   carrier                  name total
##     <chr>                <chr> <int>
## 1      UA United Air Lines Inc. 15165
```

0.3.6 Caso D: ¿Que compañia acumula más tiempo de vuelo en todo el año?

0.3.6.1 Limpieza de datos

Aqui nos encontramos el primer tratamiento para limpieza de datos, este debe afectar a las variables de estudio *air_time* y *carrier*. La función is.na(), nos devuelve un vector booleano de TRUE y FALSE, esta función en el caso de encontrar un *NA* nos devolvera un TRUE, en caso contrario FALSE. De esta forma podemos fijar la siguiente condición para limpiar los datos de una forma sencilla.

```
flights_cln <- flights[is.na(flights$air_time) == FALSE &
                       is.na(flights$carrier) == FALSE,]
```

0.3.6.2 Tratamiento de datos

Realizada la fase de limpieza pasamos a responder la cuestión. La solución se encuentra en la columna *carrier* y el número de vuelos correspondiente a esta aerolinea en la columna *total*.

```
airline_max_air_time <-flights_cln %>%
                        group_by(year,carrier) %>%
                         summarise(total = sum(air_time)) %>%
                          filter(total == max(total))

inner_join(airlines,airline_max_air_time, by = "carrier")
```

```
## # A tibble: 1 x 4
##   carrier                name  year    total
##     <chr>               <chr> <int>    <dbl>
## 1      UA United Air Lines Inc.  2013 12237728
```

0.3.7 Caso E.1: ¿Que compañia registra los mayores retrasos de salida en sus vuelos?

0.3.7.1 Limpieza de datos

De nuevo realizamos un desarrollo acorde con las especificaciones del enunciado. En este caso afecta a *dep_delay* y *carrier*.

```
flightslime<- flights[(is.na(flights$dep_delay) == FALSE) &
                      (is.na(flights$carrier) == FALSE),]
```

0.3.7.2 Tratamiento de datos

Se opera de una forma muy parecida al caso anterior. Con el fin de obtener un código más facil de seguir y legible sacamos la definición de *flights_delay* de la expresión solución. Tomamos los retrasos como el *dep_delay >0*, pero para obtener un resultado que englobe todos los datos, calcularemos este apartado con los datos que también cumplan *dep_delay<0*. Estos valores se corresponden con los vuelos que despegan antes de tiempo.

```
flights_delay <- flightslime %>%
                 select(carrier, dep_delay)
```

```
flights_delay_end <- flights_delay %>%
                      select(carrier, dep_delay) %>%
                       group_by(carrier) %>%
                        summarise(tot_delay = sum(dep_delay)) %>%
                         filter(tot_delay == max(tot_delay))

inner_join(airlines,flights_delay_end, by = "carrier")
```

```
## # A tibble: 1 x 3
##   carrier                     name tot_delay
##     <chr>                    <chr>     <dbl>
## 1      EV ExpressJet Airlines Inc.   1024829
```

0.3.8 Caso E.2: ¿Tienen los retrasos alguna correlación con la duración de los vuelos?

Una vez que se ha obtenido la aerolinea con máyor acumulación de retrasos de salida, *EV*, trabajamos sobre ella en este apartado.

0.3.8.1 Limpieza de datos

```
flightslimpe3 <- flights[ (is.na(flights$dep_delay) == FALSE) &
                          (is.na(flights$air_time) == FALSE),]
```

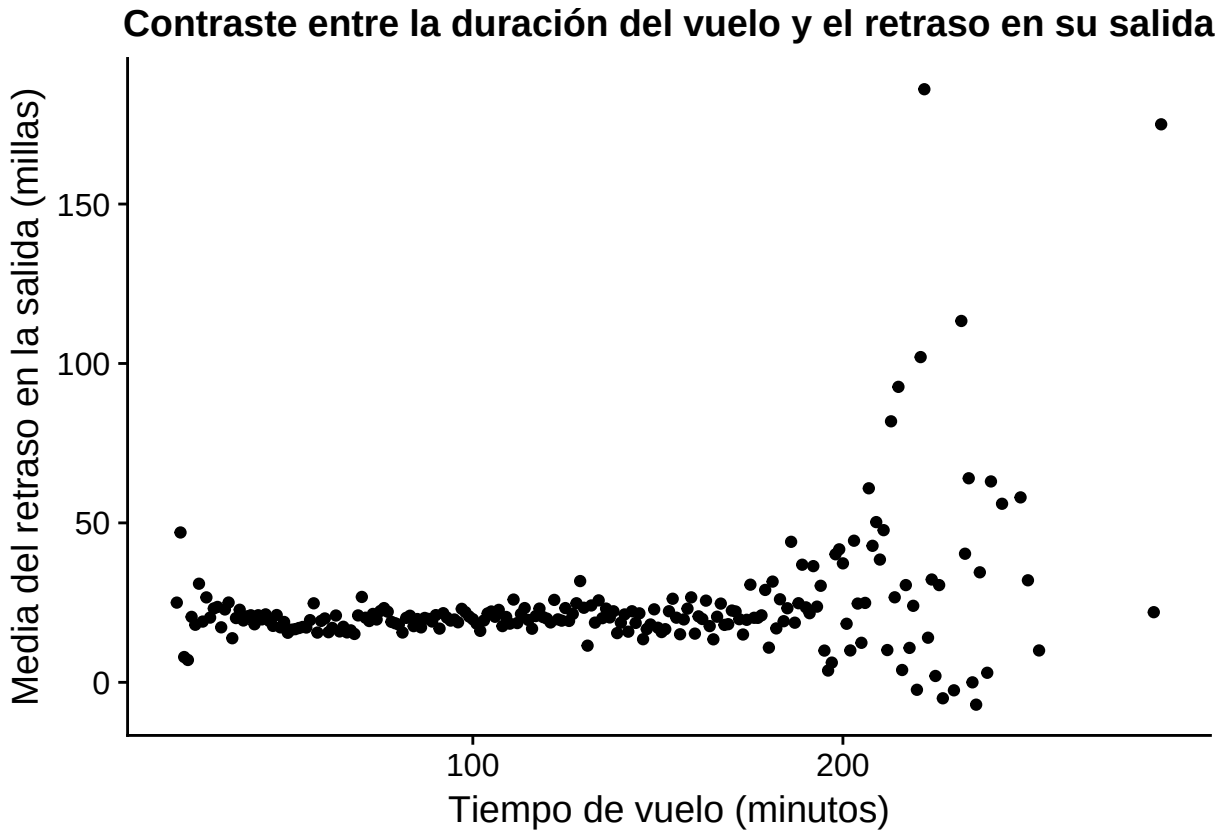
0.3.8.2 Tratamiento de datos

```
flights_contrast2 <- flightslimpe3%>%
                     select(air_time, dep_delay, carrier) %>%
                      group_by(air_time, carrier) %>%
                       filter(carrier=="EV") %>%
                       summarise(mediana_dep_delay = mean(dep_delay))
```

0.3.8.3 Visualización

En esta parte para poder sacar conclusiones sobre la correlación utilizaré diferentes gráficas. La primera de ellas solo tratará de los puntos reguistrados y concluir a través de ella si todos los puntos son correctos para el estudio(identificación de anomalias), en las gráficas utilizaré el modelo *gam* y *loess*.

```
ggplot(flights_contrast2, aes(air_time, mediana_dep_delay)) +
geom_point()+
 xlab("Tiempo de vuelo (minutos)") +
 ylab("Media del retraso en la salida (millas)") +

  ggtitle("Contraste entre la duración del vuelo y el retraso en su salida") +

  theme(plot.title = element_text(hjust = 0.5))
```

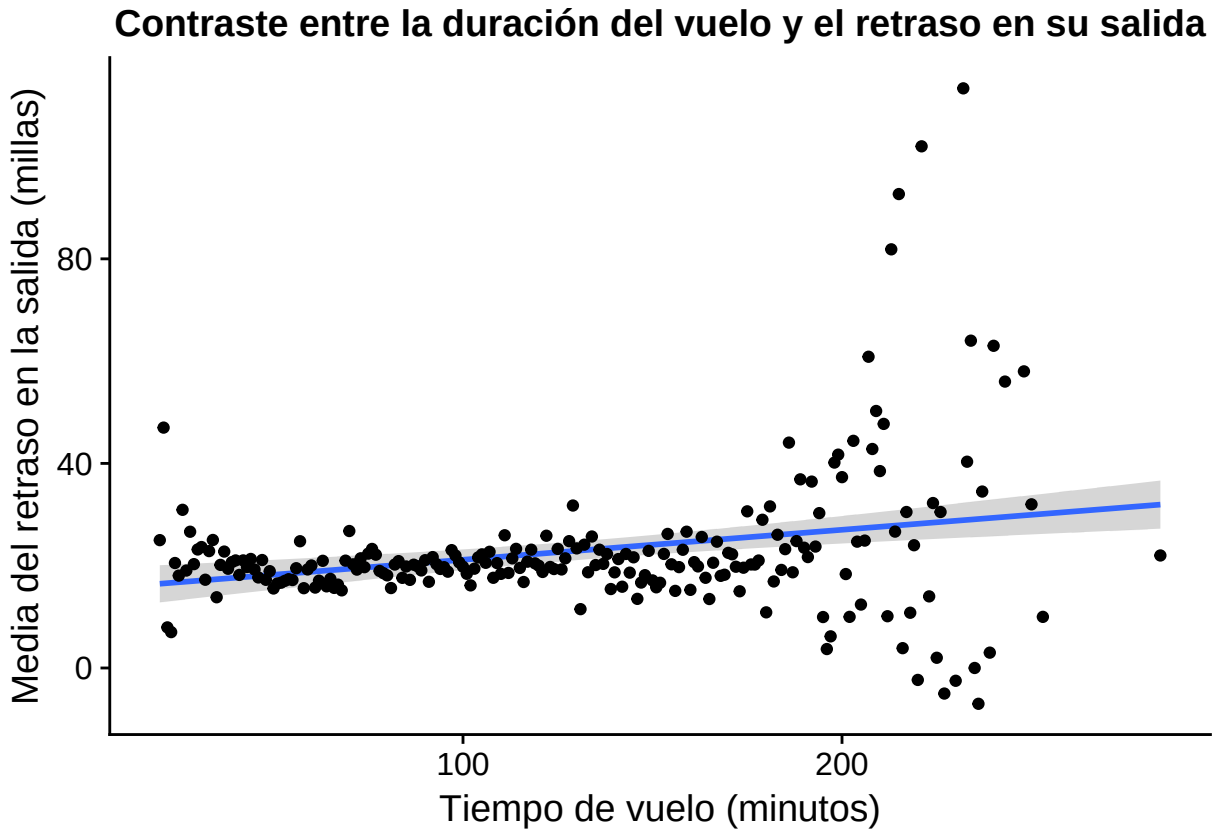


Como se observa tenemos dos puntos muy alejados del resto de la gráfica y una tendencia al aumento de los retrasos. Para poder realizar un estudio más centrado en la mayoria de puntos, elimino las dos observaciones de mayor retraso.

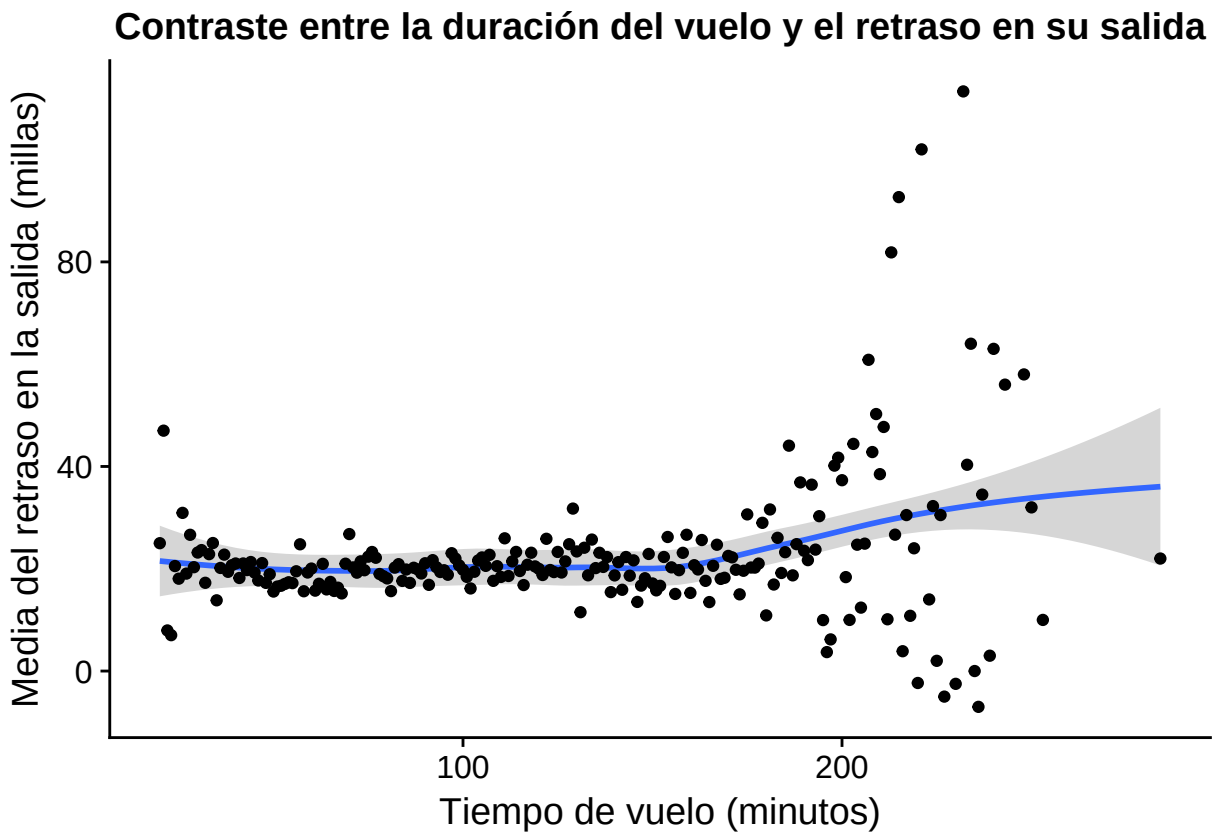
```
flights_contrast3 <- flights_contrast2[(flights_contrast2$mediana_dep_delay<150),]
```

```
ggplot(flights_contrast3, aes(air_time, mediana_dep_delay)) +
 geom_smooth(method = 'gam') +
geom_point()+
 xlab("Tiempo de vuelo (minutos)") +
 ylab("Media del retraso en la salida (millas)") +
```

```
  ggtitle("Contraste entre la duración del vuelo y el retraso en su salida") +

  theme(plot.title = element_text(hjust = 0.5))
```




```
ggplot(flights_contrast3, aes(air_time, mediana_dep_delay)) +
 geom_smooth(method = 'loess') +
geom_point()+
 xlab("Tiempo de vuelo (minutos)") +
 ylab("Media del retraso en la salida (millas)") +

  ggtitle("Contraste entre la duración del vuelo y el retraso en su salida") +

  theme(plot.title = element_text(hjust = 0.5))
```



0.3.8.4 Conclusiones

Lo primero de todo es explicar que tipo de ajuste hacen sobre los datos sendos modelos, *loess* y *gam*. El modelo *gam* en este caso nos realizará un ajuste lineal, en el caso del modelo *loess* nos realiza un ajuste no lineal de tipo local.

Las conclusiones que podemos sacar de estas gráficas son varias, el modélo *gam* nos devuelve una recta de ajuste con cierta tendencia positiva que se ajusta de una manera adecuada a tiempos de vuelo inferiores a 200 minutos de vuelo, sin embargo a partir de este valor encontramos datos dispersos en ambas direcciones pero con mucah mayor fuerza en el sentido correpondiente al aumento de los retrasos. En el caso de hacer el mismo estudio con este modelo incluyendo los puntos eliminados la recta de ajuste se moveria en tendencia positiva con mayor potencia. Debido a esta distribución de puntos no se puede asegurar que a partir de 200 minutos de vuelo el ajuste sea correcto, pero esta claro que la tendencia positiva existe.

En el caso de *loess* como vemos hasta el valor de 200 minutos de vuelo, nos devuelve un resultado no muy alejado del ofrecido por el anterior estudio, por tanto, es acertado ese ajuste creciente hasta la zona de 200 con esa tendencia. Comparando ambas representaciones hasta este punto el modelo que mejor se ajusta es *loess*. El valor de 200 minutos de vuelo es un punto de inflexión debido a que a partir de el la tendencia cambia, y como en el caso anterior

este método tampoco nos ofrece un resultado correcto, pero la tendencia positiva existe.

En conclusión, si se puede decir que existe cierta correlación entre ambas variables, aunque es cierto que se pueden encontrar vuelos con menor retraso en tiempos superiores a 200 minutos, el valor global de los retrasos en esa zona es claramente positivo.

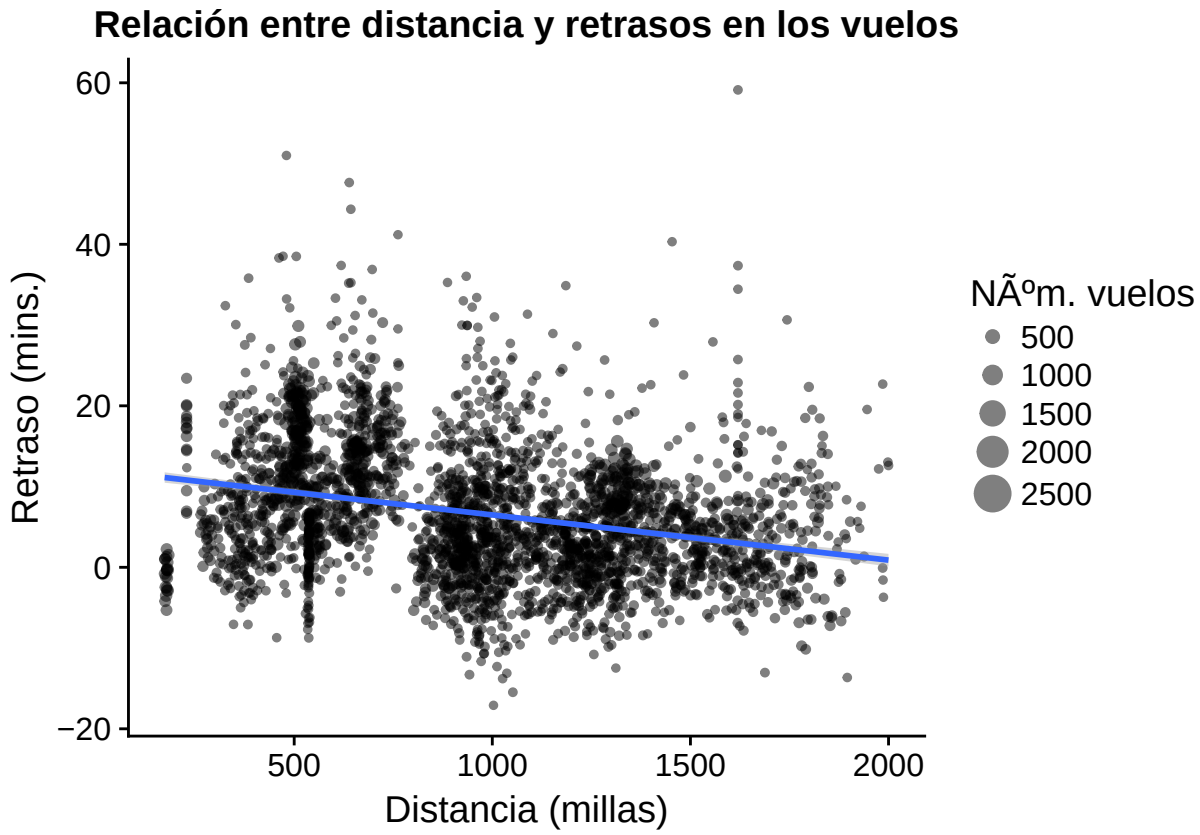
En principio no tendrian porque estar relacionados el retraso en el despegue con la duración del vuelo, pero este pensamiento ha sido rechazado por el análisis de datos llevado a cabo. En cuanto a los vuelos que salen del aeropuerto antes de tiempo, esta es una situación que parece poco probable y tal y como muestran los datos en el estudio en este campo si se corrobora la tesis inicial, esta situación solo está reguistrada en un puñado de observaciones cuyo tiempo de vuelo supera los 200 minutos.

0.4 Ejercicio 2 (*30 puntos*)

La siguiente figura, tomada de la introducción a dplyr, muestra un gráfico en `ggplot2` de la relación entre distancia de los vuelos y retraso experimentado para todos los aeropuertos de NYC.

```
by_tailnum <- group_by(flights, tailnum)
delay <- summarise(by_tailnum,
  count = n(),
  dist = mean(distance, na.rm = TRUE),
  delay = mean(arr_delay, na.rm = TRUE))
delay <- filter(delay, count > 20, dist < 2000)

# Interestingly, the average delay is only slightly related to the
# average distance flown by a plane.
ggplot(delay, aes(dist, delay)) +

  geom_point(aes(size = count), alpha = 1/2) +

  labs(x="Distancia (millas)", y="Retraso (mins.)") +

  geom_smooth(method = 'gam') +

  scale_size_area() +

  ggtitle("Relación entre distancia y retrasos en los vuelos") +

  scale_radius(name="NÃºm. vuelos")
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```



A la vista del resultado, parece que exista una cierta correlación negativa, aunque no muy fuerte, entre ambas variables. Sin embargo, veamos que sucede si desglosamos los datos utilizando otras variables disponibles.

En este ejercicio, se propone **representar el retraso de llegadas en función de la distancia recorrida**, utilizando una gráfica como la anterior, pero desglosado por meses (es decir, una gráfica como la anterior para cada mes).

La solución óptima debería construir un panel de 12 gráficas, una para cada mes. Cada gráfica se debe etiquetar con el nombre abreviado de ese mes, no con el nÃºmero de mes. Además, se debe presentar las gráficas en el orden correcto de los meses del calendario (primero el gráfico de enero, luego febrero, etc.), no por orden alfabético de los nombres del mes.

¿Qué conclusiones puedes extraer a la vista de estos gráficos? Intenta ofrecer argumentos basados en los resultados obtenidos para elaborar la respuesta.

0.4.1 Limpieza de datos

Como se ha comentado anteriormente el data.frame *fligh*s contiene *NA* que debemos eliminar para un correcto estudio de la situación. El comando se va ejecutar en el apartado siguiente en conjunto con el tratamiento de datos.

0.4.2 Tratamiento de datos

Antes de pasar a la visualización que es el fin de este ejercicio, se lleva a cabo un tratamiento de datos adecuado. Dividimos el año en los meses a estudio, lo agrupamos por las variables a estudio y por último eliminamos los *NA* para calcular *dist* y *delay* tal y como se muestra en el enunciado.

```
by_tailnum <- group_by(flights, tailnum, month)
delay <- summarise(by_tailnum,
                   count = n(),
                   dist = mean(distance, na.rm = TRUE),
                   delay = mean(arr_delay, na.rm = TRUE),)
```

```
delay1 <- delay %>%
           filter(month==1, count > 20, dist < 2000)


delay2 <- delay %>%
           filter(month==2, count > 20, dist < 2000)


delay3 <- delay %>%
           filter(month==3, count > 20, dist < 2000)


delay4 <- delay %>%
           filter(month==4, count > 20, dist < 2000)


delay5 <- delay %>%
           filter(month==5, count > 20, dist < 2000)


delay6 <- delay %>%
           filter(month==6, count > 20, dist < 2000)
```

```
delay7 <- delay %>%
           filter(month==7, count > 20, dist < 2000)


delay8 <- delay %>%
           filter(month==8, count > 20, dist < 2000)


delay9 <- delay %>%
           filter(month==9, count > 20, dist < 2000)


delay10 <- delay %>%
            filter(month==10, count > 20, dist < 2000)


delay11 <- delay %>%
            filter(month==11, count > 20, dist < 2000)


delay12 <- delay %>%
            filter(month==12, count > 20, dist < 2000)
```

0.4.3 Visualización

Se representan los conjuntos creados mediante el paquete *ggplot2* y por último para obtener la gráfica de los 12 meses se hará uso de *cowplot* junto con la función *plot_grid* incluida en el.

Cumpliendo con las peticiones del enunciado se representan con el título que esta compuesto por una abreviatura de su nombre completo usando solo las dos primeras letras, en caso de que coincidan se añadirá la tercera para evitar confusiones.

```
En <-  ggplot(delay1, aes(dist, delay)) +
  geom_point(aes(size = count), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada (mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("En") +
```

```
  scale_radius(name="Núm. vuelos")
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

```
Fe <-  ggplot(delay2, aes(dist, delay)) +
  geom_point(aes(size = count), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada (mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("Fe") +
  scale_radius(name="Núm. vuelos")
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

```
Mar <-  ggplot(delay3, aes(dist, delay)) +
  geom_point(aes(size = count), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada (mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("Mar") +
  scale_radius(name="Núm. vuelos")
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

```
Ab <-  ggplot(delay4, aes(dist, delay)) +
  geom_point(aes(size = count), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada (mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("Ab") +
  scale_radius(name="Núm. vuelos")
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

```
May <-  ggplot(delay5, aes(dist, delay)) +
  geom_point(aes(size = count), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada (mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("May") +
```

```
  scale_radius(name="Núm. vuelos")
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

```
Jun <-  ggplot(delay6, aes(dist, delay)) +
  geom_point(aes(size = count), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada(mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("Jun") +
  scale_radius(name="Núm. vuelos")
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

```
Jul <-  ggplot(delay7, aes(dist, delay)) +
  geom_point(aes(size = count), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada(mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("Jul") +
  scale_radius(name="Núm. vuelos")
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

```
Ag <-  ggplot(delay8, aes(dist, delay)) +
  geom_point(aes(size = count), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada (mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("Ag") +
  scale_radius(name="Núm. vuelos")
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

```
Se <-  ggplot(delay9, aes(dist, delay)) +
  geom_point(aes(size = count), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada(mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("Se") +
```

```
  scale_radius(name="Núm. vuelos")
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

```
Oc <-  ggplot(delay10, aes(dist, delay)) +
  geom_point(aes(size = count), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada (mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("Oc") +
  scale_radius(name="Núm. vuelos")
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

```
No <-  ggplot(delay11, aes(dist, delay)) +
  geom_point(aes(size = count), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada(mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("No") +
  scale_radius(name="Núm. vuelos")
```

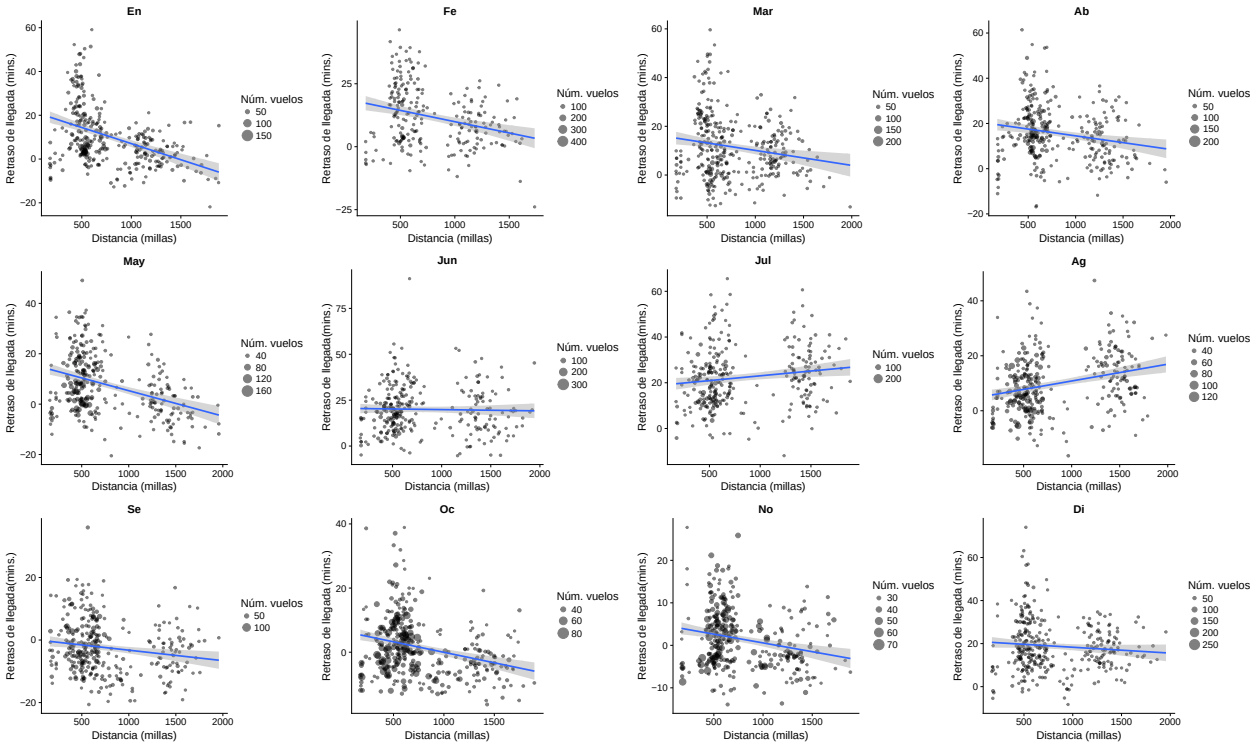
```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

```
Di<-  ggplot(delay12, aes(dist, delay)) +
  geom_point(aes(size = count), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada (mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("Di") +
  scale_radius(name="Núm. vuelos")
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

Una vez creadas las graficas las representamos juntas tal y como se demanda.

```
graficafin <- plot_grid (En, Fe, Mar, Ab, May, Jun, Jul, Ag, Se, Oc, No, Di)

graficafin
```



Lo primero que se puede resaltar de los datos es el corte que se aprecia en la mayoria de los meses, se detecta una franja de valores de distancia en la cual no tenemos ningún dato o muy pocos. Además entre estos dos conjuntos de datos encontramos diferentes comportamientos y a lo largo de los meses se encuentran cierta correpondencia. Debido a este hecho dividimos el conjunto en vuelos grupo A y grupo B, en este último se encuentran los vuelos de mayor distancia. Durante los meses la franja de separación en millas no es constante, por esta razón, prefiero elegir esta denominación grupos A y B que una más ilustrativa como vuelos de larga y corta distancia, ya que para algunos meses un vuelo entraría en una categoria y para otros en la contraria.

Podríamos estar hablando de dos clusters pero no se puede asegurar sin un desarrollo matemático. Para hacer más visible esta carazteristica se ha realizado en la gráfica siguiente la misma representación, pero se han utilizado colores para diferenciar los conjuntos de datos de la siguiente manera: rojo grupo A y verde grupo B.

```
En2 <-  ggplot(delay1, aes(dist, delay)) +
geom_point(aes(size = count, colour= cut(dist,c( 0,770, 2000))), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada(mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("En") +
  scale_radius(name="Núm. vuelos")+
  scale_color_manual(name = "distancia",
```

```
                    breaks = c("770","771"), values = c("red", "green"),
                    labels = c("A", "B"))
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

```
Fe2 <-  ggplot(delay2, aes(dist, delay)) +
geom_point(aes(size = count, colour= cut(dist,c( 0, 770, 2000))), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada (mins.)") +geom_smooth(method = 'ga
  scale_size_area() +
  ggtitle("Fe") +
  scale_radius(name="Núm. vuelos")+
  scale_color_manual(name = "distancia",
                    breaks = c("770","771"), values = c("red", "green"),
                    labels = c("A", "B"))
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

```
Mar2 <-  ggplot(delay3, aes(dist, delay)) +
  geom_point(aes(size = count, colour= cut(dist,c( 0,945, 2000))), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada(mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("Mar") +
  scale_radius(name="Núm. vuelos")+
  scale_color_manual(name = "distancia",
                    breaks = c("945","946"), values = c("red", "green"),
                    labels = c("A", "B"))
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

```
Ab2 <-  ggplot(delay4, aes(dist, delay)) +
  geom_point(aes(size = count, colour= cut(dist,c( 0,980,2000))), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada(mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("Ab") +
  scale_radius(name="Núm. vuelos")+
  scale_color_manual(name = "distancia",
                    breaks = c("980","981"), values = c("red", "green"),
                    labels = c("A", "B"))
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

```
May2 <-  ggplot(delay5, aes(dist, delay)) +
  geom_point(aes(size = count, colour= cut(dist,c( 0,1000,2000))), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada (mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("May") +
  scale_radius(name="Núm. vuelos")+
  scale_color_manual(name = "distancia",
                     breaks = c("1000","1001"), values = c("red", "green"),
                     labels = c("A", "B"))
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

```
Jun2 <-  ggplot(delay6, aes(dist, delay)) +
  geom_point(aes(size = count, colour= cut(dist,c( 0,980,2000))), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada (mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("Jun") +
  scale_radius(name="Núm. vuelos")+
  scale_color_manual(name = "distancia",
                     breaks = c("980","981"), values = c("red", "green"),
                     labels = c("A", "B"))
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

```
Jul2 <-  ggplot(delay7, aes(dist, delay)) +
  geom_point(aes(size = count, colour= cut(dist,c( 0,1000,2000))), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada (mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("Jul") +
  scale_radius(name="Núm. vuelos")+
  scale_color_manual(name = "distancia",
                     breaks = c("1000","1001"), values = c("red", "green"),
                     labels = c("A", "B"))
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
```

```
## which will replace the existing scale.
```

```
Ag2 <-  ggplot(delay8, aes(dist, delay)) +
  geom_point(aes(size = count, colour= cut(dist,c( 0,1000,2000))), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada(mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("Ag") +
  scale_radius(name="Núm. vuelos")+
  scale_color_manual(name = "distancia",
                     breaks = c("1000","1001"), values = c("red", "green"),
                     labels = c("A", "B"))
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

```
Se2 <-  ggplot(delay9, aes(dist, delay)) +
  geom_point(aes(size = count, colour= cut(dist,c( 0,1000,2000))), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada(mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("Se") +
  scale_radius(name="Núm. vuelos")+
  scale_color_manual(name = "distancia",
                     breaks = c("1000","1001"), values = c("red", "green"),
                     labels = c("A", "B"))
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

```
Oc2 <-  ggplot(delay10, aes(dist, delay)) +
  geom_point(aes(size = count, colour= cut(dist,c( 0,780,2000))), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada(mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("Oc") +
  scale_radius(name="Núm. vuelos")+
  scale_color_manual(name = "distancia",
                     breaks = c("780","781"), values = c("red", "green"),
                     labels = c("A", "B"))
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

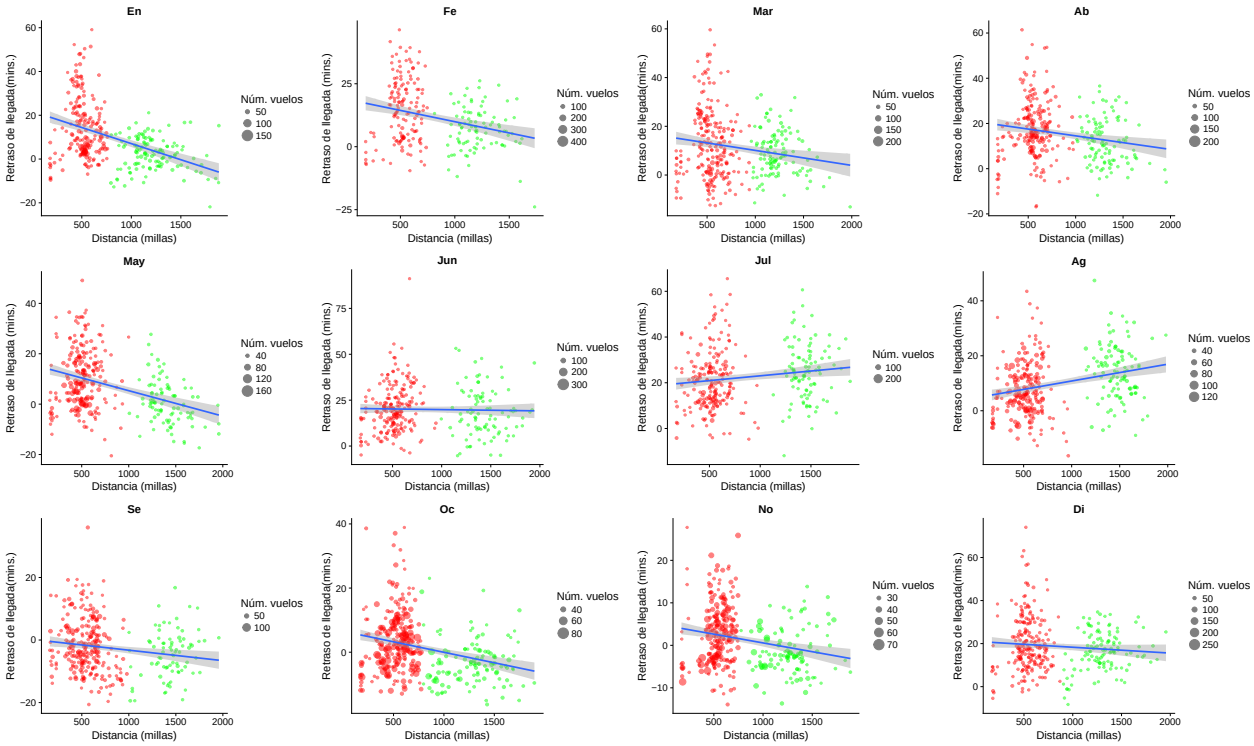
```
No2 <-  ggplot(delay11, aes(dist, delay)) +
  geom_point(aes(size = count, colour= cut(dist,c( 0,750, 2000))), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada (mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("No") +
  scale_radius(name="Núm. vuelos")+
  scale_color_manual(name = "distancia",
                     breaks = c("750","751"), values = c("red", "green"),
                     labels = c("A", "B"))
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

```
Di2 <-  ggplot(delay12, aes(dist, delay)) +
  geom_point(aes(size = count, colour= cut(dist,c( 0,900,2000))), alpha = 1/2) +
  labs(x="Distancia (millas)", y="Retraso de llegada(mins.)") +
  geom_smooth(method = 'gam') +
  scale_size_area() +
  ggtitle("Di") +
  scale_radius(name="Núm. vuelos")+
  scale_color_manual(name = "distancia",
                     breaks = c("900","901"), values = c("red", "green"),
                     labels = c("A", "B"))
```

```
## Scale for 'size' is already present. Adding another scale for 'size',
## which will replace the existing scale.
```

```
graficafin2 <- plot_grid(En2, Fe2, Mar2, Ab2, May2, Jun2, Jul2, Ag2, Se2, Oc2, No2, Di2)

graficafin2
```



0.4.4 Conclusiones

El motivo de realizar esta separación es debido a que encontramos un cierto patrón en estos datos. Los puntos rojos de la gráfica correspondiente a valores más altos de acumulación de retraso y en la mayoria de graficas según nos desplazamos hacia la zona verde la distribución de puntos tiende a alcanzar el valor de retraso cercano a cero.

En la gráfica anual se obtiene este comportamiento, vamos a entrar ahora a discutir los detalles una vez explicado el porque de la separación.

La tendencia de la gráfica anual marca una disminución de los retrasos en la llegada de los vuelos según aumentamos la logitud de los vuelos, en las gráficas mensuales no todas siguen esta tendecia:

- Obtenemos tendencias similares en los meses de enero, febrero, marzo, abril, mayo, octubre y noviembre. Todas ellas cumplen la condición de disminución en retraso de llegada frente al aumento de millas de vuelo. De estas gráficas cabe destacar que en todas ellas los valores de máximos retrasos se encuentran en la zona roja, alcanzando máximos en los meses de enero, marzo y abril donde el tiempo de retraso llega a valores cercanos a 60 minutos. Además tenemos una carazterística común en estos meses, salvo en octubre y noviembre, donde la diferecia entre los grupos A y B no alcanza un valor muy elevado debido a que de partida estos meses tienen retrasos muy pequeños y la delta de tiempo entre máximo y mínimo marca un valor cercano a los 10 minutos, el

resto de meses encontramos un valor inicial de retrasos en conjunto de 20 minutos, al ir aumentando las millas lo que vemos es que ese valor disminuye drásticamente alcanzando valores de retrasos cercanos a cero. El mes de mayor variación es enero donde empezamos con un retraso de 20 minutos y alcanzamos valores en 2000 millas donde el avión llegó antes de tiempo al aeropuerto. Por tanto, este conjunto de meses se puede ver como una tendencia a disminución del retraso de llegada y valores cercanos a la puntualidad segun nos acercamos a la barrera de las 2000 millas.

- El siguiente grupo de meses atendiendo a la tendencia, esta formado por los meses de junio, septiembre y diciembre. En estos meses la tendencia de tiempos es casi horizontal, no tenemos gran diferencia entre los retrasos de 0 a 2000 millas, pero a su vez debido al valor de estos, se pueden realizar dos subgrupos. Junio y diciembre tiene retrasos de valores 25 y 20 minutos respectivamente y son constantes a lo largo de la distancia, para septiembre cambia la situción este se encuentra en una zona de retrasos cero durante todo el espectro. En estos meses la separación en grupos A y B trae leves diferencias respecto al caso anterior, aunque la tendencia es casi constante a lo largo del año los máximos de retraso se encuentran en el grupo A.

- Por último los meses con tendencia al aumento de los retrasos según aumentan las millas de vuelo. Este grupo lo forman unicamente los meses de julio y agosto, rompiendo la tendencia anual. Julio comienza con valores de 20 minutos de retraso y agosto valores cercanos a cero. Los grupos A y B estan claramente separados aunque tienen una ditribución de puntos casi identica. Cabe destacar que se rompe un tónica habitual en este estudio, en el mes de agosto el máximo retraso se encuentra dentro del grupo B.

Por tanto atendiendo la distribución de los meses se puede decir que cumplen en computo general la tendencia anual aunque no por separado. Un modelo lineal como el llevado a cabo en este ejercicio nos devuleve un resultado en tendencia acertado, pero la recta no se ajusta a los datos. El último factor que queda por discutir de la gráfica es la distribución de los vuelos. Debido a la elección de la representación, a simple vista la mayoria de los vuelos se encuentran en el grupo A y cuando tenemos grandes acumulaciones en el grupo B estan en valores de poco retraso de llegada.

Antes de comenzar el análisis, era de esperar que los aviones que recorren mayor distancia tuvieran mayor retraso, sin embargo nos encontramos en la posición contraria. Una explicación posible para este fenómeno podria ser que los aviones de ambos grupos tengan retrasos ya en la salida de los mismos, en esta situación los vuelos cortos no tendrian tiempo de vuelo suficiente para compensar este retraso inicial, pero si los de larga distancia.

0.5 Ejercicio 3 (*20 puntos*)

Representar el retrasos de salida de los vuelos que parten del aeropuerto JFK (código 'JFK'), desglosado por meses (como en el ejercicio anterior). Se mostrarán solo los vuelos domésticos, imponiendo como condición de filtrado de datos: `distancia recorrida < 1.000 millas`.

¿Qué conclusiones puedes extraer a la vista de estos gráficos?

0.5.1 Limpieza de datos

Como se ha comentado anteriormente el data.frame *fligh*s contiene *NA* que debemos eliminar para un correcto estudio de la situación. El comando se va ejecutar en el apartado siguiente en conjunto con el tratamiento de datos.

0.5.2 Tratamiento de datos

Se realiza el tratamiento de datos acorde con las especificaciones, filtraremos el aeropuerto *JFK* como única salida de vuelos y distancias menores a 1000 millas.

Como en situaciones anteriores sacamos la definición de *by_dist* de la expresión principal *delay_ej3* para mejorar el seguimiento y legibilidad del código.

```
by_dist <- flights %>%
            filter(origin=='JFK') %>%
             group_by( month,tailnum)

delay_ej3 <- summarise(by_dist,
                   count = n(),
                   dist = mean(distance, na.rm = TRUE),
                   delay = mean(dep_delay, na.rm = TRUE))
```

Filtramos por los distintos meses con las condiciones impuestas en el enunciado.

```
delay_anu <- delay_ej3 %>%
              filter(count > 20, dist < 1000)

delay01 <- delay_ej3 %>%
           filter(count > 20, dist < 1000, month==1)

delay02 <- delay_ej3 %>%
            filter(count > 20, dist < 1000, month==2)
```

```
delay03 <- delay_ej3 %>%
            filter(count > 20, dist < 1000, month==3)

delay04 <- delay_ej3 %>%
            filter(count > 20, dist < 1000, month==4)

delay05 <- delay_ej3 %>%
            filter(count > 20, dist < 1000, month==5)

delay06 <- delay_ej3 %>%
            filter(count > 20, dist < 1000, month==6)

delay07 <- delay_ej3 %>%
            filter(count > 20, dist < 1000, month==7)

delay08 <- delay_ej3 %>%
            filter(count > 20, dist < 1000, month==8)

delay09 <- delay_ej3 %>%
            filter(count > 20, dist < 1000, month==9)

delay10 <- delay_ej3 %>%
            filter(count > 20, dist < 1000, month==10)

delay11 <- delay_ej3 %>%
            filter(count > 20, dist < 1000, month==11)

delay12 <- delay_ej3 %>%
            filter(count > 20, dist < 1000, month==12)

# Interestingly, the average delay is only slightly related to the
# average distance flown by a plane.
```

0.5.3 Visualización

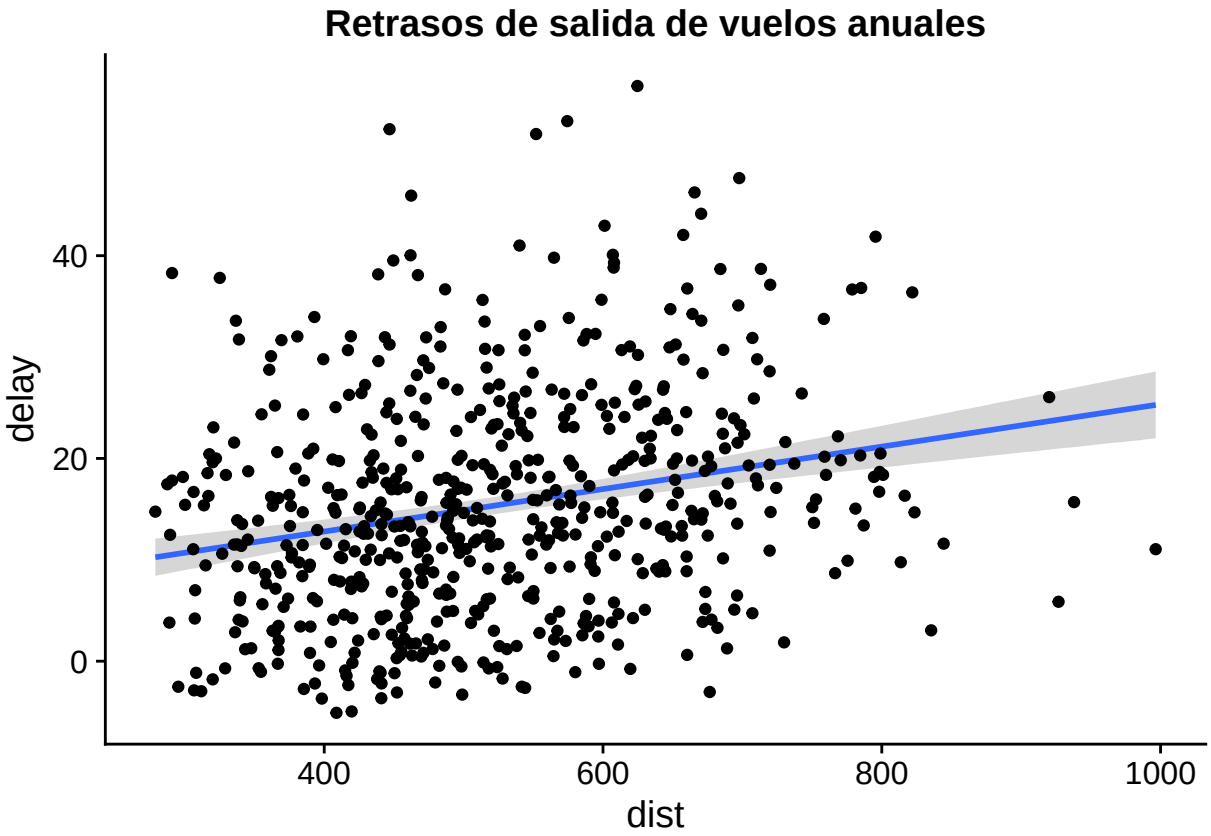
Antes de presentar las graficas mensuales con el fin de poder realizar un mejor análisis, se representa la gráfica anual.

```
anualej3delay <- ggplot(delay_anu, aes(dist, delay)) +
  geom_smooth(method="gam")+
```

```
  geom_point() +
  ggtitle("Retrasos de salida de vuelos anuales")
  labs(x="Distancia(millas)", y="Retraso de salida(minutos)")
```

```
## $x
## [1] "Distancia(millas)"
##
## $y
## [1] "Retraso de salida(minutos)"
##
## attr(,"class")
## [1] "labels"
```

```
anualej3delay
```



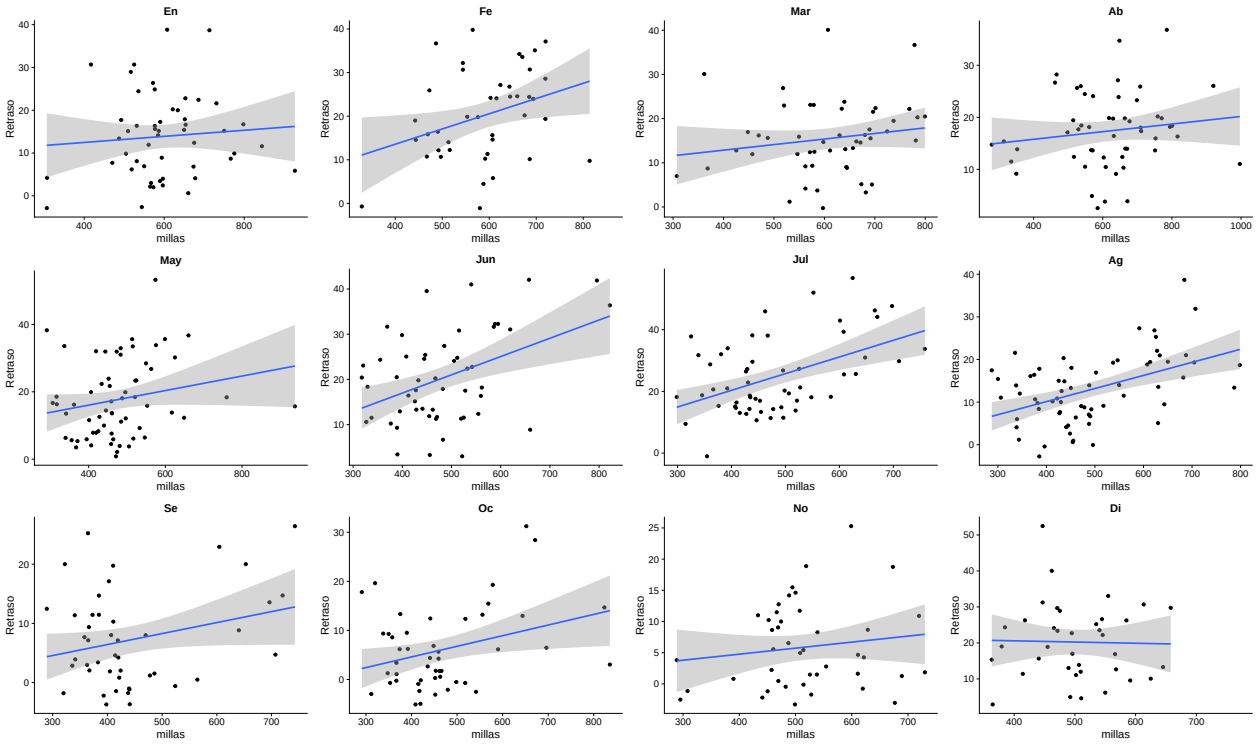
Nos encontramos con una distribución de puntos anual con pocos valores por debajo de cero, es decir, pocos aviones salieron antes de tiempo del aeropuerto. También cabe destacar que los valores altos de retrasos basándonos en esta gráfica se observan que no son atípicos. Sin embargo, no encontramos muchos vuelos largos, por tanto, los valores altos de distancia habra que tratarlos con cuidado.

```
prot_delay01 <- ggplot(delay01, aes(dist, delay)) +
  geom_point() +
  labs(x="millas", y="Retraso") +
  geom_smooth(method = 'gam')+
  ggtitle("En")

prot_delay02 <- ggplot(delay02, aes(dist, delay)) +
  geom_point() +
  labs(x="millas", y="Retraso") +
  geom_smooth(method = 'gam')+
  ggtitle("Fe")

prot_delay03 <- ggplot(delay03, aes(dist, delay)) +
  geom_point() +
  labs(x="millas", y="Retraso") +
  geom_smooth(method = 'gam')+
  ggtitle("Mar")

prot_delay04 <- ggplot(delay04, aes(dist, delay)) +
  geom_point() +
  labs(x="millas", y="Retraso") +
  geom_smooth(method = 'gam')+
  ggtitle("Ab")

prot_delay05 <- ggplot(delay05, aes(dist, delay)) +
  geom_point() +
  labs(x="millas", y="Retraso") +
  geom_smooth(method = 'gam')+
  ggtitle("May")

prot_delay06 <- ggplot(delay06, aes(dist, delay)) +
  geom_point() +
  labs(x="millas", y="Retraso") +
  geom_smooth(method = 'gam')+
  ggtitle("Jun")

prot_delay07 <- ggplot(delay07, aes(dist, delay)) +
  geom_point() +
  labs(x="millas", y="Retraso") +
  geom_smooth(method = 'gam')+
```

```
  ggtitle("Jul")

prot_delay08 <- ggplot(delay08, aes(dist, delay)) +
  geom_point() +
  labs(x="millas", y="Retraso") +
  geom_smooth(method = 'gam')+
  ggtitle("Ag")

prot_delay09 <- ggplot(delay09, aes(dist, delay)) +
  geom_point() +
  labs(x="millas", y="Retraso") +
  geom_smooth(method = 'gam')+
  ggtitle("Se")

prot_delay10 <- ggplot(delay10, aes(dist, delay)) +
  geom_point() +
  labs(x="millas", y="Retraso") +
  geom_smooth(method = 'gam')+
  ggtitle("Oc")

prot_delay11 <- ggplot(delay11, aes(dist, delay)) +
  geom_point() +
  labs(x="millas", y="Retraso") +
  geom_smooth(method = 'gam') +
  ggtitle("No")

prot_delay12 <- ggplot(delay12, aes(dist, delay)) +
  geom_point() +
  labs(x="millas", y="Retraso") +
  geom_smooth(method = 'gam')+
  ggtitle("Di")
```

De nuevo siguiendo las especificaciones del enunciado, representamos los meses por orden. Como abreviaturas se ha elegido el sistema de nombrar por las dos primeras letras de cada mes, debido a que ciertos meses comparten esas letras para evitar confusiones en estos casos utilizaremos las tres primeras. Utilizamos para esclarecer la situación el modelo *gam* como en el ejercicio anterior.

```
final_ejer3 <- plot_grid(prot_delay01,prot_delay02 ,prot_delay03 ,prot_delay04 ,prot_del

final_ejer3
```



0.5.4 Conclusiones

A primera vista se puede observar la dispersión de las medidas y la distinta distancia de viajes según el mes a estudio, para vuelos de igual distancia podemos encontrar grandes diferencias de tiempo. Hasta 800 millas la concentración es mucho mayor que cuando pasamos a distancias mayores. Esto se puede explicar debido a que no todos los meses tienen medidas por encima de las 800 millas, mostrandose en esta cualidad un mes por encima del resto: en el mes de diciembre no alcanzamos ningún valor de 700 millas. Solo alcanzamos un valor de 1000 millas en el mes de abril, este comportamiento ya se vio en la gráfica anual.

En todas las gráficas tenemos puntos dispersos y con diferente concentración si nos fijamos en las millas. Debido a esta lejania de los puntos y la distinta concentración de observaciones implica que varia mucho la recta de ajuste del modelo *gam*, pero como en la gráfica anual no se detectaban valores atípicos es correcto aplicar bajo este razonamiento a todos los puntos de la gráfica. Aunque el modelo de ajuste lineal no se ajusta bien a las observaciones, nos brindará cierta información sobre la tendencia.

En general si hablamos de tendencia, existe en la grafica anual hacia el aumento del retraso de salida según aumenta la distancia, con una delta entre máximo y mínimo cercana a los 10 minutos. Tomando los resultados del modelo *gam* por meses, esta tendencia se ve reflejada en la mayoria de meses salvo en diciembre donde es una recta horizontal.

Dado que estamos hablando de tiempos de salida, a priori no debería reguistrarse ninguna

diferencia en funcion de las millas recorridas. En las gráficas obtenidas si se observa cierta diferencia aunque comparando la tendencia en la grafica anual y mensual, la tendencia mensual en muchos meses es más marcada.

Por tanto, podemos concluir que este aeropuerto tiene retrasos de salida durante todo el año que pueden variar entre valores cercanos a cero y alcanzando máximos cercanos a los cuarenta minutos. El estudio nos dice que los meses que reguistran menores retrasos de salida son septiembre y octubre en vuelos menores a 500 millas y los máximos en julio con valores muy altos por encima de las 600 millas. El máximo absoluto se encuentra en diciembre con una distancia menor a 500 millas y un retraso de salida superior a los 50 minutos.

0.6 Ejercicio 4 (*20 puntos*)

Utilizando boxplots (`geom_boxplot`), representar gráficamente una comparativa de los retrasos de salida entre las distintas compañías aéreas, en el mes de diciembre, para el aeropuerto de Newark (código 'EWR'). ¿Se observan diferencias notables?

0.6.1 Limpieza de datos

Realizamos una limpieza de datos acorde con el problema, centrada en la variable a estudio en este apartado.

```
flights4 <- flights %>%
             filter(is.na(dep_delay) == FALSE)
```

0.6.2 Tratamiento de datos

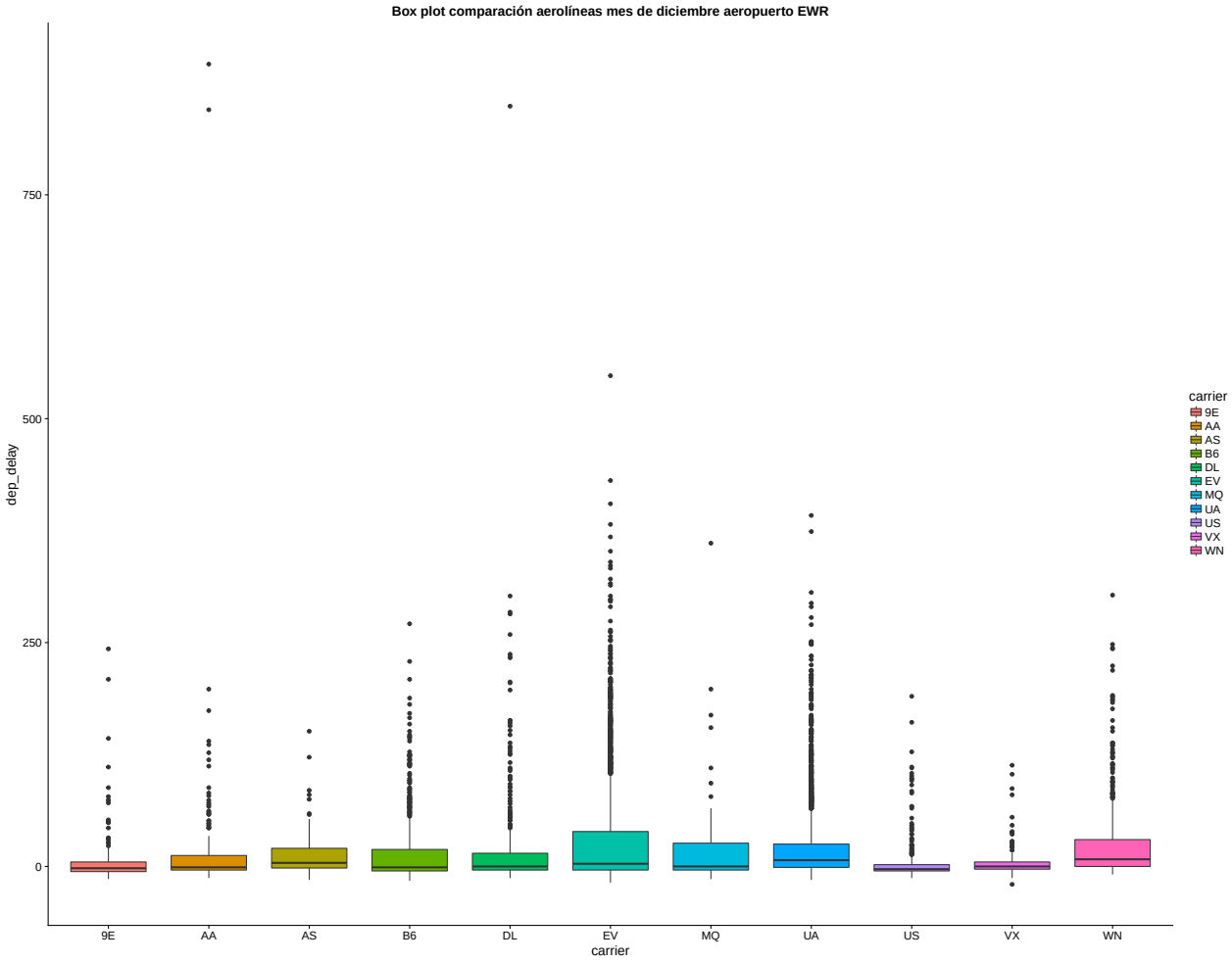
Filtramos la condición del mes de diciembre y del aeropuerto de origen *EWR*.

```
by_dic <- flights4%>%
           filter(month==12 & origin=='EWR' )
```

0.6.3 Visualización

Construimos un gráfico *box_plot*, con el código de colores explicado en la leyenda del gráfico.

```
ggplot(by_dic, aes(x=carrier, y=dep_delay, fill=carrier)) +
 geom_boxplot()+
  guides("FALSE")+
  ggtitle("Box plot comparación aerolíneas mes de diciembre aeropuerto EWR")
```



Antes de pasar a sacar conclusiones de esta representación gráfica se explicará el gráfico para una correcta interpretación.

La caja representa los datos entre el primer cuartil y el tercero, la línea que divide la caja muestra la mediana. Por debajo de la caja encontramos los datos inferiores al primer cuartil y por encima de ella los superiores al tercero. La medición de atípicos se marca de la siguiente forma: por encima es atípico todo aquel que tenga un valor superior a *Q3+1.5(Q3-Q1)* y por debajo *Q1-1.5(Q3-Q1)*, siendo *Q3* y *Q1* tercer y primer cuartil respectivamente. En el entorno gráfico es facil distinguirlo ya que se encuentran por encima o por debajo de las rectas que salen de la caja en ambas direcciones.

0.6.4 Conclusiones

Basándonos en la explicación de la representación explicada en el apartado anterior vemos que tenemos una gran cantidad de valores atípicos en todas las aerolineas de forma positiva y solo aparece en la parte inferior en la aerolínea en *VX*. En cuanto a mayores valores atípicos se encuentran en la aerolínea *DL* y *AA*, los cuales son muy lejanos respecto al resto de

observaciones de las mismas aerolíneas. Si nos centramos en la cuantia de los puntos en el cuarto cuartil la que mayor número presenta es *EV* y el mínimo es *AS*.

Si realizamos un estudio sobre las medianas no se aprecia gran diferencia de valor entre las aerolíneas, sin embargo, si en el tramo *Q3+1,5(Q3-Q1)* en el caso de la aerolínea *EV* es cercano a los 40 minutos y *US* es realmente pequeño. Cabe destacar todas las aerolíneas tienen el primer cuartil del puntos por debajo de cero o muy cercano a este valor.

Por tanto, la aerolínea que más valores atípicos acumula en *EV*, cabe destacar que en ejercicios anteriores se ha enconctrado que esta aerolínea a lo largo del año tambien es la que más retraso de salida acumula. Si hablamos de la aerolínea de menores retrasos de salida a la vista de este gráfico, estaría representada por *VX*. Para concluir, entre aerolíneas aunque los datos en el valor de mediana son muy cercanos la distribución de estos es muy diferente.