PYTHON PROGRAMMING ASSIGNMENT

BUSN5101

Aditya Singh (23175095)
Adrian Adrianto (23220175)
Chaitanya Chawla (23030293)
Sanchit Bajaj (23245744)

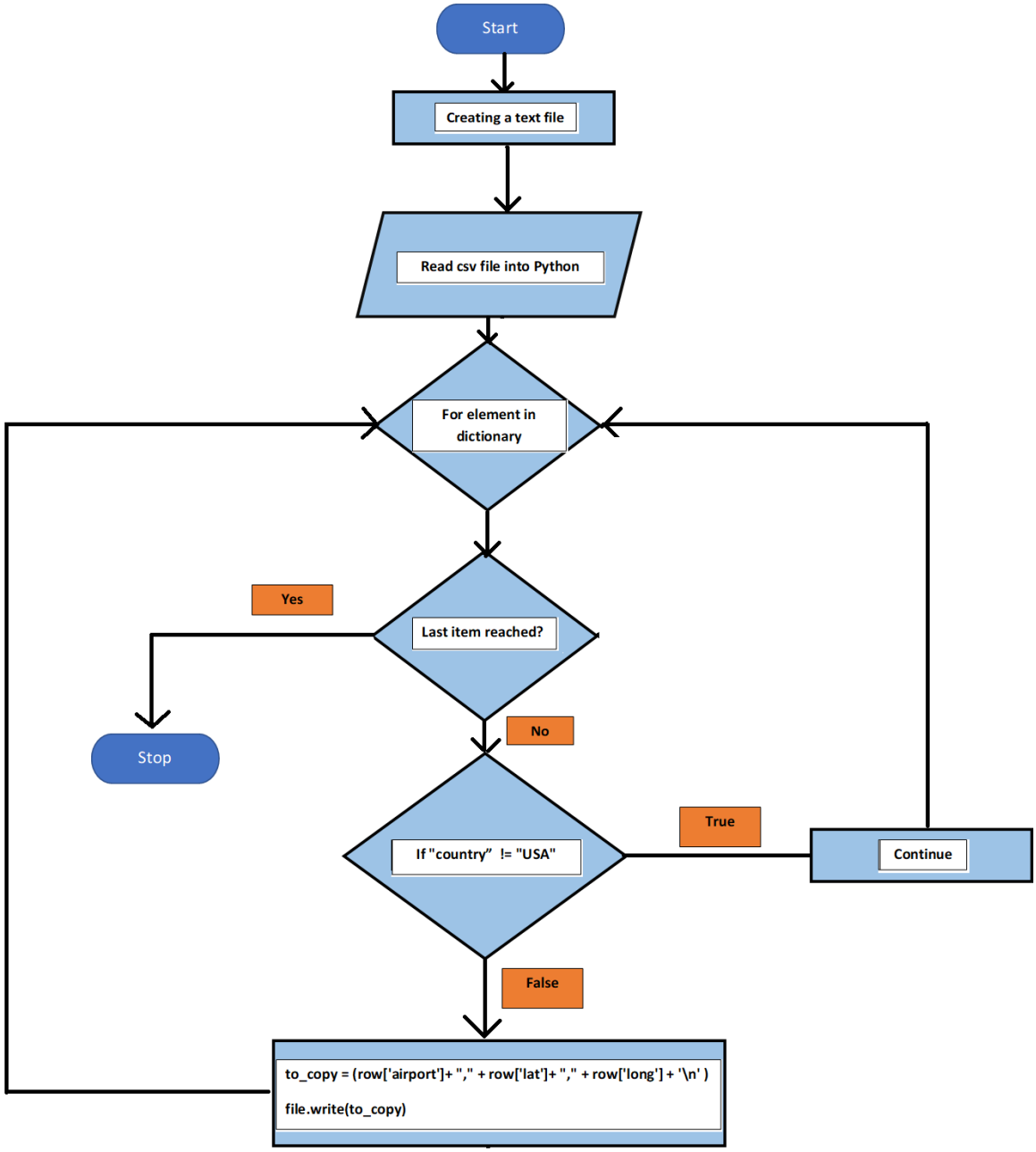
PART – A

Introduction

For this part, we are requested to import data from a file into the IDE. Data is imported from csv files and contains general information about airports in the USA, such as, name, city, latitude, and longitude. It is also requested to specifically write information of the airport name, and coordinates (latitude, and longitude), into a new text file.

Additionally, a diagram containing negative longitudinal coordinate values, is requested. This diagram would be presented in a scatter diagram to help visualize the request better.

Below is a flow chart which shows the process of creating the text file with the requested information.



Results

For the requests, Python **CSV module** is used to import data from the given file. The requests can be broken down into four parts.

- The first one is to import data from the given file into IDE.
- The second one is to write the names, latitudes, and longitudes of each airport into a new text file.
- Finally, a diagram which contains negative longitudinal values is requested for visualization purpose.

Inspecting the Data

- The data is imported using the CSV module, and into a python dictionary using the "**.DictReader**"
- Data is appended into a list to check for null values and other information.
- Checking for missing values using the string method. We were able to deduce that there are no missing values under country, longitude and latitude, country.
- We noticed there were 4 places other than the US. – Thailand, Palau, N Marina Islands, Federated States of Micronesia.

Code for importing the data to check for null values/empty cells and other information, such as finding the countries other than the US.

```
import csv
import os
os.chdir(r'C:\Users\shaul\OneDrive\Desktop\Programming for Business\Group assignment')

with open('airports.csv') as csvfile:
    file = csv.DictReader(csvfile)
    country   = []
    longitude = []
    latitude  = []
    airport   = []
    for col in file:
        country.append (col['country'])
        longitude.append (col['long'])
        latitude.append (col['lat'])
        airport.append (col['airport'])

print(len(country))
print('-'*100)

count = 0
for elm in country:
    if elm != "USA":
        count += 1
        print(elm)
print("There are", count, "rows which have values other than USA.")
print('-'*100)

for elm in country:
    if not elm.strip():
        print('Yes, there are missing values.')
```

```
3376
----------------------------------------------------------------------------------------------------
Thailand
Palau
N Mariana Islands
Federated States of Micronesia
There are 4 rows which have values other than USA.
----------------------------------------------------------------------------------------------------
```

Creating the text file

- Requested information for the text file –
  - **Airport Names** and **Coordinates** – Latitude and Longitude
- We open a new text file- “airports.txt” in the working directory.
- Read the data again with “csv.DictReader”
- We initiate a condition to filter out information for countries other than the US.
- We store the requested information in an object (to_copy)
- Write that information into the text file.

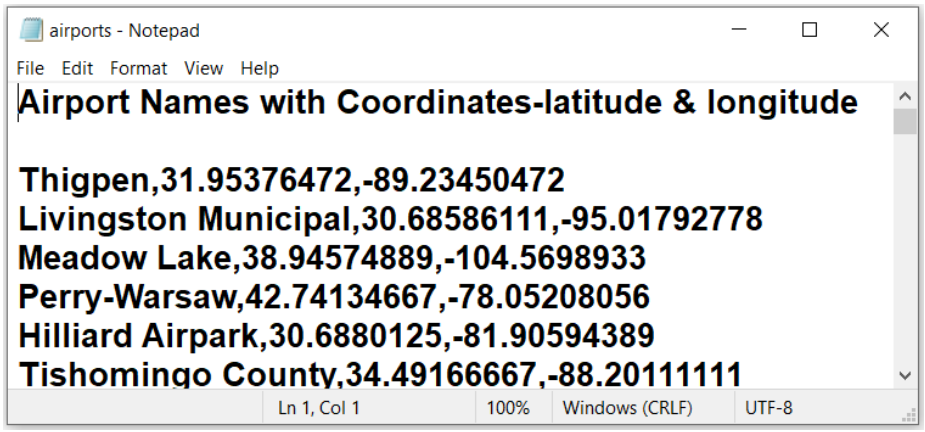
Code for creating the text file

```
file = open("airports.txt", "w")
file.write('Airport Names with Coordinates-latitude & longitude \n')
file.write('\n')

with open ('airports.csv') as csv_file:
    reader = csv.DictReader(csv_file)
    for col in reader:
        if col["country"] != "USA" :
            continue
        to_copy = (col['airport']+ "," + col['lat']+ "," + col['long'] + '\n' )
        file.write(to_copy)

file.close()
```

Text File containing the information.



Creating the scatter plot

- Requested information for the diagram –
  - Only for negative Longitudinal values
- Importing the data for longitude and latitude into a list.
- Two conditions are set to import only the requested information.
  - Only for USA- if the country is not USA the information about the longitude and latitude will not be appended in the list.
  - Only negative longitudinal values- if values for longitude is greater than 0, values will not be appended into the list.
- Values are first converted to float before appending in the list.
- Using these lists scatter plot is mapped, The matplotlib,pyplot module is used to plot the scatterplot.

Code for making the lists

```
with open('airports.csv') as csvfile:
    file = csv.DictReader(csvfile)
    longitude = []
    latitude  = []
    for col in file:
        if col["country"] != "USA" :
            continue
        if float(col["long"]) > 0:
            continue
        longitude.append(float(col['long']))
        latitude.append(float(col['lat']))

print(len(longitude))
print(len(latitude))
#print((longitude))
#print((latitude))
```

```
3372
3372
```

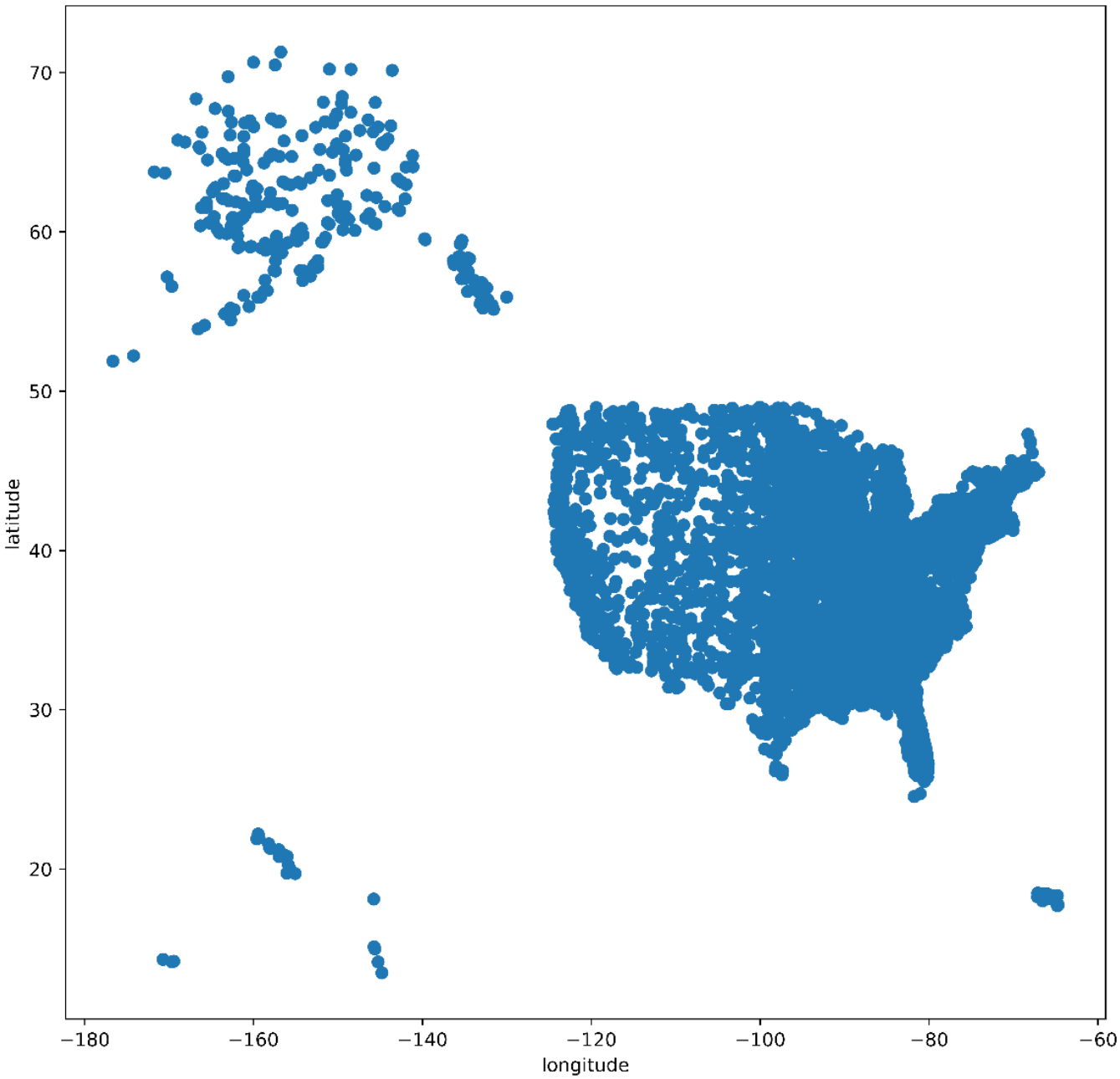
Code for making the scatterplot

```
import matplotlib.pyplot as plt

fig = plt.figure(figsize=(10, 10))
ax = fig.subplots( )

ax.scatter(x = longitude, y = latitude,)
plt.xlabel("longitude")
plt.ylabel("latitude")

plt.show()
fig.savefig('Scatterplot', dpi= 800, format= 'png')
```



- The scatter plot is made with x axis as longitude and y axis and latitude.

PART – B

Introduction

We are living in a world of huge data. However, raw data does not explain much unless it is analyzed. Our team will analyze the reasons responsible for flight delays based on two datasets. The First dataset gives information about the location (Coordinates) of a particular airport and the other dataset represents the causes of flight delays. The purpose of this analysis is to build an understanding of real-world problems. By reading our analysis, a better understanding of what are the major source of flight delays can be developed.

- **Importing Dataset**
  - Importing pandas and matplotlib module and using **pd.read_csv** command to import csv file.

```
In [1]: import pandas as pd
        from matplotlib import pyplot as plt
        from matplotlib import style

In [2]: df= pd.read_csv("airline_delay_causes_Feb2020_2.csv")

In [3]: df.head()
```

Out[3]:

	year	month	carrier	carrier_name	airport	airport_name	arr_flights	arr_del15	carrier_ct	weather_ct	...	security_ct	late_aircraft_ct	arr_can
0	2020	2	9E	Endeavor Air Inc.	ABE	Allentown/Bethlehem/Easton, PA: Lehigh Valley ...	42	7.0	1.71	0.14	...	0.0	5.08	
1	2020	2	9E	Endeavor Air Inc.	AEX	Alexandria, LA: Alexandria International	104	18.0	6.64	0.05	...	0.0	8.56	
2	2020	2	9E	Endeavor Air Inc.	AGS	Augusta, GA: Augusta Regional at Bush Field	168	27.0	8.28	0.18	...	0.0	14.22	
3	2020	2	9E	Endeavor Air Inc.	ATL	Atlanta, GA: Hartsfield-Jackson Atlanta Intern...	3365	581.0	88.22	23.66	...	0.0	294.71	
4	2020	2	9E	Endeavor Air Inc.	ATW	Appleton, WI: Appleton International	55	6.0	2.92	0.11	...	0.0	1.01	

- **Data Cleaning**
  - Data cleaning helps in improving the overall quality of our data which increases our productivity and accuracy.
  - Using **isnull ()** function to find any missing values.

Data Cleaning

```
In [72]: df.isnull().sum()
```

```
airport_name          0
 arr_flights          0
arr_del15             2
carrier_ct            0
```

The Above code is showing that arr_del15 has 2 missing values. Hence, little cleaning is needed.

```
In [73]: df[df['arr_del15'].isnull()]
```

Out[73]:

	year	month	carrier	carrier_name	airport	airport_name	arr_flights	arr_del15
568	2020	2	EV	ExpressJet Airlines LLC	PIA	Peoria, IL: General Downing - Peoria Internati...	1	NaN
1709	2020	2	YX	Republic Airline	GRB	Green Bay, WI: Green Bay Austin Straubel Inter...	1	NaN

```
In [74]: df=df.drop(labels=[568,1709], axis=0)

In [75]: df.isnull().sum()
```

After cleaning our dataset, checking once again for any missing values resulted in no NA.

- **Total number of flights**

  - Using **sum ()** function with the column arr_flights will give a total number of flights.

```
## arr_flights represents total number of arriving flights
Total_number_of_flights = df[' arr_flights'].sum()
print(Total_number_of_flights)
```

```
574266
```

```
## Total number of flights = 574266
```

- **Total number of delayed flights**
  - Using **sum ()** function with the arr_del15 column will give a total number of delayed flights.

```
## arr_del15 represents number of flights delayed
total_delayed_flights=df['arr_del15'].sum()
print(total_delayed_flights)
```

```
84616.0
```

```
## Total number of delayed flights = 84616
```

- **Total delayed time**
  - Using **sum ()** function with the arr_delay column will give the entire delayed time.

```
## arr_delay represents delay in terms of time
total_delayed_time=df['arr_delay'].sum()
print(total_delayed_time)
```

```
5819054
```

```
Total_delay_time_in_min= (total_delayed_time/60)
print(Total_delay_time_in_min)
```

```
96984.23333333334
```

- **Airport with the largest number of delayed flights**

  - Using **groupby ()** function to find total delayed flights of a particular airport. groupby () function basically divides the data into groups with respect to a particular criterion.

```
## Using groupby() function
airport_with_largest_delayed_flights=df.groupby("airport").sum()["arr_del15"]
```

```
airport_with_largest_delayed_flights.head()
```

```
airport
ABE     84.0
ABI     19.0
ABQ    219.0
ABR      4.0
ABY     10.0
```

- Using max () function to find the airport with the largest number of delayed flights which comes out to be ATL.

```
ATL    4609.0
ATW      44.0
ATY      13.0
```

```
## using max() fucntion
airport_with_largest_delayed_flights.max()
```

```
4609.0
```

```
## ATL (Atlanta, GA: Hartsfield-Jackson Atlanta International) is the airport with most number of delayed flights
```

- **Coordinates (from PART A) of the airport with the highest delayed time**

  - Before finding coordinates, firstly finding the airport with the highest delayed time which comes out to be ATL.

```
airport_with_highest_delayed_time=df.groupby("airport").sum()["arr_delay"]
```

```
print(airport_with_highest_delayed_time)
```

```
ABI       751
ABQ     10566
ABR       152
ABY      1770
ACT      1090
ACV      3755
ACY      2669
ADK         0
ADQ       210
AEX      2403
AGS      5323
ALB      7729
ALO       609
AMA      3651
ANC      9499
APN      1312
ASE     24680
ATL    352569
ATW      2828
ATY       890
```

```
airport_with_highest_delayed_time.max()
```

```
352569
```

```
## ATL is the airport with highest delayed time
## now using 1st csv file to find its coordinates
## ATL is in state Georgia (GA)
```

  - Now finding corresponding coordinates using another CSV file as shown in the code below.

```
df2=pd.read_csv("airports_1.csv")
df2.head()
```

	iata	airport	city	state	country	lat	long
0	00M	Thigpen	Bay Springs	MS	USA	31.953765	-89.234505
1	00R	Livingston Municipal	Livingston	TX	USA	30.685861	-95.017928
2	00V	Meadow Lake	Colorado Springs	CO	USA	38.945749	-104.569893
3	01G	Perry-Warsaw	Perry	NY	USA	42.741347	-78.052081
4	01J	Hilliard Airpark	Hilliard	FL	USA	30.688012	-81.905944

```
df2[df2['city']=='Atlanta']
```

	iata	airport	city	state	country	lat	long
878	ATA	Hall-Miller Municipal	Atlanta	TX	USA	33.101805	-94.195327
880	ATL	William B Hartsfield-Atlanta Intl	Atlanta	GA	USA	33.640444	-84.426944
1505	FFC	Peachtree City - Falcon	Atlanta	GA	USA	33.357250	-84.571833
1555	FTY	Fulton County - Brown	Atlanta	GA	USA	33.779139	-84.521361
2594	PDK	Dekalb-Peachtree	Atlanta	GA	USA	33.875604	-84.301968
3353	Y93	Atlanta Municipal	Atlanta	MI	USA	45.000008	-84.133337

```
###hence from the above table we can see the Coordinates of ATL which is lat-33.640444,long--84.426944
```

- **Airport in Texas which has the largest number of delayed flights.**

  - Using **str.contains ()** function to check whether a string contains a substring or not.

```
## Selecting Airport in Texas

df_tx= df[df['airport_name'].str.contains('TX')]

## creating a list
lst_tx_delay=list(df_tx['arr_del15'])

print(max(lst_tx_delay))
```

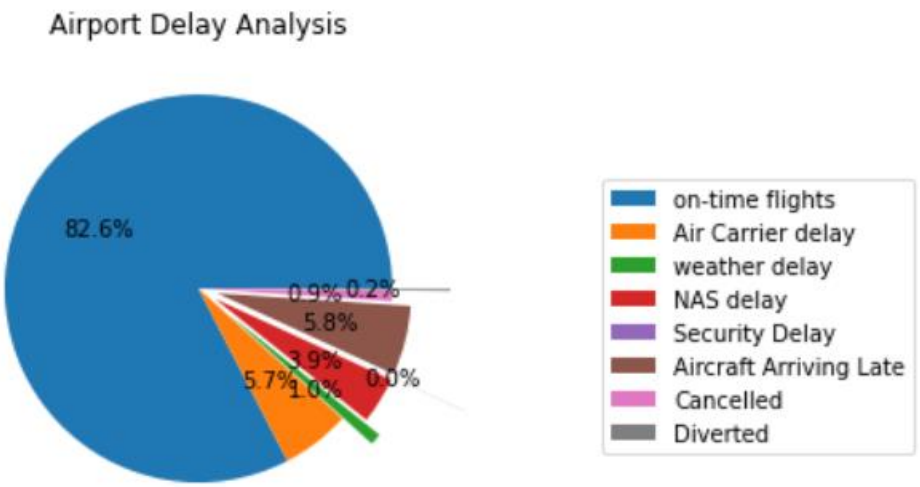
```
1812.0
```

```
# Corresponding airport in Texas
tx_delay= df_tx[df_tx['arr_del15']==1812.0]
print(tx_delay['airport_name'])
```

```
126    Dallas/Fort Worth, TX: Dallas/Fort Worth Inter...
Name: airport_name, dtype: object
```

- **<u>Pie Chart</u>**

  - A Pie chart helps in representing our data in a circular graph. Using matplotlib module for plotting pie chart.



RESULTS

- Total number of flights = 574266
- Total number of delayed flights = 84616
- Total delayed time in minutes = 96984 min
- Airport with largest number of delayed flights
  - ATL (Atlanta GA: Hartsfield – Jackson Atlanta international Airport)
- Coordinates of the airport with highest delayed time
  - latitude = 33.640444, longitude = -84.426944
- Airport in Texas that has the largest number of delayed flights.
  - Dallas/Fort Worth international Airport
- Pie chart explains the performance of flights.

CONCLUSION & RECOMMENDATIONS

- Hartsfield – Jackson international (ATL) is the busiest airport.
- Carrier delay, NAS delay, Aircraft arriving late are the main causes of delay in flights.
- Paying the team some extra head start and afterward utilizing that additional time as buffers deliberately situated all through the plans for getting work done can give a huge increase in delay reductions.
- One effective way to reduce delay is by constructing multiple airport runways.
- As per the requests, Python IDE is used to fulfil the data processing. Python csv module and matplotlib are used to fulfil part of the corresponding requests, such as importing data and visualizing requested information. All requests have been completed successfully and effectively as seen on the Results part.

REFERENCING

- Soner Yildirim (Jun 7, 2020). A practical guide for exploratory data analysis: Flight Delays. Available at: https://towardsdatascience.com/a-practical-guide-for-exploratory-data-analysis-flight-delays-f8a713ef7121.
- Python Software Foundation. (2021). CSV File Reading and Writing, Python,Org https://docs.python.org/3/library/csv.html