An Analysis of Commercial Flight Patterns in the USA: 1987-2008

Table of contents

1.Introduction

This project provides an analysis of flight arrival and departure data for all commercial flights within the USA, from October 1987 to April 2008. The data is extensive, consisting of nearly 120 million records and taking up 1.6 gigabytes of space when compressed and 12 gigabytes when uncompressed. The significance of this data lies in the wealth of information it provides about commercial flight patterns within the USA over a period of more than 20 years. This data can be used to identify trends and patterns in flight traffic, such as the most popular routes, busiest times of the year, and the

performance of different airlines. This information can be used by researchers, policymakers, and industry professionals to make data-driven decisions about the aviation industry and improve the efficiency and safety of commercial flights. Additionally, this data can also be used to analyze how weather, economic conditions, and other factors have affected flight patterns over time. Overall, this project aims to provide a comprehensive analysis of the data to inform policy and decision making, and to improve the performance of the aviation industry.

2.Dataset description

This dataset includes detailed information on the arrivals and departures of all commercial flights within the USA from October 1987 to April 2008. The dataset is quite large, containing approximately 120 million records and requiring 4.2 gigabytes of storage space in its compressed form and 12 gigabytes when uncompressed. To assist with working with such a large dataset, the user is provided with two brief introductions to useful tools such as Linux command line tools and SQLite, a simple SQL database. These tools can help the user to navigate, manipulate and analyze the data more easily.

Name	**Description**
Year	1987-2008
Month	1-12
DayofMonth	1-31
DayOfWeek	1 (Monday) - 7 (Sunday)
DepTime	actual departure time (local, hhmm)
CRSDepTime	scheduled departure time (local, hhmm)
ArrTime	actual arrival time (local, hhmm)
UniqueCarrier	unique carrier code

FlightNum	flight number
TailNum	plane tail number
ActualElapsedTime	in minutes
CRSElapsedTime	in minutes
AirTime	in minutes
ArrDelay	arrival delay, in minutes
DepDelay	departure delay, in minutes
Origin	origin IATA airport code
Dest	destination IATA airport code
Distance	in miles
TaxiIn	taxi in time, in minutes
TaxiOut	taxi out time in minutes
Cancelled	Was the flight cancelled?
CancellationCode	reason for cancellation (A = carrier, B = weather, C = NAS, D = security)
Diverted	1 = yes, 0 = no
CarrierDelay	in minutes
WeatherDelay	in minutes
NASDelay	in minutes
SecurityDelay	in minutes
LateAircraftDelay	in minutes

3.Data Analysis

a. Proposed analysis :

1- What are the main reasons for flight cancellation ?

2- What are the most airports that were the flight origin?
3- What are the most airports that were the flight destination?4- When is the best time of day/day of week/time of year to fly to minimize delays?
5- Do older planes suffer more delays?
6- What are the flight numbers with the most delays?
7- What are the delays of different companies ?
8- How does the number of people flying between differentlocations change over time?
9- How well does the weather predict plane delays?
10- What were the most years that suffered longer delays?
11-What were the most Airports combinations(origin, destination) that suffered delay?

b. Data problems :

We found that the best python object that can hold the full match airline taken using Kaggle APIs were spark data frames. The information includes arrival and departure information for all commercial flights within the USA between October 1987 and April 2008. There are over 120 million records in this enormous dataset, which requires 12 gigabytes of space when uncompressed and 1.6 gigabytes of space when compressed.

c. Most Important Columns :

- Arrival time Vs. Scheduled Arrival time
- Departure time Vs. Scheduled Departure time
- Unique carrier (1491 code)
- Flight number
- Delay reason
- Cancellation and its code ● Detailed date and time of the flight ● Origin, destination, and air time.

4.Challenges

a. Challenges in Data Acquisition:

The dataset for this project is quite large, containing nearly 120 million records and requiring 4.2 gigabytes of storage space in its compressed form and 12 gigabytes when uncompressed. To acquire the data, we intended to use the Kaggle API to download the dataset. However, due to the large size of the data,

we also employed additional measures to ensure efficient data acquisition. Specifically, we utilized Google Colab to download the data to Google Drive, which facilitated faster download speeds. Furthermore, we used a compression tool to shrink the data size. Our choice of compression algorithm was BZ2, as it is known to be highly efficient and is a separable compression algorithm. This enabled the data to be compressed from around 12.5 GB to 2.7 GB.

b. Challenges in Data Exploration:

Due to the large size of the dataset, it was challenging to perform an exploratory analysis of the data. To address this challenge, we decided to take a sample of the data to develop our logic on it. We set certain criteria to ensure that this sample was representative of the entire dataset. Specifically, we selected a sample of 100,000 rows, which is considered to be a large enough sample size. Additionally, the data was shuffled before sampling to ensure that the 100,000 rows were well distributed.

c. Challenges in Data Analysis:

The dataset did not contain certain information that was required for the analysis, such as the age of the planes or the weather conditions on every trip. To address this challenge, we devised methods to infer this information from the available data. For example, to determine the working years of the planes, we counted the unique years that the planes traveled in and identified the planes that worked for more than 11 years. We then compared the first 6 years of operation with the last 6 years of operation. Additionally, to obtain information on weather conditions, we obtained the dates of days that had flight cancellations due to weather and calculated the average delays on these days.

5.Final design pipeline

We used PySpark to process a large dataset of airline flight information. The first step is to import the necessary libraries and create a SparkSession. The data is then read in from a CSV file, and any missing values in the "ArrDelay" column are dropped or filled with either 0 or an empty string.

Next, a user-defined function (UDF) is created to extract the hour from the

"CRSDepTime" column, and a new column "Hour" is added to the dataframe with the results. The data is then grouped by this new "Hour" column, and the average arrival delay is calculated and sorted.

Similar operations are performed on the other columns of the data, such as "DayOfWeek", "DayofMonth", "Month", "Year", "FlightNum", "Origin", and "Dest". The average arrival delay is calculated and sorted for each of these groupings.

The code also looks at the effect of different reasons for flight cancellations by grouping the data by "CancellationCode" and counting the number of occurrences of each code. It also examines the impact of bad weather on flight delays by filtering the data for cancellation codes related to weather and grouping it by various columns such as "DayofMonth", "Month", and "DayOfWeek".

Finally, the code looks at the impact of older planes on flight delays by grouping the data by "TailNum" and "Year" and counting the number of flights for each combination. Then, it filters the data to only include planes that have been in service for more than 11 years and calculates the average arrival delay for each of those planes and year.

Overall, this code pipeline performs various aggregation and grouping operations on the airline flight data to explore patterns and trends in flight delays, cancellations, and the impact of various factors on these outcomes. The results are then written out to separate CSV files for further analysis.

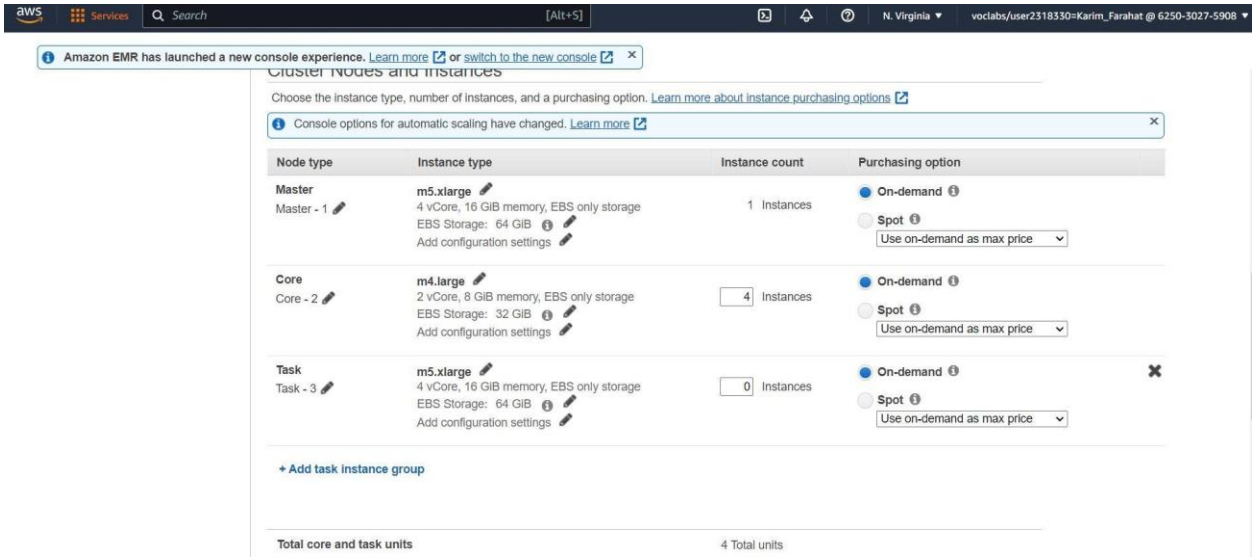
6.Optimization

some optimization techniques used in this code are:

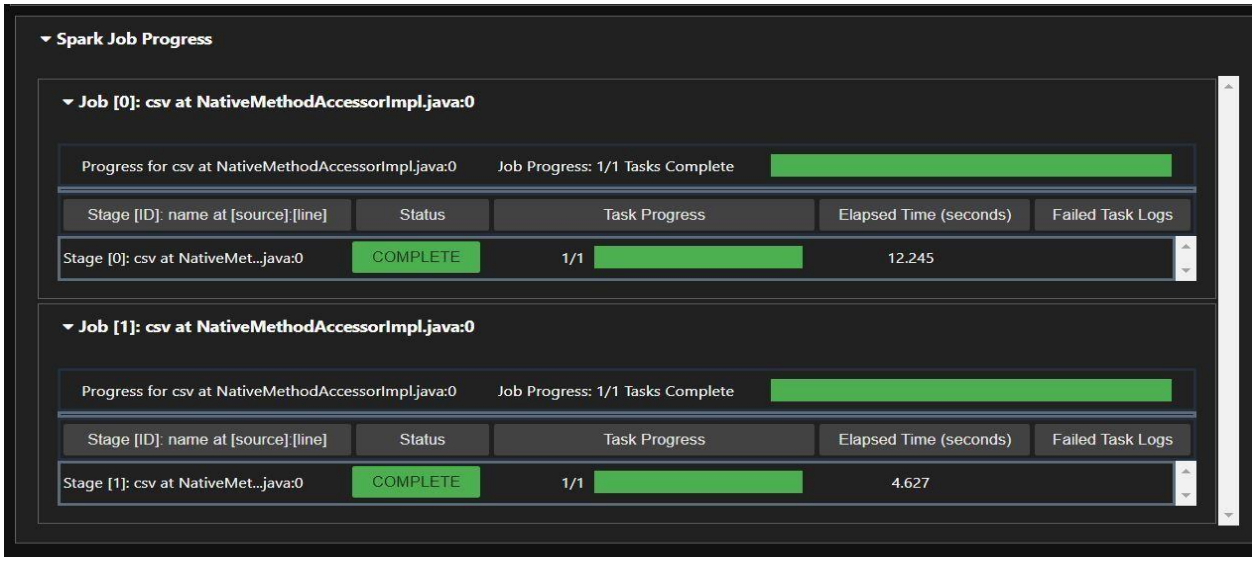
- The Data is comprised to bz2 format in order for pyspark to work on the data compresses.
- Using the na.drop() function to remove rows with null values in the 'ArrDelay' column
- Using the na.fill() function to fill null values with 0 or an empty string
- Using groupBy() and agg() functions to group and aggregate data, which allows for more efficient processing of large datasets
- Using withColumn() and alias() functions to rename columns for better readability and understanding of the data
- Using join() function to join dataframe based on specific columns, which can be more efficient than using a traditional join
- Using sort() function to sort the dataframe based on specified column, which can be more efficient than using an external sorting algorithm

- Using write.csv() to save the dataframe as a csv file, which can be more efficient than using other file formats.

7.AWS



EMR Cluster Specs

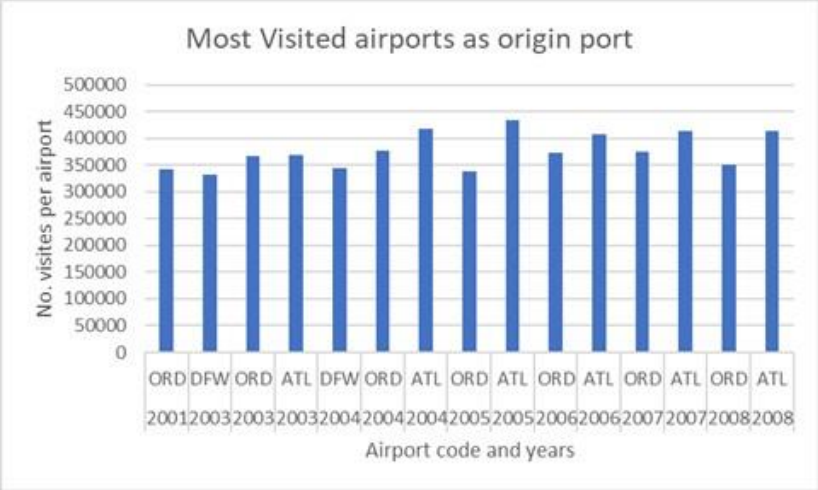


Sample of Spark Job

8.Results

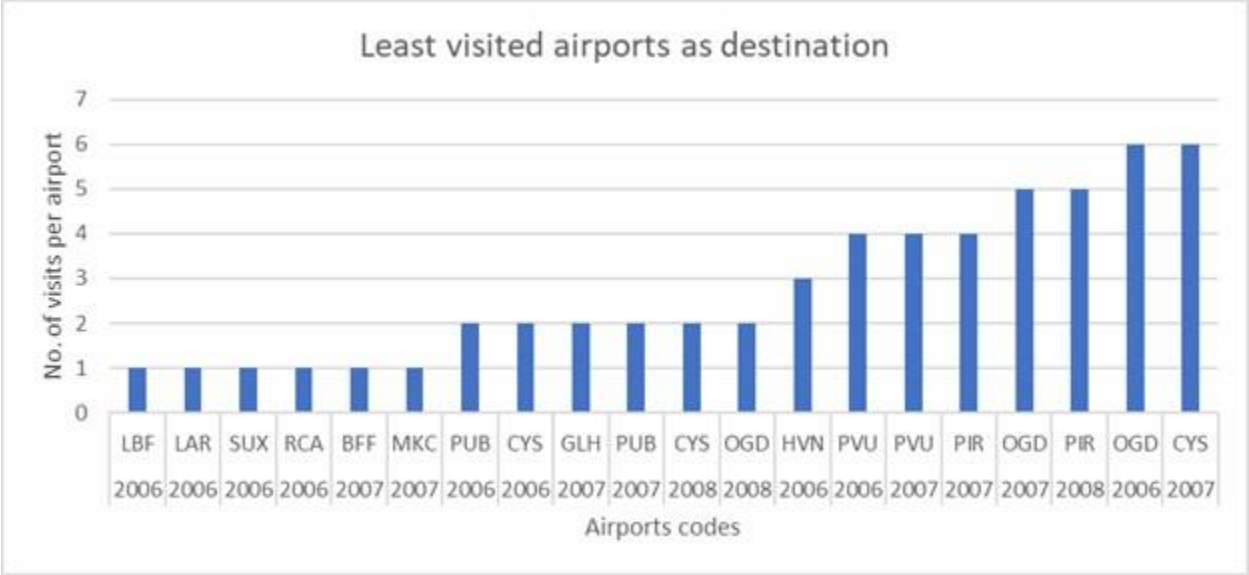
The results after applying previous procedures conclude within the following charts. They carefully analyze airlines dataset as follows:

1. Most visited airports being origin airports:

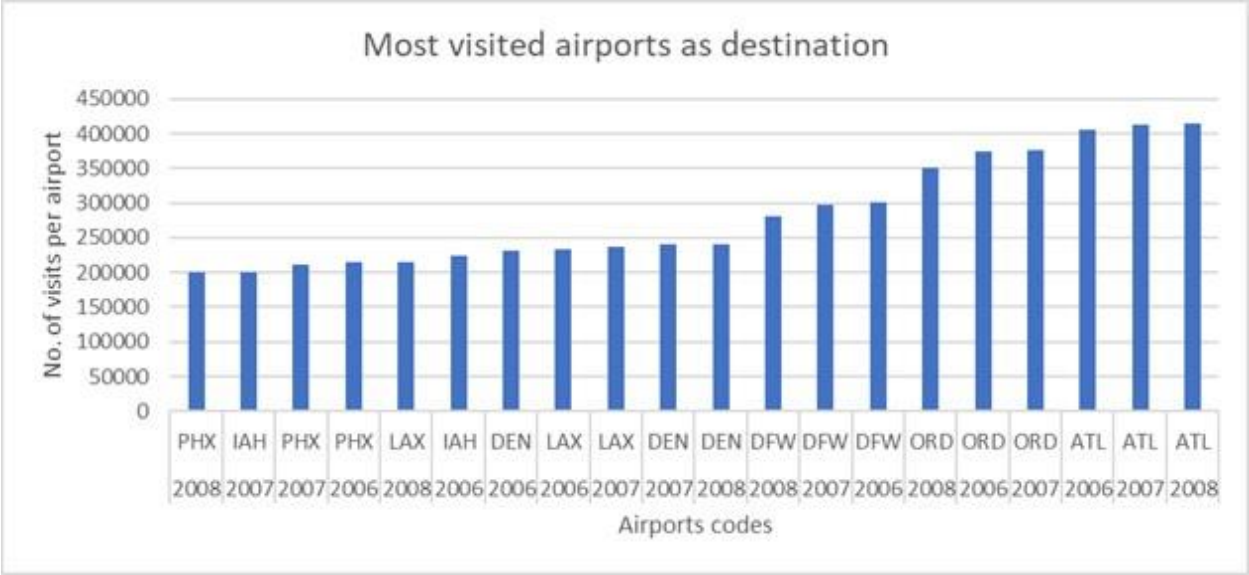


2. Least visited airports being origin airports in 2008:

3. Least visited airports being destination airports:



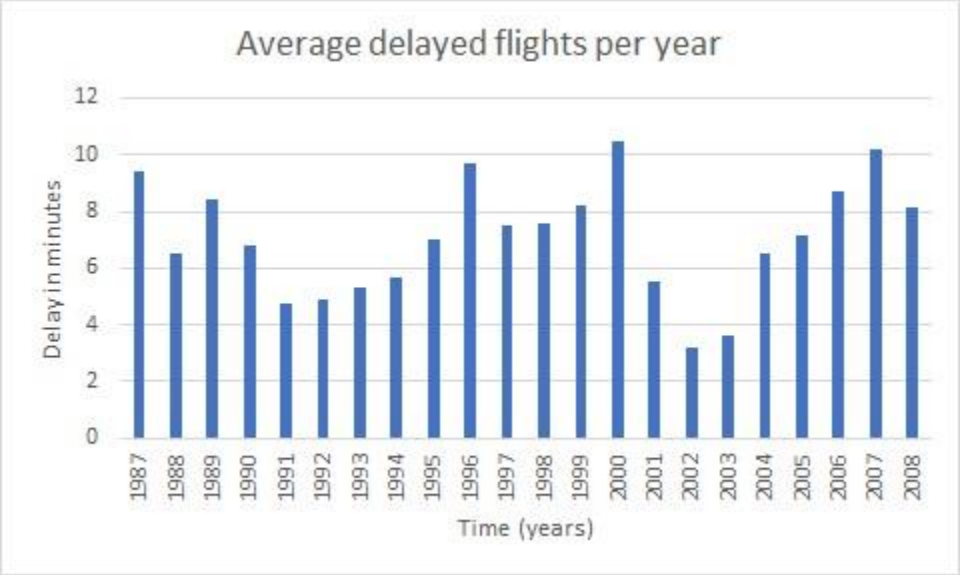
4. Most visited airports being destination airports:



5. Most delayed airports:

6. Least delayed airports:

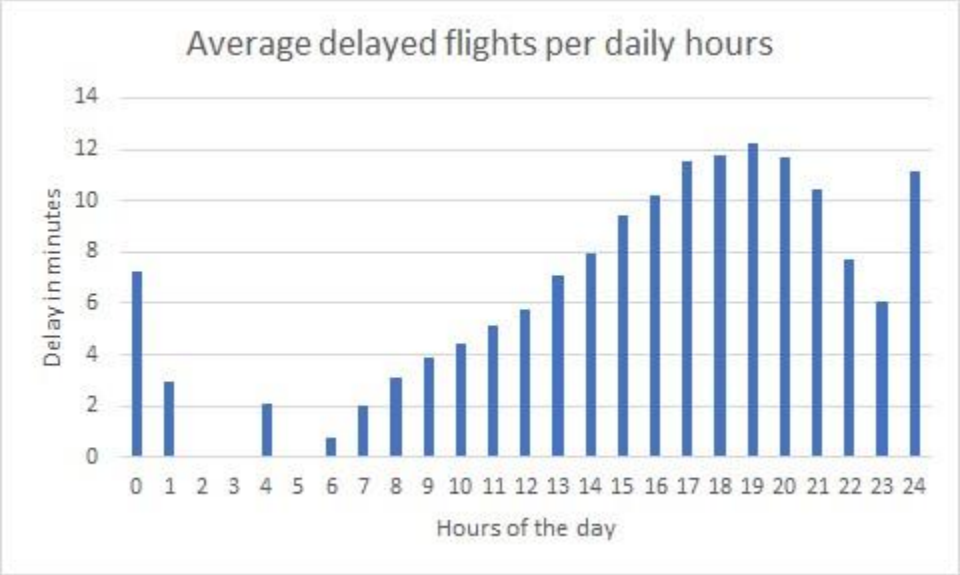
7. Average delayed flights per year:



8. Average delayed flights per month:

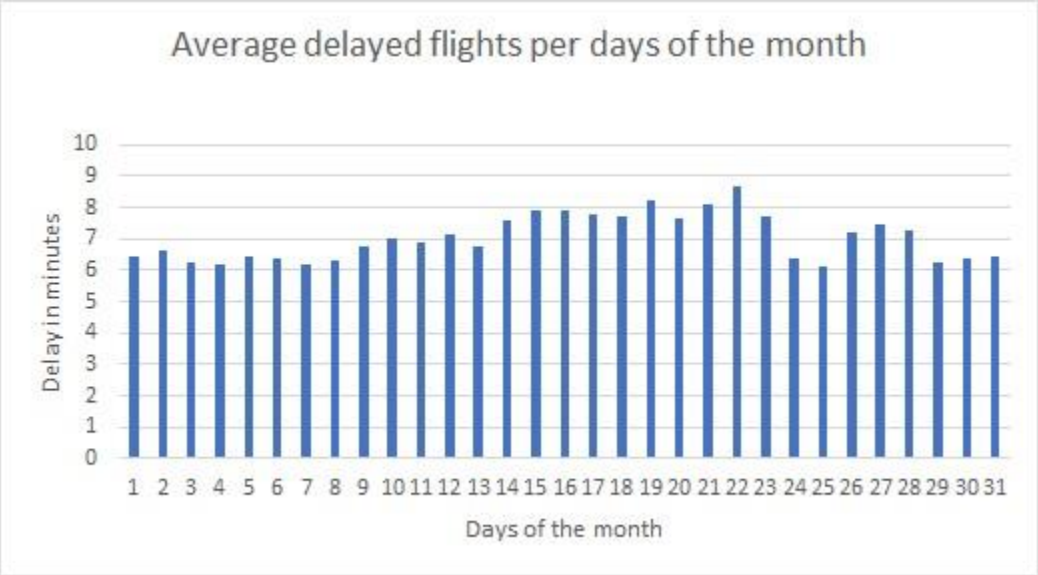
So the best Month to travel in is 9

9. Average delayed flights per hour:



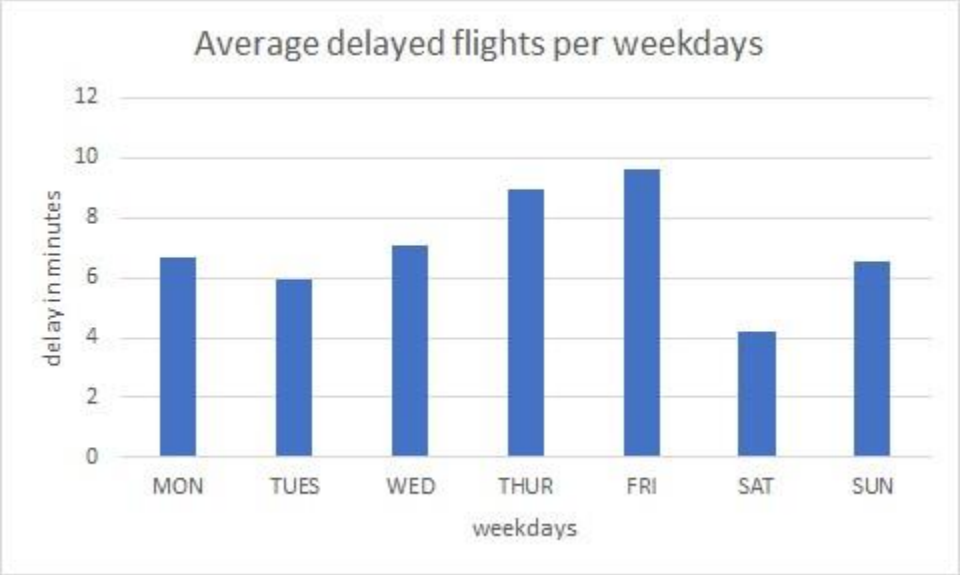
So the best hours to travel from 2 to 7

10. Average delayed flights per days of the month:



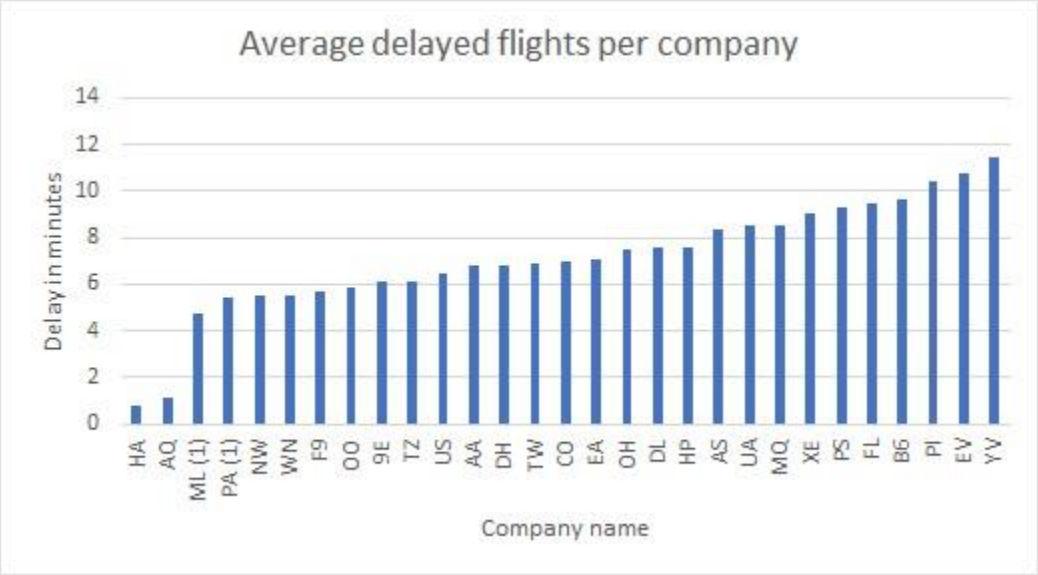
There is no big difference between the days on month

11. Average delayed flights per weekdays:



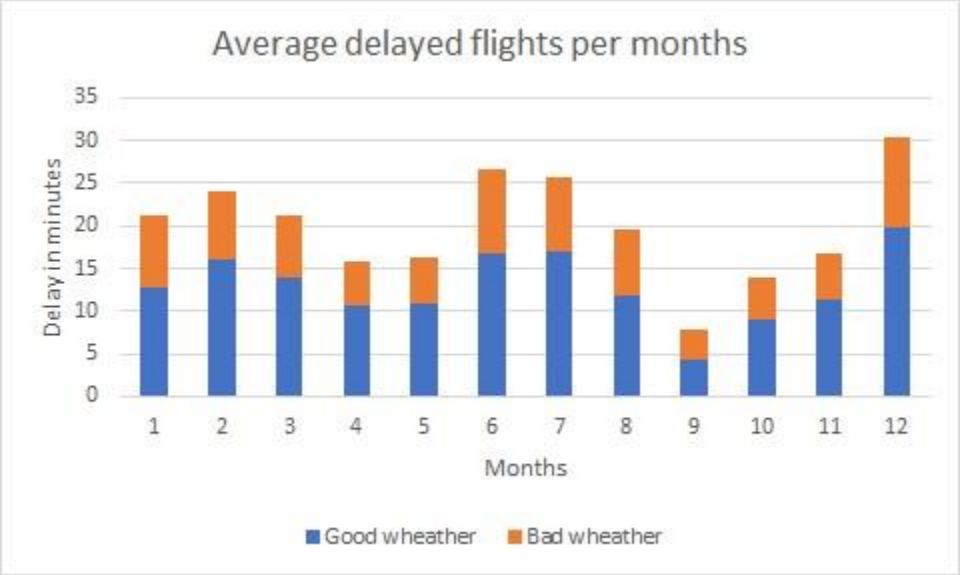
So the best day to travel in is Saturday

12. Average delayed flights per company:



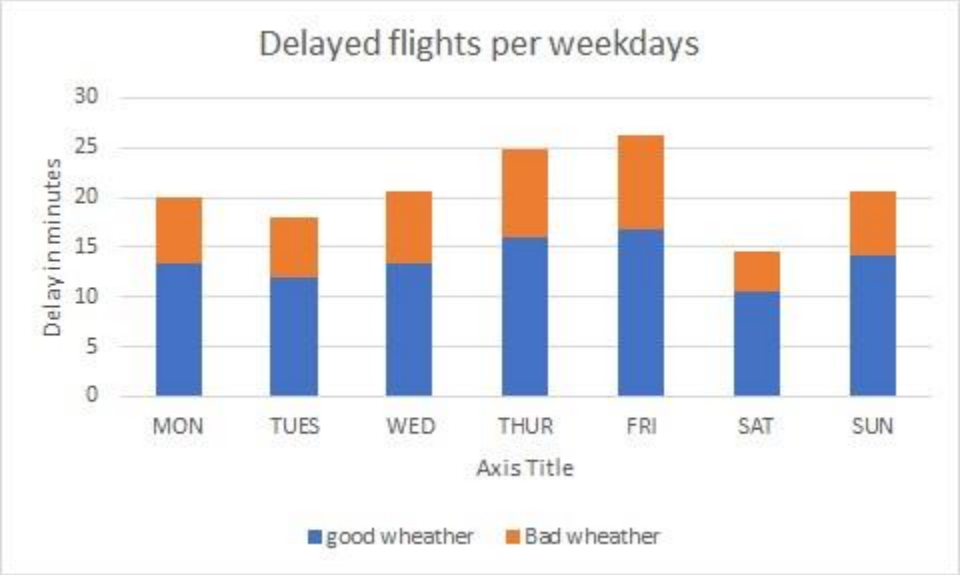
Hawaiian Airlines(HA) is the least company in delay time and Mesa Airlines(YV)

13. Average delayed flights per months due to bad weather:



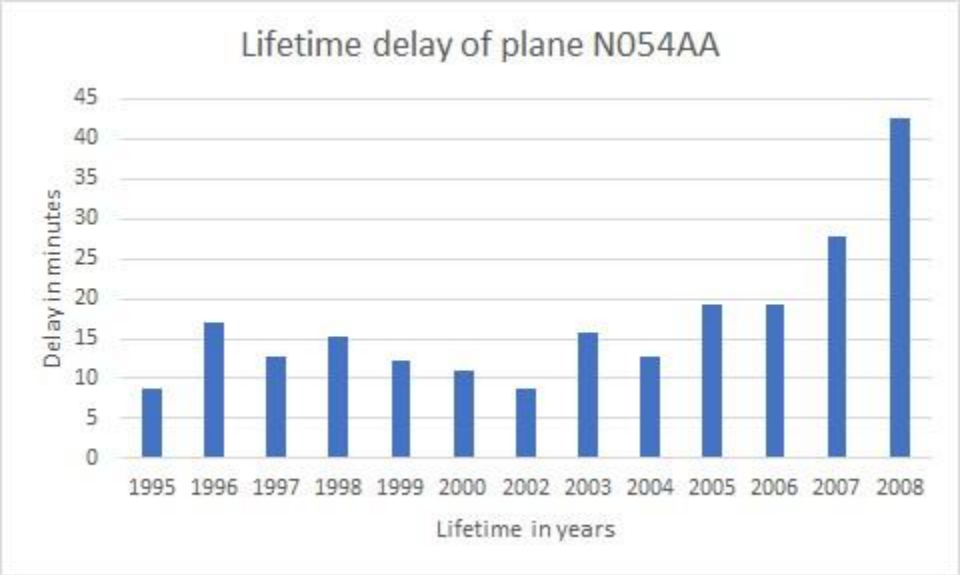
14. Average delayed flights per days of the month due to bad weather:

15. Average delayed flights per weekdays due to bad weather:

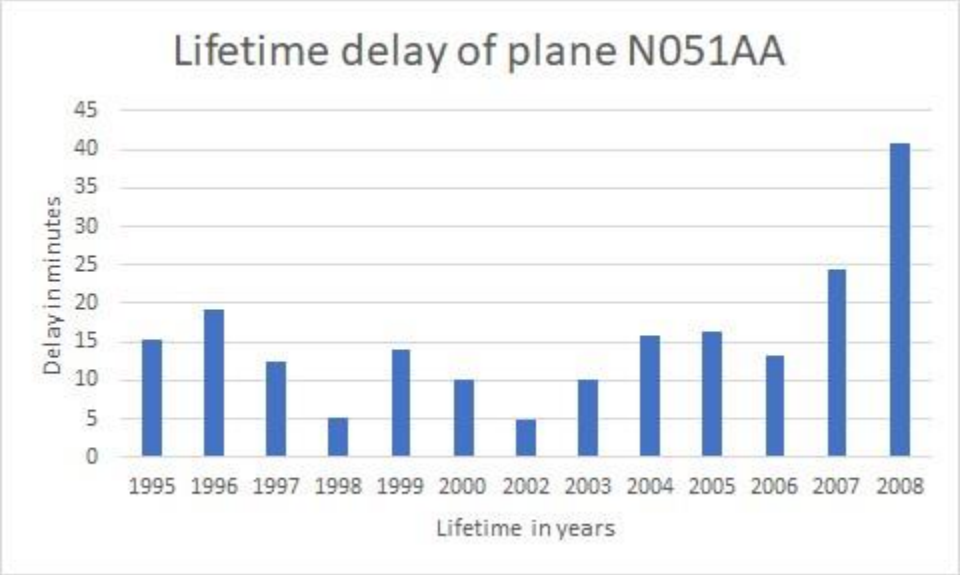


The Bad weather makes a huge difference in delay time.

16. Lifetime delay of plane “N054AA”



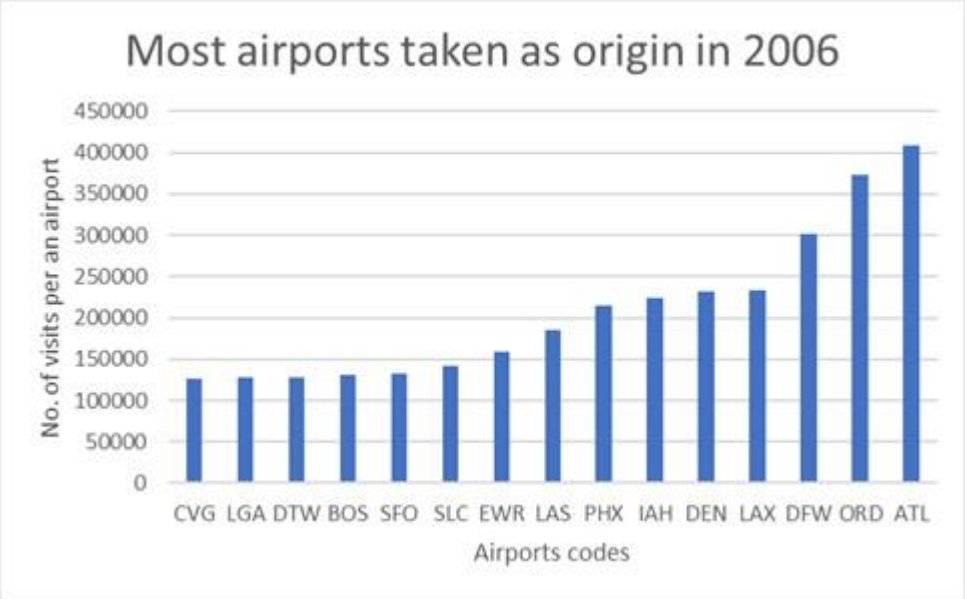
17. Lifetime delay of plane “N051AA”

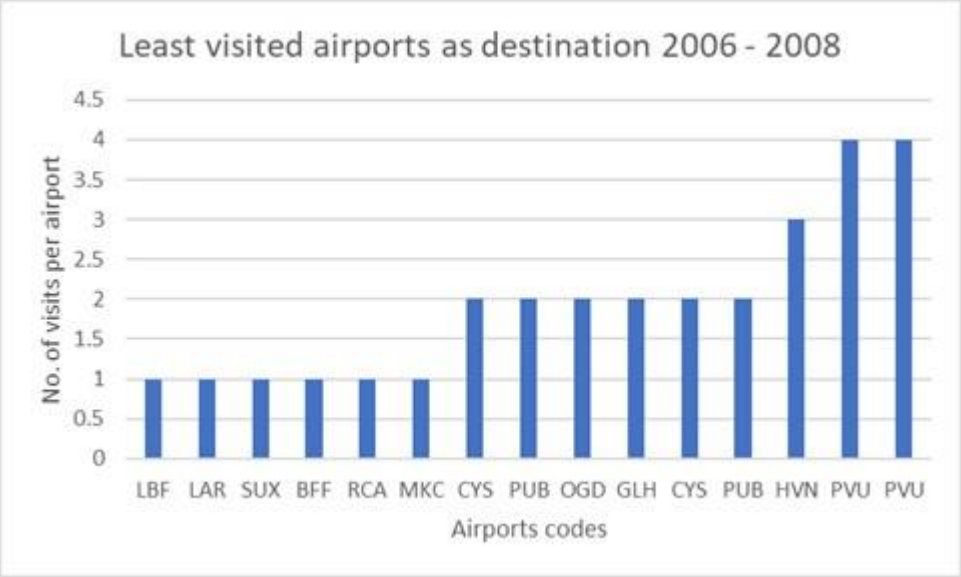


The Older the plane, the more the delay time.

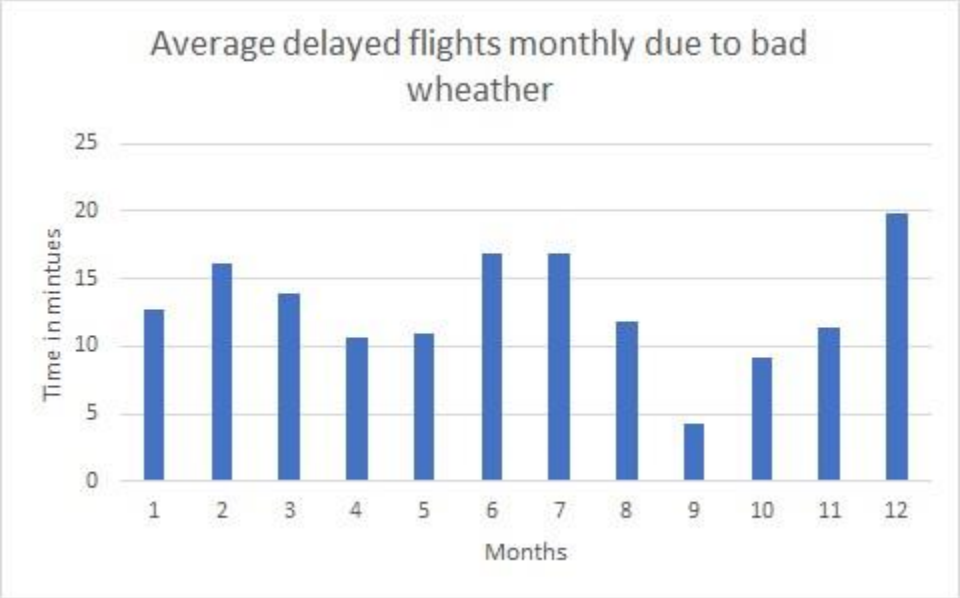
For further analysis of the data the following graphs demonstrate different relationships:

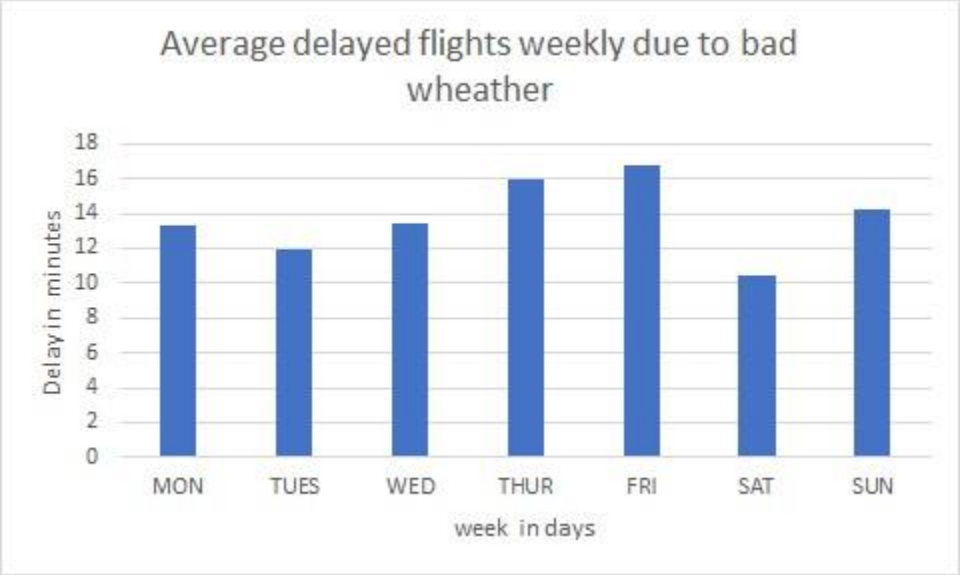

Most airports taken as origin in 2007
No. of visits per an airport
450000
400000
350000
300000
250000
200000
150000
100000
50000
0
BOS MCO SFO SLC EWR MSP DTW LAS IAH PHX LAX DEN DFW ORD ATL
Airports codes


Most airports taken as origin in 2006
No. of visits per an airport
450000
400000
350000
300000
250000
200000
150000
100000
50000
0
CVG LGA DTW BOS SFO SLC EWR LAS PHX IAH DEN LAX DFW ORD ATL
Airports codes


Least visited airports as destination 2006 - 2008
No. of visits per airport
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
LBF LAR SUX BFF RCA MKC CYS PUB OGD GLH CYS PUB HVN PVU PVU
Airports codes


Least visited airports as destination in 2008
No. of visits per airport
250
200
150
100
50
0
CYS OGD PIR TUP INL BJI GST ADK ACY AKN DLG BLI SUX ITH HTS LWB TEX RHI
Airports codes


Average delayed flights monthly due to bad wheather
Time in mintues
25
20
15
10
5
0
1
2
3
4
5
6
7
8
9
10
11
12
Months


Average delayed flights weekly due to bad wheather
Delay in minutes
18
16
14
12
10
8
6
4
2
0
MON
TUES
WED
THUR
FRI
SAT
SUN
week in days


Average delayed flights monthly
Delayed minutes
25
20
15
10
5
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Month in days