ST2195 – Programming for Data Science
(Coursework Report)

UoL Student Number: **210495821**
Page Count: **10 (Excluding Table of Contents and Cover page)**

Table of Contents

Introduction

The datasets taken for this question have been taken from the **Harvard Dataverse** and consists of the flight arrival and departure details of all major carrier flights within the USA ranging from the years 1987 to 2008.

These datasets are very large thus many wrangling and cleaning operations had to be carried out before sense could've been made out of the data in order to answer the questions given.

For our analysis the datasets from the years '2006' and '2007' were chosen since those are more of the recent years hence more up to date results can be found from them.

The following questions that are to be answered are;

1. When is the Best Time of the Day, Day of Week and Time of Year to minimize delays?

2. Do Older Planes suffer more delays?

3. How does the number of people flying between different locations change over time?

4. Can you detect cascading delays as delays in one airport create delays in others?

5. Use the available variables to construct a model that predicts delays.

The answers have been supported with tables and visualizations in Python and R respectively.

Data Cleaning Operations

The datasets from the years 2006 and 2007 were concatenated to given a dataset of 14595137 rows × 29 columns.

It was shown that approximately only 2% of flights were cancelled thus the entire column was removed along with some other irrelevant columns which wouldn't be needed to answer any of the questions.

Arrival Time and Departure time seemed to have some erroneous time values (time values greater than 2400) thus they were removed.

Afterwards all the rows which had NaN(empty) values were removed as they are of no use for the analysis.

Finally, 35 duplicated rows were identified and were removed to provide more accurate results.

The cleaned dataset was then imported to a new csv file in order to answer the questions.

Questions

1.When is the Best Time of the Day, Day of Week and Time of Year to minimize delays?

For this question, delays have been taken as the **'Arrival Delay'** with the assumption that even if a plane departs late it can arrive early thus departure and arrival delays can be contradicting, therefore it is safer to take the arrival delay. The arrival delay was then filtered to only consider the delays greater than 0. (which indicates a delay)

Best Time of the Day

To find this the departure time was grouped into 4 different times of the day and the Average Arrival delay was calculated for each timeframe respectively.

Timeframe	Average Arrival Delay
6am-12pm	20.18 mins
12pm-6pm	30.18 mins
6pm-12am	44.94 mins
12am-6pm	51.26 mins

Therefore, it is evident from the table above that **6am-12pm** is the best time of the day to minimize delays. It can be visualized as follows,

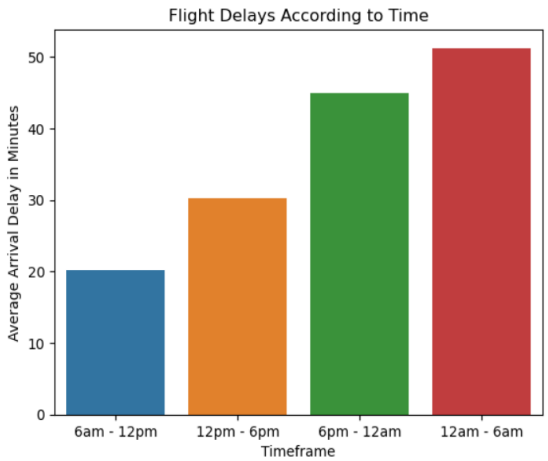


Figure 1 – Python

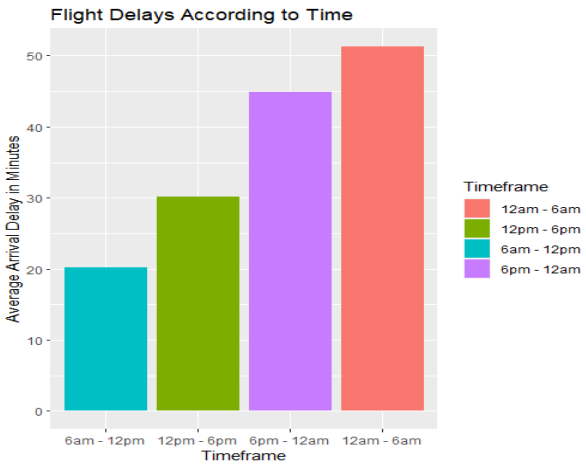


Figure 1 - R

Best Day of the Week

To find this the **'DayOfWeek'** column was used and was grouped by the Average arrival delay.

Day	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Average Arrival delay	31.5mins	29.5mins	31mins	33.4mins	33.1mins	28.1mins	31.4mins

From this table it can be observed that **Saturday** is the day with the lowest average arrival delay. A line plot can be shown to visualize this.

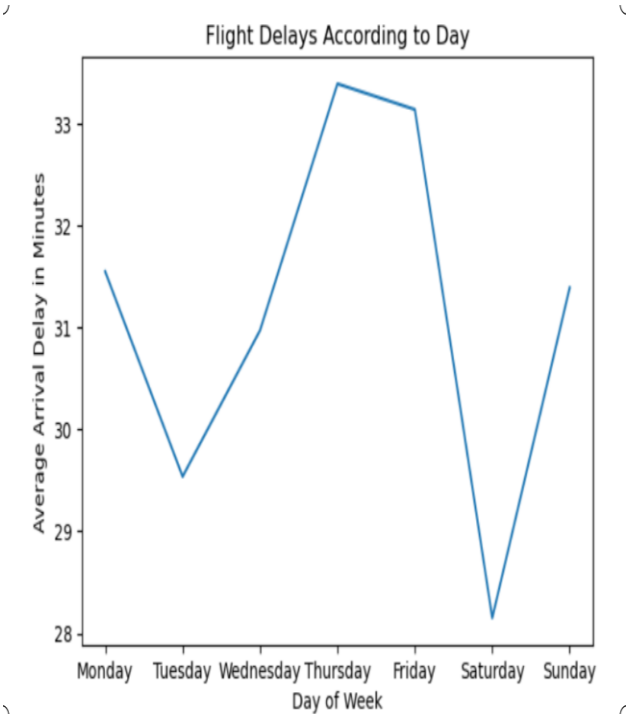


Figure 2 – Python

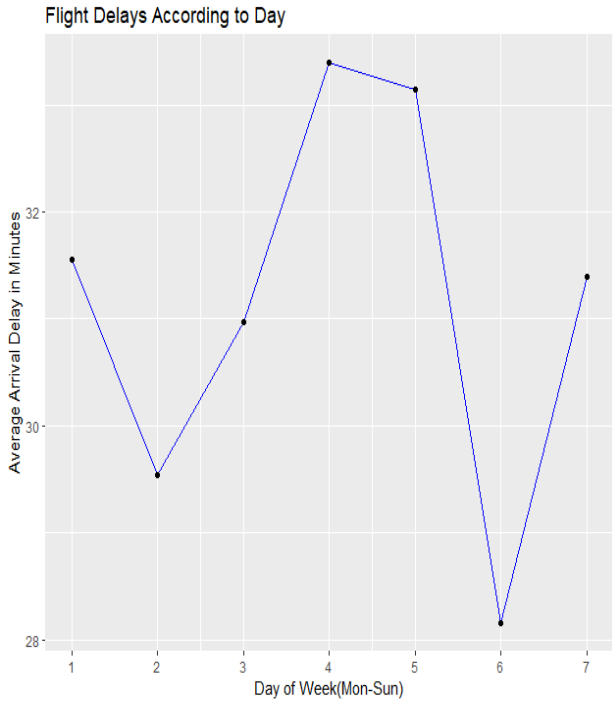


Figure 2 – R

Best Time of the Year

This question was analyzed in two ways: 1. Season Wise

2. Month Wise

Season wise analysis would be,

Season	Average Arrival Delay
Autumn	28.9 mins
Spring	29.3 mins
Winter	32.2 mins
Summer	34.5 mins

Thus **Autumn** is the best season to minimize delays.

Month – Wise analysis can be shown as follows,

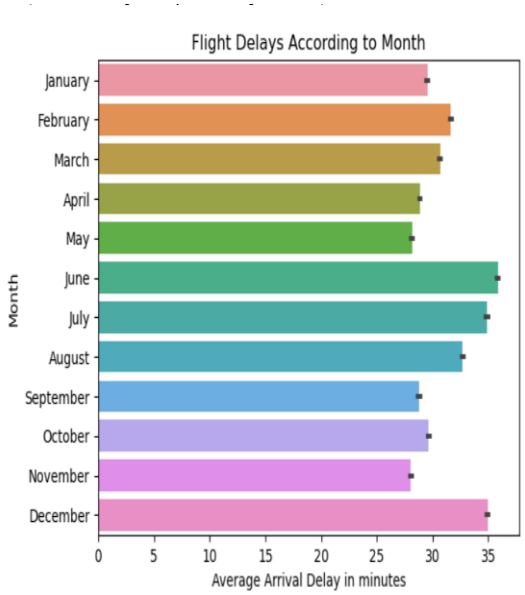


Figure 3 – Python

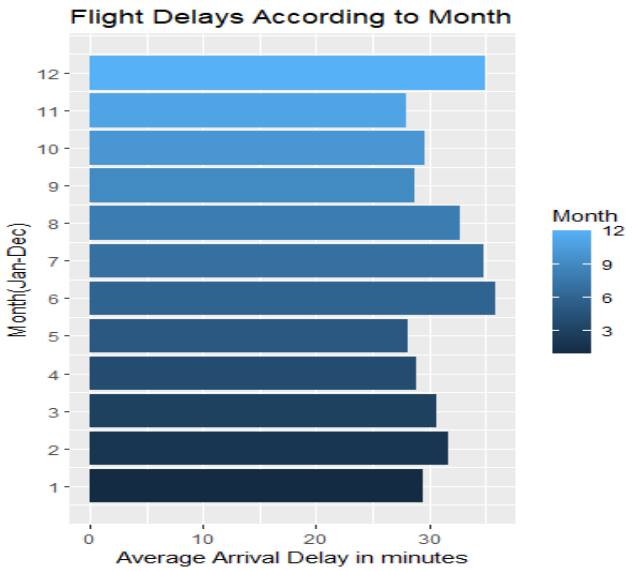


Figure 3 – R

From this we see **November** is the best month to minimize delays.

2.Do older planes suffer more delays?

In order to answer this question, we needed to import one of the supplementary information datasets namely, 'plane-data.csv' along with the cleaned dataset

After cleaning operations and relevant columns were selected, The two datasets were merged on 'TailNum' which represents the Tail Number of the plane.

Since Age of the plane was not provided in the dataset it was calculated by subtracting the manufactured year from the current year and a new column was created.

The Age of the plane was then grouped with the Average Arrival Delay similar to what was done in the first question.

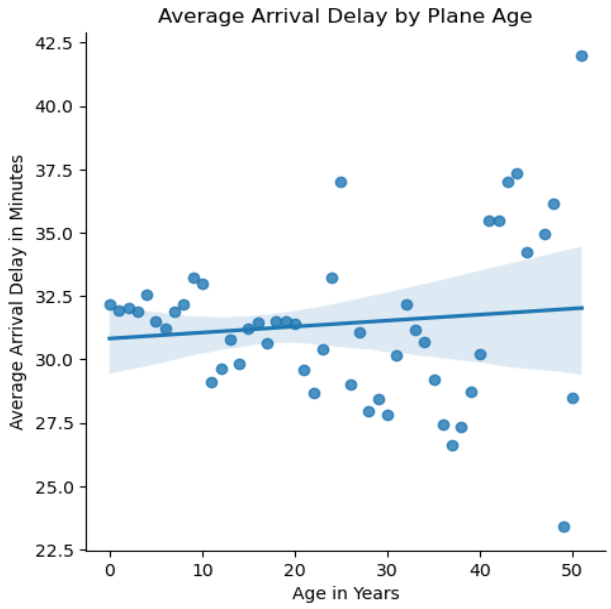


Figure 4 – Python

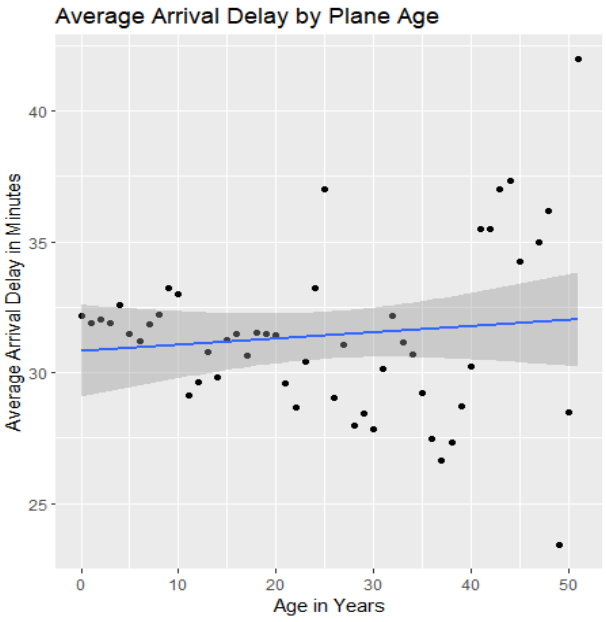


Figure 4 - R

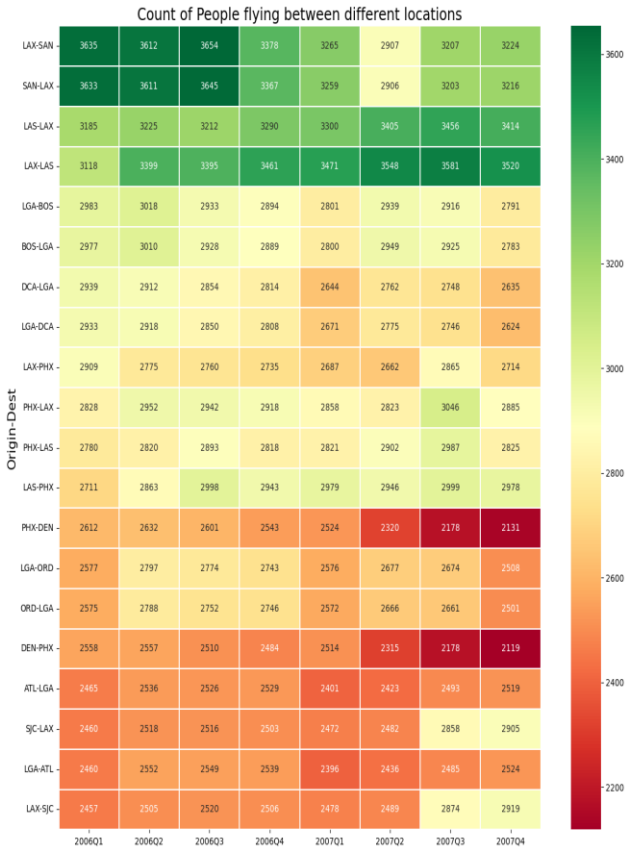
At first glance, we cannot directly jump to a conclusion since the points seem to be clustered on both sides of the line of best fit. However when calculating the correlation value between Age and Arrival Delay we get a value of 0.11 which indicates a very weak positive relationship between Age and Arrival delays. On the contrary, planes that were less than 30 years were considered new and ones greater than 30 were considered old and the mean arrival delay was computed.

New Planes	**31.5 mins**
Old Planes	**29.4 mins**

Thus from this we can deduce that Age does not have a significant impact on Arrival delays and older planes do not necessarily suffer more delays.

3. How does the number of people flying between different locations change over time?

To answer this question, the supplementary dataset airports.csv was used . In the variable descriptions it is mentioned that 'iata' refers to airport code which is used for origin and destination of the cleaned dataset. Thus it was renamed to origin and were merged together. A new column was then formed putting origin and destination in one column(org-dest). Since the number of people had not been provided in the dataset the frequency of the Origin to Destination columns was used as an alternative to carry out the analysis. To examine the change over time, the 4 quarters of the year were used. A new dataset of the frequencies of the top 20 locations travelled in each quarter was then created and a heat map was plotted.



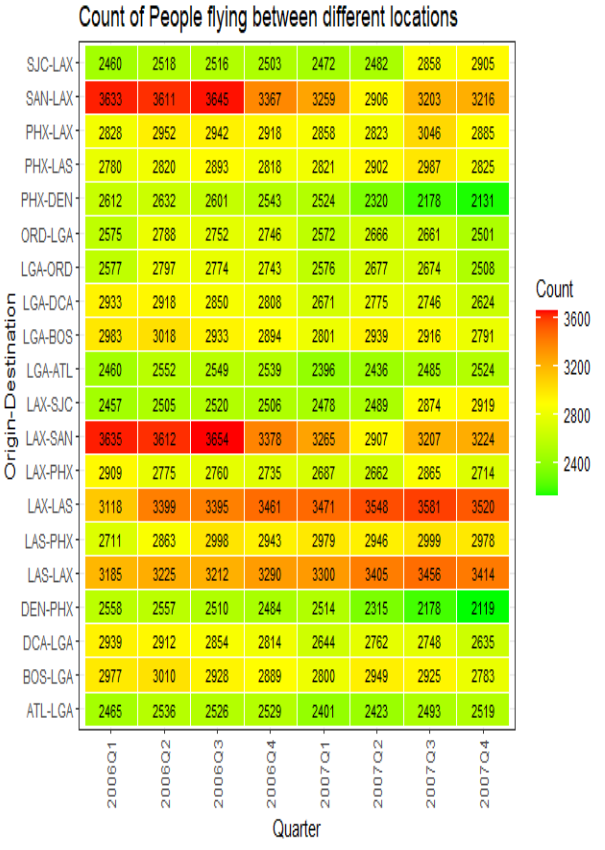


Figure 5 – Python **Figure 5 - R**

We see that the number of people flying between LAX-SAN and back seem to have dropped significantly after 2007 whereas places like LAS-LAX and back seem to have a steady increase in passengers travelled as the time moved on. We notice a significant increase from people coming from SJX to LAX starting from the 3rd quarter of 2007. It is also noticed in places like LGA-BOS there is a seasonal effect since we see in each year the highest number of passengers are travelling in the 2nd quarter and then is reducing as the next quarter approaches.

4. Can you Detect Cascading Delays as Delays in one airport create delays in others?

In order to detect cascading delays, A new date-time column was created and it was sorted with the individual Tail Numbers of the planes. To check the cascaded effect of delays a new column 'NextDelay' was created where the Arrival delays were all shifted one row down. The cascaded effect can be visualized as follows.

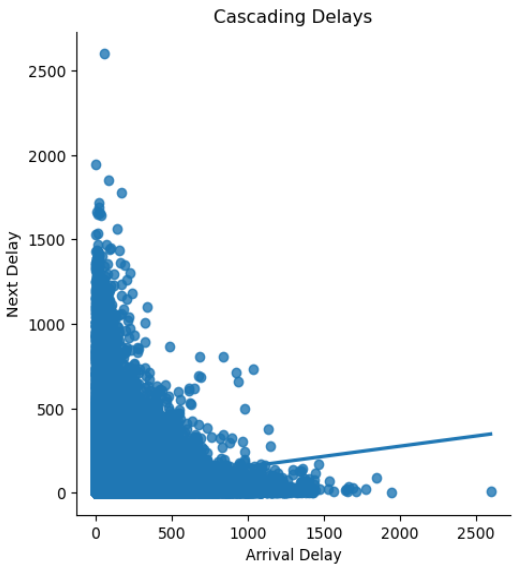


Figure 6 – Python

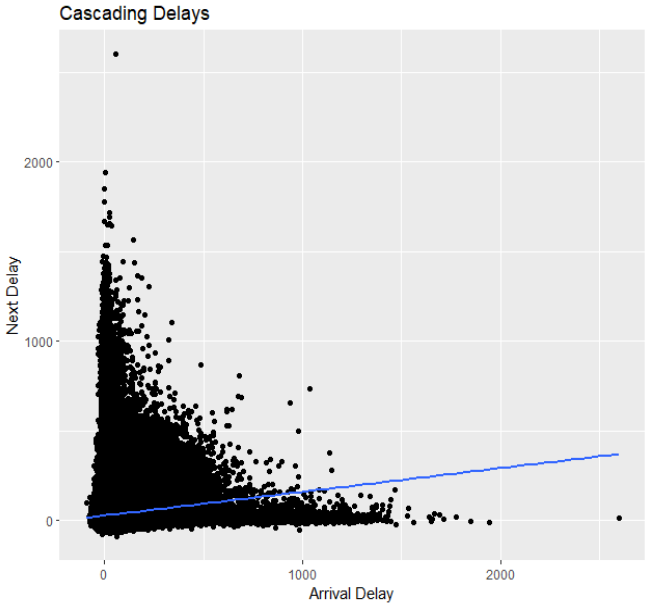


Figure 6 - R

At first glimpse we see most of the points being clustered around each other and is hard to jump to a conclusion immediately, however after calculating the correlation value of 0.13 we see a weak positive correlation between arrival delay in one airport and arrival delay in the next airport.

To analyze this further a cross tabulation table can be displayed,

	No Next Delay	Has Next Delay
No Current Delay	**0.57**	**0.49**
Has Current Delay	**0.43**	**0.51**

Thus from the above provided figures we see that 51% of the flights have experienced cascaded delays (Current and Next delays) and 57% of the flights that experience no delays in the first airport have no delays in the next airport. But since the probability of having a current delay and no delay afterwards is less than the cascaded delay probability we can come to a conclusion that indeed cascading delays are occurred consistently.

6. Use the available variables to construct a model that predicts delays

This question required the fundamentals of machine learning to create the model. A correlation heat map was first plotted after dropping non-numerical columns which would not be correlated with any other columns. The heatmaps were as follows,

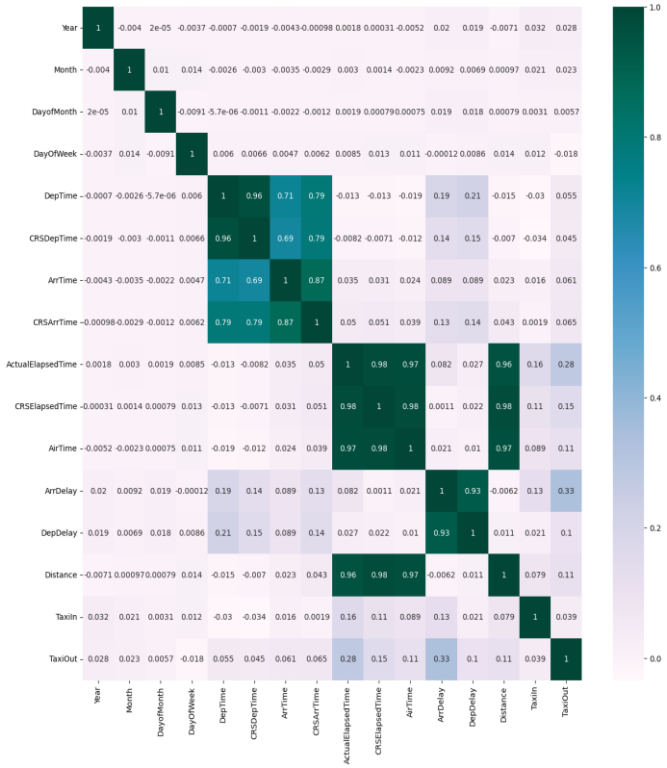


Figure 7 – Python

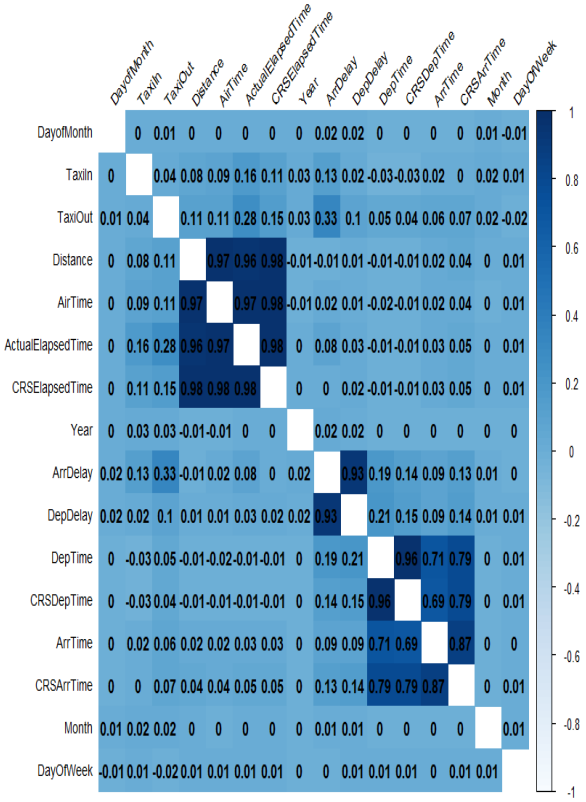


Figure 7 – R

For the delay to be predicted, it was decided to go with the Arrival Delay as the target variable. The features were chosen based on the variables which had the highest correlation with ArrDelay as well as variables which were chosen intuitively based on the previous questions such as Month and Day of Week. All the features taken namely were,

1.Month	2. Day of Week	3. DepTime	4.CRSDepTime
5. CRSArrtime	6. DepDelay	7. TaxiIn	8.TaxiOut

It was decided to execute a linear regression as the type of regression. The data was then split into Training and Testing sets where 75% of the data was allocated to training and rest to testing. The feature columns were then scaled to improve the performance and accuracy of the linear regression. The linear regression was the created and then the performance was evaluated using the following,

Mean Squared Error (MSE)	R Squared(r^2)	Mean Absolute Error (MAE)
10.51^2	0.92	7.34

The Root Mean squared error of 10.51 seems to be a low amount of error for such a big range of Arrival delay values. This can indicate higher performance of the model. The R squared value of 0.92 shows that the model has a good fit with the data and 92% of the total variation of the target variable is explained with the variation of the feature columns. The mean absolute error is less sensitive to outlier values thus giving us an even lower amount of error indicating a good model.

To evaluate this model further we can check whether the residual terms follow a normal distribution which is an MLR condition that a linear regression model for it to provide accurate predictions

A Quantile-Quantile (QQ) plot was plotted to show this.

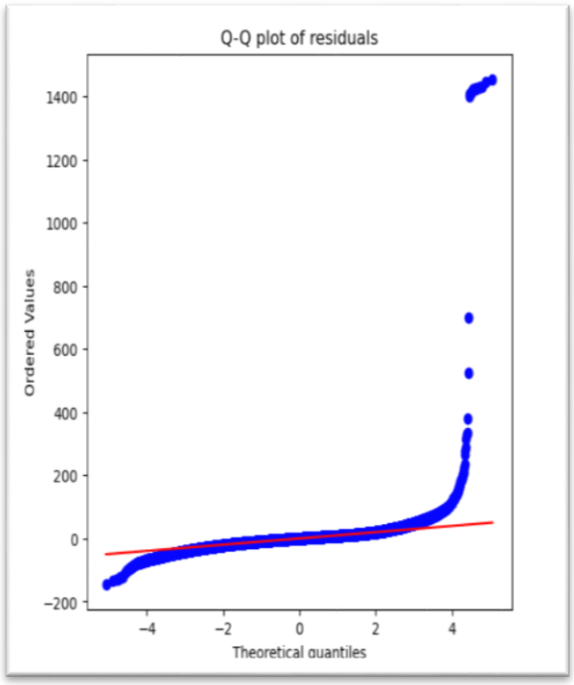


Figure 8 – Python

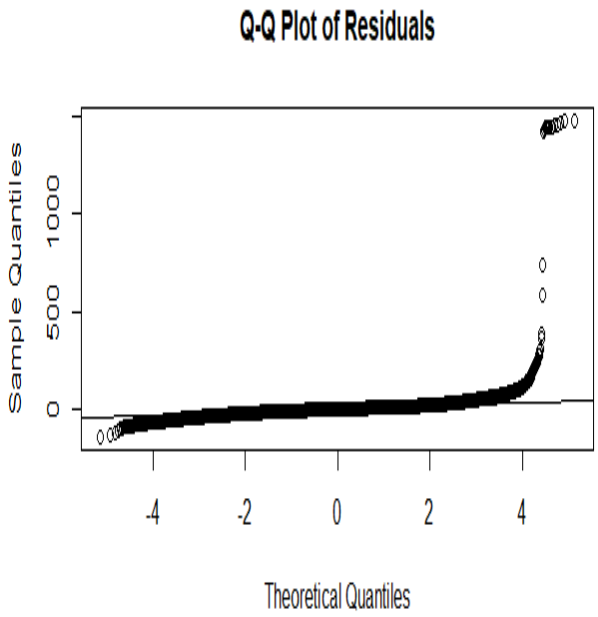


Figure 8 – R

We see from the above diagrams that majority of the residual values seem to be on or close to the normal distribution line with the exception of a few outlier values which could be analyzed further but it was decided to keep them as it is could represent legitimate data.

Overall since most of the points lie close to straight line we can conclude that the residuals indeed follow a normal distribution thus representing a good fit thus the assumptions of the model are met.