Rishabh Khorana

Rishabhkhorana@yahoo.co.in

Airline On-Time Performance and Causes of Flight Delays

Contents

[INTRODUCTION 1](#_Toc88089967)

[Python script for file creation and visualization. 2](#_Toc88089968)

[Python Pandas module to retrieve specific information. 8](#_Toc88089969)

[CONCLUSION 10](#_Toc88089970)

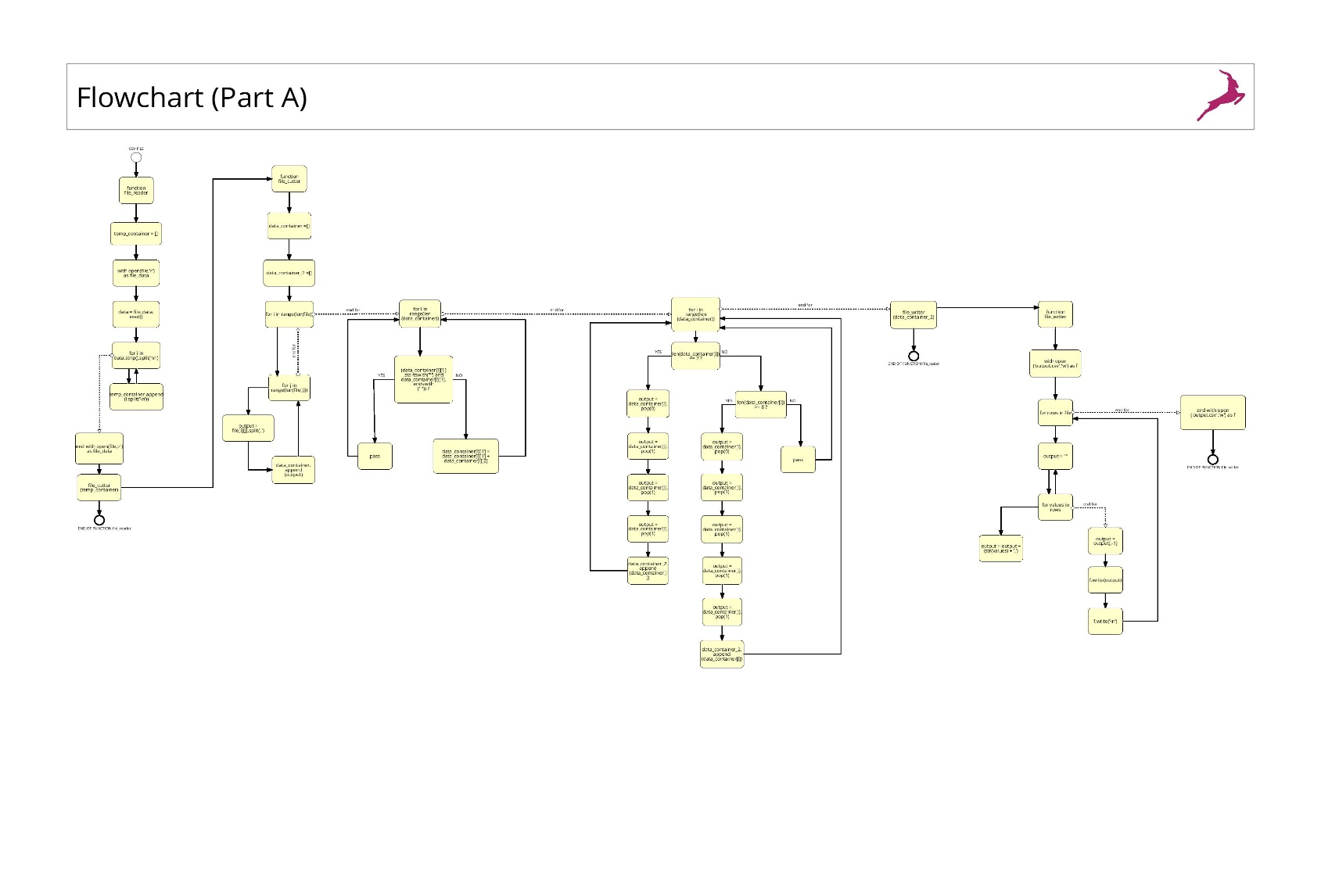
[APPENDIX 12](#_Toc88089971)

# INTRODUCTION

This main purpose of this report is to discuss pertaining to the procedure involved for the completion of the two exercises which involves utilization of the Python programming language. These exercises involve modules which include importing raw data into the programming environment, basic data wrangling which includes cleaning & formatting data and finally data visualization that helps to efficiently analyse the results enabling us to communicate with the stakeholders and provide insights based on the acquired results. This report will provide guidance to reader regarding the steps involved to accomplish the specific tasks, python script required to perform the task and representing the results through appropriate diagrams.

# Python script for file creation and visualization.

The first task in this assignment comprises of three parts which involve utilizing the provided csv file comprising of data relating to the airports and create a text file including only the names of the airports & their coordinates which involves automation of data loading into an IDE & file creation. The first task involves creating a flowchart which represents the procedure involved for undertaking the above mentioned task. Secondly, we must provide the python script for csv file and creating the text file with the requested information. The final part pertains to data visualization where we are required to plot the coordinates of the airports using a scatter diagram with only the negative longitudinal coordinate values. The flowchart for performing the above mentioned task is as follows:



**Fig 1: Flowchart [Part-A]**

Focusing on the python script for loading the csv file & creating the text file, we have divided the program into four smaller functions which simplifies the task of debugging the program and maintaining the flow.

First we import the OS module where we use the change directory method to specify the newer directory where the required files are stored. Following this, we call the first user defined function “file\_reader” with the csv file given as a parameter. Focusing on the “file\_reader” function, we access the csv file using the with function with the access mode specified as ‘r’ for reading the file. We then use the read() function which reads the complete data as a single string. Following this, we use the for loop to iterate the data stored in file, within which we use the strip() function is used to eliminate the empty spaces & the split function acts as the delimiter to split the data on the specified character . We then use the second split function to convert the data into two dimensional array which is then appended to the list declared within the function. Finally, we call the second user defined function (file\_cutter) with the list given as a parameter.

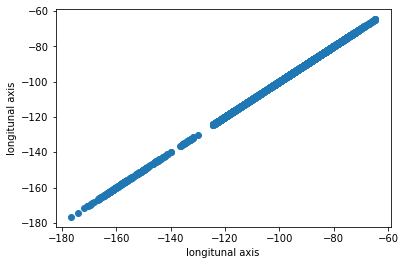
In the file\_cutter() function, we first declare two lists. Following this, we use a nested for loop which iterates over the length of the output from file\_reader() function. Here, we are trying to access the lists located within the list. Then, we use another for loop with the range given as the length of the data stored in the first list. We then use conditional statement within the list, where ‘i’ holds the values of the index which tells us the position of the rows. Here, we are checking whether the no. of elements inside the list data\_container is seven. We then use the pop function to eliminate the unnecessary columns, this would move the columns and change their index every time we pop a column from the list data\_container. Finally, this data is appended to the second list declared within the function.

During the initial slicing of the data on the delimiter’s specifically comma(‘ ,’) , few rows had more than 7 elements . This would append the values of latitude and longitude outside their respective columns. To tackle this issue, we use an additional Boolean statement. Finally, we append the obtained results to the list data\_container\_2. We then call the third user defined function “file\_writer()” with the list “data\_container\_2” as the parameter.

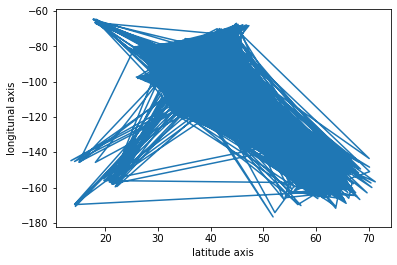
Focusing on the file\_writer() function, we create the text file using the with function and specify the access mode as ‘w’ for writing the data. Then we have a nested loop to access the values in each row which stored as a 2 dimensional list. Since strings are immutable , we create an empty string which would be iterated the same number of times as the rows in the data and the second loop adds the values for each row at that particular index to the variable output as string . It also adds a comma at the end of each element added to the string. When the row is finished, it would add comma after the last element. For removing this, we slice the last element at index -1. Then “f.write” writes the values to the rows, inserts a newline ‘/n’ and moves to the next iteration.

Following this, we move to the final part of the program which involves data visualization using a scatter plot diagram. For undertaking this task , we define the last user-defined function “scatterplot()” with the newly created text file as the input. Focusing on the scatterplot() function, we first import all the modules from the Pandas library and the Matplotlib which is a basic visualization package and plotting library in Python where we use the Pyplot module to automatically create figure objects and call the output file. Following this, we first filter those longitudinal values that are less than zero as per the requirement and convert them from strings to float values using the map() function. Then , we create two variables (x & y) to store the latitude values in x axis & longitudinal values in y axis.

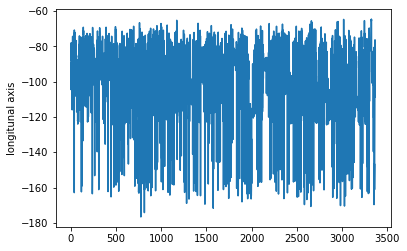
We then move towards the creation of scatter plot diagrams. In this scenario, we create five scatter plot diagrams where the first diagram represents the negative longitudinal values on both axes creating a plot y using x axis 0 ….. n-1 . For the second plot diagram, we represent the negative longitudinal values with latitudes for airport. In the first two diagrams, we use the plot function where a change in the attributes would be applied to all the elements. Following this, for the third diagram we use the scatter function to plot longitudinal values in both axis where we should get a linear relationship. For the last two diagrams, we use the scatter function to represent both the latitude and longitude values interchanging the positions of x and y axis.



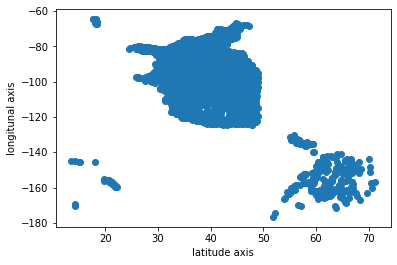
***Fig 2: Plot between negative longitudinal data on both axis***



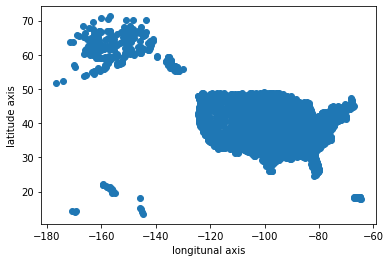
***Fig 3: Plot with negative longitudinal values & latitudes for the airport***



***Fig 4: Plot with only negative longitudinal values***



***Fig 5: Scatter plot with latitude values as x axis & longitude values as y axis.***



***Fig 6: Scatter plot with axis interchanged.***

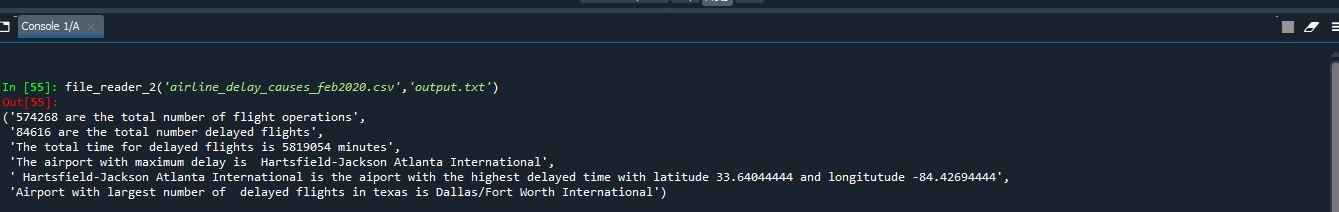
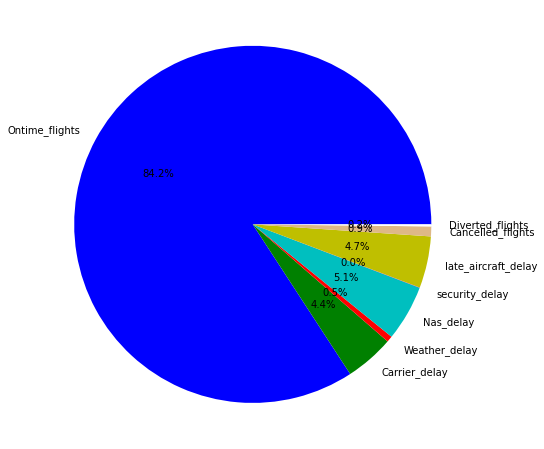
**[ For detailed breakdown of the python script to achieve the above tasks, please refer to the Appendix. ]**

# Python Pandas module to retrieve specific information.

In this second part of the exercise, we use the Python pandas module to retrieve specific information. We utilize the second csv file comprising of information pertaining to airline delays. This csv file comprises of 21 fields comprising of qualitative & quantitative information. For the qualitative information, we have data pertaining to carrier designator & name and airport details. For the quantitative information, we have information pertaining to no. of arrivals, delayed flights, no. of delays due to problems involving career issue, weather issue, national air system issue, security issue. We also have information regarding cancelled & diverted flights and delayed minutes due to the above mentioned issues. This activity comprises of seven sub-tasks pertaining to information retrieval which involves the following.

* Find the total number of flights (operations).
* Find the total number of delayed flights.
* Find the total delayed time in minutes.
* Find the airport with largest number of delayed flights.
* Find the coordinates (from Part A) of the airport with highest delayed time.
* Find the airport in Texas which has the largest number of delayed flights.
* Using a pie chart present a display the percentage of on-time flights, and the items listed in the table above.

The results for the above mentioned activities are as follows:



**Figure 1: Results dashboard (Q1-Q6)**

**Figure 2: Pie chart (Q7)**

**[ For detailed breakdown of the python script to achieve the above tasks, please refer to the Appendix. ]**

# CONCLUSION

The first part of this report involved utilizing the python CSV module to sort out the data provided in the airport CSV file. The primary task that we conducted was making a flow chart to convert the details on the airport CSV file to create a text file to include the airport names and their coordinates. After making the flow chart, we then focused on developing a python script to load the details in the airport csv file and convert it into a text file. We were successful in converting the file by focusing on three functions which are file\_reader, file\_cutter, and file\_writer.

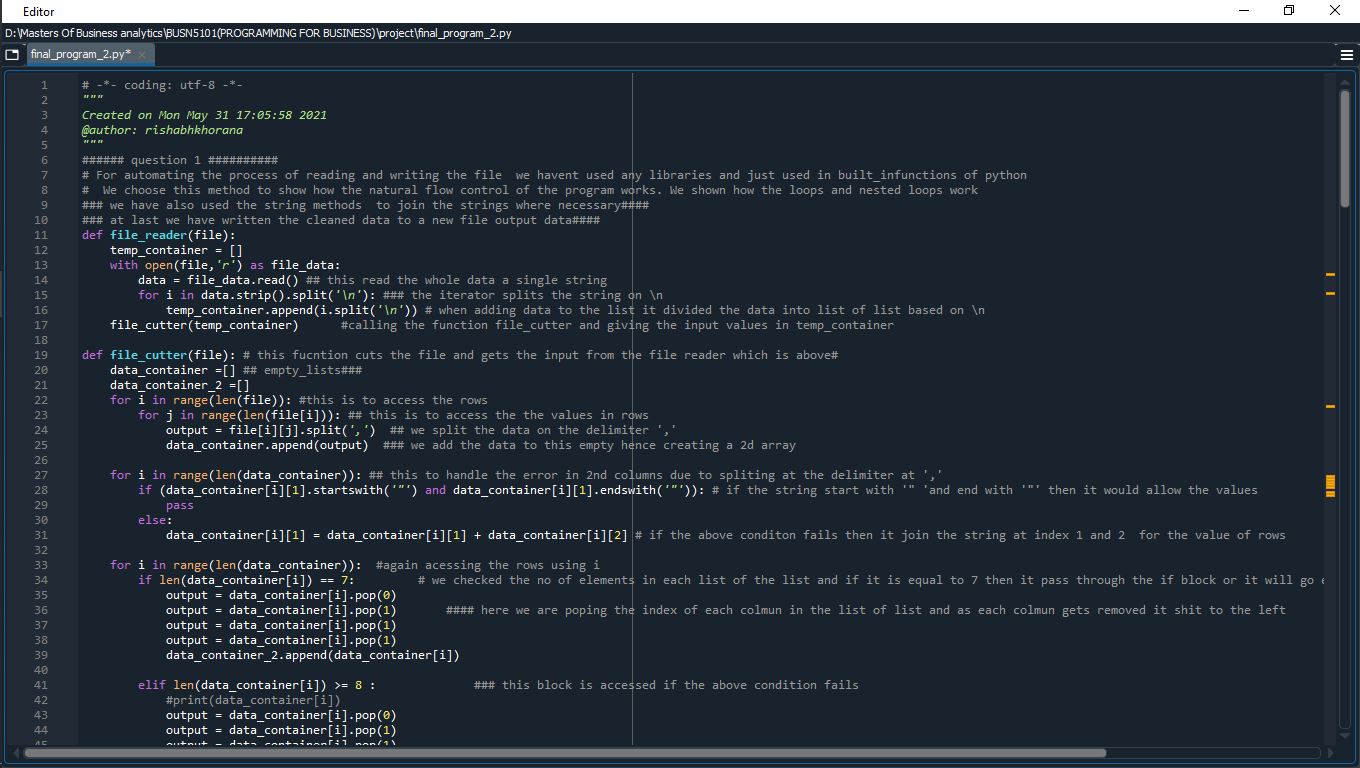
After the data was converted into a text file, we then utilized the scatterplot function to visualize the various coordinates of the airports with only the negative longitudinal coordinate. We imported all the modules from the Pandas library and the Matplotlib, a basic visualization package and plotting library in Python, where we automatically use the Pyplot module to create figure objects and call the output file. We generated five scatter plot diagrams focusing on the negative longitudinal values on both axes creating a plot y using x-axis 0…. n-1, negative longitudinal values with latitudes for the airport, longitudinal values in both the axis where we should get a linear relationship and representation of both the latitude and longitude values interchanging the x and y-axis positions, respectively.

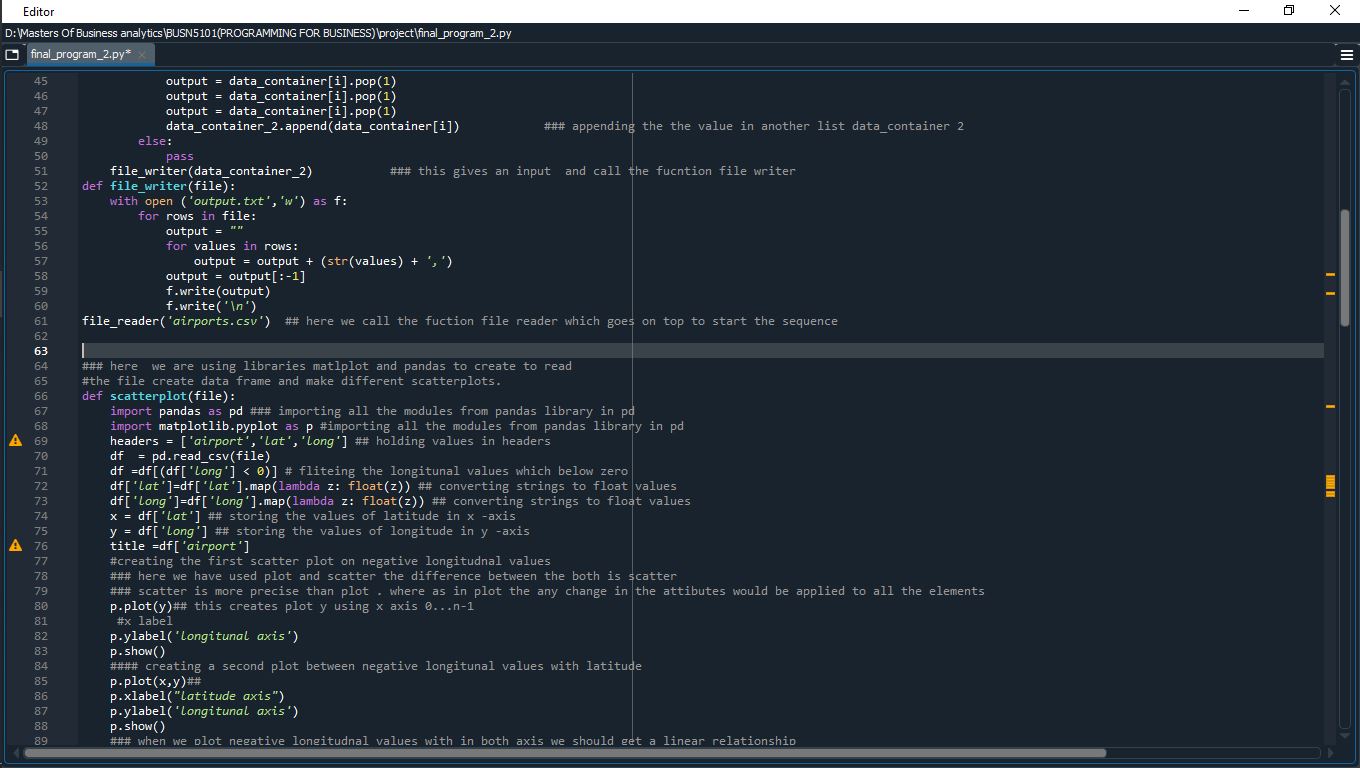
In the second part of this report, we used the python panda module to retrieve specific information. For this part of this project, we utilized the second CSV sheet provided to us, which contained qualitative & quantitative data to develop scripts to answer the questions in the assignment and present all the answers and time flights in a pie chart. We obtained the following results :

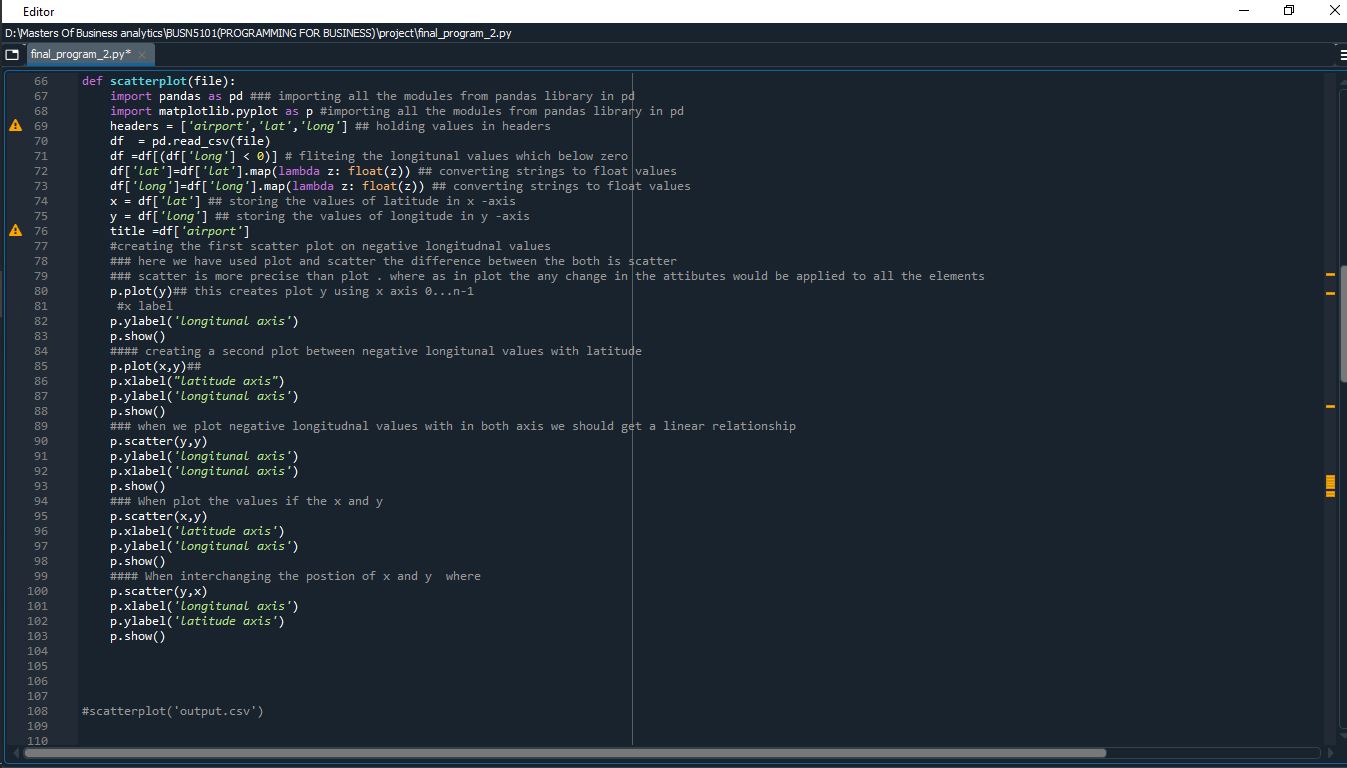
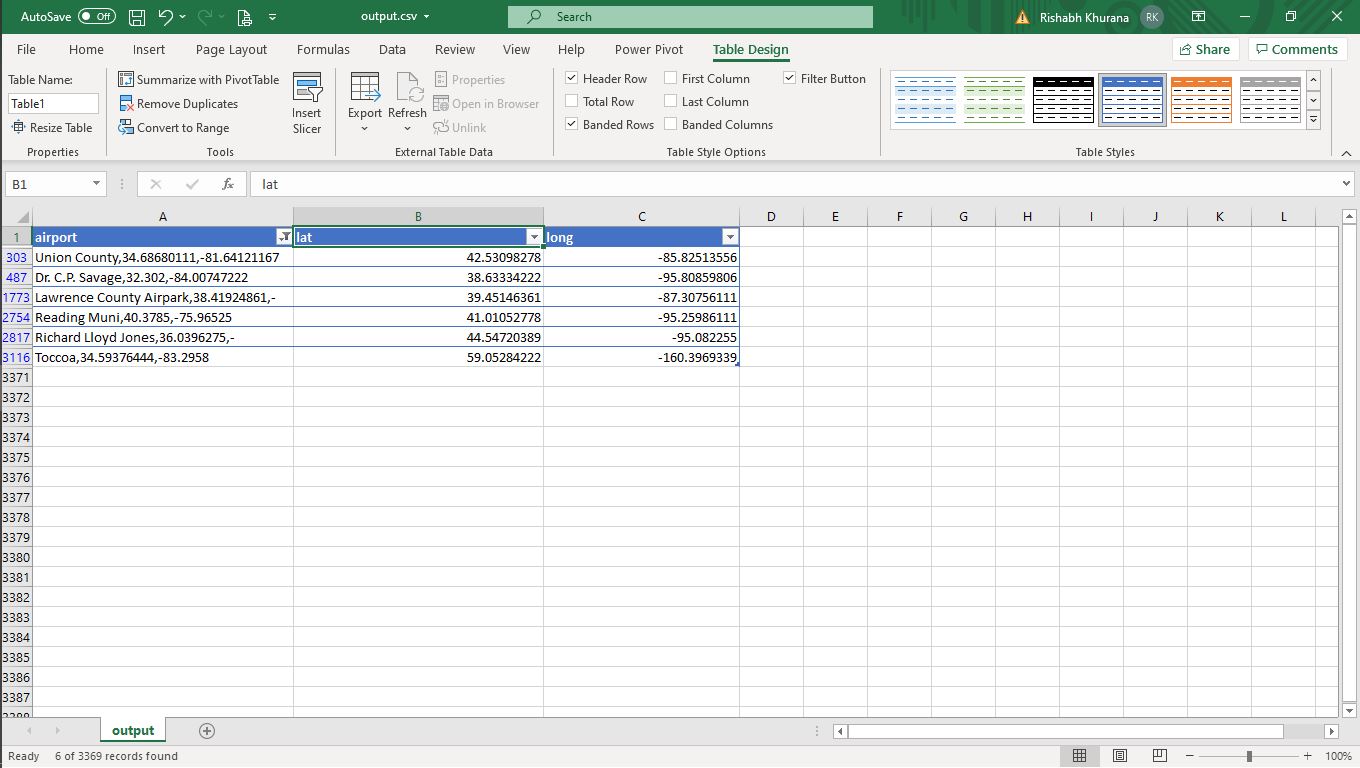
* The total number of flights (operations): 483460
* The total number of delayed flights: 84616
* The total delayed time in minutes: 5819054 minutes
* The airport with the largest number of delayed flights: Hartsfield – Jackson Atlanta International.
* The coordinates of the airport with the highest delayed time: latitude 33.6404444, longitude -84.42694.
* The airport in Texas which has the largest number of delayed flights: Dallas / Fort Worth International.

Hence to conclude, we can state that this case study helped us understand the real-world scenario. We utilized our programming skills to solve real-world scenarios and business-related problems and visualize data to display results effectively.

# APPENDIX

* Python script (Part A)

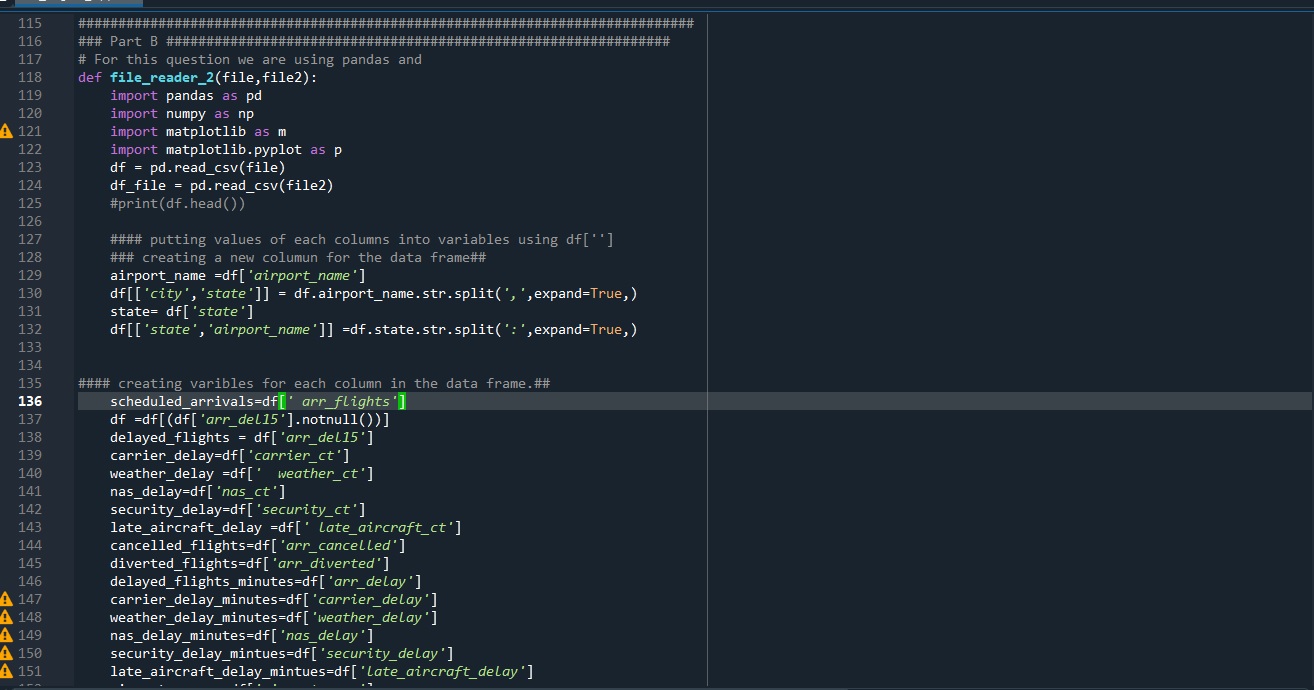


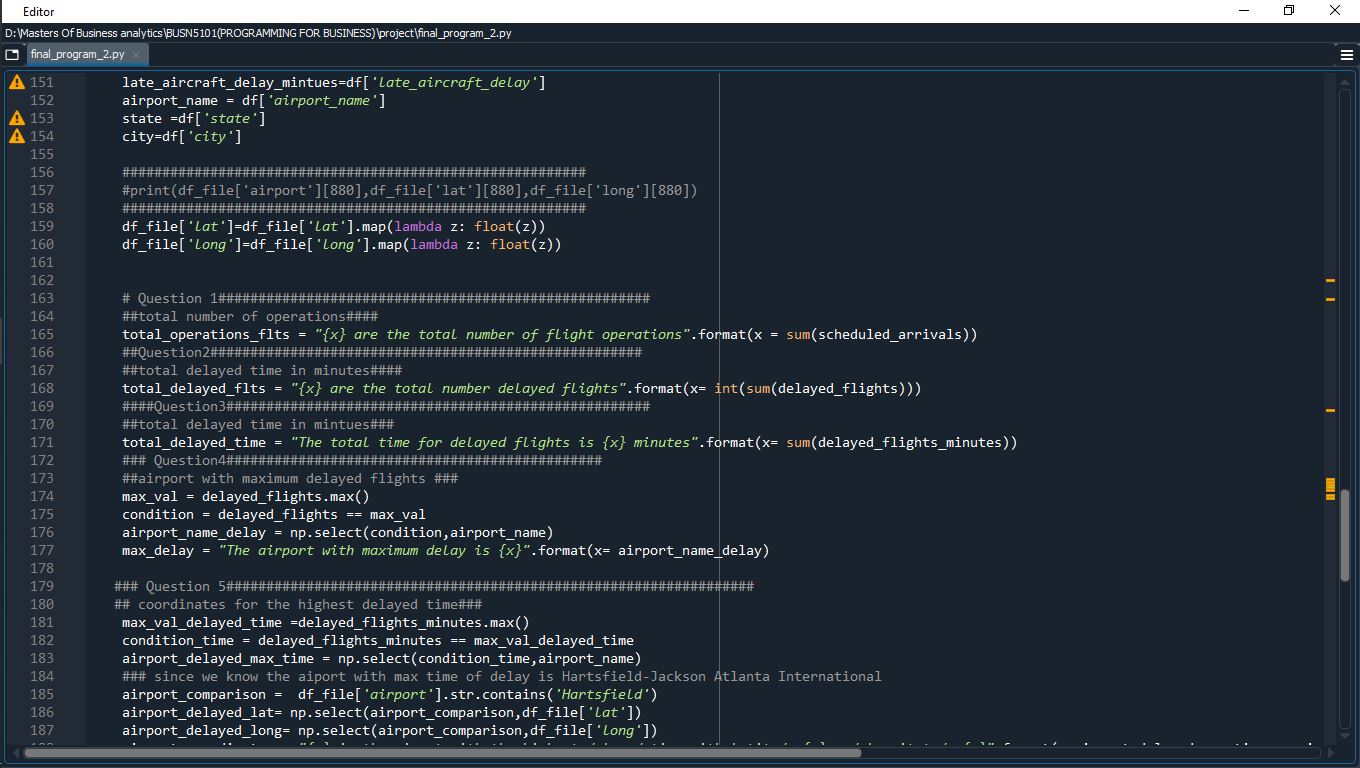


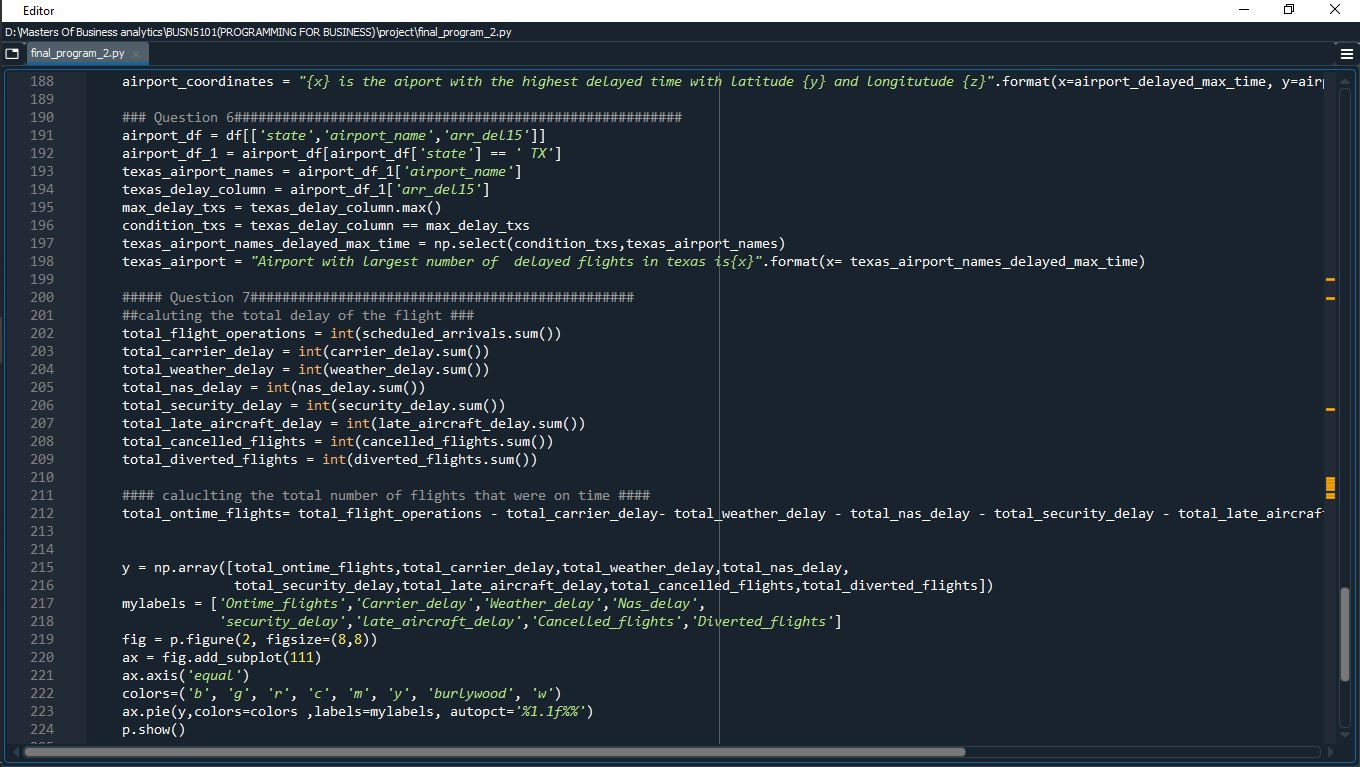
**Appendix: Error Occurrence (delimiter)**

This was handled in the final output.

* Python script (Part B)







# References:

Transtats.bts.gov. 2021. *OST\_R | BTS | Title from h2*. [online] Available at: <https://www.transtats.bts.gov/OT\_Delay/OT\_DelayCause1.asp> [Accessed 17 November 2021].