Table of Contents

1. Introduction

Based on the dataset downloaded from the Harvard Dataverse, flight data between January 2005 and December 2007 from all commercial flights on major airlines across the United States will be analysed using R and Python to answer the following questions:

1. When is the best time of day, day of week, and time of year to fly to minimise 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 failures as delays in one airport create delays in others?
5. Use the available resources to construct a model that predicts delays.

This report will showcase programming languages such as R & Python to facilitate the analysis of the data provided. SQL would be used within R & Python using libraries such as DBI from R & sqlite3 from Python to query results from databases created.

2. Answering of Questions

2.1 Setting up the workspace

Before answering the questions, the necessary R and Python packages were loaded and the following SQL databases named coursework_r and coursework_py were created. As mentioned earlier, only flight data from January 2005 and December 2007 will be utilized in this analysis. The following tables are created in each respective database:

- "airport" table contains data on the geographical locations of airports, cities & states.
- "carrier" table contains data on the airline codes and names of flight carriers.
- "planes" table contains data on the different types of planes manufactured.
- "ontime" table contains data on commercial flights from January 2005 to December 2007.

2.2 When is the best time of day, day of week, and time of year to fly to minimise delays?

2.3.1 Code Explanation

To handle the anomalies identified in the "ontime" table, arriving and departing times that exceeded 2400 were adjusted by subtracting 2400 during the data wrangling process.

Four metrics were identified to measure air travel performance for the best time of day, day of the week, and time of year to fly and to minimise delays. These metrics include average arrival delay, the percentage of cancelled and delayed flights, and on-time performance, which measures punctuality within 15 minutes of the scheduled arrival time. Arrival delay is measured instead of departure delay as arriving late can be more disruptive to passengers and airlines working on a schedule. The percentage of diverted flights was not considered as the percentages were found to be negligible.

To improve recommendations on the time to fly, additional columns were introduced into the "ontime" table in the database. The first column introduced was named "DepTimeInterval", which groups flights into 2-hour intervals based on their scheduled departure time. The scheduled departure times were divided into 2-hour intervals as it would be more appropriate to recommend flying within a time interval rather than a specific departure time. The second introduced was "Season", which categorizes flights into seasons based on their month of operation. Flights from March to May are classified as Spring, flights from June to August as Summer, flights from September to November as Autumn, and flights from December to February as Winter.

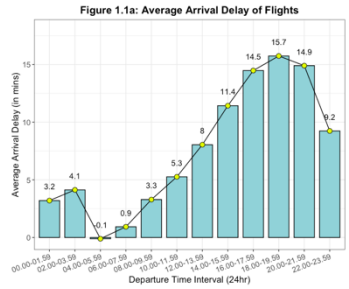
SQL queries are executed on the "ontime" table to obtain the average arrival delay, the percentage of cancelled and delayed flights, and the on-time performance of flights. To

determine the best time of day to fly, the four metrics were calculated for each departure time interval (in 2-hour intervals), excluding cancelled and diverted flights. The results were then grouped and ordered by their departure time intervals to provide insight into which time intervals experience the least delays to help identify the best time of day to fly.

To identify the best day of the week and time of year to fly, the same approach mentioned above was replicated. However, the results will be further grouped and ordered by season on top of the day of the week and month respectively. This was not introduced to identify the best time of day to fly as it affects the interpretability of the data visualizations due to the number of time intervals. Although the time intervals could have been broadened to improve interpretability, it would affect the precision of the recommendation, e.g. 0000 to 0159 compared to 0000 to 0259.

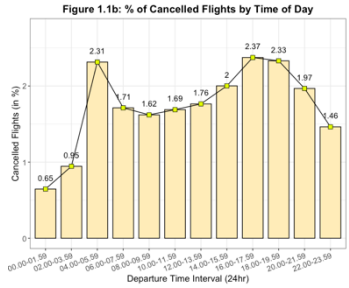
To illustrate the data obtained via the SQL queries, packages such as ggplot2 in R and matplotlib and seaborn in Python were utilised.

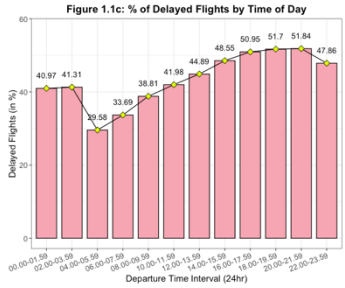
2.3.2 Analysis of Results

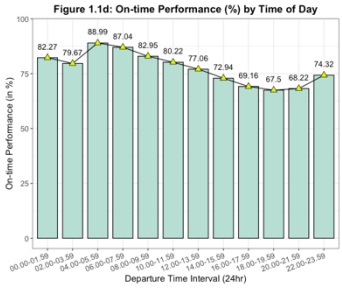


Best time to fly to minimise delays

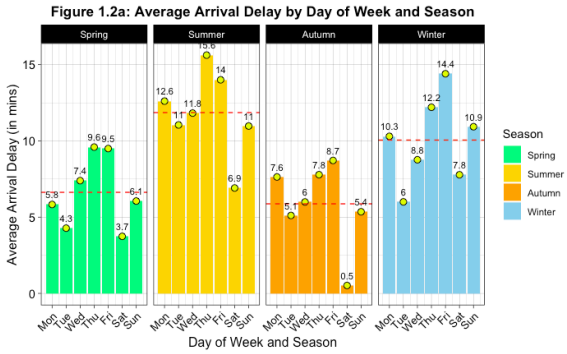
Figure 1.1a illustrates the average arrival delay by departure time interval from 2005 to 2007. Flights departing between 0400 to 0559 are observed to be the most ideal as they pretty much arrive on time (-0.1 mins). Another departure time interval worth noting is 0600 to 0759, whereby flights experienced a little less than a minute's delay (0.9 mins) on average. These two time intervals will be compared to determine the best time to fly for minimizing delays.





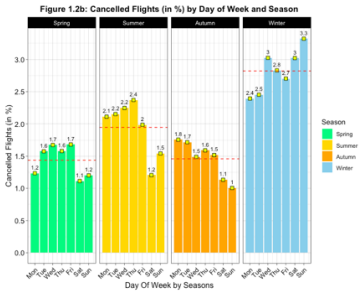


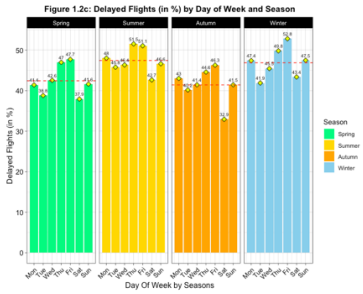
The figures above displays the percentages of cancelled, delayed and on-time performance of flights by departure time interval from 2005 to 2007. Comparing the two departure time intervals of 0400 to 0559 and 0600 to 0759, flights departing between 0400 to 0559 had a higher cancellation rate of 2.31% compared to 1.71% for flights departing between 0600 to 0759. Though the percentage of delayed flights was lower for flights departing between 0400 to 0559 at 29.58% compared to 33.69% for flights departing between 0600 to 0759. In terms of on-time performance, flights departing between 0400-0559 had a higher likelihood of arriving within 15 minutes of the scheduled time at 88.99%, compared to 87.04% for flights departing between 0600-0759. Based on the findings, the best time to fly to minimize delays is between 0400 to 0559.

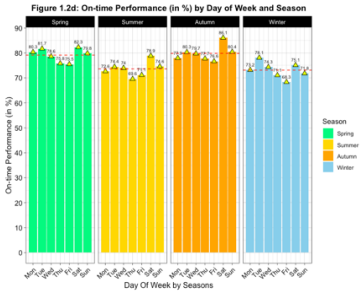


Best day of week to fly to minimise delays

Figure 1.2a illustrates the average arrival delay by day of week and season from 2005 to 2007. Saturdays have the lowest average arrival delay (4.7 mins) of all days, with the lowest delay on Saturdays in Autumn at only 0.5 mins. Therefore, to identify the best day of the week to fly to minimise delays, the air travel performance on Saturdays will be further looked at. Additionally, since Spring and Autumn experience similar average arrival delays, these two seasons will be compared to determine the best day to fly across all four seasons.

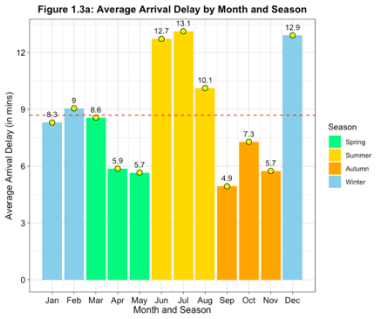






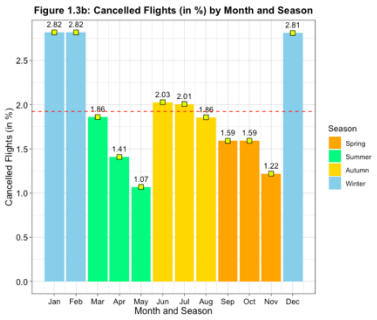
Looking at figure 1.2b, when considering the percentage of cancelled flights, Saturdays tend to have the lowest (1.6%) while there isn't a significant difference between Spring and Autumn, with both having relatively low figures on average. Figures 1.2c and 1.2d further imply that Saturday is the most ideal day to fly with the lowest percentage of delayed flights (39.2%) and the highest on-time performance on average (80.6%). However, the two figures also reveal that Saturdays in Autumn have the lowest percentage of delayed flights (32.9%) compared to Saturdays in Spring (37.9%), a considerable difference of 5%. In terms of on-time performance, Saturdays in Autumn (86.1%) rank the highest, 3.8% higher than Saturdays in Spring (82.3%).

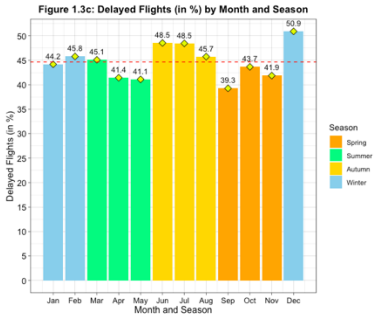
Hence, Saturdays are the best day of the week to fly as they have consistently demonstrated the most favourable results across the four metrics while it can be concluded that Saturdays in Autumn are the best day to fly across all four seasons.

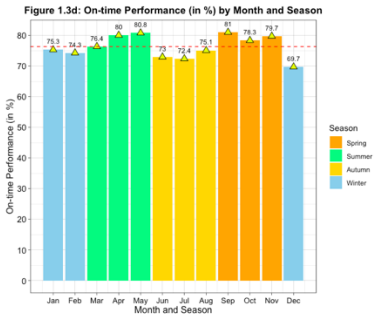


<u>Best time of year to fly to minimise delays</u>

The bar chart on the left shows the average arrival delay by month and season between January 2005 to December 2007. September had the lowest average arrival delay (4.9 min), followed by May and November (both 5.7 mins). Autumn had the lowest average delay (6 mins) of all seasons, as compared to Spring (6.7 mins). To determine the best time of year for minimising flight delays, the performance metrics of May, September and November will be compared, while also evaluating the seasons of Spring and Autumn.







In order to determine the best time of year to fly, the percentage of delayed flights and on-time performance will be prioritised over the percentage of cancelled flights as it is generally low. September stands out as the best month to fly with the least percentage of delayed flights (39.3%) and the best on-time performance (81%) when compared to other months. Thus, September is the best month to fly in order to minimise delays.

By taking the dotted red line as a benchmark, which represents the average of each of the four metrics, it can be observed that Spring had favourable results in all four charts. However, only Autumn consistently performed better than the benchmark in all four metrics, proving that Autumn is the best season to fly to minimise delays as established previously.

2.3.3 Conclusion

Based on the findings from the analysis, it can be concluded that:

1. 0400 to 0559 is the best time of day to fly to minimise delays.
2. Saturdays, particularly in Autumn, are the best day of the week to fly.
3. September is the best month to fly for minimal delays.
4. Autumn is the most favourable season to fly.

2.3 Do older planes suffer more delays?

2.3.4 Code Explanation

As there are no specific columns in the “planes” table that explicitly state the manufacturing year of the planes, the “year” column in the “planes” table is assumed to be the year the planes are manufactured moving forward.

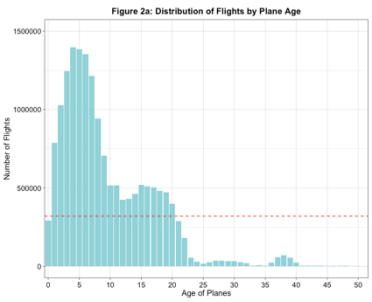
To understand whether older planes suffer more delays, the age of planes, average arrival delays of flights, as well as the percentage of delayed flights based on their age, will be calculated for flights between January 2005 and December 2007, average arrival delay is measured as it can be more disruptive to passengers in terms of subsequent travels as mentioned previously.

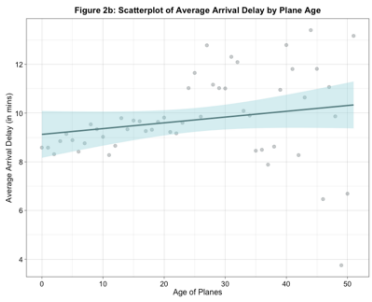
SQL queries are performed on the “planes” and “ontime” tables using an inner join referencing the tail number of planes as the foreign key to join the two tables. To gain an overview of the number of flights by plane age, the age of planes will be calculated by subtracting the “Year” column in the “ontime” and “year” column in the “planes” table, where “year” indicates the year of flight. Anomalies in the age of planes were identified due to abnormality in the “year” column in the “planes” table, as there were missing values and instances where the manufacturing year of planes was newer than the year of flight.

Upon further data exploration, it was discovered that the oldest plane in operation was manufactured in 1956, meaning that the oldest plane should be 51 years old. To deal with the erroneous data, the range for the age of planes will be specified in the query to only include planes aged between 0 and 51. Thus, only flights that experience arrival delays greater than 0 will be included in the queries used to obtain the average arrival delays and percentage of delayed flights by plane age.

To illustrate the data obtained via the SQL queries, packages such as ggplot2 in R and matplotlib and seaborn in Python were utilised.

2.3.5 Analysis of Results





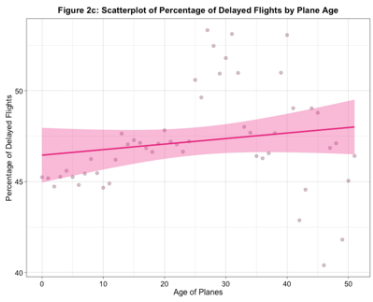
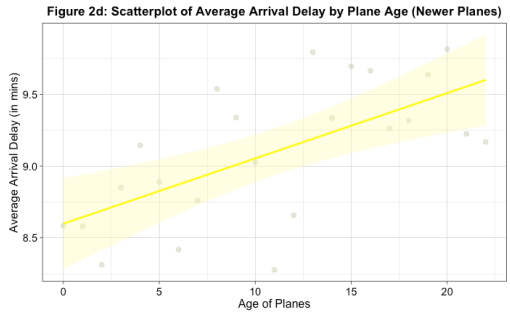
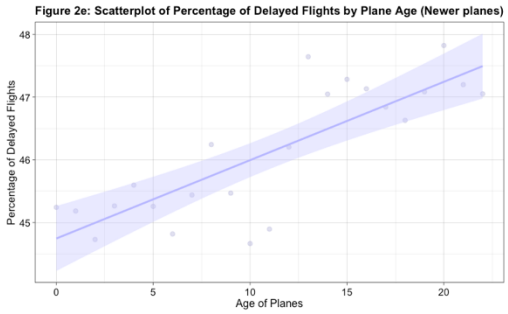


Figure 2a illustrates the distribution of flights during the period January 2005 to December 2007, categorized by the age of the planes used. The chart highlights that the majority of flights (16,050,698) were operated using planes aged 22 years or younger, while only a small portion (579,324) used planes older than 22 years old.

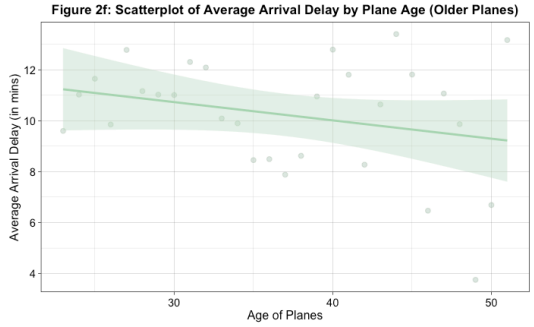
Figures 2b and 2c demonstrate that the average arrival delay and percentage of delayed flights rise with the age of planes, as indicated by the best-fit line using linear regression. However, the scatterplots exhibit a disparity as points on the left halves of the plots are closer to the regression line, while the points are the right halves are dispersed.

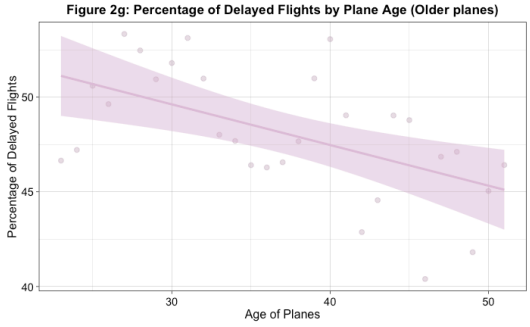
To determine whether older planes tend to encounter more delay, flights will be divided into two groups, one with only planes aged 22 or younger, and the other with planes older than 22 years, based on the distribution of flights in Figure 2a. The former group will be referred to as “Newer planes”, while the latter will be called “Older planes”.





Replicating the same scatterplots for newer planes (0 to 22 years) demonstrates that older planes within the group do tend to suffer more delays since the regression line is much steeper than in figures 2b and 2c. This is important because this group of newer planes has a considerably higher number of flights than planes aged older than 22 years, making it more representative of the dataset.





However, in the older group of planes (23 to 51 years), the scatterplots present a different scenario, as the average arrival delay and percentage of delayed flights decrease as the age of planes increases, contradicting the findings of this analysis thus far. Although, it is important to note that this group of planes had a significantly smaller volume of flights compared to the group of newer planes, which may make it less representative of the overall dataset.

Overall, considering the trends observed in the analysis, along with the clearer relationship between age and delay in the scatterplots for the group of newer planes, provides justification that older planes do suffer delays. While there may be some variation in the scatterplots for the group of older planes, it is important to consider the overall trend and the size of the groups being compared.

2.3.6 Conclusion

The analysis shows that older planes do tend to suffer more delays. This is further reinforced by the fact that the group of newer planes (0 to 22 years) constitutes a significant portion of the dataset, making it more representative.

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

2.4.1 Code Explanation

Before answering the question, the assumption is made that each flight carries the same number of passengers for each trip. This assumption is made due to the lack of information on the number of passengers on each flight. Having established that, to explain how the number of people flying between different locations changes over time, the following will be identified:

1. Top 10 US airports based on air traffic volumes
2. Top 10 US states based on air traffic volumes
3. Top 10 US cities based on air traffic volumes
4. Top 10 US flight routes based on air traffic volumes

Using SQL in R and Python to obtain the traffic volumes on the top 10 airports, states and cities based on the number of inbound flights. The “airports” and “ontime” tables were joined using an

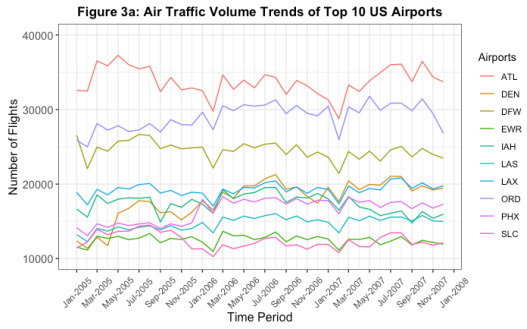
inner join to obtain the names and number of flights from the airports, states and cities, excluding cancelled and diverted flights, sorting the top 10 results for each respective variable in descending order.

To create a sequential timeline of the results in terms of date and time, a new DateTime column named “YearMonth” was created, which combined the values of the “Year” and “Month”. The day value was set to “01” to standardize the dates, and the resulting string was converted to a DateTime object using the as.Date() function in R and pd.to.datetime() in Python. Doing so allows the results of findings to be easily plotted over time, to project how the number of people (flights) between different locations changes over time.

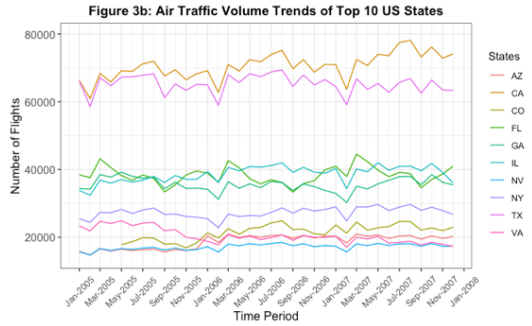
To illustrate the data obtained via the SQL queries, packages such as ggplot2 in R and matplotlib and seaborn in Python were utilised.

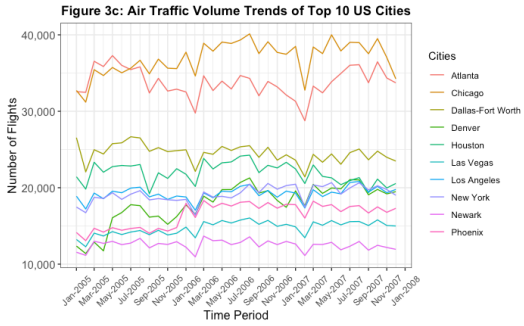
2.4.2 Analysis of Results

Figure 3a shows the top 10 airports in the US based on air traffic volumes from January 2005 to December 2007. The line chart shows a general decline in the number of flights from January to February each year, followed by a sharp increase in March likely due to the spring break holidays in the US, resulting in higher domestic travel. Similarly, flight volume tended to rise from May to August before declining in September, corresponding to the end of the summer break.



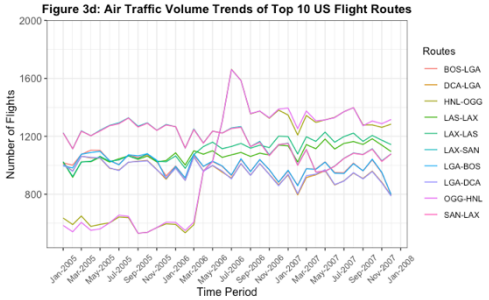
In addition, there was an increase in flight volume between September and October in each of the three years, which could be attributed to Labour Day being a popular weekend travel in the US to mark the end of summer. These trends can be attributed to seasonal effects related to school holidays and public holidays in the US as mentioned.





Figures 3b and 3c illustrate the air traffic volume trends for the top 10 US states and cities respectively from January 2005 to December 2007. Similar trends can be observed in both line charts when compared to figure 3a, where the number of flights generally tends to decline from January to February for each year, followed by the increase in flight volume in March and the dip in flights can also be observed from August to September. Additionally, the same trend in increased flights between September and October can also be observed in both charts.

Figure 3d depicts the top 10 flight routes in the US based on air traffic volumes from January 2005 to December 2007. While the line chart confirms the trends previously observed, it also reveals a significant increase in the popularity of flights between OGG (Kahului Airport) and HNL (Honolulu International Airport) in Hawaii.



Despite being the least popular of the top 10 routes, the number of flights between the two airports skyrocketed from March to July of 2006, making it the most popular route for the rest of 2006 and 2007 as Hawaii visitor numbers peaked in 2006.

2.4.3 Conclusion

Based on flight data in the US between 2005 and 2007, the changes in the number of people flying between the top 10 US airports, states, cities, and flight routes based on air traffic volumes are due to the influence of seasonal effects such as school and public holidays and travellers' preferences, attributed to the following trends:

- Flight volumes generally decline in January and February, followed by a sharp increase in March, likely due to spring break.
- Summer months also see higher flight volumes, with a decline in September following the end of summer break.
- There is an increase in flight volume between September and October due to the popularity of Labour Day weekend travels in the US.
- The increased popularity of flights between OCG and HNL in Hawaii highlights the impact of traveller preferences on air traffic volume.

2.5 Can you detect cascading failures as delays in one airport create delays in others?

2.5.1 Code Explanation

To detect whether cascading failures as delays in one airport will lead to delays in others, the journey of a flight that experienced an arrival delay will be analyzed. Specifically, observing any disruptions on the plane's subsequent departure and flight journeys in other airports to determine whether there are cascading failures.

In order to select the most appropriate airport and flight route for this analysis, network analysis was conducted to identify the critical nodes and edges within the flight network, using the igraph package in both R and Python. These critical nodes and edges have a greater potential of disrupting the network given their influence on the overall functioning of the flight network. In the context of this question, airports are represented by nodes while flight routes are represented by edges, existing within the flight network of the USA.

The most important node was identified by evaluating its degree and betweenness centrality scores, with the node exhibiting the highest scores deemed the most crucial. Meanwhile, critical edges were identified based on their betweenness centrality scores. Only flight records with ArrDelay greater than zero and not null were included in the analysis to ensure that only complete data was used. To account for the effect of delays on traffic flow throughout the airport network, arrival delays were chosen as the sole weight in the calculation of node and edge betweenness centrality scores as it provides a more accurate measure of the actual delay experienced by a flight.

Airport	Degree
ATL	380
ORD	296
DFW	286
CVG	272
MSP	261

Airport	Betweenness Centrality Score
ATL	28302.319600
SLC	18331.655073
DFW	16095.283222
CVG	14662.382787
IAH	14347.080481

From	To	Betweenness Centrality Score
MDW	ATL	6055.638925
TPA	SLC	5903.322294
CVG	DFW	5190.037118
CLT	TPA	5103.759343
SLC	ANC	3765.456421

Based on the information presented in the tables above, ATL, William B Hartsfield-Atlanta Intl, emerges as the most crucial airport with the highest degree and betweenness centrality scores, signifying that it has the most connections and is the most critical intermediary in facilitating the flow of traffic between other airports in the network. Additionally, the edge between MDW-ATL has the highest betweenness centrality score, indicating its significant role in connecting other airports in the network, not just the origin and destination airports.

Therefore, flights involving ATL and MDW-ATL will be analyzed as there is an increased probability of detecting cascading effects of delays since these are the critical nodes and edges in the flight network.
Having identified the most critical airport and flight route in the network, the next step is to monitor the flight journey of a plane that has experienced an arrival delay and has interacted with the critical airport.

Three different flights of the same flight route with low, medium and high arrival delays were selected to compare the impact of different delays on cascading delays to other flights and airports in the network using SQL in both R and Python. These flights are:

1. On February 22, 2005, flight N3736C experienced a 15-minute delay upon arrival at ATL. (This delay is considered the lowest degree of delay as the average delay for the MDW-ATL route was 14.69 minutes.)
2. On March 17, 2005, flight N948AT experienced a 60-minute delay upon arrival at ATL. (This delay is considered medium.)
3. On October 21, 2006, flight N878AS experienced a 120-minute delay upon arrival at ATL.

2.5.2 Analysis of Results

	Year	Month	DayofMonth	DepTime	CRSDepTime	ArrTime	CRSArrTime	TailNum	DepDelay	ArrDelay	Origin	Dest	Departure Airport	Arrival Airport
0	2005	1	22	947.0	920	1228.0	1213	N3736C	27.0	15.0	MDW	ATL	Chicago Midway	William B Hartsfield-Atlanta Intl
1	2005	1	22	1314.0	1303	1513.0	1417	N3736C	11.0	56.0	ATL	MCI	William B Hartsfield-Atlanta Intl	Kansas City International
2	2005	1	22	1551.0	1440	1839.0	1734	N3736C	71.0	65.0	MCI	ATL	Kansas City International	William B Hartsfield-Atlanta Intl

Figure 4a: Flight journey of flight N3736C on 22nd January 2005 (Low degree of arrival delay)

As seen above, flight N3736C experienced delays in both departure and arrival times for its flight from Chicago Midway airport to William B Hartsfield-Atlanta International airport. The plane was scheduled to depart at 0920 hours but departed 27 minutes late at 0947 hours instead, causing the plane to only arrive at William B Hartsfield-Atlanta International airport at 1228 hours instead of the scheduled 1213 hours, resulting in an arrival delay of 15 minutes.

In its subsequent flight, flight N3736C was scheduled to depart from William B Hartsfield-Atlanta International airport at 1303 hours but departed late at 1314 hours, resulting in a departure delay of 11 minutes. It was scheduled to arrive at Kansas City International airport at 1417 hours but only arrived at 1513 hours, 56 minutes later than scheduled.

The resulting delay led to a departure delay of 71 minutes from Kansas City International airport, departing at 1551 hours instead of 1440 hours. Although the delay was slightly reduced, the plane only arrived at 1839 hours instead of the scheduled 1734 hours, resulting in an arrival delay of 65 minutes. Therefore, cascading failures were detected as delays in one airport created delays in others.

Year	Month	DayofMonth	DepTime	CRSDepTime	ArrTime	CRSArrTime	TailNum	DepDelay	ArrDelay	Origin	Dest	Departure Airport	Arrival Airport
2005	1	22	1632.0	1520	1915.0	1815	N948AT	72.0	60.0	MDW	ATL	Chicago Midway	William B Hartsfield-Atlanta Intl
2005	1	22	1940.0	1905	2005.0	1926	N948AT	35.0	39.0	ATL	MEM	William B Hartsfield-Atlanta Intl	Memphis International
2005	1	22	2022.0	1958	2230.0	2214	N948AT	24.0	16.0	MEM	ATL	Memphis International	William B Hartsfield-Atlanta Intl

Figure 4b: Flight journey of flight N948AT on 22nd January 2005 (Medium degree of arrival delay)

Flight N948AT was scheduled to depart from Chicago Midway airport at 1520 hours but only did so 72 minutes later at 1632 hours which led to its delayed arrival at William B Hartsfield-Atlanta International airport at 1915 hours, an hour later than scheduled. This delay then caused a departure delay of 35 minutes for its subsequent flight, resulting in an arrival delay of 39 minutes at Memphis International airport. Furthermore, the subsequent flight back to William B Hartsfield-Atlanta International airport was delayed for 24 minutes, despite the arrival delay being reduced to 16 minutes late.

Year	Month	DayofMonth	DepTime	CRSDepTime	ArrTime	CRSArrTime	TailNum	DepDelay	ArrDelay	Origin	Dest	Departure Airport	Arrival Airport
2006	10	21	1225.0	1020	1511.0	1311	N878AS	125.0	120.0	MDW	ATL	Chicago Midway	William B Hartsfield-Atlanta Intl
2006	10	21	1547.0	1455	1700.0	1613	N878AS	52.0	47.0	ATL	RDU	William B Hartsfield-Atlanta Intl	Raleigh-Durham International
2006	10	21	1725.0	1645	1850.0	1810	N878AS	40.0	40.0	RDU	ATL	Raleigh-Durham International	William B Hartsfield-Atlanta Intl

Figure 4c: Flight journey of flight N878AS on 21st October 2006 (High degree of arrival delay)

Similarly to the two flights prior, flight N878AS experienced persistent departure and arrival delays throughout its flight journey, albeit the size of the delays was consistently decreasing, cascading failures were detected for the third time as delays in one airport create delays in others.

To sum up, the flight journeys of N3736C, N948AT, and N878AS provide evidence of how delays in one airport can indeed create cascading failures and affect flight journeys in other airports despite the varying degrees of arrival delays experienced.

2.5.3 Conclusion

Cascading failures as delays can be detected as delays in one airport will create delays in others, as proved on three occasions with the flight journeys of N3736C, N948AT, and N878AS.

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

2.6.1 Code Explanation

The models were constructed to predict the arrival delays of flights as the target variable based on flight data between January 2005 and December 2007, using packages such as ml3 and sklearn in R and Python. The features chosen to build models are:

Numerical features									
1.	Year	2.	Month	3.	DayofMonth	4.	DayOfWeek	5.	DepTime
6.	CRSDepTime	7.	ArrTime	8.	CRSArrTime	9.	AirTime	10.	DepDelay
11.	Distance	12.	CarrierDelay	13.	WeatherDelay	14.	NASDelay	15.	SecurityDelay
16.	LateAircraftDelay	17.	PlaneAge (New)						

The models to be constructed for the prediction of arrival delays are:

1. Linear regression model
2. Lasso regression model
3. Ridge regression model
4. Random forest regression model

Using SQL in R and Python, the dataset consisting of the target and feature variable columns was selected. To calculate the age of planes, the “ontime” and “planes” tables were joined based on the tail number of flights as the foreign key. To ensure that only complete data was used in the construction of the models, cancelled or diverted flights were excluded. The age of planes was limited to a maximum of 51 years, as this was the oldest age mong the flights in the dataset, as determined in question 2. The dataset was then inspected for missing values using the skim function in R and Python.

200,000 flight records were randomly sampled from the dataset. The sample was then split into train and test sets, with the train set representing 70% of the total sample size and the test set representing the remaining 30%. For the data sampling and splitting, the set.seed function and random_state function in R and Python respectively were set to 123 to ensure the reproducibility of results.

To prepare the sample data for modelling, both in R and Python, a pipeline of computational steps were defined to pre-process the data. This pipeline includes imputing any missing values and scaling the numerical features to have zero mean and unit variance, before being used to pre-process both the training and testing data for each model.

Hyperparameter tuning was performed in R and Python by defining the tuning environment, specifying a range of values for the hyperparameters and using cross-validation to evaluate the performance of each combination of hyperparameters in order to find the optimal value of the regularization parameter lambda. Grid search was used to tune hyperparameters for all models except linear regression. Additionally, a limited number of evaluations were performed for each combination of hyperparameters, in order to avoid overfitting the training data.

Finally, learners were created for each model and trained on the pre-processed data, using the best parameters found during tuning. These learners were then used to predict the target variable on the testing data.

2.6.2 Analysis of Results

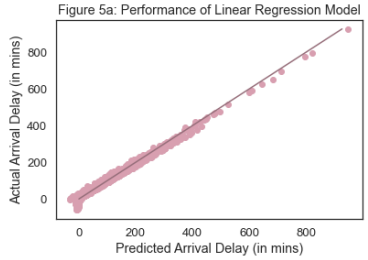


Figure 5a: Performance of Linear Regression Model

Figures 5a, 5b, 5c and 5d provide a visual aid to get a gist of the performances of each model by comparing the predicted values (x-axis) to the actual values (y-axis). It is important to note that these illustrations do not necessarily provide a definitive assessment of the models' performance. A perfect prediction would be indicated by all points on the scatter plot falling exactly on the diagonal line that runs from the origin and splits the plot in half.

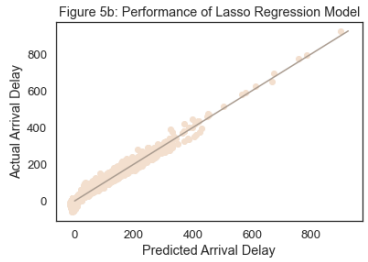


Figure 5b: Performance of Lasso Regression Model

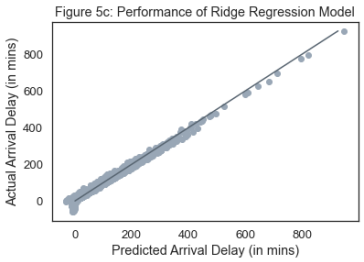


Figure 5c: Performance of Ridge Regression Model

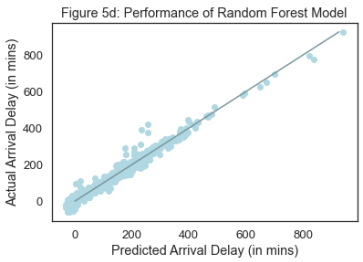


Figure 5d: Performance of Random Forest Model

To evaluate the performances of each model on its ability to predict arrival delay, the metrics used to assess the models are the root mean squared error (RMSE), mean squared error (MSE), mean absolute error (MAE), R-squared (r2). Lower values for RMSE, MSE and MAE indicate better performance, while higher values for R-squared indicate better performance.

	Linear	Lasso	Ridge	Random Forest
RMSE	8.345442	8.839547	8.344866	7.141793
MSE	69.646397	78.137589	69.636781	51.005214
MAE	6.428607	6.861608	6.428862	4.895192
r2	0.947328	0.940907	0.947336	0.961426

Figure 5e: Table comparing the metrics of each model

Based on the results presented in Figure 5e, the random forest model outperformed the other models in predicting arrival delays, with the lowest values of RMSE (7.141793), MSE (51.005214), and MAE (4.895192), and the highest R-squared value (0.961426). It is noteworthy that the metrics of the linear and ridge regression models were very similar, while the lasso regression model had the least desirable performance.

2.6.3 Conclusion

Of the four models created to predict arrival delays, the random forest model has proved to be the most capable based on the metrics used to evaluate the performances of the models.

3 Additional Information

To ensure high-quality visualizations, only the most visually appealing plots generated using either R or Python have been selected for the analysis of results.