Programming For Data Science Coursework Project

Analysis of Flight Delays in 2006 and 2007:

Table of Contents:

Student Number : 210500000

Introduction:

This draft report aims to investigate a subset of the 2009 ASA Statistical Computing and Graphics Data Expo, which consists of flight arrival and departure details for all commercial flights on major carriers within the USA, from October 1987 to April 2008. Specifically, this report will answer the five questions by performing analysis on flight data from 2006 and 2007. Each dataset contains approximately 7 million rows of information which was combined to form one large dataset for analysis. In addition to the yearly flight data, three supplementary datasets were provided which were used to aid analysis.

The report aims to address 5 main questions and present the analysis in a clear and concise manner along with suitable visualisations. The five questions it aims to address are:

- When is the best time to fly in order to minimise delays?
- Do older planes suffer more delays?
- How do the number of people flying between different locations change over time?
- Can you detect cascading failures, as delays in one airport create delays in others?
- Use the available variables to construct a model to predict delays.

The analysis for each of the questions was conducted in both R script and python, with the scripts accompanying this report. The visualisations in this report are taken from the python script.

Data pre processing/ Cleaning:

Before starting the analysis, it is important that all relevant data is read, checked for errors and cleaned so that meaningful insights can be extracted from it. To begin with, the 2 main datasets were imported (‘2006.csv’ and 2007.csv) along with the three supplementary datasets (‘ariports.csv’, ‘carriers.csv’ and ‘plane-data.csv’) and were read into dataframes. The dimension or shape of each of the flight datasets were analysed. Each of the dataframes had nearly 7 millions rows and 29 columns. The two dataframes were merged into a single dataframe called ‘flight_data’. Analysis of the variables was required as not all the variables are relevant in our analysis. For example, cancelled and diverted flights were found to have missing delay entries. Therefore they were dropped as it could cause issues while trying to accurately analyse delay. Similarly other columns that were not considered relevant were dropped such as cancellation code, taxi in and taxi out. The resultant dataframe after dropping these rows was checked to ensure that no null values were present. After cleaning the data, new features were added. For instance, most questions require analysis of delay. Therefore new columns were created to help analyse delay. First a new column called ‘TotalDelay’ was created which was calculated by finding the sum of arrival and departure delay. For the sake of clarity, only arrival and departure delays were used to calculate TotalDelay. While arrival and departure delays can often overlap, it is simply a measure to analyse delays in most questions. Secondly, ‘isDelay’ was created which determines if a flight was delayed or not by examining if the value of TotalDelay was more than zero.

The data cleaning and preprocessing was performed in detail in the script for the first question. Since each question’s code was split into separate notebooks, the data cleaning and preprocessing had to be run for each of the questions.

Question 1:

Before starting the analysis, the datasets were read, cleaned and new columns of ‘TotalDelay’ and ‘isDelay’ were created

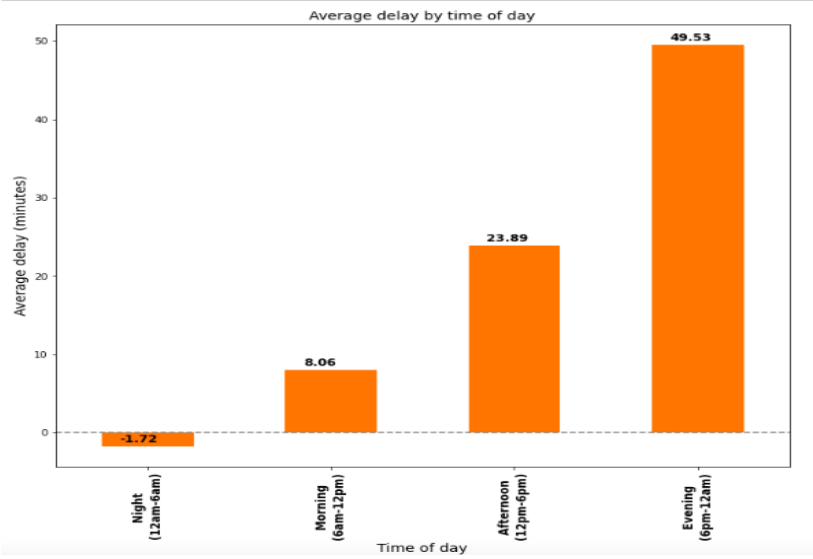
In order to determine what is the best month of the year, day of the week and time of the day to fly, the delay times during each day, week and month were analysed. In order to do this, new columns were created in the ‘flight_data’ dataframe to group the data by day of week and month (whose values were already present for each flight.) For the time of day however, a new column had to be created to store the hour of departure (in 24hr format). This new column was grouped into four categories for time of the day:

1. Morning : 6am - 12pm
2. Afternoon : 12pm - 6pm
3. Evening : 6pm - 12am
4. Night 12am - 6am

Finally, the data in the ‘flight_data’ dataframe was grouped by each time period. From this, the total number of flights in each time period was calculated and the total delay (in minutes) was calculated.

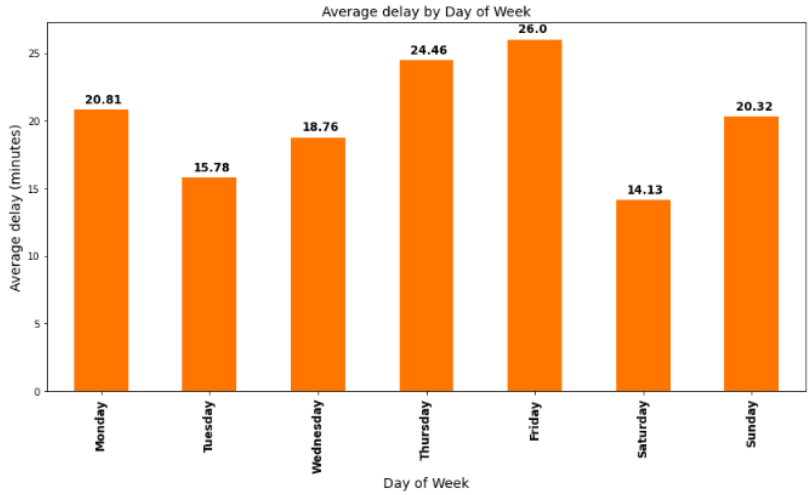
In order to show which period of time has the least delay times, total delay can not be used as its value is influenced by the number of flights in that time period. Therefore in order to have a more accurate comparison of delay times, the average delay times of each time period was calculated and displayed. The results were as follows:

When is the best time of day to fly to minimise delays?



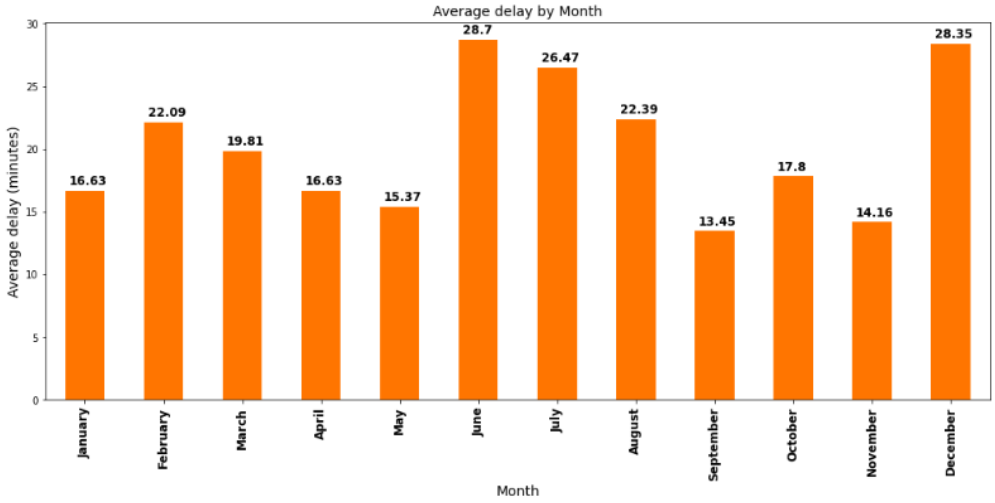
From the graph we find that the best time of day to fly is at night (12 am – 6 am) with an average delay time of only-1.7 minutes. Which would mean that on average flights at this time will leave earlier than their scheduled departure times. On the other hand flights during the evening experience an average delay of nearly 50 minutes.

When is the best day of the week to fly to minimise delays?



The graph shows that the best day to fly in order to minimise delays is Saturday with an average delay time of 14 minutes followed closely by Tuesday at 15.8 minutes. Friday and Thursday appear to be the days with the highest delay times with 26 and 24.5 minutes respectively.

When is the month of the year to fly to minimise delays?



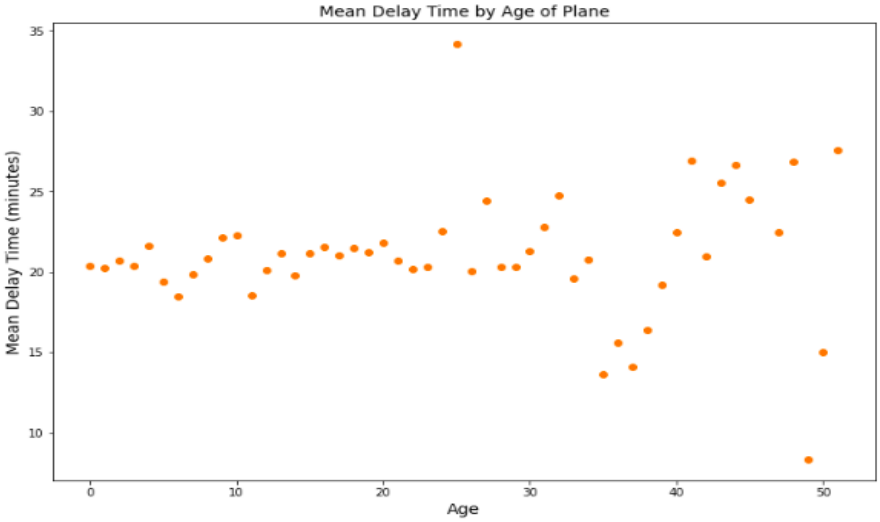
It appears that the best time to fly is September with an average delay time of 13.5 minutes, whereas June and December see the highest average delay times of around 28 minutes each.This could possibly be explained by the fact that June and December are during the peak of the Summer and Winter holiday seasons.

Question 2:

As was previously done, the datasets were read, cleaned and new columns of ‘TotalDelay’ and ‘isDelay‘ were created.

In order to analyse the correlation between age of an aircraft and its delay time additional data was required from the ‘plane-data’ supplementary dataset which was imported as ‘plane_data’. ‘plane_data’ was cleaned and any null values were dropped. The 'Year' column in flight_data was renamed to 'FlightYear' and the ‘tailnum’ column in plane_data was renamed to ‘TailNum’ so it corresponds with the tail number column in ‘flight_data’. ‘plane_data’ was then merged with ‘flight_data’ based on the common column ‘TailNum’.The merged data was stored in a dataframe from which only relevant columns - 'FlightYear' 'year', 'ArrDelay', 'DepDelay'and 'TailNum' were kept. Any null values in the merged data were dropped. However additional cleaning was required as on inspecting the data type of year, it appears to be an object which indicates that there are still certain non numeric data types that need to be removed. Values such as ‘None’ and ‘0’ were converted to null values and then were successfully removed. A new column was created called ‘PlaneAge’ which subtracts the year of the flight (‘FlightYear’) from the manufacturing year of the plane (‘year’).

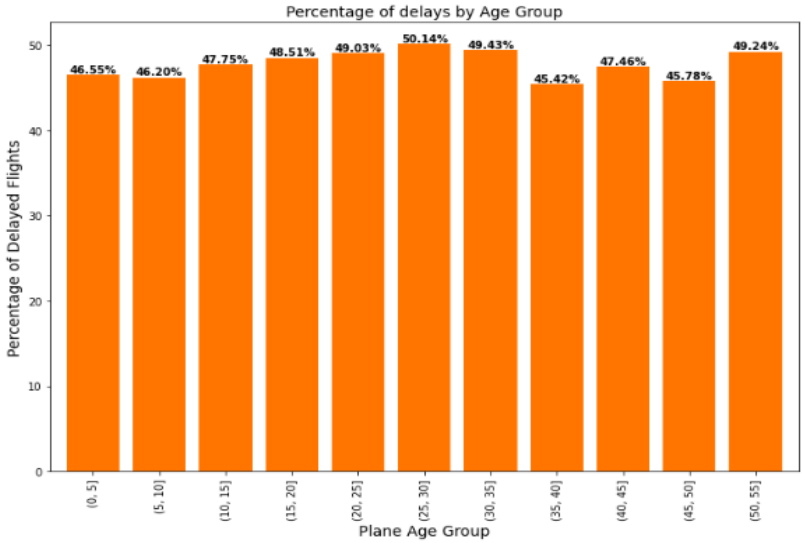
Now the data was ready to be visualised. To try and show if older planes suffer more delays, a scatter plot was created of every flight’s total delay vs its age. However due to the number of entries of data, this graph is difficult to interpret but it does appear that newer planes (up to 10 years old) are likely to depart earlier than their scheduled departure times. Therefore a new scatter plot was created where the flights were grouped by year and the mean delay times for each year was calculated.



This graph appears to be easier to interpret. While it does show that older planes have greater variation in delay times but it does not conclusively show that older planes suffer more delays.

To confirm this we find the correlation between plane age and total delay:
The correlation coefficient was calculated and was found to be 0.0031 – showing a very weak, almost non - existent positive correlation. Therefore suggesting that older planes do not suffer more delays. (At least in terms of average delay times.) However, it may be possible that older planes may have a higher

percentage of their flights delayed. To analyse this, the data was grouped by plane age in increments of 5 and the percentage of total delayed flights was calculated and plotted for each of the age ranges.



The bar graph does not show too much variation in percentage of delayed flights. Despite the oldest planes having one of the highest delay percentage, the age groups of 35-40, 40-45 and 45-50 have the lowest delayed flights percentage .Therefore we can not say that older planes suffer a higher percentage of delayed flights either.

Question 3:

To begin with the data was read, cleaned and TotalDelay and isDelay was calculated.

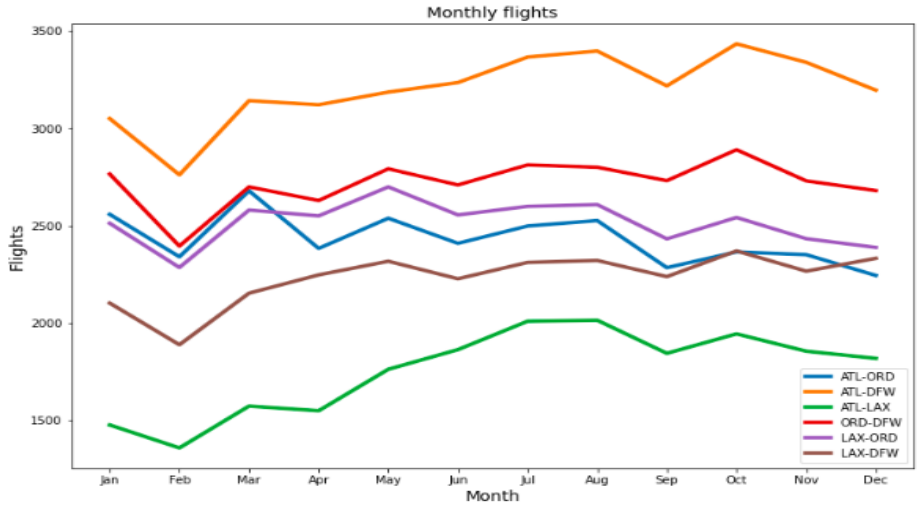
From the available datasets, there is no information on the number of passengers flying, hence why the number of flights was used as an approximation. This section will aim to highlight the trends in the number of flights across :

a) The different months
b) The two years (2006 and 2007)

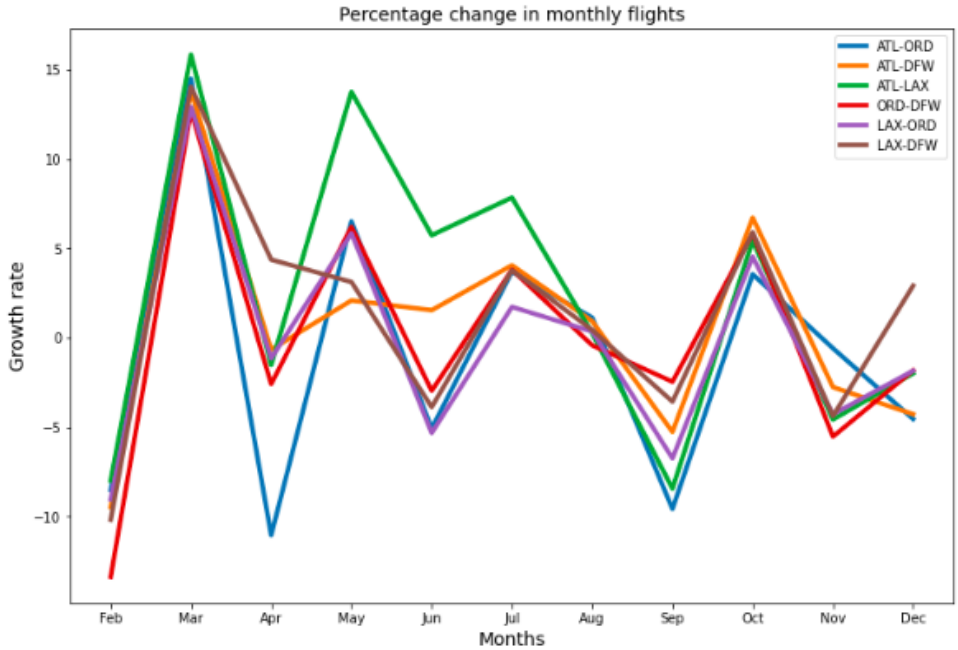
The different locations that were analysed were chosen by finding the top 4 most popular destinations by the number of flights. The top 4 destinations were : 'ATL', 'ORD', 'DFW', 'LAX' . From these airports six unique pairs of origins and destinations could be created. Therefore, by analysing the trends in the number of flights (an approximation for number of passengers flying) between these 6 pairs of airports over different months and years we can understand how the number of people flying between locations changes over time.

First a dictionary was created to store the flight data of each pair while ensuring that both combinations of the pair were stored together (For example the pair LAX - ATL and LAX - ATL were treated as the same pair.) Once the dictionary of the pair's flight data was created, only relevant columns of 'Year', 'Origin', 'Dest' and 'Month' were kept. Each of the pairs's flight data was then grouped by month and year for further analysis. Trends in monthly flights were analysed first. The graphs of each of the pairs was plotted individually first. The graphs show trends with sharp declines in monthly flights in February and huge spikes in the holiday season. These trends are seen across almost all the pairs. However, plotting the

graphs individually makes it difficult to accurately interpret the trends. Therefore the monthly flights of all six pairs were plotted on a single line graph :



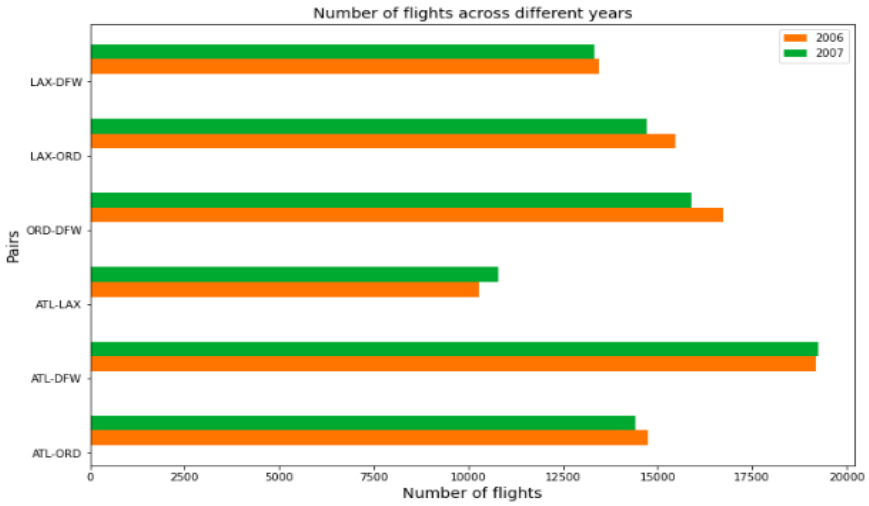
We see a sharp decrease in monthly flights in March for all 6 pairs. The pairs also show increases in monthly flights during the summer months - between June and August. However, in September there is a decline in the number of flights. Following this, there is again an increase in the holiday season between October and December. However since each of the pairs have a different number of total flights it becomes difficult to accurately determine trends in the monthly flights. To solve this we can look at the percentage change in flights between months. This will show us the change (in %) in the number of flights between the months as it takes into account the relative changes in the number of flights, rather than being influenced by the total number of flights.



This supports the conclusions made from the previous graphs but also highlights a few divergences from the trends for some of the pairs.

For instance, flights between ATL and LAX show a much higher increase in flights than other pairs in the early summer months fromMay to July. This could be due to the fact that Los Angeles is a popular holiday destination. Flights between ATL and ORD show a much more significant drop in flights in April. Lastly, flights between ATL and DFW and LAX and DFW do not show the characteristic drop in April and May that the other pairs show.

The total number of flights in each year was plotted with a stacked bar graph:



Using a stacked bar chart, we can see that most pairs see a decline in the number of flights from 2006 to 2007. However, the pair, ATL - LAX is an exception with its flights significantly increasing from 2006 to 2007.
It appears that the pair of locations - ATL - LAX is an anomaly and does not follow the same trends that other pairs follow.

Question 4:
To begin with the 2 datasets were imported, cleaned and merged to form flight_data.

For this question, the flights of individual planes between different locations were analysed. A criteria for ‘Cascading Failure’ was made and if that particular flight met that criteria it was classified as a cascading delay.

To begin with, the ten aircrafts with the most number of flights were found by sorting by ‘TailNum’. In order to observe the flights in chronological order. The flights needed to be sorted by the date and time of departure. Hence a new column called ‘DateTime’ was created to store the date and the exact time of departure of each flight. Now taking a particular aircraft: ‘N485HA’ had a total of 8055 flights. When sorting its flights by the newly created ‘DateTime’ column we see this:

	TailNum	ArrDelay	DepDelay	LateAircraftDelay	isDelay	DateTime	DepTime	Origin	Dest	next_arr	next_lateaircraft	CascadingDelay
347853	N485HA	-2.0	-4.0	0	0	2005-01-01 05:16:00	05:16:00	HNL	ITO	1.0	0.0	0
347827	N485HA	1.0	-8.0	0	0	2005-01-01 06:32:00	06:32:00	ITO	HNL	0.0	0.0	0
350492	N485HA	0.0	-9.0	0	0	2005-01-01 07:56:00	07:56:00	HNL	LIH	-2.0	0.0	0
350523	N485HA	-2.0	-7.0	0	0	2005-01-01 09:03:00	09:03:00	LIH	HNL	2.0	0.0	0
348959	N485HA	2.0	-4.0	0	0	2005-01-01 10:06:00	10:06:00	HNL	KOA	1.0	0.0	0
348928	N485HA	1.0	-3.0	0	0	2005-01-01 11:22:00	11:22:00	KOA	HNL	3.0	0.0	0
349083	N485HA	3.0	3.0	0	1	2005-01-01 12:38:00	12:38:00	HNL	KOA	16.0	3.0	1
349114	N485HA	16.0	5.0	3	1	2005-01-01 13:53:00	13:53:00	KOA	OGG	35.0	13.0	1
349145	N485HA	35.0	22.0	13	1	2005-01-01 15:07:00	15:07:00	OGG	HNL	29.0	27.0	1
350461	N485HA	29.0	27.0	27	1	2005-01-01 16:22:00	16:22:00	HNL	KOA	27.0	21.0	1
350430	N485HA	27.0	21.0	21	1	2005-01-01 17:31:00	17:31:00	KOA	HNL	17.0	17.0	1
350213	N485HA	17.0	18.0	17	1	2005-01-01 18:38:00	18:38:00	HNL	ITO	10.0	0.0	0
350244	N485HA	10.0	8.0	0	1	2005-01-01 19:48:00	19:48:00	ITO	HNL	8.0	0.0	0
350276	N485HA	8.0	9.0	0	1	2005-01-02 14:19:00	14:19:00	HNL	LIH	12.0	0.0	0
349053	N485HA	12.0	6.0	0	1	2005-01-02 15:21:00	15:21:00	LIH	HNL	2.0	0.0	0
350855	N485HA	2.0	-1.0	0	1	2005-01-02 16:34:00	16:34:00	HNL	OGG	-1.0	0.0	0
350827	N485HA	-1.0	-3.0	0	0	2005-01-02 17:37:00	17:37:00	OGG	HNL	-3.0	0.0	0

The aircraft is continuously flying between locations with the destination of its current flight becoming the origin of its subsequent flight. Therefore we can analyse cascading failures by seeing if delays in one flight causes subsequent delays in the immediate next flight.

Cascading delays were classified on the following criteria :

1. There was a delay in departure from the first airport
2. There was a delay in arrival in the following airport
3. Along with the arrival delay, a late aircraft delay was detected in the second airport

If all three conditions were met, that particular flight would be deemed to have caused a cascading delay Two new columns were created - next_arr - to store the value of the next flight's arrival delay and next_lateaircraft - to store the value of the next flight's late aircraft delay. If departure delay, next_arr and next_lateaircraft were all found to be more than 0, then that flight would be considered to have caused a cascading failure. The calculation for cascading failure was performed in a for loop which iterates through a list of tail numbers (in this case 10). Besides cascading failure, the flight's total number of delayed flights, total number of cascading flights, percentage of delayed flights, percentage of cascading failures and percentage of cascading failures of total failures was calculated inside the for loop and the results were stored in a dataframe:

The table shows a summary of delayed and cascading failure statistics for each of the top 10 most popular aircrafts. The mean cascading failures from delayed flights is : 9.5%
Which means on average 1 out of every 10 flights are cascading failures caused by delays in previous airports.

	TailNum	Flights	Delayed	% Delayed	% Cascade	% cascade of Delayed flights
0	N308SW	8560	3864	45.140187	6.530374	14.466874
1	N478HA	8195	1196	14.594265	1.366687	9.364548
2	N479HA	8079	1147	14.197302	1.027355	7.236269
3	N480HA	8078	1220	15.102748	1.609309	10.655738
4	N485HA	8055	1298	16.114215	1.477343	9.167951
5	N484HA	7953	1158	14.560543	1.257387	8.635579
6	N481HA	7947	1237	15.565622	1.321253	8.488278
7	N487HA	7861	1142	14.527414	1.424755	9.807356
8	N475HA	7844	1130	14.405915	1.236614	8.584071
9	N477HA	7842	1132	14.435093	1.275185	8.833922

Question 5:

For this question, delays were predicted using regression and classification models. For both classification and regression models, multiple models were created to find the optimal model to predict delays. For the classification models, the target variable is DelayClass, which is a new column created which classifies the total delay times into moderate/no delay (for delay times less than 15 minutes) and significant delay(for delay times over 15 minutes). For the regression models, the target variable is TotalDelay.
Only relevant features had to be selected to be used in the models for prediction of delay. Out of the 29 variables, features such as FlightNum, TailNum and UniqueCarrier contained too many unique values to be considered in the model. As explained earlier, cancelled and diverted flights had no delay entries. Therefore they were not considered along with CancellationCode. If the model's aim is to predict delays in future flights there are some values in the dataset that will not be known before the flight. For example: DepTime, ArrTime, ActualElapsedTime, AirTime, ArrDelay and DepDelay can only be determined once the flight has either taken off or reached its destination. In addition to this some variables that have time series data such as departure hour were not considered as they are difficult to correctly fit into the models.

Before the models could be created, dummy variables needed to be assigned to the Destination and Origin variables.

A sample of the data was taken -150,000 rows (approximately 10% of the total dataset.) The dependent variables were then defined and the independent or target variable was also defined. A 70 - 30 train -test split was made of the sample. The data was now prepared and ready. Three classification models were used to predict DelayClass - Decision Tree Classifier (CART), Gradient Boosting Classifier and Random Forest Classifier. For performance of each of the models was assessed by calculating the accuracy (the measure of the number of correct predictions), precision (the proportion of true positives out of the actual values predicted), recall(a measure of the true positive out of actual positives) and F1 Score (a mean of precision and recall). The data was trained on the train set and the predictions were made on the test set. The results were as follows:

	accuracy	precision	recall	F1 score
Random Forest	0.929044	0.931288	0.929044	0.926842
XGBoost	0.931889	0.935703	0.931889	0.929441
Decision Tree	0.881267	0.881212	0.881267	0.881239

It appears that all models are good predictors of Delay Class with high accuracy, precision and recall. Random Forest and gradient boosting appear to be better suited to this data than Decision tree classifier.

The three regression models used to predict the total delay were - Random Forest regressor, Decision Tree Regressor and Lasso Regression. Like the classification models, the train set was used to fit the models and the test set was used to make predictions. The metrics used to evaluate model performance were r-square (the proportion of variance in the dependent variable that is explained by the independent variables) and root mean square error (the measure of the differences between the predicted and actual values). The results for the three models were as follows:

	RMSE	R2
Lasso Regression	16.902318	0.945431
Random Forest Regression	18.122606	0.937267
Gradient Boosting Regression	17.026029	0.944629

All three models perform very well with very low mean square error and high r-square value.

(The results for all the models were taken from python code)