PH125.9x Data Science. Capstone. Barcelona Airport (BCN) Performance

Jordi Candela

February 2022

Introduction

This file contains the second exercise, an original and personally built from scratch exercise, of the edx course PH125.9x. Capstone of the Data Science Professional Certificate of HarvardX.

The exercise analizes the performance of a particular airport, Barcelona in the example, thanks to the data achieved through a data scrapping algorithm that gets the data supplied by flightradar.com.

Flightradar.com is a webpage that acquires the data from the aircraft transponders and delivers information of each flight of the world that uses a transponder. This data, in parallel, is compared with the airlines scheduled, delivering information about delays, cancelled flights, flight times, etc.

In the case of the airport of Barcelona, the airport suffered massive delays since 2016 until the beginning of the Covid-19 pandemy, being very difficult to allocate the responsibilities of such disruptions.

The exercise aims to find if Barcelona airport delays follow any particular relationship with the major data from flightradar, wich are mainly the flightcode, the scheduled time, or the airline.

To perform this exercise, only airport departures will be considered, since they have a miminum value of delay equal to zero and may provide a better picture in the event of poor airort performance (airport arrivals are usually more influenced by the airport origin than by the destination airport).

Several linear regression will be tested, trying to correlate delays with scheduled time under different data sets. On the top of that, a machine learning algorith will be also tested to see if there is a trustworthy estimation of delays from the variables of the dataset.

The different methodologies used follow the contents of the previous courses of the program, and mainly the techniques explained in the Linear Regression and Machine Learning courses, PH125.7x and PH125.8x.

Overview

The data used in this exercise are the total number of departures of Barcelona airport of September 2019, which was on the busiest months in the hostory of the airport. This data is extracted into an xlx file that becomes the initial dataset for the exercise.

The first stage of the work consists in understanding the structure of the dataset and the nature of the data. Under this topic, the major airlines of the airport is achieved and a benchark of delays between the major airlines of the airport is performed.

Once the major features of the dataset are undertood, some data transformation are required in order to be able to analyst the correlation between delays and a particular hour of the day.

Moreover, additional complementary datasets are built in order to identify whether there is a specific trend for the dominant carrier of the airport (Vueling in the case of Barcelona) or by the major carriers of the airport.

The next stage is performing several linear regression exercises in order to find clear correlations of delays with time of operation.

Finally, a machine learning algorithm is performed trying to minimize the RMSE, starting with a naive approach, which is complemented by trying to minimize the RMSE correlation the delay with the most relevant variables of the dataset.

Executive Summary

The results of the exercise conclude that there is not a clear relationship between delays and time of operation or the airline, showing a quite homogeneous dispersion as long as the operation begins. This could be explained either by the lack of the relevant variable that leads to a delay (for instance, weather conditions) or because of the fact that the airport suffers structural delays as long as the daily operations start.

Methods & Analysis

Techniques - approach

Methodologically, the different steps of the exercise are summarized as follows:

1. Installation of R packages.
2. Reading the datafile.
3. Data tyding.
4. Understanding the dataset.
5. Linear regression models.
6. Machine learning algorithm

- **1. Installation of R packages**

```
# Package Installs
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if(!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.org")
if(!require(lubridate)) install.packages("lubridate", repos = "http://cran.us.r-project.org")
if(!require(stringr)) install.packages("stringr", repos = "http://cran.us.r-project.org")
if(!require(recosystem)) install.packages("recosystem", repos = "http://cran.us.r-project.org")
if(!require(kableExtra)) install.packages("kableExtra", repos = "http://cran.us.r-project.org")
if(!require(readxl)) install.packages("readxl", repos = "http://cran.us.r-project.org")

library(tidyverse)
library(caret)
library(data.table)
library(lubridate)
library(stringr)
library(recosystem)
library(kableExtra)
library(readxl)
```

- **2. Reading the xlsx datafile**

Once installed the required package, the dataset is read and a sample of the file is achieved in order to understand the data structure of the file:

```
BCN_September_2019 <- read_excel("C:/Users/Jordi Candela/Desktop/Edx_CP_Data Science_Harvard/
9.- PH125.9x Data Science. Capstone/2.- own project/BCN_September_2019.xlsx")
```

- **3. Data tyding**

In order to ease the further manipulation of the data several operations contains NA values, which will be omitted in this analysis, since they belong to flight that finally are not being flown: canceled, diverted, descheduled, etc. and my distort the final findings of the work.

```
BCN_September_2019<- na.omit(BCN_September_2019)
```

- **4. Understanding the dataset**

The initial dataset is a table with 14.862 airport departures, each regiser / row corresponding to a single departure.

The different variables of the dataset are:

- Flight (chr): Flight code of the operation.
- From (chr): Departure airport IATA code. Always BCN in this case.
- To (chr): Destination airport IATA code.
- AL_CODE (chr): Airline IATA code.
- AC_TYPE (chr): Aircraft type.
- STATUS (chr): Indicates if the flight has effectively departed.
- DATE (POSIXct): Date of operation.
- STD (POSIXct): Scheduled time of operation.
- ATD (POSIXct): Actual time of operation.
- fr22_dd (num): Operation delay in minutes (ATD-STD)
- AIRLINE (chr): Airline commercial name

```
head(BCN_September_2019)
```

```
## # A tibble: 6 x 11
##   FLIGHT FROM  TO    AL_CODE AC_TYPE STATUS   DATE
##   <chr>  <chr> <chr> <chr>   <chr>   <chr>    <dttm>
## 1 ZF7756 BCN   VKO   ZF      B763    departed 2019-09-05 00:00:00
## 2 ZF7756 BCN   VKO   ZF      B763    departed 2019-09-06 00:00:00
## 3 U27208 BCN   LPL   U2      A319    departed 2019-09-02 00:00:00
## 4 U27208 BCN   LPL   U2      A319    departed 2019-09-09 00:00:00
## 5 VY1702 BCN   VGO   VY      A20N    departed 2019-09-04 00:00:00
## 6 EW5921 BCN   VIE   EW      A319    departed 2019-09-04 00:00:00
## # ... with 4 more variables: STD <dttm>, ATD <dttm>, fr24_dd <dbl>,
## #   AIRLINE <chr>
```

Understanding delays in relation to airlines The following step is to understand the distribution of the traffic grouped by airlines, what shows that there is a lot of dispersion within in many carriers.

```
# Behaviour per airlines and delays
BCN_September_2019_airline <- BCN_September_2019 %>%
  group_by(AIRLINE) %>%
  summarise (mean_delay = mean (fr24_dd), n=n())
str(BCN_September_2019_airline)
```

```
## tibble [108 x 3] (S3: tbl_df/tbl/data.frame)
##  $ AIRLINE   : chr [1:108] "Aegean Airlines" "Aer Lingus" "Aero4M" "Aeroflot" ...
##  $ mean_delay: num [1:108] 32.7 38.9 17 29.3 44.9 ...
##  $ n         : int [1:108] 47 83 1 124 13 3 3 53 59 1 ...
```

Thus, the previous dataset is filtered in order to work with the major carriers. In the case of Barcelona airport there are only 19 carriers that had more than 100 departures in September 2019, and thus have enough critical mass to be analysed.

```
# Filter of Airlines with more than 100 monthly departures
BCN_September_2019_airline <- filter(BCN_September_2019_airline, n >= 100)
BCN_September_2019_airline
```

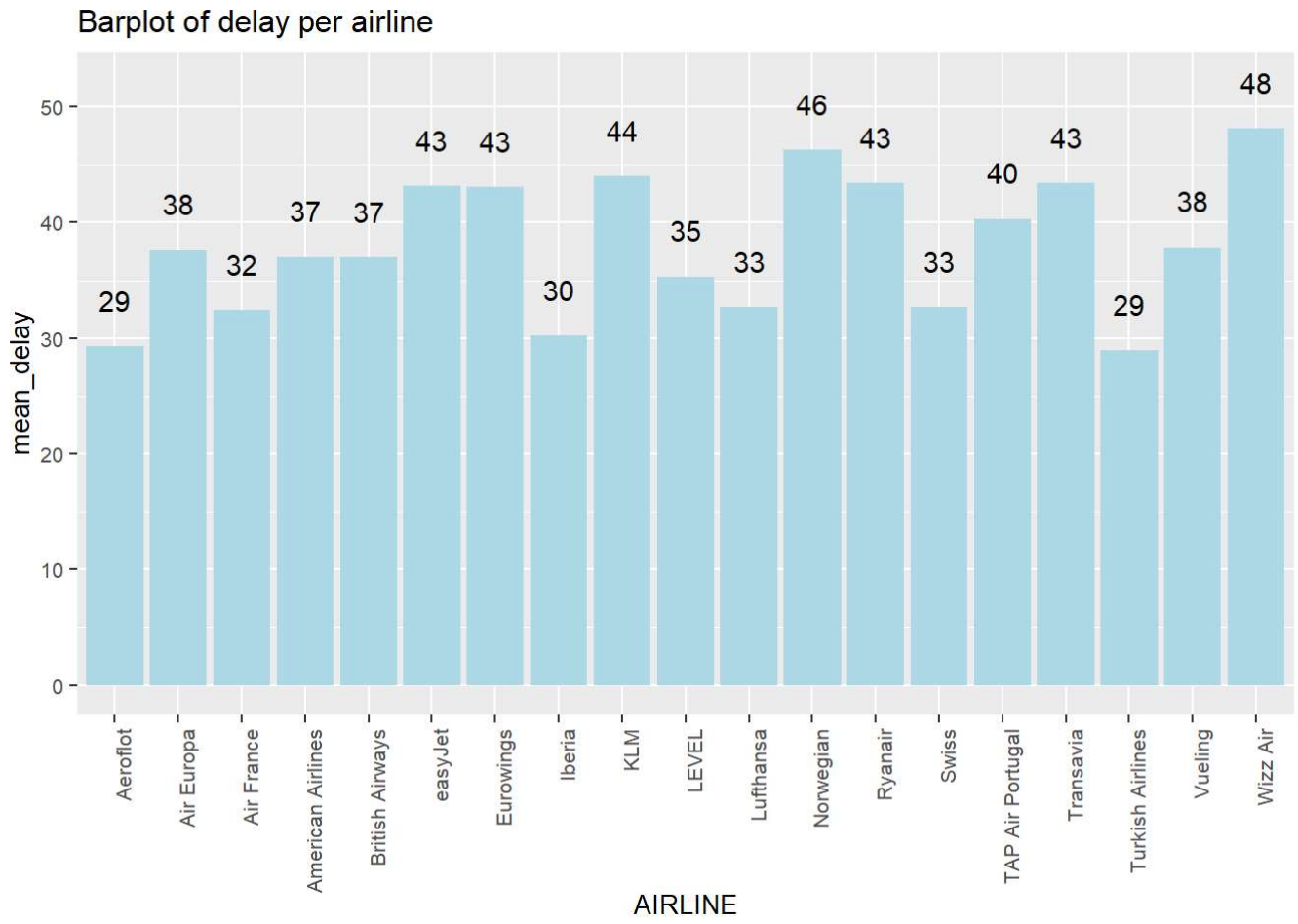
```
## # A tibble: 19 x 3
##    AIRLINE           mean_delay     n
##    <chr>                  <dbl> <int>
##  1 Aeroflot                29.3   124
##  2 Air Europa              37.6   238
##  3 Air France              32.4   204
##  4 American Airlines       37.0   148
##  5 British Airways         37.0   220
##  6 easyJet                 43.2  1083
##  7 Eurowings               43.0   240
##  8 Iberia                  30.2   352
##  9 KLM                     44.0   151
## 10 LEVEL                   35.3   138
## 11 Lufthansa               32.6   415
## 12 Norwegian               46.3   332
## 13 Ryanair                 43.4  1971
## 14 Swiss                   32.7   153
## 15 TAP Air Portugal        40.3   225
## 16 Transavia               43.4   168
## 17 Turkish Airlines        29.0   112
## 18 Vueling                 37.9  6112
## 19 Wizz Air                48.2   258
```

The next step is to arrange the top airlines that have better performances, which are shown in the corresponding barplot .

```
#Obtaining the major airlines with better performance
top_airlines_arranged <- arrange(BCN_September_2019_airline, mean_delay)
top_airlines_arranged
```

```
## # A tibble: 19 x 3
##    AIRLINE           mean_delay     n
##    <chr>                  <dbl> <int>
##  1 Turkish Airlines        29.0   112
##  2 Aeroflot                29.3   124
##  3 Iberia                  30.2   352
##  4 Air France              32.4   204
##  5 Lufthansa               32.6   415
##  6 Swiss                   32.7   153
##  7 LEVEL                   35.3   138
##  8 British Airways         37.0   220
##  9 American Airlines       37.0   148
## 10 Air Europa              37.6   238
## 11 Vueling                 37.9  6112
## 12 TAP Air Portugal        40.3   225
## 13 Eurowings               43.0   240
## 14 easyJet                 43.2  1083
## 15 Ryanair                 43.4  1971
## 16 Transavia               43.4   168
## 17 KLM                     44.0   151
## 18 Norwegian               46.3   332
## 19 Wizz Air                48.2   258
```

```
#Barplot arplot delay of TOP airlines (n > 100 monthly departures)
top_airlines_arranged %>%
  ggplot(aes(AIRLINE, mean_delay))+
  geom_col(fill="lightblue") +
  labs(title="Barplot of delay per airline")+
  geom_text(aes(label = signif(mean_delay, digits = 2)), nudge_y = 4)+
  theme(text = element_text(size=10),
        axis.text.x = element_text(angle=90, hjust=1))
```

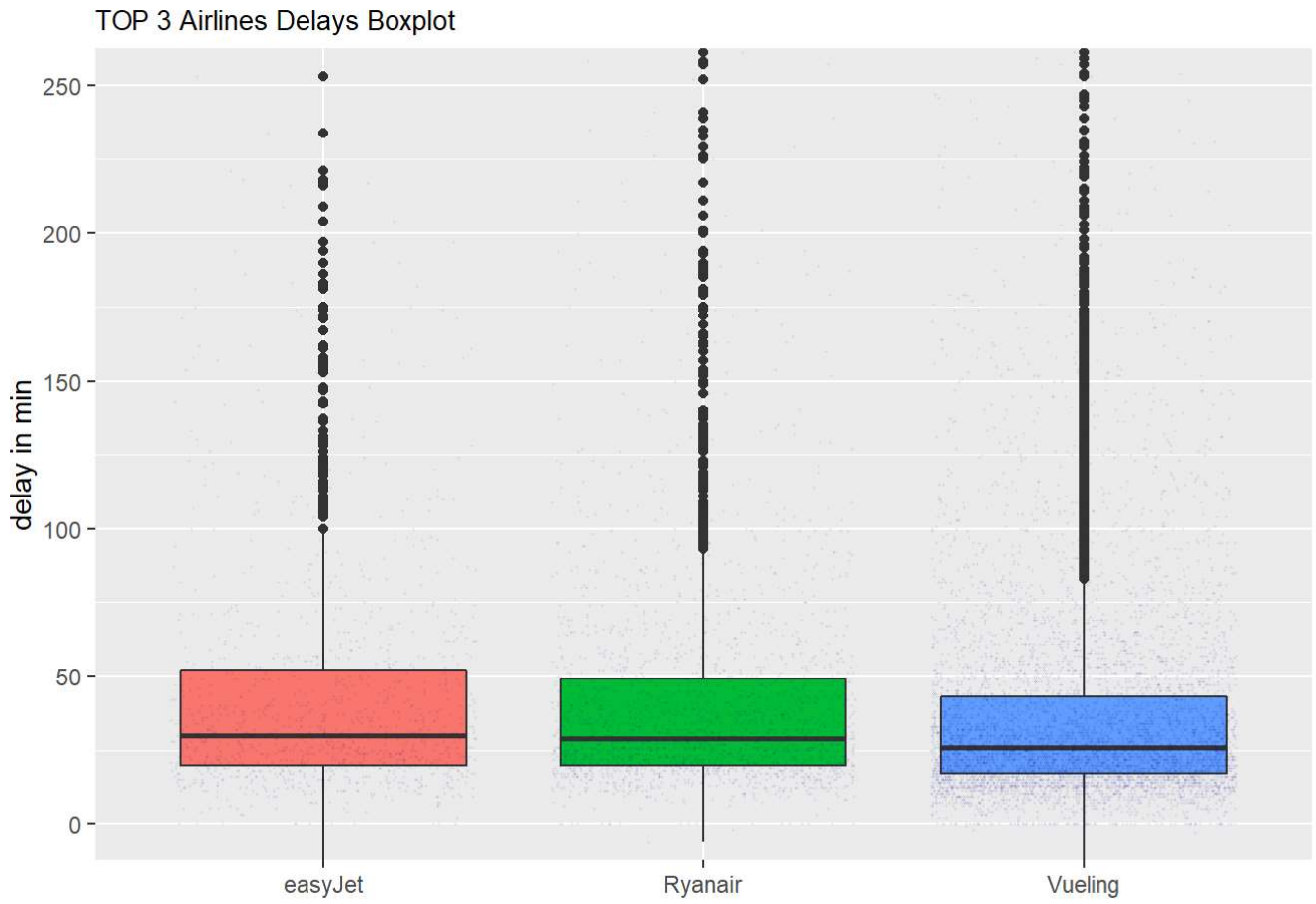


According to the date, one would conclude that there are is a remarkable difference between airlines, with a better performance of legacy carriers (flag or national carriers like Iberia, Air France, Lufthansa).

Besides, the breakdown of traffic of the airport shows a concentration around three major players: Vueling, Ryanair and Easyjet, whose delays are particularly analyzed through a boxplot. It is curious the fact the results of the boxplot shows a very similar performance of the three major players of the airport, although the local social perception is that Vueling is the airline that usually records poorer performances in Barcelona. Data denies this perception.

```
# Behavior of TOP3 Airlines
BCN_September_2019_TOP3 <- filter(BCN_September_2019, AIRLINE %in% c("easyJet", "Ryanair", "V
ueling"))

BCN_September_2019_TOP3 %>%
  ggplot(aes(x=AIRLINE, y=fr24_dd, fill=AIRLINE)) +
  geom_boxplot() +
  coord_cartesian(ylim=c(0,250))+
  geom_jitter(color="darkblue", size=0.2, alpha=0.05) +
  theme(
    legend.position="none",
    plot.title = element_text(size=10)
  ) +
  ggtitle("TOP 3 Airlines Delays Boxplot") +
  xlab("") +
  ylab("delay in min")
```



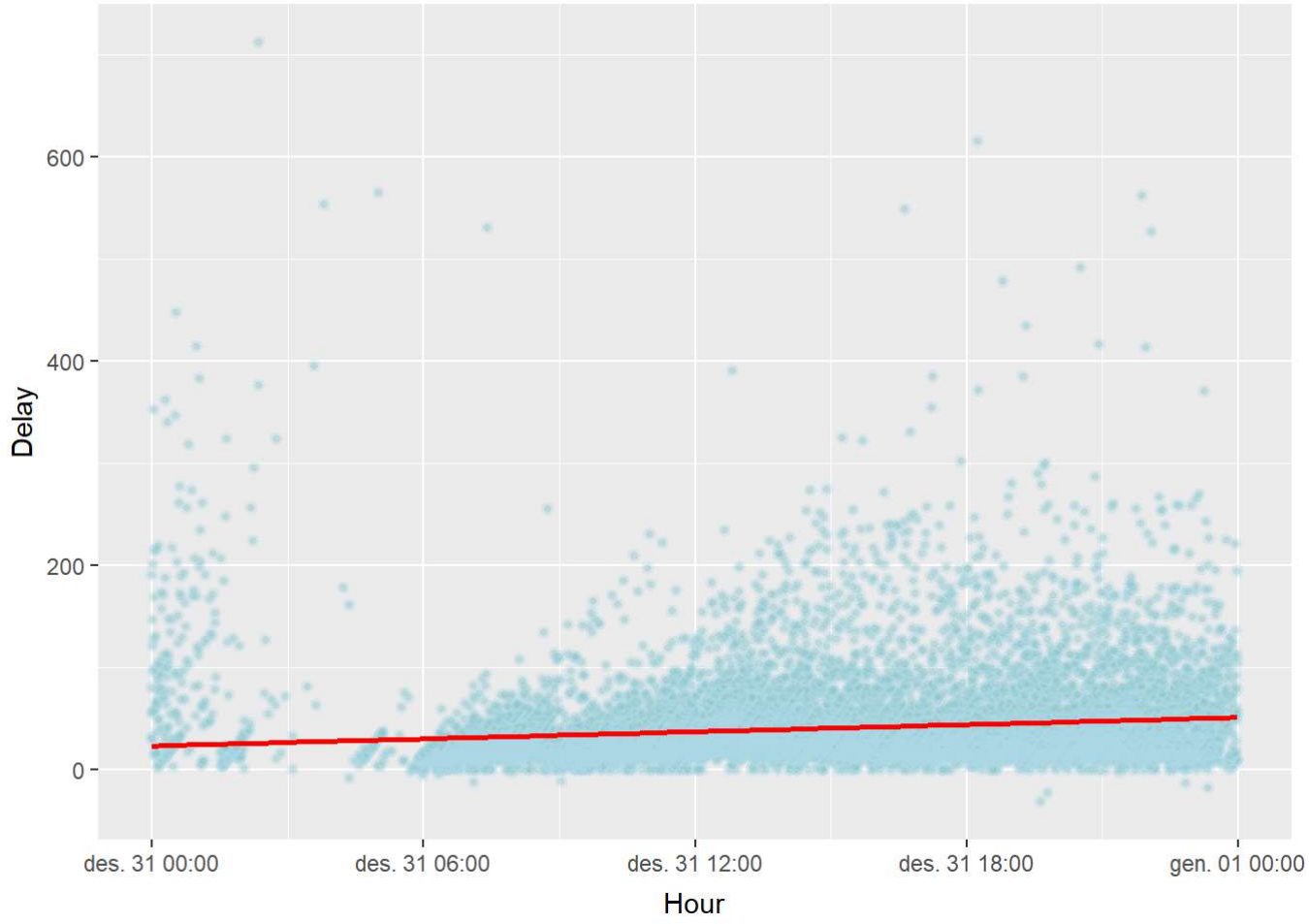
Understanding delays in relation to hour of operation It make sense to continue the analysis only considering the airlines that account more than 100 monthly operations in Barcelona, since smaller carriers would add a lot of dispersion to the results due to its outlying performance usually originated by external factors not related to Barcelona airport.

It is important to understand the spread of the evolution of the delays along the day, which is shown in the next plot.

```
# Evolution of delays within the day
dailydelay <- ggplot(BCN_September_2019, aes(x=ATD, y= fr24_dd)) +
  geom_point(
    color="lightblue",
    fill="#69b3a2",
    shape=21,
    alpha=0.3,
    size=0.1,
    stroke = 2
  )+
  geom_smooth(method=lm , color="red", se=FALSE) +
  labs(x = "Hour",y = "Delay")+
  theme(axis.title.x = element_text(vjust=-0.5, size=rel(1))) +
  theme(axis.title.y = element_text(vjust=1.5, size=rel(1)))

dailydelay
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Since the airport operations start from 6:00 AM in the morning, it is more accurate to follow the analysis not considering outlying operations from 0 AM to 6 AM, which are usually made by cargo operators and follow very different patterns and business models compared to passenger operations.

In order to do that, and considering that the information of the time of operation is related each one to a particular date, the hour of operation has been extracted and a new dataset filtering the hours above 6 AM is built.

```
# Evolution of delays within the day without considering early morning/late night
BCN_September_2019_nomorning <- cbind(BCN_September_2019, hour = hour(BCN_September_2019$AT
D))

BCN_September_2019_nomorning <- BCN_September_2019_nomorning %>% filter (BCN_September_2019_n
omorning$hour >= 6)

head(BCN_September_2019_nomorning)
```

```
##   FLIGHT FROM  TO AL_CODE AC_TYPE   STATUS       DATE                STD
## 1 ZF7756  BCN VKO      ZF    B763 departed 2019-09-05 1899-12-31 20:10:00
## 2 ZF7756  BCN VKO      ZF    B763 departed 2019-09-06 1899-12-31 20:10:00
## 3 U27208  BCN LPL      U2    A319 departed 2019-09-02 1899-12-31 23:38:00
## 4 U27208  BCN LPL      U2    A319 departed 2019-09-09 1899-12-31 23:04:00
## 5 VY1702  BCN VGO      VY    A20N departed 2019-09-04 1899-12-31 07:20:00
## 6 EW5921  BCN VIE      EW    A319 departed 2019-09-04 1899-12-31 09:15:00
##                   ATD fr24_dd   AIRLINE hour
## 1 1899-12-31 19:38:00     -32  Azur Air   19
## 2 1899-12-31 19:47:00     -23  Azur Air   19
## 3 1899-12-31 23:20:00     -18   easyJet   23
## 4 1899-12-31 22:50:00     -14   easyJet   22
## 5 1899-12-31 07:07:00     -13   Vueling    7
## 6 1899-12-31 09:03:00     -12 Eurowings    9
```

- **5. Linear regression models**

The methodology applied follows the contents of the course PH125.7x Linear Regression, coordinated by the professor Rafael Irizarry.

The first linear regression aims to find the relation between the delay and the hour of operation, delivering poor results of the R.

```
# Linear regression delay vs hour without considering early morning / late night
lmdelay = lm(fr24_dd ~ hour, data = BCN_September_2019_nomorning)
summary(lmdelay)
```

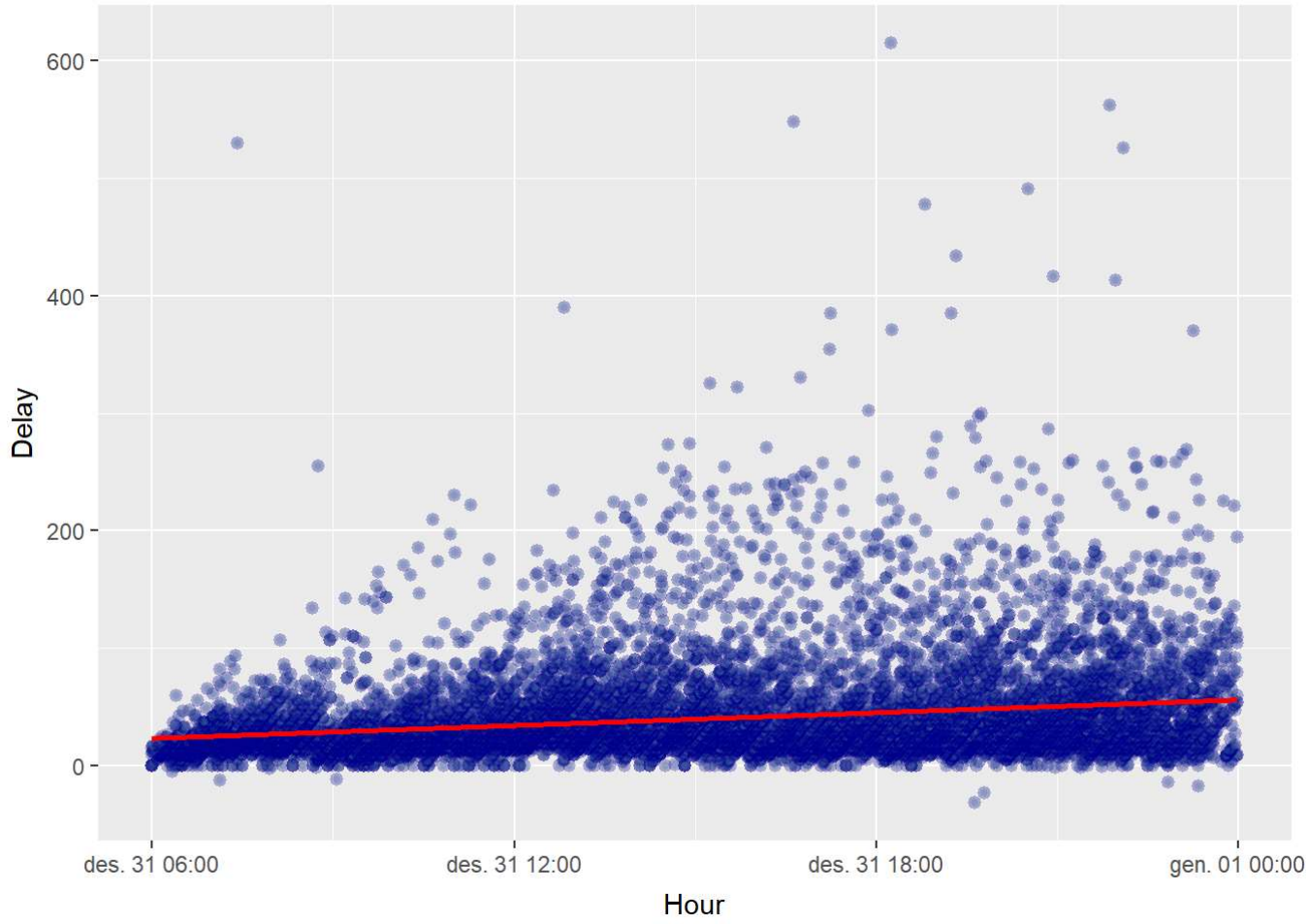
```
##
## Call:
## lm(formula = fr24_dd ~ hour, data = BCN_September_2019_nomorning)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -79.60 -19.26  -8.26   5.93 569.21
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.09110    0.93376   14.02   <2e-16 ***
## hour         1.81643    0.06189   29.35   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37.28 on 14515 degrees of freedom
## Multiple R-squared:  0.05602,    Adjusted R-squared:  0.05595
## F-statistic: 861.3 on 1 and 14515 DF,  p-value: < 2.2e-16
```

The plot shows the huge dispersion of the delay is continuous along the day.

```
# Plot of Delays and Linear regression delay of departures vs hour without considering early
 morning / late night
dailydelay_nomorning <- ggplot(BCN_September_2019_nomorning, aes(x=ATD, y= fr24_dd)) +
  geom_point(
    color="darkblue",
    fill="#69b3a2",
    shape=21,
    alpha=0.3,
    size=0.1,
    stroke = 2
  )+
  geom_smooth(method=lm , color="red", se=FALSE) +
  labs(x = "Hour",y = "Delay")+
  theme(axis.title.x = element_text(vjust=-0.5, size=rel(1))) +
  theme(axis.title.y = element_text(vjust=1.5, size=rel(1)))

dailydelay_nomorning
```

```
## `geom_smooth()` using formula 'y ~ x'
```



A second linear regression aims to put light on the performance of the major carrier of the airport, Vueling, who eventually may have a singular pattern due to the important amount of based aircrafts.

However, the results of this second exercise of linear regression still show a poor correlation.

```
### Linear regression 2
# Linear regression delay of Vueling vs hour without considering early morning / late night
BCN_September_2019_nomorning_vy <- BCN_September_2019_nomorning %>% filter (BCN_September_201
9_nomorning$AIRLINE %in% c("Vueling"))
lmdelay_vy = lm(fr24_dd ~ hour, data = BCN_September_2019_nomorning_vy)
summary(lmdelay_vy)
```

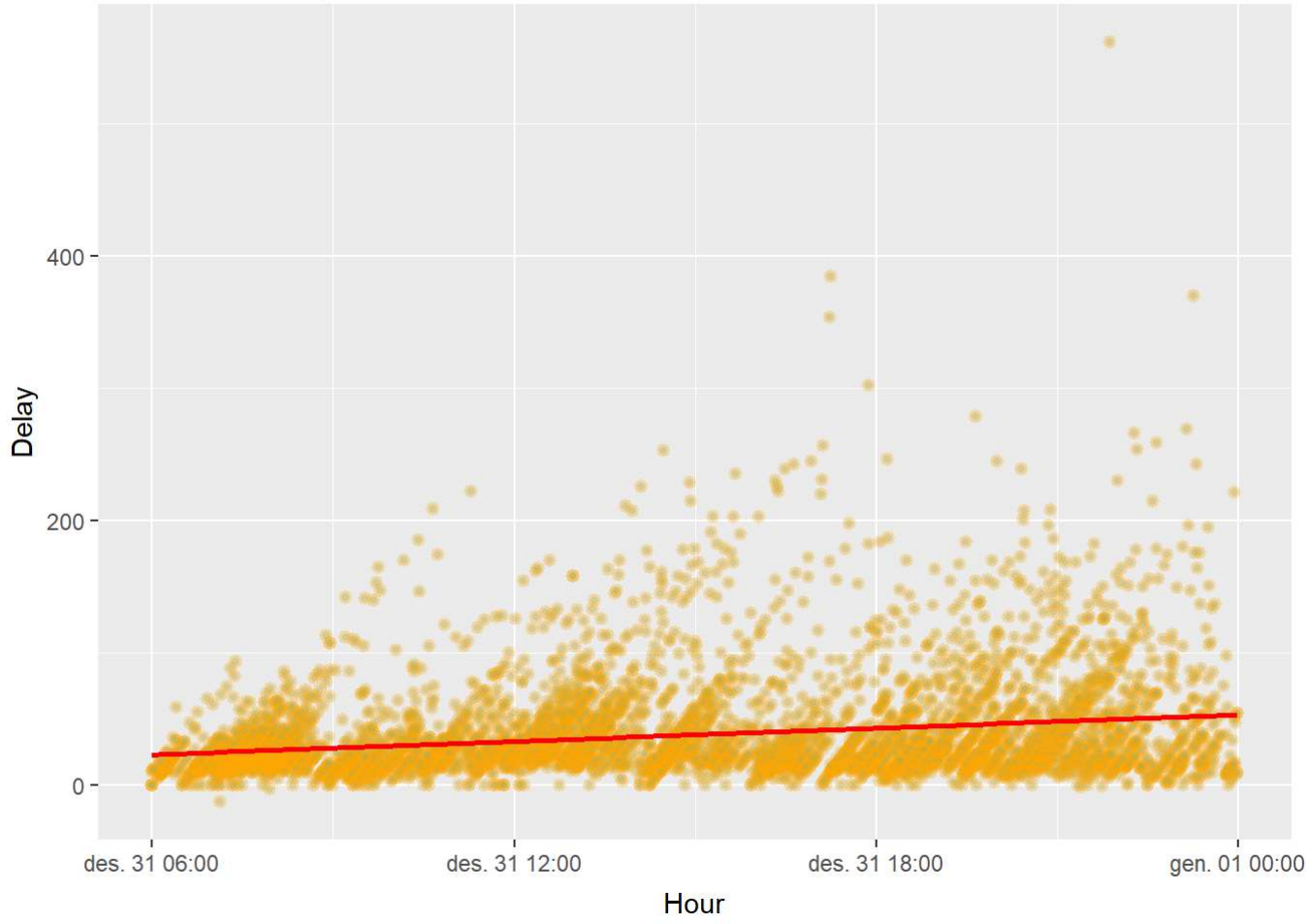
```
##
## Call:
## lm(formula = fr24_dd ~ hour, data = BCN_September_2019_nomorning_vy)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -52.19 -19.12  -7.47   5.91 513.16
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.68950    1.28554   10.65   <2e-16 ***
## hour         1.67380    0.08696   19.25   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.89 on 6008 degrees of freedom
## Multiple R-squared:  0.05808,    Adjusted R-squared:  0.05792
## F-statistic: 370.5 on 1 and 6008 DF,  p-value: < 2.2e-16
```

The dispersion of Vueling's delays along the day is shown in the plot below.

```
# Plot of Delays and Linear regression delay of Vueling vs hour without considering early mor
ning / late night
dailydelay_nomorning_vy <- ggplot(BCN_September_2019_nomorning_vy, aes(x=ATD, y= fr24_dd)) +
  geom_point(
    color="orange",
    fill="#69b3a2",
    shape=21,
    alpha=0.3,
    size=0.1,
    stroke = 2
  )+
  geom_smooth(method=lm , color="red", se=FALSE) +
  labs(x = "Hour",y = "Delay")+
  theme(axis.title.x = element_text(vjust=-0.5, size=rel(1))) +
  theme(axis.title.y = element_text(vjust=1.5, size=rel(1)))

dailydelay_nomorning_vy
```

```
## `geom_smooth()` using formula 'y ~ x'
```



- **6. Machine learning algorithm**

The methodology applied follows the contents of the chapter 6.2, Recommendation Systems, of the course PH125.8x Machine Learning, coordinated by the professor Rafael Irizarry.

The target of the modelling when estimating is to minimize the error function of the estimation function, the Residual MEan Square Error (RMSE).

Lower values of RMSE mean higher accuracies of the prediction. Thus the approach will follow a successive approximation approach, trying the most relevant variables and, in the event that an improvement is achieved within a particular variable, progressively adding new variables/bias that contribute to decrease values of RMSE.

0.- Preparation of the dataset

The machine learning datased is created from BCN_September_2019 data and considering a validation dataset of 10% of the data.

```
# Validation set will be 10% of BCN_September_2019 data
set.seed(1, sample.kind="Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
test_index <- createDataPartition(BCN_September_2019_nomorning$fr24_dd, p = 0.1, list = FALSE
)
tester_bcn <- BCN_September_2019_nomorning[-test_index,]
validation_bcn <- BCN_September_2019_nomorning[test_index,]
```

i.- Simple Naive Prediction

The simple or naive prediction model does not consider any bias ; in other words, does not consider the “quality” of a particular airlines

Therefore, the model strictly uses the mean of the dataset to predict the delay, assuming that all differences are due to a random error;

$$Y_{u,i} = \mu + \epsilon_{u,i}$$

where $Y_{u,i}$ is the prediction, $\epsilon_{u,i}$ is the independent error, and μ the expected “true” delay for all flights Predicting the mean gives the following naive RMSE.

```
### Simple Naive Prediction based on Mean Delay
mu <- mean(tester_bcn$fr24_dd)
mu
```

```
## [1] 39.06384
```

```
rmse_naive <- RMSE(tester_bcn$fr24_dd, mu)
rmse_naive
```

```
## [1] 38.76736
```

```
# Create a Data Tibble and Save Results
rmse_results = tibble(method = "Naive Analysis by Mean", RMSE = rmse_naive)
rmse_results %>% knitr::kable()
```

method	RMSE
Naive Analysis by Mean	38.76736

ii.- Airline Effects(b_i) Prediction

The Hour Effects Prediction considers that the prediction of a rating of a delay has a bias linked to airline.

This bias of the airline (b_i) is estimated as the difference between the airline mean rating and the overall mean rating.

The mathematical formula of this model is stated as:

$$Y_{u,i} = \mu + b_i + \epsilon_{u,i}$$

where $Y_{u,i}$ is the prediction, $\epsilon_{u,i}$ is the independent error, and μ the mean delay for all fligths, and b_i is the bias for each airline i.

```
### Simple model taking into account the airline effects (b_i)
mu <- mean(tester_bcn$fr24_dd)
airline_avgs <- tester_bcn %>%
  group_by(AIRLINE) %>%
  summarise(b_i = mean(fr24_dd - mu))
predicted_delays <- mu + validation_bcn %>%
  left_join(airline_avgs, by='AIRLINE') %>%
  pull(b_i)
rmse_model_airline_effects <- RMSE(predicted_delays, validation_bcn$fr24_dd)
rmse_model_airline_effects
```

```
## [1] 34.1493
```

```
rmse_results <- bind_rows(rmse_results,
                          tibble(method="Airline Effects Model",
                                     RMSE = rmse_model_airline_effects))
rmse_results %>% knitr::kable()
```

method	RMSE
Naive Analysis by Mean	38.76736
Airline Effects Model	34.14930

iii.- Hour Effects(b_h) Prediction

The Hour Effects Prediction considers that the prediction of a rating of a delay has a bias linked to hour of the flight.

This bias of the hour (b_h) is estimated as the difference between the hour mean rating and the overall mean rating.

The mathematical formula of this model is stated as:

$$Y_{u,h} = \mu + b_h + \epsilon_{u,h}$$

where $Y_{u,h}$ is the prediction, $\epsilon_{u,h}$ is the independent error, and μ the mean delay for all fligths, and b_h is the bias for each hour i.

```
### Simple model taking into account the hour effect (b_h)
mu <- mean(tester_bcn$fr24_dd)
airline_avgs <- tester_bcn %>%
  group_by(hour) %>%
  summarise(b_h = mean(fr24_dd - mu))
predicted_delays <- mu + validation_bcn %>%
  left_join(airline_avgs, by='hour') %>%
  pull(b_h)
rmse_model_hour_effects <- RMSE(predicted_delays, validation_bcn$fr24_dd)
rmse_model_hour_effects
```

```
## [1] 33.24636
```

```
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Hour Effects Model",
                                     RMSE = rmse_model_hour_effects))
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
rmse_results %>% knitr::kable()
```

method	RMSE
Naive Analysis by Mean	38.76736
Airline Effects Model	34.14930
Hour Effects Model	33.24636

Both the airline (b_i) and hour (b_h) improve the initial naive model, achieving a lower RMSE value, although with very slight results that lead to the conclusion that with the information from flightradar.com cannot be successfully deployed a machina learning algorithm to prodict the delays of Barcelona airport.

Summary of results The values of the R achieved in the linear models show values close to 0, illustrating the no-relationship of delays with the airlines or the hour of operation.

The results applying the machine learning algorithm are consistent with the linear regression models, which are stated below;

```
rmse_results %>% knitr::kable()
```

method	RMSE
Naive Analysis by Mean	38.76736
Airline Effects Model	34.14930
Hour Effects Model	33.24636

Results and Conclusions

- **Summmary**

The performance of Barcelona airport, although it is very difficult to be modelled and thus predicted from the public data available in platforms such flightradar.com; can be objectively explained and understood through the analysis of available information.

Some examples of non-evident conclusions of this report are:

- Legacy carriers in Barcelona suffer less delays tha low-cost carriers - The three major airlines of Barcelona, all low-cost carriers, follow very similar delay patterns, although socially Vueling is perceived as the less punctual airline, whereas Easyjet is perceived the most efficient of the three and really has the lowest performance (This can be surely explained by the efffect of media, PR and social media). - Along the day, delays follow a stagnated patter as soon as the first operational wave of the airport begins, which is usually about 6:00 AM.

On the other hand, the different methodologies applied to try to identify behaviors or patters of delays correlated to data such as arline of time of operation do not provide sturdy conclusions.

- **Limitations**

The major limitation faced in the study is the lack of computational power to include mode data in the analysis, such as more operational months.

- **Future Work**

As stated before, adding more variables in future operations would add new textures that could help achieving more accurate estimations and relationships of the airport delays, otherwise the all the conclusions would tend to depict an scenario with structural airport delays and thus a lack of airport operational capacity.

Examples of this could include meteorological conditions (wind speed, wind direction, rain, etc.) or operational conditions from the tower of control.

Thanks to IOT, additional data from GPS receivers will be more easily attached and complement the intitial set of variables used.

- **References**

Irizarry,R., 2019. Introduction to Data Science.

Irizarry,R., github site: rafalab