By : Lazoueche, Youssouf/Ngantung, Edmond Hendrik Jacob/Tian, Pei
Group : A3 M

Introduction

We are using the data (*Flights dataset.csv*) given from our Bishops Moodle at CS503 Data Visualization Assignment 3 and applied it to the one of Python compiler, *Jupyter*, for data processing. The data mainly describe the information about flights among the states of the United States of America in 2009.

In this assignment, we used such Python libraries mainly using the *plotly* to perform the data visualization into our code (*Assignment3.ipynb*) as a below

- pandas : to convert the data in the csv file into python
- numpy : to process the data with the math calculation
- plotly : to represent the output of interactive data visualization

This report shows the relationship between columns such as the relationship between air time and distance, the delays and US states following with their origins, etc following with the reason why we need to apply data cleaning as *NaN* values before analysing further to the data.

Preprocessing (Data Cleaning)

	YEAR	MONTH	DAY_OF_MONTH	DAY_OF_WEEK	UNIQUE_CARRIER	AIRLINE_ID	CARRIER	TAIL_NUM	FL_NUM	ORIGIN	...	CANCELLATION_CODE	DIVER
0	2009	12	2	3	9E	20363	9E	91879E	850	ATL	...	NaN	
1	2009	12	3	4	9E	20363	9E	92289E	850	ATL	...	NaN	
2	2009	12	4	5	9E	20363	9E	91629E	850	ATL	...	NaN	
3	2009	12	6	7	9E	20363	9E	91709E	850	ATL	...	NaN	
4	2009	12	7	1	9E	20363	9E	92289E	850	ATL	...	NaN	
5	2009	12	9	3	9E	20363	9E	92009E	850	ATL	...	NaN	
6	2009	12	10	4	9E	20363	9E	91539E	850	ATL	...	NaN	
7	2009	12	11	5	9E	20363	9E	92289E	850	ATL	...	NaN	
8	2009	12	13	7	9E	20363	9E	91629E	850	ATL	...	NaN	
9	2009	12	14	1	9E	20363	9E	91869E	850	ATL	...	NaN	

Data cleaning has to be done since some certain columns have NaN values which can impact the data analysis to be inaccurate. We then located and counted the number of NaN values between columns by using the python command of *data.isna().sum()* such as below

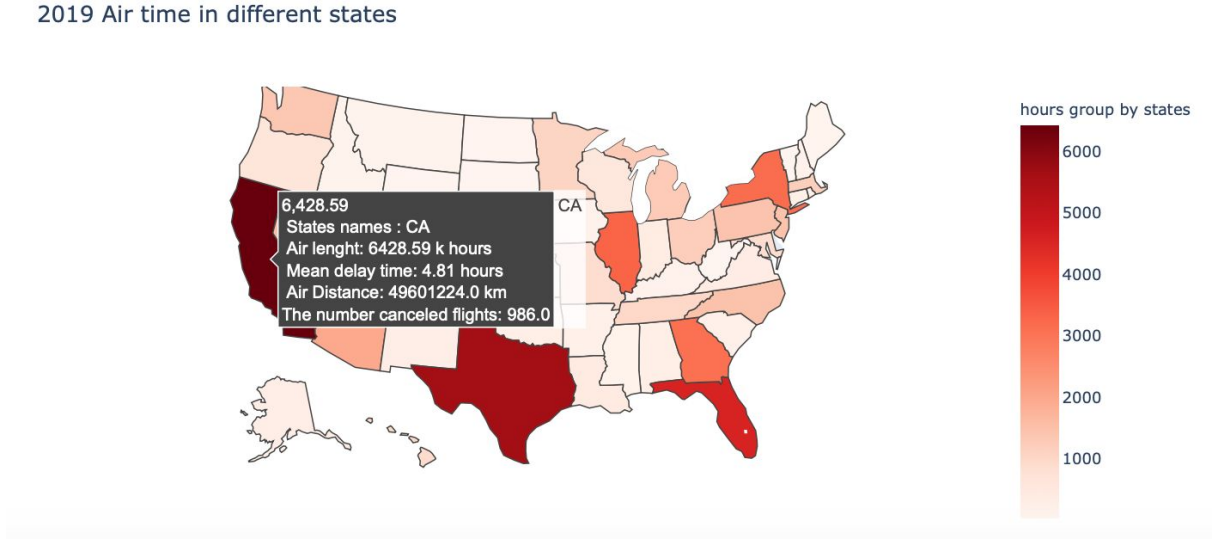
```
In [4]: data.isna().sum()
DEST_STATE_NM              0
DEST_WAC                   0
CRS_DEP_TIME               0
DEP_TIME               14193
DEP_DELAY              14193
CRS_ARR_TIME               0
ARR_TIME               15178
ARR_DELAY              16218
CANCELLED                  0
CANCELLATION_CODE     514539
DIVERTED                   0
AIR_TIME               16218
DISTANCE                   0
CARRIER_DELAY         397249
WEATHER_DELAY         397249
NAS_DELAY             397249
SECURITY_DELAY        397249
LATE_AIRCRAFT_DELAY   397249
Unnamed: 35           529269
dtype: int64
```

In this case, we only delete the *CANCELLATION_CODE* and *Unnamed: 35* as both have the highest number of NaN values in between columns.

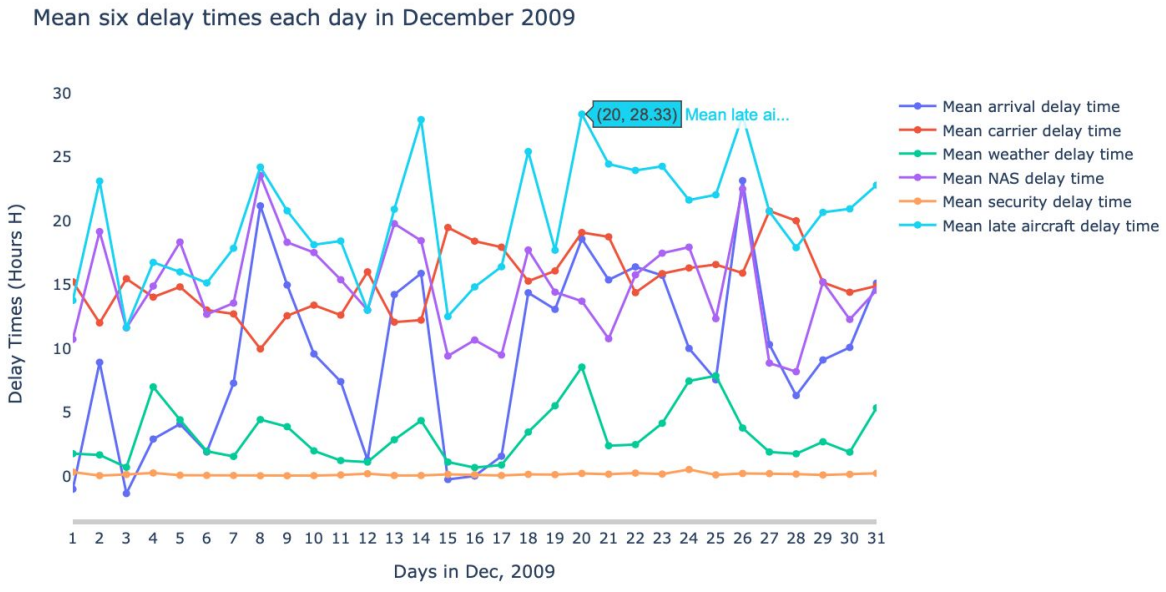
By : Lazoueche, Youssouf/Ngantung, Edmond Hendrik Jacob/Tian, Pei
Group : A3 M

Data Analysis

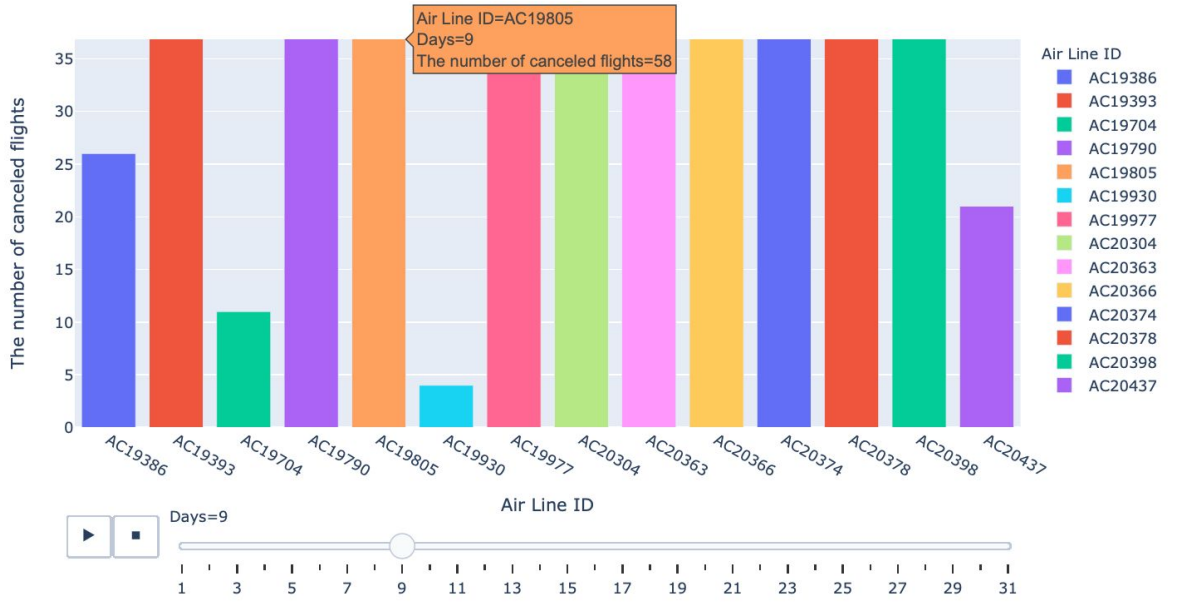
The first data analysis is to discuss the relationship between US states and the air time. In this analysis, it specifically observes the comparison of the air time in between the states. Thanks to plotly library in python, we can not only easily compare the air time hours based on the color, but also to see further the detail of the states we are pointing to. As you can see, California is at the very top using flights in the US as it follows with other datas supported including mean delay time, air distance and the number canceled flights. All the queries are set in the python code.



The second graph shows the different types of delay time in December 2009. The mean security security delay time is mostly constant compared to other graphs. Those other graphs are always fluctuating in December 2009. As you can see below, the flight departure (aircraft) has the biggest number in delay time in the month. We can easily see the detail of the coordinate to the highest point (20, 28.33) which belongs to the aircraft.



The third relationship is between the number of the cancelled flights and the airline code. This graph actually needs to be represented in 31 different types of graphs as the bar graph goes to different when it goes day by day. In this representation, we use plotly to compress multiple graphs into only one graph. This can be done by using animation. It is much more interactive and tidy than the other previous graphs as they have the animation located at the below of the bar graph. We took one specific case on day 9th, mostly all flights are cancelled.



CONCLUSION

- It is important to do data cleaning before analysing as the original data has too many NaN values which can be affecting the accuracy of the analysis.

- Plotly output could be the best to represent the analysis in the very compact and dynamic way as it can reduce many redundant figures and show the significance into the desired details shown as coordinates..