

2015 TRAVERSE AGENCY FLIGHT DELAYS ANALYSIS

Importing Libraries

```
In [1]: import pandas as pd  
import matplotlib.pyplot as plt  
import numpy as np  
import datetime as datetime
```

```
In [ ]:
```

Importing Datasets

```
In [2]: df_airlines = pd.read_csv('Airlines.csv')
```

```
In [3]: df_airlines
```

Out[3]:

	IATA_CODE	AIRLINE
0	UA	United Air Lines Inc.
1	AA	American Airlines Inc.
2	US	US Airways Inc.
3	F9	Frontier Airlines Inc.
4	B6	JetBlue Airways
5	OO	Skywest Airlines Inc.
6	AS	Alaska Airlines Inc.
7	NK	Spirit Air Lines
8	WN	Southwest Airlines Co.
9	DL	Delta Air Lines Inc.
10	EV	Atlantic Southeast Airlines
11	HA	Hawaiian Airlines Inc.
12	MQ	American Eagle Airlines Inc.
13	VX	Virgin America

In []:

In [4]: df_airports = pd.read_csv('Airports.csv')

In [5]: df_airports

Out[5]:

	IATA_CODE	AIRPORT	CITY	STATE	COUNTRY	LATITUDE	LONGITUDE
0	ABE	Lehigh Valley International Airport	Allentown	PA	USA	40.65236	-75.44040
1	ABI	Abilene Regional Airport	Abilene	TX	USA	32.41132	-99.68190
2	ABQ	Albuquerque International Sunport	Albuquerque	NM	USA	35.04022	-106.60919
3	ABR	Aberdeen Regional Airport	Aberdeen	SD	USA	45.44906	-98.42183
4	ABY	Southwest Georgia Regional Airport	Albany	GA	USA	31.53552	-84.19447
...
317	WRG	Wrangell Airport	Wrangell	AK	USA	56.48433	-132.36982
318	WYS	Westerly State Airport	West Yellowstone	MT	USA	44.68840	-111.11764
319	XNA	Northwest Arkansas Regional Airport	Fayetteville/Springdale/Rogers	AR	USA	36.28187	-94.30681
320	YAK	Yakutat Airport	Yakutat	AK	USA	59.50336	-139.66023
321	YUM	Yuma International Airport	Yuma	AZ	USA	32.65658	-114.60597

322 rows × 7 columns

In []:

In [6]: df_cancellation_codes = pd.read_csv('cancellation_codes.csv')

In [7]: df_cancellation_codes

Out[7]: CANCELLATION_REASON CANCELLATION_DESCRIPTION

0	A	Airline/Carrier
1	B	Weather
2	C	National Air System
3	D	Security

In []:

In [8]: `df_flights = pd.read_csv('flights.csv')`

```
C:\Users\HP\AppData\Local\Temp\ipykernel_1756\772266734.py:1: DtypeWarning: Columns (7,8) have mixed types. Specify dtype option on import or set low_memory=False.  
df_flights = pd.read_csv('flights.csv')
```

In [9]: `df_flights`

Out[9]:

	YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATION_AIRPORT
0	2015	1	1	4	AS	98	N407AS	ANC	
1	2015	1	1	4	AA	2336	N3KUAA	LAX	
2	2015	1	1	4	US	840	N171US	SFO	
3	2015	1	1	4	AA	258	N3HYAA	LAX	
4	2015	1	1	4	AS	135	N527AS	SEA	
...
5819074	2015	12	31	4	B6	688	N657JB	LAX	
5819075	2015	12	31	4	B6	745	N828JB	JFK	
5819076	2015	12	31	4	B6	1503	N913JB	JFK	
5819077	2015	12	31	4	B6	333	N527JB	MCO	
5819078	2015	12	31	4	B6	839	N534JB	JFK	

5819079 rows × 31 columns



In []:

In [10]: `flight_total = df_flights`

In []:

Extract Flights Not Cancelled From (df_flights) DataFrame

In [11]: `df = df_flights[df_flights['CANCELLED']== 0]`In [12]: `df`

Out[12]:

	YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATION_AIRPORT
0	2015	1	1	4	AS	98	N407AS	ANC	
1	2015	1	1	4	AA	2336	N3KUAA	LAX	
2	2015	1	1	4	US	840	N171US	SFO	
3	2015	1	1	4	AA	258	N3HYAA	LAX	
4	2015	1	1	4	AS	135	N527AS	SEA	
...
5819074	2015	12	31	4	B6	688	N657JB	LAX	
5819075	2015	12	31	4	B6	745	N828JB	JFK	
5819076	2015	12	31	4	B6	1503	N913JB	JFK	
5819077	2015	12	31	4	B6	333	N527JB	MCO	
5819078	2015	12	31	4	B6	839	N534JB	JFK	

5729195 rows × 31 columns



In []:

Drop Irrelevant Columns in df DataFrame

```
In [13]: df = df.drop(['DAY_OF_WEEK', 'FLIGHT_NUMBER', 'TAIL_NUMBER', 'TAXI_OUT', 'WHEELS_OFF', 'TAXI_IN', 'CANCELLATION_REASON'])
```

```
In [14]: df
```

```
Out[14]:
```

	YEAR	MONTH	DAY	AIRLINE	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	DEPARTURE_TIME
0	2015	1	1	AS	ANC	SEA	5	2354.0
1	2015	1	1	AA	LAX	PBI	10	2.0
2	2015	1	1	US	SFO	CLT	20	18.0
3	2015	1	1	AA	LAX	MIA	20	15.0
4	2015	1	1	AS	SEA	ANC	25	24.0
...
5819074	2015	12	31	B6	LAX	BOS	2359	2355.0
5819075	2015	12	31	B6	JFK	PSE	2359	2355.0
5819076	2015	12	31	B6	JFK	SJU	2359	2350.0
5819077	2015	12	31	B6	MCO	SJU	2359	2353.0
5819078	2015	12	31	B6	JFK	BQN	2359	14.0

5729195 rows × 18 columns

```
In [ ]:
```

Creating a new Column('ARRIVAL_DELAYED') in df DataFrame

```
In [15]: def arrival(x):
    if x['ARRIVAL_DELAY'] > 0:
        return 'DELAYED'
    else:
        return 'ON_TIME'

df['ARRIVAL_DELAYED'] = df.apply(arrival, axis = 1)
```

```
In [16]: df
```

Out[16]:

	YEAR	MONTH	DAY	AIRLINE	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	DEPARTURE_TIME
0	2015	1	1	AS	ANC	SEA	5	2354.0
1	2015	1	1	AA	LAX	PBI	10	2.0
2	2015	1	1	US	SFO	CLT	20	18.0
3	2015	1	1	AA	LAX	MIA	20	15.0
4	2015	1	1	AS	SEA	ANC	25	24.0
...
5819074	2015	12	31	B6	LAX	BOS	2359	2355.0
5819075	2015	12	31	B6	JFK	PSE	2359	2355.0
5819076	2015	12	31	B6	JFK	SJU	2359	2350.0
5819077	2015	12	31	B6	MCO	SJU	2359	2353.0
5819078	2015	12	31	B6	JFK	BQN	2359	14.0

5729195 rows × 19 columns



```
In [ ]:
```

Renaming the IATA_CODE and AIRLINE Columns in (df_airlines) DataFrame

```
In [17]: df_airlines = df_airlines.rename(columns = {'IATA_CODE':'AIRLINE_CODE', 'AIRLINE':'AIRLINE_NAME'})
```

```
In [18]: df_airlines
```

```
Out[18]:
```

	AIRLINE_CODE	AIRLINE_NAME
0	UA	United Air Lines Inc.
1	AA	American Airlines Inc.
2	US	US Airways Inc.
3	F9	Frontier Airlines Inc.
4	B6	JetBlue Airways
5	OO	Skywest Airlines Inc.
6	AS	Alaska Airlines Inc.
7	NK	Spirit Air Lines
8	WN	Southwest Airlines Co.
9	DL	Delta Air Lines Inc.
10	EV	Atlantic Southeast Airlines
11	HA	Hawaiian Airlines Inc.
12	MQ	American Eagle Airlines Inc.
13	VX	Virgin America

```
In [ ]:
```

Creating IATA_CODE Column in (df_flights) DataFrame

```
In [19]: df_flights['IATA_CODE'] = df_flights['ORIGIN_AIRPORT']
```

```
In [20]: df_flights
```

```
Out[20]:
```

	YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATION_AIRPORT
0	2015	1	1	4	AS	98	N407AS	ANC	
1	2015	1	1	4	AA	2336	N3KUAA	LAX	
2	2015	1	1	4	US	840	N171US	SFO	
3	2015	1	1	4	AA	258	N3HYAA	LAX	
4	2015	1	1	4	AS	135	N527AS	SEA	
...
5819074	2015	12	31	4	B6	688	N657JB	LAX	
5819075	2015	12	31	4	B6	745	N828JB	JFK	
5819076	2015	12	31	4	B6	1503	N913JB	JFK	
5819077	2015	12	31	4	B6	333	N527JB	MCO	
5819078	2015	12	31	4	B6	839	N534JB	JFK	

5819079 rows × 32 columns

```
In [ ]:
```

Create a DATE Column in (df_flights) DataFrame and convert to Timestamp

```
In [21]: df_flights['DATE'] = df_flights['YEAR'].astype(str) + '-' + df_flights['MONTH'].astype(str) + '-' + df_flights['DAY']
df_flights['DATE'] = pd.to_datetime(df_flights['DATE'])
```

```
In [22]: df_flights
```

Out[22]:

	YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATION_AIRPORT
0	2015	1	1		4	AS	98	N407AS	ANC
1	2015	1	1		4	AA	2336	N3KUAA	LAX
2	2015	1	1		4	US	840	N171US	SFO
3	2015	1	1		4	AA	258	N3HYAA	LAX
4	2015	1	1		4	AS	135	N527AS	SEA
...
5819074	2015	12	31		4	B6	688	N657JB	LAX
5819075	2015	12	31		4	B6	745	N828JB	JFK
5819076	2015	12	31		4	B6	1503	N913JB	JFK
5819077	2015	12	31		4	B6	333	N527JB	MCO
5819078	2015	12	31		4	B6	839	N534JB	JFK

5819079 rows × 33 columns



In []:

Extract YEAR, MONTH_NAME, DAY_NAME from the DATE Column in (df_flights) DataFrame

```
In [23]: df_flights['YEARS'] = (df_flights['DATE']).dt.strftime('%Y')
df_flights['MONTHS'] = df_flights['DATE'].dt.month_name()
df_flights['DAYS'] = df_flights['DATE'].dt.day_name()
```

```
In [24]: df_flights
```

Out[24]:

	YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATION_AIR
0	2015	1	1		4	AS	98	N407AS	ANC
1	2015	1	1		4	AA	2336	N3KUAA	LAX
2	2015	1	1		4	US	840	N171US	SFO
3	2015	1	1		4	AA	258	N3HYAA	LAX
4	2015	1	1		4	AS	135	N527AS	SEA
...
5819074	2015	12	31		4	B6	688	N657JB	LAX
5819075	2015	12	31		4	B6	745	N828JB	JFK
5819076	2015	12	31		4	B6	1503	N913JB	JFK
5819077	2015	12	31		4	B6	333	N527JB	MCO
5819078	2015	12	31		4	B6	839	N534JB	JFK

5819079 rows × 36 columns



In []:

Rename the AIRLINE Column in (df_flights) DataFrame as AIRLINE_CODE

```
In [25]: df_flights = df_flights.rename(columns = {'AIRLINE':'AIRLINE_CODE',})
```

```
In [26]: df_flights
```

Out[26]:

	YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE_CODE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATIO
0	2015	1	1	4	AS	98	N407AS	ANC	
1	2015	1	1	4	AA	2336	N3KUAA	LAX	
2	2015	1	1	4	US	840	N171US	SFO	
3	2015	1	1	4	AA	258	N3HYAA	LAX	
4	2015	1	1	4	AS	135	N527AS	SEA	
...
5819074	2015	12	31	4	B6	688	N657JB	LAX	
5819075	2015	12	31	4	B6	745	N828JB	JFK	
5819076	2015	12	31	4	B6	1503	N913JB	JFK	
5819077	2015	12	31	4	B6	333	N527JB	MCO	
5819078	2015	12	31	4	B6	839	N534JB	JFK	

5819079 rows × 36 columns



In []:

Join (df_airlines) & (df_flights) DataFrame on (AIRLINE_CODE) Column

```
In [27]: df_flights = df_flights.merge(df_airlines, on = 'AIRLINE_CODE', how ='inner')
```

```
In [28]: df_flights
```

Out[28]:

	YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE_CODE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATIO
0	2015	1	1	4	AS	98	N407AS	ANC	
1	2015	1	1	4	AS	135	N527AS	SEA	
2	2015	1	1	4	AS	108	N309AS	ANC	
3	2015	1	1	4	AS	122	N413AS	ANC	
4	2015	1	1	4	AS	130	N457AS	FAI	
...
5819074	2015	12	31	4	VX	769	N622VA	LGA	
5819075	2015	12	31	4	VX	357	N284VA	BOS	
5819076	2015	12	31	4	VX	1916	N853VA	SFO	
5819077	2015	12	31	4	VX	490	N840VA	LAX	
5819078	2015	12	31	4	VX	48	N281VA	OGG	

5819079 rows × 37 columns



In []:

Join (df_airports) & (df_flights) DataFrame on (IATA_CODE) Column

```
In [29]: df_flights = df_flights.merge(df_airports, on = 'IATA_CODE', how ='inner')
```

```
In [30]: df_flights
```

Out[30]:

	YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE_CODE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATIO
0	2015	1	1	4	AS	98	N407AS	ANC	
1	2015	1	1	4	AS	108	N309AS	ANC	
2	2015	1	1	4	AS	122	N413AS	ANC	
3	2015	1	1	4	AS	136	N431AS	ANC	
4	2015	1	1	4	AS	134	N464AS	ANC	
...
5332909	2015	12	31	4	WN	1471	N200WN	ISP	
5332910	2015	12	31	4	WN	5086	N903WN	ISP	
5332911	2015	12	31	4	WN	1178	N551WN	ISP	
5332912	2015	12	31	4	WN	1140	N427WN	ISP	

YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE_CODE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATIO
5332913	2015	12	31	4	WN	5472	N7723E	ISP

5332914 rows × 43 columns

In []:

Join (df_cancellation_codes) & (df_flights) DataFrame on (CANCELLATION_REASON) Column

In [31]: `df_flights = df_flights.merge(df_cancellation_codes, on = 'CANCELLATION_REASON', how ='inner')`

In [32]: `df_flights`

Out[32]:

	YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE_CODE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATION_
0	2015	1	1	4	AS	136	N431AS	ANC	
1	2015	1	11	7	AS	144	N535AS	ANC	
2	2015	1	25	7	AS	106	N767AS	ANC	
3	2015	1	26	1	AS	82	N577AS	ANC	
4	2015	1	27	2	AS	43	N768AS	ANC	
...
87425	2015	2	2	1	NK	197	N619NK	LGA	
87426	2015	1	10	6	B6	1348	N562JB	SJU	
87427	2015	5	27	3	EV	5802	N14543	MKE	

	YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE_CODE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATION_
87428	2015	3	20	5	WN	1972	N520SW	HOU	
87429	2015	7	7	2	MQ	3455	N843MQ	CHO	

87430 rows × 44 columns

In []:

Drop Irrelevant Columns in (df_flights) DataFrame

```
In [33]: df_flights = df_flights.drop(df_flights[['YEAR','DAY','FLIGHT_NUMBER','TAIL_NUMBER','TAXI_OUT','WHEELS_OFF','TAXI_IN']])  
In [34]: df_flights = df_flights.drop(df_flights[['IATA_CODE','DIVERTED','AIR_SYSTEM_DELAY','SECURITY_DELAY','AIRLINE_DELAY']])  
In [35]: df_flights = df_flights.drop(df_flights[['ELAPSED_TIME','AIR_TIME','DISTANCE','WHEELS_ON','STATE',]], axis = 1)  
In [36]: df_flights
```

Out[36]:

	MONTH	DAY_OF_WEEK	AIRLINE_CODE	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	DEPARTURE_DELAY
0	1	4	AS	ANC	SEA	135	
1	1	7	AS	ANC	PDX	200	
2	1	7	AS	ANC	SEA	2140	
3	1	1	AS	ANC	SEA	1750	
4	1	2	AS	ANC	BET	1110	
...
87425	2	1	NK	LGA	FLL	1050	
87426	1	6	B6	SJU	DCA	1904	
87427	5	3	EV	MKE	IAH	1732	

	MONTH	DAY_OF_WEEK	AIRLINE_CODE	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	DEPARTURE_TIME
87428	3	5	WN	HOU	MSY		2055
87429	7	2	MQ	CHO	LGA		1500

87430 rows × 22 columns

In []:

Replace missing values in (df_flights) DataFrame with 0

In [37]: `df_flights = df_flights.fillna(0)`

In [38]: `df_flights`

Out[38]:

	MONTH	DAY_OF_WEEK	AIRLINE_CODE	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	DEPARTURE_DELAY
0	1	4	AS	ANC	SEA	135	
1	1	7	AS	ANC	PDX	200	
2	1	7	AS	ANC	SEA	2140	
3	1	1	AS	ANC	SEA	1750	
4	1	2	AS	ANC	BET	1110	
...
87425	2	1	NK	LGA	FLL	1050	
87426	1	6	B6	SJU	DCA	1904	
87427	5	3	EV	MKE	IAH	1732	

	MONTH	DAY_OF_WEEK	AIRLINE_CODE	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	DEPARTURE_DELAY
87428	3	5	WN	HOU	MSY	2055	-15
87429	7	2	MQ	CHO	LGA	1500	0

87430 rows × 22 columns

In []:

Creating a new Column('DEPARTURE_DELAYED') in (df_flights) DataFrame for Flights that were delayed.

In [39]:

```
def delay(x):
    if x['DEPARTURE_DELAY'] > 0:
        return 'DELAYED'
    else:
        return 'ON_TIME'

df_flights[ 'DEPARTURE_DELAYED' ] = df_flights.apply(delay, axis = 1)
```

In [40]:

```
df_flights
```

Out[40]:

	MONTH	DAY_OF_WEEK	AIRLINE_CODE	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	DEPARTURE_DELAY
0	1	4	AS	ANC	SEA	135	
1	1	7	AS	ANC	PDX	200	
2	1	7	AS	ANC	SEA	2140	
3	1	1	AS	ANC	SEA	1750	
4	1	2	AS	ANC	BET	1110	
...
87425	2	1	NK	LGA	FLL	1050	
87426	1	6	B6	SJU	DCA	1904	
87427	5	3	EV	MKE	IAH	1732	

MONTH	DAY_OF_WEEK	AIRLINE_CODE	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	DEPARTURE_DELAY
87428	3	5	WN	HOU	MSY	2055
87429	7	2	MQ	CHO	LGA	1500

87430 rows × 23 columns

In []:

Drop Irrelevant Columns in (df_flights) DataFrame and Reset Index

```
In [41]: df_flights = df_flights.drop(['AIRLINE_CODE', 'SCHEDULED_TIME', 'SCHEDULED_ARRIVAL', 'ARRIVAL_TIME', 'ARRIVED_DELAYED'], axis=1)
df_flights = df_flights.reset_index(drop = True)
```

```
In [42]: df_flights
```

Out[42]:

	MONTH	DAY_OF_WEEK	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	DEPARTURE_TIME	DEPART
0	1	4	ANC	SEA	135		0.0
1	1	7	ANC	PDX	200		0.0
2	1	7	ANC	SEA	2140		0.0
3	1	1	ANC	SEA	1750		0.0
4	1	2	ANC	BET	1110		0.0
...
87425	2	1	LGA	FLL	1050		1152.0
87426	1	6	SJU	DCA	1904		0.0
87427	5	3	MKE	IAH	1732		0.0

MONTH	DAY_OF_WEEK	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	DEPARTURE_TIME	DEPART
87428	3	5	HOU	MSY	2055	0.0
87429	7	2	CHO	LGA	1500	0.0

87430 rows × 18 columns

In []:

Total Flights Volume is 5,819,079

In [43]: Total_Flights_Volume = len(flight_total)

In [44]: Total_Flights_Volume

Out[44]: 5819079

In []:

Total Flight Volume Variations By Month And Day

```

In [45]: Total_Flights_Variation_Month = df_flights.groupby('MONTHS')['MONTHS'].count()
month_cat = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'November', 'December']
Total_Flights_Variation_Month = Total_Flights_Variation_Month.reindex(month_cat)

Total_Flights_Variation_Day = df_flights.groupby('DAYS')['DAYS'].count()
day_cat = ['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday']
Total_Flights_Variation_Day = Total_Flights_Variation_Day.reindex(day_cat)

```

```
In [46]: Total_Flights_Variation_Month
```

```
Out[46]: MONTHS
January      11982
February     20517
March        11002
April         4520
May          5694
June         9120
July         4806
August        5052
September    2075
November     4599
December     8063
Name: MONTHS, dtype: int64
```

```
In [ ]:
```

```
In [47]: Total_Flights_Variation_Day
```

```
Out[47]: DAYS
Sunday       13020
Monday      20926
Tuesday      14965
Wednesday    10472
Thursday     11996
Friday        7681
Saturday      8370
Name: DAYS, dtype: int64
```

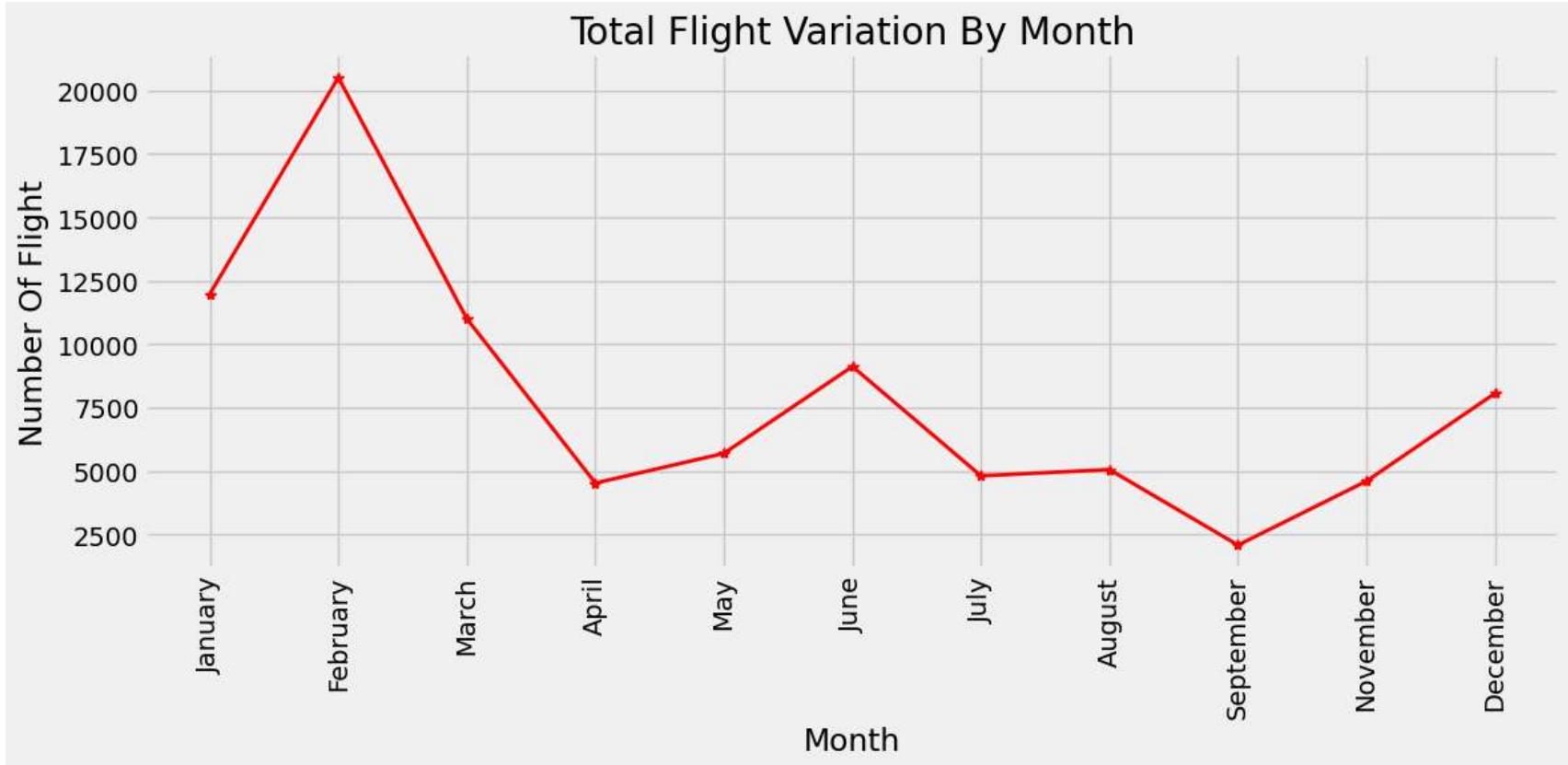
```
In [48]: plt.style.use('fivethirtyeight')
plt.tight_layout()
plt.figure(figsize = (12,6))
plt.title('Total Flight Variation By Month')
plt.xlabel('Month')
plt.ylabel('Number Of Flight')
plt.plot(Total_Flights_Variation_Month, color = 'red', marker = '*', linewidth = 2)
plt.xticks(rotation = 90)

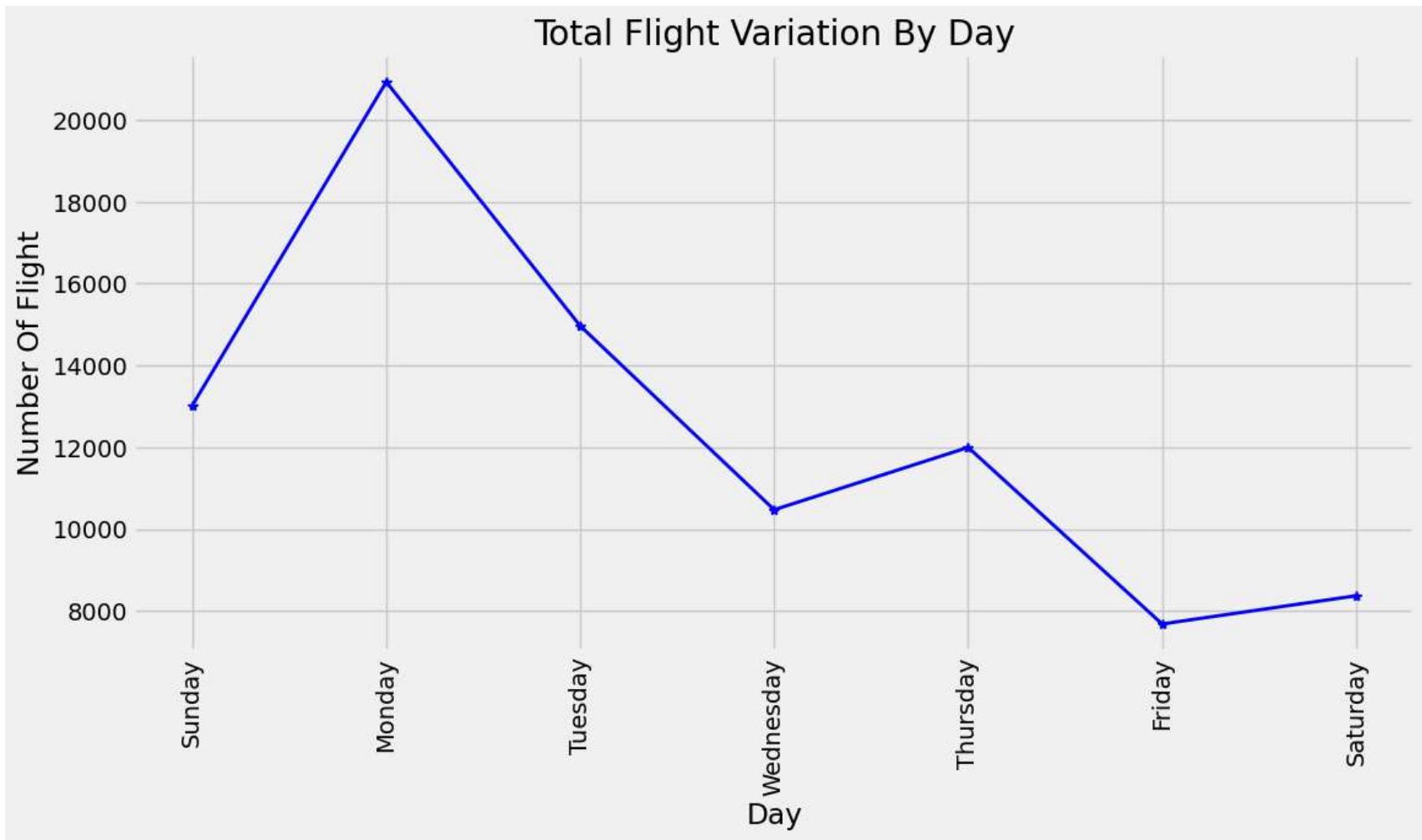
plt.style.use('fivethirtyeight')
plt.tight_layout()
```

```
plt.figure(figsize = (12,6))
plt.title('Total Flight Variation By Day')
plt.xlabel('Day')
plt.ylabel('Number Of Flight')
plt.plot(Total_Flights_Variation_Day, color = 'blue', marker ='*', linewidth = 2)
plt.xticks(rotation = 90)
plt.show()

plt.show()
```

<Figure size 640x480 with 0 Axes>





In []:

Percentage Of Flights-In That Experience Departure Delay is 36.42%

In [49]: `Arrival_Delay = (df['ARRIVAL_DELAYED'] == 'DELAYED').sum()`

```
In [50]: Arrival_Delay_Pct = Arrival_Delay/Total_Flights_Volume *100
```

```
In [51]: Arrival_Delay_Pct
```

```
Out[51]: 35.8629948141278
```

```
In [ ]:
```

Average Delay Time is 0.36(Minutes)

```
In [52]: Average_Arrival_Delay_Time = (df['ARRIVAL_DELAYED'] == 'DELAYED').mean()
```

```
In [53]: Average_Arrival_Delay_Time
```

```
Out[53]: 0.36425640949557486
```

```
In [ ]:
```

Variation Of Delayed Flight Throughout The Year

```
In [54]: Delay_Departure = df_flights[df_flights['DEPARTURE_DELAYED'] == 'DELAYED']
```

```
In [55]: Delay_Departure
```

Out[55]:

	MONTH	DAY_OF_WEEK	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	DEPARTURE_TIME	DEPART
44	8	3	ANC	DFW	2040	2048.0	
52	6	3	ANC	IAH	2040	2109.0	
99	6	4	SEA	SAN	1955	2100.0	
119	7	2	SEA	CLT	700	701.0	
121	8	6	SEA	PHL	35	313.0	
...
87126	7	2	CRP	IAH	1254	1713.0	
87164	6	1	MQT	ORD	858	920.0	
87332	8	1	CMI	ORD	1818	2013.0	

MONTH	DAY_OF_WEEK	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	DEPARTURE_TIME	DEPART
87333	11	3	CMI	ORD	1818	1915.0
87425	2	1	LGA	FLL	1050	1152.0

2480 rows × 18 columns

```
In [56]: Delay = Delay_Departure.groupby('MONTHS')[['DEPARTURE_DELAYED']].count()

month_cat = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'November', 'December']
Delay = Delay.reindex(month_cat)
```

```
In [ ]:
```

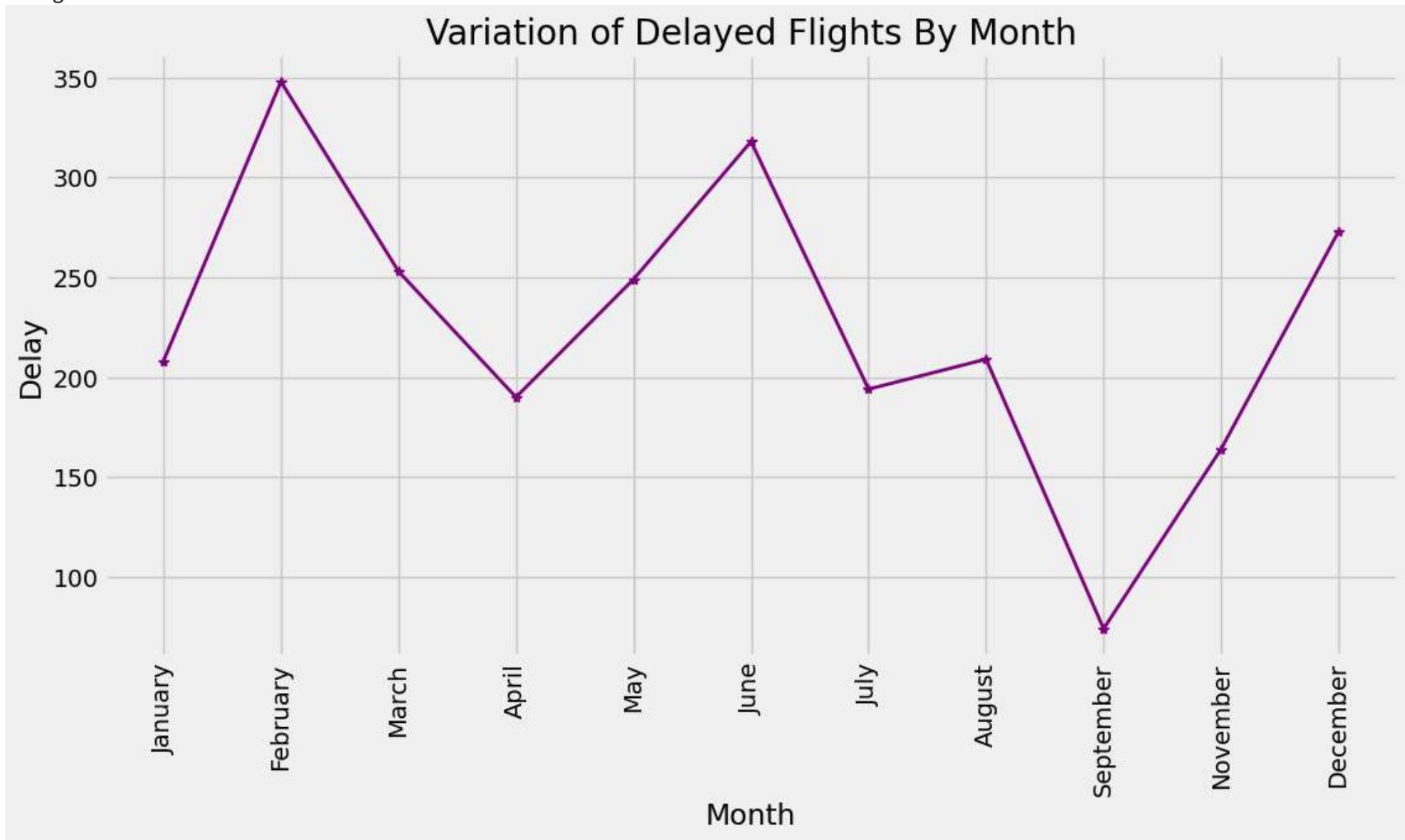
```
In [57]: Delay
```

```
Out[57]: MONTHS
January      208
February     348
March        253
April         190
May          249
June         318
July         194
August        209
September     74
November      164
December      273
Name: DEPARTURE_DELAYED, dtype: int64
```

```
In [58]: plt.style.use('fivethirtyeight')
plt.tight_layout()
plt.figure(figsize = (12,6))
```

```
plt.title('Variation of Delayed Flights By Month')
plt.xlabel('Month')
plt.ylabel('Delay')
plt.plot(Delay, color = 'purple', marker = '*', linewidth = 2)
plt.xticks(rotation = 90)
plt.show()
```

<Figure size 640x480 with 0 Axes>



In []:

Variation Of Delayed Flight Leaving Boston(BOS) Airport

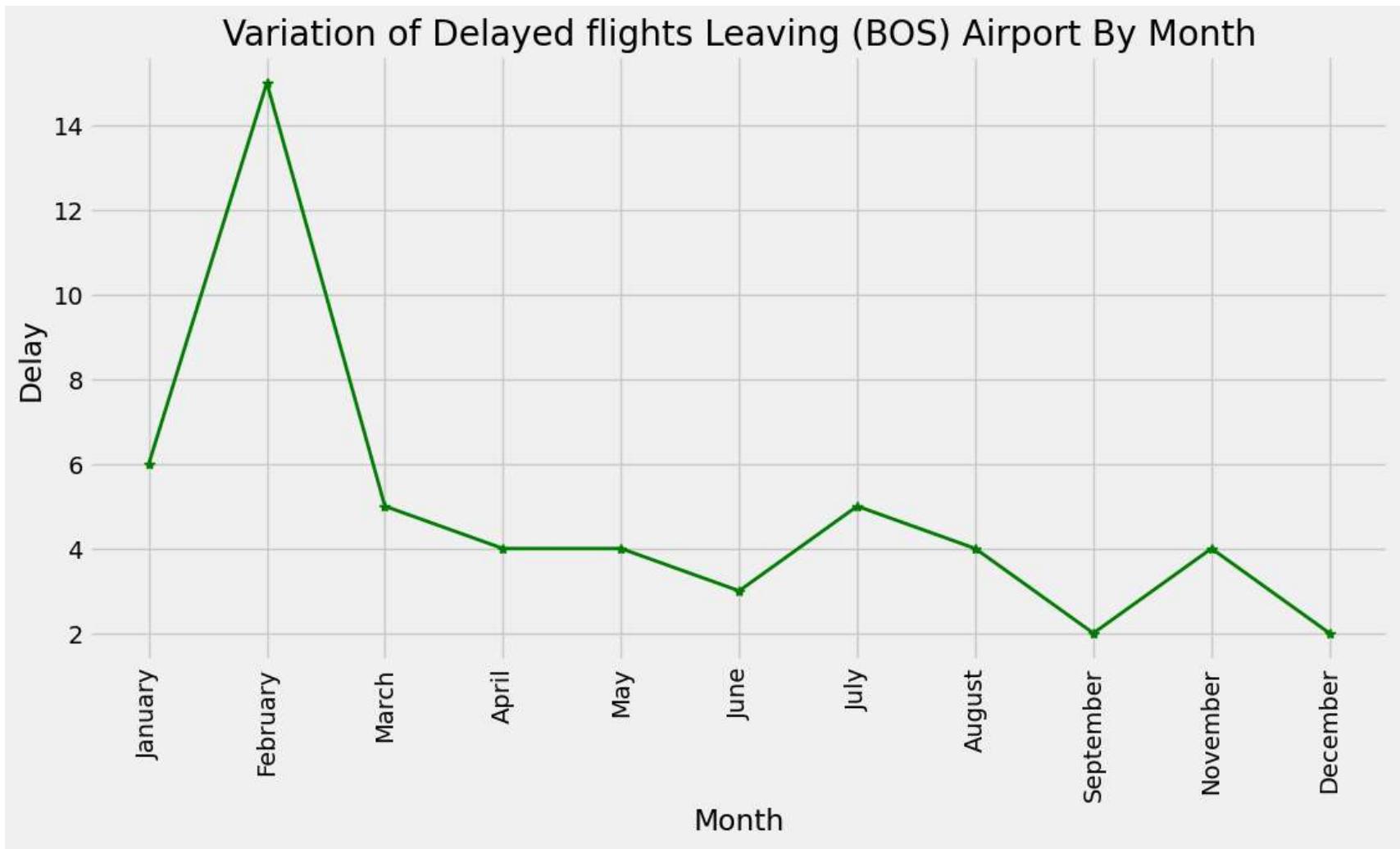
```
In [59]: Delay_Departure_BOS = df_flights[(df_flights['ORIGIN_AIRPORT'] == 'BOS') & (df_flights['DEPARTURE_DELAYED'] == 'DELAYED')]
Delay_BOS = Delay_Departure_BOS.groupby('MONTHS')['DEPARTURE_DELAYED'].count()
month_cat = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'November', 'December']
Delay_BOS = Delay_BOS.reindex(month_cat)
```

```
In [60]: Delay_BOS
```

```
Out[60]: MONTHS
January      6
February     15
March        5
April         4
May          4
June         3
July         5
August        4
September    2
November     4
December     2
Name: DEPARTURE_DELAYED, dtype: int64
```

```
In [61]: plt.style.use('fivethirtyeight')
plt.tight_layout()
plt.figure(figsize = (12,6))
plt.title('Variation of Delayed flights Leaving (BOS) Airport By Month')
plt.xlabel('Month')
plt.ylabel('Delay')
plt.plot(Delay_BOS, color = 'green', marker = '*', linewidth = 2)
plt.xticks(rotation = 90)
plt.show()
```

<Figure size 640x480 with 0 Axes>



In []:

Total_Cancelled_Flights is 87,430

Percentage Of Cancellation Due To Weather is 54.8%

Percentage Of Cancellation Due To Airline/Carrier is 27.8%

```
In [62]: Total_Cancelled_Flights = df_flights['CANCELLED'].count()  
Total_Cancelled_Flights
```

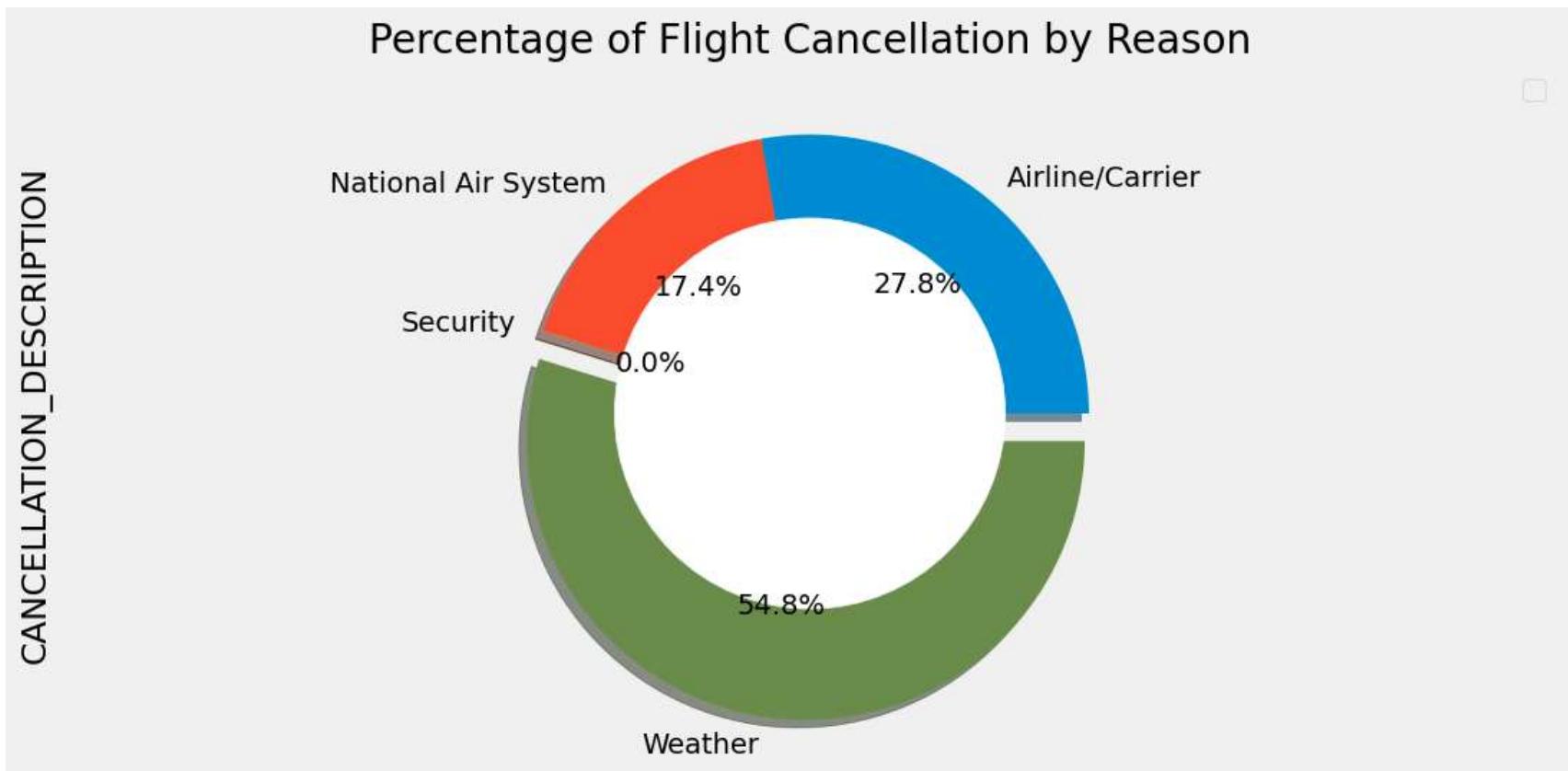
```
Out[62]: 87430
```

```
In [63]: Cancelled_Flights = df_flights.groupby('CANCELLATION_DESCRIPTION')['CANCELLATION_DESCRIPTION'].count()  
Cancelled_Flights
```

```
Out[63]: CANCELLATION_DESCRIPTION  
Airline/Carrier      24309  
National Air System  15225  
Security             22  
Weather              47874  
Name: CANCELLATION_DESCRIPTION, dtype: int64
```

```
In [64]: plt.style.use('fivethirtyeight')  
plt.tight_layout()  
plt.figure(figsize = (12,6))  
plt.title('Percentage of Flight Cancellation by Reason')  
plt.legend('Reason')  
plt.axis('equal')  
explode = [0,0,0,0.1]  
Cancelled_Flights.plot(kind = 'pie', autopct = '%1.1f%%', explode = explode, shadow = True)  
circle = plt.Circle((0,0),0.7,color = 'white')  
p = plt.gcf()  
p.gca().add_artist(circle)  
plt.show()
```

```
<Figure size 640x480 with 0 Axes>
```



In []:

Most Reliable Airline In Terms Of One-Time Departure is SouthWest Airlines Co

Least Reliable Airline In Terms Of One-Time Departure is Hawaiian Airlines Inc

In [65]: `On_Time_Departure = df_flights[df_flights['DEPARTURE_DELAYED'] == 'ON_TIME']`

In [66]: On_Time_Departure

Out[66]:

	MONTH	DAY_OF_WEEK	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	DEPARTURE_TIME	DEPART
0	1	4	ANC	SEA	135	0.0	
1	1	7	ANC	PDX	200	0.0	
2	1	7	ANC	SEA	2140	0.0	
3	1	1	ANC	SEA	1750	0.0	
4	1	2	ANC	BET	1110	0.0	
...
87424	11	1	MIA	MCO	2150	0.0	
87426	1	6	SJU	DCA	1904	0.0	
87427	5	3	MKE	IAH	1732	0.0	

MONTH	DAY_OF_WEEK	ORIGIN_AIRPORT	DESTINATION_AIRPORT	SCHEDULED_DEPARTURE	DEPARTURE_TIME	DEPART
87428	3	5	HOU	MSY	2055	0.0
87429	7	2	CHO	LGA	1500	0.0

84950 rows × 18 columns

```
In [67]: On_Time = On_Time_Departure.groupby('AIRLINE_NAME')['AIRLINE_NAME'].count().sort_values(ascending = True)
On_Time = On_Time.sort_values(ascending = True)
```

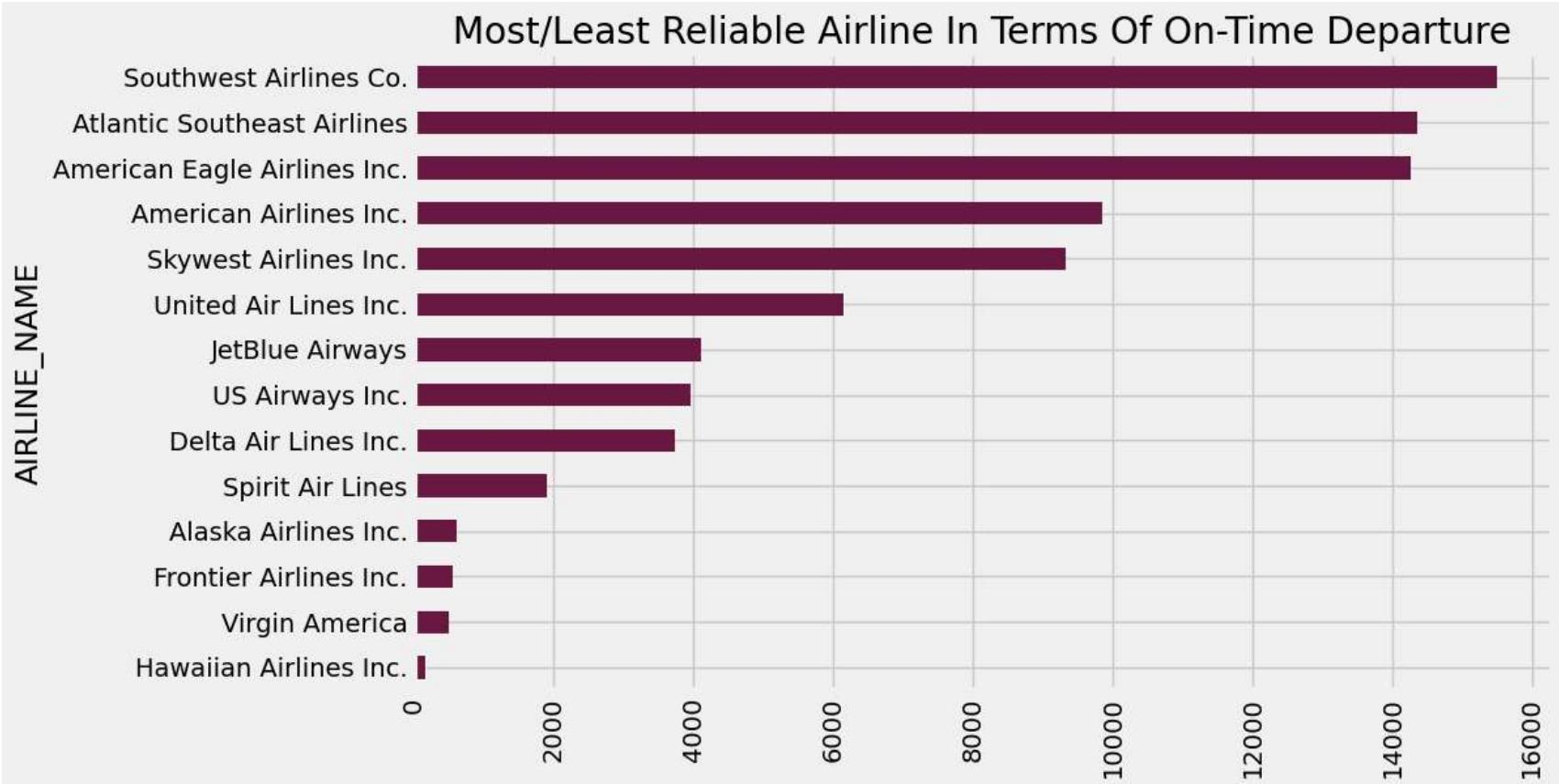
```
In [68]: On_Time
```

```
Out[68]: AIRLINE_NAME
Hawaiian Airlines Inc.          153
Virgin America                  498
Frontier Airlines Inc.          542
Alaska Airlines Inc.            614
Spirit Air Lines                1889
Delta Air Lines Inc.            3731
US Airways Inc.                 3954
JetBlue Airways                 4100
United Air Lines Inc.           6142
Skywest Airlines Inc.           9333
American Airlines Inc.          9856
American Eagle Airlines Inc.    14271
Atlantic Southeast Airlines     14366
Southwest Airlines Co.          15501
Name: AIRLINE_NAME, dtype: int64
```

```
In [69]: plt.style.use('fivethirtyeight')
plt.tight_layout()
plt.figure(figsize = (10,6))
plt.title('Most/Least Reliable Airline In Terms Of On-Time Departure')
plt.xlabel('AIRLINE_NAME')
plt.ylabel('NUMBER OF FLIGHT')
On_Time.plot(kind = 'barh',color = '#6C1D45')
```

```
plt.xticks(rotation = 90)  
plt.show()
```

<Figure size 640x480 with 0 Axes>



In []:

In []: