Práctica 1. FMAD 2021-2022

ICAI. Master en Big Data. Fundamentos Matemáticos del Análisis de Datos (FMAD).

Monsalve Rodilla, Ignacio

Curso 2021-22. Última actualización: 2021-09-15

Contenido

Ejercicio I

```
library(tidyverse)
```

En primer lugar, se debe guardar el fichero. Para ello se utilizan los comandos del script 'herramientas'.

Cargo el conjunto de datos

```
chlstrl = read_csv('cholesterol.csv')
```

```
## Rows: 403 Columns: 7
```

```
## -- Column specification --------------------------------------------
----------
## Delimiter: ","
## chr (1): gender
## dbl (6): chol, age, height, weight, waist, hip
```

```
##
## i Use `spec()` to retrieve the full column specification for this data
.
## i Specify the column types or set `show_col_types = FALSE` to quiet th
is message.
```

```
chlstrl
```

```
## # A tibble: 403 x 7
##     chol   age gender height weight waist   hip
##    <dbl> <dbl> <chr>   <dbl>  <dbl> <dbl> <dbl>
##  1   203    46 female     62    121    29    38
##  2   165    29 female     64    218    46    48
##  3   228    58 female     61    256    49    57
##  4    78    67 male       67    119    33    38
##  5   249    64 male       68    183    44    41
##  6   248    34 male       71    190    36    42
##  7   195    30 male       69    191    46    49
##  8   227    37 male       59    170    34    39
##  9   177    45 male       69    166    34    40
## 10   263    55 female     63    202    45    50
## # ... with 393 more rows
```

```
View(chlstrl)
```

La información básica la vemos aquí

```
str(chlstrl)
```

```
## spec_tbl_df [403 x 7] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ chol  : num [1:403] 203 165 228 78 249 248 195 227 177 263 ...
##  $ age   : num [1:403] 46 29 58 67 64 34 30 37 45 55 ...
##  $ gender: chr [1:403] "female" "female" "female" "male" ...
##  $ height: num [1:403] 62 64 61 67 68 71 69 59 69 63 ...
##  $ weight: num [1:403] 121 218 256 119 183 190 191 170 166 202 ...
##  $ waist : num [1:403] 29 46 49 33 44 36 46 34 34 45 ...
##  $ hip   : num [1:403] 38 48 57 38 41 42 49 39 40 50 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   chol = col_double(),
##   ..   age = col_double(),
##   ..   gender = col_character(),
##   ..   height = col_double(),
##   ..   weight = col_double(),
##   ..   waist = col_double(),
##   ..   hip = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Para ver datos ausentes

```
# is.na(chlstrl) Nos devuelve TRUE / FALSE de todos
sum(is.na(chlstrl))

## [1] 11
```

Para localizarlos en la tabla, podemos utilizar which

```
which(is.na(chlstrl))

##  [1]   28 1273 1296 1405 1441 1527 1774 2352 2409 2755 2812
```

Empezamos con una variable continua.

Es cierto que esto puede causar problemas, como ya se comentó en las sesiones de clase, ya que uno puede decidir si una variable es continua o discreta en función de lo que quiera analizar. En este caso, se cree que todas las variables pueden ser consideradas como continuas, salvo la variable gender que será un factor.

Analizamos la variable 'chol' El tipo de dato es un double

```
mean(chlstrl$chol, na.rm= TRUE)

## [1] 207.8458

median(chlstrl$chol, na.rm= TRUE)

## [1] 204
```

Podemos ver como la media y la mediana son similares, por lo que es un indicativo de que no existen valores atípicos que estén 'ensuciando' la media.

Los valores máximos y mínimos son:

```
max(chlstrl$chol, na.rm= TRUE)

## [1] 443

min(chlstrl$chol, na.rm= TRUE)

## [1] 78
```

La desviación típica

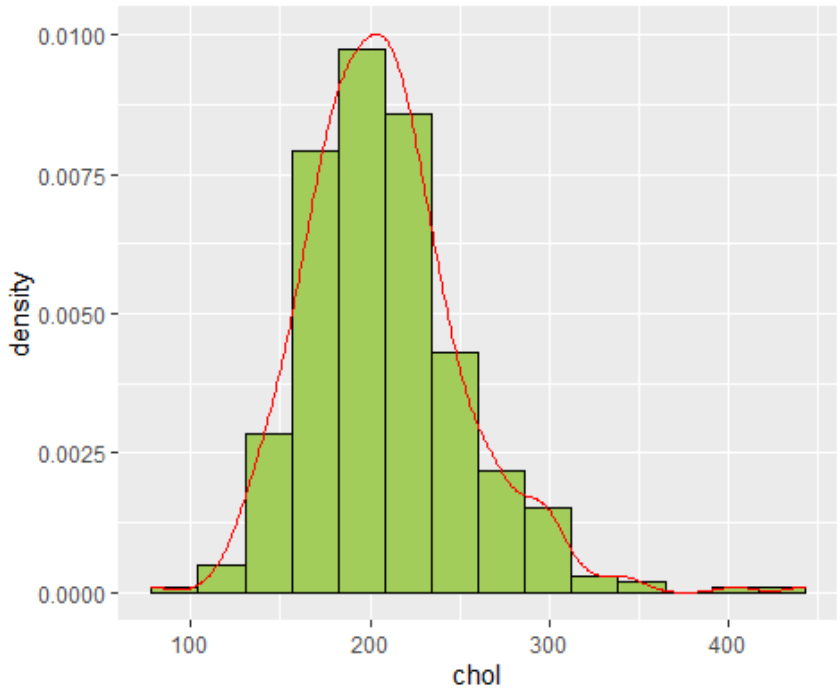
```
sd(chlstrl$chol, na.rm= TRUE)

## [1] 44.44556
```

Realizamos los gráficos

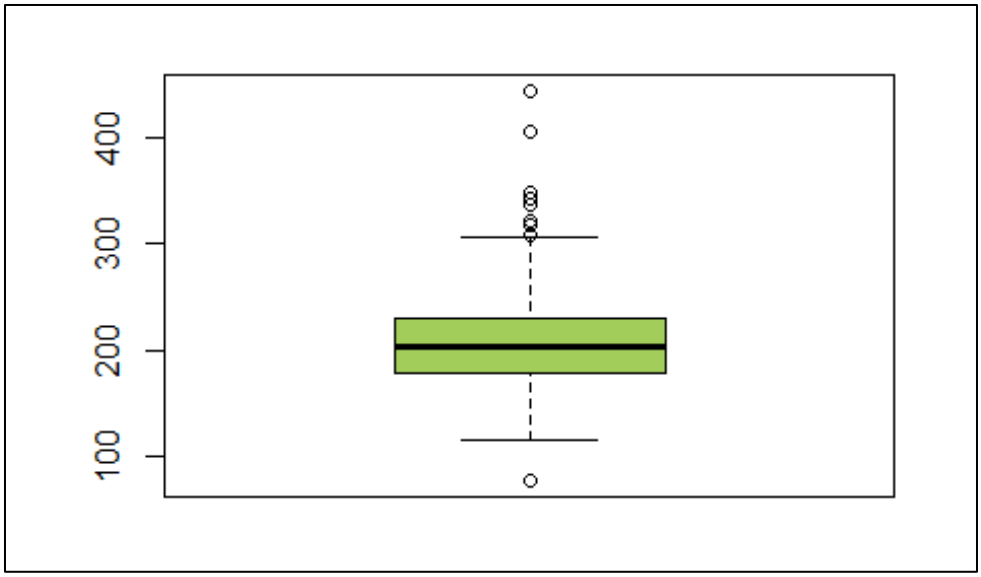
Primero se define 'cortes' como se vio en clase, para fijar los extremos

```
cortes = seq(min(chlstrl$chol,na.rm=TRUE), max(chlstrl$chol,na.rm=TRUE),
length.out = 15)
ggplot(data = chlstrl, mapping = aes(x=chol)) +
  geom_histogram(breaks = cortes,aes(y=stat(density)),
                 fill = "darkolivegreen3", color="black") +
  geom_density(col='red')
```



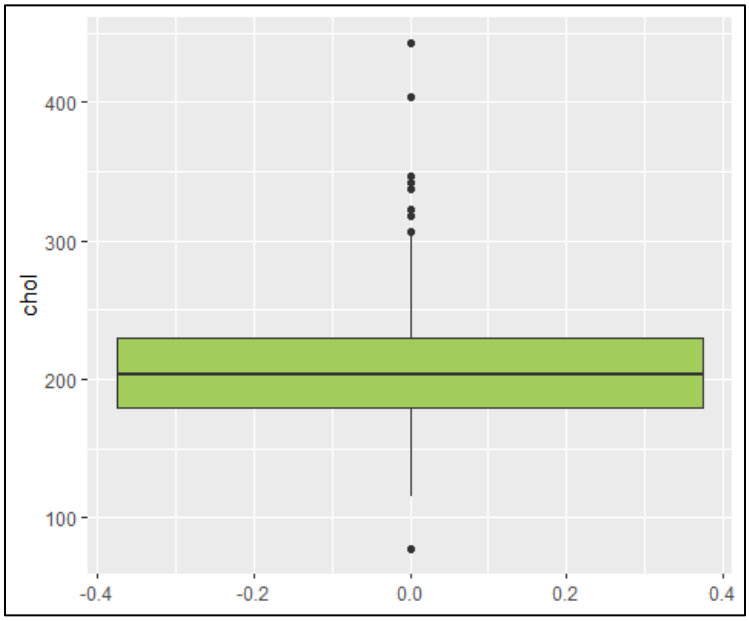
A continuación, realizamos un boxplot:

```
bxp_chol = boxplot(chlstrl$chol, col="darkolivegreen3")
```



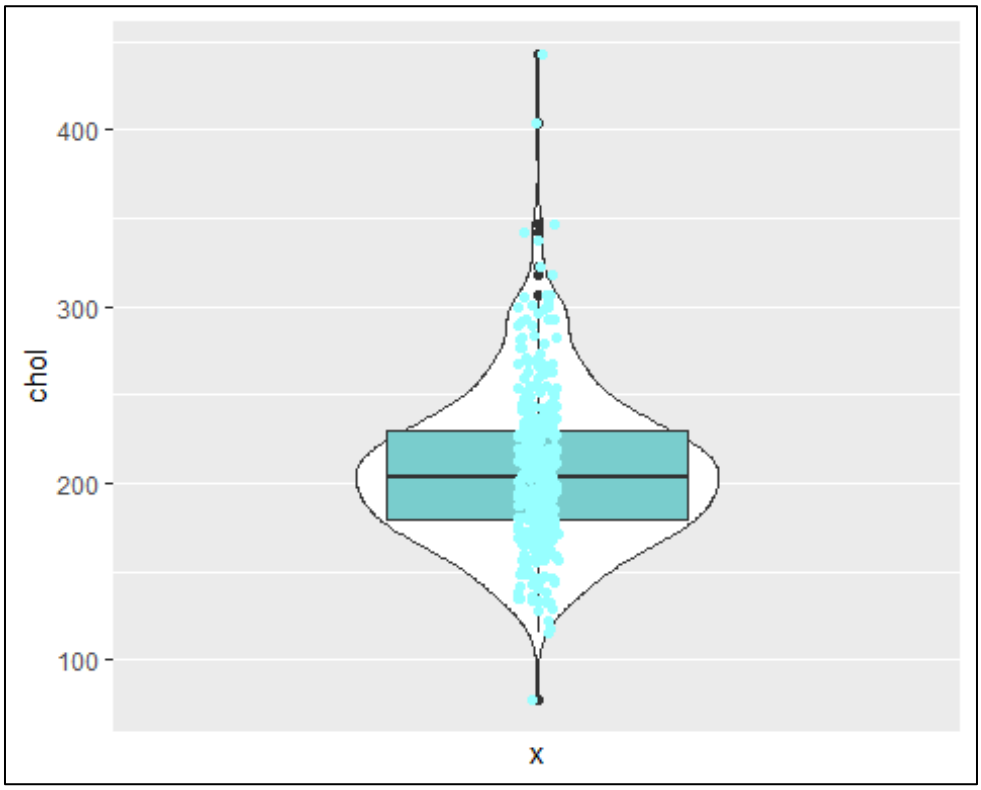
Con tidyverse

```
ggplot(chlstrl) +
  geom_boxplot(mapping = aes(y = chol), fill="darkolivegreen3")
```



Otro gráfico que es interesante visualizar es el de violín, como se muestra a continuación.

```
ggplot(chlstrl) +
  geom_violin(mapping = aes(x=0, y = chol)) +
  scale_x_discrete(breaks = c()) +
  geom_boxplot(mapping = aes(y = chol), fill="darkslategray3") +
  geom_jitter(aes(x=0, y = chol),
              position = position_jitter(w=0.05, h= 0), col="darkslategra
y1")
```



Analizamos las variables categóricas:

Se analiza la variable gender:

Hay que pasarlo a factor

Aquí podemos ver las personas que son hombres y mujeres:

```
table(chlstrl$gender)
```

```
##
## female   male
##    234    169
```

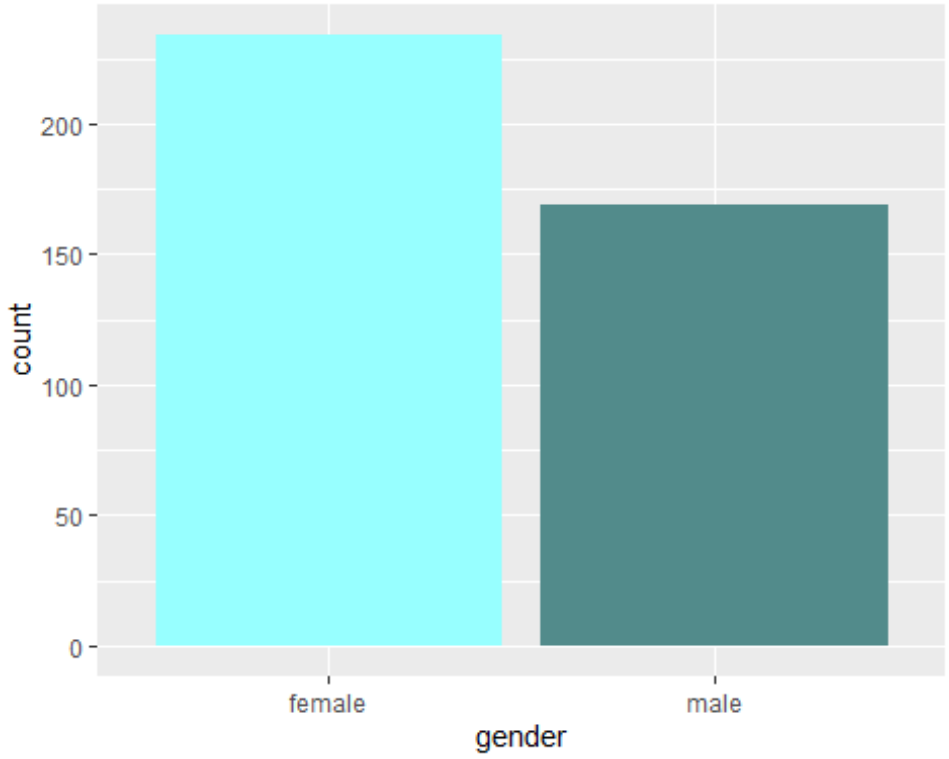
Aquí se ven las proporciones:

```
prop.table(table(chlstrl$gender))

##
##    female      male
## 0.5806452 0.4193548
```

El diagrama correspondiente:

```
ggplot(chlstrl) +
  geom_bar(mapping = aes(x = gender), fill= c('darkslategray1','darkslate
gray4'))
```



Vamos a convertirlo, pero manteniendo la tabla en caso de necesidad. Es cierto que en el enunciado se pide 'sobreescribir', y se podría 'machacar' chlstrl, sin embargo como se cumple el mismo propósito, se decide realizar una nueva asignación a chlstrl_si.

```
chlstrl_si <- chlstrl %>%
  mutate("height" = height*0.0254, "weight" = weight*0.454 )

chlstrl_si %>%
  mutate("BMI" = weight/(height)^2) -> chlstrl_si

View(chlstrl_si)
```

Creamos ahora los intervalos

```
vector_edades = seq(10,100,30)
vector_edades
```

```
## [1]  10  40  70 100
```

Una nueva asignación en chlstrl_sii con el nuevo mutate. En este caso, la única diferencia con el anterior es la columna añadida ageGroup

```
chlstrl_sii <- chlstrl_si %>%
  mutate('ageGroup'=cut(chlstrl_si$age, breaks = seq(10,100,30)))

View(chlstrl_sii)

chlstrl_sii %>%
  group_by(ageGroup) %>%
    count()
```

```
## # A tibble: 3 x 2
## # Groups:   ageGroup [3]
##   ageGroup     n
##   <fct>    <int>
## 1 (10,40]    160
## 2 (40,70]    207
## 3 (70,100]    36
```

Se puede hacer un nuevo tibble en el que únicamente se seleccione a las mujeres.

```
chlstrl_sii_mujeres = chlstrl_sii[chlstrl_sii$gender=='female', ]
View(chlstrl_sii_mujeres)
```

Esto se puede hacer también facilmente con dplyr:

```
chlstrl_sii %>%
  group_by(ageGroup) %>%
    filter(gender=="female")  %>%
      summarise(media_col = mean(chol,na.rm=TRUE),media_bmi = mean(BMI,na
.rm=TRUE))
```

```
## # A tibble: 3 x 3
##   ageGroup media_col media_bmi
##   <fct>        <dbl>     <dbl>
## 1 (10,40]       189.      30.5
## 2 (40,70]       221.      30.3
## 3 (70,100]      230.      29.4
```

Ejercicio II

En primer lugar, a la hora de crear el vector hay que tener una precaución.

Debemos evitar el 0, por lo que se concatena entre -100 y 100, evitando el 0

```
v=sample(c(-100:-1,1:100),9,replace = TRUE)

numero_de_cambios =function(v){
i=0
  for(p in seq(length(v)-1)){

    if( v[p]*v[p+1]<0 ){
      i=i+1
    }

  }
return(i)
}

numero_de_cambios(v)
```

```
## [1] 6
```

```
numero_de_cambios = function(v){
  pos=c()

  for(p in seq(length(v)-1)){

    if( v[p]*v[p+1]<0 ){
      pos=append(pos,p+1)
    }

  }

  if( is.null(pos) == TRUE){
    print("No hay ningún cambio de signo")
    }else{
      return(pos)
    }
}

numero_de_cambios(v)
```

```
## [1] 2 3 4 6 7 9
```

Ejercicio III

Ejercicio a

Realizamos los gráficos.

Es importante señalar que se van a guardar en **6 variables** ya que luego se visualizarán de manera conjunto, como se muestra en el enunciado.

Gráfico 1

```
g1 = ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point() +
  geom_smooth(se = FALSE)
```

Gráfico 2

```
g2 = ggplot(data = mpg, mapping = aes(x = displ, y = hwy, group = drv)) +
  geom_point() +
  geom_smooth(se = FALSE)
```

Gráfico 3

```
g3 = ggplot(data = mpg, mapping = aes(x = displ, y = hwy, colour = drv))
+
  geom_point() +
  geom_smooth(se = FALSE)
```

Gráfico 4

```
g4 = ggplot() +
  geom_point(data = mpg, mapping = aes(x = displ, y = hwy, colour = drv))
+
  geom_smooth(data = mpg, mapping = aes(x = displ, y = hwy),se = FALSE)
```

Gráfico 5

```
g5 = ggplot() +
  geom_point(data = mpg, mapping = aes(x = displ, y = hwy, colour = drv))
+
```

```
  geom_smooth(data = mpg, mapping = aes(x = displ, y = hwy, linetype = dr
v),se = FALSE)
```

Gráfico 6

En este último caso, notar que se realizan dos geom_point para realizar esa 'sombra' que se encuentra sobre los puntos

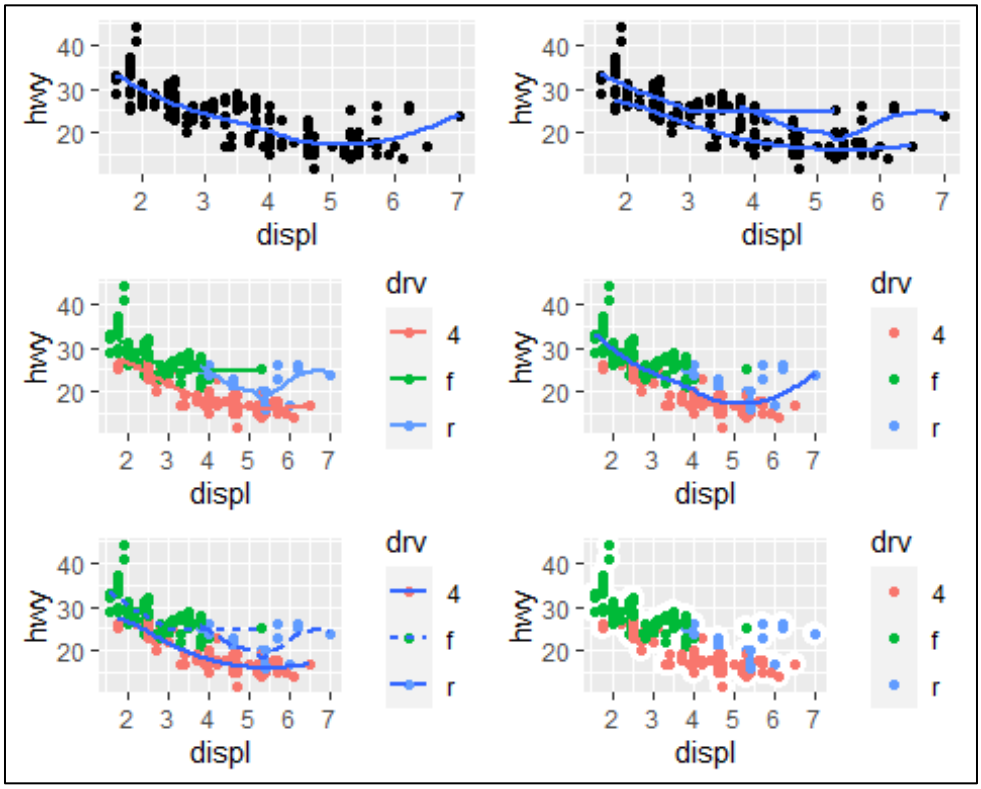
```
g6 = ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(size = 4, color = "white") +
  geom_point(aes(colour = drv))

library(gridExtra)

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine

grid.arrange(g1, g2, g3, g4, g5, g6, nrow = 3)
```



Ejercicio b

```
library(nycflights13)

View(flights)
```

En este ejercicio se pide encontrar vuelos que cumplan ciertas condiciones.

I

Un retraso de dos o más horas

Es importante conocer las UNIDADES. En este caso la variable que se necesita está en minutos, por lo que: 2h = 120'

```
filter(flights, arr_delay >= 120)
```

```
## # A tibble: 10,200 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      811            630       101     1047
830
##  2  2013     1     1      848           1835       853     1001
1950
##  3  2013     1     1      957            733       144     1056
853
##  4  2013     1     1     1114            900       134     1447
1222
##  5  2013     1     1     1505           1310       115     1638
1431
##  6  2013     1     1     1525           1340       105     1831
1626
##  7  2013     1     1     1549           1445        64     1912
1656
##  8  2013     1     1     1558           1359       119     1718
1515
##  9  2013     1     1     1732           1630        62     2028
1825
## 10  2013     1     1     1803           1620       103     2008
1750
## # ... with 10,190 more rows, and 11 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <ch
r>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hou
r <dttm>
```

||

Vuelos a Houston (IAH / HOU)

Con un operador OR (|) se puede realizar fácilmente

```
filter(flights, dest == "IAH" | dest == "HOU")
```

```
## # A tibble: 9,313 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      517            515         2      830
819
##  2  2013     1     1      533            529         4      850
830
##  3  2013     1     1      623            627        -4      933
932
##  4  2013     1     1      728            732        -4     1041
1038
##  5  2013     1     1      739            739         0     1104
1038
##  6  2013     1     1      908            908         0     1228
1219
##  7  2013     1     1     1028           1026         2     1350
1339
##  8  2013     1     1     1044           1045        -1     1352
1351
##  9  2013     1     1     1114            900       134     1447
1222
## 10  2013     1     1     1205           1200         5     1503
1505
## # ... with 9,303 more rows, and 11 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <ch
r>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hou
r <dttm>
```

III

Fueron manejados por United / America / Delta

```
filter(flights, carrier %in% c("AA", "DL", "UA"))
```

```
## # A tibble: 139,504 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      517            515         2      830
819
##  2  2013     1     1      533            529         4      850
830
##  3  2013     1     1      542            540         2      923
850
##  4  2013     1     1      554            600        -6      812
837
##  5  2013     1     1      554            558        -4      740
728
##  6  2013     1     1      558            600        -2      753
745
##  7  2013     1     1      558            600        -2      924
917
##  8  2013     1     1      558            600        -2      923
937
##  9  2013     1     1      559            600        -1      941
910
## 10  2013     1     1      559            600        -1      854
902
## # ... with 139,494 more rows, and 11 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <ch
r>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hou
r <dttm>
```

IV

Salieron en verano (meses de julio, agosto o septiembre)

```
filter(flights, month >= 7, month <= 9)
```

```
## # A tibble: 86,326 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     7     1        1           2029       212      236
2359
##  2  2013     7     1        2           2359         3      344
344
##  3  2013     7     1       29           2245       104      151
1
##  4  2013     7     1       43           2130       193      322
14
##  5  2013     7     1       44           2150       174      300
100
##  6  2013     7     1       46           2051       235      304
2358
##  7  2013     7     1       48           2001       287      308
2305
##  8  2013     7     1       58           2155       183      335
43
##  9  2013     7     1      100           2146       194      327
30
## 10  2013     7     1      100           2245       135      337
135
## # ... with 86,316 more rows, and 11 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <ch
r>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hou
r <dttm>
```

V

Llegaron tarde pero no salieron tarde (más de dos horas)

```
filter(flights, arr_delay > 120, dep_delay <= 0)
```

```
## # A tibble: 29 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    27     1419           1420        -1     1754
1550
##  2  2013    10     7     1350           1350         0     1736
1526
##  3  2013    10     7     1357           1359        -2     1858
1654
##  4  2013    10    16      657            700        -3     1258
1056
##  5  2013    11     1      658            700        -2     1329
1015
##  6  2013     3    18     1844           1847        -3       39
2219
##  7  2013     4    17     1635           1640        -5     2049
1845
##  8  2013     4    18      558            600        -2     1149
850
##  9  2013     4    18      655            700        -5     1213
950
## 10  2013     5    22     1827           1830        -3     2217
2010
## # ... with 19 more rows, and 11 more variables: arr_delay <dbl>, carri
er <chr>,
## #   flight <int>, tailnum <chr>, origin <chr>, dest <chr>, air_time <d
bl>,
## #   distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```

VI

Se retrasaron como mínimo una hora, pero durante el vuelo recuperaron 30'

```
filter(flights, dep_delay >= 60, dep_delay - arr_delay > 30)
```

```
## # A tibble: 1,844 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     2205           1720       285       46
2040
##  2  2013     1     1     2326           2130       116      131
18
##  3  2013     1     3     1503           1221       162     1803
1555
##  4  2013     1     3     1839           1700        99     2056
1950
##  5  2013     1     3     1850           1745        65     2148
2120
##  6  2013     1     3     1941           1759       102     2246
2139
##  7  2013     1     3     1950           1845        65     2228
2227
##  8  2013     1     3     2015           1915        60     2135
2111
##  9  2013     1     3     2257           2000       177       45
2224
## 10  2013     1     4     1917           1700       137     2135
1950
## # ... with 1,834 more rows, and 11 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <ch
r>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hou
r <dttm>
```

VII

Salieron entre medianoche y las 6am

```
filter(flights, dep_time <= 600 | dep_time == 2400)
```

```
## # A tibble: 9,373 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      517            515         2      830
819
##  2  2013     1     1      533            529         4      850
830
##  3  2013     1     1      542            540         2      923
850
##  4  2013     1     1      544            545        -1     1004
1022
##  5  2013     1     1      554            600        -6      812
837
##  6  2013     1     1      554            558        -4      740
728
##  7  2013     1     1      555            600        -5      913
854
##  8  2013     1     1      557            600        -3      709
723
##  9  2013     1     1      557            600        -3      838
846
## 10  2013     1     1      558            600        -2      753
745
## # ... with 9,363 more rows, and 11 more variables: arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <ch
r>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hou
r <dttm>
```