Clase 1 - Practico 1

Felix Rojo Lapalma

11 de mayo de 2018

Ejercicios:

Visualizacion es una herramienta muy importante para la generacion de intuicion, pero raramente uno tiene los datos en la forma necesaria. Frecuentemente se necesitara crear nuevas variables o simplemente reordenarlas.

Exploraremos ahora la manipulacion basica utilizando un conjunto de datos sobre los vuelos en Nueva York en 2013.

```
library(nycflights13)
fligths<-nycflights13::flights
fligths
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_t~ sched_~ dep_d~ arr_~ sched~ arr_d~ carr~ flig~
##    <int> <int> <int>  <int>   <int>  <dbl> <int>  <int>  <dbl> <chr> <int>
##  1  2013     1     1    517     515   2.00   830    819  11.0  UA     1545
##  2  2013     1     1    533     529   4.00   850    830  20.0  UA     1714
##  3  2013     1     1    542     540   2.00   923    850  33.0  AA     1141
##  4  2013     1     1    544     545  -1.00  1004   1022 -18.0  B6      725
##  5  2013     1     1    554     600  -6.00   812    837 -25.0  DL      461
##  6  2013     1     1    554     558  -4.00   740    728  12.0  UA     1696
##  7  2013     1     1    555     600  -5.00   913    854  19.0  B6      507
##  8  2013     1     1    557     600  -3.00   709    723 -14.0  EV     5708
##  9  2013     1     1    557     600  -3.00   838    846 - 8.00 B6       79
## 10  2013     1     1    558     600  -2.00   753    745   8.00 AA      301
## # ... with 336,766 more rows, and 8 more variables: tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

Practico 1: Entregar un Rmd donde se encuentren todos los vuelos que:

- Que arribaron con un retraso de mas de dos horas.
- Volaron hacia Houston (IAH o HOU)
- Fueron operados por United, American o Delta.
- Salieron en Verano (Julio, Agosto y Septiembre)
- Arrivaron mas de dos horas tarde, pero salieron bien.
- Salieron entre medianoche y las 6 am.

Revisamos un poco el dataframe. Empezamos con un summary.

```
##       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
##
```

Arribo con retraso > 120 min

De nycflights13 tenemos que el retraso lo podemos encontrar en unidades de *[minutos]* bajo la columna *arr_delay* del dataframe. Asimismo del *summary* observamos que tenemos NA's :9430 . Podriamos reemplazarlos por los valores medios sin embargo observemos que la cantidad constituye un 2.5871796 % por lo cual vamos a no considerarlos simplemente. Entonces, buscamos aquellos que arribaron con un retraso de mas de dos horas, es decir la condición es que arr_delay>120

```
## # A tibble: 19,464 x 19
##     year month   day dep_~ sche~ dep_~ arr_~ sche~ arr_~ carr~ flig~ tail~
##    <int> <int> <int> <int> <int> <dbl> <int> <int> <dbl> <chr> <int> <chr>
##  1  2013     1     1   811   630 101    1047   830   137 MQ     4576 N531~
##  2  2013     1     1   848  1835 853    1001  1950   851 MQ     3944 N942~
##  3  2013     1     1   957   733 144    1056   853   123 UA      856 N534~
##  4  2013     1     1  1114   900 134    1447  1222   145 UA     1086 N765~
##  5  2013     1     1  1505  1310 115    1638  1431   127 EV     4497 N179~
```

```
##  6  2013     1     1  1525  1340 105    1831  1626   125 B6      525 N231~
##  7    NA    NA    NA    NA    NA  NA      NA    NA    NA <NA>     NA <NA>
##  8    NA    NA    NA    NA    NA  NA      NA    NA    NA <NA>     NA <NA>
##  9  2013     1     1  1549  1445  64.0  1912  1656   136 EV     4181 N211~
## 10  2013     1     1  1558  1359 119    1718  1515   123 EV     5712 N826~
## # ... with 19,454 more rows, and 7 more variables: origin <chr>,
## #   dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>,
## #   time_hour <dttm>
```

Esta ultima forma no parece correcta puesto se ve en forma explicita que no desechamos los *NAs*. Alternativamente hacemos Ref2

```
flightsdelayed2hs_2=subset(flights,arr_delay>120)
flightsdelayed2hs_2
```

```
## # A tibble: 10,034 x 19
##     year month   day dep_~ sche~ dep_~ arr_~ sche~ arr_~ carr~ flig~ tail~
##    <int> <int> <int> <int> <int> <dbl> <int> <int> <dbl> <chr> <int> <chr>
##  1  2013     1     1   811   630 101    1047   830   137 MQ     4576 N531~
##  2  2013     1     1   848  1835 853    1001  1950   851 MQ     3944 N942~
##  3  2013     1     1   957   733 144    1056   853   123 UA      856 N534~
##  4  2013     1     1  1114   900 134    1447  1222   145 UA     1086 N765~
##  5  2013     1     1  1505  1310 115    1638  1431   127 EV     4497 N179~
##  6  2013     1     1  1525  1340 105    1831  1626   125 B6      525 N231~
##  7  2013     1     1  1549  1445  64.0  1912  1656   136 EV     4181 N211~
##  8  2013     1     1  1558  1359 119    1718  1515   123 EV     5712 N826~
##  9  2013     1     1  1732  1630  62.0  2028  1825   123 EV     4092 N169~
## 10  2013     1     1  1803  1620 103    2008  1750   138 MQ     4622 N504~
## # ... with 10,024 more rows, and 7 more variables: origin <chr>,
## #   dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>,
## #   time_hour <dttm>
```

Esta ultima parece correcta en cuanto subset descuenta por defecto los casos con NA. Alternativamente de Ref3

```
completeFun <- function(data, desiredCols) {
  completeVec <- complete.cases(data[, desiredCols])
  return(data[completeVec, ])
}
flightsdelayed2hs_int=completeFun(flights,'arr_delay')
flightsdelayed2hs_3=flightsdelayed2hs_int[flightsdelayed2hs_int$arr_delay>120,]
flightsdelayed2hs_3
```

```
## # A tibble: 10,034 x 19
##     year month   day dep_~ sche~ dep_~ arr_~ sche~ arr_~ carr~ flig~ tail~
##    <int> <int> <int> <int> <int> <dbl> <int> <int> <dbl> <chr> <int> <chr>
##  1  2013     1     1   811   630 101    1047   830   137 MQ     4576 N531~
##  2  2013     1     1   848  1835 853    1001  1950   851 MQ     3944 N942~
##  3  2013     1     1   957   733 144    1056   853   123 UA      856 N534~
##  4  2013     1     1  1114   900 134    1447  1222   145 UA     1086 N765~
##  5  2013     1     1  1505  1310 115    1638  1431   127 EV     4497 N179~
##  6  2013     1     1  1525  1340 105    1831  1626   125 B6      525 N231~
##  7  2013     1     1  1549  1445  64.0  1912  1656   136 EV     4181 N211~
##  8  2013     1     1  1558  1359 119    1718  1515   123 EV     5712 N826~
##  9  2013     1     1  1732  1630  62.0  2028  1825   123 EV     4092 N169~
## 10  2013     1     1  1803  1620 103    2008  1750   138 MQ     4622 N504~
```

```
## # ... with 10,024 more rows, and 7 more variables: origin <chr>,
## #   dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>,
## #   time_hour <dttm>
```

Estas ultimas formas vemos que dan los resultados requeridos. Tenemos un total de 10034 vuelos que sufrieron el retraso mencionado.

Volaron hacia Houston (IAH o HOU)

Buscamos aquellos vuelos con *dest* = IAH o HOU.

```
toHouston=flights[flights$dest=='IAH' | flights$dest=='HOU',]
toHouston
```

```
## # A tibble: 9,313 x 19
##     year month   day dep_t~ sched~ dep_de~ arr_~ sche~ arr_de~ carr~ flig~
##    <int> <int> <int>  <int>  <int>   <dbl> <int> <int>   <dbl> <chr> <int>
##  1  2013     1     1    517    515    2.00   830   819   11.0  UA     1545
##  2  2013     1     1    533    529    4.00   850   830   20.0  UA     1714
##  3  2013     1     1    623    627 -  4.00   933   932    1.00 UA      496
##  4  2013     1     1    728    732 -  4.00  1041  1038    3.00 UA      473
##  5  2013     1     1    739    739    0     1104  1038   26.0  UA     1479
##  6  2013     1     1    908    908    0     1228  1219    9.00 UA     1220
##  7  2013     1     1   1028   1026    2.00  1350  1339   11.0  UA     1004
##  8  2013     1     1   1044   1045 -  1.00  1352  1351    1.00 UA      455
##  9  2013     1     1   1114    900  134     1447  1222  145    UA     1086
## 10  2013     1     1   1205   1200    5.00  1503  1505 -  2.00 UA     1461
## # ... with 9,303 more rows, and 8 more variables: tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

Tenemos entonces 9313 vuelos entrantes.

Fueron operados por United, American o Delta.

Tenemos que encontrar los codigos que corresponden a cada carrier. Nuevamente de nycflights13

```
carriers<-nycflights13::airlines
carriers
```

```
## # A tibble: 16 x 2
##    carrier name
##    <chr>   <chr>
##  1 9E      Endeavor Air Inc.
##  2 AA      American Airlines Inc.
##  3 AS      Alaska Airlines Inc.
##  4 B6      JetBlue Airways
##  5 DL      Delta Air Lines Inc.
##  6 EV      ExpressJet Airlines Inc.
##  7 F9      Frontier Airlines Inc.
##  8 FL      AirTran Airways Corporation
##  9 HA      Hawaiian Airlines Inc.
## 10 MQ      Envoy Air
## 11 OO      SkyWest Airlines Inc.
## 12 UA      United Air Lines Inc.
```

```
## 13 US      US Airways Inc.
## 14 VX      Virgin America
## 15 WN      Southwest Airlines Co.
## 16 YV      Mesa Airlines Inc.
```

```
carriers[grepl("United", carriers$name) | grepl("American", carriers$name) | grepl("Delta", carriers$nam
```

```
## # A tibble: 3 x 2
##   carrier name
##   <chr>   <chr>
## 1 AA      American Airlines Inc.
## 2 DL      Delta Air Lines Inc.
## 3 UA      United Air Lines Inc.
```

Buscamos aquellos que cumplan con el *carrier name* encontrado para cada caso.

```
carriersmatch=flights[flights$carrier=='AA' | flights$carrier=='DL'| flights$carrier=='UA',]
carriersmatch
```

```
## # A tibble: 139,504 x 19
##     year month   day dep_t~ sched_~ dep_d~ arr_~ sched~ arr_d~ carr~ flig~
##    <int> <int> <int>  <int>   <int>  <dbl> <int>  <int>  <dbl> <chr> <int>
##  1  2013     1     1    517     515   2.00   830    819  11.0  UA     1545
##  2  2013     1     1    533     529   4.00   850    830  20.0  UA     1714
##  3  2013     1     1    542     540   2.00   923    850  33.0  AA     1141
##  4  2013     1     1    554     600  -6.00   812    837 -25.0  DL      461
##  5  2013     1     1    554     558  -4.00   740    728  12.0  UA     1696
##  6  2013     1     1    558     600  -2.00   753    745   8.00 AA      301
##  7  2013     1     1    558     600  -2.00   924    917   7.00 UA      194
##  8  2013     1     1    558     600  -2.00   923    937 -14.0  UA     1124
##  9  2013     1     1    559     600  -1.00   941    910  31.0  AA      707
## 10  2013     1     1    559     600  -1.00   854    902 - 8.00 UA     1187
## # ... with 139,494 more rows, and 8 more variables: tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

Entre las tres operaron 139504 vuelos.

Salieron en Verano (Julio, Agosto y Septiembre)

Para esto podriamos usar la informacion de *month*

```
flight_month=completeFun(flights,'month')
flight_summer=flight_month[flight_month$month>=7  & flight_month$month<=9,];
flight_summer
```

```
## # A tibble: 86,326 x 19
##     year month   day dep_t~ sched_~ dep_de~ arr_~ sched~ arr_~ carr~ flig~
##    <int> <int> <int>  <int>   <int>   <dbl> <int>  <int> <dbl> <chr> <int>
##  1  2013     7     1      1    2029  212       236   2359   157 B6      915
##  2  2013     7     1      2    2359    3.00    344    344     0 B6     1503
##  3  2013     7     1     29    2245  104       151      1   110 B6      234
##  4  2013     7     1     43    2130  193       322     14   188 B6     1371
##  5  2013     7     1     44    2150  174       300    100   120 AA      185
##  6  2013     7     1     46    2051  235       304   2358   186 B6      165
##  7  2013     7     1     48    2001  287       308   2305   243 VX      415
```

```
##  8  2013     7     1     58    2155  183      335     43   172 B6      425
##  9  2013     7     1    100    2146  194      327     30   177 B6     1183
## 10  2013     7     1    100    2245  135      337    135   122 B6      623
## # ... with 86,316 more rows, and 8 more variables: tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

En esta condicion estamos utilizando explicitamente el mes y no las horas. Podria ocurrir el caso que se informe,por ejemplo, mes de septiembre fecha 30 y hora de salida 2359 pero que ese caso sufra un delay con lo cual podria tener que ser removido del set. Esa condicion la revisamos mas adelante.

Arrivaron mas de dos horas tarde, pero salieron bien.

```
flight_dep_delay_arr_delay=completeFun(flights,c("dep_delay", "arr_delay"))
flight_dep_delayOK_arr_delay_Fail=flight_dep_delay_arr_delay[flight_dep_delay_arr_delay$dep_delay==0 & 
flight_dep_delayOK_arr_delay_Fail
```

```
## # A tibble: 3 x 19
##    year month   day dep_t~ sche~ dep_~ arr_~ sche~ arr_~ carr~ flig~ tail~
##   <int> <int> <int>  <int> <int> <dbl> <int> <int> <dbl> <chr> <int> <chr>
## 1  2013    10     7   1350  1350     0  1736  1526   130 EV     5181 N611~
## 2  2013     5    23   1810  1810     0  2208  2000   128 MQ     4626 N525~
## 3  2013     7     1    905   905     0  1443  1223   140 DL     1057 N337~
## # ... with 7 more variables: origin <chr>, dest <chr>, air_time <dbl>,
## #   distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```

Tenemos entonces 3 casos.

Salieron entre medianoche y las 6 am.

Para esta situacion ampliamos el dataframe calculando el deptime a seg y seguidamente ampliamos el campo a time_hour

```
flight_dep_time=completeFun(flights,c('dep_time'))
flight_dep_time$dep_real <-flight_dep_time$time_hour+flight_dep_time$dep_delay*60+flight_dep_time$minut
flight_dep_time
```

```
## # A tibble: 328,521 x 20
##     year month   day dep_t~ sched_~ dep_d~ arr_~ sched~ arr_d~ carr~ flig~
##    <int> <int> <int>  <int>   <int>  <dbl> <int>  <int>  <dbl> <chr> <int>
##  1  2013     1     1    517     515   2.00   830    819  11.0  UA     1545
##  2  2013     1     1    533     529   4.00   850    830  20.0  UA     1714
##  3  2013     1     1    542     540   2.00   923    850  33.0  AA     1141
##  4  2013     1     1    544     545  -1.00  1004   1022 -18.0  B6      725
##  5  2013     1     1    554     600  -6.00   812    837 -25.0  DL      461
##  6  2013     1     1    554     558  -4.00   740    728  12.0  UA     1696
##  7  2013     1     1    555     600  -5.00   913    854  19.0  B6      507
##  8  2013     1     1    557     600  -3.00   709    723 -14.0  EV     5708
##  9  2013     1     1    557     600  -3.00   838    846 - 8.00 B6       79
## 10  2013     1     1    558     600  -2.00   753    745   8.00 AA      301
## # ... with 328,511 more rows, and 9 more variables: tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>, dep_real <dttm>
```

Y el set que buscamos es:

```
flight_zero_to_six <- subset(flight_dep_time, format(flight_dep_time$dep_real, '%H') %in% c('00', '01',
```

Revisemos nuevamente aquellos casos que salieron en verano

Revisemos nuevamente los meses de la actividad anterior

```
flight_summer_2 <- subset(flight_dep_time, format(flight_dep_time$dep_real, '%m') %in% c('07', '08', '0
```

Es decir que en la instancia anterior contamos 1869 casos de mas.