Tarea Examen 1

Yimmy Eman

2022-07-09

Pregunta 1

Intenta describir con frases entendibles el conjunto de vuelos retrasados. Intenta dar afirmaciones como por ejemplo:

- Un vuelo tiende a salir unos 20 minutos antes el 50% de las veces y a salir tarde el 50% de las veces restantes.
- Los vuelos de la compañía XX llegan siempre 20 minutos tarde
- El 95% de los vuelos a HNL llegan a tiempo, pero el 5% restante se retrasan más de 3 horas. Intenta dar por lo menos 5 afirmaciones verídicas en base a los datos que tenemos disponibles.

Planteamiento 1:

- El top 3 de los vuelos más lento pertenecen a las compañías US, B6 y 9E.
- El top 3 de los vuelos más rápidos pertenecen a las compañías DL y EV.

```
slow.flights <- group_by(flights,carrier) %>%
  summarise(speed = distance/ air_time * 60) %>%
  arrange(speed)

fast.flights <- group_by(flights,carrier) %>%
  summarise(speed = distance/ air_time * 60) %>%
  arrange(desc(speed))

head(slow.flights, 3)
```

```
## # A tibble: 3 x 2
## # Groups:   carrier [3]
##   carrier speed
##   <chr>   <dbl>
## 1 US       76.8
## 2 B6       84.7
## 3 9E       92.5
```

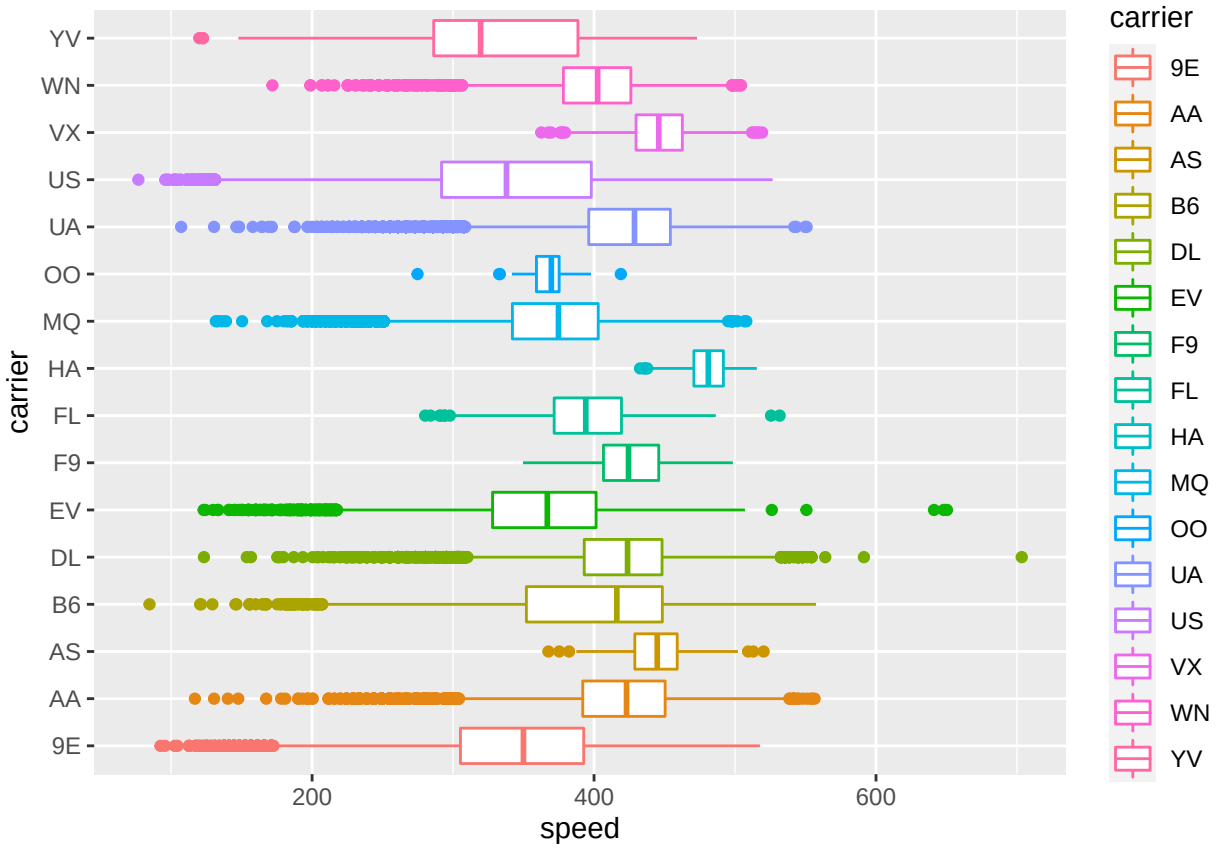
```
head(fast.flights, 3)
```

```
## # A tibble: 3 x 2
## # Groups:   carrier [2]
```

```
##   carrier speed
##   <chr>   <dbl>
## 1 DL       703.
## 2 EV       650.
## 3 EV       648
```

```
ggplot(data = slow.flights, mapping = aes(x = speed, y = carrier, col = carrier)) +
  geom_boxplot()
```

```
## Warning: Removed 9430 rows containing non-finite values (stat_boxplot).
```

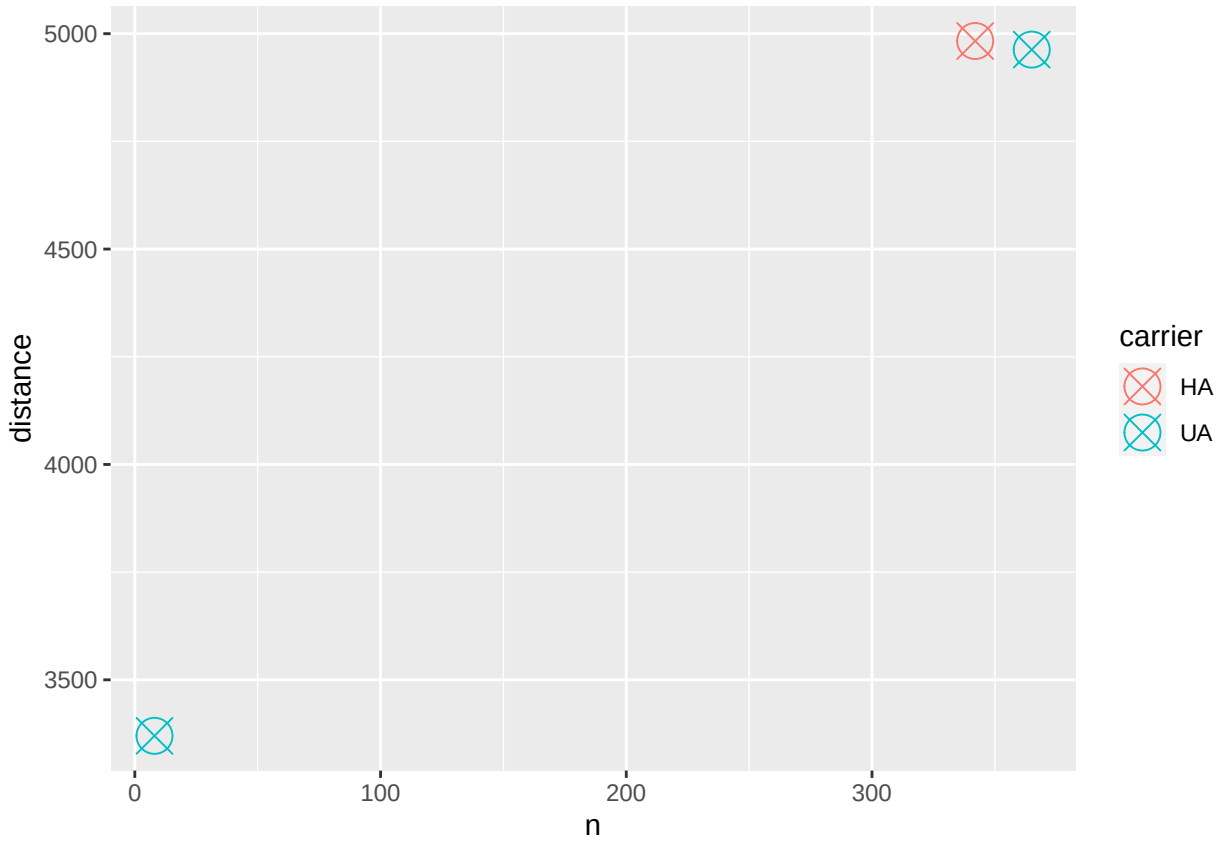


Planteamiento 2:

- El top 3 de los vuelos con mas distancia son de las compañia HA con 4983 Km y 342 vuelos, y la compañias UA con 4963 km con 365 vuelos y 3370 km y 8 vuelos respectivamente.

```
(group_by(flights, carrier, distance) %>%
  summarise(n = n()) %>%
  arrange(desc(distance)))[1:3,] %>%
  ggplot(mapping = aes(x = n, y = distance, type = 21, col = carrier)) +
    geom_point(shape = 13, size = 6)
```

```
## `summarise()` has grouped output by 'carrier'. You can override using the
## `.groups` argument.
```



Pregunta 2

Da una versión equivalente a las pipes siguientes sin usar la función count:

not_cancelled %>% count(dest) not_cancelled %>% count(tailnum, wt = distance)

```
not_cancelled <- flights %>%
  filter(!is.na(dep_delay), !is.na(arr_delay))
```

```
# not_cancelled %>% count(dest)
summarise(group_by(not_cancelled, dest), n())
```

```
## # A tibble: 104 x 2
##    dest  `n()`
##    <chr> <int>
##  1 ABQ     254
##  2 ACK     264
##  3 ALB     418
##  4 ANC       8
##  5 ATL   16837
##  6 AUS    2411
##  7 AVL     261
##  8 BDL     412
##  9 BGR     358
## 10 BHM     269
## # ... with 94 more rows
```

```
# not_cancelled %>% count(tailnum, wt = distance)
summarise(group_by(not_cancelled, tailnum), n = sum(distance))
```

```
## # A tibble: 4,037 x 2
##    tailnum      n
##    <chr>    <dbl>
##  1 D942DN    3418
##  2 N0EGMQ  239143
##  3 N10156  109664
##  4 N102UW   25722
##  5 N103US   24619
##  6 N104UW   24616
##  7 N10575  139903
##  8 N105UW   23618
##  9 N107US   21677
## 10 N108UW   32070
## # ... with 4,027 more rows
```

Pregunta 3

Para definir un vuelo cancelado hemos usado la función

(is.na(dep_delay) | is.na(arr_delay))

Intenta dar una definición que sea mejor, ya que la nuestra es un poco subóptima. ¿Cuál es la columna más importante?

```
cancelled <- flights %>%
  filter(is.na(dep_delay) | is.na(arr_delay))
cancelled
```

```
## # A tibble: 9,430 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
##  1  2013     1     1     1525           1530        -5     1934           1805
##  2  2013     1     1     1528           1459        29     2002           1647
##  3  2013     1     1     1740           1745        -5     2158           2020
##  4  2013     1     1     1807           1738        29     2251           2103
##  5  2013     1     1     1939           1840        59       29           2151
##  6  2013     1     1     1952           1930        22     2358           2207
##  7  2013     1     1     2016           1930        46       NA           2220
##  8  2013     1     1       NA           1630        NA       NA           1815
##  9  2013     1     1       NA           1935        NA       NA           2240
## 10  2013     1     1       NA           1500        NA       NA           1825
## # ... with 9,420 more rows, and 11 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```

```
# Ambas columnas son importantes
```

Pregunta 4

Investiga si existe algún patrón del número de vuelos que se cancelan cada día. Investiga si la proporción de vuelos cancelados está relacionada con el retraso promedio por día en los vuelos. Investiga si la proporción de vuelos cancelados está relacionada con el retraso promedio por aeropuerto en los vuelos. ¿Qué compañía aérea sufre los peores retrasos?

Sol

Los días 8, 9 y 10 de cada mes hay mas cancelaciones de vuelos.

```
cancelled_prop = round(dim(cancelled)[1] / dim(flights)[1]*100,2)
cancelled_prop
```

```
## [1] 2.8
```

```
group_by(cancelled, day) %>%
  summarise(n = n(),
            daily_prop = round(n/sum(day)*100,2),
            mean_dep_delay = mean(dep_delay > 0, na.rm = T)) %>%
  arrange(desc(n))
```

```
## # A tibble: 31 x 4
##      day     n daily_prop mean_dep_delay
##    <int> <int>      <dbl>          <dbl>
##  1     8   963      12.5           0.738
##  2     9   626      11.1           0.545
##  3    10   613      10             0.744
##  4    12   473       8.33          0.763
##  5    23   455       4.35          0.756
##  6    28   379       3.57          0.548
##  7    11   376       9.09          0.424
##  8     7   374      14.3           0.589
##  9    22   360       4.55          0.679
## 10     6   334      16.7           0.526
## # ... with 21 more rows
```

```
# ¿Qué compañía aérea sufre los peores retrasos?
# R: HA y MQ
(select(flights, carrier, dep_delay) %>%
  arrange(desc(dep_delay)))[1:3,]
```

```
## # A tibble: 3 x 2
##   carrier dep_delay
##   <chr>       <dbl>
## 1 HA           1301
## 2 MQ           1137
## 3 MQ           1126
```

Pregunta 5

Difícil: Intenta desentrañar los efectos que producen los retrasos por culpa de malos aeropuertos vs malas compañías aéreas. Por ejemplo, intenta usar flights %>% group_by(carrier, dest) %>% summarise(n())

```
flights %>%
  group_by(carrier, dest) %>%
  summarise(n = n()) %>%
  arrange(desc(n))
```

```
## `summarise()` has grouped output by 'carrier'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 314 x 3
## # Groups:   carrier [16]
##    carrier dest      n
##    <chr>   <chr> <int>
##  1 DL      ATL   10571
##  2 US      CLT    8632
##  3 AA      DFW    7257
##  4 AA      MIA    7234
##  5 UA      ORD    6984
##  6 UA      IAH    6924
##  7 UA      SFO    6819
##  8 B6      FLL    6563
##  9 B6      MCO    6472
## 10 AA      ORD    6059
## # ... with 304 more rows
```

Pregunta 6

¿Qué hace el parámetro sort como argumento de count()? ¿Cuándo puede sernos útil? Vuelve a la lista de funciones útiles para filtrar y mutar y describe cómo cada operación cambia cuando la juntamos con un group_by.

```
not_cancelled %>% count(dest)
```

```
## # A tibble: 104 x 2
##    dest      n
##    <chr> <int>
##  1 ABQ     254
##  2 ACK     264
##  3 ALB     418
##  4 ANC       8
##  5 ATL   16837
##  6 AUS    2411
##  7 AVL     261
##  8 BDL     412
##  9 BGR     358
## 10 BHM     269
## # ... with 94 more rows
```

```r
# Usando sort = TRUE dentro de count ordenda de mayor a menor,
not_cancelled %>% count(dest, sort = T)
```

```
## # A tibble: 104 x 2
##    dest      n
##    <chr> <int>
##  1 ATL   16837
##  2 ORD   16566
##  3 LAX   16026
##  4 BOS   15022
##  5 MCO   13967
##  6 CLT   13674
##  7 SFO   13173
##  8 FLL   11897
##  9 MIA   11593
## 10 DCA    9111
## # ... with 94 more rows
```

```r
# Combinando con group_by nos ordena el número de ocurrencias de las agrupaciones
not_cancelled %>%
  group_by(carrier, dest) %>%
  count(dest, sort = T)
```

```
## # A tibble: 312 x 3
## # Groups:   carrier, dest [312]
##    carrier dest      n
##    <chr>   <chr> <int>
##  1 DL      ATL   10452
##  2 US      CLT    8498
##  3 AA      MIA    7143
##  4 AA      DFW    6966
##  5 UA      IAH    6814
##  6 UA      ORD    6744
##  7 UA      SFO    6728
##  8 B6      FLL    6466
##  9 B6      MCO    6409
## 10 AA      ORD    5846
## # ... with 302 more rows
```

Pregunta 7

Vamos a por los peores aviones. Investiga el top 10 de qué aviones (número de cola y compañía) llegaron más tarde a su destino.

```r
(not_cancelled %>%
  group_by(carrier, tailnum, arr_delay) %>%
  count() %>%
  arrange(desc(arr_delay)))[1:10,]
```

```
## # A tibble: 10 x 4
```

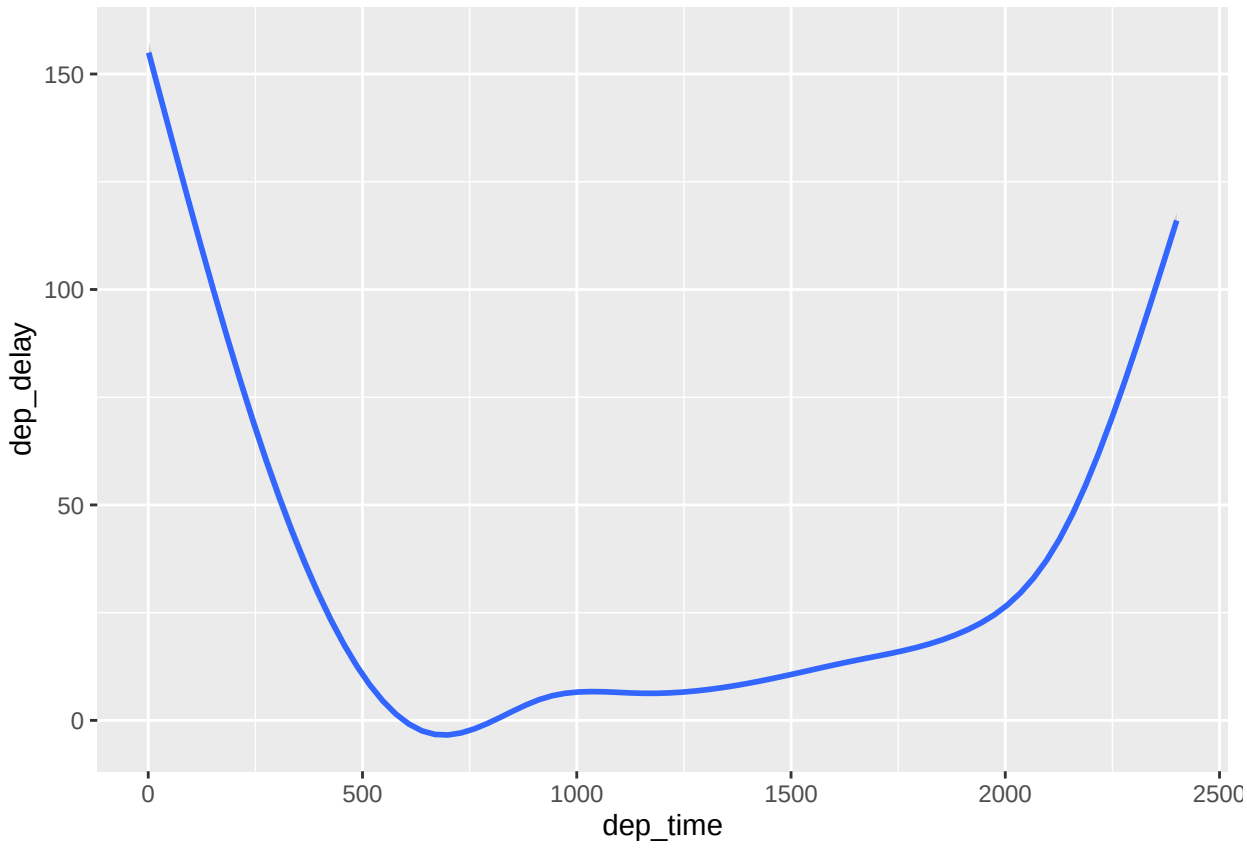
```
## # Groups:   carrier, tailnum, arr_delay [10]
##    carrier tailnum arr_delay     n
##    <chr>   <chr>       <dbl> <int>
##  1 HA      N384HA       1272     1
##  2 MQ      N504MQ       1127     1
##  3 MQ      N517MQ       1109     1
##  4 AA      N338AA       1007     1
##  5 MQ      N665MQ        989     1
##  6 DL      N959DL        931     1
##  7 DL      N927DA        915     1
##  8 DL      N6716C        895     1
##  9 AA      N5DMAA        878     1
## 10 MQ      N523MQ        875     1
```

Pregunta 8

Queremos saber qué hora del día nos conviene volar si queremos evitar los retrasos en la salida. Difícil: Queremos saber qué día de la semana nos conviene volar si queremos evitar los retrasos en la salida.

```
not_cancelled %>%
  ggplot(mapping = aes(x = dep_time, y = dep_delay)) +
    geom_smooth()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Del gráfico anterior podemos concluir que las horas en donde hay menos retraso sen encuentran entre las 5:00 y 7:30.

```
#Función para calcular día de la semana.
weekday <- function(year,month,day) {
  a = (14-month)%/%12
  y = year - a
  m = month + 12*a - 2
  d = (day + y + y%/%4 - y%/%100 + y%/%400 + (31* m)%/%12) %% 7
  return(d)
}
```

```
weekday(1988,11,27)
```

```
## [1] 0
```

```
data <- not_cancelled %>%
  group_by(year,month, day, dep_delay) %>%
  summarise(week.day = weekday(year, month, day))
```

```
## `summarise()` has grouped output by 'year', 'month', 'day', 'dep_delay'. You
## can override using the `.groups` argument.
```

```
data
```

```
## # A tibble: 327,346 x 5
## # Groups:   year, month, day, dep_delay [42,806]
##     year month   day dep_delay week.day
##    <int> <int> <int>     <dbl>    <dbl>
##  1  2013     1     1       -15        2
##  2  2013     1     1       -15        2
##  3  2013     1     1       -14        2
##  4  2013     1     1       -13        2
##  5  2013     1     1       -12        2
##  6  2013     1     1       -11        2
##  7  2013     1     1       -11        2
##  8  2013     1     1       -10        2
##  9  2013     1     1       -10        2
## 10  2013     1     1       -10        2
## # ... with 327,336 more rows
```

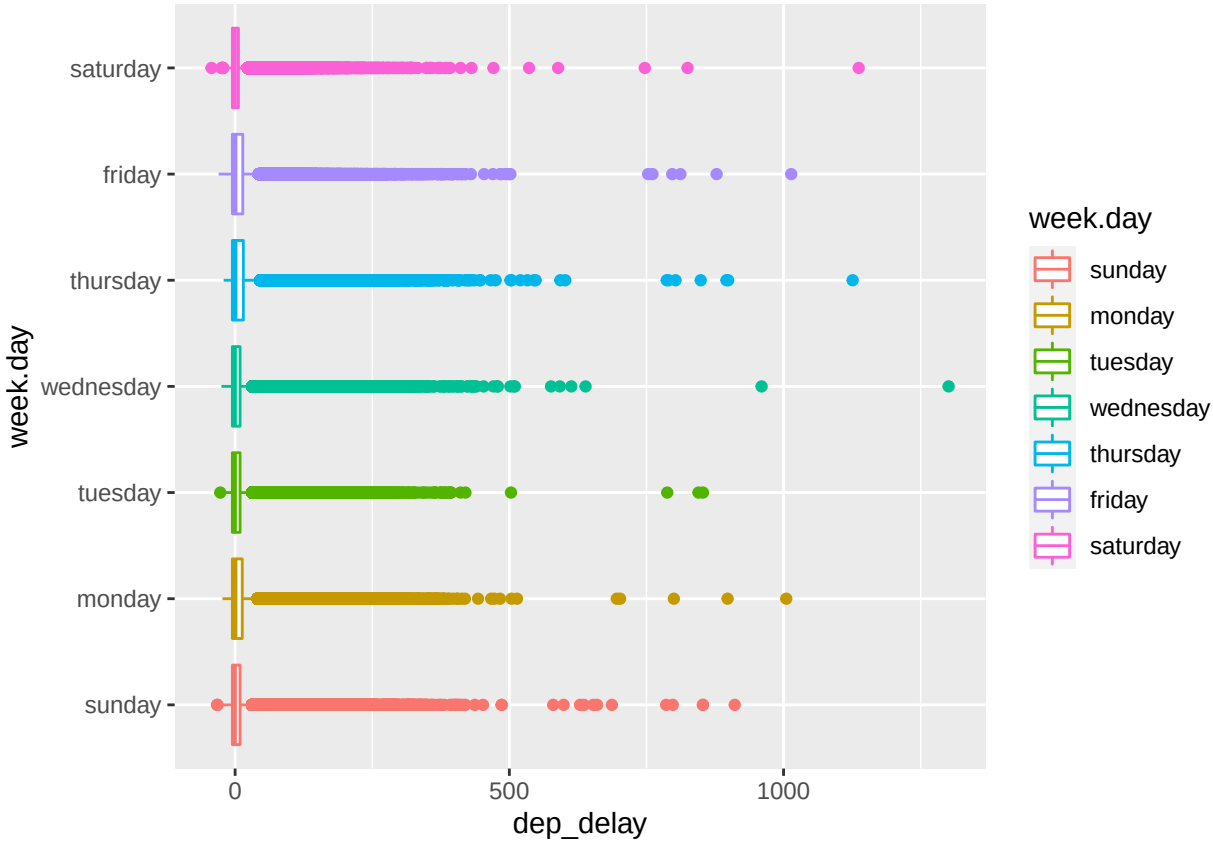
```
data$week.day = factor(data$week.day, labels = c("sunday", "monday",
                                                 "tuesday", "wednesday",
                                                 "thursday", "friday", "saturday"))

data
```

```
## # A tibble: 327,346 x 5
## # Groups:   year, month, day, dep_delay [42,806]
##     year month   day dep_delay week.day
##    <int> <int> <int>     <dbl> <fct>
```

```
##  1  2013     1     1       -15 tuesday
##  2  2013     1     1       -15 tuesday
##  3  2013     1     1       -14 tuesday
##  4  2013     1     1       -13 tuesday
##  5  2013     1     1       -12 tuesday
##  6  2013     1     1       -11 tuesday
##  7  2013     1     1       -11 tuesday
##  8  2013     1     1       -10 tuesday
##  9  2013     1     1       -10 tuesday
## 10  2013     1     1       -10 tuesday
## # ... with 327,336 more rows
```

El día de la semana que hay menos cantidad de retrasos son los jueves.

```
ggplot(data = data)+
  geom_boxplot(mapping = aes(y = week.day, x = dep_delay, color = week.day))
```



Pregunta 9

Para cada destino, calcula el total de minutos de retraso acumulado. Para cada uno de ellos, calcula la proporción del total de retraso para dicho destino.

```
not_cancelled %>%
  group_by(dest) %>%
```

```
  summarise(dep_delay,
            sum_delay = sum(dep_delay),
            prop = round(dep_delay / sum_delay,2)) %>%
  arrange(sum_delay)
```

```
## `summarise()` has grouped output by 'dest'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 327,346 x 4
## # Groups:   dest [104]
##    dest  dep_delay sum_delay  prop
##    <chr>     <dbl>     <dbl> <dbl>
##  1 PSP           6       -53 -0.11
##  2 PSP         -10       -53  0.19
##  3 PSP          -8       -53  0.15
##  4 PSP          -4       -53  0.08
##  5 PSP          10       -53 -0.19
##  6 PSP           7       -53 -0.13
##  7 PSP          -4       -53  0.08
##  8 PSP          -5       -53  0.09
##  9 PSP          -2       -53  0.04
## 10 PSP           0       -53  0
## # ... with 327,336 more rows
```

Pregunta 10

Los retrasos suelen estar correlacionados con el tiempo. Aunque el problema que ha causado el primer retraso de un avión se resuelva, el resto de vuelos se retrasan para que salgan primero los aviones que debían haber partido antes. Intenta usar la función lag() y explora cómo el retraso de un avión se relaciona con el retraso del avión inmediatamente anterior o posterior.

```
# Espero resolverlas cuando esté mas avanzado en el curso
```

Pregunta 11

Vamos a por los destinos esta vez. Localiza vuelos que llegaron 'demasiado rápido' a sus destinos. Seguramente, el becario se equivocó al introducir el tiempo de vuelo y se trate de un error en los datos. Calcula para ello el cociente entre el tiempo en el aire de cada vuelo relativo al tiempo de vuelo del avión que tardó menos en llegar a dicho destino. ¿Qué vuelos fueron los que más se retrasaron en el aire?

```
# Espero resolverlas cuando esté mas avanzado en el curso
```