PRAC2: Neteja i anàlisi de les dades

Marc Ferrer Margarit (mferrermargarit@uoc.edu) i Marc Ramos Bruach (mramosbru@uoc.edu)

5/15/2021

1. Descripció del dataset. Perquè és important i quina pregunta/problema pretén respondre? (0.5) (MF)

El dataset que farem servir per a aquesta pràctica conté els retards i cancel · lacions dels vols 2015 que surten de l'aeroport de San Francisco. Normalment els motius principals dels retards de vol són relacionats amb el temps, però en alguns casos també hi ha retards de vols relacionats amb les companyies aèries o aeroports. Aquest document examina i mostra les causes de retard i cancel · lació en diversos aspectes. Així doncs aquest dataset és important per saber quines són les principals causes dels retards que s'hi han produït durant el 2015.

Les preguntes que volem respondre amb aquest dataset són quins dies de la setmana els quals es produeixen més retards, quines aerolínies són les tenen els retards i veure si la distància dels vols influeix en els retards que es produeixen.

Aquest dataset ha sigut obtingut a partir d'una pràctica anterior realitzada durant el màster de Data Sciene de la UOC. També es pot obtenir el dataset complet, amb totes les dades dels vols (aprox. 600 MB) al següent enllaç: https://www.kaggle.com/usdot/flight-delays?select=flights.csv

2. Integració i selecció de les dades d'interès a analitzar. (0.5) (MF)

Per veure les dades que conté el dataset el carregarem i mostrarem les columnes que conté i la mida del dataset:

```
flights <- read.csv("../data/flights.csv")
str(flights)
```

```
## 'data.frame':    145952 obs. of  28 variables:
##  $ YEAR               : int  2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 ...
##  $ MONTH              : int  12 1 11 5 5 3 4 4 4 6 ...
##  $ DAY                : int  31 8 27 30 23 29 11 18 25 2 ...
##  $ DAY_OF_WEEK        : int  4 4 5 6 6 7 6 6 6 2 ...
##  $ AIRLINE            : chr  "F9" "OO" "AA" "OO" ...
##  $ FLIGHT_NUMBER      : int  668 6287 2207 5647 6508 6289 304 304 317 5459 ...
##  $ ORIGIN_AIRPORT     : chr  "SFO" "SFO" "SFO" "SFO" ...
##  $ DESTINATION_AIRPORT: chr  "DEN" "SBA" "DFW" "SBA" ...
##  $ SCHEDULED_DEPARTURE: int  2000 1350 1200 1840 2230 831 2135 2135 2145 900 ...
##  $ DEPARTURE_TIME     : int  1926 1318 1129 1814 2205 809 2113 2113 2123 838 ...
##  $ DEPARTURE_DELAY    : int  -34 -32 -31 -26 -25 -22 -22 -22 -22 -22 ...
##  $ TAXI_OUT           : int  14 52 43 36 39 25 23 15 20 35 ...
##  $ WHEELS_OFF         : int  1940 1410 1212 1850 2244 834 2136 2128 2143 913 ...
##  $ SCHEDULED_TIME     : int  150 71 208 75 92 63 84 84 124 66 ...
```

```
##  $ ELAPSED_TIME       : int  139 99 247 87 110 71 95 84 136 75 ...
##  $ AIR_TIME           : int  109 44 177 47 64 41 67 65 111 36 ...
##  $ DISTANCE           : int  967 262 1464 262 421 193 421 421 679 190 ...
##  $ WHEELS_ON          : int  2229 1454 1709 1937 2348 915 2243 2233 2334 949 ...
##  $ TAXI_IN            : int  16 3 27 4 7 5 5 4 5 4 ...
##  $ SCHEDULED_ARRIVAL  : int  2330 1501 1728 1955 2 934 2259 2259 2349 1006 ...
##  $ ARRIVAL_TIME       : int  2245 1457 1736 1941 2355 920 2248 2237 2339 953 ...
##  $ ARRIVAL_DELAY      : int  -45 -4 8 -14 -7 -14 -11 -22 -10 -13 ...
##  $ DIVERTED           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CANCELLED          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CANCELLATION_REASON: chr  "" "" "" "" ...
##  $ AIR_SYSTEM_DELAY   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ LATE_AIRCRAFT_DELAY: int  NA 0 NA NA NA NA NA NA NA NA ...
##  $ WEATHER_DELAY      : int  NA NA NA NA NA NA NA NA NA NA ...
```

Com podem veure tenim 28 columnes i un total de 145.592 dades en el dataset obtingut. També podem veure quines són les dades que conté el nostre dataset. Com que volem veure els retards o problemes que poden haver-hi, les causes i en quines aerolínies només cal que seleccionem aquelles columnes que ens proporcionin aquesta informació. En aquest cas serien:

```
col_interest = c(
  "DAY_OF_WEEK",
  "AIRLINE",
  "DEPARTURE_DELAY",
  "ARRIVAL_DELAY",
  "LATE_AIRCRAFT_DELAY",
  "DISTANCE"
)
print(col_interest)
```

```
## [1] "DAY_OF_WEEK"         "AIRLINE"             "DEPARTURE_DELAY"
## [4] "ARRIVAL_DELAY"       "LATE_AIRCRAFT_DELAY" "DISTANCE"
```

Amb aquestes dades ja podem fer un anàlisi complet per tal de donar resposta a les preguntes proposades.

```
flights <- subset(flights, select=col_interest)
```

3. Neteja de les dades. (2) (MR)

3.1. Les dades contenen zeros o elements buits? Com gestionaries aquests casos? (1)

Amb R és fàcil veure si tenim valors buits (NA) dins les nostres dades:

```
colSums(is.na(flights))
```

```
##         DAY_OF_WEEK             AIRLINE     DEPARTURE_DELAY       ARRIVAL_DELAY
##                   0                   0                   0                 461
## LATE_AIRCRAFT_DELAY            DISTANCE
##               93699                   0
```

Veiem alguns valors buits en la variable `ARRIVAL_DELAY`. Hi ha vàries estratègies per a resoldre problemes amb els elements buits, una tècnica eficaç és aplicar la funció kNN amb la qual omplirem els buits fent servir informació de k veïns més propers. Aquesta opció escollida es basa en que les variables del nostre dataset guarden certa relació i no són completament independents. Tindrem així uns valors aproximats als esperats que és millor que tenir-ne de buits. Com hi ha moltes dades, podriem eliminar els registres on `LATE_AIRCRAFT_DELAY` és nul.

```
flights <- flights[!is.na(flights$LATE_AIRCRAFT_DELAY),]
suppressWarnings(suppressMessages(library(VIM)))
flights$ARRIVAL_DELAY = kNN(flights)$ARRIVAL_DELAY
colSums(is.na(flights))
```

```
##         DAY_OF_WEEK             AIRLINE     DEPARTURE_DELAY       ARRIVAL_DELAY
##                   0                   0                   0                   0
## LATE_AIRCRAFT_DELAY            DISTANCE
##                   0                   0
```

Veiem que no tenim valors buits en les variables conflictives.

3.2. Identificació i tractament de valors extrems. (1)

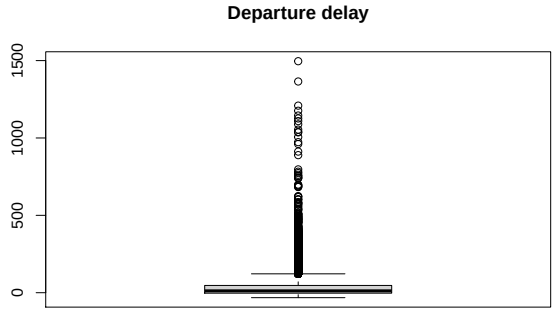
Començarem amb estadística descriptiva bàsica fent ús del summary que ens proporciona R. Aquí podem veure els valors màxims i mínims de cada variable.

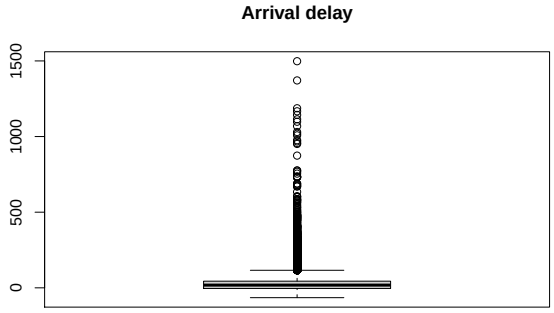
```
summary(flights)
```

```
##   DAY_OF_WEEK      AIRLINE          DEPARTURE_DELAY   ARRIVAL_DELAY
##  Min.   :1.000   Length:52253       Min.   : -32.00   Min.   : -65.00
##  1st Qu.:2.000   Class :character   1st Qu.:  -3.00   1st Qu.:  -4.00
##  Median :4.000   Mode  :character   Median :  12.00   Median :  18.00
##  Mean   :3.877                      Mean   :  32.21   Mean   :  30.74
##  3rd Qu.:5.000                      3rd Qu.:  47.00   3rd Qu.:  44.00
##  Max.   :7.000                      Max.   :1496.00   Max.   :1498.00
##  LATE_AIRCRAFT_DELAY    DISTANCE
##  Min.   :   0.00     Min.   :  77
##  1st Qu.:   0.00     1st Qu.: 372
##  Median :   0.00     Median : 679
##  Mean   :  16.47     Mean   :1145
##  3rd Qu.:  17.00     3rd Qu.:1855
##  Max.   :1102.00     Max.   :2704
```

D'aquí veiem alguns casos interessants, volem veure els que tenen mínims i màxims que s'allunyen clarament dels quartils (1r i 3r). Amb un boxplot podrem veure quants valors són extrems dins d'aquestes variables.

```
boxplot(flights$DEPARTURE_DELAY, main="Departure delay")
boxplot(flights$ARRIVAL_DELAY, main="Arrival delay")
```





Amb un gràfic de caixes podem veure clarament si tenim *outliers* o valors extrems a les dades. R representa els valors extrems com a cercles més enllà del rang interquartil. Aquests valors són normals ja que pot ser

que els vols hagin tingut gran retards. La raó per la qual quasi no podem veure la caixa (a prop de zero) és perquè la gran majoria de vols no tenen retard i els outliers coincideixen amb els vols que en tenen.

4. Anàlisi de les dades. (2.5) (MR)

4.1. Selecció dels grups de dades que es volen analitzar/comparar (planificació dels anàlisis a aplicar). (0.5)

- Com volem analitzar el retard, considerarem que els vols han tingut retard si la suma del retard de sortida més el d'arribada supera els 10 minuts. En cas que el retard hagi sigut de l'avió mirarem que sigui per sobre de 15 minuts.

```
flights$TOTAL_DELAY = flights$DEPARTURE_DELAY + flights$ARRIVAL_DELAY
flights <- within(flights, {
  DELAYED <- NA
  DELAYED[TOTAL_DELAY > 10] <- 1
  DELAYED[TOTAL_DELAY <= 10] <- 0
   }
)

flights <- within(flights, {
  late_delay_SFO <- NA
  late_delay_SFO[LATE_AIRCRAFT_DELAY > 15] <- 1
  late_delay_SFO[LATE_AIRCRAFT_DELAY <= 15] <- 0
   } )
```

- Analitzarem els vols també per aerolinia i per dia de la setmana. Categoritzem les variables.

```
flights$AIRLINE = factor(flights$AIRLINE)
flights$DAY_OF_WEEK_FAC = factor(flights$DAY_OF_WEEK)
levels = c("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun")
levels(flights$DAY_OF_WEEK_FAC) <- levels

flights$DELAYED = factor(flights$DELAYED)
levels = c("No", "Yes")
levels(flights$DELAYED) <- levels

flights$late_delay_SFO = factor(flights$late_delay_SFO)
levels = c("No", "Yes")
levels(flights$late_delay_SFO) <- levels
str(flights)
```

```
## 'data.frame':    52253 obs. of  10 variables:
##  $ DAY_OF_WEEK        : int  4 2 6 2 2 3 1 6 2 7 ...
##  $ AIRLINE            : Factor w/ 11 levels "AA","AS","B6",..: 7 6 7 2 8 7 7 9 7 7 ...
##  $ DEPARTURE_DELAY    : int  -32 -20 -20 -19 -19 -19 -19 -19 -19 -18 ...
##  $ ARRIVAL_DELAY      : int  -4 -19 -22 21 -7 -33 5 -50 -28 21 ...
##  $ LATE_AIRCRAFT_DELAY: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ DISTANCE           : int  262 2338 620 679 337 262 421 2521 158 1504 ...
##  $ TOTAL_DELAY        : int  -36 -39 -42 2 -26 -52 -14 -69 -47 3 ...
##  $ DELAYED            : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ late_delay_SFO     : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ DAY_OF_WEEK_FAC    : Factor w/ 7 levels "Mon","Tue","Wed",..: 4 2 6 2 2 3 1 6 2 7 ...
```

- Separarem en grups de vols llargs (més de 500 mi) de vols curts (menys de 500 mi).

```
long_index = flights$DISTANCE > 500
flights.long = flights[long_index,]
flights.short = flights[-long_index,]
```

4.2. Comprovació de la normalitat i homogeneïtat de la variància. (1)

Per fer la comprovació de la normalitat de les dades un dels mètodes més habituals és fer servir la funció de Shapiro-Wilk. Mirem en eles variables numèriques:

```
shapiro.test(flights[c(1:5000), "TOTAL_DELAY"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  flights[c(1:5000), "TOTAL_DELAY"]
## W = 0.93261, p-value < 2.2e-16
```

```
shapiro.test(flights[c(1:5000), "DAY_OF_WEEK"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  flights[c(1:5000), "DAY_OF_WEEK"]
## W = 0.92372, p-value < 2.2e-16
```

```
shapiro.test(flights[c(1:5000), "DISTANCE"])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  flights[c(1:5000), "DISTANCE"]
## W = 0.79216, p-value < 2.2e-16
```

Aquest test és més restrictiu que el test de Kolmogorov-Smirnov. Només podem presentar 5000 mostres en el test. Com el p valor que resulta és inferior a 0.05 es rebutja l'hipòtesi nul · la i considera que la distribució no és normal.

Veiem el test de l'homoscedasticitat per asegurar igualtat de variàncies. Aplicarem el test de Fligner-Killeen, que es tracta de l'alternativa no paramètrica, utilitzada quan les dades no compleixen amb la condició de normalitat.

```
fligner.test(TOTAL_DELAY~DISTANCE, data  = flights)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  TOTAL_DELAY by DISTANCE
## Fligner-Killeen:med chi-squared = 837.46, df = 79, p-value < 2.2e-16
```

```
fligner.test(TOTAL_DELAY~DAY_OF_WEEK, data  = flights)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  TOTAL_DELAY by DAY_OF_WEEK
## Fligner-Killeen:med chi-squared = 641.78, df = 6, p-value < 2.2e-16
```

Atès que les proves resulten en un p-valor inferior al nivell de significació ($< 0{,}05$), es rebutja la hipòtesi

nul · la d'homoscedasticitat i es conclou que les variables presenta variàncies estadísticament diferents per als diferents grups de spray.

4.3. Aplicació de proves estadístiques per comparar els grups de dades. En funció de les dades i de l'objectiu de l'estudi, aplicar proves de contrast d'hipòtesis, correlacions, regressions, etc. Aplicar almenys tres mètodes d'anàlisi diferents. (1)

Comprovarem si podem entendre el retard de sortida es pot entendre a partir del retard d'arribada i amb el de l'avió a través d'una regressió lineal. Podem afegir la distancia i el dia de la setmana per millorar el model?

```
# Agafem el valor de referencia de dia de la setmana dilluns
day_monday = relevel(factor(flights$DAY_OF_WEEK), ref = 1)

# Agafem com a aeroport de referencia el AA.
airline_aa=relevel(factor(flights$AIRLINE), ref = 'AA')

model1 <- lm(DEPARTURE_DELAY ~ ARRIVAL_DELAY + late_delay_SFO, data=flights)
model2 <- lm(DEPARTURE_DELAY ~ ARRIVAL_DELAY + late_delay_SFO + DISTANCE, data=flights)
model3 <- lm(DEPARTURE_DELAY ~ ARRIVAL_DELAY + late_delay_SFO + day_monday, data=flights)
model4 <- lm(DEPARTURE_DELAY ~ ARRIVAL_DELAY + late_delay_SFO + airline_aa, data=flights)


tabla.coeficientes <-matrix(c(1,summary(model1)$r.squared,2,summary(model2)$r.squared,3,summary(model3)$

colnames(tabla.coeficientes) <-c("Modelo", "R^2")
print(tabla.coeficientes)
```

```
##      Modelo       R^2
## [1,]      1 0.9405958
## [2,]      2 0.9418051
## [3,]      3 0.9407873
## [4,]      4 0.9433441
```

```
summary(model4)
```

```
##
## Call:
## lm(formula = DEPARTURE_DELAY ~ ARRIVAL_DELAY + late_delay_SFO +
##     airline_aa, data = flights)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -164.790   -6.544    1.207    7.904  268.310
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -1.516394   0.233085  -6.506 7.80e-11 ***
## ARRIVAL_DELAY       0.961966   0.001188 809.752  < 2e-16 ***
## late_delay_SFOYes   6.655755   0.155461  42.813  < 2e-16 ***
## airline_aaAS       -0.678734   0.398921  -1.701  0.08887 .
## airline_aaB6        3.174432   0.395768   8.021 1.07e-15 ***
## airline_aaDL        1.681012   0.345065   4.872 1.11e-06 ***
## airline_aaF9       -4.561676   0.540676  -8.437  < 2e-16 ***
## airline_aaHA      -12.289558   0.845159 -14.541  < 2e-16 ***
```

```
## airline_aaOO        0.182728   0.259577   0.704  0.48147
## airline_aaUA        6.298152   0.254282  24.768  < 2e-16 ***
## airline_aaUS        1.533259   0.468528   3.273  0.00107 **
## airline_aaVX        0.307630   0.296591   1.037  0.29964
## airline_aaWN        3.528508   0.301999  11.684  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.79 on 52240 degrees of freedom
## Multiple R-squared:  0.9433, Adjusted R-squared:  0.9433
## F-statistic: 7.248e+04 on 12 and 52240 DF,  p-value: < 2.2e-16
```

Obtenim una R-squared de 0.94 per a tots els models. El model 4 sembla que té una millor capacitat descriptiva, però tampoc és una millora significativa. Amb un valor p inferior al valor de significació podem dir que es pot explicar la variable DEPARTURE_DELAY amb ARRIVAL_DELAY i late_delay_SFO. Aquesta correlació ens indica que sabent el retard del vol d'arribada podriem predir el retard del següent vol de sortida.

Amb un model de regressió logística veurem si és possible obtenir el retard a partir del dia de la setmana i de la distancia. Preveurem només el retard de sortida:

```
flights <- within(flights, {
  delay_SFO <- NA
  delay_SFO[DEPARTURE_DELAY >= 15] <- 1
  delay_SFO[DEPARTURE_DELAY < 15] <- 0
   } )

flights$delay_SFO <- factor(flights$delay_SFO, levels = c(0, 1))
str(flights$delay_SFO)
```

```
##  Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

```
model5 <- glm(flights$delay_SFO ~ day_monday + flights$DISTANCE, family="binomial")
summary(model5)
```

```
##
## Call:
## glm(formula = flights$delay_SFO ~ day_monday + flights$DISTANCE,
##     family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2664  -1.1475  -0.9567   1.1842   1.4405
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        2.107e-01  2.472e-02   8.526  < 2e-16 ***
## day_monday2       -2.645e-01  3.188e-02  -8.296  < 2e-16 ***
## day_monday3       -4.514e-01  3.235e-02 -13.956  < 2e-16 ***
## day_monday4       -2.049e-01  3.095e-02  -6.621 3.56e-11 ***
## day_monday5       -1.885e-01  3.131e-02  -6.019 1.75e-09 ***
## day_monday6       -6.692e-01  3.590e-02 -18.641  < 2e-16 ***
## day_monday7       -1.745e-02  3.195e-02  -0.546    0.585
## flights$DISTANCE -5.238e-05  9.915e-06  -5.283 1.27e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 72354  on 52252  degrees of freedom
## Residual deviance: 71790  on 52245  degrees of freedom
## AIC: 71806
##
## Number of Fisher Scoring iterations: 4
```

Podem fer un segon test i respondre a la pregunta, "els vols llargs tenen més retards que els curts?", o dit d'altra manera, "la mitjana dels retards dels vols llargs és major a la dels vols curts?". Així plantegem el següent test de contrast d'hipòtesis de dos mostres sobre la diferencia de mitjanes, amb valor de significació $\alpha = 0.05$.

```
flights.short.late <- flights.short$TOTAL_DELAY
flights.long.late <- flights.long$TOTAL_DELAY
t.test(flights.short.late, flights.long.late, alternative = "less")
```

```
##
##  Welch Two Sample t-test
##
## data:  flights.short.late and flights.long.late
## t = -2.1617, df = 61928, p-value = 0.01532
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##        -Inf -0.4434798
## sample estimates:
## mean of x mean of y
##  62.94781  64.80285
```

Amb el valor de p superior a 0.05 no podem rebutjar la hipòtesis nul · la i per tant no podem concloure que siguin diferents.

Amb un anàlisis de correlació podem investigar quina de les variables influeix més en el retard dels vols.

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
columns_to_compare = c("DISTANCE", "DAY_OF_WEEK")
corr_matrix <-matrix(nc = 2, nr = 0)
colnames(corr_matrix) <-c("estimate", "p-value")


# Calculem el coeficient de correlació per a cada camp respecte la variable quantitativa TOTAL_DELAY
for (i in columns_to_compare) {
  spearman_test =cor.test(flights[,i], flights$TOTAL_DELAY, method = "spearman", exact = FALSE )
  corr_coef = spearman_test$estimate
```

```
  p_val = spearman_test$p.value
  # Add row to matrix
  pair =matrix(ncol = 2, nrow = 1)
  pair[1][1] = corr_coef
  pair[2][1] = p_val
  corr_matrix <-rbind(corr_matrix, pair)
  rownames(corr_matrix)[nrow(corr_matrix)] <-i
}
print(corr_matrix)
```

```
##                   estimate     p-value
## DISTANCE    -0.0005035221 0.908368289
## DAY_OF_WEEK -0.0128001600 0.003433205
```

La correlació és molt baixa en qualsevol cas i no veiem cap relació directa entre les variables explicatives i el retard.

5. Representació dels resultats a partir de taules i gràfiques. (2) (MF)

Un cop ja tenim les dades netes i preparades ens centrarem en les variables més importants i les quals ens ajudaran a respondre les preguntes més endavant. Les variables són les aerolínies i els dies de la setmana. D'aquesta forma, a continuació mostrarem les taules d'aquestes variables segons els retards.

```
table(flights$AIRLINE, flights$delay_SFO)
```

```
##
##         0    1
##   AA 2061 1541
##   AS  996  803
##   B6  917  920
##   DL 1664 1212
##   F9  374  421
##   HA  234   54
##   OO 6693 6597
##   UA 7769 8335
##   US  928  219
##   VX 3117 2333
##   WN 2423 2642
```

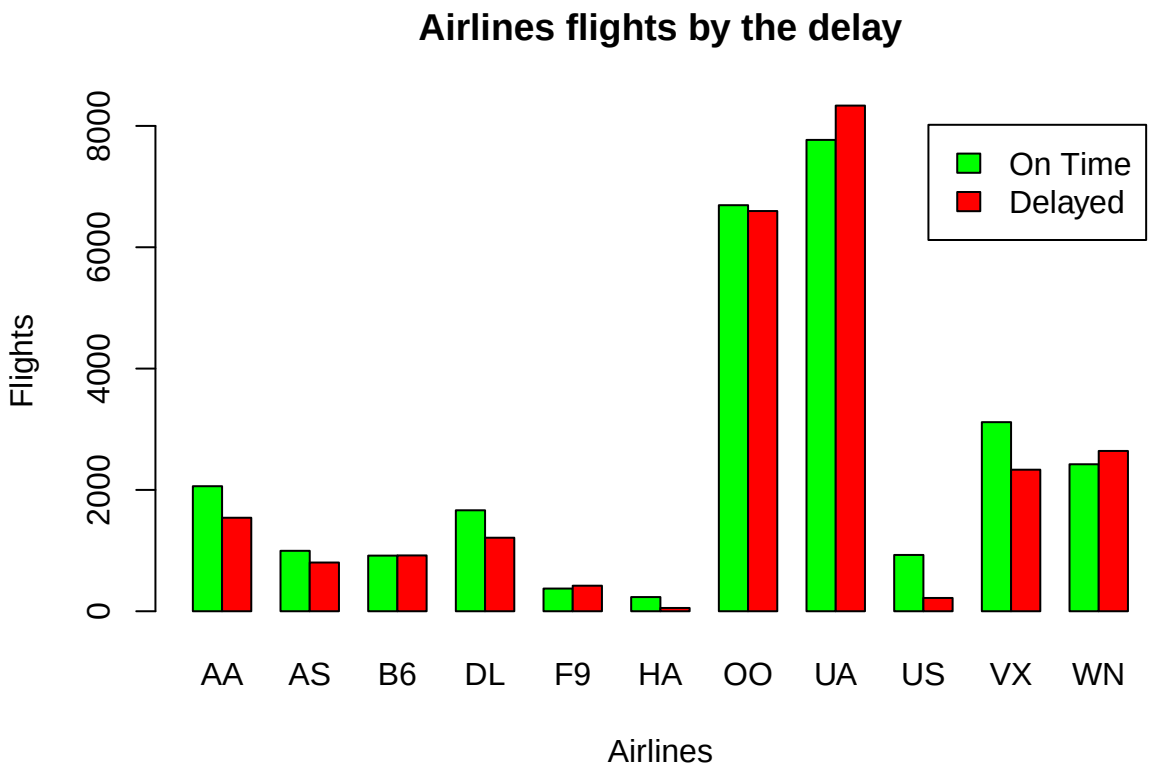
```
table(flights$DAY_OF_WEEK, flights$delay_SFO)
```

```
##
##        0    1
##   1 3871 4500
##   2 3962 3534
##   3 4159 3079
##   4 4312 4082
##   5 4084 3931
##   6 3316 1976
##   7 3472 3975
```

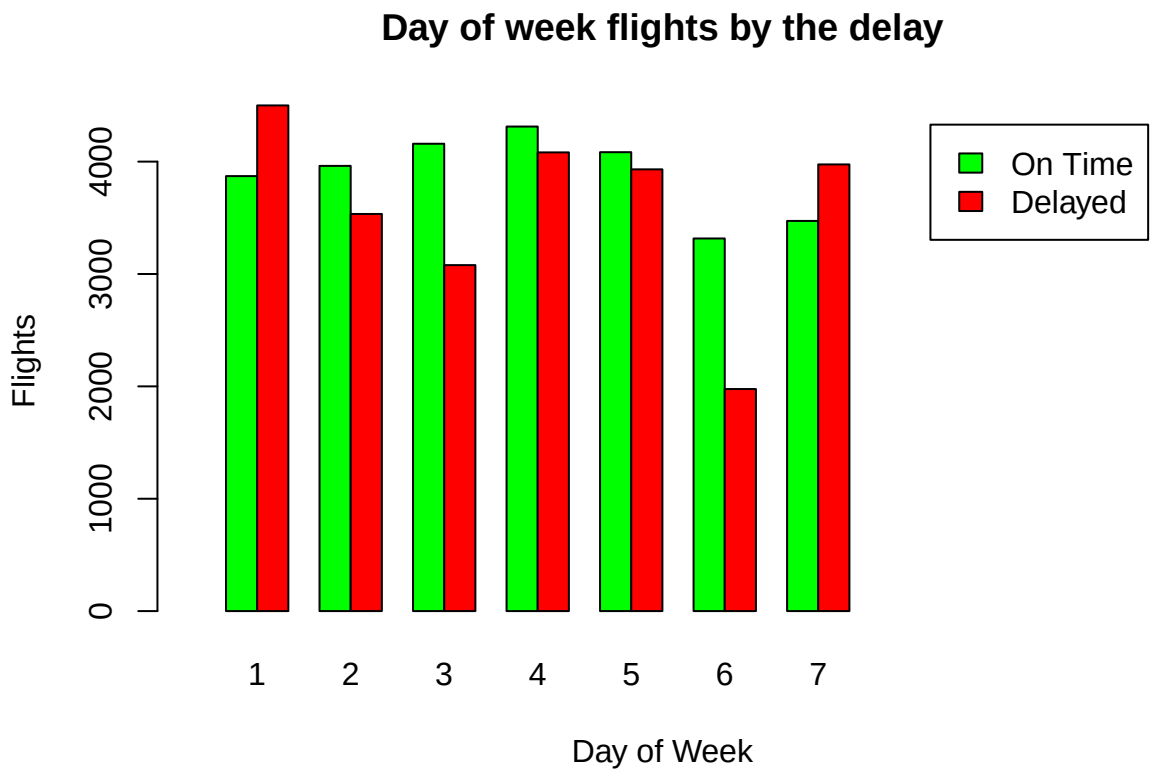
Com podem veure les dues taules ens mostren per a cada una de les variables el nombre de vols que han tingut retard i el nombre de vols que no n'han tingut. Partint d'aquestes taules que hem generat, representarem els valors utilitzant uns gràfics de barres de forma que podem veure més clarament la distribució de tots els

valors.

```
table_airlines <- table(flights$delay_SFO, flights$AIRLINE)
table_dayweek <- table(flights$delay_SFO, flights$DAY_OF_WEEK)
barplot(table_airlines, main="Airlines flights by the delay", xlab = "Airlines", ylab = "Flights", col =
```




```
barplot(table_dayweek, main="Day of week flights by the delay", xlab = "Day of Week", ylab = "Flights",
```



Com podem veure els gràfics realitzats ens mostren per cada una de les variables que volem el nombre de vols que no han tingut retards, de color verd, i el nombre de vols que han tingut retard, de color vermell, d'aquesta forma podem veure d'una manera molt ràpida i clara com estan distribuïts els valors en les variables escollides.

6. Resolució del problema. A partir dels resultats obtinguts, quines són les conclusions? Els resultats permeten respondre al problema? (0.5) (MR)

7. Contribucions al treball

Contribucions	Accuracy
Recerca prèvia	M.R./ M.F.
Redacció de les respostes	M.R./ M.F.
Desenvolupament codi	M.R./ M.F.

8. Codi: Cal adjuntar el codi, preferiblement en R, amb el que s'ha realitzat la neteja, anàlisi i representació de les dades. Si ho preferiu, també podeu treballar en Python. (2) (MF)