

Part_I_exploration_template

November 4, 2022

1 Part I - (Airline on time data Visualization)

1.1 by (Olawamiwa Adeoluwa)

1.2 Introduction

The dataset name is "Airline On-Time Performance Data". This dataset reports flights in the United States, including carriers, arrival and departure delays, and reasons for delays, from 1987 to 2008. But we will be analyzing from 2006 to 2008 due to the high volume of the data.

1.3 Preliminary Wrangling

```
In [1]: # import all packages and set plots to be embedded inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
import seaborn as sns
import requests
import tarfile
import bz2
import shutil
import urllib
import time
import datetime
import zipfile
import os
from io import BytesIO

%matplotlib inline

In [2]: unzip = zipfile.ZipFile('dataverse_files.zip')

In [3]: unzip.extractall('flight_data')

In [4]: destination = 'flight_data'
for path, subdir, files in os.walk(destination):
```

```

        for name in files:
            print(os.path.join(path,name))

flight_data/2007.csv.bz2
flight_data/2003.csv.bz2
flight_data/2004.csv.bz2
flight_data/1990.csv.bz2
flight_data/1992.csv.bz2
flight_data/1998.csv.bz2
flight_data/1996.csv.bz2
flight_data/variable-descriptions.csv
flight_data/2000.csv.bz2
flight_data/1988.csv.bz2
flight_data/1987.csv.bz2
flight_data/1989.csv.bz2
flight_data/plane-data.csv
flight_data/2006.csv.bz2
flight_data/airports.csv
flight_data/2008.csv.bz2
flight_data/1995.csv.bz2
flight_data/1994.csv.bz2
flight_data/1999.csv.bz2
flight_data/carriers.csv
flight_data/2002.csv.bz2
flight_data/1991.csv.bz2
flight_data/1997.csv.bz2
flight_data/2001.csv.bz2
flight_data/1993.csv.bz2
flight_data/2005.csv.bz2

```

Load in your dataset and describe its properties through the questions below. Try and motivate your exploration goals through this section.

```

In [2]: #Will be using the last 3 data in the extracted folder 2006-2008
        data06= pd.read_csv('flight_data/2006.csv.bz2')
        data07= pd.read_csv('flight_data/2007.csv.bz2')
        data08= pd.read_csv('flight_data/2008.csv.bz2')

In [3]: #Append all three dataframes and save into master df
        df_data=data06.append([data07,data08])

In [4]: #Getting number of rows and columns of appended data table to see structure
        df_data.shape

Out[4]: (16984354, 29)

In [5]: #Checking what df_data looks like
        df_data.head()

```

```

Out[5]:   Year  Month  DayOfMonth  DayOfWeek  DepTime  CRSDepTime  ArrTime  \
0  2006      1         11          3      743.0         745    1024.0
1  2006      1         11          3     1053.0        1053    1313.0
2  2006      1         11          3     1915.0        1915    2110.0
3  2006      1         11          3     1753.0        1755    1925.0
4  2006      1         11          3      824.0         832    1015.0

      CRSArrTime UniqueCarrier  FlightNum  ...  TaxiIn  TaxiOut  \
0          1018            US        343  ...    45.0    13.0
1          1318            US        613  ...    27.0    19.0
2          2133            US        617  ...     4.0    11.0
3          1933            US        300  ...    16.0    10.0
4          1015            US        765  ...    27.0    12.0

      Cancelled  CancellationCode  Diverted  CarrierDelay  WeatherDelay  NASDelay  \
0             0                NaN         0           0.0           0.0       0.0
1             0                NaN         0           0.0           0.0       0.0
2             0                NaN         0           0.0           0.0       0.0
3             0                NaN         0           0.0           0.0       0.0
4             0                NaN         0           0.0           0.0       0.0

      SecurityDelay  LateAircraftDelay
0              0.0              0.0
1              0.0              0.0
2              0.0              0.0
3              0.0              0.0
4              0.0              0.0

```

[5 rows x 29 columns]

```
In [4]: df_data.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6513922 entries, 0 to 6513921
Data columns (total 29 columns):
Year                int64
Month              int64
DayOfMonth          int64
DayOfWeek           int64
DepTime             float64
CRSDepTime          int64
ArrTime             float64
CRSArrTime          int64
UniqueCarrier       object
FlightNum           int64
TailNum             float64
ActualElapsedTime   float64
CRSElapsedTime      int64

```

```

AirTime          float64
ArrDelay         float64
DepDelay         float64
Origin           object
Dest             object
Distance         float64
TaxiIn           float64
TaxiOut          float64
Cancelled        int64
CancellationCode float64
Diverted         int64
CarrierDelay     float64
WeatherDelay     float64
NASDelay         float64
SecurityDelay    float64
LateAircraftDelay float64
dtypes: float64(16), int64(10), object(3)
memory usage: 1.4+ GB

```

1.3.1 Change the time Datatype

```

In [5]: # converting input time value to datetime.
def conv_time(time_val):
    if pd.isnull(time_val):
        return np.nan
    else:
        # replace 24:00 o'clock with 00:00 o'clock:
        if time_val == 2400: time_val = 0
        # creating a 4 digit value out of input value:
        time_val = "{0:04d}".format(int(time_val))
        # creating a time datatype out of input value:
        time_formatted = datetime.time(int(time_val[0:2]), int(time_val[2:4]))
    return time_formatted

In [6]: df_data['ArrTime'] = df_data['ArrTime'].apply(conv_time)
df_data['DepTime'] = df_data['DepTime'].apply(conv_time)
df_data['CRSDepTime'] = df_data['CRSDepTime'].apply(conv_time)
df_data['CRSArrTime'] = df_data['CRSArrTime'].apply(conv_time)

```

1.3.2 What is the structure of your dataset?

There are over 15 million observations in the flight for the year **2006-2008** dataset with 29 variables. Most variables are numeric in nature (Integer, Float), but the variables like flight number, Tail number, Year are Object datatypes

1.3.3 What is/are the main feature(s) of interest in your dataset?

The main features are factors that are responsible for causing delays and cancellations. These will be used to gauge the on time performance and characteristics of flights to different destinations.

1.3.4 What features in the dataset do you think will help support your investigation into your feature(s) of interest?

The features of interest are factors that are attributed to delays in arrival and departure (given in time, minutes) and cancellation codes and reasons. We can tie these up to flight carriers and destination see if we can spot patterns and relationships.

1.4 Univariate Exploration

This flight prediction univariate exploration will be focusing on their distribution.

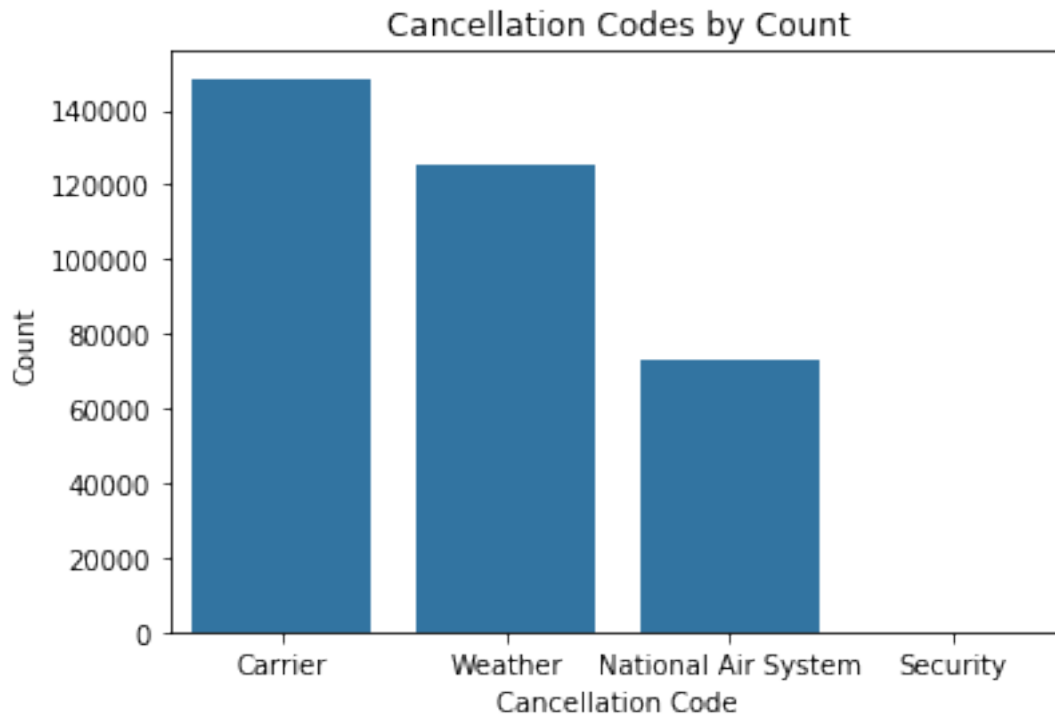
1.4.1 Which cancellation reason is most common?

In [4]: *#Lets first create a function that will rename x,y axis and chart title. We can re-use t*

```
def chart_labels(x_label,y_label,title):  
    plt.title(title)  
    plt.xlabel(x_label)  
    plt.ylabel(y_label)
```

In [7]: *#Getting distribution using Seaborn bar plot*

```
#Define Plot  
base_color=sns.color_palette()[0]  
g=sns.countplot(data=df_data,x='CancellationCode',color=base_color,order=['A','B','C','D'])  
chart_labels('Cancellation Code','Count','Cancellation Codes by Count')  
g.set_xticklabels(['Carrier','Weather','National Air System','Security']);
```



From the graph above, it is clear that **Carrier** and **Weather** are the highest reasons for flight cancellations.

1.4.2 Which carrier had the most cancellations?

Once we have looked at the top reasons for cancellations, we can move on to see which carrier had the most cancellations. This can be observed by a bar plot for carrier codes for which flights were cancelled.

```
In [ ]: #Getting a subset of original dataframe that contains only cancellation date
df_data_sub=df_data.query('CancellationCode in ["A","B","C","D"]')
```

```
In [10]: #Getting distribution of carriers that had most cancellations
```

```
#Saving value counts for various carriers
```

```
carriers_cancellations=df_data_sub.UniqueCarrier.value_counts()
```

```
#Define Plot
```

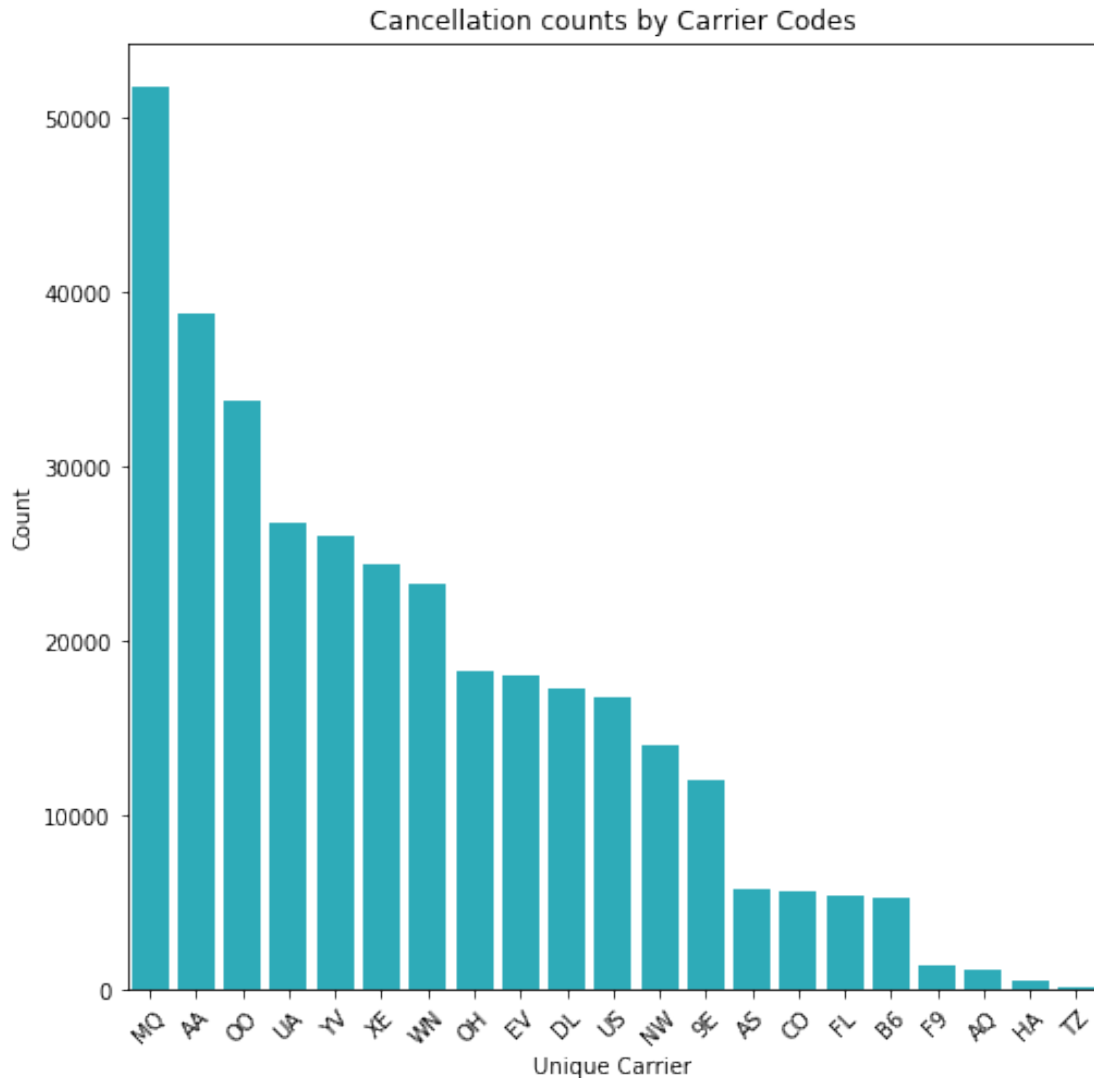
```
base_color=sns.color_palette()[9]
```

```
plt.figure(figsize=(8,8))
```

```
g=sns.countplot(data=df_data_sub,x='UniqueCarrier',color=base_color,order=carriers_canc
```

```
g.set_xticklabels(labels=carriers_cancellations.index,rotation=45)
```

```
chart_labels('Unique Carrier','Count','Cancellation counts by Carrier Codes')
```

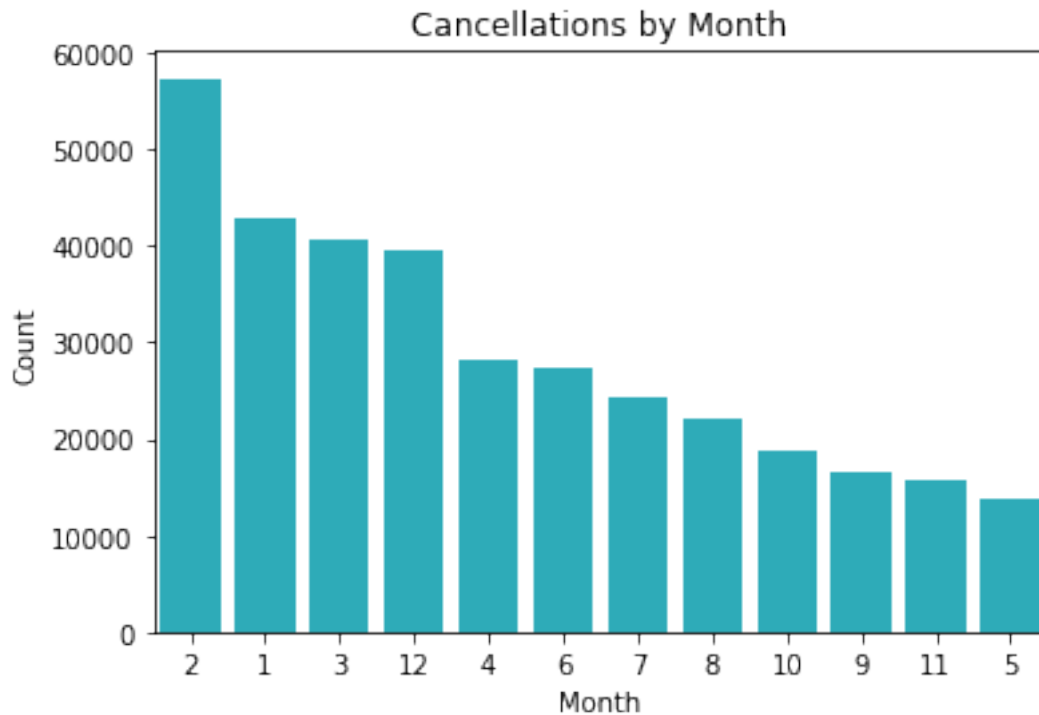


The plot above shows that MQ(Envoy Air), AA(American Airlines), OO(SkyWest Airlines) are the top three carriers with highest cancellations. These cancellations include all four types (A,B,C,D).

1.4.3 Which month of the year had most cancellations?

Moving onto the time factor, lets try to understand which months have the most cancellations for all three years.

```
In [11]: #Define Plot
month_order=df_data_sub.Month.value_counts()
sns.countplot(data=df_data_sub,x='Month',color=base_color,order=month_order.index)
chart_labels('Month','Count','Cancellations by Month');
```



It is interesting to note that the first three months see most cancellations including December and April. We can explore this more in bivariate distributions where we can filter by weather cancellations to see if this is attributed to seasons or states in US.

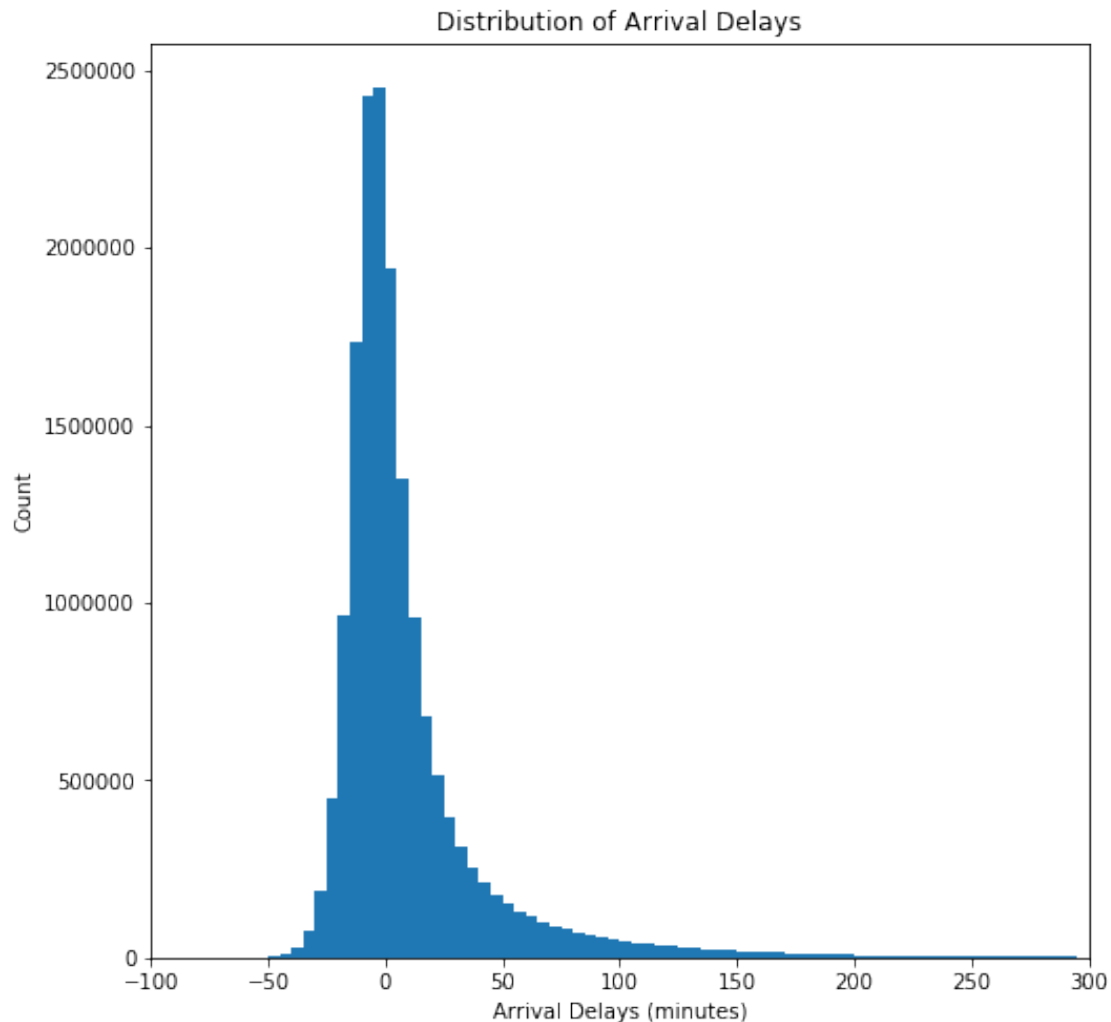
1.4.4 What does the distribution of arrival delays look like?

Now, let's take a look at arrival delays by day of the month. We will use a histogram.

```
In [7]: #Filtering only data with arrival_delay information, excluding NaN
df_arrival_delay= df_data[df_data.ArrDelay.notnull()]
```

We will plot a histogram of arrival delays. Since most of the data is concentrated between -50 and 300 we will set the x limits accordingly.

```
In [14]: #Define plot
plt.figure(figsize=(8,8))
bin_edges=np.arange(-50,300,5)
plt.hist(df_arrival_delay.ArrDelay,bins=bin_edges)
plt.xlim(-100,300)
chart_labels('Arrival Delays (minutes)', 'Count', 'Distribution of Arrival Delays')
```

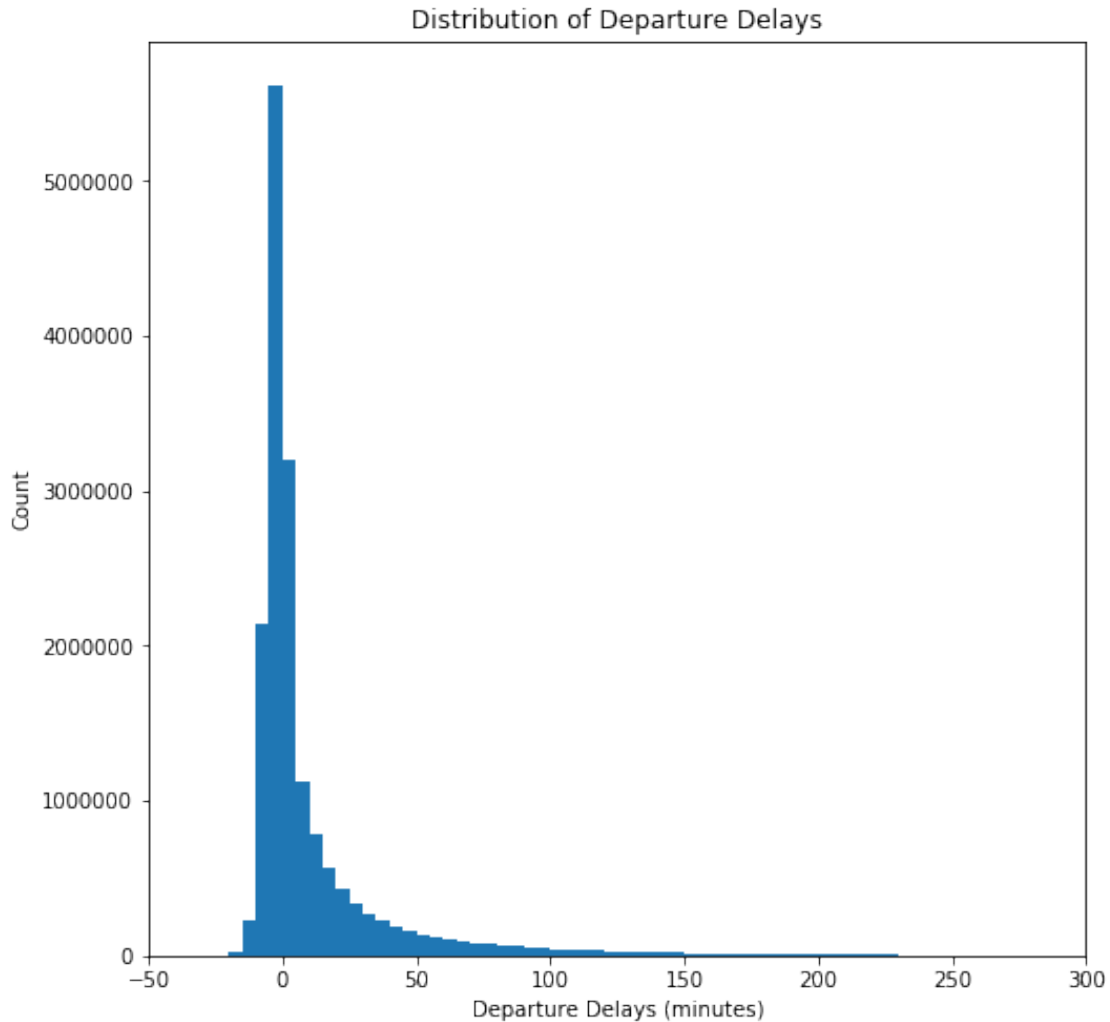
From the histogram we can see that most of the delays are concentrated between 0 and 50 minutes. The distribution is skewed to the right.

1.4.5 What does the distribution of departure delays look like?

Now, let's take a look at the distribution of departure delays.

```
In [5]: #Filtering only data with arrival_delay information, excluding NaN
df_depart_delay=df_data[df_data.DepDelay.notnull()]
```

```
In [34]: #Define plot
plt.figure(figsize=(8,8))
bin_edges=np.arange(-50,300,5)
plt.hist(df_depart_delay.DepDelay,bins=bin_edges)
plt.xlim(-50,300)
chart_labels('Departure Delays (minutes)', 'Count', 'Distribution of Departure Delays')
```



The histogram above shows the distribution of departure delays. From the graph, it can be observed that most of the delays are between 0 and 50. The graph is skewed to the right.

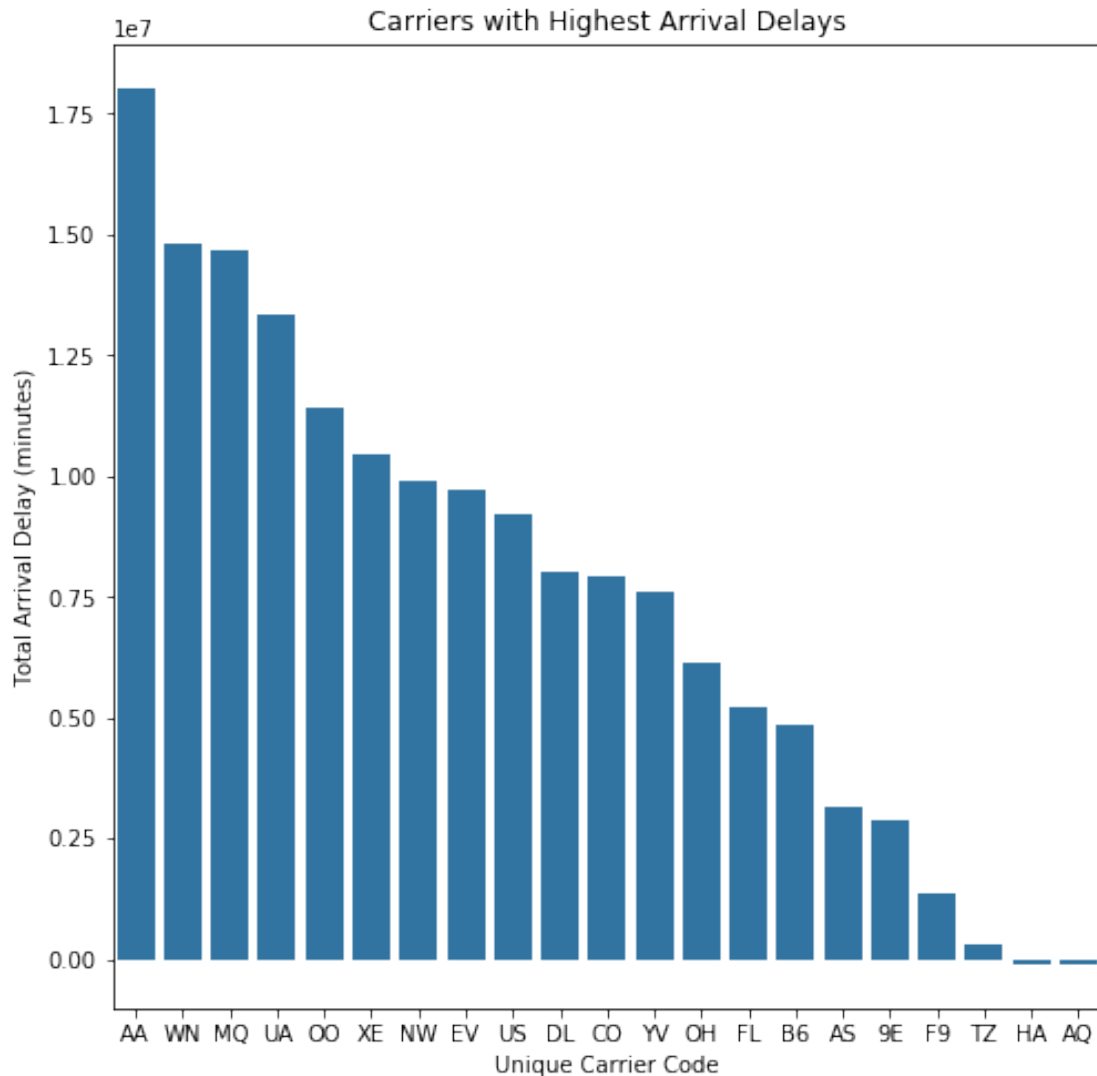
1.4.6 Which carriers had the highest delays at arrivals?

Now, let's take a look at on-time performance of flights by carriers (based on arrival delays)

```
In [28]: #Group by carrier codes and get sum of delays for each carrier
delays_carrier=df_arrival_delay.groupby('UniqueCarrier').ArrDelay.sum().sort_values(asc
```

```
In [29]: #Define bar plot
```

```
plt.figure(figsize=(8,8))
base_color=sns.color_palette()[0]
sns.barplot(x=delays_carrier.index,y=delays_carrier.values,color=base_color)
chart_labels('Unique Carrier Code','Total Arrival Delay (minutes)','Carriers with Hight
```



From the plot it can be seen that American Airlines has the highest arrival delays.

In []:

1.4.7 Discuss the distribution(s) of your variable(s) of interest. Were there any unusual points? Did you need to perform any transformations?

In the graphs above, we explored cancellation codes, carrier codes with cancellations and time of the year when cancellations were highest.

Cancellation code "D" **Security** is very less as compared to other codes. This means there have been very few instances of delays/cancellations due to security reasons.

We also explored delays in departure and arrival of flights to understand their distribution. We had to transform the histogram for these by placing limits on x axis since most of the data is concentrated there.

1.4.8 Of the features you investigated, were there any unusual distributions? Did you perform any operations on the data to tidy, adjust, or change the form of the data? If so, why did you do this?

To explore the carriers that had cancellations, we modified the original dataset and filtered only those rows that had cancellations (A,B,C or D). We would then be able to determine and observe cancellations by carriers.

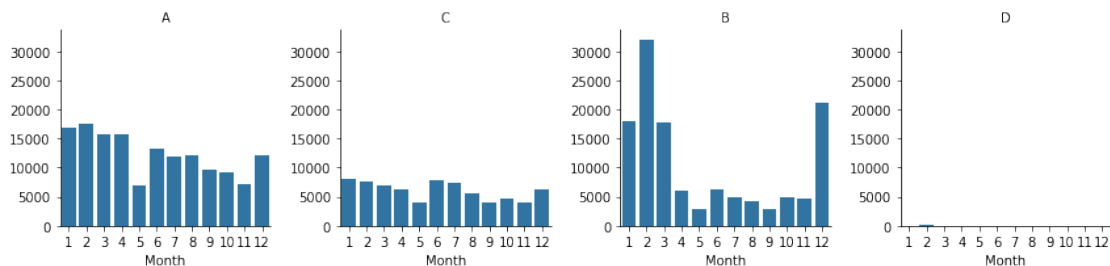
To explore delays, we created new dataframes that excluded NaN (missing) values in delays to plot histogram.

1.5 Bivariate Exploration

1.5.1 For each cancellation code (reason) which month had the most cancellations? Does this information convey anything about seasons?

This is an extension of the univariate plot where we were trying to see which months had most cancellations.

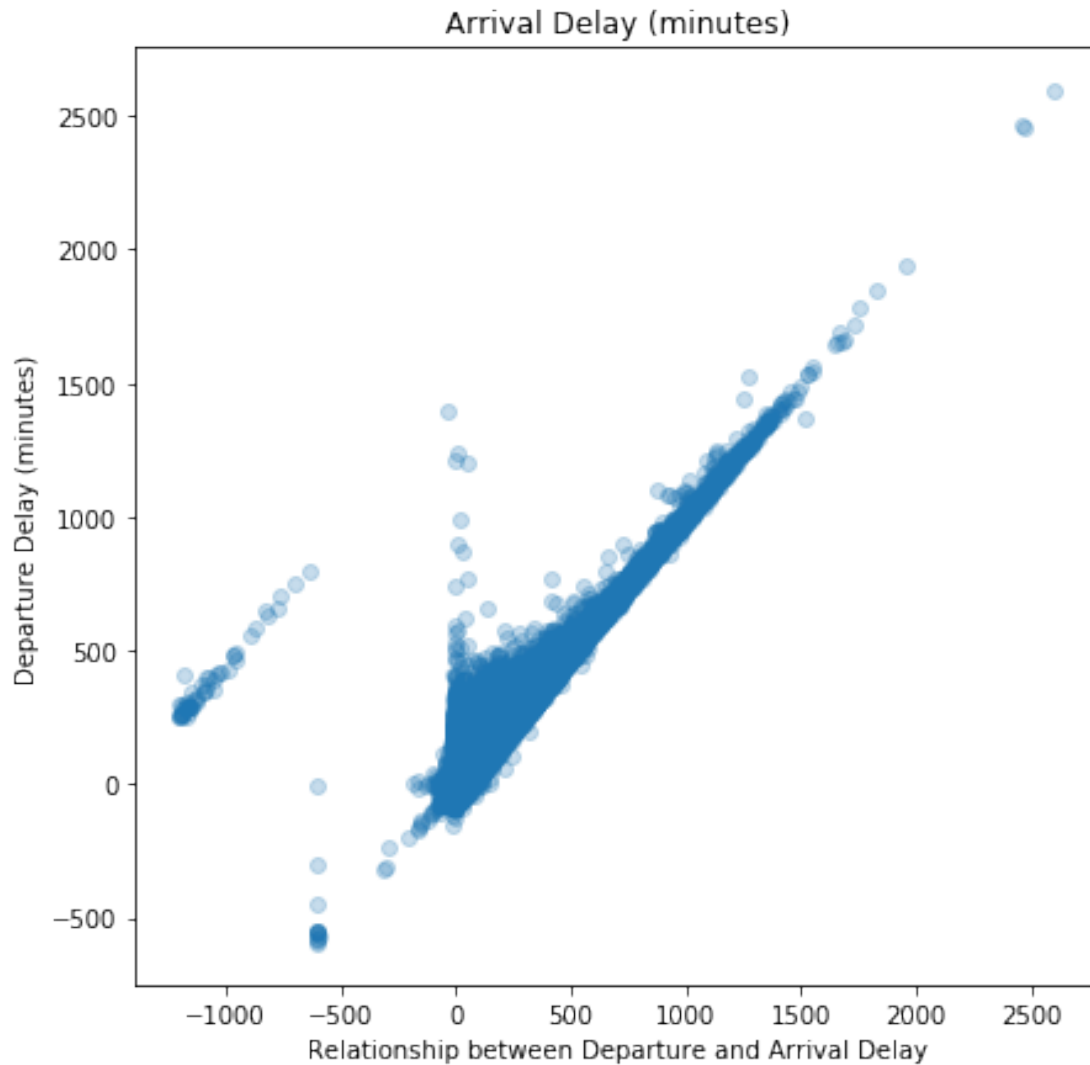
```
In [30]: #Define plot
month_order=[1,2,3,4,5,6,7,8,9,10,11,12]
g=sns.FacetGrid(data=df_data_sub,col='CancellationCode')
g.map(sns.countplot,'Month',order=month_order)
g.set_titles("{col_name}");
```



The plot above for "B" shows that the months of December, January, February and March had most cancellations. This can be due to bad weather (Since "B" is cancellations due to weather) but we will explore more plots on this later.

1.5.2 What is the relationship between departure and arrival delays?

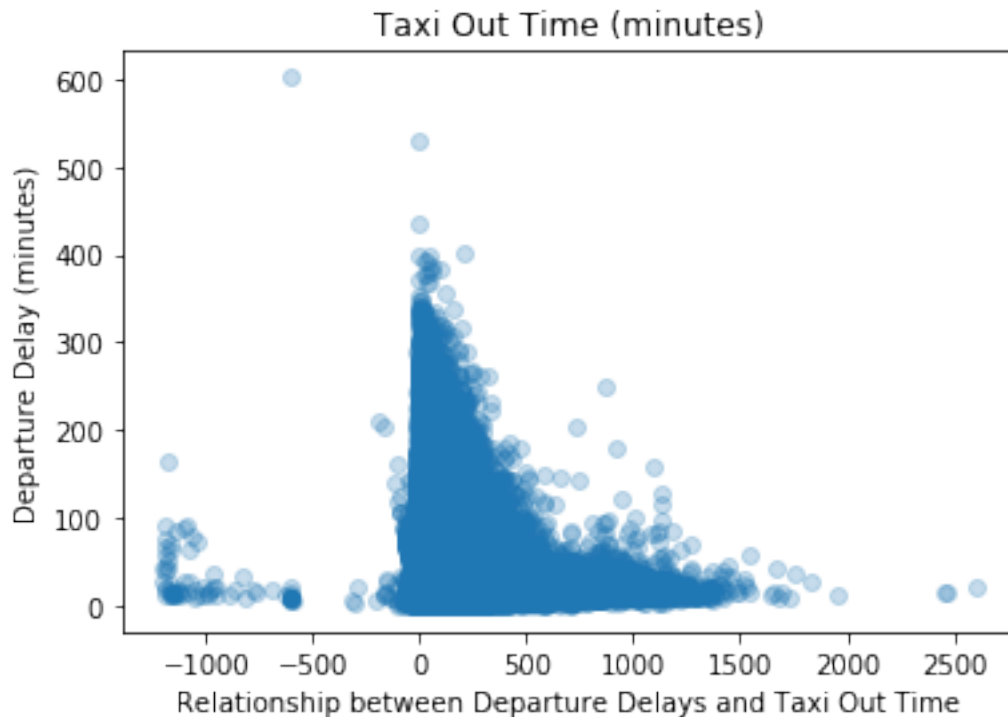
```
In [32]: #Define plot
plt.figure(figsize=(7,7))
plt.scatter(data=df_data.query('Cancelled==0'),x='DepDelay',y='ArrDelay',alpha=1/4)
chart_labels('Relationship between Departure and Arrival Delay','Departure Delay (minutes)')
```



The plot above shows a linear relationship between departure and arrival delays.

1.5.3 What is the relationship between departure delay and taxi out time?

```
In [5]: #Define plot
plt.scatter(data=df_data.query('Cancelled==0'),x='DepDelay',y='TaxiOut',alpha=1/4)
chart_labels('Relationship between Departure Delays and Taxi Out Time','Departure Delay
```



There is no linear relationship between departure delay and taxi out time.

1.5.4 What does each carrier's performance look like in terms of on time arrivals?

Lets prepare a stacked bar chart that shows each carrier's on time performance.

Note: For on time performance calculations, we will remove all cancelled flights since we only want to gauge flights that took off.

We will use "ArrDelay" i.e **Arrival Delay** for this calculation. The flights that have ArrDelay greater than 0 were delayed. We will get the proportion of flights (for each carrier) that made it on time.

```
In [6]: #Getting a dataframe with number of flights that were delayed and not cancelled by each
arrival_delayed=df_depart_delay.query('Cancelled==0 & ArrDelay>0').groupby('UniqueCarrier')

In [7]: #Getting a dataframe of uncanceled flight
uncancelled_flight=df_data.query('Cancelled==0')

In [8]: #Getting a dataframe with number of total flights that took off (not cancelled)
uncancelled_flights=uncancelled_flight.groupby('UniqueCarrier').size().reset_index(name='total')

In [9]: #Merging both dataframes on unique_carrier_code
df_merged_delay=pd.merge(arrival_delayed,uncancelled_flights,on='UniqueCarrier')

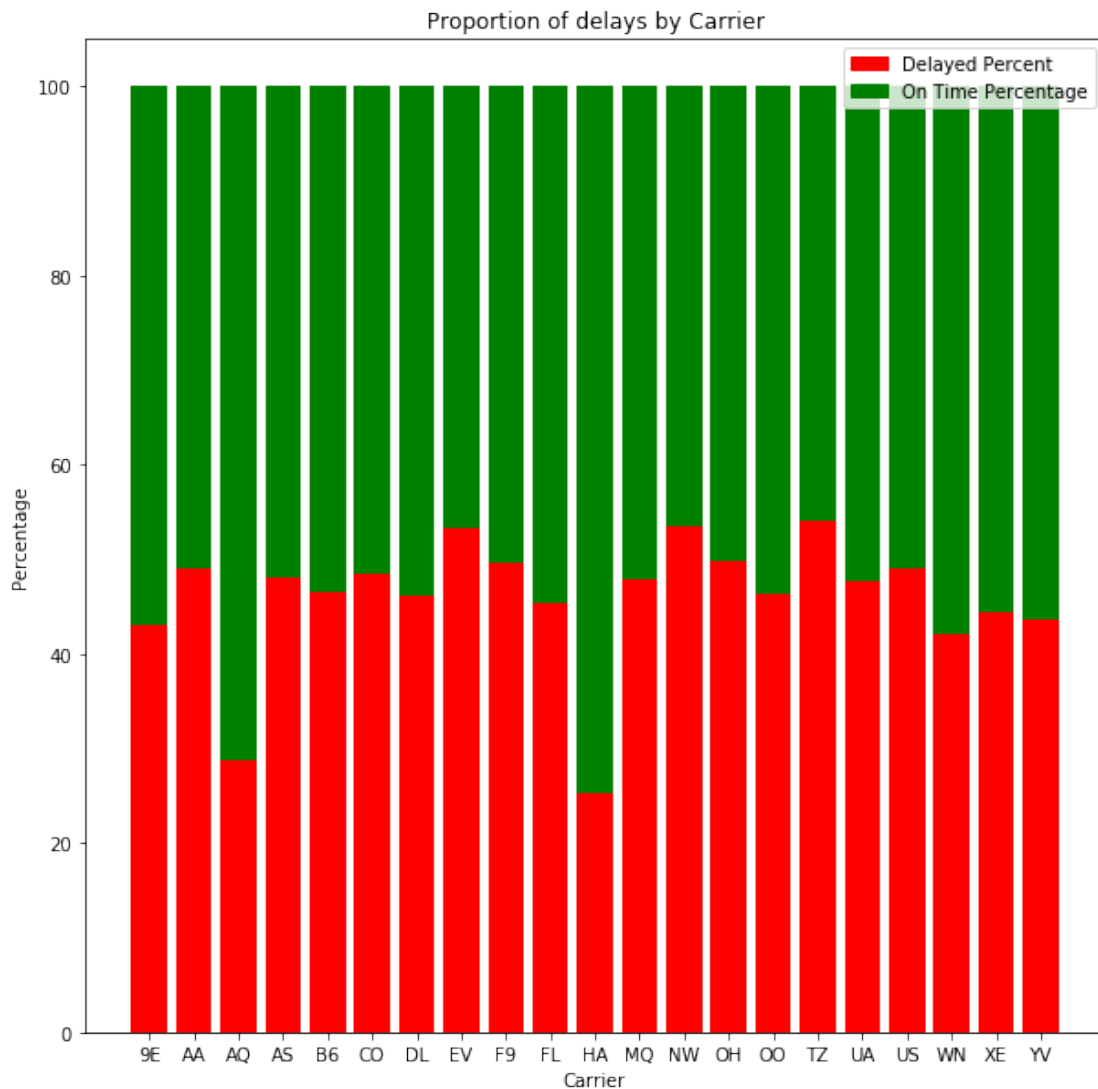
In [10]: #Calculating the percentage of delayed flights
df_merged_delay['delayed_percent']=df_merged_delay['delayed_flights']/df_merged_delay['total']
```

```

In [11]: #Creating a new column on time percentage
df_merged_delay['on_time_percent']=100-df_merged_delay.delayed_percent
df_merged_delay=df_merged_delay.sort_values(by='delayed_percent',ascending=False)

In [14]: #Define plot
plt.figure(figsize=(10,10))
plt.bar(data=df_merged_delay,x='UniqueCarrier',height='on_time_percent',color='green',b
plt.bar(data=df_merged_delay,x='UniqueCarrier',height='delayed_percent',color='red')
red_patch=mpatches.Patch(color='red',label='Delayed Percent')
green_patch=mpatches.Patch(color='green',label='On Time Percentage')
chart_labels('Carrier','Percentage','Proportion of delays by Carrier')
plt.legend(handles=[red_patch,green_patch]);

```



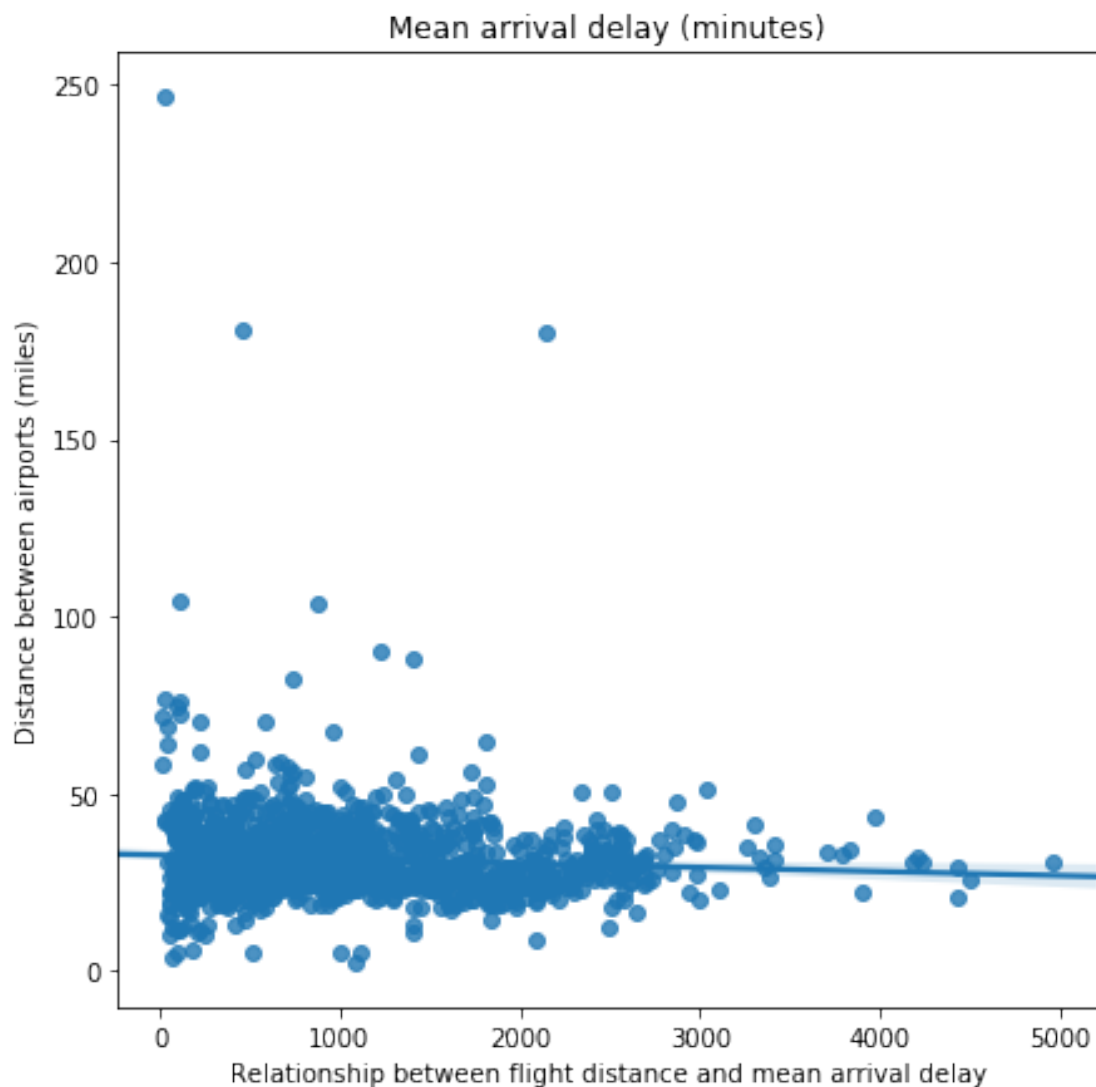
The red bars show delays as a percentage of total flights for each carrier. Larger the red bar, worse the performance.

1.5.5 What is the relationship between flight distances and mean arrival delay?

```
In [13]: #Pandas series with average arrival delay by distance
distance_arrival=df_data.query('ArrDelay>0').groupby('Distance')['ArrDelay'].mean()

In [14]: plt.figure(figsize=(7,7))

#Converting the series for seaborn plotting
x,y=pd.Series(distance_arrival.index,name='Distance'),pd.Series(distance_arrival.values)
sns.regplot(x=x,y=y)
chart_labels('Relationship between flight distance and mean arrival delay','Distance be
```



No increasing or decreasing trend in average delay as the distance between airports increases.

1.5.6 Talk about some of the relationships you observed in this part of the investigation. How did the feature(s) of interest vary with other features in the dataset?

We found a linear relationship between departure and arrival delays. No such relationship was found between departure delay and taxi out time.

1.5.7 Did you observe any interesting relationships between the other features (not the main feature(s) of interest)?

The relationship between departure and arrival delays needs to be further analyzed in multivariate exploration. Particularly, we need to see if there is pattern in taxi out time and departure delays due to bad weather conditions and so on. Correlation from the plot cannot definitively imply causation.

1.6 Multivariate Exploration

Investigating variables of interest between

- Weather and Aircraft related delays
- Variation in delay from months to months
- Delays due to security reason

1.6.1 Does the number of flights to destinations have an effect on arrival delays?

```
In [15]: #Creating two series that groups by destination and gets number of flights and average
group_destination=df_data.groupby('Dest').size().sort_values(ascending=False)
group_arrival=df_data.groupby('Dest')['ArrDelay'].mean()
```

```
In [16]: #Create a dataframe of top 50 busiest airports and average arrival delays for each by d
delay_destination=pd.concat([group_destination,group_arrival],axis=1).reset_index()
delay_destination.columns=['Dest','flights','avg_arrival_delay']
```

/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:2: FutureWarning: Sorting because n of pandas will change to not sort by default.

To accept the future behavior, pass 'sort=False'.

To retain the current behavior and silence the warning, pass 'sort=True'.

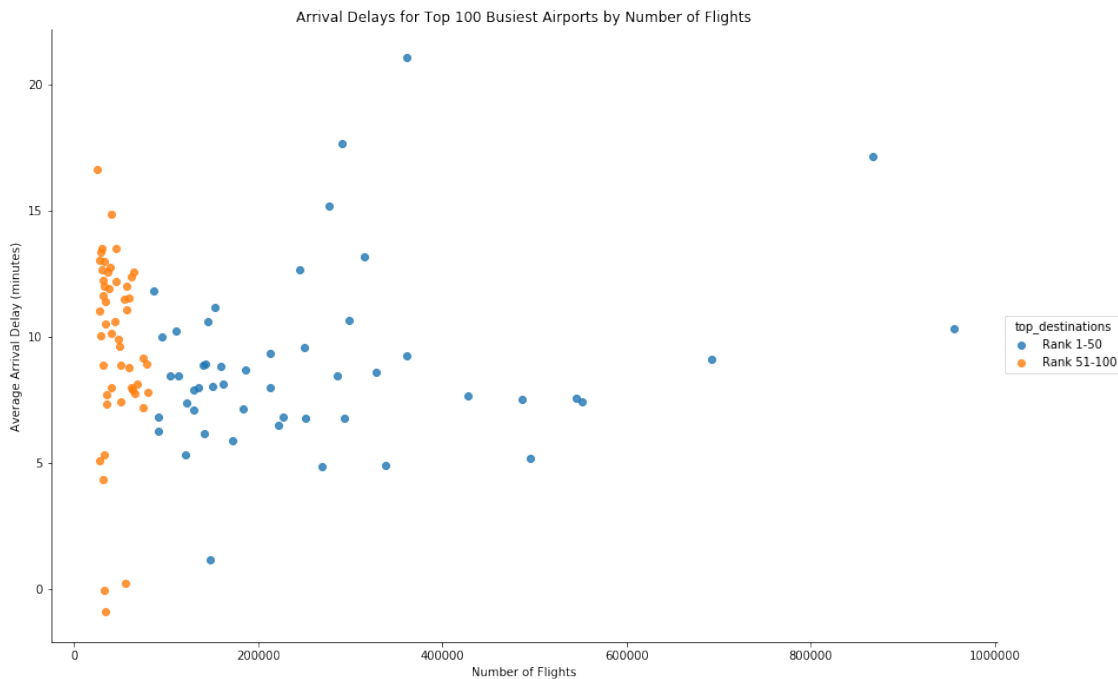
```
In [17]: #Sorting number of flight values by descending order (busiest destinations first)
delay_destination=delay_destination.sort_values('flights',ascending=False)
```

```
In [18]: #Feature Engineering: Create a new column 'top_destinations' to create two groups: Top
delay_destination['top_destinations']=np.nan
```

```
In [19]: #Setting new column for top 50
delay_destination.iloc[:50,3]='Rank 1-50'
```

```
In [20]: #Setting new column for rest
delay_destination.iloc[50:100,3]='Rank 51-100'

In [22]: #Define scatter plot with color used for ranking
g=sns.FacetGrid(data=delay_destination,hue='top_destinations',hue_order=['Rank 1-50','Rank 51-100'])
g=g.map(sns.regplot,'flights','avg_arrival_delay',x_jitter=0.4,fit_reg=False)
g.add_legend()
chart_labels('Number of Flights','Average Arrival Delay (minutes)','Arrival Delays for
```



The graph above shows that for the top 50 busiest destinations, the number of flights does have an effect on average arrival delay. This is not so much true for less busy airports that rank from 51-100.

1.6.2 What is the trend of each type of delay from month to month? Which type of delay contributes the most to flight delays? Is there a lot of variation in any of these?

```
In [24]: #Creating dataframes that will group each delay by month and calculate average delay for
carrier_by_month=df_data.query('CarrierDelay>0').groupby('Month')['CarrierDelay'].mean()

In [25]: weather_by_month=df_data.query('WeatherDelay>0').groupby('Month')['WeatherDelay'].mean()

In [26]: nas_by_month=df_data.query('NASDelay>0').groupby('Month')['NASDelay'].mean().reset_index()

In [27]: security_by_month=df_data.query('SecurityDelay>0').groupby('Month')['SecurityDelay'].mean()

In [28]: lateaircraft_by_month=df_data.query('LateAircraftDelay>0').groupby('Month')['LateAircraftDelay'].mean()
```

```

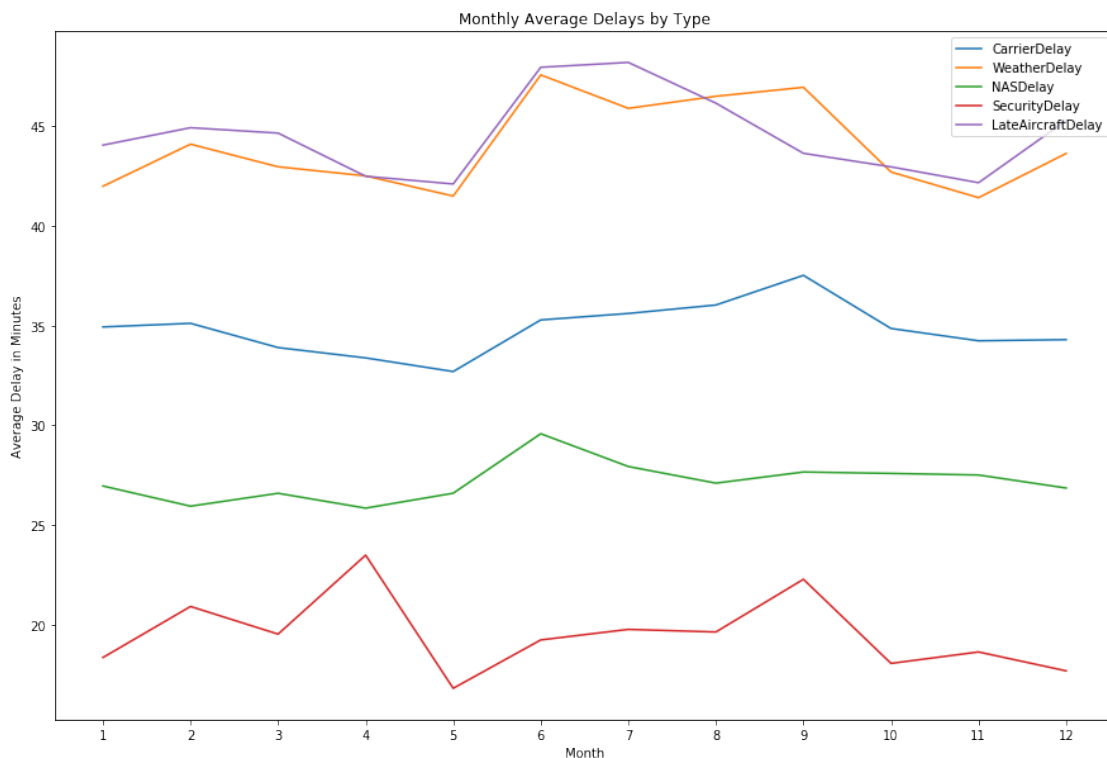
In [29]: #Joining all these dataframes into one
temp=carrier_by_month.merge(weather_by_month,on='Month')
temp=temp.merge(nas_by_month,on='Month')
temp=temp.merge(security_by_month,on='Month')
delays_by_month=temp.merge(lateaircraft_by_month,on='Month')

In [31]: #Define line plot

plt.figure(figsize=(15,10))
plt.errorbar(x=delays_by_month.Month,y=delays_by_month.CarrierDelay)
plt.errorbar(x=delays_by_month.Month,y=delays_by_month.WeatherDelay)
plt.errorbar(x=delays_by_month.Month,y=delays_by_month.NASDelay)
plt.errorbar(x=delays_by_month.Month,y=delays_by_month.SecurityDelay)
plt.errorbar(x=delays_by_month.Month,y=delays_by_month.LateAircraftDelay)
plt.legend(loc='upper right')

chart_labels('Month','Average Delay in Minutes','Monthly Average Delays by Type')
plt.xticks(delays_by_month.Month);

```



From the graph above, it can be seen that both weather and aircraft related delays contribute more to overall delays as compared to others.

It can be also observed that there is large variation in delays for these from month to month. Delays due to security reasons is the least common with less variation.

Multivariate analysis further supports the finding that weather conditions were the top reasons for delays. It can also be seen that longer flights with longer distances experience longer arrival (average) delays than shorter flights.

1.6.3 Talk about some of the relationships you observed in this part of the investigation. Were there features that strengthened each other in terms of looking at your feature(s) of interest?

- It can be seen that both weather and aircraft related delays contribute more to overall delays as compared to others.
- It can be also observed that there is large variation in delays for these from month to month.
- Delays due to security reasons is the least common with less variation.

1.6.4 Were there any interesting or surprising interactions between features?

This analysis further supports the finding that weather conditions were the top reasons for delays. It can also be seen that longer flights with longer distances experience longer arrival (average) delays than shorter flights.

1.7 Conclusions

This analysis further supports the finding that weather conditions were the top reasons for delays. It can also be seen that longer flights with longer distances experience longer arrival (average) delays than shorter flights.

After analyzing this data, we found that the highest cause of delays came from the carrier or weather related problems. American Airlines was the worst performing carrier with highest delays and cancellations from 2006 to 2008.

In []: