Part_I_Exploration

March 23, 2022

Part I - Airline On-Time Performance Data ## by Odai Alsalieti

0.1 Introduction

This database contains scheduled and actual departure and arrival times reported by certified U.S. air carriers that account for at least one percent of domestic scheduled passenger revenues. The data is collected by the Office of Airline Information, Bureau of Transportation Statistics (BTS).

Flights 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. You can see the database description here

Since this is a large dataset; there are approximately 120 million records in total, and takes up to 12 GiB storage space. So, I choose to deal with moving average for last 3 years(2006-2007-2008) downloaded here

0.2 Exploration sections:

- ### 0.2.1 Univariate Exploration
- ### 0.2.2 Bivariate Exploration
- ### 0.2.3 Multiivariate Exploration
- ### 0.2.4 Conclusion
- ### 0.2.5 Sources

0.3 Preliminary Wrangling

Go Down

[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 seaborn as sb

%matplotlib inline
```

Because the data volume is very large, I downloaded the files for the last three years and merged them into one table and selected canceled or delayed flights only to analyze them to find the reasons for that After that I saved the new extracted data in another file and deleted the rest of the data so I commented the following codes

[2]:
```
#df_06 = pd.read_csv("2006.csv", encoding ='ISO-8859-1')
#df_07 = pd.read_csv("2007.csv", encoding ='ISO-8859-1')
#df_08 = pd.read_csv("2008.csv", encoding ='ISO-8859-1')
```

[3]:
```
#print(df_06.shape)
#df_06.head()
```

[4]:
```
#print(df_07.shape)
#df_07.head()
```

[5]:
```
#print(df_08.shape)
#df_08.head()
```

[6]:
```
#create combined data frame for delayed and cancelled flights for the years␣
↪2008 , 2007, 2008
#df = df_06.append(df_07, ignore_index=True)
#df = df.append(df_08, ignore_index=True)
```

[7]:
```
#df = df.query('ArrDelay >= 15 or DepDelay >=15 or Cancelled == 1')
```

[8]:
```
#print(df.shape)
#df.head()
```

[9]:
```
#export clean dataframe to csv for later use
#df.to_csv('flights.csv', index = False)
```

[10]:
```
# Read combined data frame Csv file
df = pd.read_csv("flights.csv", encoding ='ISO-8859-1')
```

[11]:
```
# Load 5 rows and print shape
print(df.shape)
df.head()
```

```
(6060821, 29)
```

[11]:
```
   Year  Month  DayofMonth  DayOfWeek  DepTime  CRSDepTime  ArrTime  \
0  2006      1          11          3    825.0         820   1041.0
1  2006      1          11          3      NaN        1725      NaN
2  2006      1          11          3   1752.0        1540   1923.0
3  2006      1          11          3   1153.0        1145   1324.0
4  2006      1          11          3    806.0         810   1035.0

   CRSArrTime UniqueCarrier  FlightNum  … TaxiIn  TaxiOut  Cancelled  \
0        1021            US        349  …    4.0     21.0          0
1        1845            US         69  …    0.0      0.0          1
2        1654            US        127  …    3.0     19.0          0
3        1259            US        637  …    3.0     38.0          0
4        1020            US        218  …    8.0     13.0          0

   CancellationCode  Diverted  CarrierDelay WeatherDelay NASDelay  \
0               NaN         0           0.0          0.0     20.0
1                 A         0           0.0          0.0      0.0
2               NaN         0           0.0          0.0    149.0
3               NaN         0           0.0          0.0     25.0
4               NaN         0           0.0          0.0     15.0

   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]
```

[12]:
```python
# Get info about df
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6060821 entries, 0 to 6060820
Data columns (total 29 columns):
 #   Column             Dtype
---  ------             -----
 0   Year               int64
 1   Month              int64
 2   DayofMonth         int64
 3   DayOfWeek          int64
 4   DepTime            float64
 5   CRSDepTime         int64
 6   ArrTime            float64
 7   CRSArrTime         int64
 8   UniqueCarrier      object
 9   FlightNum          int64
```

```
 10  TailNum            object
 11  ActualElapsedTime  float64
 12  CRSElapsedTime     float64
 13  AirTime            float64
 14  ArrDelay           float64
 15  DepDelay           float64
 16  Origin             object
 17  Dest               object
 18  Distance           int64
 19  TaxiIn             float64
 20  TaxiOut            float64
 21  Cancelled          int64
 22  CancellationCode   object
 23  Diverted           int64
 24  CarrierDelay       float64
 25  WeatherDelay       float64
 26  NASDelay           float64
 27  SecurityDelay      float64
 28  LateAircraftDelay  float64
dtypes: float64(14), int64(10), object(5)
memory usage: 1.3+ GB
```

[13]:

```python
# Describtion of df
df.describe()
```

[13]:

```
               Year         Month    DayofMonth     DayOfWeek       DepTime  \
count  6.060821e+06  6.060821e+06  6.060821e+06  6.060821e+06  5.641893e+06
mean   2.006983e+03  6.407828e+00  1.588541e+01  3.961709e+00  1.509631e+03
std    7.970499e-01  3.491902e+00  8.718390e+00  1.974145e+00  4.652097e+02
min    2.006000e+03  1.000000e+00  1.000000e+00  1.000000e+00  1.000000e+00
25%    2.006000e+03  3.000000e+00  8.000000e+00  2.000000e+00  1.150000e+03
50%    2.007000e+03  6.000000e+00  1.600000e+01  4.000000e+00  1.548000e+03
75%    2.008000e+03  9.000000e+00  2.300000e+01  6.000000e+00  1.902000e+03
max    2.008000e+03  1.200000e+01  3.100000e+01  7.000000e+00  2.930000e+03

         CRSDepTime       ArrTime    CRSArrTime     FlightNum  \
count  6.060821e+06  5.623909e+06  6.060821e+06  6.060821e+06
mean   1.453387e+03  1.608323e+03  1.619454e+03  2.275277e+03
std    4.370080e+02  5.560598e+02  4.695433e+02  2.015150e+03
min    0.000000e+00  1.000000e+00  0.000000e+00  1.000000e+00
25%    1.117000e+03  1.304000e+03  1.303000e+03  6.230000e+02
50%    1.509000e+03  1.720000e+03  1.701000e+03  1.567000e+03
75%    1.815000e+03  2.036000e+03  2.010000e+03  3.788000e+03
max    2.359000e+03  2.955000e+03  2.400000e+03  9.741000e+03

       ActualElapsedTime  …      Distance        TaxiIn       TaxiOut  \
count       5.622963e+06  …  6.060821e+06  5.918291e+06  5.923763e+06
```

```
mean        1.401712e+02  …  7.558384e+02  7.815988e+00  2.061965e+01
std         7.578788e+01  …  5.768816e+02  3.181661e+01  1.805588e+01
min         1.200000e+01  …  1.100000e+01  0.000000e+00  0.000000e+00
25%         8.500000e+01  …  3.340000e+02  4.000000e+00  1.100000e+01
50%         1.220000e+02  …  5.990000e+02  6.000000e+00  1.600000e+01
75%         1.730000e+02  …  9.870000e+02  8.000000e+00  2.500000e+01
max         1.879000e+03  …  4.962000e+03  1.501000e+03  4.350000e+02

          Cancelled      Diverted  CarrierDelay  WeatherDelay      NASDelay  \
count  6.060821e+06  6.060821e+06  5.709770e+06  5.709770e+06  5.709770e+06
mean   6.931668e-02  2.927326e-03  1.350526e+01  2.666869e+00  1.400809e+01
std    2.539919e-01  5.402553e-02  3.683084e+01  1.758650e+01  2.891299e+01
min    0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
25%    0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
50%    0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  1.000000e+00
75%    0.000000e+00  0.000000e+00  1.400000e+01  0.000000e+00  1.800000e+01
max    1.000000e+00  1.000000e+00  2.580000e+03  1.429000e+03  1.392000e+03

       SecurityDelay  LateAircraftDelay
count   5.709770e+06       5.709770e+06
mean    8.977472e-02       1.785028e+01
std     2.115847e+00       3.676868e+01
min     0.000000e+00       0.000000e+00
25%     0.000000e+00       0.000000e+00
50%     0.000000e+00       0.000000e+00
75%     0.000000e+00       2.100000e+01
max     3.920000e+02       1.366000e+03

[8 rows x 24 columns]
```

[14]:
```python
# Check the number of unique values
df.nunique()
```

[14]:
```
Year                    3
Month                  12
DayofMonth             31
DayOfWeek               7
DepTime              1552
CRSDepTime           1241
ArrTime              1634
CRSArrTime           1429
UniqueCarrier          21
FlightNum            7684
TailNum              6219
ActualElapsedTime     779
CRSElapsedTime        611
AirTime               900
```

```
ArrDelay             1378
DepDelay             1405
Origin                314
Dest                  314
Distance             1498
TaxiIn                288
TaxiOut               368
Cancelled               2
CancellationCode        4
Diverted                2
CarrierDelay         1226
WeatherDelay          775
NASDelay              697
SecurityDelay         237
LateAircraftDelay     694
dtype: int64
```

[15]:
```
# Check number of duplicated values
df.duplicated().sum()
```

[15]:
```
12
```

[16]:
```
# Check the number of null values
df.isnull().sum()
```

[16]:
```
Year                      0
Month                     0
DayofMonth                0
DayOfWeek                 0
DepTime              418928
CRSDepTime                0
ArrTime              436912
CRSArrTime                0
UniqueCarrier             0
FlightNum                 0
TailNum               83384
ActualElapsedTime    437858
CRSElapsedTime          860
AirTime              437858
ArrDelay             437858
DepDelay             418928
Origin                    0
Dest                      0
Distance                  0
TaxiIn               142530
TaxiOut              137058
Cancelled                 0
```

```
CancellationCode     5640704
Diverted                   0
CarrierDelay          351051
WeatherDelay          351051
NASDelay              351051
SecurityDelay         351051
LateAircraftDelay     351051
dtype: int64
```

[17]:
```
# Drop duplicated rows
df.drop_duplicates()
```

[17]:
```
         Year  Month  DayofMonth  DayOfWeek  DepTime  CRSDepTime  ArrTime  \
0        2006      1          11          3    825.0         820   1041.0
1        2006      1          11          3      NaN        1725      NaN
2        2006      1          11          3   1752.0        1540   1923.0
3        2006      1          11          3   1153.0        1145   1324.0
4        2006      1          11          3    806.0         810   1035.0
…         …     …         …         …       …        …       …
6060816  2008     12          13          6    848.0         850   1024.0
6060817  2008     12          13          6    657.0         600    904.0
6060818  2008     12          13          6   1007.0         847   1149.0
6060819  2008     12          13          6    638.0         640    808.0
6060820  2008     12          13          6    612.0         615    923.0

         CRSArrTime UniqueCarrier  FlightNum  … TaxiIn  TaxiOut  Cancelled  \
0              1021            US        349  …    4.0     21.0          0
1              1845            US         69  …    0.0      0.0          1
2              1654            US        127  …    3.0     19.0          0
3              1259            US        637  …    3.0     38.0          0
4              1020            US        218  …    8.0     13.0          0
…               …             …         …  …    …       …
6060816        1005            DL       1628  …    4.0     44.0          0
6060817         749            DL       1631  …   15.0     34.0          0
6060818        1010            DL       1631  …    8.0     32.0          0
6060819         753            DL       1632  …   14.0     26.0          0
6060820         907            DL       1635  …    5.0     23.0          0

         CancellationCode  Diverted  CarrierDelay WeatherDelay NASDelay  \
0                     NaN         0           0.0          0.0     20.0
1                       A         0           0.0          0.0      0.0
2                     NaN         0           0.0          0.0    149.0
3                     NaN         0           0.0          0.0     25.0
4                     NaN         0           0.0          0.0     15.0
…                    …       …           …          …       …
6060816               NaN         0           0.0          0.0     19.0
6060817               NaN         0           0.0         57.0     18.0
```

```
6060818                 NaN          0            1.0           0.0      19.0
6060819                 NaN          0            0.0           0.0      15.0
6060820                 NaN          0            0.0           0.0      16.0

         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
…                …                …
6060816            0.0                0.0
6060817            0.0                0.0
6060818            0.0               79.0
6060819            0.0                0.0
6060820            0.0                0.0

[6060809 rows x 29 columns]
```

[18]:
```
# Update NaN time values in variables of interest to 0 to avoid this error␣
↪where ploting
# SVD did not converge in Linear Least Squares
df['AirTime'] = df['AirTime'].fillna(0)
df['DepDelay']= df['DepDelay'].fillna(0)
df['ArrDelay'] = df['ArrDelay'].fillna(0)
df['TaxiIn'] = df['TaxiIn'].fillna(0)
df['TaxiOut'] = df['TaxiOut'].fillna(0)
```

[19]:
```
# Change dtype of scheduled departure and arrival times to hours only
df['CRSDepTime'] = pd.to_datetime(df.CRSDepTime, format='%H', exact=False).dt.
↪hour
df['CRSArrTime'] = pd.to_datetime(df.CRSArrTime, format='%H', exact=False).dt.
↪hour
```

0.3.1 What is the structure of your dataset?

The extracted data contains 6060809 rows $\times$ 29 columns that are described here in detail

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

I am more interested in researching delayed and canceled flights and researching their causes

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

The features in the dataset that I think will help support my investigation are that all possible information is provided in the columns: **('Year', 'Month', 'DayofMonth', 'DayOfWeek', 'DepTime', 'CRSDepTime', 'ArrTime', 'CRSArrTime', 'UniqueCarrier', 'FlightNum', 'TailNum',, 'AirTime', 'ArrDelay','DepDelay', 'Origin', 'Dest', 'Distance', 'TaxiIn', 'TaxiOut', 'Cancelled' , 'CancellationCode')**

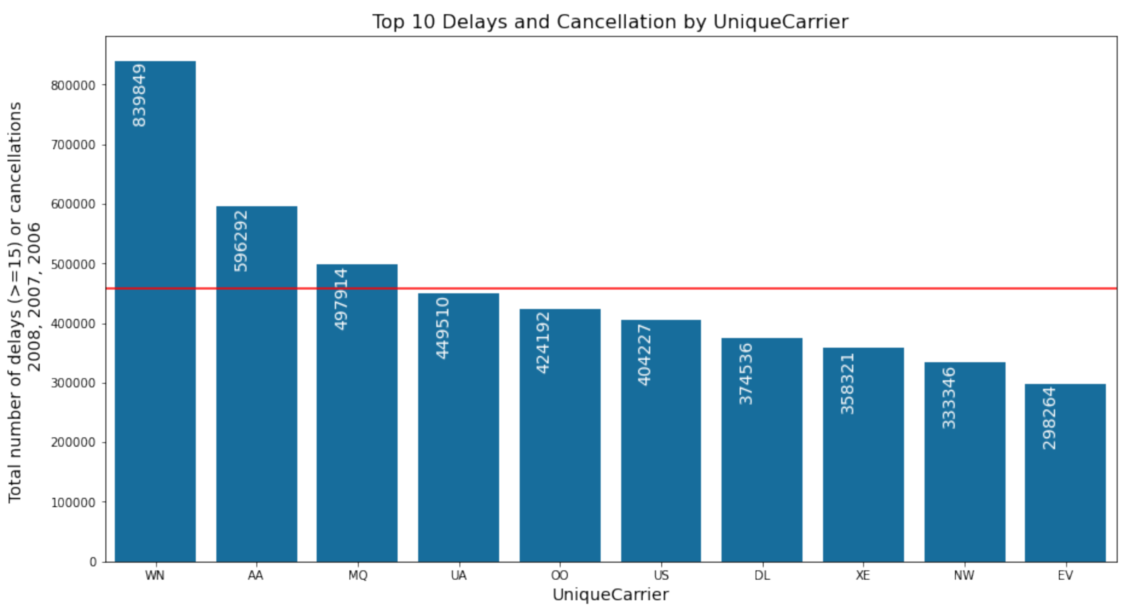
Univariate Exploration

[20]:
```python
# Define plot function To avoid re-writing the code
def plot (i):
    # Set plot size
    plt.subplots(figsize=(15,8))
    # Extract data
    plt_data = df[i].value_counts().head(10)
    # Set order values
    plt_order = plt_data.index
    # Set Plot Color
    c_palette = sb.color_palette("colorblind",10)
    colors = c_palette[0]
    # Set columns of interest
    col = ['DayofMonth', 'Month', 'CancellationCode', 'CRSDepTime',␣
↪'CRSArrTime']
    if i in col:
        data = df[i].value_counts().sort_index(ascending=True)
        ax = sb.countplot(data=df, x=i,color=colors)
        plt.axhline(data.mean(), c='red')
    elif i=='TailNum':
        # Extract data without tail numbers 0, and 000000
        tail_data = df[(df['TailNum'] != "0") & (df['TailNum'] !=␣
↪"000000")]['TailNum'].value_counts().head(10)
        # Set order values
        orders = tail_data.head(10).index
        #Plot data
        ax = sb.countplot(data=df, x = 'TailNum',color=colors,order = orders)
        plt.axhline(tail_data.mean(), c='red')
    else:
        ax = sb.countplot(data=df, x=i, color=colors, order=plt_order);
        plt.axhline(plt_data.mean(), c='red')
    #add values to bars
    for p in ax.patches:
        ax.annotate('{:}'.format(p.get_height()), (p.get_x()+0.25, p.
↪get_height()+0.03), rotation = 90,
                    color = 'white', horizontalalignment='center',␣
↪verticalalignment='top',
```

```
                    size=14)
    # Add xticklabels
    if i=='DayOfWeek':
        week_day = ['Friday','Thursday','Monday','Sunday','Wednesday',␣
↪'Tuesday',  'Saturday']
        ax.set_xticklabels(week_day)
    if i=='CancellationCode':
        cancelled_code = ['Carrier', 'Weather', 'NAS', 'Security']
        ax.set_xticklabels(cancelled_code)
    #set title and axis
    titles = 'Top 10 Delays and Cancellation by ' + i
    plt.title(titles, fontsize=16);
    plt.xlabel(i, fontsize=14);
    plt.ylabel('Total number of delays (>=15) or cancellations \n 2008, 2007,␣
↪2006', fontsize=14);
    #display plot
    plt.show();
```

0.4 What are the Top 10 Delays and Cancellation by UniqueCarrier?

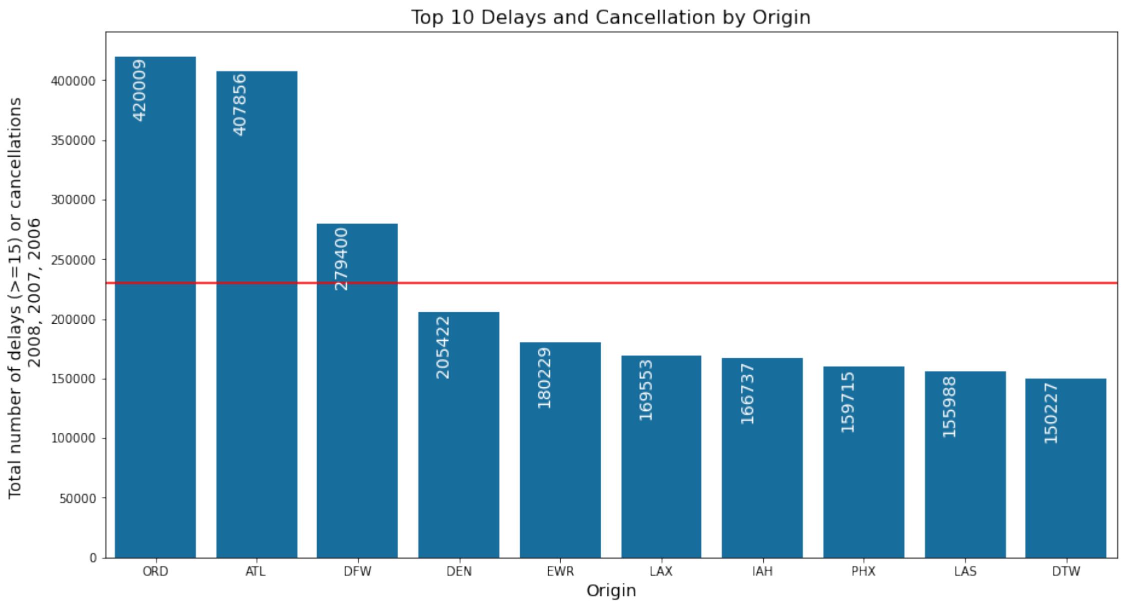
[21]:
```
# Top 10 Delays and Cancellation by UniqueCarrier
plot('UniqueCarrier')
```



Southwest Airlines , **American Airlines** and **Envoy Air**: had the most delays and cancellation over mean by Carrier in **2008**, **2007**, **2006**

0.5 What are the Top 10 Delays and Cancellation by Origin?

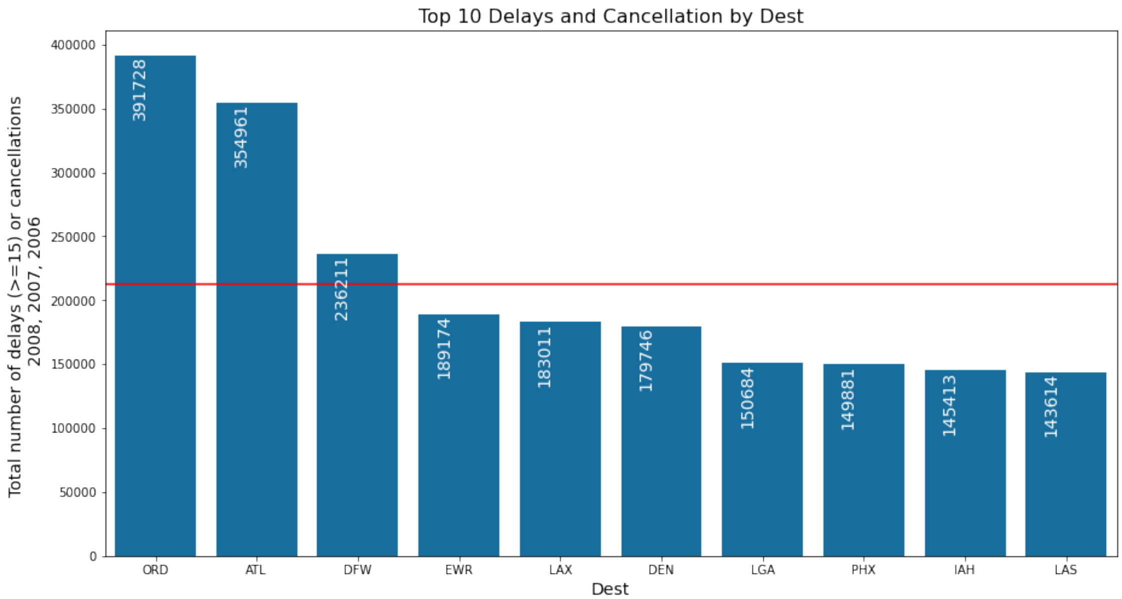
[22]:
```
# Top 10 Delays and Cancellation by Origin
plot('Origin')
```



Chicago O'Hare International Airport (ORD) , **Atlanta Airport (ATL)** and **Dallas/Ft Worth Intl(DFW)**: had the most delays and cancellation over mean by Origin in **2008, 2007, 2006**

0.6 What are the Top 10 Delays and Cancellation by Dest?

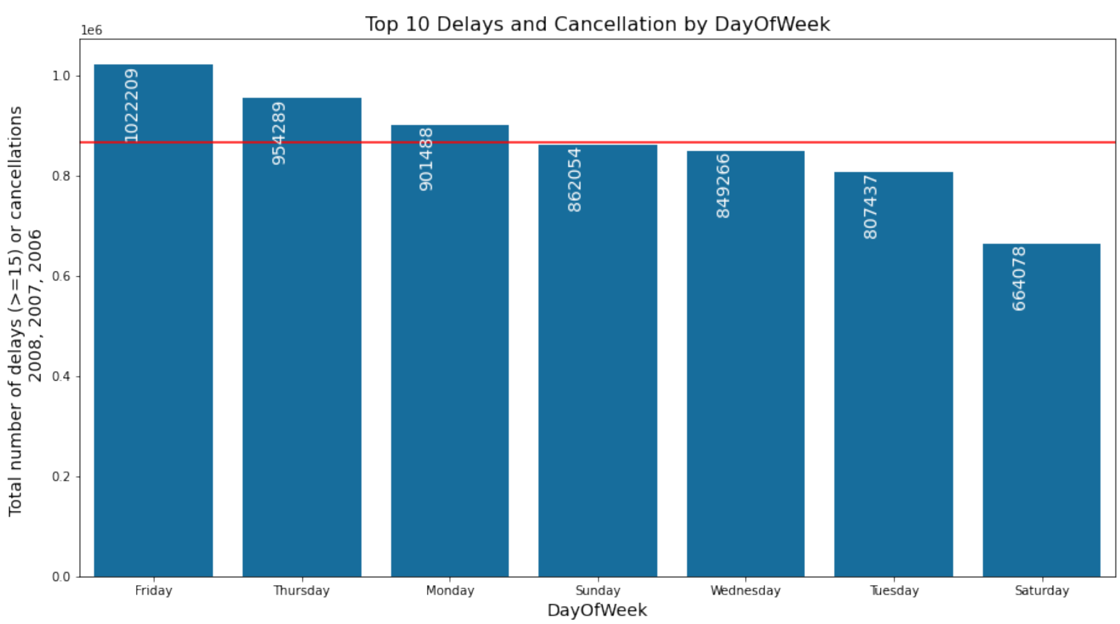
[23]:
```
# Top 10 Delays and Cancellation by Dest
plot('Dest')
```



Chicago O'Hare International Airport (ORD) , **Atlanta Airport (ATL)** and **Dallas/Ft Worth Intl(DFW)**: had the most delays and cancellation over mean by Dest in **2008, 2007, 2006**

0.7 What are the Top 10 Delays and Cancellation by DayofWeek?

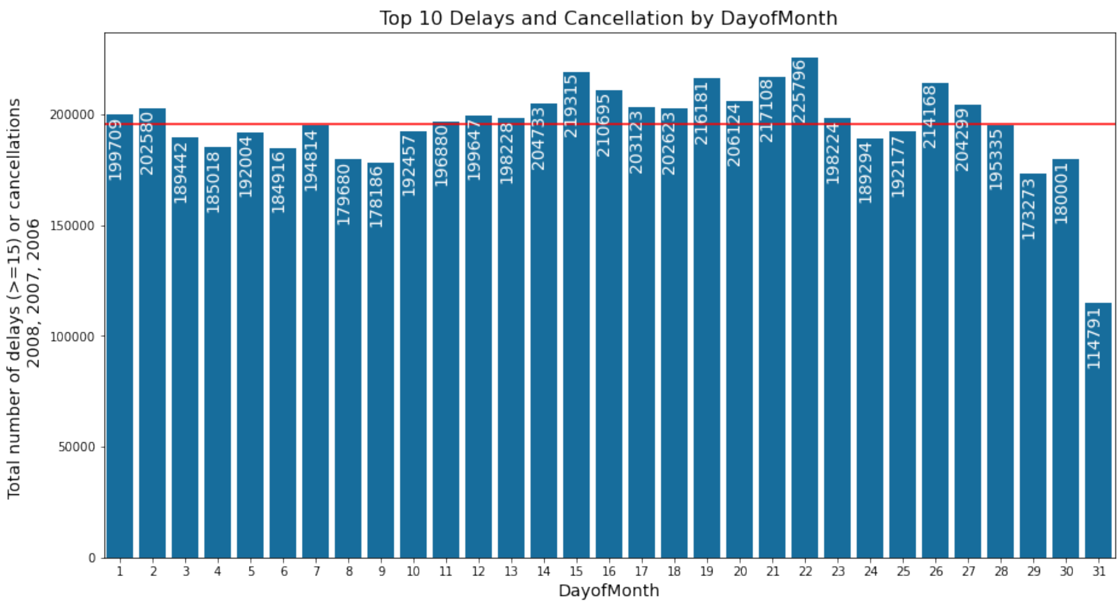
[24]:
```
# Top 10 Delays and Cancellation by DayofWeek
plot('DayOfWeek')
```



Friday had the most delays and cancellation over mean in **2008, 2007, 2006**

0.8 What are the Top 10 Delays and Cancellation by DayofMonth?

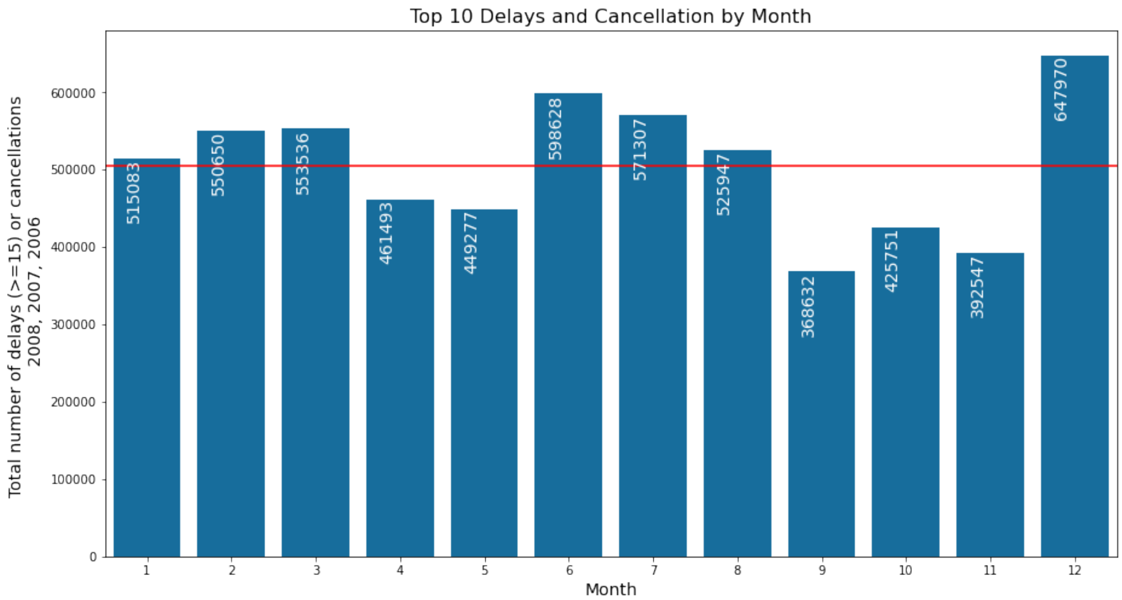
[25]:
```python
# Top 10 Delays and Cancellation by DayofMonth
plot('DayofMonth')
```



the **22th** day of month had the most delays and cancellation over mean in **2008, 2007, 2006**

0.9 What are the Top 10 Delays and Cancellation by Month?

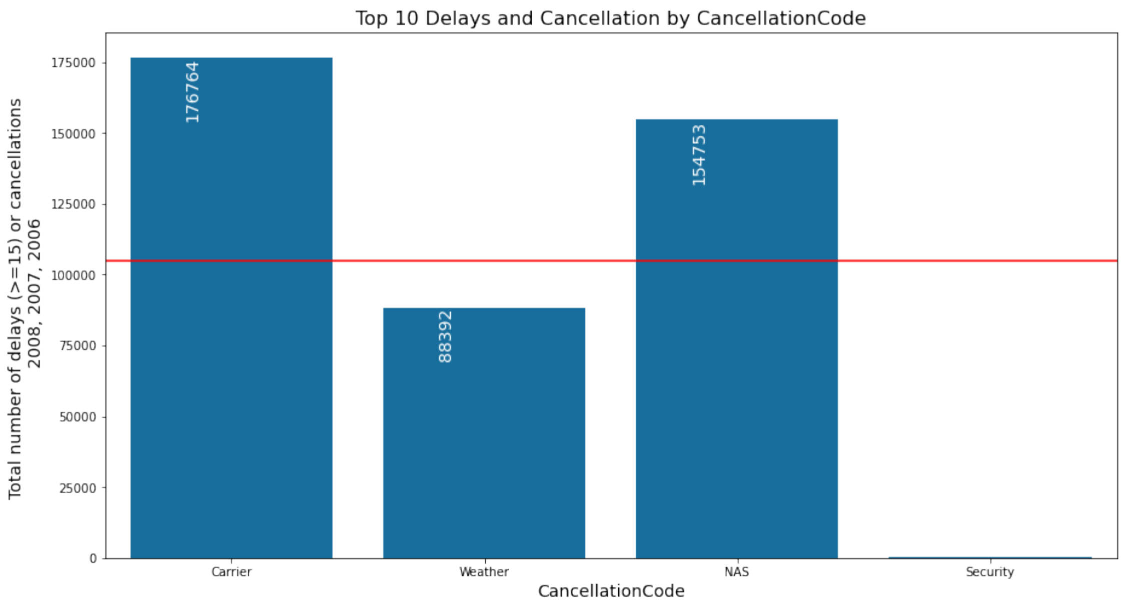
[26]:
```python
# Top 10 Delays and Cancellation by Month
plot('Month')
```



December had the most delays and cancellation over mean in **2008, 2007, 2006**

0.10 What are the Top 10 Delays and Cancellation by Cancellation Code?

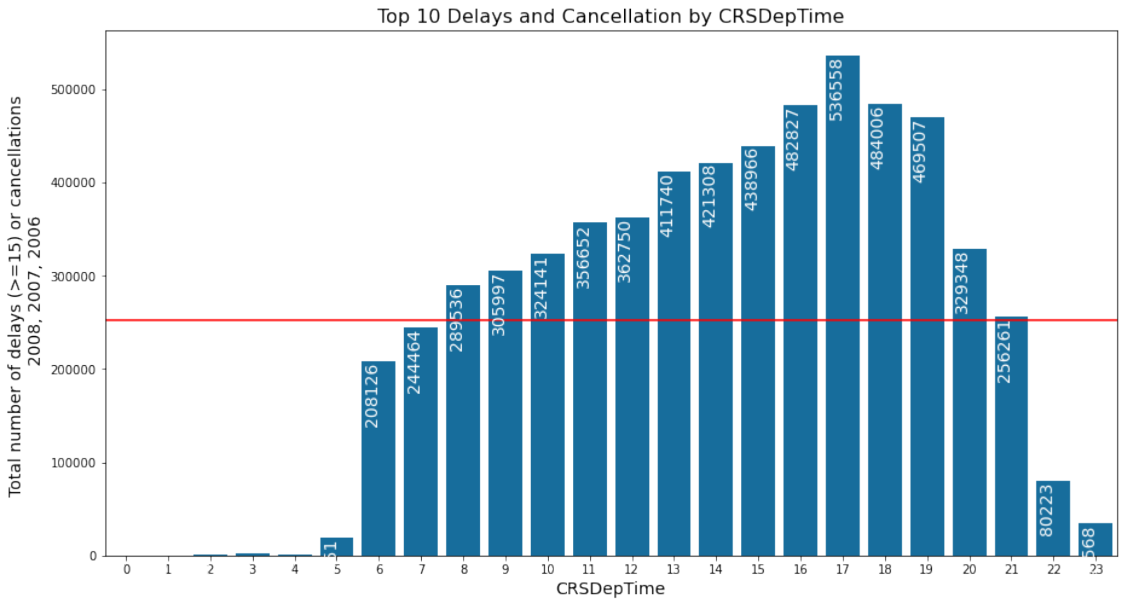
[27]:
```
# Top 10 Delays and Cancellation by Cancellation Code
plot('CancellationCode')
```



Cancellation by Carrier and National Aviation System are the top two reasons for cancellations over mean

0.11 What are the Top 10 Delays and Cancellation by Scheduled Departure time?

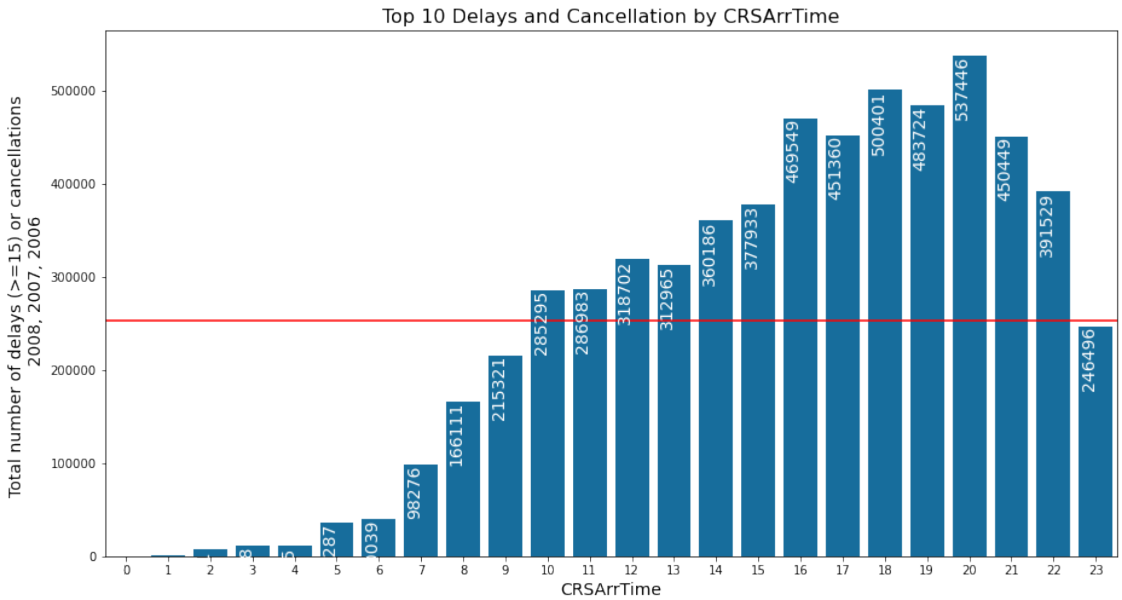
[28]:
```
# Top 10 Delays and Cancellation by Scheduled Departure time
plot('CRSDepTime')
```



Scheduled departure time Maximum delay or cancellation is at 17:00

0.12 What are the Top 10 Delays and Cancellation by Scheduled Arrival time?

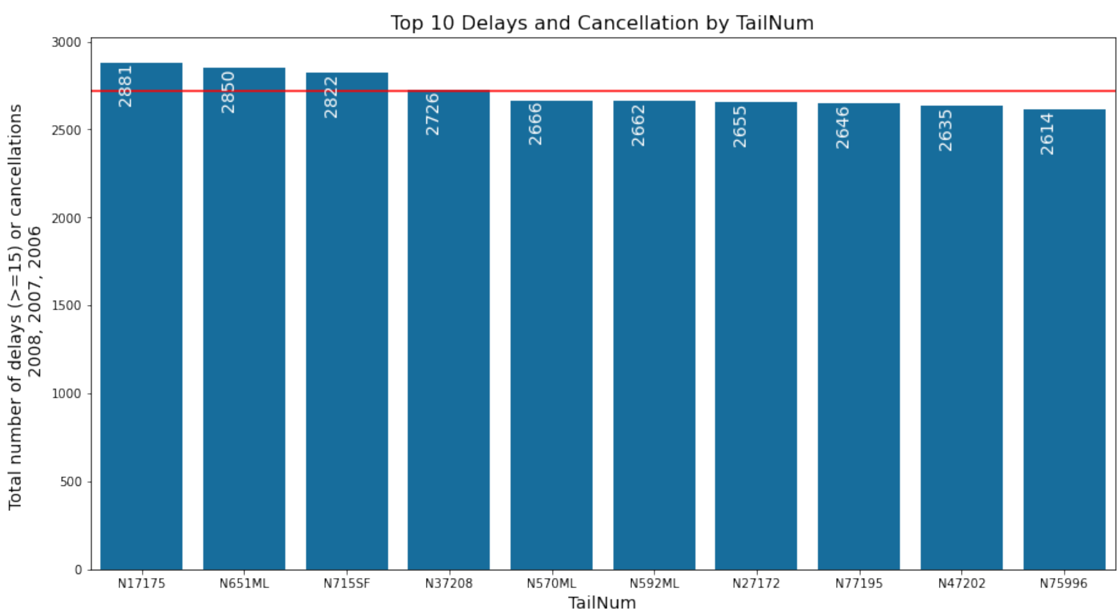
[29]:
```
# Top 10 Delays and Cancellation by Scheduled Arrival time
plot('CRSArrTime')
```



Scheduled Arrival time Maximum delay or cancellation is at 20:00

0.13 What are the Top 10 Delays and Cancellation by Tail Number?

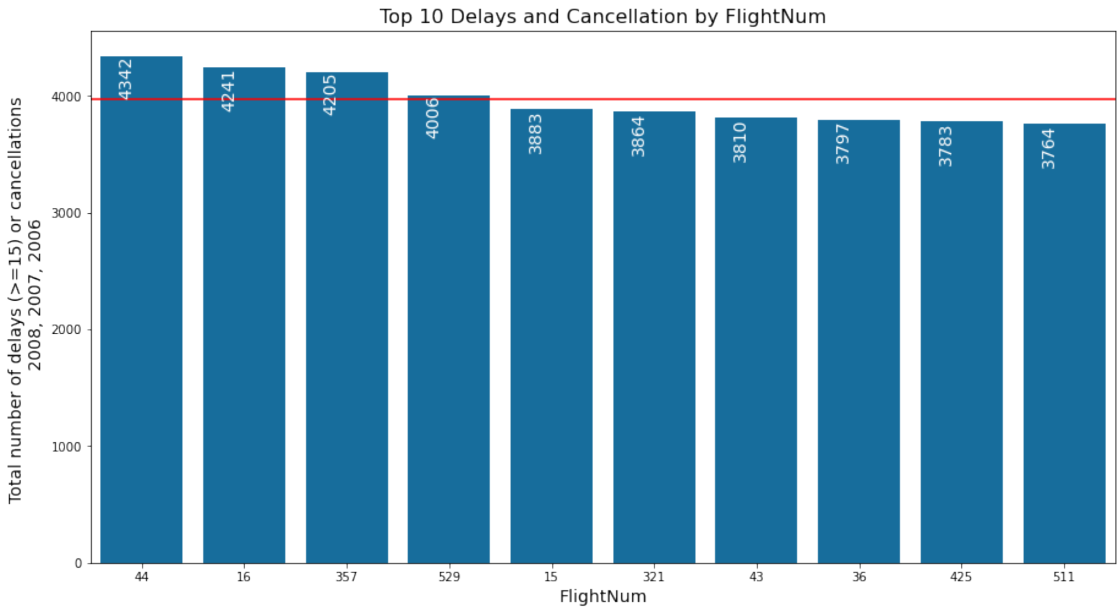
[30]:
```python
# Top 10 Delays and Cancellation by Tail Number
plot('TailNum')
```



Plans with tail numbers : **N17175**, **N651ML**, **N715SF**, had the most delays and cancellation

0.14 What are the Top 10 Delays and Cancellation by Flight Number?

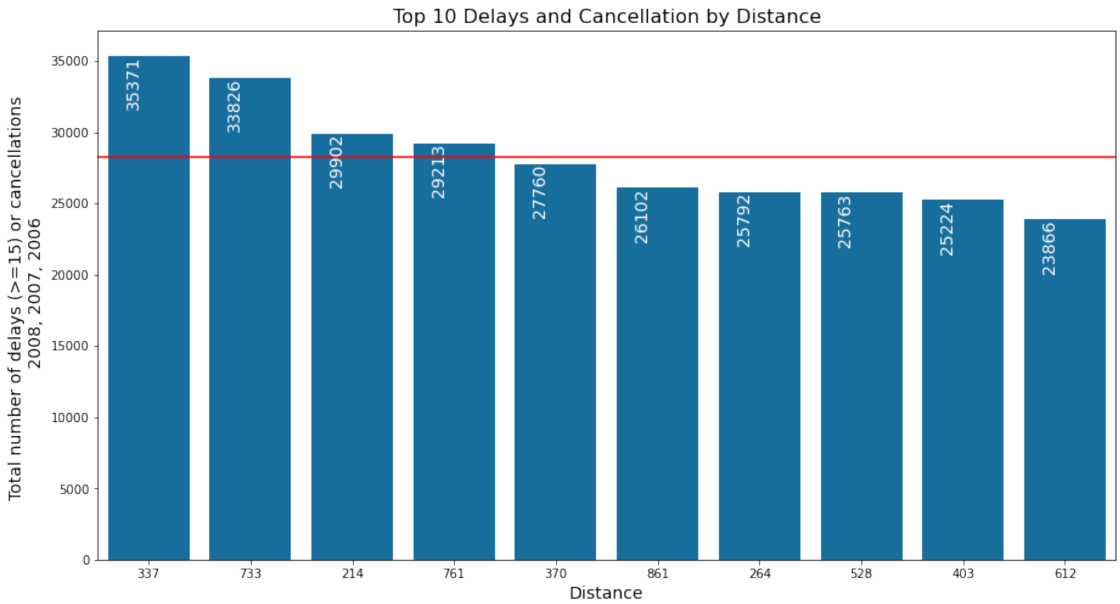
[31]:
```python
# Top 10 Delays and Cancellation by Flight Number
plot('FlightNum')
```



Flights number **44** , **16**, **357** had the most delays

0.15 What are the Top 10 Delays and Cancellation by Distance?

[32]:
```python
# Top 10 Delays and Cancellation by Distance
plot('Distance')
```



Distance = **337 miles** had the most delays and cancellation

0.16 What are the Delays and Cancellation by Airtime?

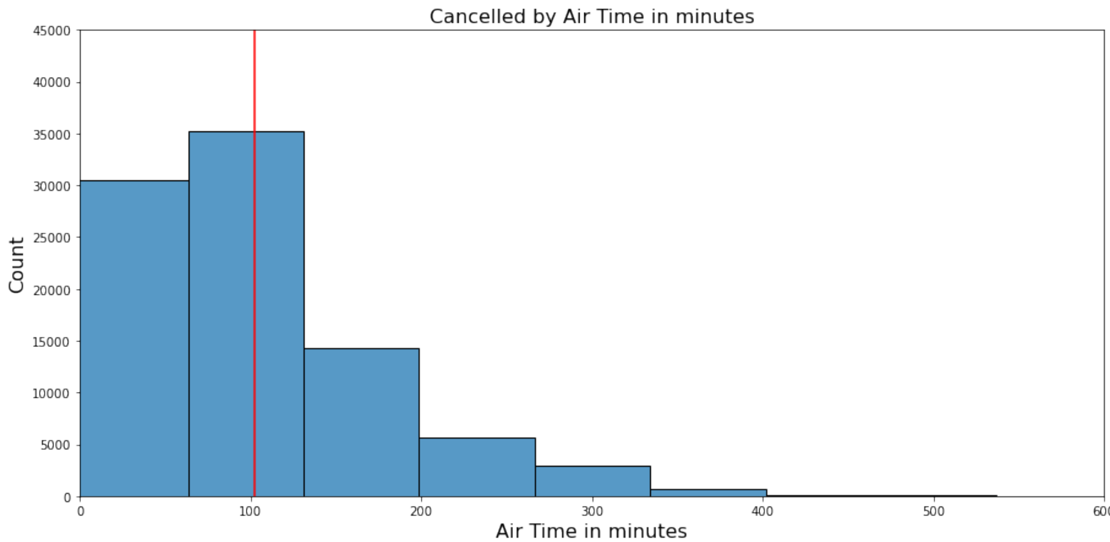
[33]:
```
# Delay or cancellation flight by Air Time
#define plot
fig, ax = plt.subplots(figsize=(15,7))

#generate data
ar_data = df[(df['Cancelled']== 0) & (df['AirTime']> 0)]
sb.histplot(data=df, x = 'AirTime',bins=50, stat = "frequency")

#set title and axis
plt.title('Cancelled by Air Time in minutes', fontsize=16);
plt.xlabel('Air Time in minutes', fontsize=16);
plt.ylabel('Count', fontsize=16);

#plot mean line
plt.axvline(x=df.AirTime.mean(), c='red')
plt.axis([0,600, 0, 45000])

#display plot
plt.show()
```



Airtime on short flights of **100 minutes** or less has the Greatest cancelled flights

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

The data analysis showed several types of distribution that correspond to the reality of canceled or delayed flights. ### 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?

This data does not contain outliers and does not need any changes

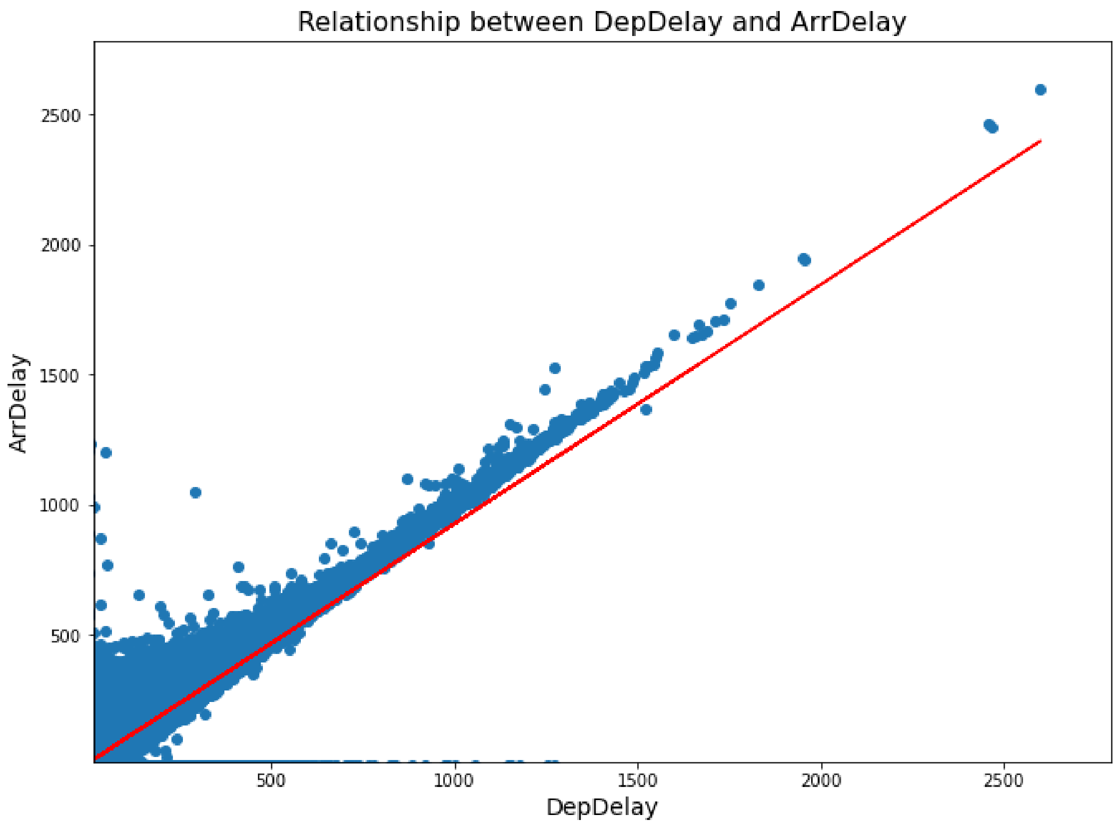
Bivariate Exploration

[34]:
```
# Define function to plot scatterplot to show relationship between two␣
↪variables
def relation(i,z):
    # Set plot size
    f,ax = plt.subplots(figsize=(11, 8));
    # Set x,y
    x = np.array(df[i])
    y = np.array(df[z])
    # plot data
    plt.scatter(x,y)
    # plot line regression
    #obtain m (slope) and b(intercept) of linear regression line
    m, b = np.polyfit(x, y, 1)
    #add linear regression line to scatterplot
    plt.plot(x, m*x + b,"r-")
    #Set Title
```

```
    titl = 'Relationship between '+ i +' and ' + z
    plt.title(titl, fontsize=16);
    #Set labels
    plt.xlabel(i, fontsize=14);
    plt.ylabel(z, fontsize=14);
    # Focus on the delays >=15
    plt.xlim(14,);
    plt.ylim(14,);
    plt.show();
```

0.17 What is the relationship between DepDelay' and 'ArrDelay'?

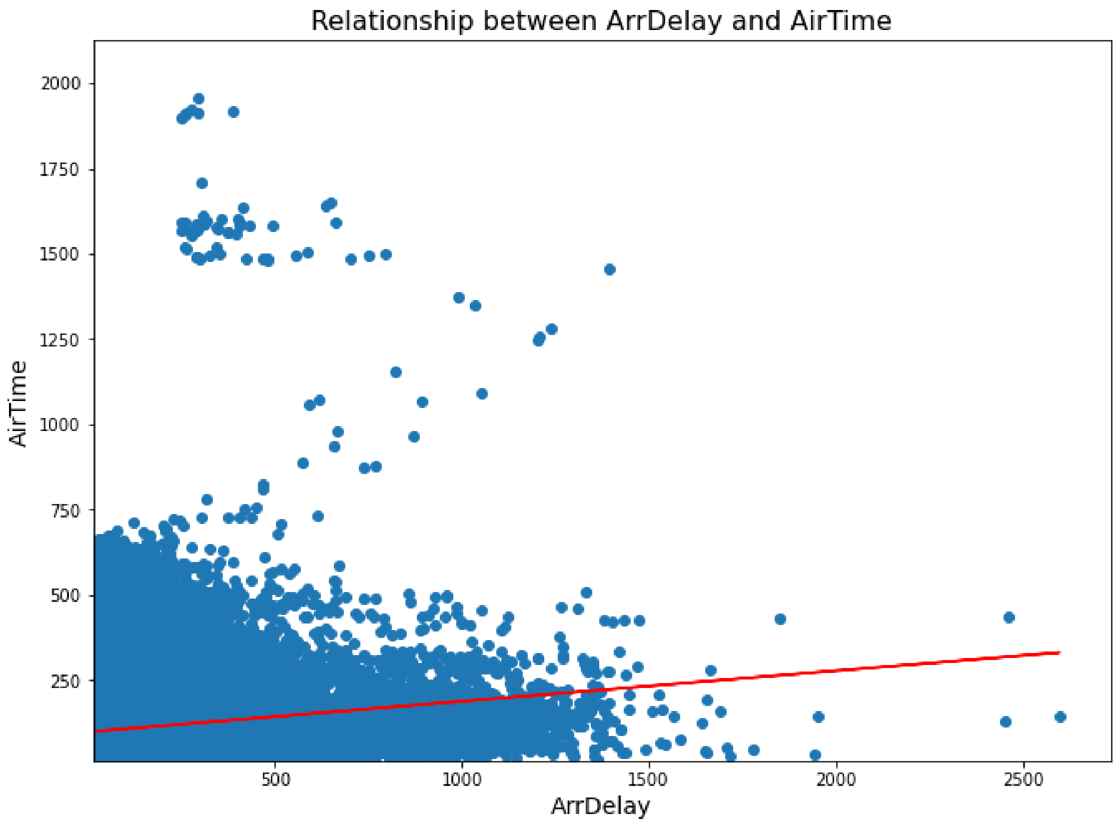
[35]:
```
# Plot Relation between 'DepDelay' and 'ArrDelay'
relation('DepDelay','ArrDelay')
```



There is a strong relationship between 'DepDelay' and 'ArrDelay'

0.18 What is the relationship between 'ArrDelay' and 'AirTime'?

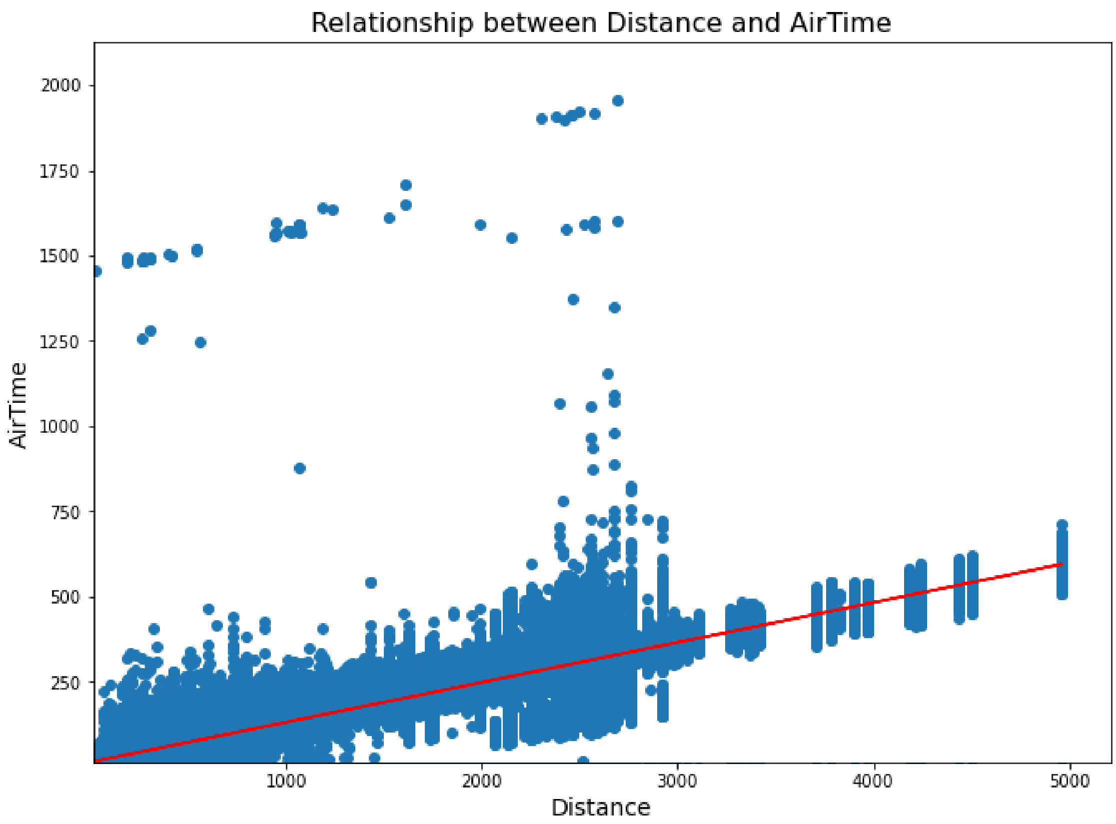
[36]:
```
# Plot Relation between 'ArrDelay' and 'AirTime'
relation('ArrDelay','AirTime')
```



There is a positive Relation between 'ArrDelay' and 'AirTime'

0.19 What is the relationship between 'Distance' and 'AirTime'?

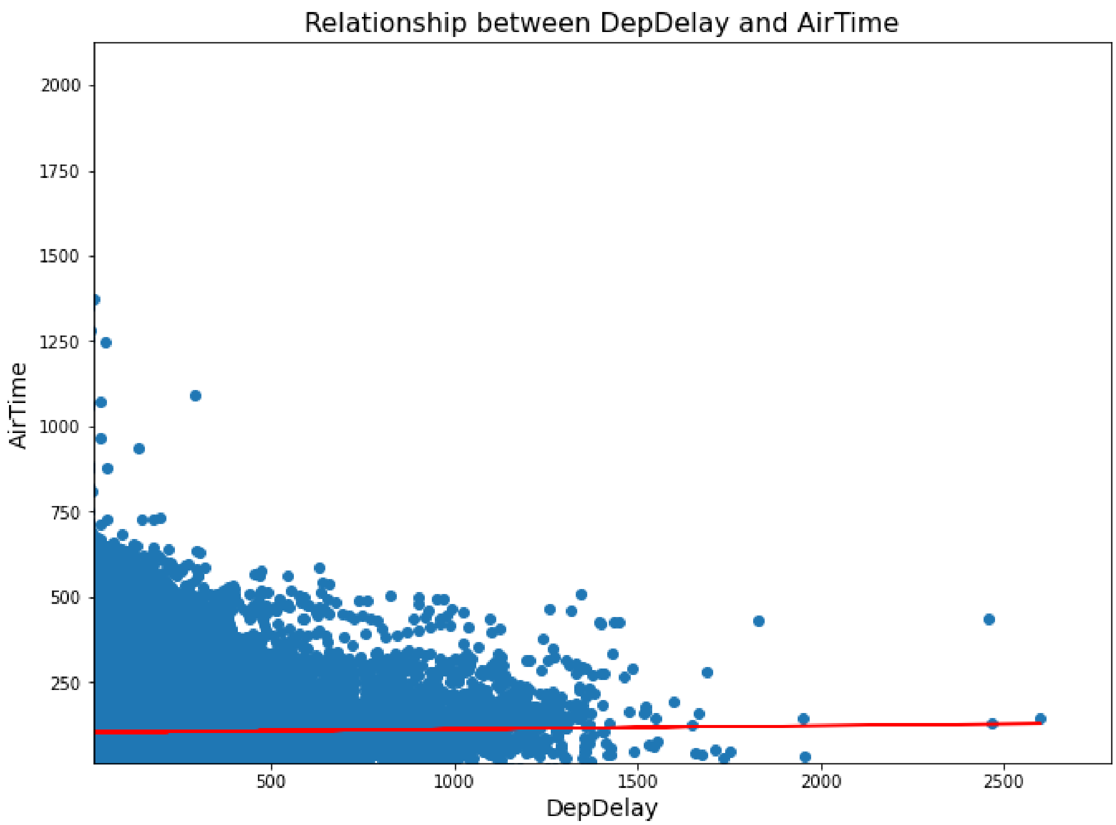
[37]:
```
# Plot Relation between 'Distance' and 'AirTime'
relation('Distance','AirTime')
```



There is a positive relationship between 'Distance' and 'AirTime'

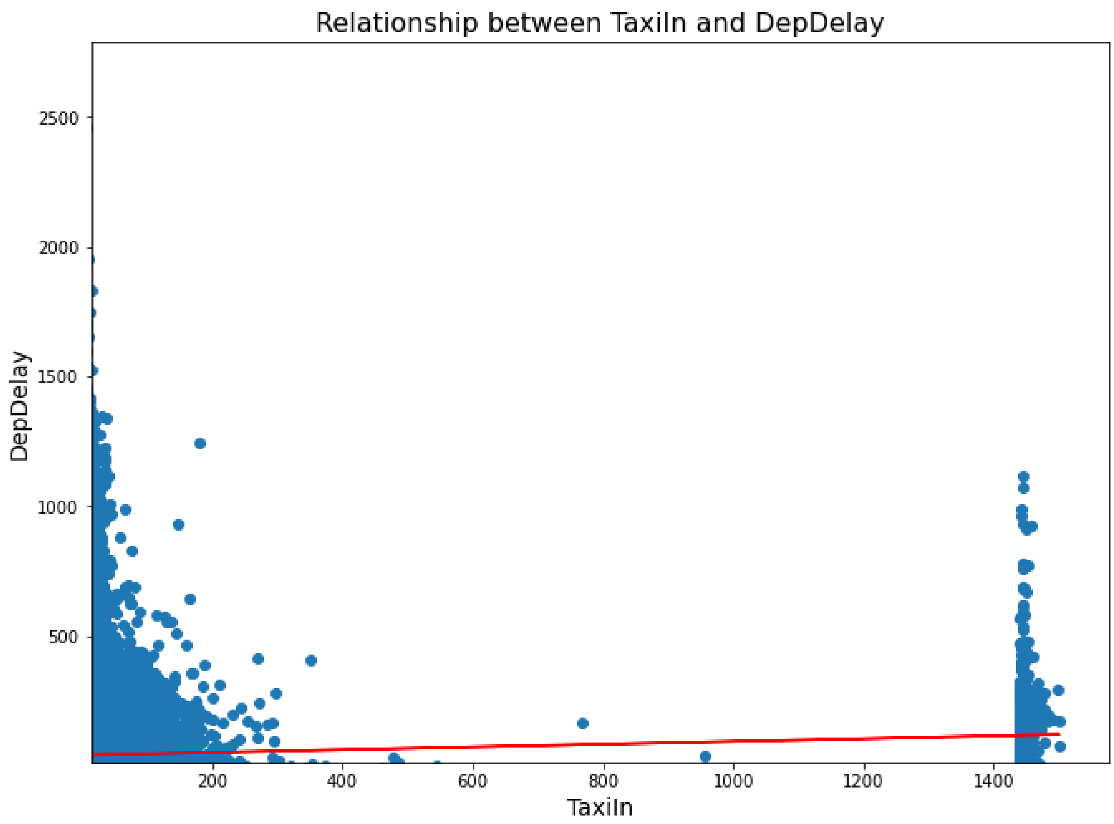
0.20 What is the relationship between 'DepDelay' and 'AirTime'?

[38]:
```
# Plot Relation between 'DepDelay' and 'AirTime'
relation('DepDelay','AirTime')
```



0.21 What is the relationship between 'TaxiIn' and 'DepDelay'?

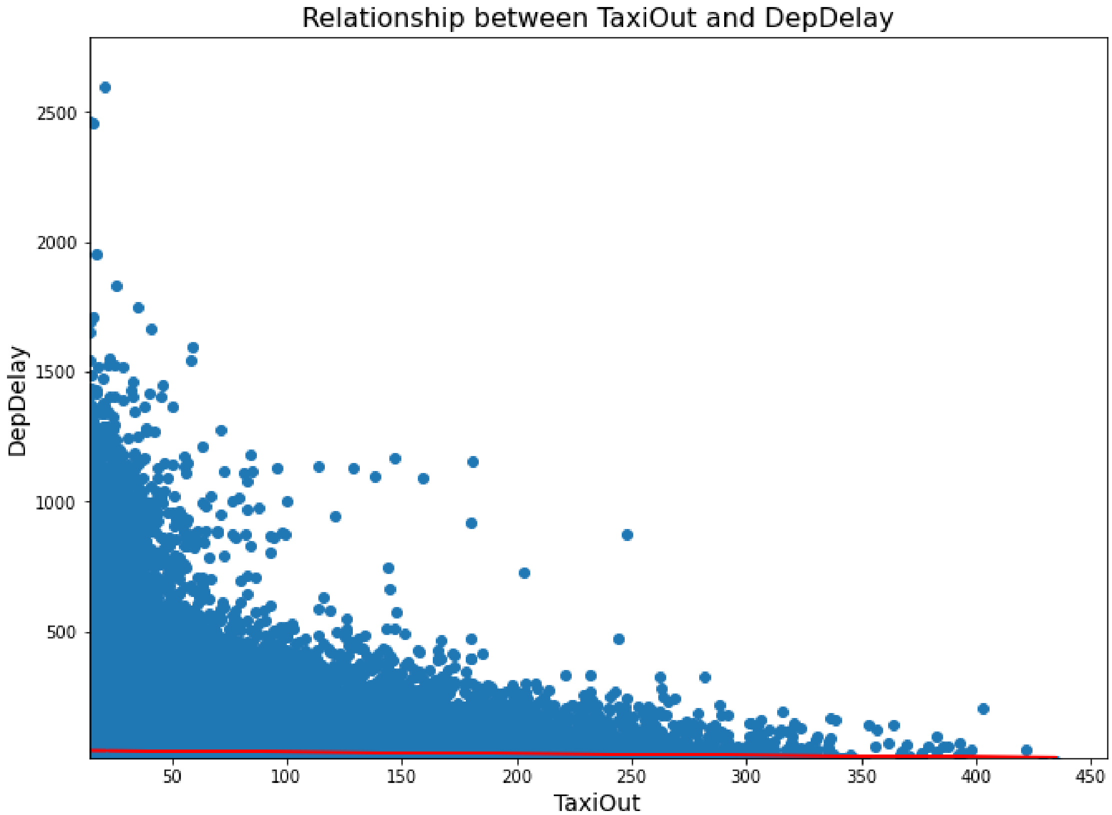
[39]:
```
# Plot Relation between 'TaxiIn' and 'DepDelay'
relation('TaxiIn','DepDelay')
```



There is a relationship between 'TaxiIn' and 'DepDelay'

0.22 What is the relationship between 'TaxiOut' and 'DepDelay'?

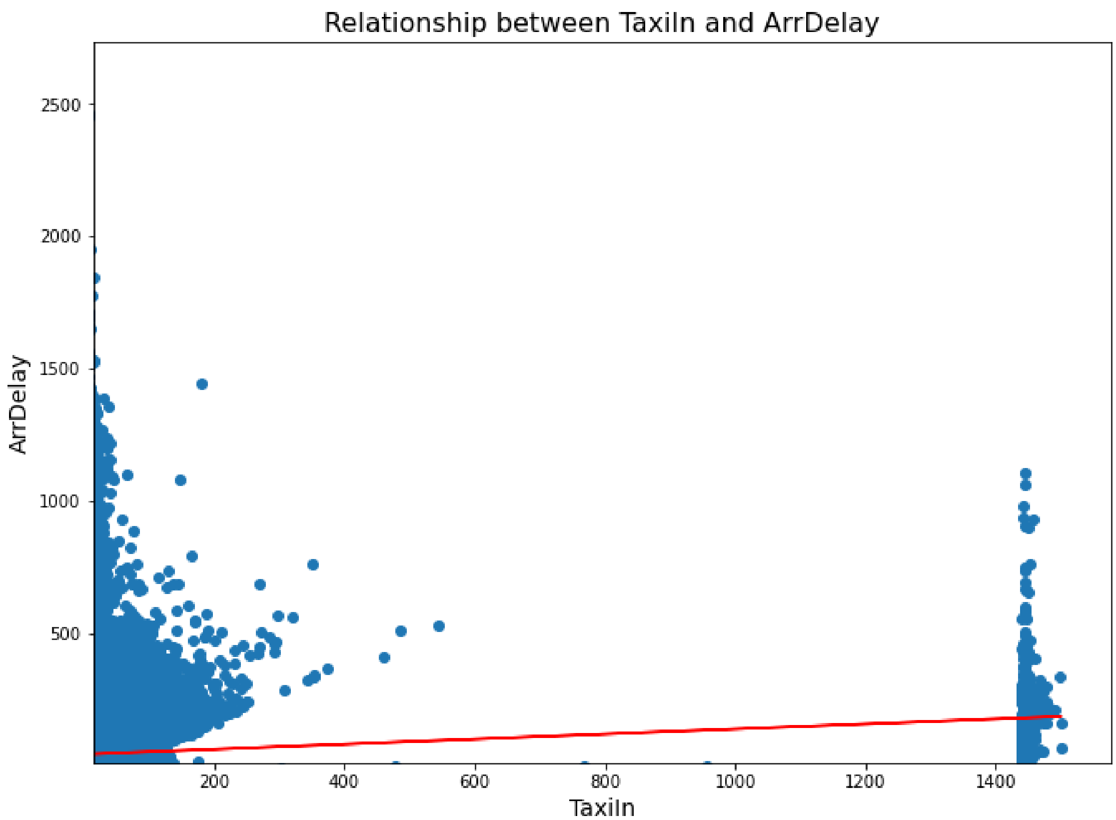
[40]:
```
# Plot Relation between 'TaxiOut' and 'DepDelay'
relation('TaxiOut','DepDelay')
```



There is Inverse relationship between ‘TaxiOut’ and ‘DepDelay’ since the linear regression had negative slop

0.23 What is the relationship between ‘TaxiIn’ and ‘ArrDelay’?

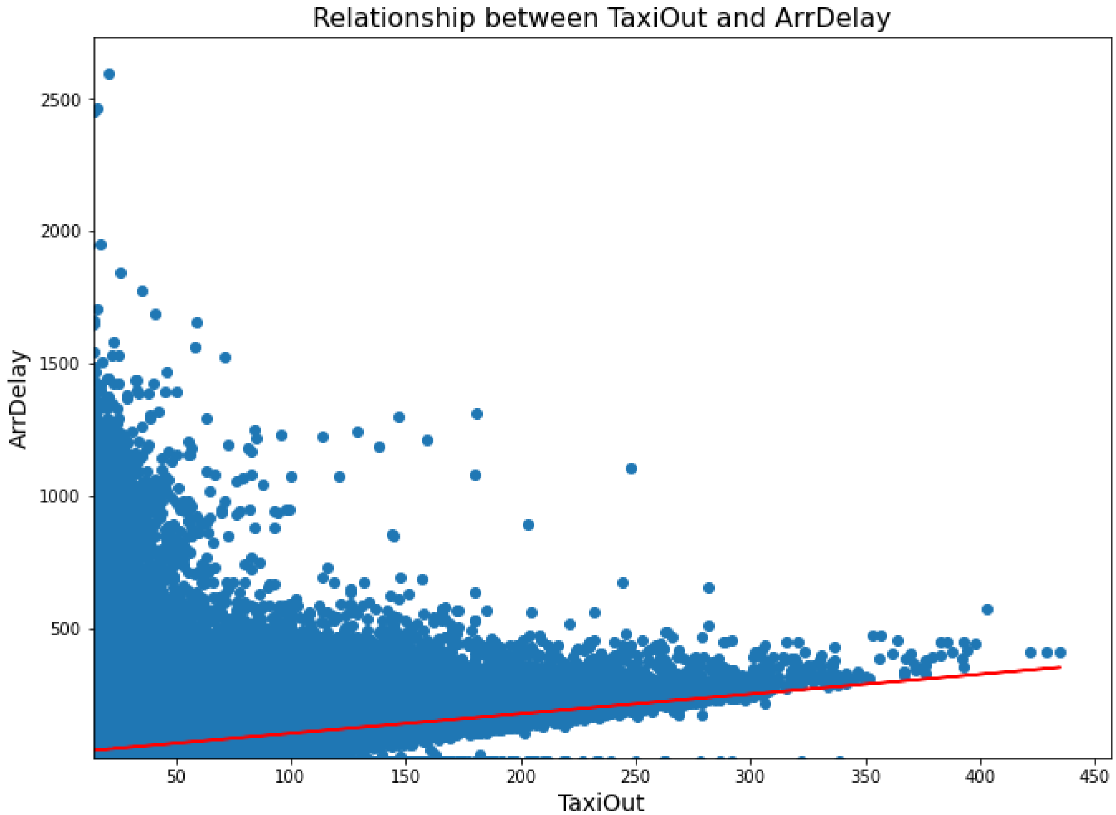
[41]:
```
# Plot Relation between 'TaxiIn' and 'ArrDelay'
relation('TaxiIn', 'ArrDelay')
```



There is Positive relationShip between 'TaxiIn' and 'ArrDelay'

0.24 What is the relationship between 'TaxiOut' and 'ArrDelay'?

[42]:
```
# Plot Relation between 'TaxiOut' and 'ArrDelay'
relation('TaxiOut','ArrDelay')
```



There is positive Relationship between 'TaxiIn' and 'ArrDelay'

0.24.1 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?

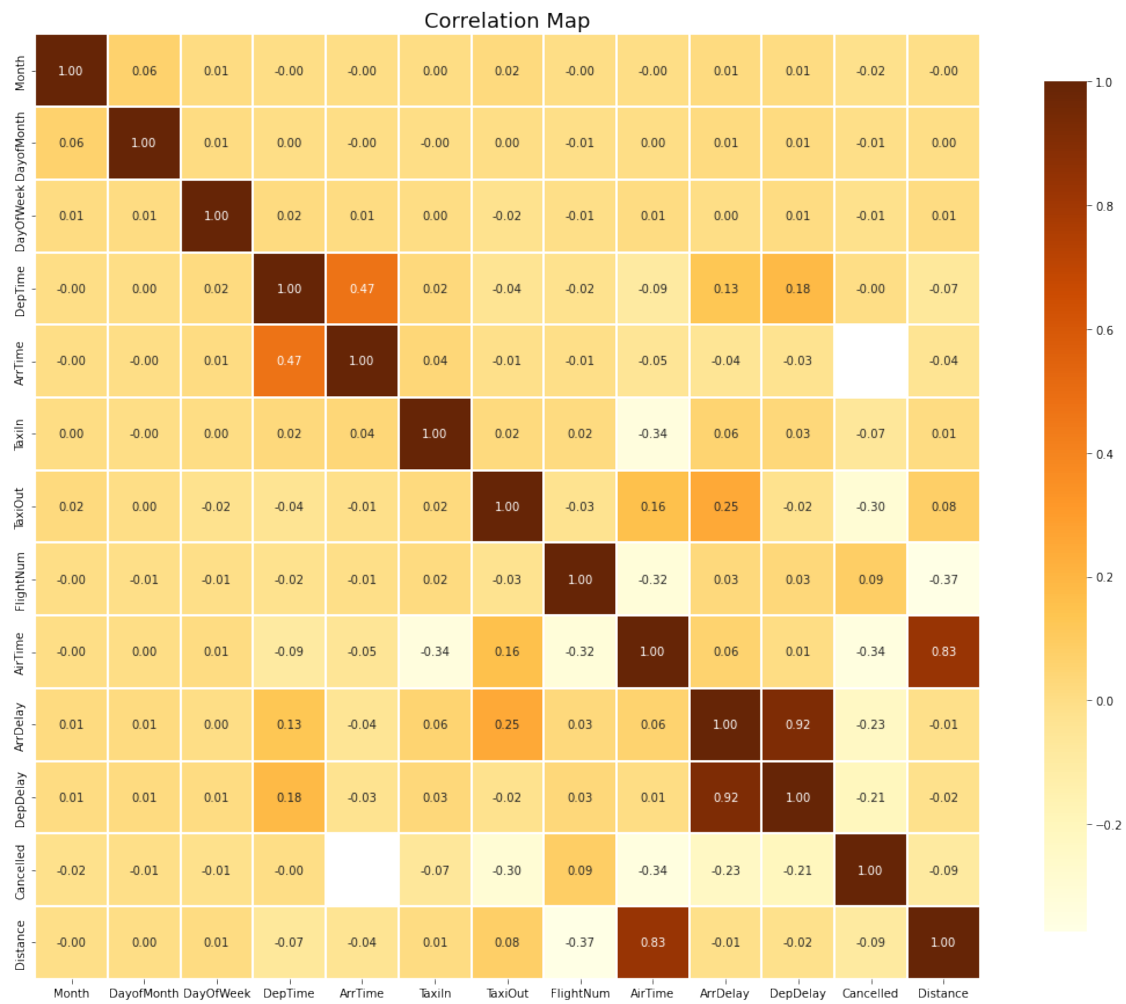
- There is positive relationship between: . 'DepDelay' and 'ArrDelay' . 'TaxiOut','ArrDelay' . 'Distance','AirTime' . 'ArrDelay','AirTime' . 'TaxiIn' and 'ArrDelay'
- There is Inverse relationship between 'TaxiOut' and 'DepDelay' ### Did you observe any interesting relationships between the other features (not the main feature(s) of interest)? Air time less than 1500 minutes has both departure and arrival delays

Multivariate Exploration

0.25 What is the relationship between all variables of interest?

[43]:
```
#plot correlation map between all variables of interest
# set size of plot
f,ax = plt.subplots(figsize=(20, 15));
# Define plot of all interesting variables
```

```python
sb.heatmap(df[['Month', 'DayofMonth', 'DayOfWeek', 'DepTime', 'ArrTime','TaxiIn',
                'TaxiOut','UniqueCarrier', 'FlightNum', 'TailNum','AirTime', 'ArrDelay',
                'DepDelay', 'Origin', 'Dest', 'Cancelled', 'Distance']].corr(),
           cmap="YlOrBr", square=True, annot=True, fmt= '.2f',ax=ax,
           linewidth=0.3, cbar_kws={"shrink": 0.9});
# set title
plt.title('Correlation Map', fontsize=18);
```



Correlation Map: there is a very strong relationship between: . 'DepDelay' and 'ArrDelay' . 'Distance','AirTime'

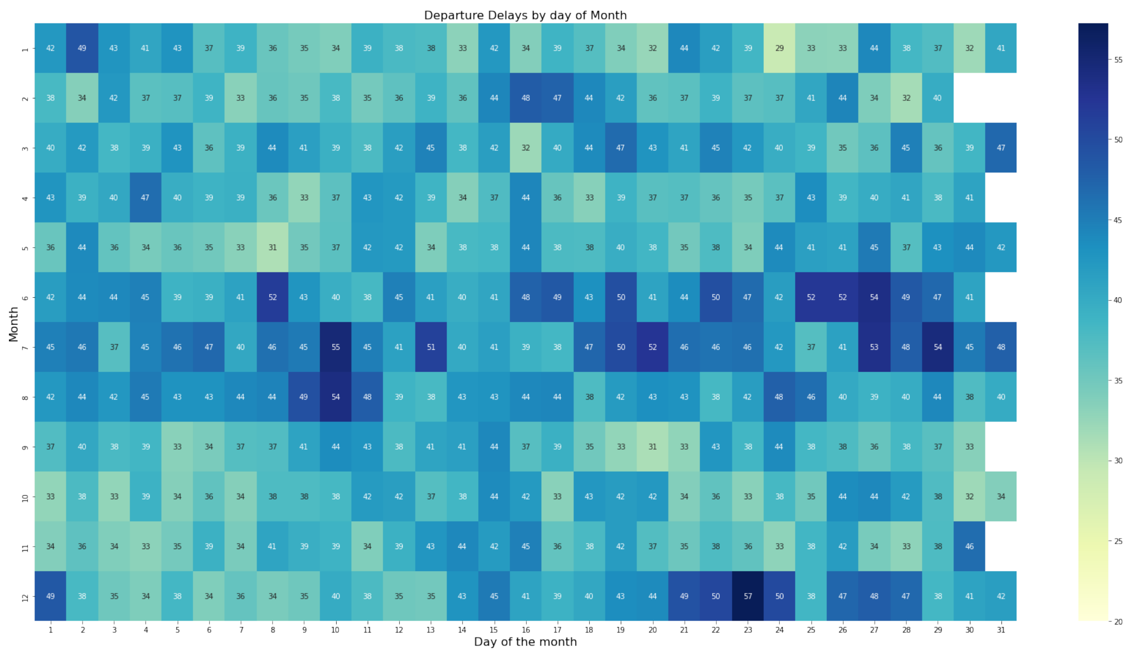
0.26 What is the Day of Month has the most Departure Delays?

[44]:
```
#pivot variables of interest
pl = df.pivot_table(index='Month',columns='DayofMonth', values='DepDelay',␣
↪aggfunc='mean')

#generate plot
plt.figure(figsize=(30,15));
sb.heatmap(pl, annot=True, cmap='YlGnBu', vmin=20);

#set title and axis

plt.title('Departure Delays by day of Month', fontsize=16);
plt.xlabel('Day of the month', fontsize=16);
plt.ylabel('Month', fontsize=16);
```



Departure Delays by day of Month: 23th December had the highest Average of Departure delays

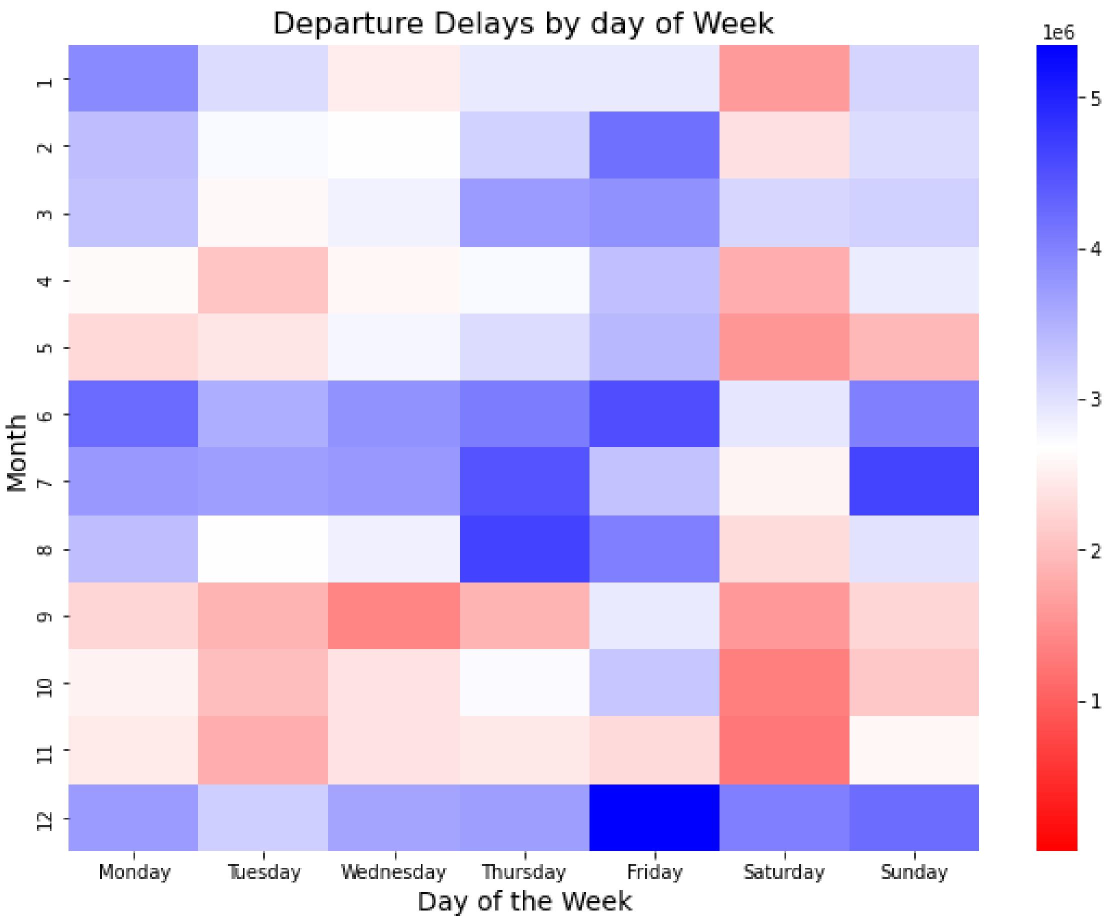
0.27 What is the Day of Week has the most Departure Delays?

[45]:
```
#pivot variables of interest
pl = df.pivot_table(index='Month',columns='DayOfWeek', values='DepDelay',␣
↪aggfunc='sum')

#generate plot
```

```
plt.figure(figsize=(11,8));
graph = sb.heatmap(pl, cmap='bwr_r', vmin=15);

#set title and axis
week_day = ['Monday','Tuesday','Wednesday','Thursday','Friday',␣
 ↪'Saturday','Sunday']
graph.set_xticklabels(week_day);
#SET Title
plt.title('Departure Delays by day of Week', fontsize=16);
plt.xlabel('Day of the Week', fontsize=14);
plt.ylabel('Month', fontsize=14);
```



Departure Delays by day of Week: Friday in December is the day that had the highest Average of Departure delays

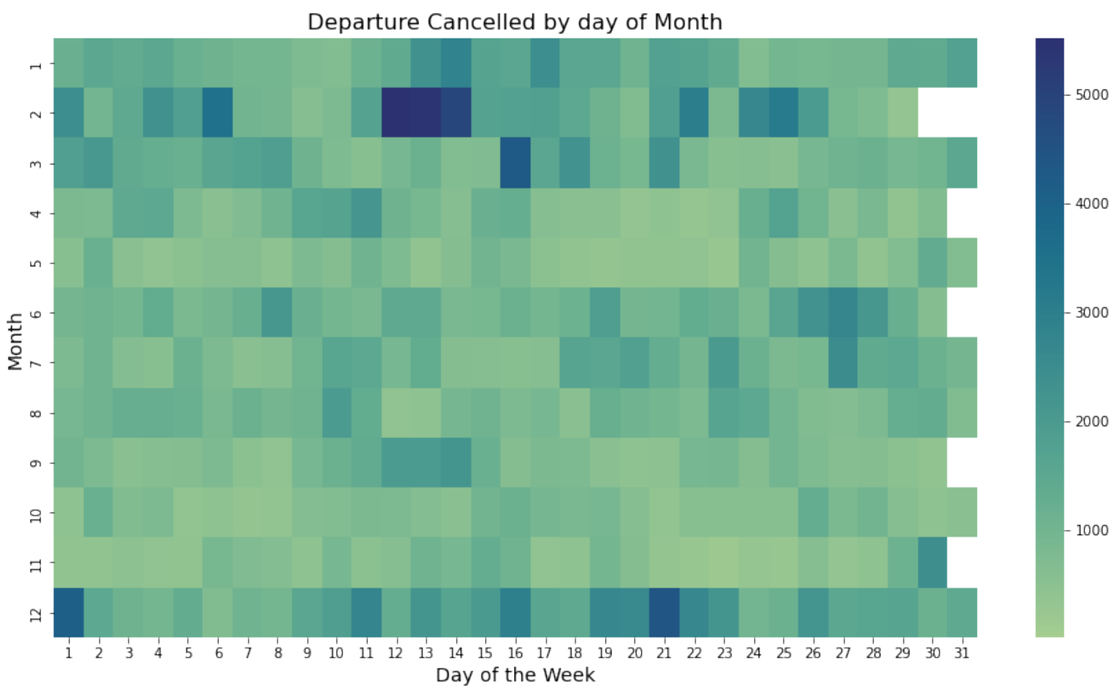
0.28 What is the Day of Month has the most Cancelled Flights?

[46]:
```
#pivot variables of interest
pl = df.pivot_table(index='Month',columns='DayofMonth', values='Cancelled',␣
↪aggfunc='sum')

#generate plot
plt.figure(figsize=(15,8));
sb.heatmap(pl,cmap='crest', vmin=15);

#set title and axis

plt.title('Departure Cancelled by day of Month', fontsize=16);
plt.xlabel('Day of the Week', fontsize=14);
plt.ylabel('Month', fontsize=14);
```



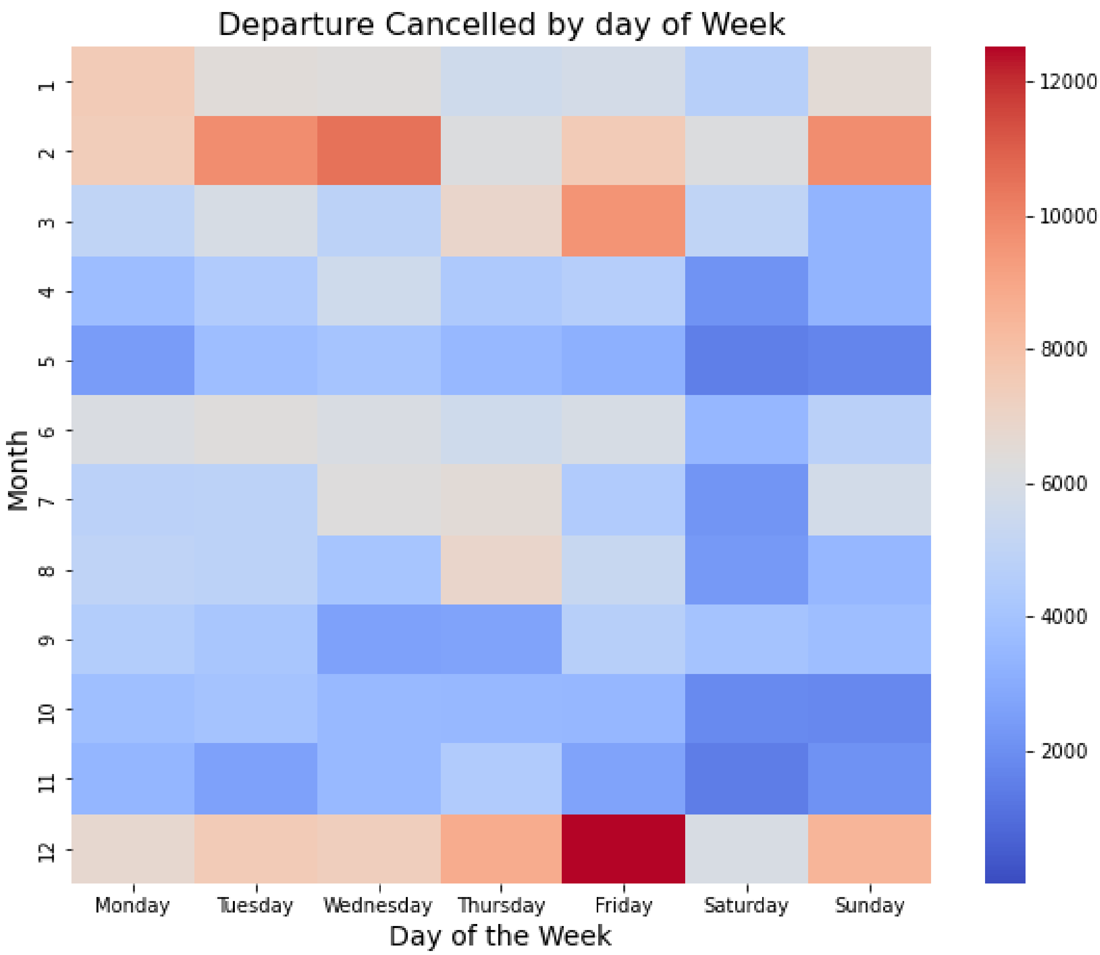
Departure Cancelled by day of Month: 12th, 13th, 14th February had the most Cancelled flights

0.29 What is the Day of Week has the most Cancelled Flights?

[47]:
```
#pivot variables of interest
pl = df.pivot_table(index='Month',columns='DayOfWeek', values='Cancelled',␣
↪aggfunc='sum')
```

```python
#generate plot
plt.figure(figsize=(10,8));
graph=sb.heatmap(pl, cmap='coolwarm', vmin=15);
week_day = ['Monday','Tuesday','Wednesday','Thursday','Friday',␣
↪'Saturday','Sunday']
graph.set_xticklabels(week_day);
#set title and axis

plt.title('Departure Cancelled by day of Week', fontsize=16);
plt.xlabel('Day of the Week', fontsize=14);
plt.ylabel('Month', fontsize=14);
```



Departure Cancelled by day of Week: Friday in December had the heighest average of cancelled flights

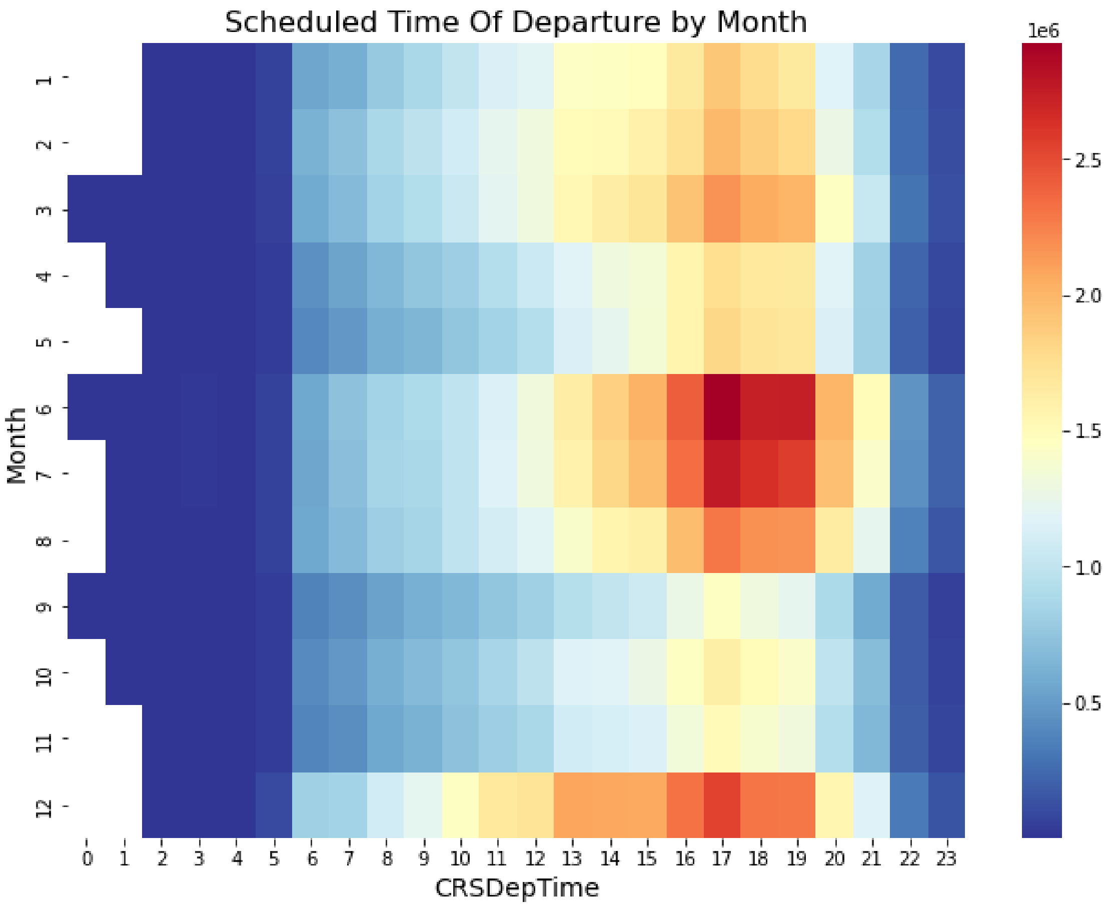
0.30 What is the Scheduled Time Of Departure has the most Departure Delay Flights?

[48]:
```
#pivot variables of interest
pl = df.pivot_table(index = 'Month', columns = 'CRSDepTime', values='DepDelay',␣
↪aggfunc='sum')

#generate plot
plt.figure(figsize=(11,8));
sb.heatmap(pl, cmap='RdYlBu_r', vmin=15);

#set title and axis

plt.title('Scheduled Time Of Departure by Month', fontsize=16);
plt.xlabel('CRSDepTime', fontsize=14);
plt.ylabel('Month', fontsize=14);
```



Scheduled Time Of Departure by Month: in June at 17:00 had the heighest average of delayed flights

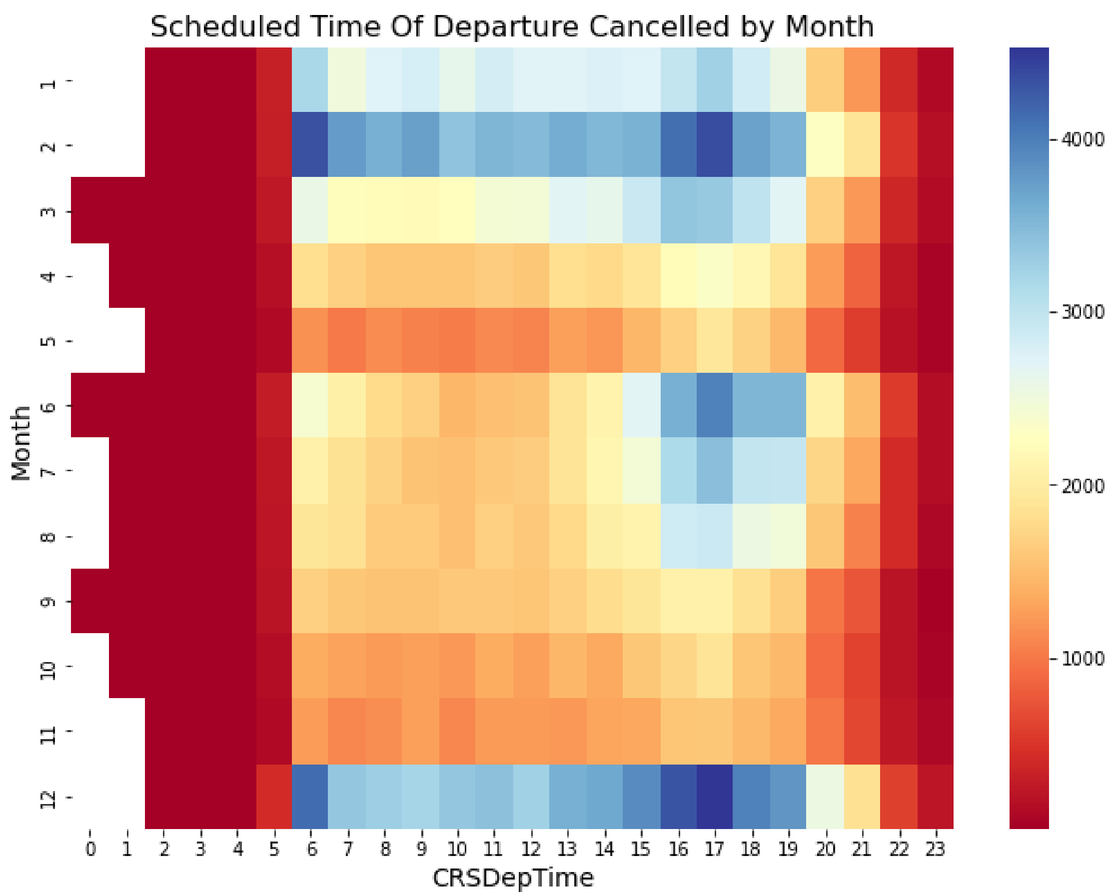
0.31 What is the Scheduled Time Of Departure has the most Departure Cancelled Flights?

[49]:
```
#pivot variables of interest
pl = df.pivot_table(index = 'Month', columns = 'CRSDepTime',␣
↪values='Cancelled', aggfunc='sum')

#generate plot
plt.figure(figsize=(11,8));
sb.heatmap(pl, cmap='RdYlBu', vmin=15);

#set title and axis

plt.title('Scheduled Time Of Departure Cancelled by Month', fontsize=16);
plt.xlabel('CRSDepTime', fontsize=14);
plt.ylabel('Month', fontsize=14);
```



Scheduled Time Of Departure Cancelled by Month: 17:00 in February

0.31.1 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?

Sunday in july is the day that had the highest Average of Departure delays

0.31.2 Were there any interesting or surprising interactions between features?

12,13,14 February had the highest average of Cancelled flights (Mayebe because the Valentine's Day)

Conclusions > These data are the data of the last three years (2008, 2007, 2006) from the data that were submitted, which were combined and analyzed, and the results of the analysis were divided into

0.31.3 Univariate Exploration Result:

Southwest Airlines , **American Airlines** and **Envoy Air**: had the most delays and cancellation over mean by Carrier in **2008**, **2007**, **2006 Chicago O'Hare International Airport (ORD)** , **Atlanta Airport (ATL)** and **Dallas/Ft Worth Intl(DFW)**: had the most delays and cancellation over mean by Origin in **2008, 2007, 2006 Chicago O'Hare International Airport (ORD)** , **Atlanta Airport (ATL)** and **Dallas/Ft Worth Intl(DFW)**: had the most delays and cancellation over mean by Dest in **2008, 2007, 2006 Friday** had the most delays and cancellation over mean in **2008, 2007, 2006** the **22th** day of month had the most delays and cancellation over mean in **2008, 2007, 2006 December** had the most delays and cancellation over mean in **2008, 2007, 2006** Cancellation by Carrier and National Aviation System are the top two reasons for cancellations over mean Scheduled departure time Maximum delay or cancellation is at 17:00 Scheduled Arrival time Maximum delay or cancellation is at 20:00 Plans with tail numbers : **N17175**, **N651ML**, **N715SF**, had the most delays and cancellation Flights number **44** , **16**, **357** had the most delays Distance = **337 miles** had the most delays and cancellation Airtime on short flights of **100 minutes** or less has the Greatest cancelled flights

0.31.4 Bivariate Exploration Results:

There is a strong relationship between 'DepDelay' and 'ArrDelay' There is a positive Relation between 'ArrDelay' and 'AirTime' There is a positive relationship between 'Distance' and 'AirTime' There is a relationship between 'TaxiIn' and 'DepDelay' There is Inverse relationship between 'TaxiOut' and 'DepDelay' since the linear regression had negative slop There is Positive relationShip between 'TaxiIn' and 'ArrDelay' There is positive Relationship between 'TaxiIn' and 'ArrDelay'

0.31.5 Multivariate Exploration results:

Correlation Map: there is a very strong relationship between: . 'DepDelay' and 'ArrDelay' . 'Distance','AirTime' Departure Delays by day of Month: 23th December had the highest Average of Departure delays Departure Delays by day of Week: Friday in December is the day that had the highest Average of Departure delays Departure Cancelled by day of Month: 12th, 13th, 14th February had the most Cancelled flights

Departure Cancelled by day of Week: Friday in December had the heighest average of cancelled flights Scheduled Time Of Departure by Month: in June at 17:00 had the heighest average of delayed flights Scheduled Time Of Departure Cancelled by Month: 17:00 in February

Sources: - https://knowledge.udacity.com/questions/523432

- https://knowledge.udacity.com/questions/412638
- https://seaborn.pydata.org/generated/seaborn.heatmap.html
- http://www.aiandhumans.com/papers/RosenthalRojas_LDAV17.pdf
- https://www.statology.org/scatterplot-with-regression-line-python/

Go up