Consultancy Report

US Flights Delays

Introduction to Programming

Fall Semester 2018/2019

Group 4

Haorun Cheng nr. 13881

João Fernandes nr. 14066

Maria Henriques nr. 14317

Table of Content

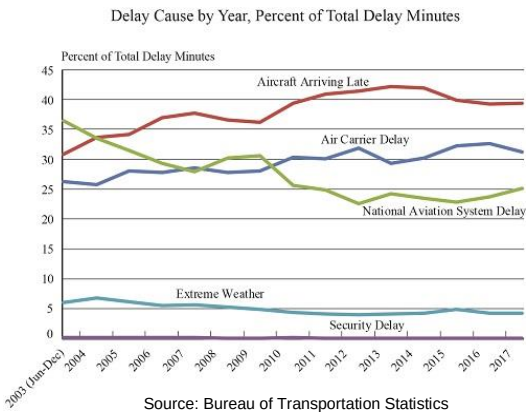
Introduction

This report aims to understand the delays in domestic flights in the USA, by analyzing "the temporal evolution of the delays, their geographical distribution, patterns in the aggravation or even to identify if the times of delays follow some known probabilistic distribution".[1] In order to achieve this goal, it was necessary to evaluate data related to the flights' "Time", such as arrival (ARR_TIME) and departure (DEP_TIME) time, "Destination", like the origin (ORIGIN) and destination (DEST) airports, and also "Performance", for what concerns delays (ARR_DELAY; DEP_DELAY) and airtime (AIR_TIME). This data was made available by the US Government Department of Transportation, that provides information on all domestic flights from 1987 to the present[1]. For the purpose of this report, it was only used data referred to January 2018.

Background

Airline companies carry around 10 billion passengers worldwide, per year, a number that is expected to double by 2035. The United States represents, alone, 536 million passengers, per year, and has seen the largest number of passengers ever (800 million) in 2016. However, even though this is an industry that represents 3,5% of the world GDP, it has seen significant financial losses, estimated at $ 33 billion/ year, due to delays of passengers' flights. In fact, "in US domestic flights alone, in the last 10 years, on average, 19% of flights were delayed by more than 15 minutes, which corresponds to more than 800,000 backward flights".[1]

In general, there are five reasons for delays of passengers' flights that are more evident. The first one, and the most problematic one, is "Late-arriving Aircraft" or, in other words, a flight that is obligated to depart late as the previous flight, with the same aircraft, arrived late. This problem represented, on average, in the last 10 years,



Source: Bureau of Transportation Statistics

[1] GroupWork_Case

around 40% of the delays' causes. The second and third reasons are "Air Carrier" and the "National Aviation Systems", which represent 30% and 25% of delays, respectively. While "Air Carrier" consist of delays caused by circumstances within the airline's control, "NAS" represents delays instigated by the national aviation system. The fourth and fifth reasons are not that relevant as they only represent 5% and 0,1% for delays causes. They are "Extreme Weather" and "Security", respectively.[2] More values related to this issue can be found in the appendix.

Methodology

In order to put this project forward, it was necessary to download the required datasets. For that purpose, three files were transferred from the Introduction to Programing's Moodle page: flights.csv, holidays.csv, and carriers.csv. The first step was, then, to import all the libraries that could become useful, such as pandas, matplotlib.pyplot, datetime, and others. After having all the datasets needed, it was time to evaluate the importance of columns with a high percentage of null attributes. As the column 'Unnamed:23' had solely none – *NaN* – values, the column was considered irrelevant and, therefore, was eliminated. With the purpose of evaluating the importance of other columns with null values, all null numbers were summed up, per column, using the expression *flights.isnull ( ).sum(axis=0)*. In consequence of the results obtained, the column 'TAIL_NUM' was removed, using *flights.drop (['name of column'], 1)*, as it was dispensable. In what concerns the other columns that had a fair amount of null values, these were not removed as they were quite relevant for the study in hands. In order to resolve this problem, the null values were substituted by the respective columns' mean. Therefore, *meanx = int (flights ['column name'].mean ( ), values = {'column name': meanx, ...}, flights = flights.fillna (value=values)*. With the purpose of testing if all values had been correctly substituted, a *query* was defined: *query = flights['ARR_TIME'] == None*. If the query is shown no results, which ended up happening, then the column being tested has no longer null values.

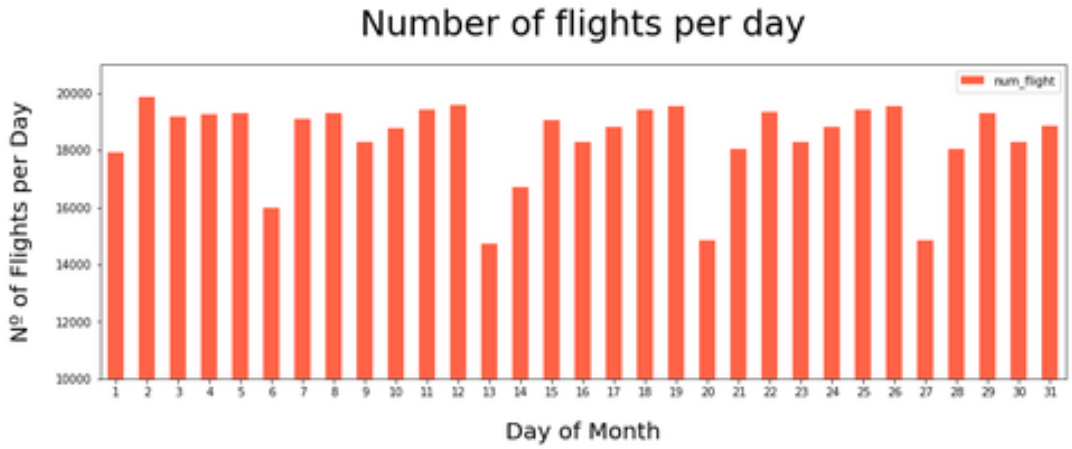
The next step was to obtain information about the US air transportation sector size as well as data related to delays.[1]

[2] Bureau of Transportation Statistics: https://www.bts.gov/topics/airlines-and-airports/understanding-reporting-causes-flight-delays-and-cancellations

Data Analysis (Jan 2018)

- Number of Flights per Day

The 570,188 flights that occurred in January 2018 were distributed along the month with a certain seasonality. While, on one hand, Mondays, Thursday and Fridays had, normally, the biggest number of flights, Saturdays, on the other hand, had the lowest amount. The only exception found was in the first week, when Monday had one of the lowest numbers. This can be easily explained as that specific Monday was January 1st, New Year's Day. Saturday's lower numbers can also be easily explained as people typically like to take advantage of weekends to go on vacations. Therefore, they are more prompted to travel on Thursday or Fridays with that purpose.



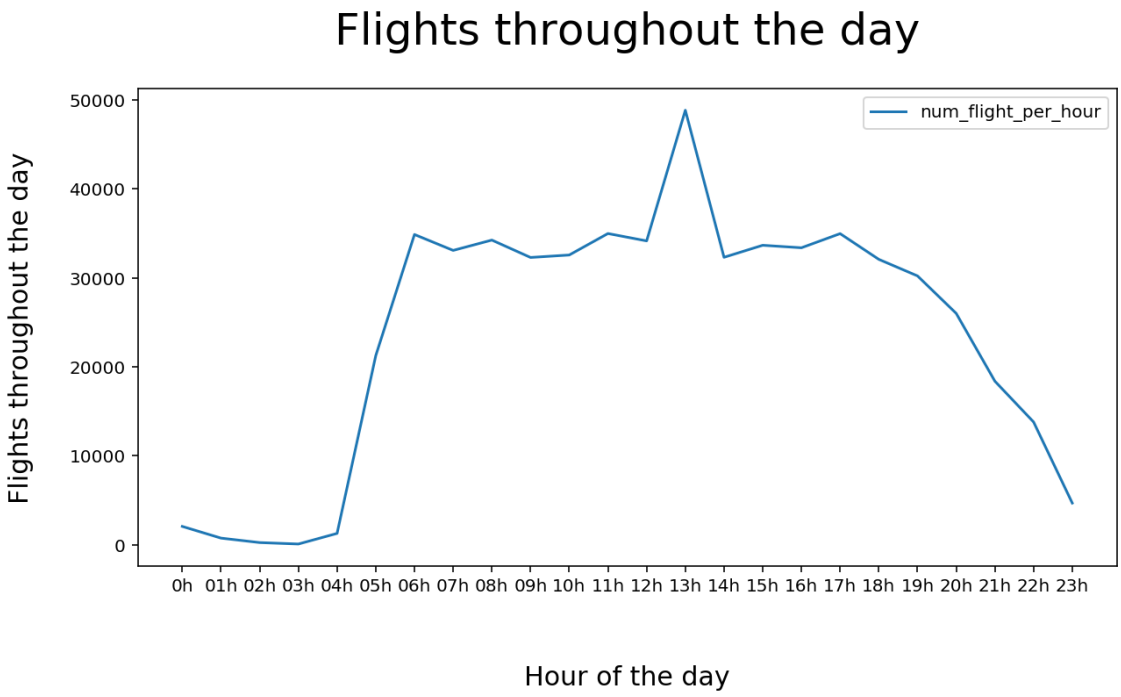
It is also interesting to observe that, accordingly to some data analysis[3], Saturdays are the best days to travel, as they represent the least amount of delays (18,11%, on average). Piecing these two concepts together, it makes sense that, on days with less air traffic, the number of delays will also be smaller. On the other hand, Fridays seem to be the worst day to travel, as it represents 29,75%, on average, of flights delays.

In order to obtain the number of flights per day, it was necessary to group the data by day. In other words, it was necessary to create a new variable that would be grouped by the 'DAY_OF_MONTH'. For that purpose, it was used a *flights.groupby (by= 'DAY_OF_MONTH').size( )* command. In order to obtain the graph above, a *flight_per_day.plot.bar* instruction was given.

[3] https://triphackr.com/how-to-avoid-flight-delays/

- Number of Flights per Hour

On average, during the month of January of 2018, there were 766 flights per hour. After evaluating the graph presented below, it is clear that the number of flights varies a lot with the hour of the day. It is also possible to verify that during midnight through 4 A.M. the number of flights is substantially low. In contrast, the highest peak can be found at 1 P.M., with almost 1000 times more flights than those at 3 A.M. Nevertheless, the most stable periods of the day are from 6 A.M. to 12 P.M and between 2 P.M. and 7 P.M. Amid 4 A.M. – 6 A.M. and 7 P.M. – 12 A.M is when the highest increases can be found.



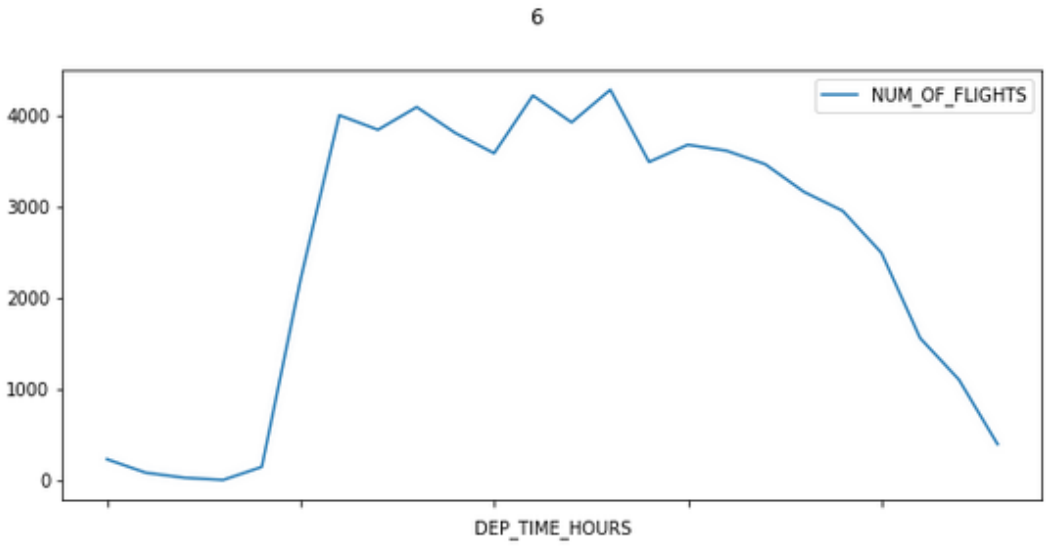
Once again, accordingly to data studies[4], the best time of the day to travel is at 6 A.M., when not that many flights have occurred and therefore, there is not yet a big dependence on arrival times. It also makes sense that, as the day passes, more and more flights are going to depart and arrive and, consequently, the probability of having a delay will also be higher. Nevertheless, after 6 P.M. – time at which delays reach its peak – as the number of flights begins to diminish, the number of delays also decreases.

In order to obtain the average number of flights per hour a simple division of the total number of flights by 31 and 24 was carried out. As in the 'hours' were appearing

[4] https://www.businessinsider.com/best-time-of-day-to-fly-to-avoid-delays-2015-9

both 00h and 24h, it was necessary to create a function called “transform” that would convert ‘24h00m’ values into ‘00h00m’ values. With the purpose of obtaining the number of flights per departure hour, it was necessary to create a new column – ‘DEP_TIME_HOURS’ – where a lambda would be applied in order to use “transform” function having as inputs all the values of the ‘DEP_TIME’ row. Since times such as 00h05 would only be shown as ‘5’ in this new column, a value of 10000 was added to every value of the said column and just the second and third letter were selected. The following step was to group the values by the respective hours, using *groupby*.

Evaluating the number of flights, per hour, per day of week, it was possible to observe that the pattern is similar among them. Nevertheless, Saturday (day 6, as referred in the graph’s title) is the exception where there was a regular fluctuation within the period of 6 A.M. to 3 P.M., differing from others where a peak increase is present at 1 P.M.



The remaining graphs will be in the Appendix.

- Flights’ Duration

On average, a flight trip takes 111 minutes. In order to be easier to evaluate the distribution of the flights’ duration, a categorization system was adopted. Using this system, flight’ times were divided into three sections: “Short-haul flights: Under 180 minutes”, “Medium-haul flights: amid 180 to 360 minutes”, and “Long-haul flights: between 360 and 720 minutes”. This division was decided based on the document “Flight

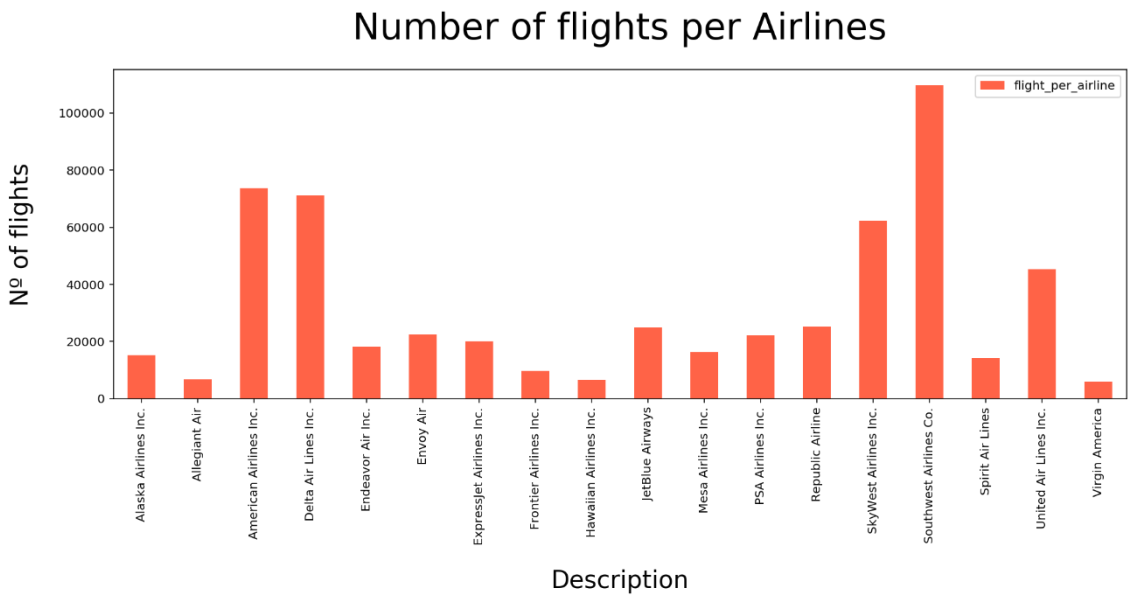
Length" that can be found here [5][5]. Evaluating the results obtained, it was possible to observe that the majority of flights – more precisely 490,513 flights or 86.03% – had a duration of less than 3 hours. A way smaller number of flights – 77,260 or 13.55% – took between 3 to 6 hours. The remaining 2345 flights (0,41%) took more than 6 hours to be completed. In the file used to characterize flights' duration, it is also mention flights with more than 720 minutes. However, this section was not used as it was possible to assess that, in January 2018, there were no flights taking more than 12 hours.

In order to obtain these results, it was necessary to, in the first instance, group together the number of flights in accordance with the respective 'AIR_TIME'. For that purpose, a "groupby" function was used, once again. The next step was, then, to sum the number of flights that belonged to the same category. In other words, all flights with a duration smaller than 180 minutes were summed together. (1st Step: filter through the query, *Short_haul_flight = flights_duration ['AIR_TIME']<180*; 2nd Step: Sum all the filtered values, using *.sum ( ).*) The procedure was repeated for the other 2 groupings. To find the respective percentages, a simple division by the total of flights, 570118, was made.

- Airlines Analysis

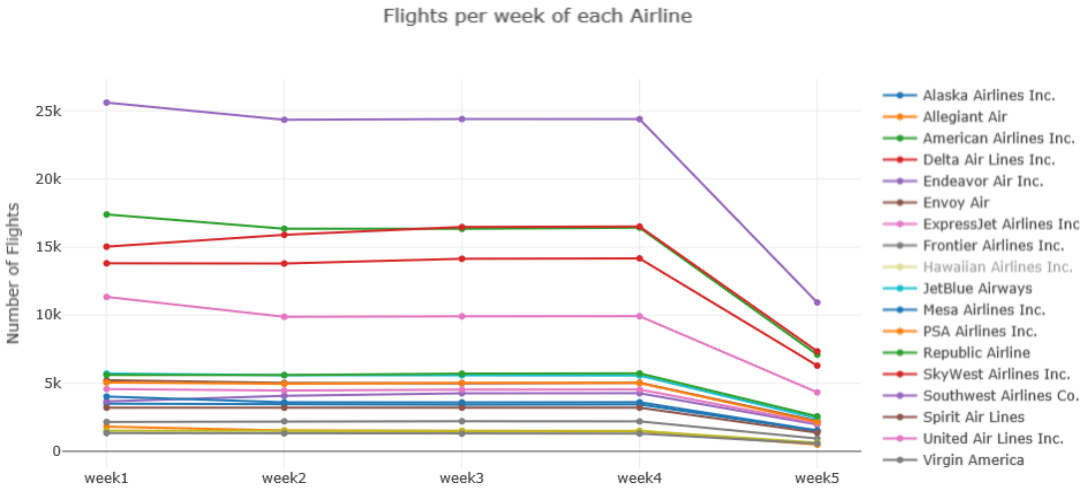
With the purpose of knowing the airlines we are working with, rather than the code that does not say much, it was necessary to add to the original document (flights.csv) the carriers name. This was made by creating a new csv, where carries.csv was merged with the flights.csv: *final_csv = flights.merge (carriers, how = 'left', on = 'OP_UNIQUE_CARRIER')*. Having, now, the airlines names it was easier to evaluate each company's number of flights during the first month of 2018. In this document, were found 18 airlines, being the five biggest the SouthWest Airlines, American Airlines, Delta Airlines, SkyWest Airlines, and United Airlines.

[5] https://ipfs.io/ipfs/QmXoypizjW3WknFiJnKLwHCnL72vedxjQkDDP1mXWo6uco/wiki/Flight_length.html



In order to obtain the graph above, it was necessary to group the number of flights by the Airlines' name. For that purpose, a *groupby* was used, once again.

The number of flights per company was also well distributed throughout the month, as all weeks had almost the same number of trips. The fifth week only had a significantly smaller number of flights as it is composed only by three days.



In order to be able to understand this number of flights variation throughout the weeks, it was required to, firstly, define a function – "num_week" – that would identify the corresponding week of a flight, using as inputs the flights' 'YEAR', 'MONTH', and 'DAY_OF_MONTH'. Afterwards, this function was applied and a new column named 'NUM_OF_WEEK' was created. Subsequently, a new variable – "flights_airlines_per_week" – was created by grouping the number of flights by the

'DESCRIPTION' variable, in order to get the number of flights per airline, and then by the 'NUM_OF_WEEK' with the purpose of having the said number organized by week.

With the purpose of having all 18 different airlines represented in the same graph, it was necessary to create, in the first instance, a new variable "airline" with all the unique airline name, then "for loop" with "i" in range 0 to len (airline) was used in order to obtain the corresponding parts related to each airline. Inside the loop, a query was created to verify if the name "i" of "airline" is equal to the "Description" of "flights_airlines_per_week", by applying this query to "flights_airlines_per_week" information about flights per week of airline "i" will be filtered and filled to the "i" position of "airline".

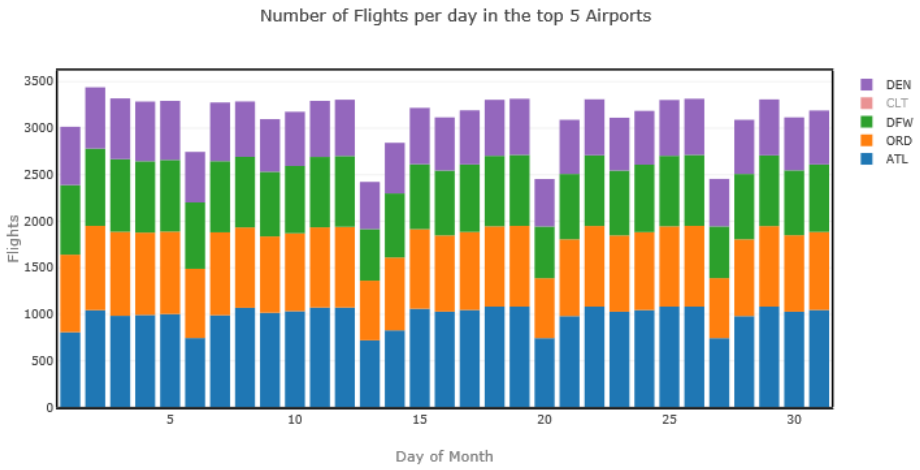
The next step was then to organize this information in a dictionary that would contain the airlines' names as *keys*, as well as, the corresponding number of flights per week as *values*. Instead of inserting 18 variables of trace to create the graph, a new list called "data" was created. This list will fill traces info by using a "for loop" of the different airlines' names. "i" was then defined as each airline info (go.scatter). Inside it, we would have "x=x1", where x1 is a list of 5 weeks so all the traces will have on their x label the five weeks. In the y-axis, we have "y=dict_flight_airline_week[i]", and each y will be the values of the different *keys* on the dictionary of the number of flights per airline per week explained above. Each time that "for loop" runs, in the end, "i" (the info of trace of each airline) will be added to the list "data". Then applying plotly.offine.plot with "data" mentioned above and a layout with title of graph on it, a plotly graph will show.

- Airports Analysis

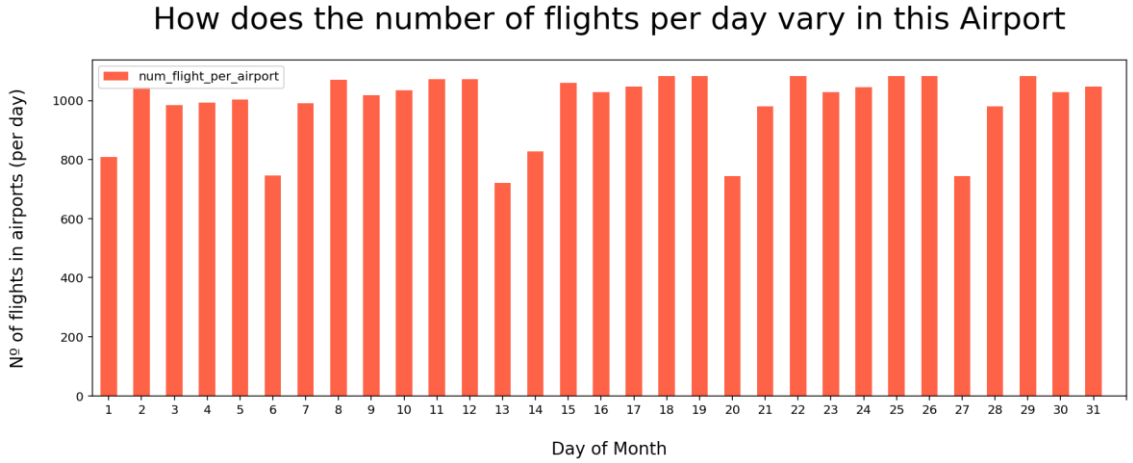
Top Five with Most Flights

Using, once again, the *groupby* function to create a new variable that would group the number of flights by the "ORIGIN" airport, it was possible to identify the top five airports. In order to obtain the intended values per day, it was necessary to apply a lambda to the column 'Nº of flights', that would divide the total number of flights, in the month of January, for 31 days. (Graph in the Appendix)

After having the average number of flights per day, an analysis of how these flights were distributed throughout the month was carried out. For that purpose, it was created a list that would memorize the info of traces. This was made by defining a *query* that would seek solely the flights from airport i. Then, the query was applied to flights1 and a new variable, that only contained airport's flights – "flights_actualized" – was created. The following step was to group the "flights_actualized" by 'DAY_OF_MONTH' in order to get how many flights airport "i" had, per day.

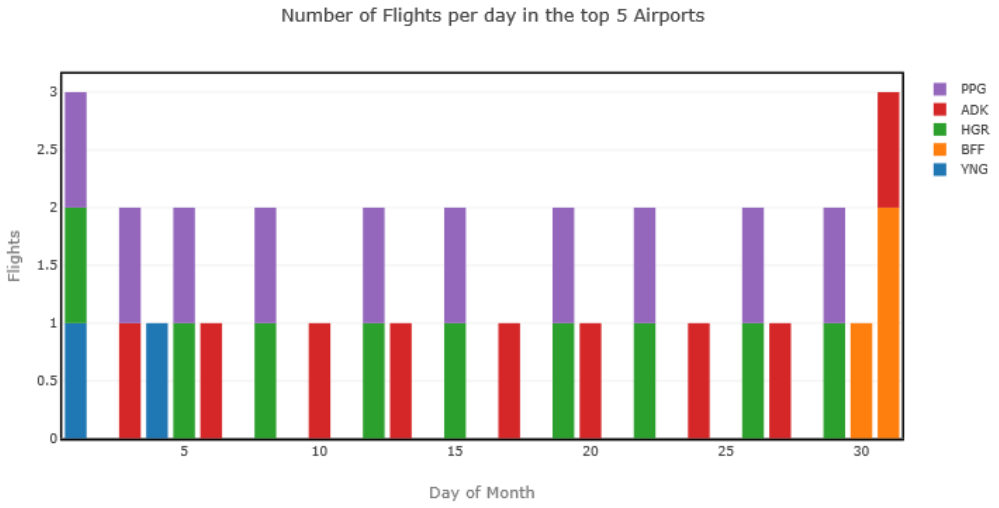


When choosing one of the largest airports, in this case, ATL, in Atlanta, Georgia, it was possible to gauge that its number of flights per day of the week had a similar pattern as the overall number of flights/day, studied in the "Number of flights per day" section. This way, it was possible to verify that on Saturdays the number of flights was substantially lower than in the rest of the week, where the results were more homogeneous.



Top Five with Fewer Flights

A similar process was used when trying to evaluate what were the airports with the smallest amount of flights. However, for this specific case, the number of flights in the first month of 2018 were not divided by 31, as the results would be smaller than 1 and, therefore, would become somewhat irrelevant for the analysis in hands. (Graph in Appendix)



Airport Connections

With the purpose of analysing existing connections from a specific airport, a function had to be created. The first step was to create a variable "connection_airline" to group flights by 'DESCRIPTION', 'ORIGIN', and 'DEST', using *groupby*. Then, it was created an array variable which contains airlines' names and that is filled with the respective sections of 'connection_airline' defined in the first step. For this purpose, a "for loop" by airline's name was used. With this function, it is possible to identify all destination airports that receive flights from all airports from which a specific airline departs. In other words, for example, airline[0] would represent all the flights of all the airports from where Alaska Airlines Inc. departs, to all the possible destination airports. Therefore, 'num_of_connection_per_airline' will be a dictionary where the airlines' names would be represented by *keys*, while the different connections between airports would be characterized by the respective *values*.

Two more variables were created – 'airline_name2' and 'airline_name3'. While both of them have the airlines' names, the first one would be filled with information

regarding the unique ORIGIN airports, while the second would be filled with the unique DESTINATION airports. We also defined a function called "airport_conn" with 3 arguments – *input_airline*, *input_airport*, and *input_answer_origin_dest* – which are going to be defined by the user's responses. Initially a *query* was, once again, defined based on input_airline, in order to select from "connection_airlines" only those with the same airline description. Then basing on the answer on answer_origin_dest, there will be two symmetric function, for example if the answer is "origin", another query will be created basing on both "answer_origin_dest" and "input_airport" to select from the column ORIGIN of "connection_airlines" all the airport equal to input_airport introduced by user. In the end, by applying these two queries to "connection_airline", a dataframe with only flights from input_airline and origin airport input_airport will be shown. The function will print all the unique destination airport of this dataframe shown. The opposite is done when the user chooses destination as answer_origin_dest, where in the end a data frame with only flights of airline input_airline and destination airport input_airport will be shown and then print all the unique origin airport on this dataframe.

After all this process is complete, is time to ask the user what he/she wants to do. The first question would be "From the airlines above, which one do you want to know about?". Here, a "while loop" is used in order to give the user the opportunity to insert an airline name until a correct name is written, or if a no answer – "n" – is given. After having selected an existing airline, the next step is to choose an origin or destination airport. If the user wants to know more about Origin airports, he/she is going to be presented with a display with all the possible Origin airports from where the chosen airline departs. The next step is, then, to choose one of those airports. Then, once again, a "while loop" is going to be used in order to give the user multiple opportunities to insert a valid answer. In the end, after having all three inputs chosen by the user, these are going to be inserted in the initial function "airport_conn". Even if, on the other hand, the user chooses Destination airports, the logic applied is going to be the same.

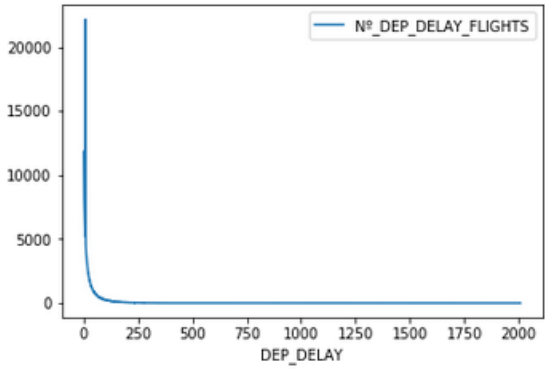
- Flights' Delays Analysis

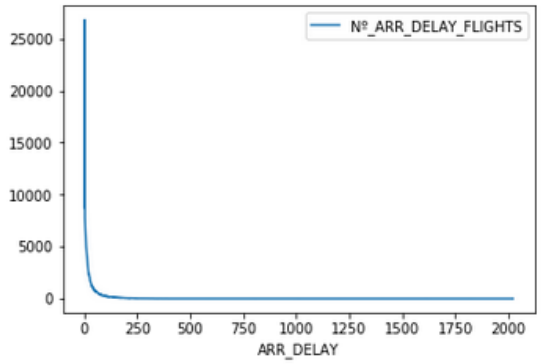
In January 2018, 17169 flights were cancelled and 1249 were made with a change in destination. From the remaining 551700, 201046 departed with at least 15 minutes delay. Therefore, the number of flights departing with some kind of delay represented

35.26% of the total number of flights. Nevertheless, the number of flights arriving ahead of time was almost two times higher than the ones departing late, with a percentage of 63.5%.

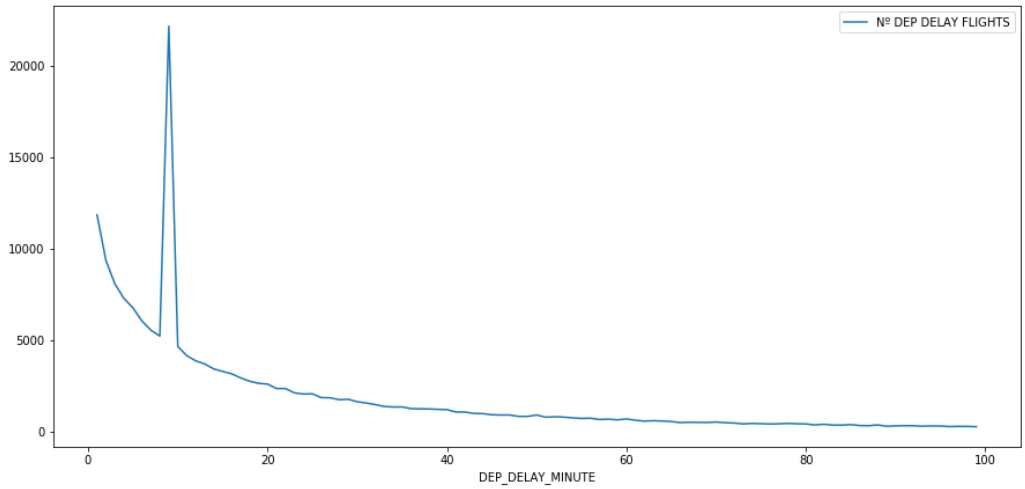
In order to obtain these values, it was necessary to, first, define queries in order to detach 'CANCELED' and 'DIVERTED' flights. The next step was to create a list with the total number of cancelled, diverted and with a correct destination flights. The same logic was applied to each airline by creating a dictionary that would be filled with the respective values of cancelled, diverted, and correct destination flights. This was made to each airline by applying the same query to a grouped by "name of airline" variable.

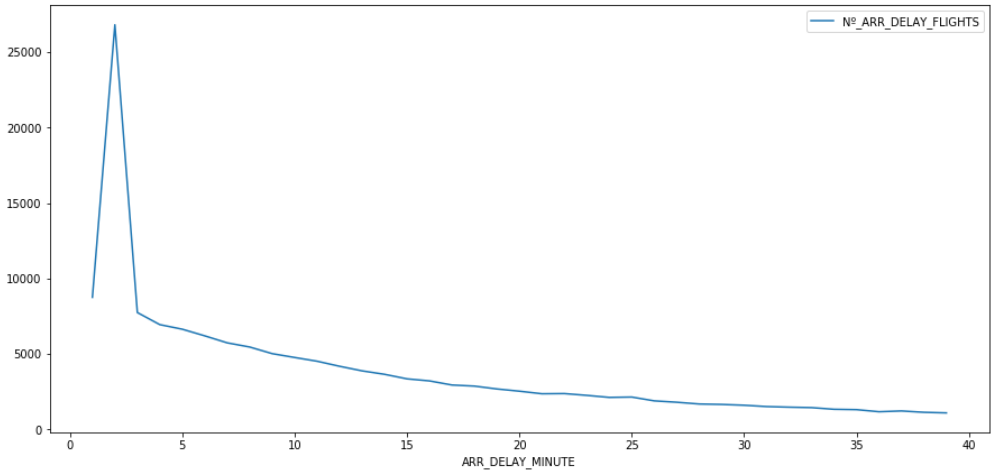
When trying to evaluate the frequency of delayed flights, two graphs were initially made. However, these were not a good representation of the intended results, as there were too many single flights with extremely high delay times.





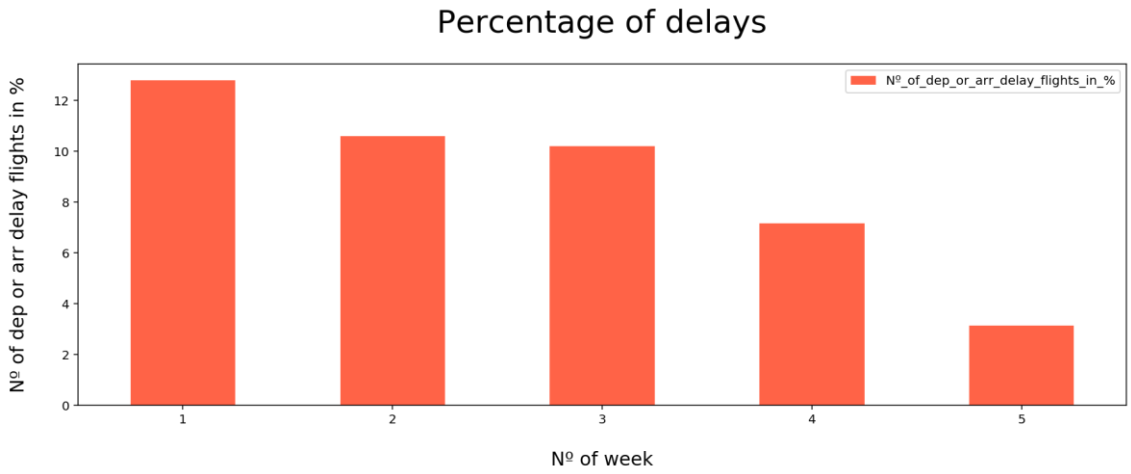
Evaluating the graphs below, that are a better representation of the flights' delays distribution, it is possible to acknowledge that the majority of flights depart with ten minutes delay, while a majority of flights arrive with less than five minutes delay.





Nevertheless, these delay times can be mitigated if the airlines can manage to reduce the air time.

The first week of January was the one with the highest number of both departure and arrival delays. A number that then tended to decrease as the month went by. As it was expected, the fifth week reflected the least amount of delays. One of the reasons that can explain this is the fact that this fifth week is composed solely by 3 days. Nevertheless, there was a big difference in what concerns the number of flights in the first and fifth week, as the first week had 56,278 more flights than the fifth one. This can be in the origin of the delays' percentage as, if there are more flights, then it is expected that the number of delays would be higher as well.



- Correlation between variables

In order to be easier to identify the most important variables for the study on hands, it was decided that the columns 'YEAR', 'MONTH', 'OP_UNIQUE_CARRIER', 'OP_CARRIER_FL_NUM', 'ORIGIN_CITY_NAME', 'DEST_CITY_NAME', 'DEP_TIME_HOURS', and 'CRS_ELAPSED_TIME', were going to be eliminated as they were not seen as relevant.

The next step, in order to have a more in dept study, was to create four new variables. These were 'DEP_DELAY_DUMMY', 'ARR_DELAY_DUMMY', 'ON_LAND_DELAY', and 'DISTANCE_PER_MINUTE_OF_FLIGHT'. The first two were originated through the definition of a dummy function that would return the number 1, in the case of a delay occurring, and a 0, for the opposite case. By transforming these first two variables into dummy variables, a fair number of outliers were eliminated, and consequently the results became more accurate. The third one, resulted from the difference between the 'ACTUAL_ELAPSED_TIME' and 'AIR_TIME', and is a representation of the number of minutes a flight is held on land before departing and after landing. The fourth, and last one, was the result of a division between the 'DISTANCE' variable and the 'AIR_TIME' variable, that had as a purpose to help us understand if there was a significant correlation between the flights speed while on air and the arrival delays (see matrix of correlation in the appendix).

Evaluating the correlation matrix, it is possible to assess that, as it was expected, the arrival delay is strongly correlated with the departure delay. This result goes in line with what was stated in the background section and, therefore, can be easily explained as, if one flight arrives late, then, the next flight that uses that same aircraft, has a high probability of departing late as well.

Assessing the four new variables that were created, there are four correlations that are relevant enough to be explained. The first one is the correlation between the "Dep_Delay_Dummy" and the "Day_of_Month" that had a value of -0,21. This value does not represent a strong correlation. However, it permits us to state that as the month goes by, the probability of a delay occurring becomes lower. This can be explained as it is in the begging of the month of January that two holidays occur. Nevertheless, this relationship was corroborated when evaluating the percentage of delays throughout the

weeks, as the first and second weeks were the ones with the highest percentages. At the first sight, a correlation of -0.21 might not seem that much, however, it is higher than the initial correlation between "Dep_Delay" and "Day_Of_Month" as the first one was still counting with numerous outliers. The same had happened in what concerns the "Arr_Delay_Dummy" variable.

Another variable to have into consideration is the "On_Land_Delay" variable that has shown smaller, yet positive correlations with both "Dep_Delay" and "Arr_Delay". Nevertheless, the second correlation is stronger, with a value of 0.25, compared to the first one that had a value of 0.073. This difference can be explained as the variable "Arr_Delay" has into consideration both the time on land before a flight departs and the time on land after the flight has arrived at its destination. Therefore, if no measures are taken in order to improve on-air performance, by reducing the air time, the latest a flight departs, the latest it is going to arrive, being, therefore, the arrival delay strongly influence by the departure delay.

The last variable being studied is the "Distance_per_minute_of_flight" that has shown a negative correlation with the "Arr_Delay". Such result makes sense when thinking that the greater the distance travelled per minute, the faster the plane is going to travel and, consequently, the probability of a late arrival is lower.

- Regression Analysis

In order to compute a regression model, changes in the database were needed. Firstly, by applying "query_dep_delay" and "query_arr_delay" in the same database, two new variables with only flights that were delayed on departure or on arrive were created. These two new variables will be the Dependent variable of the regression.

Secondly, for a better analyze purpose, the outliers from two dependent variables must be removed. In this case, by begin to define the correspondent percentile of 25% and 75% as "flights3. DEP/ARR_DELAY.percentile (0.25/0.75)", then subtracting percentile 25% to 75% and multiply this value by 1.5 (for moderate outliers), and by subtracting the final value to percentile 25% and adding the same value to percentile 75%, the limit for outliers will eventually be defined: (-42.5, 89.5) for DEP_DELAY and (-48.5, 91,5) for ARR_DELAY. Thus, using the outliers' values, four new queries will be

defined, DEP_DELAY<89.5 and >-48.5, ARR_DELAY<91.5 and >-48.5, and will be applied to the respective database of the dependent variable to eliminate the outliers.

Finally, since there are two types of delay: departure and arrive, two different regressions would be convenient to create, in order to analyze the reason behind these two types of delay.

On the Departure Delay side, four variables were selected as independent variable of the regression: ON_LAND_DELAY, CRS_DEP_TIME, DAY_OF_MONTH, and DISTANCE.

A R^2 of 0.01452, which means 1.45% of departure delay is explained by independent variables, was obtained, along with 4 regression coefficients: 0.1264, 0.0038, -0.0793, and -0.0013, respectively with order of independent variable, and an interception of 13.7882. These result in coefficient can be interpreted as if one unit of increase in the corresponding independent variable would lead to a variation of 0.1264, 0.0038, -0.0793, and -0.0013 minutes respectively on departure delay.

Looking to the Arrive Delay side, ON_LAND_DELAY, DISTANCE_PER_MINUTE_OF_FLY, DAY_OF_MONTH, and "DEP_TIME" were selected as independent variable.

A R^2 of 0.02466, which represents almost as twice as R^2 of departure delay, was obtained. The corresponding regression coefficients are 0.5629, -0.0418, -0.3867, and 0.01498, meaning that: an increase of 1 unit in the respective independent variable will lead to an increase/decrease on arrive delay time. The interception is 2.9153.

Conclusions

From the result of regression analyse, we can conclude that remaining everything else constant, the delay on arriving is much smaller than the delay on departure (interception of arrive delay < interception of departure delay).

The fact of low value of R^2 can be explained by the low correlation between independent variables with delays, in other words, these variables are not the most appropriate to explain or to predict the delay times. Nevertheless, the model of arrive delay shows an R^2 almost double than the departure delay one, thus the credibility of

this model, correctly predict the arrive delay time, would be higher than departure delay model.

Limitations

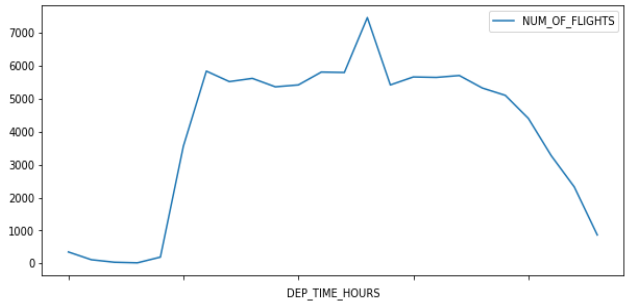
The analysis and conclusions presented in this report might not be the most representative ones as, for this purpose, it was only studied one month and, therefore, the results presented cannot be taken as a representation of the overall US flights' delays.

Appendix

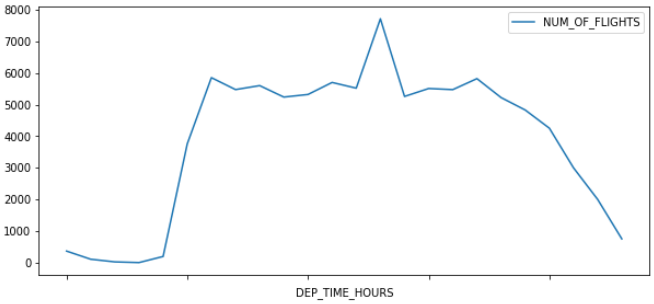
	Delay Cause by Year, as a Percent of Total Delay Minutes										
	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
Air Carrier Delay	28,5%	27,8%	28,0%	30,4%	30,1%	31,9%	29,4%	30,2%	32,2%	32,6%	31,2%
Aircraft Arriving Late	37,7%	36,6%	36,2%	39,4%	40,8%	41,4%	42,1%	41,9%	39,8%	39,2%	39,4%
National Aviation System Delay	27,9%	30,2%	30,6%	25,7%	24,8%	22,5%	24,2%	23,5%	22,9%	23,7%	25,1%
Security Delay	0,2%	0,1%	0,1%	0,2%	0,1%	0,1%	0,1%	0,1%	0,1%	0,1%	0,1%
Extreme Weather	5,7%	5,4%	5,0%	4,4%	4,1%	4,0%	4,1%	4,3%	5,0%	4,4%	4,3%

SOURCE: Bureau of Transportation Statistics

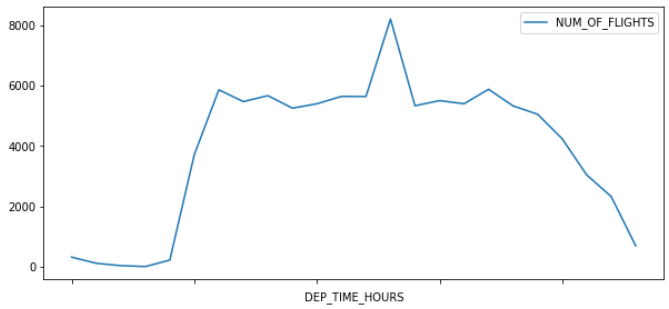
1



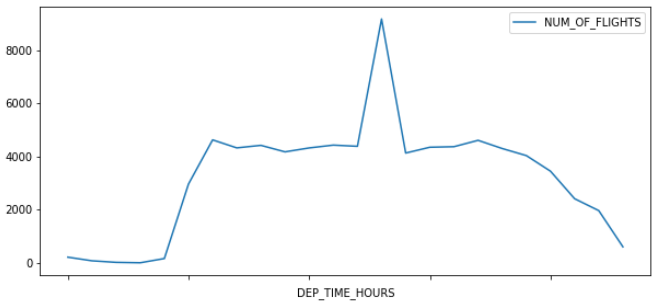
2



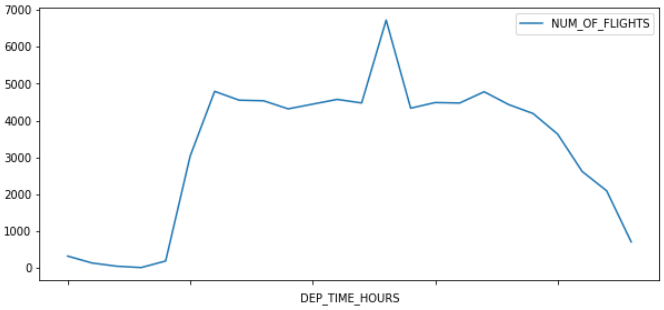
3

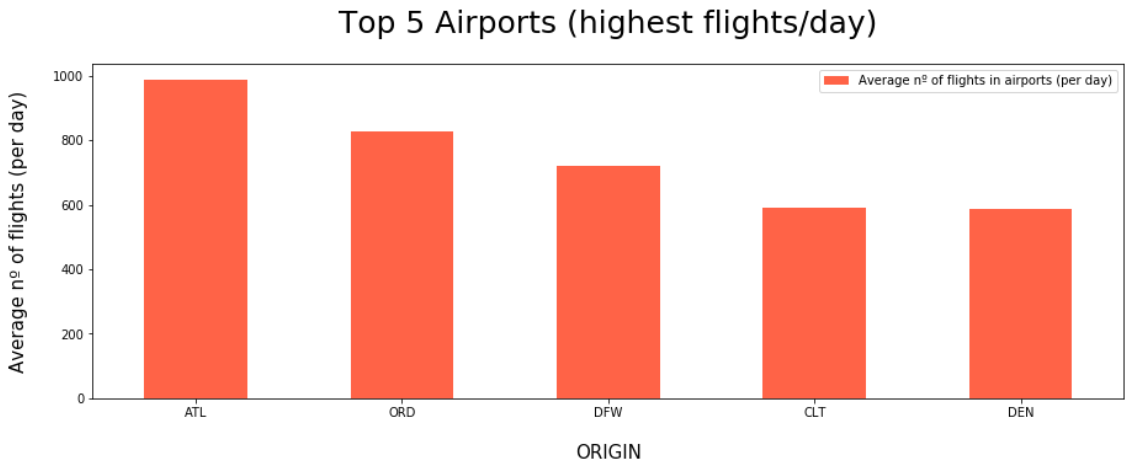


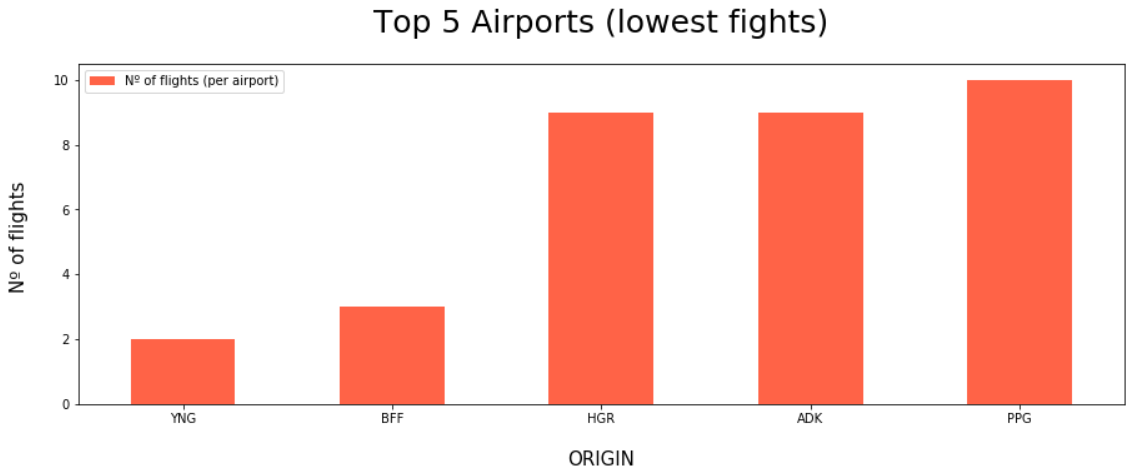
4



5




Top 5 Airports (highest flights/day)
Average nº of flights in airports (per day)
Average nº of flights (per day)
1000
800
600
400
200
0
ATL
ORD
DFW
CLT
DEN
ORIGIN


Top 5 Airports (lowest fights)
Nº of flights (per airport)
Nº of flights
10
8
6
4
2
0
YNG
BFF
HGR
ADK
PPG
ORIGIN

Correlation Matrix: