

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 als EUA. 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.

La pregunta o problema que es vol respondre és quines són les causes més comunes que han provocat els retards o les cancel·lacions dels vols i també veure quins són els llocs o el conjunt de destinacions que és més comú que es produeixi una cancel·lació o quines són les aerolínies més propenses a tenir cancel·lacions.

Aquest dataset ha sigut obtingut a partir d'una pràctica anterior realitzada durant el màster de Data Science 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  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        : Factor w/ 11 levels "AA","AS","B6",...: 5 7 1 7 7 7 2 2 2 7 ...
##  $ FLIGHT_NUMBER  : int   668 6287 2207 5647 6508 6289 304 304 317 5459 ...
##  $ ORIGIN_AIRPORT : Factor w/ 1 level "SFO": 1 1 1 1 1 1 1 1 1 1 ...
##  $ DESTINATION_AIRPORT: Factor w/ 80 levels "ABQ","ACV","ANC",...: 20 70 21 70 63 45 63 63 72 71 ...
##  $ 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: Factor w/ 4 levels "", "A", "B", "C": 1 1 1 1 1 1 1 1 1 1 ...
## $ 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(
  "AIRLINE",
  "SCHEDULED_DEPARTURE",
  "DEPARTURE_TIME",
  "DEPARTURE_DELAY",
  "SCHEDULED_TIME",
  "SCHEDULED_ARRIVAL",
  "ARRIVAL_TIME",
  "ARRIVAL_DELAY",
  "DIVERTED",
  "CANCELLED",
  "CANCELLATION_REASON",
  "AIR_SYSTEM_DELAY",
  "LATE_AIRCRAFT_DELAY",
  "WEATHER_DELAY",
  "DISTANCE"
)
print(col_interest)
```

```
## [1] "AIRLINE"          "SCHEDULED_DEPARTURE" "DEPARTURE_TIME"
## [4] "DEPARTURE_DELAY"  "SCHEDULED_TIME"      "SCHEDULED_ARRIVAL"
## [7] "ARRIVAL_TIME"     "ARRIVAL_DELAY"       "DIVERTED"
## [10] "CANCELLED"        "CANCELLATION_REASON" "AIR_SYSTEM_DELAY"
## [13] "LATE_AIRCRAFT_DELAY" "WEATHER_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))
```

```
##          AIRLINE SCHEDULED_DEPARTURE    DEPARTURE_TIME    DEPARTURE_DELAY
##              0              0              0              0
##    SCHEDULED_TIME    SCHEDULED_ARRIVAL    ARRIVAL_TIME    ARRIVAL_DELAY
##              0              0              173              461
##      DIVERTED          CANCELLED CANCELLATION_REASON    AIR_SYSTEM_DELAY
##              0              0              0              0
## LATE_AIRCRAFT_DELAY    WEATHER_DELAY    DISTANCE
##      93699          116418              0
```

Veiem molts valors buits en les columnes LATE_AIRCRAFT_DELAY i WEATHER_DELAY:

```
num_na = colSums(is.na(flights))["LATE_AIRCRAFT_DELAY"]
num_no_na = colSums(!is.na(flights))["LATE_AIRCRAFT_DELAY"]
print(num_no_na/(num_no_na+num_na)*100)
```

```
## LATE_AIRCRAFT_DELAY
##              35.8015
```

```
num_na = colSums(is.na(flights))["WEATHER_DELAY"]
num_no_na = colSums(!is.na(flights))["WEATHER_DELAY"]
print(num_no_na/(num_no_na+num_na)*100)
```

```
## WEATHER_DELAY
##              20.23542
```

Tenim només un 35% i un 20% de les dades no nul·les en aquests camps. Per a no introduir masses errors en l'estimació de nous valors, els exclourem del nostre anàlisi.

```
col_interest = c(
  "AIRLINE",
  "SCHEDULED_DEPARTURE",
  "DEPARTURE_TIME",
  "DEPARTURE_DELAY",
  "SCHEDULED_TIME",
  "SCHEDULED_ARRIVAL",
  "ARRIVAL_TIME",
  "ARRIVAL_DELAY",
  "DIVERTED",
  "CANCELLED",
  "CANCELLATION_REASON",
  "AIR_SYSTEM_DELAY",
  "DISTANCE"
)
print(col_interest)
```

```
## [1] "AIRLINE"          "SCHEDULED_DEPARTURE" "DEPARTURE_TIME"
## [4] "DEPARTURE_DELAY"  "SCHEDULED_TIME"     "SCHEDULED_ARRIVAL"
## [7] "ARRIVAL_TIME"     "ARRIVAL_DELAY"      "DIVERTED"
## [10] "CANCELLED"        "CANCELLATION_REASON" "AIR_SYSTEM_DELAY"
## [13] "DISTANCE"
```

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

com també alguns valors buits en altres variables com ARRIVAL_TIME o ARRIVAL_DELAY. Hi ha diverses 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. No té massa sentit però aplicar això a la variable `ARRIVAL_TIME`.

```
suppressWarnings(suppressMessages(library(VIM)))
flights$ARRIVAL_DELAY = kNN(flights)$ARRIVAL_DELAY

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

```
##          AIRLINE SCHEDULED_DEPARTURE DEPARTURE_TIME DEPARTURE_DELAY
##              0              0              0              0
## SCHEDULED_TIME SCHEDULED_ARRIVAL ARRIVAL_TIME ARRIVAL_DELAY
##              0              0              173              0
## DIVERTED CANCELLED CANCELLATION_REASON AIR_SYSTEM_DELAY
##              0              0              0              0
## DISTANCE
##              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.

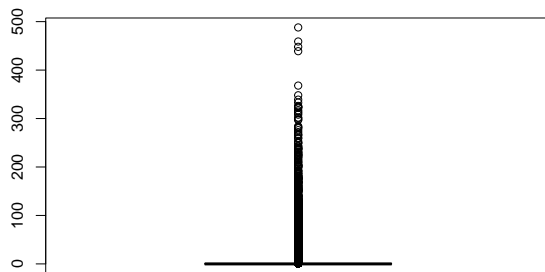
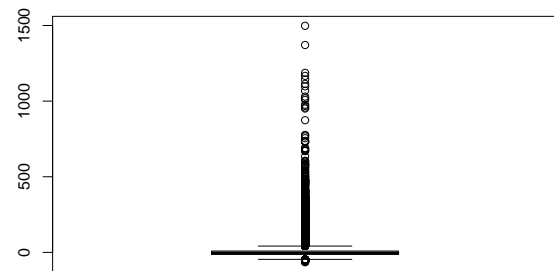
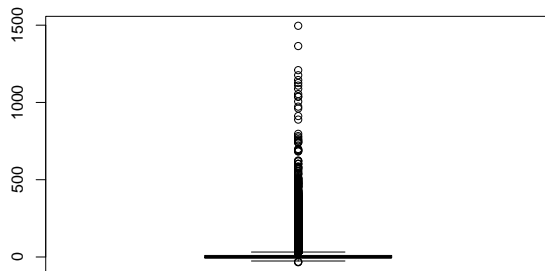
```
summary(flights)
```

```
##          AIRLINE SCHEDULED_DEPARTURE DEPARTURE_TIME DEPARTURE_DELAY
## UA      :45135 Min. : 5 Min. : 1 Min. : -34.00
## OO      :34223 1st Qu.: 920 1st Qu.: 925 1st Qu.: -5.00
## VX      :15848 Median :1319 Median :1328 Median : -1.00
## WN      :13790 Mean :1366 Mean :1367 Mean : 11.19
## AA      :12062 3rd Qu.:1800 3rd Qu.:1805 3rd Qu.: 10.00
## DL      : 9659 Max. :2359 Max. :2400 Max. :1496.00
## (Other):15235
## SCHEDULED_TIME SCHEDULED_ARRIVAL ARRIVAL_TIME ARRIVAL_DELAY
## Min. : 43.0 Min. : 1 Min. : 1 Min. : -65.000
## 1st Qu.: 90.0 1st Qu.:1135 1st Qu.:1120 1st Qu.: -13.000
## Median :126.0 Median :1602 Median :1556 Median : -4.000
## Mean :179.6 Mean :1503 Mean :1487 Mean : 5.814
## 3rd Qu.:277.0 3rd Qu.:1948 3rd Qu.:1946 3rd Qu.: 9.000
## Max. :367.0 Max. :2359 Max. :2400 Max. :1498.000
## NA's :173
## DIVERTED CANCELLED CANCELLATION_REASON AIR_SYSTEM_DELAY
## Min. :0.000000 Min. :0.000000 :145860 Min. : 0.000
## 1st Qu.:0.000000 1st Qu.:0.000000 A: 39 1st Qu.: 0.000
## Median :0.000000 Median :0.000000 B: 46 Median : 0.000
## Mean :0.002528 Mean :0.0006303 C: 7 Mean : 1.973
## 3rd Qu.:0.000000 3rd Qu.:0.000000 3rd Qu.: 0.000
## Max. :1.000000 Max. :1.000000 Max. :488.000
##
## DISTANCE
## Min. : 77
## 1st Qu.: 414
## Median : 679
## Mean :1201
```

```
## 3rd Qu.:2139
## 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$DEPARTURE_DELAY)
boxplot(flights$ARRIVAL_DELAY)
boxplot(flights$AIR_SYSTEM_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. També existeix la funció `boxplot.stats()` amb la qual es mostren només els valors atípics de les variables que les contenen.

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

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

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

7. 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)