Airline Data Exploration

by (Abeer Mousa)

Research Questions

1. What percentage of flights are canceled or diverted out of the total number of flights?
2. Is there a relationship between the number of cancellations and the time of year (quarters) or days of the week?
3. How do the causes of cancellations vary by quarter and day of the week?
4. What is the distribution pattern of flight delays?

Dataset Overview

The airline dataset provides a detailed record of flight information, compiled from multiple flights across different years. It includes a wide range of features related to various aspects of flight operations:

1. General Flight Information: This includes temporal details such as Year, Quarter, Month, Day of the Month, and Day of the Week.
2. Origin and Destination Information : Origin, OriginCityName, OriginState, OriginStateFips, OriginStateName, OriginWac: Various details about the origin airport, including codes, city name, state name, and geographic information.
3. Dest, DestCityName, DestState, DestStateFips, DestStateName, DestWac: Details about the destination airport, including codes, city name, state name, and geographic information.
4. Departure Information: CRSDepTime, DepTime, DepDelay, DepDelayMinutes, DepDel15, DepartureDelayGroups: Scheduled and actual departure times, along with various delay metrics, providing insights into how on-time or delayed departures were.
5. Arrival Information: CRSArrTime, ArrTime, ArrDelay, ArrDelayMinutes, ArrDel15, ArrivalDelayGroups: Scheduled and actual arrival times, along with various delay metrics, providing insights into arrival performance.
6. Cancelled, CancellationCode, Diverted: Indicators of whether a flight was canceled or diverted, along with codes explaining the reason for cancellation.
7. Flights, Distance, DistanceGroup: Metrics related to the flight count and the distance covered, with DistanceGroup likely categorizing flights into distance ranges.
8. Delay Breakdown: CarrierDelay, WeatherDelay, NASDelay, SecurityDelay, LateAircraftDelay: These columns break down the reasons for delays into categories like carrier issues, weather, air traffic control (NAS), security, and delays due to late aircraft.

Imports

```
# 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
import os

%matplotlib inline
```

Reusable functions - lib

Check unique Values

```
## Check unique Values
def check_unique_values(df, columns):
  print('Number of unique values per column')
  for column in columns:
    print('-----------------------------------')
    print('{} has {} unique
values:'.format(column,len(df[column].unique())))
    print('-----------------------------------')
    print(df[column].unique())
```

Check for Outliers

```
# Check for Outliers
def Check_for_Outliers(df, column, sort_by, upper_bound=None ):
    summaries = df.describe().loc[['mean', 'std']]
    if (upper_bound is None):
      upper_bound = summaries[column]['mean'] + summaries[column]
['std']
    lower_bound = summaries[column]['mean'] - summaries[column]['std']

    print('upper_bound = {} | lower_bound = {}'.format(upper_bound,
lower_bound))
    print('Count of outlier more than upper_bound =
{}'.format(len(df[df[column] > upper_bound ])))
    print('percentage  of outlier more than upper_bound = {}
%'.format(round((len(df[df[column] > upper_bound ])
                                                                        /
len(df))*100,2)))
```

Set Default Figure Size

```
def get_fig_size(fig_size=None):
    default_size = (10, 15)

    if fig_size is None:
        return default_size
    else:
        return fig_size
```

Pie chart

```
def pie_chart(df, column, title, labels=None):
   sorted_counts = df[column].value_counts()
   size = get_fig_size()
   if labels is None or len(labels) == 0:
      labels = sorted_counts.index

   wedges, texts, autotexts = plt.pie( sorted_counts, startangle = 90,
autopct='%1.1f%%', counterclock = False)
   plt.title(title, pad= 20)
   plt.axis('equal')
   plt.legend(wedges, labels, title="Categories", loc="center left",
bbox_to_anchor=(1, 0, 0.5, 1))
```

Bar Chart

```
def bar_chart(df, column, title, labels=None):

   # Return the Series having unique values
   x = df[column].unique()

   # Return the Series having frequency count of each unique value
   y = df[column].value_counts(sort=False)

   if(labels == None):
       bars = plt.bar(df[column].unique(), y)
   else:
       bars = plt.bar(labels, y)

   for bar in bars:
        height = bar.get_height()
        plt.text(bar.get_x() + bar.get_width() / 2, height,
f'{height}', ha='center', va='bottom')

   #plt.figure(figsize=get_fig_size())
   plt.ylabel('count')
   plt.title(title, pad= 20)
```

Two Histogram Plots

```
def two_hist_chart(df, columns, titles, xyLabels, bin_size=1):
  plt.figure(figsize = [20, 5])

  # histogram on left, example of too-large bin size
  # 1 row, 2 cols, subplot 1
  plt.subplot(1, 2, 1)
  bins = np.arange(-2, df[columns[0]].max()+bin_size, bin_size)
  plt.title(titles[0], pad= 20)
  plt.xlabel(xyLabels[0])
  plt.ylabel('Frequency')
```

```
  plt.hist(data = df, x = columns[0], bins = bins);


  # histogram on right, example of too-small bin size
  plt.subplot(1, 2, 2) # 1 row, 2 cols, subplot 2
  bins = np.arange(-2, df[columns[1]].max()+bin_size, bin_size)
  plt.title(titles[1], pad= 20)
  plt.xlabel(xyLabels[1])
  plt.ylabel('Frequency')
  plt.hist(data = df, x = columns[1], bins = bins);
```

Clustered Bar Charts

```
def clustered_bar_chart(df, value_column, class_column, title,
xyLabels):
  sns.countplot(data=df, x=value_column, hue=class_column)
  plt.legend(loc='upper right', bbox_to_anchor=(1.25, 1),
fontsize='small', title='')
  plt.xticks(ticks=np.arange(len(df[value_column].unique())) + 0.2,
labels=df[value_column].unique())
  plt.title(title)
  plt.xlabel(xyLabels[0])
  plt.ylabel(xyLabels[1])
```

Scatter plot

```
def Scatter_plot(df, columns, title, xyLabels):
  plt.scatter(data=df, x=columns[0], y=columns[1])
  plt.title(title)
  plt.xlabel(xyLabels[0])
  plt.ylabel(xyLabels[1])
```

Regression Plot

```
def regression_scatter_plot(df, columns, title, xyLabels):
  sns.regplot(data=df, x=columns[0], y=columns[1]);
  plt.title(title)
  plt.xlabel(xyLabels[0])
  plt.ylabel(xyLabels[1])
```

Box Plot

```
def box_plot(df, class_column, classes, value_column, title,
xyLabels):


   ax1 = sns.boxplot(data=df, x=class_column, y=value_column,
color='tab:blue')
   plt.xticks(rotation=15);
```

```
    plt.title(title)
    plt.xlabel(xyLabels[0])
    plt.ylabel(xyLabels[1])
    plt.ylim(ax1.get_ylim())
```

Heat Map

```
def heat_map(df, columns, title, xyLabels):
    # Specify bin edges
    # bins_x = np.arange(0.6, 7+0.3, 0.3)
    # bins_y = np.arange(12, 58+3, 3)

   plt.hist2d(data=df, x=columns[0], y=columns[1], cmin=1,
cmap='viridis_r' )
   plt.colorbar()
   plt.title(title)
   plt.xlabel(xyLabels[0])
   plt.ylabel(xyLabels[1]);
```

FacetGrid

```
def FacetGrid(df, value_column, class_column, bin_size, title,
xyLabels):

    bins = np.arange(-2, df[value_column].max() + bin_size, bin_size)
    g = sns.FacetGrid(data=df, col=class_column, col_wrap=2)


    g.map(plt.hist, value_column, bins=bins)
    g.set_axis_labels(xyLabels[0], xyLabels[1])

    plt.show()

# 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 sns
import os
```

Loading Data

```
print(os.getcwd())

C:\Users\User\Udacity\data_analysis\finalProject
```

Change the work directory

```
os.chdir('C:/Users/User/Udacity/data_analysis/finalProject')
```

Load Data

```
# load in the dataset into a pandas dataframe, print statistics
df =
pd.read_csv('c:/Users/User/Udacity/data_analysis/finalProject/airline_
2m/airline_2m.csv',encoding='ISO-8859-1')
```

```
C:\Users\User\AppData\Local\Temp\ipykernel_18896\1473759456.py:2:
DtypeWarning: Columns (69,76,77,84) have mixed types. Specify dtype
option on import or set low_memory=False.
  df =
pd.read_csv('c:/Users/User/Udacity/data_analysis/finalProject/airline_
2m/airline_2m.csv',encoding='ISO-8859-1')
```

Browsing Data

```
# high-level overview of data shape and composition
print(df.shape)
print(df.dtypes)
print(df.head(10))
```

```
(2000000, 109)
Year                      int64
Quarter                   int64
Month                     int64
DayofMonth                int64
DayOfWeek                 int64
                          ...
Div5WheelsOn            float64
Div5TotalGTime          float64
Div5LongestGTime        float64
Div5WheelsOff           float64
Div5TailNum             float64
Length: 109, dtype: object
   Year  Quarter  Month  DayofMonth  DayOfWeek  FlightDate
Reporting_Airline  \
0  1998        1      1           2          5  1998-01-02
NW
1  2009        2      5          28          4  2009-05-28
FL
2  2013        2      6          29          6  2013-06-29
MQ
3  2010        3      8          31          2  2010-08-31
DL
4  2006        1      1          15          7  2006-01-15
US
5  1995        4     11          29          3  1995-11-29
DL
6  2006        3      8           7          1  2006-08-07
CO
```

```
7  2019         2      6          11          2  2019-06-11
9E
8  2008         3      8           3          7  2008-08-03
YV
9  2018         1      2           8          4  2018-02-08
WN

   DOT_ID_Reporting_Airline IATA_CODE_Reporting_Airline
Tail_Number  ...  \
0                     19386                          NW
N297US  ...
1                     20437                          FL
N946AT  ...
2                     20398                          MQ
N665MQ  ...
3                     19790                          DL
N6705Y  ...
4                     20355                          US
N504AU  ...
5                     19790                          DL
N925DL  ...
6                     19704                          CO
N27724  ...
7                     20363                          9E
N927XJ  ...
8                     20378                          YV
N522LR  ...
9                     19393                          WN
N8688J  ...

   Div4WheelsOff  Div4TailNum  Div5Airport  Div5AirportID
Div5AirportSeqID  \
0            NaN          NaN          NaN            NaN
NaN
1            NaN          NaN          NaN            NaN
NaN
2            NaN          NaN          NaN            NaN
NaN
3            NaN          NaN          NaN            NaN
NaN
4            NaN          NaN          NaN            NaN
NaN
5            NaN          NaN          NaN            NaN
NaN
6            NaN          NaN          NaN            NaN
NaN
7            NaN          NaN          NaN            NaN
NaN
8            NaN          NaN          NaN            NaN
```

```
NaN
9            NaN          NaN          NaN            NaN
NaN

  Div5WheelsOn Div5TotalGTime  Div5LongestGTime Div5WheelsOff
Div5TailNum
0           NaN            NaN                NaN           NaN
NaN
1           NaN            NaN                NaN           NaN
NaN
2           NaN            NaN                NaN           NaN
NaN
3           NaN            NaN                NaN           NaN
NaN
4           NaN            NaN                NaN           NaN
NaN
5           NaN            NaN                NaN           NaN
NaN
6           NaN            NaN                NaN           NaN
NaN
7           NaN            NaN                NaN           NaN
NaN
8           NaN            NaN                NaN           NaN
NaN
9           NaN            NaN                NaN           NaN
NaN

[10 rows x 109 columns]
```

```
df.iloc[:,:20].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000000 entries, 0 to 1999999
Data columns (total 20 columns):
 #   Column                           Dtype
---  ------                           -----
 0   Year                             int64
 1   Quarter                          int64
 2   Month                            int64
 3   DayofMonth                       int64
 4   DayOfWeek                        int64
 5   FlightDate                       object
 6   Reporting_Airline                object
 7   DOT_ID_Reporting_Airline         int64
 8   IATA_CODE_Reporting_Airline      object
 9   Tail_Number                      object
 10  Flight_Number_Reporting_Airline  int64
 11  OriginAirportID                  int64
 12  OriginAirportSeqID               int64
 13  OriginCityMarketID               int64
```

```
 14  Origin                            object
 15  OriginCityName                    object
 16  OriginState                       object
 17  OriginStateFips                   float64
 18  OriginStateName                   object
 19  OriginWac                         int64
dtypes: float64(1), int64(11), object(8)
memory usage: 305.2+ MB
```

```
df.iloc[:, :20].head()
```

```
   Year  Quarter  Month  DayofMonth  DayOfWeek  FlightDate
Reporting_Airline  \
0  1998        1      1           2          5  1998-01-02
NW
1  2009        2      5          28          4  2009-05-28
FL
2  2013        2      6          29          6  2013-06-29
MQ
3  2010        3      8          31          2  2010-08-31
DL
4  2006        1      1          15          7  2006-01-15
US

   DOT_ID_Reporting_Airline IATA_CODE_Reporting_Airline Tail_Number  \
0                     19386                          NW      N297US
1                     20437                          FL      N946AT
2                     20398                          MQ      N665MQ
3                     19790                          DL      N6705Y
4                     20355                          US      N504AU

   Flight_Number_Reporting_Airline  OriginAirportID
OriginAirportSeqID  \
0                              675            13487
1348701
1                              671            13342
1334202
2                             3297            11921
1192102
3                             1806            12892
1289201
4                              465            11618
1161801

   OriginCityMarketID Origin      OriginCityName OriginState
OriginStateFips  \
0               31650    MSP     Minneapolis, MN          MN
27.0
1               33342    MKE       Milwaukee, WI          WI
55.0
```

```
2                 31921    GJT  Grand Junction, CO          CO
8.0
3                 32575    LAX     Los Angeles, CA          CA
6.0
4                 31703    EWR          Newark, NJ          NJ
34.0

  OriginStateName  OriginWac
0       Minnesota         63
1       Wisconsin         45
2        Colorado         82
3      California         91
4      New Jersey         21
```

```
df.iloc[:, :20].describe()
```

```
               Year       Quarter         Month    DayofMonth
DayOfWeek  \
count  2.000000e+06  2.000000e+06  2.000000e+06  2.000000e+06
2.000000e+06
mean   2.004314e+03  2.501267e+00  6.500761e+00  1.572202e+01
3.937445e+00
std    9.228930e+00  1.118022e+00  3.443460e+00  8.778412e+00
1.990369e+00
min    1.987000e+03  1.000000e+00  1.000000e+00  1.000000e+00
1.000000e+00
25%    1.997000e+03  1.000000e+00  3.000000e+00  8.000000e+00
2.000000e+00
50%    2.005000e+03  3.000000e+00  7.000000e+00  1.600000e+01
4.000000e+00
75%    2.012000e+03  3.000000e+00  9.000000e+00  2.300000e+01
6.000000e+00
max    2.020000e+03  4.000000e+00  1.200000e+01  3.100000e+01
7.000000e+00

       DOT_ID_Reporting_Airline  Flight_Number_Reporting_Airline  \
count              2.000000e+06                     2.000000e+06
mean               1.992450e+04                     1.719375e+03
std                3.665827e+02                     1.659726e+03
min                1.938600e+04                     1.000000e+00
25%                1.970400e+04                     5.220000e+02
50%                1.980500e+04                     1.170000e+03
75%                2.035500e+04                     2.211000e+03
max                2.117100e+04                     9.794000e+03

       OriginAirportID  OriginAirportSeqID  OriginCityMarketID  \
count     2.000000e+06        2.000000e+06        2.000000e+06
mean      1.271899e+04        1.271901e+06        3.173373e+04
std       1.534529e+03        1.534527e+05        1.302432e+03
min       1.013500e+04        1.013501e+06        3.007000e+04
```

```
25%       1.129200e+04       1.129202e+06       3.064700e+04
50%       1.289200e+04       1.289201e+06       3.145300e+04
75%       1.405700e+04       1.405702e+06       3.257500e+04
max       1.686900e+04       1.686901e+06       3.610100e+04

       OriginStateFips     OriginWac
count     1.999354e+06  2.000000e+06
mean      2.687446e+01  5.522946e+01
std       1.643874e+01  2.682221e+01
min       1.000000e+00  1.000000e+00
25%       1.200000e+01  3.400000e+01
50%       2.600000e+01  5.200000e+01
75%       4.200000e+01  8.100000e+01
max       7.800000e+01  8.410000e+02
```

lets check 'Flight_Number_Reporting_Airline' or 'DOT_ID_Reporting_Airline' by checking the number of unique values which should equal the number of observation if this is the unique identifier.

```
check_unique_values(df,
['Flight_Number_Reporting_Airline','DOT_ID_Reporting_Airline'])

Number of unique values per column
----------------------------------
Flight_Number_Reporting_Airline has 8050 unique values:
----------------------------------
[ 675  671 3297 ... 9519 7917 7645]
----------------------------------
DOT_ID_Reporting_Airline has 34 unique values:
----------------------------------
[19386 20437 20398 19790 20355 19704 20363 20378 19393 19805 20304
19977
 19822 19930 20366 20452 20397 19991 20211 20404 20409 19690 20374
20416
 20436 19391 19707 20368 20417 21171 20312 19678 20384 20295]
```

Result : both are not

```
df.iloc[:, 20:40].info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000000 entries, 0 to 1999999
Data columns (total 20 columns):
 #   Column                Dtype
---  ------                -----
 0   DestAirportID         int64
 1   DestAirportSeqID      int64
 2   DestCityMarketID      int64
 3   Dest                  object
```

```
 4   DestCityName          object
 5   DestState             object
 6   DestStateFips         float64
 7   DestStateName         object
 8   DestWac               int64
 9   CRSDepTime            int64
 10  DepTime               float64
 11  DepDelay              float64
 12  DepDelayMinutes       float64
 13  DepDel15              float64
 14  DepartureDelayGroups  float64
 15  DepTimeBlk            object
 16  TaxiOut               float64
 17  WheelsOff             float64
 18  WheelsOn              float64
 19  TaxiIn                float64
dtypes: float64(10), int64(5), object(5)
memory usage: 305.2+ MB
```

```
df.iloc[:, 20:40].head()
```

```
   DestAirportID  DestAirportSeqID  DestCityMarketID Dest  \
0          14869           1486902             34614  SLC
1          13204           1320401             31454  MCO
2          11298           1129803             30194  DFW
3          11433           1143301             31295  DTW
4          11057           1105702             31057  CLT

            DestCityName DestState  DestStateFips   DestStateName
DestWac  \
0     Salt Lake City, UT        UT           49.0            Utah
87
1            Orlando, FL        FL           12.0         Florida
33
2  Dallas/Fort Worth, TX        TX           48.0           Texas
74
3            Detroit, MI        MI           26.0        Michigan
43
4          Charlotte, NC        NC           37.0  North Carolina
36

   CRSDepTime  DepTime  DepDelay  DepDelayMinutes  DepDel15  \
0        1640   1659.0      19.0             19.0       1.0
1        1204   1202.0      -2.0              0.0       0.0
2        1630   1644.0      14.0             14.0       0.0
3        1305   1305.0       0.0              0.0       0.0
4        1820   1911.0      51.0             51.0       1.0

   DepartureDelayGroups DepTimeBlk  TaxiOut  WheelsOff  WheelsOn
TaxiIn
```

```
0                    1.0  1600-1659     24.0     1723.0    1856.0
3.0
1                   -1.0  1200-1259     10.0     1212.0    1533.0
8.0
2                    0.0  1600-1659      9.0     1653.0    1936.0
6.0
3                    0.0  1300-1359     23.0     1328.0    2008.0
7.0
4                    3.0  1800-1859     19.0     1930.0    2050.0
8.0
```

```
df.iloc[:, 20:40].describe()
```

```
       DestAirportID  DestAirportSeqID  DestCityMarketID
DestStateFips  \
count   2.000000e+06      2.000000e+06      2.000000e+06
1.999406e+06
mean    1.271924e+04      1.271925e+06      3.173239e+04
2.685666e+01
std     1.534860e+03      1.534858e+05      1.302004e+03
1.643312e+01
min     1.013500e+04      1.013501e+06      3.007000e+04
1.000000e+00
25%     1.129200e+04      1.129202e+06      3.064700e+04
1.200000e+01
50%     1.289200e+04      1.289201e+06      3.145300e+04
2.600000e+01
75%     1.405700e+04      1.405702e+06      3.257500e+04
4.200000e+01
max     1.686900e+04      1.686901e+06      3.610100e+04
7.800000e+01

            DestWac    CRSDepTime       DepTime      DepDelay  \
count  2.000000e+06  2.000000e+06  1.963995e+06  1.963932e+06
mean   5.526029e+01  1.332350e+03  1.343248e+03  8.587405e+00
std    2.678134e+01  4.765702e+02  4.818427e+02  3.272473e+01
min    1.000000e+00  0.000000e+00  1.000000e+00 -9.900000e+02
25%    3.400000e+01  9.250000e+02  9.300000e+02 -3.000000e+00
50%    5.200000e+01  1.325000e+03  1.331000e+03  0.000000e+00
75%    8.100000e+01  1.728000e+03  1.737000e+03  7.000000e+00
max    8.410000e+02  2.400000e+03  2.400000e+03  1.878000e+03

       DepDelayMinutes      DepDel15  DepartureDelayGroups
TaxiOut  \
count     1.963932e+06  1.963932e+06          1.963932e+06
1.584358e+06
mean      1.049667e+01  1.696362e-01          6.643356e-02
1.580659e+01
std       3.196467e+01  3.753130e-01          1.824514e+00
1.023564e+01
```

```
min       0.000000e+00  0.000000e+00         -2.000000e+00
0.000000e+00
25%       0.000000e+00  0.000000e+00         -1.000000e+00
1.000000e+01
50%       0.000000e+00  0.000000e+00          0.000000e+00
1.300000e+01
75%       7.000000e+00  0.000000e+00          0.000000e+00
1.800000e+01
max       1.878000e+03  1.000000e+00          1.200000e+01
1.412000e+03

          WheelsOff      WheelsOn        TaxiIn
count  1.584323e+06  1.582042e+06  1.582153e+06
mean   1.362872e+03  1.479911e+03  6.714089e+00
std    4.855511e+02  5.065056e+02  7.948352e+00
min    1.000000e+00  1.000000e+00  0.000000e+00
25%    9.440000e+02  1.105000e+03  4.000000e+00
50%    1.344000e+03  1.513000e+03  5.000000e+00
75%    1.751000e+03  1.910000e+03  8.000000e+00
max    2.400000e+03  2.400000e+03  1.439000e+03
```

```
df.iloc[:, 40:60].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000000 entries, 0 to 1999999
Data columns (total 20 columns):
 #   Column              Dtype  
---  ------              -----  
 0   CRSArrTime          int64  
 1   ArrTime             float64
 2   ArrDelay            float64
 3   ArrDelayMinutes     float64
 4   ArrDel15            float64
 5   ArrivalDelayGroups  float64
 6   ArrTimeBlk          object 
 7   Cancelled           float64
 8   CancellationCode    object 
 9   Diverted            float64
 10  CRSElapsedTime      float64
 11  ActualElapsedTime   float64
 12  AirTime             float64
 13  Flights             float64
 14  Distance            float64
 15  DistanceGroup       int64  
 16  CarrierDelay        float64
 17  WeatherDelay        float64
 18  NASDelay            float64
 19  SecurityDelay       float64
dtypes: float64(16), int64(2), object(2)
memory usage: 305.2+ MB
```

```
df.iloc[:, 40:60].head(20)
```

```
    CRSArrTime  ArrTime  ArrDelay  ArrDelayMinutes  ArrDel15  \
0         1836   1859.0      23.0             23.0       1.0
1         1541   1541.0       0.0              0.0       0.0
2         1945   1942.0      -3.0              0.0       0.0
3         2035   2015.0     -20.0              0.0       0.0
4         2026   2058.0      32.0             32.0       1.0
5          730    741.0      11.0             11.0       0.0
6         2000   2002.0       2.0              2.0       0.0
7         2057     31.0     214.0            214.0       1.0
8         1810   1820.0      10.0             10.0       0.0
9         2250   2319.0      29.0             29.0       1.0
10        1325   1331.0       6.0              6.0       0.0
11        1300   1255.0      -5.0              0.0       0.0
12        1521   1511.0     -10.0              0.0       0.0
13        1705   1646.0     -19.0              0.0       0.0
14        1206   1150.0     -16.0              0.0       0.0
15         728    737.0       9.0              9.0       0.0
16        1246   1239.0      -7.0              0.0       0.0
17        1855   1832.0     -23.0              0.0       0.0
18        2327   2313.0     -14.0              0.0       0.0
19        1700      NaN       NaN              NaN       NaN

    ArrivalDelayGroups ArrTimeBlk  Cancelled CancellationCode
Diverted  \
0                  1.0  1800-1859        0.0              NaN
0.0
1                  0.0  1500-1559        0.0              NaN
0.0
2                 -1.0  1900-1959        0.0              NaN
0.0
3                 -2.0  2000-2059        0.0              NaN
0.0
4                  2.0  2000-2059        0.0              NaN
0.0
5                  0.0  0700-0759        0.0              NaN
0.0
6                  0.0  2000-2059        0.0              NaN
0.0
7                 12.0  2000-2059        0.0              NaN
0.0
8                  0.0  1800-1859        0.0              NaN
0.0
9                  1.0  2200-2259        0.0              NaN
0.0
10                 0.0  1300-1359        0.0              NaN
0.0
11                -1.0  1300-1359        0.0              NaN
0.0
```

```
12                   -1.0  1500-1559        0.0              NaN
0.0
13                   -2.0  1700-1759        0.0              NaN
0.0
14                   -2.0  1200-1259        0.0              NaN
0.0
15                    0.0  0700-0759        0.0              NaN
0.0
16                   -1.0  1200-1259        0.0              NaN
0.0
17                   -2.0  1800-1859        0.0              NaN
0.0
18                   -1.0  2300-2359        0.0              NaN
0.0
19                    NaN  1700-1759        1.0                A
0.0

    CRSElapsedTime  ActualElapsedTime  AirTime  Flights  Distance  \
0            176.0              180.0    153.0      1.0     991.0
1            157.0              159.0    141.0      1.0    1066.0
2            135.0              118.0    103.0      1.0     773.0
3            270.0              250.0    220.0      1.0    1979.0
4            126.0              107.0     80.0      1.0     529.0
5             51.0               62.0     28.0      1.0     190.0
6            125.0              131.0     94.0      1.0     563.0
7             67.0               60.0     35.0      1.0     192.0
8             80.0               88.0     59.0      1.0     316.0
9            140.0              153.0    114.0      1.0     793.0
10            40.0               44.0      NaN      1.0     109.0
11            95.0               90.0     77.0      1.0     562.0
12           156.0              149.0      NaN      1.0    1045.0
13           124.0              109.0     95.0      1.0     677.0
14           126.0              115.0     99.0      1.0     733.0
15            58.0               66.0      NaN      1.0     278.0
16            45.0               49.0     24.0      1.0      98.0
17           130.0              119.0    102.0      1.0     689.0
18           282.0              271.0    255.0      1.0    2288.0
19            85.0                NaN      NaN      1.0     373.0

    DistanceGroup  CarrierDelay  WeatherDelay  NASDelay  SecurityDelay

0               4           NaN           NaN       NaN            NaN

1               5           NaN           NaN       NaN            NaN

2               4           NaN           NaN       NaN            NaN

3               8           NaN           NaN       NaN            NaN

4               3           0.0           0.0       0.0            0.0
```

5	1	NaN	NaN	NaN	NaN
6	3	NaN	NaN	NaN	NaN
7	1	0.0	0.0	0.0	0.0
8	2	NaN	NaN	NaN	NaN
9	4	0.0	0.0	13.0	0.0
10	1	NaN	NaN	NaN	NaN
11	3	NaN	NaN	NaN	NaN
12	5	NaN	NaN	NaN	NaN
13	3	NaN	NaN	NaN	NaN
14	3	NaN	NaN	NaN	NaN
15	2	NaN	NaN	NaN	NaN
16	1	NaN	NaN	NaN	NaN
17	3	NaN	NaN	NaN	NaN
18	10	NaN	NaN	NaN	NaN
19	2	NaN	NaN	NaN	NaN

```
df.iloc[:, 40:60].describe()
```

```
         CRSArrTime       ArrTime      ArrDelay  ArrDelayMinutes  \
count  2.000000e+06  1.960449e+06  1.958922e+06     1.958922e+06
mean   1.492285e+03  1.487321e+03  6.205467e+00     1.179442e+01
std    4.955542e+02  5.062998e+02  3.483340e+01     3.197121e+01
min    0.000000e+00  1.000000e+00 -7.060000e+02     0.000000e+00
25%    1.115000e+03  1.111000e+03 -1.000000e+01     0.000000e+00
50%    1.520000e+03  1.518000e+03 -1.000000e+00     0.000000e+00
75%    1.913000e+03  1.915000e+03  1.000000e+01     1.000000e+01
max    2.400000e+03  2.400000e+03  1.898000e+03     1.898000e+03

           ArrDel15  ArrivalDelayGroups     Cancelled      Diverted  \
count  1.958922e+06        1.958922e+06  2.000000e+06  2.000000e+06
mean   1.980349e-01       -7.384521e-02  1.823100e-02  2.295000e-03
std    3.985187e-01        1.994990e+00  1.337858e-01  4.785117e-02
min    0.000000e+00       -2.000000e+00  0.000000e+00  0.000000e+00
25%    0.000000e+00       -1.000000e+00  0.000000e+00  0.000000e+00
50%    0.000000e+00       -1.000000e+00  0.000000e+00  0.000000e+00
75%    0.000000e+00        0.000000e+00  0.000000e+00  0.000000e+00
```

```
max    1.000000e+00       1.200000e+01  1.000000e+00  1.000000e+00

       CRSElapsedTime  ActualElapsedTime       AirTime    Flights  \
count    1.999719e+06       1.958948e+06  1.580651e+06  2000000.0
mean     1.271275e+02       1.249893e+02  1.059533e+02        1.0
std      7.040894e+01       7.038500e+01  6.859287e+01        0.0
min      0.000000e+00      -1.480000e+02 -7.030000e+02        1.0
25%      7.500000e+01       7.300000e+01  5.600000e+01        1.0
50%      1.090000e+02       1.060000e+02  8.700000e+01        1.0
75%      1.590000e+02       1.560000e+02  1.350000e+02        1.0
max      7.050000e+02       9.750000e+02  9.650000e+02        1.0

           Distance  DistanceGroup   CarrierDelay   WeatherDelay  \
count  2.000000e+06   2.000000e+06  221803.000000  221803.000000
mean   7.334963e+02   3.409396e+00      16.892580       2.939929
std    5.684968e+02   2.242753e+00      46.222289      21.101110
min    1.100000e+01   1.000000e+00       0.000000       0.000000
25%    3.250000e+02   2.000000e+00       0.000000       0.000000
50%    5.800000e+02   3.000000e+00       0.000000       0.000000
75%    9.670000e+02   4.000000e+00      17.000000       0.000000
max    5.095000e+03   1.100000e+01    1878.000000    1847.000000

            NASDelay  SecurityDelay
count  221803.000000  221803.000000
mean       15.389395       0.084873
std        30.538782       2.109449
min         0.000000       0.000000
25%         0.000000       0.000000
50%         4.000000       0.000000
75%        19.000000       0.000000
max      1343.000000     219.000000
```

```
df.iloc[:, 60:80].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000000 entries, 0 to 1999999
Data columns (total 20 columns):
 #   Column                Dtype
---  ------                -----
 0   LateAircraftDelay     float64
 1   FirstDepTime          float64
 2   TotalAddGTime         float64
 3   LongestAddGTime       float64
 4   DivAirportLandings    float64
 5   DivReachedDest        float64
 6   DivActualElapsedTime  float64
 7   DivArrDelay           float64
 8   DivDistance           float64
 9   Div1Airport           object
 10  Div1AirportID         float64
```

```
 11  Div1AirportSeqID      float64
 12  Div1WheelsOn          float64
 13  Div1TotalGTime        float64
 14  Div1LongestGTime      float64
 15  Div1WheelsOff         float64
 16  Div1TailNum           object
 17  Div2Airport           object
 18  Div2AirportID         float64
 19  Div2AirportSeqID      float64
dtypes: float64(17), object(3)
memory usage: 305.2+ MB
```

```
df.iloc[:, 60:80].head()
```

```
   LateAircraftDelay  FirstDepTime  TotalAddGTime  LongestAddGTime  \
0                NaN           NaN            NaN              NaN
1                NaN           NaN            NaN              NaN
2                NaN           NaN            NaN              NaN
3                NaN           NaN            NaN              NaN
4               32.0           NaN            NaN              NaN

   DivAirportLandings  DivReachedDest  DivActualElapsedTime
DivArrDelay  \
0                 NaN             NaN                   NaN
NaN
1                 0.0             NaN                   NaN
NaN
2                 0.0             NaN                   NaN
NaN
3                 0.0             NaN                   NaN
NaN
4                 NaN             NaN                   NaN
NaN

   DivDistance Div1Airport  Div1AirportID  Div1AirportSeqID
Div1WheelsOn  \
0          NaN         NaN            NaN               NaN
NaN
1          NaN         NaN            NaN               NaN
NaN
2          NaN         NaN            NaN               NaN
NaN
3          NaN         NaN            NaN               NaN
NaN
4          NaN         NaN            NaN               NaN
NaN

   Div1TotalGTime  Div1LongestGTime  Div1WheelsOff Div1TailNum
Div2Airport  \
0             NaN               NaN            NaN         NaN
```

```
NaN
1               NaN                 NaN             NaN          NaN
NaN
2               NaN                 NaN             NaN          NaN
NaN
3               NaN                 NaN             NaN          NaN
NaN
4               NaN                 NaN             NaN          NaN
NaN

   Div2AirportID  Div2AirportSeqID
0            NaN               NaN
1            NaN               NaN
2            NaN               NaN
3            NaN               NaN
4            NaN               NaN
```

```
df.iloc[:, 60:80].describe()
```

```
       LateAircraftDelay  FirstDepTime  TotalAddGTime  LongestAddGTime
\
count      221803.000000   4454.000000    4454.000000      4454.000000

mean           22.054170   1324.179838      36.215088        35.504490

std            41.631429    490.605125      32.909992        31.155685

min             0.000000      1.000000       1.000000         1.000000

25%             0.000000    859.000000      16.000000        16.000000

50%             0.000000   1331.000000      27.000000        26.000000

75%            27.000000   1728.000000      43.000000        43.000000

max          1407.000000   2400.000000     339.000000       208.000000


       DivAirportLandings  DivReachedDest  DivActualElapsedTime
DivArrDelay  \
count       746114.000000     1775.000000           1501.000000
1501.000000
mean             0.003674        0.845634            351.501666
208.504997
std              0.117989        0.361401            165.854855
160.300462
min              0.000000        0.000000             84.000000
2.000000
25%              0.000000        1.000000            249.000000
121.000000
50%              0.000000        1.000000            314.000000
```

```
169.000000
75%                0.000000        1.000000            407.000000
243.000000
max                9.000000        1.000000           1420.000000
1603.000000

       DivDistance  Div1AirportID  Div1AirportSeqID  Div1WheelsOn  \
count  1775.000000    1881.000000      1.881000e+03   1881.000000
mean     40.184225   12697.503987      1.269753e+06   1505.026050
std     145.714770    1617.893469      1.617892e+05    536.936115
min       0.000000   10135.000000      1.013502e+06      1.000000
25%       0.000000   11203.000000      1.120302e+06   1133.000000
50%       0.000000   12478.000000      1.247802e+06   1602.000000
75%       0.000000   14107.000000      1.410702e+06   1919.000000
max    2122.000000   16229.000000      1.622902e+06   2359.000000

       Div1TotalGTime  Div1LongestGTime  Div1WheelsOff  Div2AirportID
\
count     1881.000000       1881.000000    1512.000000      14.000000

mean        34.927698         28.360447    1558.488757   12846.071429

std         34.177899         30.210587     581.816735    1366.140575

min          1.000000          1.000000       1.000000   10397.000000

25%         14.000000         10.000000    1157.750000   12264.500000

50%         22.000000         16.000000    1703.000000   12579.000000

75%         44.000000         34.000000    2019.250000   13783.000000

max        280.000000        211.000000    2359.000000   14771.000000


       Div2AirportSeqID
count      1.400000e+01
mean       1.284610e+06
std        1.366139e+05
min        1.039705e+06
25%        1.226452e+06
50%        1.257903e+06
75%        1.378303e+06
max        1.477101e+06
```

```
df.iloc[:, 80:100].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000000 entries, 0 to 1999999
Data columns (total 20 columns):
 #   Column            Dtype
```

```
---  ------            -----
 0   Div2WheelsOn      float64
 1   Div2TotalGTime    float64
 2   Div2LongestGTime  float64
 3   Div2WheelsOff     float64
 4   Div2TailNum       object
 5   Div3Airport       float64
 6   Div3AirportID     float64
 7   Div3AirportSeqID  float64
 8   Div3WheelsOn      float64
 9   Div3TotalGTime    float64
 10  Div3LongestGTime  float64
 11  Div3WheelsOff     float64
 12  Div3TailNum       float64
 13  Div4Airport       float64
 14  Div4AirportID     float64
 15  Div4AirportSeqID  float64
 16  Div4WheelsOn      float64
 17  Div4TotalGTime    float64
 18  Div4LongestGTime  float64
 19  Div4WheelsOff     float64
dtypes: float64(19), object(1)
memory usage: 305.2+ MB
```

```
df.iloc[:, 80:100].head()
```

```
   Div2WheelsOn  Div2TotalGTime  Div2LongestGTime  Div2WheelsOff
Div2TailNum  \
0           NaN             NaN               NaN            NaN
NaN
1           NaN             NaN               NaN            NaN
NaN
2           NaN             NaN               NaN            NaN
NaN
3           NaN             NaN               NaN            NaN
NaN
4           NaN             NaN               NaN            NaN
NaN

   Div3Airport  Div3AirportID  Div3AirportSeqID  Div3WheelsOn
Div3TotalGTime  \
0          NaN            NaN               NaN           NaN
NaN
1          NaN            NaN               NaN           NaN
NaN
2          NaN            NaN               NaN           NaN
NaN
3          NaN            NaN               NaN           NaN
NaN
4          NaN            NaN               NaN           NaN
```

```
NaN

   Div3LongestGTime  Div3WheelsOff  Div3TailNum  Div4Airport  
Div4AirportID  \
0               NaN            NaN          NaN          NaN            
NaN  
1               NaN            NaN          NaN          NaN            
NaN  
2               NaN            NaN          NaN          NaN            
NaN  
3               NaN            NaN          NaN          NaN            
NaN  
4               NaN            NaN          NaN          NaN            
NaN  

   Div4AirportSeqID  Div4WheelsOn  Div4TotalGTime  Div4LongestGTime  \
0               NaN           NaN             NaN               NaN   
1               NaN           NaN             NaN               NaN   
2               NaN           NaN             NaN               NaN   
3               NaN           NaN             NaN               NaN   
4               NaN           NaN             NaN               NaN   

   Div4WheelsOff  
0            NaN  
1            NaN  
2            NaN  
3            NaN  
4            NaN  
```

```
df.iloc[:, 80:100].describe()
```

```
       Div2WheelsOn  Div2TotalGTime  Div2LongestGTime  
Div2WheelsOff  \
count     14.000000       14.000000         14.000000       3.000000   

mean    1318.142857       17.214286         15.642857    1470.000000   

std      691.330474       15.126027         12.767791     660.218146   

min       17.000000        4.000000          4.000000     954.000000   

25%     1086.750000        5.250000          5.250000    1098.000000   

50%     1530.500000       13.500000         13.500000    1242.000000   

75%     1710.500000       19.750000         19.750000    1728.000000   

max     2055.000000       55.000000         44.000000    2214.000000   


       Div3Airport  Div3AirportID  Div3AirportSeqID  Div3WheelsOn  \
```

```
count           0.0             0.0                0.0           0.0
mean            NaN             NaN                NaN           NaN
std             NaN             NaN                NaN           NaN
min             NaN             NaN                NaN           NaN
25%             NaN             NaN                NaN           NaN
50%             NaN             NaN                NaN           NaN
75%             NaN             NaN                NaN           NaN
max             NaN             NaN                NaN           NaN

       Div3TotalGTime  Div3LongestGTime  Div3WheelsOff  Div3TailNum  \
count             0.0               0.0            0.0          0.0
mean              NaN               NaN            NaN          NaN
std               NaN               NaN            NaN          NaN
min               NaN               NaN            NaN          NaN
25%               NaN               NaN            NaN          NaN
50%               NaN               NaN            NaN          NaN
75%               NaN               NaN            NaN          NaN
max               NaN               NaN            NaN          NaN

       Div4Airport  Div4AirportID  Div4AirportSeqID  Div4WheelsOn  \
count          0.0            0.0               0.0           0.0
mean           NaN            NaN               NaN           NaN
std            NaN            NaN               NaN           NaN
min            NaN            NaN               NaN           NaN
25%            NaN            NaN               NaN           NaN
50%            NaN            NaN               NaN           NaN
75%            NaN            NaN               NaN           NaN
max            NaN            NaN               NaN           NaN

       Div4TotalGTime  Div4LongestGTime  Div4WheelsOff
count             0.0               0.0            0.0
mean              NaN               NaN            NaN
std               NaN               NaN            NaN
min               NaN               NaN            NaN
25%               NaN               NaN            NaN
50%               NaN               NaN            NaN
75%               NaN               NaN            NaN
max               NaN               NaN            NaN
```

```
df.iloc[:, 100:120].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000000 entries, 0 to 1999999
Data columns (total 9 columns):
 #   Column            Dtype
---  ------            -----
 0   Div4TailNum       float64
 1   Div5Airport       float64
 2   Div5AirportID     float64
 3   Div5AirportSeqID  float64
```

```
 4   Div5WheelsOn      float64
 5   Div5TotalGTime    float64
 6   Div5LongestGTime  float64
 7   Div5WheelsOff     float64
 8   Div5TailNum       float64
dtypes: float64(9)
memory usage: 137.3 MB
```

```
df.iloc[:, 100:120].head()
```

```
   Div4TailNum  Div5Airport  Div5AirportID  Div5AirportSeqID
Div5WheelsOn  \
0          NaN          NaN            NaN               NaN
NaN
1          NaN          NaN            NaN               NaN
NaN
2          NaN          NaN            NaN               NaN
NaN
3          NaN          NaN            NaN               NaN
NaN
4          NaN          NaN            NaN               NaN
NaN

   Div5TotalGTime  Div5LongestGTime  Div5WheelsOff  Div5TailNum
0             NaN               NaN            NaN          NaN
1             NaN               NaN            NaN          NaN
2             NaN               NaN            NaN          NaN
3             NaN               NaN            NaN          NaN
4             NaN               NaN            NaN          NaN
```

```
df.iloc[:,  100:120].describe()
```

```
       Div4TailNum  Div5Airport  Div5AirportID  Div5AirportSeqID  \
count          0.0          0.0            0.0               0.0
mean           NaN          NaN            NaN               NaN
std            NaN          NaN            NaN               NaN
min            NaN          NaN            NaN               NaN
25%            NaN          NaN            NaN               NaN
50%            NaN          NaN            NaN               NaN
75%            NaN          NaN            NaN               NaN
max            NaN          NaN            NaN               NaN

       Div5WheelsOn  Div5TotalGTime  Div5LongestGTime
Div5WheelsOff  \
count           0.0             0.0               0.0           0.0

mean            NaN             NaN               NaN           NaN

std             NaN             NaN               NaN           NaN

min             NaN             NaN               NaN           NaN
```

```
25%                NaN              NaN                NaN             NaN
50%                NaN              NaN                NaN             NaN
75%                NaN              NaN                NaN             NaN
max                NaN              NaN                NaN             NaN

       Div5TailNum
count          0.0
mean           NaN
std            NaN
min            NaN
25%            NaN
50%            NaN
75%            NaN
max            NaN
```

```
df.shape
```

```
(2000000, 109)
```

The data frame contains 2,000,000 observation and 109 feature

Data Wrangling

In this part the following will be done

1. creating a copy of data frame
2. create a data frame for cancelled trips
3. create a data frame for diverted trips
4. create a data frame for others

```
cleaned_airline_df = df.copy()
cleaned_airline_df.shape
```

```
(2000000, 109)
```

Drop columns with zero count

```
cleaned_airline_df.drop(cleaned_airline_df.columns[85:], axis=1, inplace=True)
cleaned_airline_df.shape
```

```
(2000000, 85)
```

```
cleaned_airline_df.iloc[:, 70:85].describe()
```

	Div1AirportID	Div1AirportSeqID	Div1WheelsOn	Div1TotalGTime
count	1881.000000	1.881000e+03	1881.000000	1881.000000
mean	12697.503987	1.269753e+06	1505.026050	34.927698
std	1617.893469	1.617892e+05	536.936115	34.177899
min	10135.000000	1.013502e+06	1.000000	1.000000
25%	11203.000000	1.120302e+06	1133.000000	14.000000
50%	12478.000000	1.247802e+06	1602.000000	22.000000
75%	14107.000000	1.410702e+06	1919.000000	44.000000
max	16229.000000	1.622902e+06	2359.000000	280.000000

	Div1LongestGTime	Div1WheelsOff	Div2AirportID	Div2AirportSeqID
count	1881.000000	1512.000000	14.000000	1.400000e+01
mean	28.360447	1558.488757	12846.071429	1.284610e+06
std	30.210587	581.816735	1366.140575	1.366139e+05
min	1.000000	1.000000	10397.000000	1.039705e+06
25%	10.000000	1157.750000	12264.500000	1.226452e+06
50%	16.000000	1703.000000	12579.000000	1.257903e+06
75%	34.000000	2019.250000	13783.000000	1.378303e+06
max	211.000000	2359.000000	14771.000000	1.477101e+06

	Div2WheelsOn	Div2TotalGTime	Div2LongestGTime	Div2WheelsOff
count	14.000000	14.000000	14.000000	3.000000
mean	1318.142857	17.214286	15.642857	1470.000000
std	691.330474	15.126027	12.767791	660.218146
min	17.000000	4.000000	4.000000	954.000000
25%	1086.750000	5.250000	5.250000	1098.000000
50%	1530.500000	13.500000	13.500000	1242.000000
75%	1710.500000	19.750000	19.750000	1728.000000
max	2055.000000	55.000000	44.000000	2214.000000

Adding New Columns

Add a new categorical column for the day of the week description.

```
day_of_week_mapping = {
    1: 'Monday',
    2: 'Tuesday',
    3: 'Wednesday',
    4: 'Thursday',
    5: 'Friday',
    6: 'Saturday',
    7: 'Sunday'
}

# Apply the mapping to the 'DayOfWeek' column
cleaned_airline_df['DayOfWeek_Desc'] =
cleaned_airline_df['DayOfWeek'].map(day_of_week_mapping)
```

Add a new categorical column for the quarter of the year description.

```
quarter_mapping = {1: 'Q1', 2: 'Q2', 3: 'Q3', 4: 'Q4'}
cleaned_airline_df['Quarter_Desc'] =
cleaned_airline_df['Quarter'].map(quarter_mapping)
```

Check Unique Values

```
check_unique_values(df=cleaned_airline_df,columns=[
    'CarrierDelay','WeatherDelay','NASDelay',
    'SecurityDelay','LateAircraftDelay'])
```

```
Number of unique values per column
----------------------------------
CarrierDelay has 706 unique values:
----------------------------------
[      nan 0.000e+00 4.400e+01 9.000e+00 1.000e+00 2.000e+00 2.400e+01
 3.600e+01 2.600e+01 1.000e+01 1.200e+01 2.200e+01 7.000e+00 4.200e+01
 3.000e+01 4.000e+00 6.000e+00 1.100e+01 1.500e+01 4.300e+01 1.130e+02
 3.400e+01 1.800e+01 2.100e+01 5.000e+00 4.100e+01 2.000e+01 3.200e+01
 6.400e+02 3.100e+01 1.400e+01 1.600e+01 8.000e+00 6.100e+01 2.300e+01
 1.330e+02 1.120e+02 8.600e+01 1.280e+02 2.500e+01 3.300e+01 7.600e+01
 8.500e+01 5.500e+01 7.000e+01 5.200e+01 3.000e+00 4.900e+01 1.050e+02
 3.500e+01 3.700e+01 7.700e+01 2.900e+01 1.260e+02 1.300e+01 4.500e+01
 2.980e+02 2.430e+02 1.930e+02 1.900e+01 5.700e+01 9.200e+01 2.700e+01
 9.500e+01 7.100e+01 4.600e+01 1.800e+02 2.800e+01 2.160e+02 7.500e+01
 3.800e+01 6.900e+01 5.320e+02 4.690e+02 6.500e+01 8.300e+01 4.000e+01
 5.540e+02 1.700e+01 6.400e+01 1.680e+02 3.900e+01 7.900e+01 1.410e+02
 1.110e+02 1.610e+02 1.300e+02 1.390e+02 2.550e+02 1.630e+02 4.700e+01
 9.540e+02 4.800e+01 6.700e+01 6.200e+01 1.150e+02 2.180e+02 2.560e+02
 2.150e+02 1.040e+02 5.300e+01 1.470e+02 5.800e+01 7.200e+01 7.300e+01
```

1.380e+02	1.160e+02	2.870e+02	5.600e+01	1.200e+02	9.300e+01	9.900e+01
1.030e+02	1.080e+02	5.000e+01	2.250e+02	4.630e+02	6.600e+01	8.700e+01
2.420e+02	5.400e+01	6.300e+01	3.270e+02	8.400e+01	1.020e+02	5.100e+01
2.710e+02	1.860e+02	8.000e+01	8.900e+01	5.900e+01	5.050e+02	7.400e+01
7.800e+01	4.830e+02	1.520e+02	1.450e+02	8.520e+02	1.170e+02	6.000e+01
1.500e+02	1.990e+02	1.810e+02	1.360e+02	4.450e+02	1.660e+02	2.990e+02
4.540e+02	1.840e+02	1.090e+02	1.420e+02	1.640e+02	1.070e+02	2.660e+02
8.800e+01	5.360e+02	1.240e+02	3.020e+02	1.190e+02	3.100e+02	8.200e+01
2.960e+02	6.800e+01	9.700e+01	2.450e+02	8.100e+01	1.560e+02	9.100e+01
1.620e+02	4.300e+02	1.700e+02	1.510e+02	2.110e+02	9.400e+01	6.020e+02
1.820e+02	9.000e+01	2.780e+02	1.590e+02	1.440e+02	1.022e+03	9.600e+01
2.400e+02	2.670e+02	1.180e+02	2.040e+02	1.010e+02	2.600e+02	1.460e+02
1.350e+02	3.330e+02	1.060e+02	1.550e+02	1.100e+02	9.800e+01	1.690e+02
1.780e+02	2.930e+02	2.540e+02	3.030e+02	3.140e+02	2.330e+02	5.170e+02
1.000e+02	3.630e+02	1.220e+02	1.340e+02	2.570e+02	1.270e+02	1.950e+02
2.060e+02	5.900e+02	6.950e+02	5.730e+02	1.910e+02	1.710e+02	9.220e+02
3.290e+02	1.580e+02	1.890e+02	7.360e+02	7.380e+02	1.290e+02	9.920e+02
2.850e+02	2.210e+02	1.430e+02	1.770e+02	2.090e+02	1.530e+02	1.760e+02
1.400e+02	1.600e+02	3.350e+02	1.940e+02	1.250e+02	2.720e+02	2.860e+02
9.760e+02	1.870e+02	2.380e+02	2.240e+02	3.010e+02	1.540e+02	2.120e+02
1.125e+03	3.130e+02	2.030e+02	4.180e+02	3.870e+02	2.080e+02	2.750e+02
1.370e+02	2.530e+02	1.880e+02	2.500e+02	2.170e+02	2.020e+02	2.000e+02
1.740e+02	1.210e+02	2.640e+02	1.320e+02	2.320e+02	3.530e+02	1.480e+02
2.370e+02	7.720e+02	2.810e+02	2.350e+02	1.830e+02	2.310e+02	1.730e+02
3.910e+02	1.140e+02	1.310e+02	2.800e+02	1.069e+03	4.880e+02	2.070e+02
1.230e+02	2.620e+02	8.080e+02	5.190e+02	4.040e+02	2.230e+02	4.410e+02
4.320e+02	1.720e+02	1.970e+02	5.990e+02	2.700e+02	8.920e+02	6.100e+02
2.140e+02	5.200e+02	3.460e+02	1.850e+02	4.150e+02	3.730e+02	4.000e+02
6.120e+02	2.460e+02	2.280e+02	3.440e+02	2.290e+02	1.790e+02	4.310e+02
1.650e+02	6.340e+02	5.310e+02	4.580e+02	3.570e+02	2.830e+02	1.750e+02
6.290e+02	8.470e+02	1.900e+02	5.910e+02	1.490e+02	8.580e+02	7.270e+02
2.650e+02	2.100e+02	3.180e+02	4.990e+02	2.970e+02	6.470e+02	2.360e+02
3.390e+02	3.940e+02	7.920e+02	1.670e+02	3.430e+02	4.460e+02	3.040e+02
1.920e+02	1.570e+02	2.480e+02	3.660e+02	1.194e+03	2.340e+02	2.490e+02
5.370e+02	2.610e+02	5.870e+02	3.230e+02	3.450e+02	6.060e+02	3.370e+02
3.770e+02	1.960e+02	9.240e+02	3.310e+02	5.180e+02	5.030e+02	2.050e+02
2.270e+02	3.170e+02	9.460e+02	3.560e+02	2.220e+02	2.200e+02	2.520e+02
3.280e+02	3.470e+02	5.150e+02	2.260e+02	9.040e+02	2.410e+02	2.010e+02
3.090e+02	2.300e+02	4.640e+02	3.620e+02	4.260e+02	3.650e+02	2.390e+02
2.510e+02	2.440e+02	6.230e+02	1.980e+02	3.750e+02	4.850e+02	1.120e+03
5.660e+02	2.880e+02	4.250e+02	6.150e+02	2.890e+02	2.730e+02	2.130e+02
2.740e+02	3.190e+02	3.860e+02	2.690e+02	2.950e+02	3.300e+02	5.800e+02
9.120e+02	5.340e+02	5.600e+02	8.050e+02	3.080e+02	7.030e+02	5.720e+02
4.790e+02	2.910e+02	1.292e+03	6.040e+02	3.790e+02	7.350e+02	2.190e+02
8.430e+02	4.160e+02	2.630e+02	2.840e+02	3.210e+02	3.380e+02	4.050e+02
1.037e+03	5.140e+02	5.070e+02	1.016e+03	5.840e+02	8.070e+02	6.600e+02
4.030e+02	8.040e+02	3.760e+02	3.800e+02	4.090e+02	5.160e+02	4.170e+02
3.520e+02	4.070e+02	4.730e+02	2.590e+02	3.250e+02	3.880e+02	3.410e+02
3.060e+02	3.120e+02	4.520e+02	6.660e+02	3.110e+02	2.820e+02	6.350e+02

```
 3.400e+02 8.800e+02 3.200e+02 3.980e+02 7.370e+02 5.430e+02 7.450e+02
 5.700e+02 3.720e+02 2.680e+02 3.690e+02 3.480e+02 5.230e+02 3.220e+02
 8.310e+02 4.370e+02 1.235e+03 8.200e+02 7.300e+02 5.590e+02 4.200e+02
 1.137e+03 3.550e+02 1.878e+03 3.680e+02 3.590e+02 4.020e+02 3.600e+02
 2.900e+02 5.490e+02 1.085e+03 4.980e+02 7.230e+02 5.330e+02 7.470e+02
 4.610e+02 1.404e+03 9.310e+02 3.320e+02 4.480e+02 4.060e+02 3.260e+02
 2.940e+02 3.920e+02 2.790e+02 3.340e+02 1.138e+03 2.470e+02 4.760e+02
 8.030e+02 9.640e+02 3.500e+02 4.290e+02 3.360e+02 1.031e+03 5.750e+02
 3.710e+02 9.480e+02 7.580e+02 3.810e+02 4.430e+02 1.018e+03 3.640e+02
 7.800e+02 5.760e+02 4.350e+02 5.080e+02 4.680e+02 7.160e+02 4.650e+02
 3.160e+02 8.760e+02 3.930e+02 3.850e+02 6.430e+02 7.260e+02 5.010e+02
 6.580e+02 4.230e+02 4.190e+02 4.860e+02 5.470e+02 1.088e+03 6.300e+02
 3.150e+02 5.860e+02 2.770e+02 5.090e+02 2.580e+02 1.628e+03 6.620e+02
 2.760e+02 9.130e+02 7.440e+02 6.900e+02 8.930e+02 5.620e+02 3.970e+02
 6.910e+02 7.790e+02 1.105e+03 1.015e+03 4.810e+02 5.270e+02 8.500e+02
 6.980e+02 6.930e+02 7.510e+02 3.740e+02 9.980e+02 3.510e+02 3.580e+02
 1.094e+03 7.890e+02 4.600e+02 7.250e+02 5.690e+02 7.320e+02 8.710e+02
 5.040e+02 5.940e+02 5.970e+02 6.520e+02 7.050e+02 1.185e+03 4.080e+02
 7.940e+02 9.680e+02 8.870e+02 1.458e+03 8.550e+02 3.000e+02 4.910e+02
 1.154e+03 1.402e+03 8.720e+02 3.830e+02 6.050e+02 4.340e+02 3.610e+02
 7.760e+02 5.680e+02 4.890e+02 7.000e+02 4.470e+02 3.780e+02 1.079e+03
 1.038e+03 4.950e+02 6.800e+02 9.160e+02 5.290e+02 8.850e+02 1.532e+03
 4.550e+02 7.460e+02 1.034e+03 3.050e+02 3.670e+02 1.099e+03 8.270e+02
 4.820e+02 7.810e+02 8.110e+02 5.420e+02 6.090e+02 6.990e+02 5.400e+02
 4.280e+02 4.590e+02 4.770e+02 8.890e+02 6.650e+02 6.560e+02 1.025e+03
 4.390e+02 3.070e+02 5.820e+02 5.110e+02 9.020e+02 6.270e+02 8.280e+02
 6.790e+02 8.940e+02 3.490e+02 4.010e+02 3.990e+02 9.850e+02 7.500e+02
 3.820e+02 1.145e+03 3.420e+02 3.890e+02 3.240e+02 2.920e+02 7.930e+02
 4.940e+02 4.210e+02 3.540e+02 6.370e+02 4.400e+02 9.290e+02 1.238e+03
 6.410e+02 1.108e+03 6.530e+02 7.750e+02 7.310e+02 1.071e+03 9.820e+02
 5.780e+02 4.140e+02 3.840e+02 4.740e+02 9.650e+02 6.570e+02 8.350e+02
 4.700e+02 8.610e+02 6.390e+02 1.467e+03 4.800e+02 8.510e+02 4.100e+02
 4.440e+02 7.090e+02 6.140e+02 8.010e+02 9.180e+02 4.840e+02 1.316e+03
 1.280e+03 9.390e+02 9.300e+02 5.960e+02 5.440e+02 4.500e+02 4.220e+02
 8.210e+02 5.000e+02 1.068e+03 4.270e+02 7.880e+02 9.410e+02 4.560e+02
 6.030e+02 5.350e+02 8.480e+02 5.300e+02 1.006e+03 6.780e+02 5.770e+02
 1.007e+03 5.380e+02 4.710e+02 5.710e+02 9.490e+02 4.670e+02]
----------------------------------
WeatherDelay has 395 unique values:
----------------------------------
[      nan 0.000e+00 3.700e+01 1.900e+01 4.800e+01 4.400e+01 2.700e+01
 9.000e+00 1.000e+00 1.560e+02 1.700e+01 3.900e+01 8.000e+00 4.000e+01
 5.400e+01 1.500e+01 5.000e+00 6.100e+01 4.900e+01 7.400e+01 7.000e+00
 2.200e+01 1.620e+02 2.000e+00 5.000e+01 4.700e+01 8.500e+01 3.500e+01
 3.200e+01 6.000e+00 3.800e+01 2.500e+01 9.400e+01 3.000e+01 1.400e+01
 3.000e+00 2.400e+01 1.950e+02 1.600e+01 1.200e+01 4.000e+00 3.600e+01
 5.700e+01 1.990e+02 4.200e+01 5.470e+02 1.080e+02 5.800e+01 8.400e+01
 2.190e+02 1.650e+02 1.800e+01 5.300e+01 2.100e+01 6.880e+02 6.500e+01
 9.700e+01 5.200e+01 1.300e+01 8.300e+01 1.260e+02 1.160e+02 4.300e+01
```

```
 2.900e+01 8.600e+01 6.600e+01 1.100e+01 1.930e+02 1.520e+02 2.000e+01
 4.100e+01 1.310e+02 6.900e+01 1.070e+02 9.300e+01 1.380e+02 6.200e+01
 1.000e+01 7.200e+01 3.100e+01 1.110e+02 1.860e+02 1.150e+02 8.000e+01
 5.100e+01 4.500e+01 3.400e+01 1.830e+02 5.900e+01 7.600e+01 1.680e+02
 1.030e+02 1.290e+02 5.600e+01 2.800e+01 6.300e+01 2.590e+02 1.240e+02
 2.110e+02 7.900e+01 1.350e+02 1.200e+02 1.800e+02 2.130e+02 9.100e+01
 1.510e+02 6.000e+01 1.500e+02 5.500e+01 6.700e+01 3.300e+01 7.000e+01
 1.660e+02 1.420e+02 4.600e+01 3.890e+02 2.460e+02 7.800e+01 8.700e+01
 1.920e+02 1.600e+02 1.850e+02 2.170e+02 3.230e+02 2.600e+01 9.500e+01
 1.780e+02 1.340e+02 2.270e+02 1.790e+02 1.130e+02 1.090e+02 1.450e+02
 2.140e+02 7.700e+01 1.490e+02 7.300e+01 1.020e+02 2.300e+01 2.760e+02
 1.460e+02 1.440e+02 7.100e+01 1.470e+02 1.040e+02 1.580e+02 6.400e+01
 1.610e+02 1.400e+02 2.400e+02 1.550e+02 1.590e+02 5.450e+02 1.630e+02
 9.800e+01 1.690e+02 2.670e+02 8.200e+01 1.640e+02 9.900e+01 1.140e+02
 1.360e+02 2.010e+02 1.270e+02 1.740e+02 7.500e+01 1.540e+02 1.000e+02
 9.000e+01 2.640e+02 6.800e+01 2.050e+02 1.810e+02 1.153e+03 7.040e+02
 2.290e+02 1.280e+02 1.530e+02 1.330e+02 1.910e+02 1.170e+02 1.230e+02
 1.670e+02 6.100e+02 2.240e+02 1.300e+02 1.010e+02 8.100e+01 1.190e+02
 9.200e+01 2.660e+02 2.840e+02 2.770e+02 2.120e+02 2.000e+02 2.100e+02
 2.420e+02 4.030e+02 1.120e+02 1.880e+02 4.490e+02 1.940e+02 3.030e+02
 1.060e+02 1.970e+02 2.250e+02 2.060e+02 2.790e+02 7.380e+02 1.220e+02
 1.770e+02 8.800e+01 3.500e+02 3.020e+02 1.430e+02 2.580e+02 1.050e+02
 1.410e+02 9.600e+01 6.870e+02 1.320e+02 2.330e+02 1.180e+02 2.510e+02
 2.070e+02 7.120e+02 1.250e+02 2.040e+02 2.370e+02 1.820e+02 1.720e+02
 8.900e+01 1.890e+02 3.120e+02 4.430e+02 9.310e+02 3.170e+02 2.180e+02
 1.980e+02 2.480e+02 1.900e+02 6.060e+02 1.210e+02 2.500e+02 2.280e+02
 1.100e+02 1.390e+02 5.210e+02 3.110e+02 6.650e+02 1.710e+02 2.980e+02
 2.650e+02 2.030e+02 1.870e+02 5.230e+02 3.100e+02 2.150e+02 2.160e+02
 3.520e+02 7.650e+02 1.847e+03 1.480e+02 2.730e+02 2.360e+02 3.770e+02
 6.760e+02 2.340e+02 2.600e+02 2.430e+02 3.040e+02 2.410e+02 3.630e+02
 2.200e+02 1.750e+02 3.400e+02 8.270e+02 2.020e+02 3.090e+02 2.690e+02
 1.730e+02 2.560e+02 2.380e+02 1.293e+03 2.090e+02 1.370e+02 2.850e+02
 2.220e+02 2.390e+02 2.630e+02 2.830e+02 3.840e+02 2.810e+02 3.700e+02
 2.210e+02 7.200e+02 6.540e+02 3.000e+02 2.230e+02 5.020e+02 1.223e+03
 2.720e+02 3.420e+02 5.000e+02 1.760e+02 4.160e+02 2.320e+02 8.590e+02
 1.700e+02 1.960e+02 2.300e+02 7.720e+02 2.920e+02 2.530e+02 3.080e+02
 1.019e+03 6.850e+02 2.890e+02 3.680e+02 2.820e+02 7.630e+02 3.670e+02
 1.570e+02 3.990e+02 4.080e+02 4.010e+02 2.910e+02 3.050e+02 3.780e+02
 2.490e+02 2.080e+02 3.450e+02 4.290e+02 5.310e+02 2.700e+02 8.970e+02
 5.970e+02 9.380e+02 3.350e+02 9.770e+02 3.270e+02 3.640e+02 2.680e+02
 6.990e+02 8.360e+02 2.780e+02 3.950e+02 2.990e+02 3.480e+02 4.840e+02
 2.310e+02 2.470e+02 6.790e+02 2.940e+02 1.066e+03 2.960e+02 3.370e+02
 6.260e+02 4.100e+02 6.180e+02 6.000e+02 3.510e+02 7.410e+02 3.060e+02
 2.710e+02 4.320e+02 2.450e+02 9.560e+02 4.410e+02 2.610e+02 1.410e+03
 2.520e+02 2.800e+02 5.740e+02 1.840e+02 3.240e+02 3.730e+02 4.700e+02
 3.980e+02 6.750e+02 4.650e+02 5.380e+02 3.820e+02 6.190e+02 6.970e+02
 4.510e+02 3.200e+02 3.250e+02 3.130e+02 2.860e+02 9.510e+02 4.360e+02
 3.220e+02 5.880e+02 3.540e+02]
----------------------------------
```

```
NASDelay has 430 unique values:
----------------------------------
[      nan 0.000e+00 1.300e+01 7.000e+00 3.000e+00 2.400e+01 6.000e+00
 5.000e+00 1.600e+01 1.700e+01 1.900e+01 5.700e+01 1.500e+01 5.900e+01
 6.100e+01 1.440e+02 4.000e+00 9.400e+01 4.400e+01 3.700e+01 8.000e+00
 1.100e+01 1.800e+01 4.000e+01 2.000e+00 2.300e+01 4.700e+01 2.000e+01
 2.500e+01 5.300e+01 5.500e+01 9.000e+00 1.290e+02 4.500e+01 3.200e+01
 1.200e+01 6.700e+01 1.000e+02 2.100e+01 3.900e+01 2.800e+01 9.500e+01
 6.600e+01 8.600e+01 1.000e+00 4.600e+01 2.700e+01 3.500e+01 8.000e+01
 1.000e+01 1.400e+01 3.100e+01 4.300e+01 3.800e+01 3.000e+01 1.190e+02
 1.010e+02 3.600e+01 2.200e+01 2.600e+01 5.400e+01 2.790e+02 6.800e+01
 6.500e+01 3.400e+01 1.020e+02 1.460e+02 1.250e+02 5.600e+01 1.110e+02
 2.090e+02 4.040e+02 8.100e+01 3.300e+01 1.080e+02 5.000e+01 6.400e+01
 1.750e+02 6.000e+01 3.010e+02 8.900e+01 8.500e+01 3.230e+02 6.900e+01
 7.500e+01 1.050e+02 1.400e+02 4.900e+01 1.030e+02 4.100e+01 2.710e+02
 4.800e+01 2.900e+01 7.300e+01 4.200e+01 1.730e+02 1.840e+02 5.200e+01
 5.100e+01 2.250e+02 8.800e+01 5.800e+01 1.630e+02 6.200e+01 8.400e+01
 1.120e+02 1.170e+02 9.100e+01 8.300e+01 7.700e+01 9.300e+01 7.800e+01
 1.530e+02 6.300e+01 7.400e+01 2.400e+02 8.700e+01 1.150e+02 1.330e+02
 2.990e+02 1.920e+02 1.600e+02 9.800e+01 7.100e+01 1.350e+02 1.480e+02
 3.310e+02 1.090e+02 9.600e+01 1.430e+02 7.600e+01 1.760e+02 1.130e+02
 1.070e+02 1.040e+02 2.160e+02 1.270e+02 1.340e+02 1.470e+02 1.230e+02
 1.810e+02 1.510e+02 1.850e+02 1.420e+02 1.870e+02 1.310e+02 9.700e+01
 1.880e+02 1.370e+02 1.180e+02 1.100e+02 4.650e+02 7.000e+01 1.240e+02
 1.060e+02 7.200e+01 1.410e+02 8.200e+01 1.860e+02 1.960e+02 2.500e+02
 1.540e+02 1.140e+02 1.720e+02 6.710e+02 1.160e+02 7.900e+01 1.450e+02
 1.300e+02 2.940e+02 2.190e+02 1.260e+02 1.500e+02 1.710e+02 9.900e+01
 1.680e+02 2.100e+02 2.040e+02 2.280e+02 1.320e+02 2.420e+02 1.520e+02
 2.960e+02 2.080e+02 1.200e+02 9.000e+01 1.620e+02 1.570e+02 2.430e+02
 2.930e+02 2.690e+02 2.640e+02 1.380e+02 2.030e+02 1.280e+02 1.490e+02
 9.200e+01 2.000e+02 2.300e+02 1.210e+02 2.560e+02 1.990e+02 1.640e+02
 1.940e+02 1.660e+02 1.590e+02 2.750e+02 1.740e+02 2.720e+02 2.760e+02
 1.950e+02 1.610e+02 2.570e+02 3.330e+02 3.510e+02 1.220e+02 1.670e+02
 3.400e+02 3.070e+02 1.790e+02 2.310e+02 3.120e+02 2.140e+02 3.890e+02
 1.560e+02 3.490e+02 2.260e+02 2.230e+02 2.340e+02 2.780e+02 2.070e+02
 1.930e+02 2.980e+02 2.970e+02 1.580e+02 2.050e+02 1.390e+02 1.360e+02
 1.890e+02 3.130e+02 2.830e+02 1.900e+02 2.020e+02 2.110e+02 2.450e+02
 3.220e+02 5.360e+02 2.240e+02 2.870e+02 2.460e+02 2.530e+02 1.690e+02
 2.180e+02 2.130e+02 2.270e+02 1.970e+02 4.070e+02 1.770e+02 4.660e+02
 1.980e+02 1.700e+02 4.780e+02 1.650e+02 2.480e+02 6.910e+02 1.830e+02
 2.680e+02 2.370e+02 2.010e+02 2.670e+02 3.100e+02 3.370e+02 2.060e+02
 4.200e+02 1.800e+02 1.820e+02 3.640e+02 1.550e+02 2.650e+02 2.350e+02
 2.380e+02 1.910e+02 2.360e+02 2.540e+02 3.770e+02 2.210e+02 1.194e+03
 3.470e+02 2.890e+02 2.330e+02 2.510e+02 3.540e+02 3.030e+02 2.220e+02
 3.280e+02 2.490e+02 2.800e+02 2.120e+02 2.320e+02 3.300e+02 3.760e+02
 2.410e+02 5.330e+02 2.950e+02 3.730e+02 2.390e+02 2.590e+02 2.200e+02
 3.080e+02 4.540e+02 2.630e+02 6.750e+02 3.700e+02 3.450e+02 2.290e+02
 9.440e+02 3.170e+02 3.880e+02 5.780e+02 2.600e+02 1.780e+02 3.190e+02
 3.480e+02 3.260e+02 2.740e+02 2.150e+02 3.390e+02 3.320e+02 2.810e+02
```

```
 3.140e+02 7.010e+02 1.053e+03 2.170e+02 3.830e+02 2.880e+02 2.770e+02
 3.180e+02 4.620e+02 2.730e+02 6.030e+02 3.290e+02 3.090e+02 4.140e+02
 8.950e+02 3.460e+02 6.760e+02 4.430e+02 2.700e+02 2.580e+02 2.620e+02
 2.660e+02 3.250e+02 4.470e+02 3.040e+02 2.470e+02 7.270e+02 2.860e+02
 4.710e+02 4.180e+02 7.100e+02 3.560e+02 2.820e+02 2.610e+02 5.510e+02
 2.910e+02 2.520e+02 3.740e+02 7.880e+02 6.180e+02 8.060e+02 3.870e+02
 3.900e+02 3.240e+02 2.550e+02 3.360e+02 3.630e+02 4.340e+02 4.030e+02
 3.590e+02 3.340e+02 8.520e+02 9.260e+02 4.010e+02 3.690e+02 3.410e+02
 5.630e+02 4.810e+02 5.130e+02 3.600e+02 4.130e+02 4.120e+02 2.840e+02
 9.520e+02 3.020e+02 9.840e+02 3.050e+02 3.000e+02 3.200e+02 3.750e+02
 3.580e+02 3.810e+02 4.150e+02 1.343e+03 5.180e+02 3.350e+02 5.900e+02
 4.050e+02 3.060e+02 3.270e+02 1.008e+03 3.520e+02 3.710e+02 2.850e+02
 6.820e+02 4.160e+02 8.580e+02 6.790e+02 4.410e+02 3.150e+02 4.110e+02
 4.530e+02 4.690e+02 3.210e+02 4.000e+02 3.570e+02 4.320e+02 3.420e+02
 4.020e+02 5.320e+02 3.430e+02]
----------------------------------
SecurityDelay has 100 unique values:
----------------------------------
[ nan   0.   6.  82.   8.  57.  11.  25.  26.  10.  20.   3.  23.  16.
 148.   1.  30.  12.   4.   5.  75.   7.  24.   2.   9.  17.  60.  44.
  21.  18. 208.  28.  29.  19.  15.  62.  42.  37.  22. 168.  93.  14.
  36.  13.  32.  39.  54.  56.  86. 199.  40.  38.  48.  41. 159.  27.
 106. 115.  35.  46.  53.  43.  88.  83. 219.  31. 119.  47.  92.  94.
  51.  80.  85.  52.  70.  49. 124.  45.  90.  33.  77.  66. 214.  59.
 113. 131. 180. 102. 117.  34.  58.  96.  68.  67.  98. 123.  73.  84.
  72.  71.]
----------------------------------
LateAircraftDelay has 491 unique values:
----------------------------------
[      nan 3.200e+01 2.140e+02 1.600e+01 0.000e+00 1.000e+00 2.100e+01
 1.170e+02 8.000e+00 1.790e+02 8.400e+01 2.700e+01 1.800e+01 2.200e+01
 3.800e+01 1.100e+01 3.300e+01 3.000e+01 1.000e+01 1.310e+02 7.200e+01
 1.900e+01 4.200e+01 1.750e+02 4.300e+01 3.400e+01 1.200e+01 2.000e+01
 1.760e+02 1.930e+02 2.900e+01 6.000e+00 2.800e+01 3.000e+00 8.000e+01
 8.300e+01 1.330e+02 1.700e+01 1.500e+01 7.000e+00 5.000e+00 5.200e+01
 5.800e+01 5.500e+01 9.000e+00 2.300e+01 4.400e+01 7.900e+01 8.900e+01
 6.200e+01 2.370e+02 6.800e+01 4.100e+01 6.000e+01 5.700e+01 4.000e+01
 3.700e+01 1.350e+02 2.800e+02 8.100e+01 4.800e+01 5.100e+01 9.000e+01
 1.190e+02 5.600e+01 9.500e+01 6.100e+01 1.070e+02 4.700e+01 7.300e+01
 2.600e+01 9.800e+01 1.300e+02 3.600e+01 4.500e+01 3.360e+02 3.100e+01
 2.000e+00 7.400e+01 4.900e+01 1.680e+02 1.050e+02 5.400e+01 9.400e+01
 1.300e+01 2.410e+02 1.080e+02 2.500e+01 2.400e+01 9.300e+01 8.200e+01
 8.800e+01 1.400e+01 1.220e+02 3.900e+01 9.100e+01 6.700e+01 1.040e+02
 2.170e+02 7.800e+01 7.000e+01 1.320e+02 3.500e+01 1.650e+02 2.160e+02
 6.300e+01 6.500e+01 4.000e+00 5.900e+01 1.400e+02 9.900e+01 2.070e+02
 1.630e+02 1.360e+02 7.500e+01 1.060e+02 1.620e+02 1.100e+02 9.200e+01
 1.580e+02 1.780e+02 4.340e+02 1.700e+02 6.400e+01 1.180e+02 8.700e+01
 4.600e+01 1.410e+02 1.240e+02 1.290e+02 1.150e+02 5.000e+01 1.000e+02
 2.320e+02 8.600e+01 2.360e+02 2.060e+02 5.300e+01 6.900e+01 9.600e+01
```

1.160e+02	1.230e+02	2.040e+02	3.720e+02	1.120e+02	1.590e+02	1.890e+02
1.370e+02	2.690e+02	9.700e+01	1.770e+02	1.510e+02	2.330e+02	1.530e+02
1.010e+02	1.820e+02	2.500e+02	7.600e+01	7.100e+01	1.740e+02	8.500e+01
1.480e+02	1.280e+02	1.380e+02	1.270e+02	1.030e+02	1.020e+02	1.610e+02
2.340e+02	2.080e+02	1.260e+02	1.200e+02	1.130e+02	1.250e+02	1.860e+02
1.720e+02	1.110e+02	1.950e+02	1.450e+02	1.490e+02	1.430e+02	1.550e+02
2.050e+02	1.210e+02	1.470e+02	2.270e+02	7.700e+01	2.510e+02	2.750e+02
1.690e+02	1.090e+02	1.440e+02	2.250e+02	1.660e+02	1.810e+02	1.870e+02
1.500e+02	1.540e+02	2.200e+02	3.180e+02	1.830e+02	1.570e+02	1.970e+02
2.980e+02	2.940e+02	1.420e+02	2.560e+02	2.110e+02	6.600e+01	2.710e+02
1.140e+02	1.390e+02	2.290e+02	2.010e+02	2.190e+02	1.520e+02	2.420e+02
2.280e+02	1.460e+02	2.440e+02	2.460e+02	2.550e+02	1.600e+02	2.610e+02
1.670e+02	2.490e+02	1.560e+02	2.990e+02	4.070e+02	1.850e+02	1.840e+02
1.980e+02	3.660e+02	1.640e+02	3.840e+02	1.340e+02	1.990e+02	2.210e+02
2.230e+02	3.570e+02	2.930e+02	3.350e+02	4.270e+02	1.730e+02	2.150e+02
2.470e+02	2.390e+02	2.540e+02	2.530e+02	2.920e+02	2.030e+02	1.880e+02
1.800e+02	3.970e+02	1.710e+02	3.990e+02	1.900e+02	1.940e+02	2.630e+02
3.040e+02	2.790e+02	2.660e+02	2.020e+02	3.010e+02	2.620e+02	2.700e+02
1.960e+02	1.920e+02	2.300e+02	3.940e+02	3.080e+02	2.770e+02	3.170e+02
3.150e+02	2.260e+02	2.680e+02	2.220e+02	2.650e+02	1.910e+02	2.590e+02
2.000e+02	4.180e+02	3.470e+02	3.300e+02	2.120e+02	3.620e+02	2.480e+02
2.520e+02	2.830e+02	5.280e+02	3.670e+02	2.450e+02	2.640e+02	3.680e+02
3.140e+02	4.220e+02	2.090e+02	2.970e+02	2.180e+02	2.240e+02	2.380e+02
3.240e+02	2.670e+02	5.230e+02	2.570e+02	2.820e+02	3.370e+02	3.770e+02
3.870e+02	2.950e+02	3.600e+02	4.360e+02	2.740e+02	4.540e+02	3.280e+02
2.960e+02	3.110e+02	9.120e+02	2.100e+02	2.900e+02	2.310e+02	4.000e+02
7.830e+02	6.730e+02	3.230e+02	3.130e+02	2.400e+02	3.100e+02	3.290e+02
3.390e+02	2.870e+02	3.550e+02	4.130e+02	3.960e+02	2.850e+02	2.910e+02
2.780e+02	4.280e+02	4.430e+02	2.350e+02	2.600e+02	5.700e+02	5.390e+02
2.130e+02	8.010e+02	3.000e+02	6.100e+02	2.840e+02	3.410e+02	2.760e+02
3.190e+02	3.950e+02	4.290e+02	3.090e+02	2.580e+02	4.460e+02	3.530e+02
2.720e+02	6.120e+02	6.480e+02	3.310e+02	3.590e+02	3.500e+02	3.120e+02
4.480e+02	3.380e+02	3.510e+02	4.440e+02	5.370e+02	3.250e+02	3.580e+02
3.650e+02	4.740e+02	1.407e+03	4.850e+02	5.150e+02	4.230e+02	3.480e+02
5.690e+02	5.190e+02	3.070e+02	3.910e+02	4.500e+02	5.110e+02	4.890e+02
4.410e+02	6.220e+02	3.760e+02	3.030e+02	2.730e+02	2.880e+02	5.560e+02
2.860e+02	3.050e+02	8.240e+02	3.020e+02	4.300e+02	3.640e+02	2.890e+02
3.690e+02	4.030e+02	4.020e+02	3.160e+02	4.420e+02	3.810e+02	4.620e+02
3.270e+02	7.950e+02	5.120e+02	3.980e+02	3.400e+02	3.490e+02	2.810e+02
4.570e+02	3.820e+02	3.520e+02	3.860e+02	2.430e+02	4.660e+02	3.260e+02
1.013e+03	8.620e+02	3.060e+02	5.580e+02	3.450e+02	7.270e+02	4.530e+02
7.320e+02	4.940e+02	3.330e+02	4.520e+02	4.990e+02	3.830e+02	3.320e+02
7.440e+02	1.256e+03	3.430e+02	4.870e+02	5.030e+02	4.970e+02	4.790e+02
5.790e+02	4.950e+02	4.250e+02	3.610e+02	3.200e+02	3.220e+02	3.630e+02
5.130e+02	6.440e+02	4.320e+02	3.560e+02	1.054e+03	5.510e+02	6.560e+02
4.260e+02	4.240e+02	5.640e+02	3.790e+02	1.173e+03	5.570e+02	4.350e+02
8.250e+02	3.710e+02	3.210e+02	3.930e+02	7.090e+02	4.630e+02	5.600e+02
7.300e+02	3.420e+02	4.080e+02	4.040e+02	3.440e+02	8.420e+02	4.750e+02
5.180e+02	6.650e+02	8.190e+02	3.900e+02	8.690e+02	4.050e+02	3.740e+02

```
3.540e+02 3.700e+02 4.490e+02 4.700e+02 3.460e+02 3.920e+02 4.580e+02
3.780e+02]
```

from checking the unique values, it was noticed the nan value in 'CarrierDelay','WeatherDelay','NASDelay', 'SecurityDelay','LateAircraftDelay' so it will be replaced with zero

```
cleaned_airline_df[cleaned_airline_df['Cancelled'] == 1]
['CancellationCode'].unique()

array(['A', nan, 'B', 'C', 'D'], dtype=object)
```

It is noticed that there exists null values for cancellation code when the trip is cancelled. we need to fix this data

Replace Null Values

```
cleaned_airline_df['CarrierDelay'].fillna(0, inplace=True)
cleaned_airline_df['WeatherDelay'].fillna(0, inplace=True)
cleaned_airline_df['NASDelay'].fillna(0, inplace=True)
cleaned_airline_df['SecurityDelay'].fillna(0, inplace=True)
cleaned_airline_df['LateAircraftDelay'].fillna(0, inplace=True)
cleaned_airline_df.loc[cleaned_airline_df['Cancelled'] ==
1,'CancellationCode'] =
cleaned_airline_df.loc[cleaned_airline_df['Cancelled'] ==
1,'CancellationCode'].fillna('Not Defined')

cleaned_airline_df[['CarrierDelay',
                    'WeatherDelay',
                    'NASDelay','SecurityDelay',
                    'LateAircraftDelay']].describe()

       CarrierDelay  WeatherDelay      NASDelay  SecurityDelay  \
count  2.000000e+06  2.000000e+06  2.000000e+06   2.000000e+06
mean   1.873412e+00  3.260425e-01  1.706707e+00   9.412500e-03
std    1.628119e+01  7.087432e+00  1.125969e+01   7.029902e-01
min    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
50%    0.000000e+00  0.000000e+00  0.000000e+00   0.000000e+00
75%    0.000000e+00  0.000000e+00  0.000000e+00   0.000000e+00
max    1.878000e+03  1.847000e+03  1.343000e+03   2.190000e+02

       LateAircraftDelay
count       2.000000e+06
mean        2.445841e+00
std         1.549742e+01
min         0.000000e+00
25%         0.000000e+00
50%         0.000000e+00
```

```
75%           0.000000e+00
max           1.407000e+03
```

The min value is zero

```
check_unique_values(df=cleaned_airline_df,columns=[
    'CarrierDelay','WeatherDelay','NASDelay',
    'SecurityDelay','LateAircraftDelay'])
```

```
Number of unique values per column
----------------------------------
CarrierDelay has 705 unique values:
----------------------------------
[0.000e+00 4.400e+01 9.000e+00 1.000e+00 2.000e+00 2.400e+01 3.600e+01
 2.600e+01 1.000e+01 1.200e+01 2.200e+01 7.000e+00 4.200e+01 3.000e+01
 4.000e+00 6.000e+00 1.100e+01 1.500e+01 4.300e+01 1.130e+02 3.400e+01
 1.800e+01 2.100e+01 5.000e+00 4.100e+01 2.000e+01 3.200e+01 6.400e+02
 3.100e+01 1.400e+01 1.600e+01 8.000e+00 6.100e+01 2.300e+01 1.330e+02
 1.120e+02 8.600e+01 1.280e+02 2.500e+01 3.300e+01 7.600e+01 8.500e+01
 5.500e+01 7.000e+01 5.200e+01 3.000e+00 4.900e+01 1.050e+02 3.500e+01
 3.700e+01 7.700e+01 2.900e+01 1.260e+02 1.300e+01 4.500e+01 2.980e+02
 2.430e+02 1.930e+02 1.900e+01 5.700e+01 9.200e+01 2.700e+01 9.500e+01
 7.100e+01 4.600e+01 1.800e+02 2.800e+01 2.160e+02 7.500e+01 3.800e+01
 6.900e+01 5.320e+02 4.690e+02 6.500e+01 8.300e+01 4.000e+01 5.540e+02
 1.700e+01 6.400e+01 1.680e+02 3.900e+01 7.900e+01 1.410e+02 1.110e+02
 1.610e+02 1.300e+02 1.390e+02 2.550e+02 1.630e+02 4.700e+01 9.540e+02
 4.800e+01 6.700e+01 6.200e+01 1.150e+02 2.180e+02 2.560e+02 2.150e+02
 1.040e+02 5.300e+01 1.470e+02 5.800e+01 7.200e+01 7.300e+01 1.380e+02
 1.160e+02 2.870e+02 5.600e+01 1.200e+02 9.300e+01 9.900e+01 1.030e+02
 1.080e+02 5.000e+01 2.250e+02 4.630e+02 6.600e+01 8.700e+01 2.420e+02
 5.400e+01 6.300e+01 3.270e+02 8.400e+01 1.020e+02 5.100e+01 2.710e+02
 1.860e+02 8.000e+01 8.900e+01 5.900e+01 5.050e+02 7.400e+01 7.800e+01
 4.830e+02 1.520e+02 1.450e+02 8.520e+02 1.170e+02 6.000e+01 1.500e+02
 1.990e+02 1.810e+02 1.360e+02 4.450e+02 1.660e+02 2.990e+02 4.540e+02
 1.840e+02 1.090e+02 1.420e+02 1.640e+02 1.070e+02 2.660e+02 8.800e+01
 5.360e+02 1.240e+02 3.020e+02 1.190e+02 3.100e+02 8.200e+01 2.960e+02
 6.800e+01 9.700e+01 2.450e+02 8.100e+01 1.560e+02 9.100e+01 1.620e+02
 4.300e+02 1.700e+02 1.510e+02 2.110e+02 9.400e+01 6.020e+02 1.820e+02
 9.000e+01 2.780e+02 1.590e+02 1.440e+02 1.022e+03 9.600e+01 2.400e+02
 2.670e+02 1.180e+02 2.040e+02 1.010e+02 2.600e+02 1.460e+02 1.350e+02
 3.330e+02 1.060e+02 1.550e+02 1.100e+02 9.800e+01 1.690e+02 1.780e+02
 2.930e+02 2.540e+02 3.030e+02 3.140e+02 2.330e+02 5.170e+02 1.000e+02
 3.630e+02 1.220e+02 1.340e+02 2.570e+02 1.270e+02 1.950e+02 2.060e+02
 5.900e+02 6.950e+02 5.730e+02 1.910e+02 1.710e+02 9.220e+02 3.290e+02
 1.580e+02 1.890e+02 7.360e+02 7.380e+02 1.290e+02 9.920e+02 2.850e+02
 2.210e+02 1.430e+02 1.770e+02 2.090e+02 1.530e+02 1.760e+02 1.400e+02
 1.600e+02 3.350e+02 1.940e+02 1.250e+02 2.720e+02 2.860e+02 9.760e+02
 1.870e+02 2.380e+02 2.240e+02 3.010e+02 1.540e+02 2.120e+02 1.125e+03
 3.130e+02 2.030e+02 4.180e+02 3.870e+02 2.080e+02 2.750e+02 1.370e+02
```

2.530e+02	1.880e+02	2.500e+02	2.170e+02	2.020e+02	2.000e+02	1.740e+02
1.210e+02	2.640e+02	1.320e+02	2.320e+02	3.530e+02	1.480e+02	2.370e+02
7.720e+02	2.810e+02	2.350e+02	1.830e+02	2.310e+02	1.730e+02	3.910e+02
1.140e+02	1.310e+02	2.800e+02	1.069e+03	4.880e+02	2.070e+02	1.230e+02
2.620e+02	8.080e+02	5.190e+02	4.040e+02	2.230e+02	4.410e+02	4.320e+02
1.720e+02	1.970e+02	5.990e+02	2.700e+02	8.920e+02	6.100e+02	2.140e+02
5.200e+02	3.460e+02	1.850e+02	4.150e+02	3.730e+02	4.000e+02	6.120e+02
2.460e+02	2.280e+02	3.440e+02	2.290e+02	1.790e+02	4.310e+02	1.650e+02
6.340e+02	5.310e+02	4.580e+02	3.570e+02	2.830e+02	1.750e+02	6.290e+02
8.470e+02	1.900e+02	5.910e+02	1.490e+02	8.580e+02	7.270e+02	2.650e+02
2.100e+02	3.180e+02	4.990e+02	2.970e+02	6.470e+02	2.360e+02	3.390e+02
3.940e+02	7.920e+02	1.670e+02	3.430e+02	4.460e+02	3.040e+02	1.920e+02
1.570e+02	2.480e+02	3.660e+02	1.194e+03	2.340e+02	2.490e+02	5.370e+02
2.610e+02	5.870e+02	3.230e+02	3.450e+02	6.060e+02	3.370e+02	3.770e+02
1.960e+02	9.240e+02	3.310e+02	5.180e+02	5.030e+02	2.050e+02	2.270e+02
3.170e+02	9.460e+02	3.560e+02	2.220e+02	2.200e+02	2.520e+02	3.280e+02
3.470e+02	5.150e+02	2.260e+02	9.040e+02	2.410e+02	2.010e+02	3.090e+02
2.300e+02	4.640e+02	3.620e+02	4.260e+02	3.650e+02	2.390e+02	2.510e+02
2.440e+02	6.230e+02	1.980e+02	3.750e+02	4.850e+02	1.120e+03	5.660e+02
2.880e+02	4.250e+02	6.150e+02	2.890e+02	2.730e+02	2.130e+02	2.740e+02
3.190e+02	3.860e+02	2.690e+02	2.950e+02	3.300e+02	5.800e+02	9.120e+02
5.340e+02	5.600e+02	8.050e+02	3.080e+02	7.030e+02	5.720e+02	4.790e+02
2.910e+02	1.292e+03	6.040e+02	3.790e+02	7.350e+02	2.190e+02	8.430e+02
4.160e+02	2.630e+02	2.840e+02	3.210e+02	3.380e+02	4.050e+02	1.037e+03
5.140e+02	5.070e+02	1.016e+03	5.840e+02	8.070e+02	6.600e+02	4.030e+02
8.040e+02	3.760e+02	3.800e+02	4.090e+02	5.160e+02	4.170e+02	3.520e+02
4.070e+02	4.730e+02	2.590e+02	3.250e+02	3.880e+02	3.410e+02	3.060e+02
3.120e+02	4.520e+02	6.660e+02	3.110e+02	2.820e+02	6.350e+02	3.400e+02
8.800e+02	3.200e+02	3.980e+02	7.370e+02	5.430e+02	7.450e+02	5.700e+02
3.720e+02	2.680e+02	3.690e+02	3.480e+02	5.230e+02	3.220e+02	8.310e+02
4.370e+02	1.235e+03	8.200e+02	7.300e+02	5.590e+02	4.200e+02	1.137e+03
3.550e+02	1.878e+03	3.680e+02	3.590e+02	4.020e+02	3.600e+02	2.900e+02
5.490e+02	1.085e+03	4.980e+02	7.230e+02	5.330e+02	7.470e+02	4.610e+02
1.404e+03	9.310e+02	3.320e+02	4.480e+02	4.060e+02	3.260e+02	2.940e+02
3.920e+02	2.790e+02	3.340e+02	1.138e+03	2.470e+02	4.760e+02	8.030e+02
9.640e+02	3.500e+02	4.290e+02	3.360e+02	1.031e+03	5.750e+02	3.710e+02
9.480e+02	7.580e+02	3.810e+02	4.430e+02	1.018e+03	3.640e+02	7.800e+02
5.760e+02	4.350e+02	5.080e+02	4.680e+02	7.160e+02	4.650e+02	3.160e+02
8.760e+02	3.930e+02	3.850e+02	6.430e+02	7.260e+02	5.010e+02	6.580e+02
4.230e+02	4.190e+02	4.860e+02	5.470e+02	1.088e+03	6.300e+02	3.150e+02
5.860e+02	2.770e+02	5.090e+02	2.580e+02	1.628e+03	6.620e+02	2.760e+02
9.130e+02	7.440e+02	6.900e+02	8.930e+02	5.620e+02	3.970e+02	6.910e+02
7.790e+02	1.105e+03	1.015e+03	4.810e+02	5.270e+02	8.500e+02	6.980e+02
6.930e+02	7.510e+02	3.740e+02	9.980e+02	3.510e+02	3.580e+02	1.094e+03
7.890e+02	4.600e+02	7.250e+02	5.690e+02	7.320e+02	8.710e+02	5.040e+02
5.940e+02	5.970e+02	6.520e+02	7.050e+02	1.185e+03	4.080e+02	7.940e+02
9.680e+02	8.870e+02	1.458e+03	8.550e+02	3.000e+02	4.910e+02	1.154e+03
1.402e+03	8.720e+02	3.830e+02	6.050e+02	4.340e+02	3.610e+02	7.760e+02
5.680e+02	4.890e+02	7.000e+02	4.470e+02	3.780e+02	1.079e+03	1.038e+03

```
 4.950e+02 6.800e+02 9.160e+02 5.290e+02 8.850e+02 1.532e+03 4.550e+02
 7.460e+02 1.034e+03 3.050e+02 3.670e+02 1.099e+03 8.270e+02 4.820e+02
 7.810e+02 8.110e+02 5.420e+02 6.090e+02 6.990e+02 5.400e+02 4.280e+02
 4.590e+02 4.770e+02 8.890e+02 6.650e+02 6.560e+02 1.025e+03 4.390e+02
 3.070e+02 5.820e+02 5.110e+02 9.020e+02 6.270e+02 8.280e+02 6.790e+02
 8.940e+02 3.490e+02 4.010e+02 3.990e+02 9.850e+02 7.500e+02 3.820e+02
 1.145e+03 3.420e+02 3.890e+02 3.240e+02 2.920e+02 7.930e+02 4.940e+02
 4.210e+02 3.540e+02 6.370e+02 4.400e+02 9.290e+02 1.238e+03 6.410e+02
 1.108e+03 6.530e+02 7.750e+02 7.310e+02 1.071e+03 9.820e+02 5.780e+02
 4.140e+02 3.840e+02 4.740e+02 9.650e+02 6.570e+02 8.350e+02 4.700e+02
 8.610e+02 6.390e+02 1.467e+03 4.800e+02 8.510e+02 4.100e+02 4.440e+02
 7.090e+02 6.140e+02 8.010e+02 9.180e+02 4.840e+02 1.316e+03 1.280e+03
 9.390e+02 9.300e+02 5.960e+02 5.440e+02 4.500e+02 4.220e+02 8.210e+02
 5.000e+02 1.068e+03 4.270e+02 7.880e+02 9.410e+02 4.560e+02 6.030e+02
 5.350e+02 8.480e+02 5.300e+02 1.006e+03 6.780e+02 5.770e+02 1.007e+03
 5.380e+02 4.710e+02 5.710e+02 9.490e+02 4.670e+02]
------------------------------------
WeatherDelay has 394 unique values:
------------------------------------
[0.000e+00 3.700e+01 1.900e+01 4.800e+01 4.400e+01 2.700e+01 9.000e+00
 1.000e+00 1.560e+02 1.700e+01 3.900e+01 8.000e+00 4.000e+01 5.400e+01
 1.500e+01 5.000e+00 6.100e+01 4.900e+01 7.400e+01 7.000e+00 2.200e+01
 1.620e+02 2.000e+00 5.000e+01 4.700e+01 8.500e+01 3.500e+01 3.200e+01
 6.000e+00 3.800e+01 2.500e+01 9.400e+01 3.000e+01 1.400e+01 3.000e+00
 2.400e+01 1.950e+02 1.600e+01 1.200e+01 4.000e+00 3.600e+01 5.700e+01
 1.990e+02 4.200e+01 5.470e+02 1.080e+02 5.800e+01 8.400e+01 2.190e+02
 1.650e+02 1.800e+01 5.300e+01 2.100e+01 6.880e+02 6.500e+01 9.700e+01
 5.200e+01 1.300e+01 8.300e+01 1.260e+02 1.160e+02 4.300e+01 2.900e+01
 8.600e+01 6.600e+01 1.100e+01 1.930e+02 1.520e+02 2.000e+01 4.100e+01
 1.310e+02 6.900e+01 1.070e+02 9.300e+01 1.380e+02 6.200e+01 1.000e+01
 7.200e+01 3.100e+01 1.110e+02 1.860e+02 1.150e+02 8.000e+01 5.100e+01
 4.500e+01 3.400e+01 1.830e+02 5.900e+01 7.600e+01 1.680e+02 1.030e+02
 1.290e+02 5.600e+01 2.800e+01 6.300e+01 2.590e+02 1.240e+02 2.110e+02
 7.900e+01 1.350e+02 1.200e+02 1.800e+02 2.130e+02 9.100e+01 1.510e+02
 6.000e+01 1.500e+02 5.500e+01 6.700e+01 3.300e+01 7.000e+01 1.660e+02
 1.420e+02 4.600e+01 3.890e+02 2.460e+02 7.800e+01 8.700e+01 1.920e+02
 1.600e+02 1.850e+02 2.170e+02 3.230e+02 2.600e+01 9.500e+01 1.780e+02
 1.340e+02 2.270e+02 1.790e+02 1.130e+02 1.090e+02 1.450e+02 2.140e+02
 7.700e+01 1.490e+02 7.300e+01 1.020e+02 2.300e+01 2.760e+02 1.460e+02
 1.440e+02 7.100e+01 1.470e+02 1.040e+02 1.580e+02 6.400e+01 1.610e+02
 1.400e+02 2.400e+02 1.550e+02 1.590e+02 5.450e+02 1.630e+02 9.800e+01
 1.690e+02 2.670e+02 8.200e+01 1.640e+02 9.900e+01 1.140e+02 1.360e+02
 2.010e+02 1.270e+02 1.740e+02 7.500e+01 1.540e+02 1.000e+02 9.000e+01
 2.640e+02 6.800e+01 2.050e+02 1.810e+02 1.153e+03 7.040e+02 2.290e+02
 1.280e+02 1.530e+02 1.330e+02 1.910e+02 1.170e+02 1.230e+02 1.670e+02
 6.100e+02 2.240e+02 1.300e+02 1.010e+02 8.100e+01 1.190e+02 9.200e+01
 2.660e+02 2.840e+02 2.770e+02 2.120e+02 2.000e+02 2.100e+02 2.420e+02
 4.030e+02 1.120e+02 1.880e+02 4.490e+02 1.940e+02 3.030e+02 1.060e+02
 1.970e+02 2.250e+02 2.060e+02 2.790e+02 7.380e+02 1.220e+02 1.770e+02
```

```
 8.800e+01 3.500e+02 3.020e+02 1.430e+02 2.580e+02 1.050e+02 1.410e+02
 9.600e+01 6.870e+02 1.320e+02 2.330e+02 1.180e+02 2.510e+02 2.070e+02
 7.120e+02 1.250e+02 2.040e+02 2.370e+02 1.820e+02 1.720e+02 8.900e+01
 1.890e+02 3.120e+02 4.430e+02 9.310e+02 3.170e+02 2.180e+02 1.980e+02
 2.480e+02 1.900e+02 6.060e+02 1.210e+02 2.500e+02 2.280e+02 1.100e+02
 1.390e+02 5.210e+02 3.110e+02 6.650e+02 1.710e+02 2.980e+02 2.650e+02
 2.030e+02 1.870e+02 5.230e+02 3.100e+02 2.150e+02 2.160e+02 3.520e+02
 7.650e+02 1.847e+03 1.480e+02 2.730e+02 2.360e+02 3.770e+02 6.760e+02
 2.340e+02 2.600e+02 2.430e+02 3.040e+02 2.410e+02 3.630e+02 2.200e+02
 1.750e+02 3.400e+02 8.270e+02 2.020e+02 3.090e+02 2.690e+02 1.730e+02
 2.560e+02 2.380e+02 1.293e+03 2.090e+02 1.370e+02 2.850e+02 2.220e+02
 2.390e+02 2.630e+02 2.830e+02 3.840e+02 2.810e+02 3.700e+02 2.210e+02
 7.200e+02 6.540e+02 3.000e+02 2.230e+02 5.020e+02 1.223e+03 2.720e+02
 3.420e+02 5.000e+02 1.760e+02 4.160e+02 2.320e+02 8.590e+02 1.700e+02
 1.960e+02 2.300e+02 7.720e+02 2.920e+02 2.530e+02 3.080e+02 1.019e+03
 6.850e+02 2.890e+02 3.680e+02 2.820e+02 7.630e+02 3.670e+02 1.570e+02
 3.990e+02 4.080e+02 4.010e+02 2.910e+02 3.050e+02 3.780e+02 2.490e+02
 2.080e+02 3.450e+02 4.290e+02 5.310e+02 2.700e+02 8.970e+02 5.970e+02
 9.380e+02 3.350e+02 9.770e+02 3.270e+02 3.640e+02 2.680e+02 6.990e+02
 8.360e+02 2.780e+02 3.950e+02 2.990e+02 3.480e+02 4.840e+02 2.310e+02
 2.470e+02 6.790e+02 2.940e+02 1.066e+03 2.960e+02 3.370e+02 6.260e+02
 4.100e+02 6.180e+02 6.000e+02 3.510e+02 7.410e+02 3.060e+02 2.710e+02
 4.320e+02 2.450e+02 9.560e+02 4.410e+02 2.610e+02 1.410e+03 2.520e+02
 2.800e+02 5.740e+02 1.840e+02 3.240e+02 3.730e+02 4.700e+02 3.980e+02
 6.750e+02 4.650e+02 5.380e+02 3.820e+02 6.190e+02 6.970e+02 4.510e+02
 3.200e+02 3.250e+02 3.130e+02 2.860e+02 9.510e+02 4.360e+02 3.220e+02
 5.880e+02 3.540e+02]
----------------------------------
NASDelay has 429 unique values:
----------------------------------
[0.000e+00 1.300e+01 7.000e+00 3.000e+00 2.400e+01 6.000e+00 5.000e+00
 1.600e+01 1.700e+01 1.900e+01 5.700e+01 1.500e+01 5.900e+01 6.100e+01
 1.440e+02 4.000e+00 9.400e+01 4.400e+01 3.700e+01 8.000e+00 1.100e+01
 1.800e+01 4.000e+01 2.000e+00 2.300e+01 4.700e+01 2.000e+01 2.500e+01
 5.300e+01 5.500e+01 9.000e+00 1.290e+02 4.500e+01 3.200e+01 1.200e+01
 6.700e+01 1.000e+02 2.100e+01 3.900e+01 2.800e+01 9.500e+01 6.600e+01
 8.600e+01 1.000e+00 4.600e+01 2.700e+01 3.500e+01 8.000e+01 1.000e+01
 1.400e+01 3.100e+01 4.300e+01 3.800e+01 3.000e+01 1.190e+02 1.010e+02
 3.600e+01 2.200e+01 2.600e+01 5.400e+01 2.790e+02 6.800e+01 6.500e+01
 3.400e+01 1.020e+02 1.460e+02 1.250e+02 5.600e+01 1.110e+02 2.090e+02
 4.040e+02 8.100e+01 3.300e+01 1.080e+02 5.000e+01 6.400e+01 1.750e+02
 6.000e+01 3.010e+02 8.900e+01 8.500e+01 3.230e+02 6.900e+01 7.500e+01
 1.050e+02 1.400e+02 4.900e+01 1.030e+02 4.100e+01 2.710e+02 4.800e+01
 2.900e+01 7.300e+01 4.200e+01 1.730e+02 1.840e+02 5.200e+01 5.100e+01
 2.250e+02 8.800e+01 5.800e+01 1.630e+02 6.200e+01 8.400e+01 1.120e+02
 1.170e+02 9.100e+01 8.300e+01 7.700e+01 9.300e+01 7.800e+01 1.530e+02
 6.300e+01 7.400e+01 2.400e+02 8.700e+01 1.150e+02 1.330e+02 2.990e+02
 1.920e+02 1.600e+02 9.800e+01 7.100e+01 1.350e+02 1.480e+02 3.310e+02
 1.090e+02 9.600e+01 1.430e+02 7.600e+01 1.760e+02 1.130e+02 1.070e+02
```

```
 1.040e+02 2.160e+02 1.270e+02 1.340e+02 1.470e+02 1.230e+02 1.810e+02
 1.510e+02 1.850e+02 1.420e+02 1.870e+02 1.310e+02 9.700e+01 1.880e+02
 1.370e+02 1.180e+02 1.100e+02 4.650e+02 7.000e+01 1.240e+02 1.060e+02
 7.200e+01 1.410e+02 8.200e+01 1.860e+02 1.960e+02 2.500e+02 1.540e+02
 1.140e+02 1.720e+02 6.710e+02 1.160e+02 7.900e+01 1.450e+02 1.300e+02
 2.940e+02 2.190e+02 1.260e+02 1.500e+02 1.710e+02 9.900e+01 1.680e+02
 2.100e+02 2.040e+02 2.280e+02 1.320e+02 2.420e+02 1.520e+02 2.960e+02
 2.080e+02 1.200e+02 9.000e+01 1.620e+02 1.570e+02 2.430e+02 2.930e+02
 2.690e+02 2.640e+02 1.380e+02 2.030e+02 1.280e+02 1.490e+02 9.200e+01
 2.000e+02 2.300e+02 1.210e+02 2.560e+02 1.990e+02 1.640e+02 1.940e+02
 1.660e+02 1.590e+02 2.750e+02 1.740e+02 2.720e+02 2.760e+02 1.950e+02
 1.610e+02 2.570e+02 3.330e+02 3.510e+02 1.220e+02 1.670e+02 3.400e+02
 3.070e+02 1.790e+02 2.310e+02 3.120e+02 2.140e+02 3.890e+02 1.560e+02
 3.490e+02 2.260e+02 2.230e+02 2.340e+02 2.780e+02 2.070e+02 1.930e+02
 2.980e+02 2.970e+02 1.580e+02 2.050e+02 1.390e+02 1.360e+02 1.890e+02
 3.130e+02 2.830e+02 1.900e+02 2.020e+02 2.110e+02 2.450e+02 3.220e+02
 5.360e+02 2.240e+02 2.870e+02 2.460e+02 2.530e+02 1.690e+02 2.180e+02
 2.130e+02 2.270e+02 1.970e+02 4.070e+02 1.770e+02 4.660e+02 1.980e+02
 1.700e+02 4.780e+02 1.650e+02 2.480e+02 6.910e+02 1.830e+02 2.680e+02
 2.370e+02 2.010e+02 2.670e+02 3.100e+02 3.370e+02 2.060e+02 4.200e+02
 1.800e+02 1.820e+02 3.640e+02 1.550e+02 2.650e+02 2.350e+02 2.380e+02
 1.910e+02 2.360e+02 2.540e+02 3.770e+02 2.210e+02 1.194e+03 3.470e+02
 2.890e+02 2.330e+02 2.510e+02 3.540e+02 3.030e+02 2.220e+02 3.280e+02
 2.490e+02 2.800e+02 2.120e+02 2.320e+02 3.300e+02 3.760e+02 2.410e+02
 5.330e+02 2.950e+02 3.730e+02 2.390e+02 2.590e+02 2.200e+02 3.080e+02
 4.540e+02 2.630e+02 6.750e+02 3.700e+02 3.450e+02 2.290e+02 9.440e+02
 3.170e+02 3.880e+02 5.780e+02 2.600e+02 1.780e+02 3.190e+02 3.480e+02
 3.260e+02 2.740e+02 2.150e+02 3.390e+02 3.320e+02 2.810e+02 3.140e+02
 7.010e+02 1.053e+03 2.170e+02 3.830e+02 2.880e+02 2.770e+02 3.180e+02
 4.620e+02 2.730e+02 6.030e+02 3.290e+02 3.090e+02 4.140e+02 8.950e+02
 3.460e+02 6.760e+02 4.430e+02 2.700e+02 2.580e+02 2.620e+02 2.660e+02
 3.250e+02 4.470e+02 3.040e+02 2.470e+02 7.270e+02 2.860e+02 4.710e+02
 4.180e+02 7.100e+02 3.560e+02 2.820e+02 2.610e+02 5.510e+02 2.910e+02
 2.520e+02 3.740e+02 7.880e+02 6.180e+02 8.060e+02 3.870e+02 3.900e+02
 3.240e+02 2.550e+02 3.360e+02 3.630e+02 4.340e+02 4.030e+02 3.590e+02
 3.340e+02 8.520e+02 9.260e+02 4.010e+02 3.690e+02 3.410e+02 5.630e+02
 4.810e+02 5.130e+02 3.600e+02 4.130e+02 4.120e+02 2.840e+02 9.520e+02
 3.020e+02 9.840e+02 3.050e+02 3.000e+02 3.200e+02 3.750e+02 3.580e+02
 3.810e+02 4.150e+02 1.343e+03 5.180e+02 3.350e+02 5.900e+02 4.050e+02
 3.060e+02 3.270e+02 1.008e+03 3.520e+02 3.710e+02 2.850e+02 6.820e+02
 4.160e+02 8.580e+02 6.790e+02 4.410e+02 3.150e+02 4.110e+02 4.530e+02
 4.690e+02 3.210e+02 4.000e+02 3.570e+02 4.320e+02 3.420e+02 4.020e+02
 5.320e+02 3.430e+02]
----------------------------------
SecurityDelay has 99 unique values:
----------------------------------
[  0.   6.  82.   8.  57.  11.  25.  26.  10.  20.   3.  23.  16. 148.
   1.  30.  12.   4.   5.  75.   7.  24.   2.   9.  17.  60.  44.  21.
  18. 208.  28.  29.  19.  15.  62.  42.  37.  22. 168.  93.  14.  36.
```

```
  13.  32.  39.  54.  56.  86. 199.  40.  38.  48.  41. 159.  27. 106.
 115.  35.  46.  53.  43.  88.  83. 219.  31. 119.  47.  92.  94.  51.
  80.  85.  52.  70.  49. 124.  45.  90.  33.  77.  66. 214.  59. 113.
 131. 180. 102. 117.  34.  58.  96.  68.  67.  98. 123.  73.  84.  72.
  71.]
---------------------------------
LateAircraftDelay has 490 unique values:
---------------------------------
[0.000e+00 3.200e+01 2.140e+02 1.600e+01 1.000e+00 2.100e+01 1.170e+02
 8.000e+00 1.790e+02 8.400e+01 2.700e+01 1.800e+01 2.200e+01 3.800e+01
 1.100e+01 3.300e+01 3.000e+01 1.000e+01 1.310e+02 7.200e+01 1.900e+01
 4.200e+01 1.750e+02 4.300e+01 3.400e+01 1.200e+01 2.000e+01 1.760e+02
 1.930e+02 2.900e+01 6.000e+00 2.800e+01 3.000e+00 8.000e+01 8.300e+01
 1.330e+02 1.700e+01 1.500e+01 7.000e+00 5.000e+00 5.200e+01 5.800e+01
 5.500e+01 9.000e+00 2.300e+01 4.400e+01 7.900e+01 8.900e+01 6.200e+01
 2.370e+02 6.800e+01 4.100e+01 6.000e+01 5.700e+01 4.000e+01 3.700e+01
 1.350e+02 2.800e+02 8.100e+01 4.800e+01 5.100e+01 9.000e+01 1.190e+02
 5.600e+01 9.500e+01 6.100e+01 1.070e+02 4.700e+01 7.300e+01 2.600e+01
 9.800e+01 1.300e+02 3.600e+01 4.500e+01 3.360e+02 3.100e+01 2.000e+00
 7.400e+01 4.900e+01 1.680e+02 1.050e+02 5.400e+01 9.400e+01 1.300e+01
 2.410e+02 1.080e+02 2.500e+01 2.400e+01 9.300e+01 8.200e+01 8.800e+01
 1.400e+01 1.220e+02 3.900e+01 9.100e+01 6.700e+01 1.040e+02 2.170e+02
 7.800e+01 7.000e+01 1.320e+02 3.500e+01 1.650e+02 2.160e+02 6.300e+01
 6.500e+01 4.000e+00 5.900e+01 1.400e+02 9.900e+01 2.070e+02 1.630e+02
 1.360e+02 7.500e+01 1.060e+02 1.620e+02 1.100e+02 9.200e+01 1.580e+02
 1.780e+02 4.340e+02 1.700e+02 6.400e+01 1.180e+02 8.700e+01 4.600e+01
 1.410e+02 1.240e+02 1.290e+02 1.150e+02 5.000e+01 1.000e+02 2.320e+02
 8.600e+01 2.360e+02 2.060e+02 5.300e+01 6.900e+01 9.600e+01 1.160e+02
 1.230e+02 2.040e+02 3.720e+02 1.120e+02 1.590e+02 1.890e+02 1.370e+02
 2.690e+02 9.700e+01 1.770e+02 1.510e+02 2.330e+02 1.530e+02 1.010e+02
 1.820e+02 2.500e+02 7.600e+01 7.100e+01 1.740e+02 8.500e+01 1.480e+02
 1.280e+02 1.380e+02 1.270e+02 1.030e+02 1.020e+02 1.610e+02 2.340e+02
 2.080e+02 1.260e+02 1.200e+02 1.130e+02 1.250e+02 1.860e+02 1.720e+02
 1.110e+02 1.950e+02 1.450e+02 1.490e+02 1.430e+02 1.550e+02 2.050e+02
 1.210e+02 1.470e+02 2.270e+02 7.700e+01 2.510e+02 2.750e+02 1.690e+02
 1.090e+02 1.440e+02 2.250e+02 1.660e+02 1.810e+02 1.870e+02 1.500e+02
 1.540e+02 2.200e+02 3.180e+02 1.830e+02 1.570e+02 1.970e+02 2.980e+02
 2.940e+02 1.420e+02 2.560e+02 2.110e+02 6.600e+01 2.710e+02 1.140e+02
 1.390e+02 2.290e+02 2.010e+02 2.190e+02 1.520e+02 2.420e+02 2.280e+02
 1.460e+02 2.440e+02 2.460e+02 2.550e+02 1.600e+02 2.610e+02 1.670e+02
 2.490e+02 1.560e+02 2.990e+02 4.070e+02 1.850e+02 1.840e+02 1.980e+02
 3.660e+02 1.640e+02 3.840e+02 1.340e+02 1.990e+02 2.210e+02 2.230e+02
 3.570e+02 2.930e+02 3.350e+02 4.270e+02 1.730e+02 2.150e+02 2.470e+02
 2.390e+02 2.540e+02 2.530e+02 2.920e+02 2.030e+02 1.880e+02 1.800e+02
 3.970e+02 1.710e+02 3.990e+02 1.900e+02 1.940e+02 2.630e+02 3.040e+02
 2.790e+02 2.660e+02 2.020e+02 3.010e+02 2.620e+02 2.700e+02 1.960e+02
 1.920e+02 2.300e+02 3.940e+02 3.080e+02 2.770e+02 3.170e+02 3.150e+02
 2.260e+02 2.680e+02 2.220e+02 2.650e+02 1.910e+02 2.590e+02 2.000e+02
 4.180e+02 3.470e+02 3.300e+02 2.120e+02 3.620e+02 2.480e+02 2.520e+02
```

```
 2.830e+02 5.280e+02 3.670e+02 2.450e+02 2.640e+02 3.680e+02 3.140e+02
 4.220e+02 2.090e+02 2.970e+02 2.180e+02 2.240e+02 2.380e+02 3.240e+02
 2.670e+02 5.230e+02 2.570e+02 2.820e+02 3.370e+02 3.770e+02 3.870e+02
 2.950e+02 3.600e+02 4.360e+02 2.740e+02 4.540e+02 3.280e+02 2.960e+02
 3.110e+02 9.120e+02 2.100e+02 2.900e+02 2.310e+02 4.000e+02 7.830e+02
 6.730e+02 3.230e+02 3.130e+02 2.400e+02 3.100e+02 3.290e+02 3.390e+02
 2.870e+02 3.550e+02 4.130e+02 3.960e+02 2.850e+02 2.910e+02 2.780e+02
 4.280e+02 4.430e+02 2.350e+02 2.600e+02 5.700e+02 5.390e+02 2.130e+02
 8.010e+02 3.000e+02 6.100e+02 2.840e+02 3.410e+02 2.760e+02 3.190e+02
 3.950e+02 4.290e+02 3.090e+02 2.580e+02 4.460e+02 3.530e+02 2.720e+02
 6.120e+02 6.480e+02 3.310e+02 3.590e+02 3.500e+02 3.120e+02 4.480e+02
 3.380e+02 3.510e+02 4.440e+02 5.370e+02 3.250e+02 3.580e+02 3.650e+02
 4.740e+02 1.407e+03 4.850e+02 5.150e+02 4.230e+02 3.480e+02 5.690e+02
 5.190e+02 3.070e+02 3.910e+02 4.500e+02 5.110e+02 4.890e+02 4.410e+02
 6.220e+02 3.760e+02 3.030e+02 2.730e+02 2.880e+02 5.560e+02 2.860e+02
 3.050e+02 8.240e+02 3.020e+02 4.300e+02 3.640e+02 2.890e+02 3.690e+02
 4.030e+02 4.020e+02 3.160e+02 4.420e+02 3.810e+02 4.620e+02 3.270e+02
 7.950e+02 5.120e+02 3.980e+02 3.400e+02 3.490e+02 2.810e+02 4.570e+02
 3.820e+02 3.520e+02 3.860e+02 2.430e+02 4.660e+02 3.260e+02 1.013e+03
 8.620e+02 3.060e+02 5.580e+02 3.450e+02 7.270e+02 4.530e+02 7.320e+02
 4.940e+02 3.330e+02 4.520e+02 4.990e+02 3.830e+02 3.320e+02 7.440e+02
 1.256e+03 3.430e+02 4.870e+02 5.030e+02 4.970e+02 4.790e+02 5.790e+02
 4.950e+02 4.250e+02 3.610e+02 3.200e+02 3.220e+02 3.630e+02 5.130e+02
 6.440e+02 4.320e+02 3.560e+02 1.054e+03 5.510e+02 6.560e+02 4.260e+02
 4.240e+02 5.640e+02 3.790e+02 1.173e+03 5.570e+02 4.350e+02 8.250e+02
 3.710e+02 3.210e+02 3.930e+02 7.090e+02 4.630e+02 5.600e+02 7.300e+02
 3.420e+02 4.080e+02 4.040e+02 3.440e+02 8.420e+02 4.750e+02 5.180e+02
 6.650e+02 8.190e+02 3.900e+02 8.690e+02 4.050e+02 3.740e+02 3.540e+02
 3.700e+02 4.490e+02 4.700e+02 3.460e+02 3.920e+02 4.580e+02
3.780e+02]
```

```
cleaned_airline_df[cleaned_airline_df['Cancelled'] == 1]
['CancellationCode'].unique()
```

```
array(['A', 'Not Defined', 'B', 'C', 'D'], dtype=object)
```

Create New Data Frame

create a separate data frame for canceled flights

```
canceled_airline_df =
cleaned_airline_df[cleaned_airline_df['Cancelled']==1]
canceled_airline_df.shape
```

```
(36462, 87)
```

```
diverted_airline_df =
cleaned_airline_df[cleaned_airline_df['Diverted']==1]
diverted_airline_df.shape
```

```
(4590, 87)

delay_airline_df =
cleaned_airline_df[(cleaned_airline_df['ArrDelayMinutes']>0)
                                                          |
(cleaned_airline_df['DepDelayMinutes']>0)]
delay_airline_df.shape

(1060475, 87)
```

What is the structure of your dataset?

There are 2,000,000 airline trip in the dataset with 109 features last 24 column contain no data. Most variables are float, int and objects.

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

I'm most interested in figuring out what features are best for predicting the price of the diamonds in the dataset.

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

I expect that carat will have the strongest effect on each diamond's price: the larger the diamond, the higher the price. I also think that the other big "C"s of diamonds: cut, color, and clarity, will have effects on the price, though to a much smaller degree than the main effect of carat.

Univariate Exploration

To investigate the patterns of flight cancellations, we employed both pie charts and bar charts for visual analysis. Here's a summary of the approach and findings:

1. pie chart were used to represent the proportion of canceled flights and diverted relative to the total number of flights. These charts visually demonstrates the percentage of each.
2. bar charts are used to show the distribution of cancellations by day of the week and by quarter of the year. This helped in identifying any trends or patterns in cancellations across different time periods

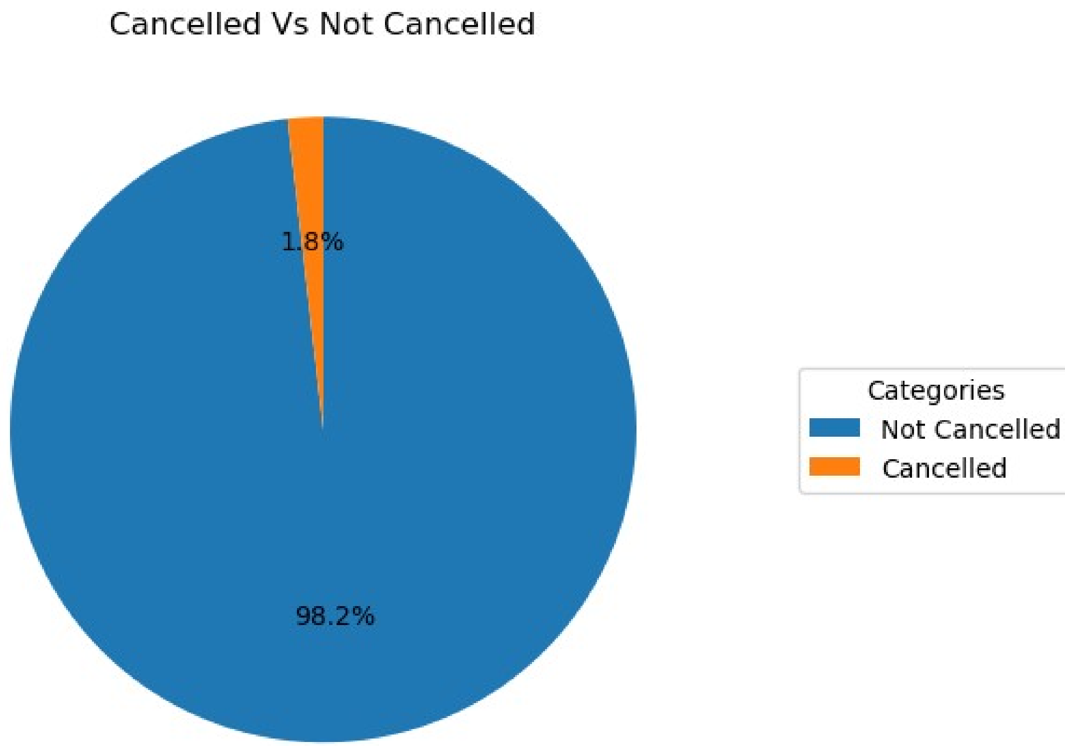
Finding:

1.8% of total trips were canceled. Analyzing cancellations by day of the week reveals that Fridays have fewer cancellations compared to other days, with the highest number of cancellations occurring on Tuesdays. When examining cancellations by quarter, it is evident that the number of cancellations is significantly higher in Q1 compared to other quarters

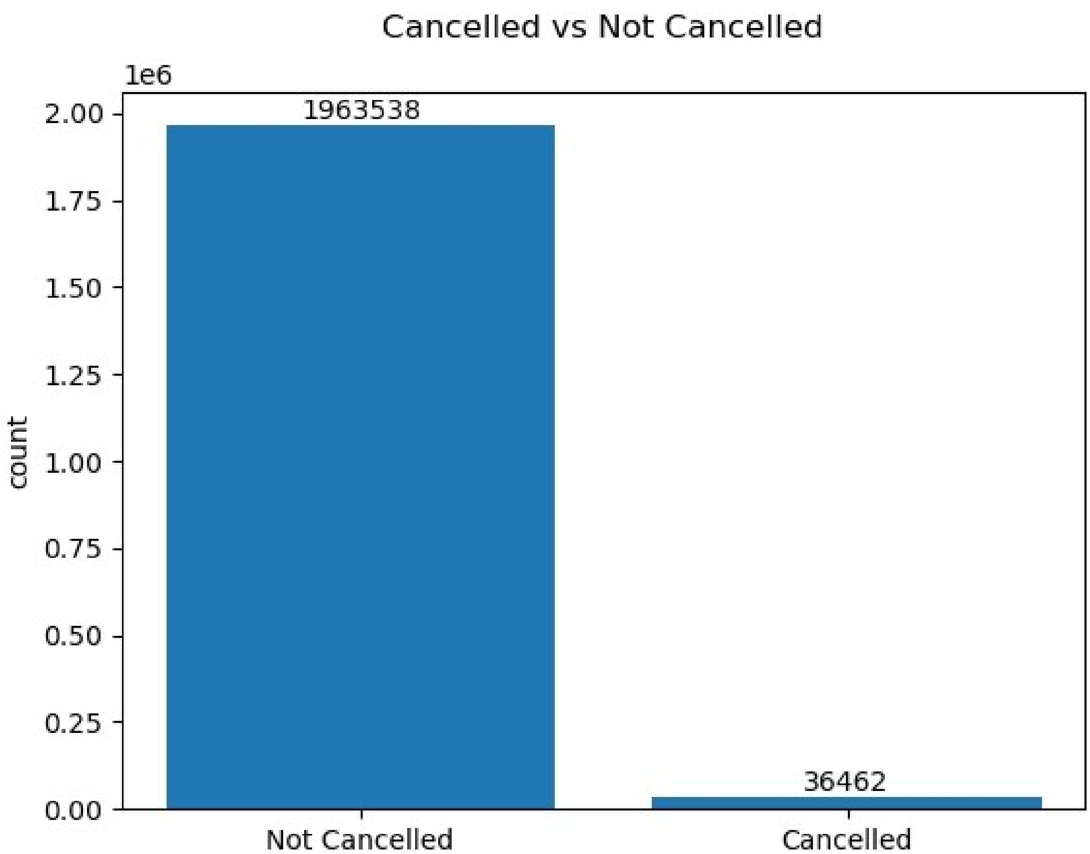
Cancelled Trips

Cancelled Vs Not Cancelled

```
pie_chart(cleaned_airline_df, 'Cancelled','Cancelled Vs Not
Cancelled', ['Not Cancelled', 'Cancelled'])
```

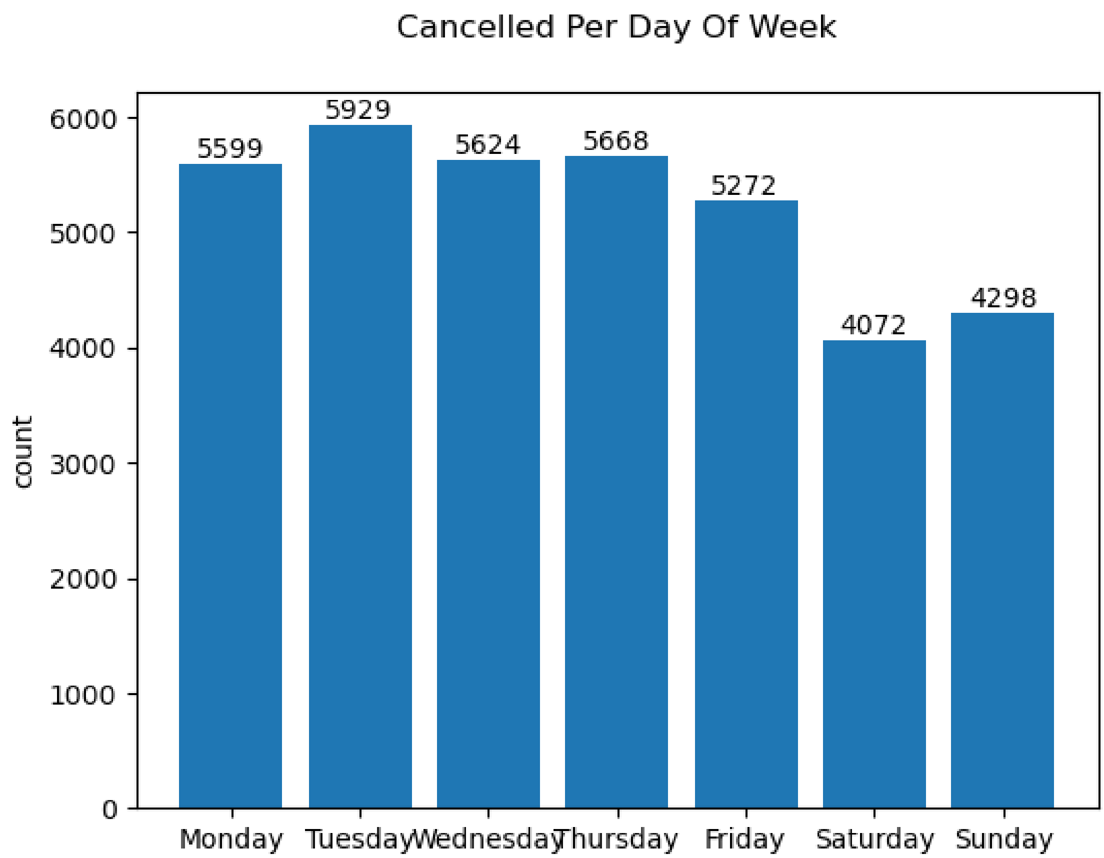



```
bar_chart(cleaned_airline_df, 'Cancelled', 'Cancelled vs Not
Cancelled', ['Not Cancelled', 'Cancelled'] )
```



Cancelled Per week Day

```
bar_chart(canceled_airline_df.sort_values(['DayOfWeek']),
        'DayOfWeek_Desc',
        'Cancelled Per Day Of Week',)
```

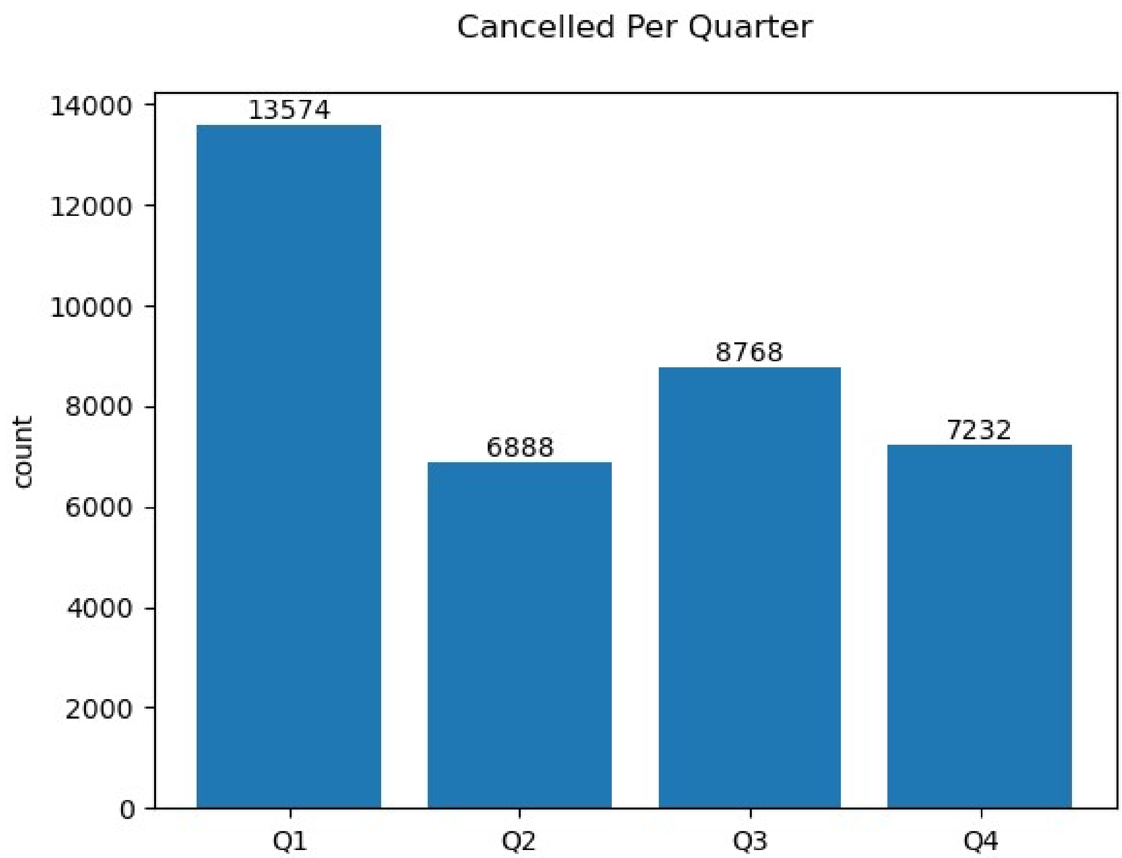


Cancelled Per Quarter

```
canceled_airline_df['Quarter_Desc'].unique()
```

```
array(['Q3', 'Q2', 'Q1', 'Q4'], dtype=object)
```

```
bar_chart(canceled_airline_df.sort_values(['Quarter']),
        'Quarter_Desc',
        'Cancelled Per Quarter',)
```

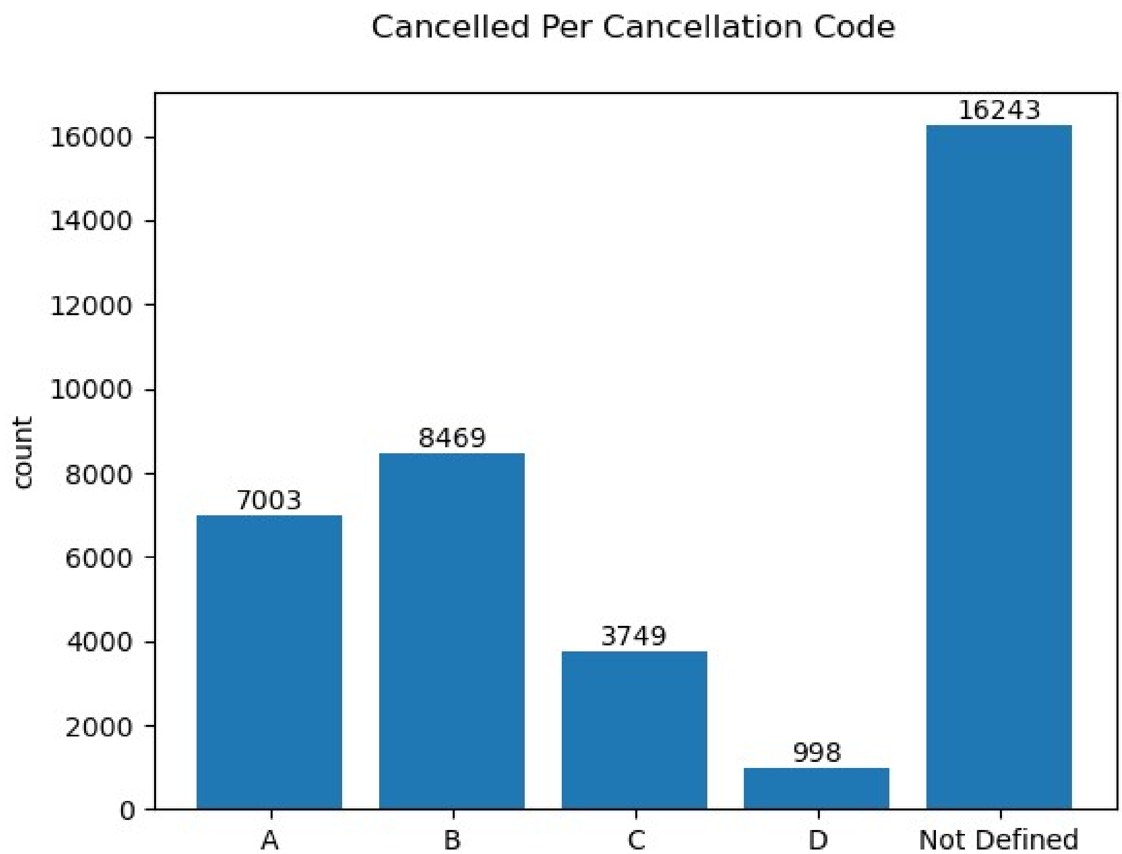


Cancelled trip based On Cancellation Code

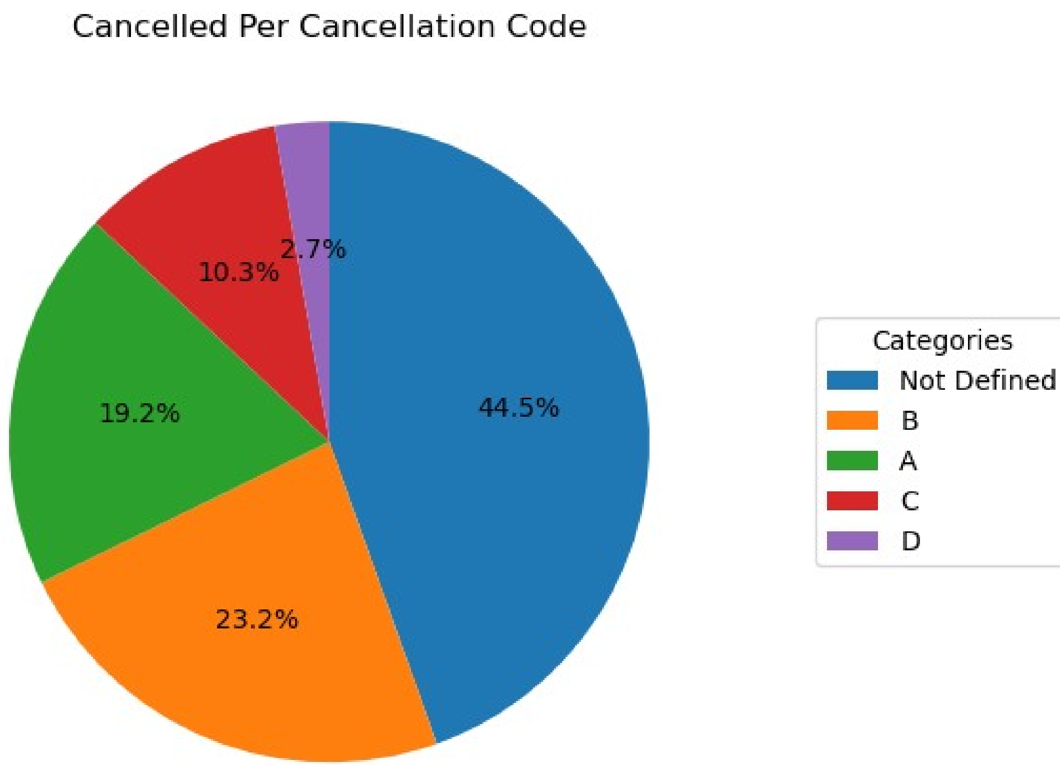
```
canceled_airline_df['CancellationCode'].unique()
```

```
array(['A', 'Not Defined', 'B', 'C', 'D'], dtype=object)
```

```
bar_chart(canceled_airline_df.sort_values(['CancellationCode']),
        'CancellationCode',
        'Cancelled Per Cancellation Code')
```




```
pie_chart(canceled_airline_df.sort_values(['CancellationCode']),
        'CancellationCode',
        'Cancelled Per Cancellation Code')
```



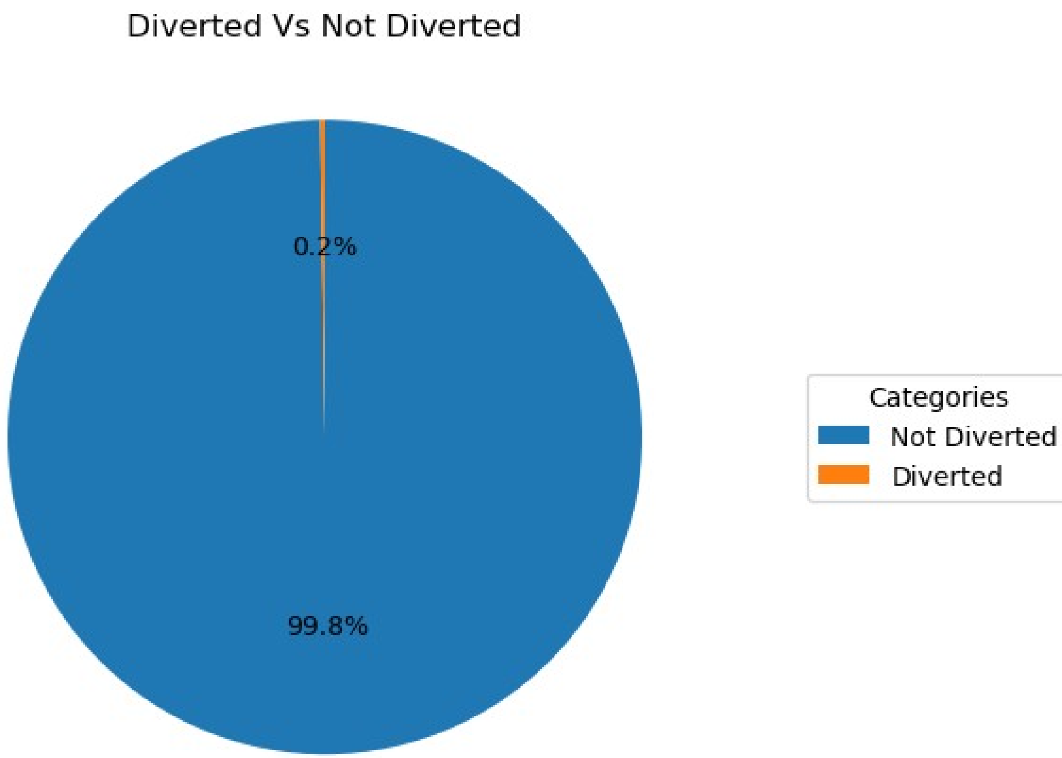
Finding : 1.8% of total trips were canceled. Analyzing cancellations by day of the week reveals that Fridays have fewer cancellations compared to other days, with the highest number of cancellations occurring on Tuesdays. When examining cancellations by quarter, it is evident that the number of cancellations is significantly higher in Q1 compared to other quarters

Diverted Trips

Diverted vs Not Diverted

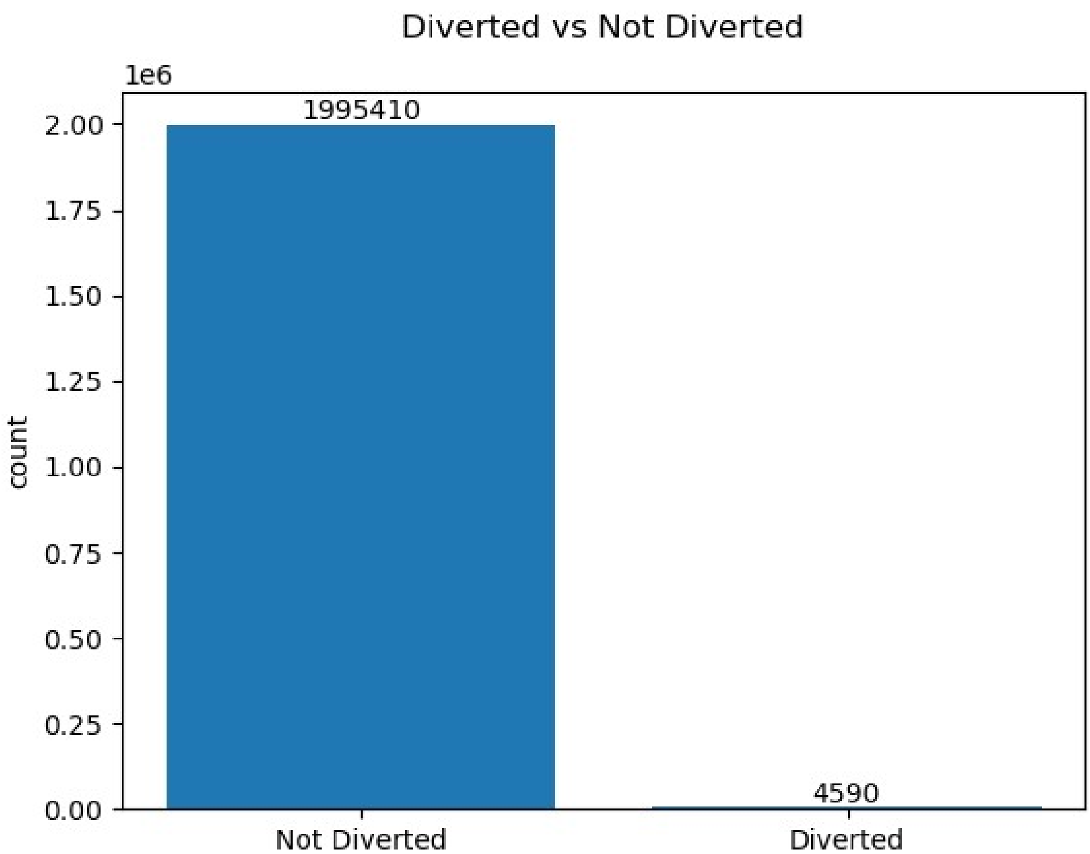
Only 0.2% of the total observation is diverted

```
pie_chart(cleaned_airline_df, 'Diverted','Diverted Vs Not Diverted',
['Not Diverted', 'Diverted'])
```



From the bar chart below the number of observation per Diverted or not was shown

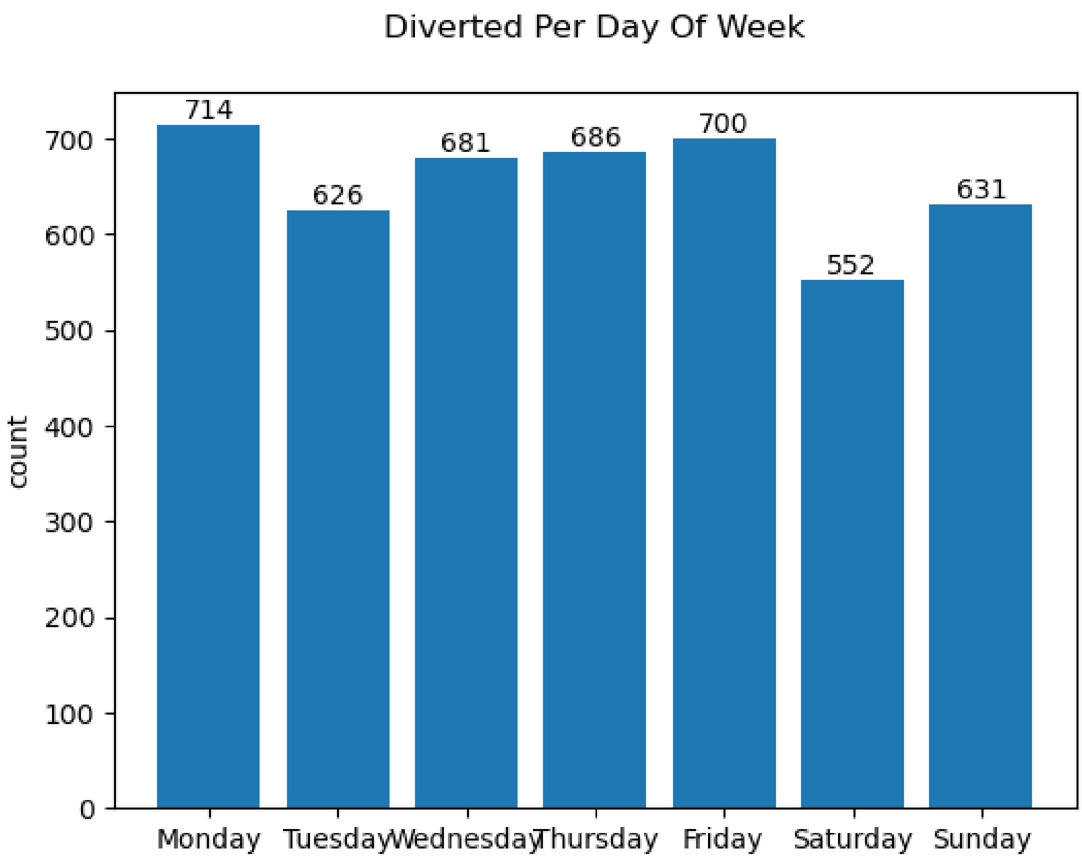
```
bar_chart(cleaned_airline_df, 'Diverted', 'Diverted vs Not Diverted',
['Not Diverted', 'Diverted'] )
```



Diverted Per week Day

In this section, we examine the pattern of diverted flights across different days of the week. However, no significant pattern emerges from the data.

```
bar_chart(diverted_airline_df.sort_values(['DayOfWeek']),
        'DayOfWeek_Desc',
        'Diverted Per Day Of Week',)
```



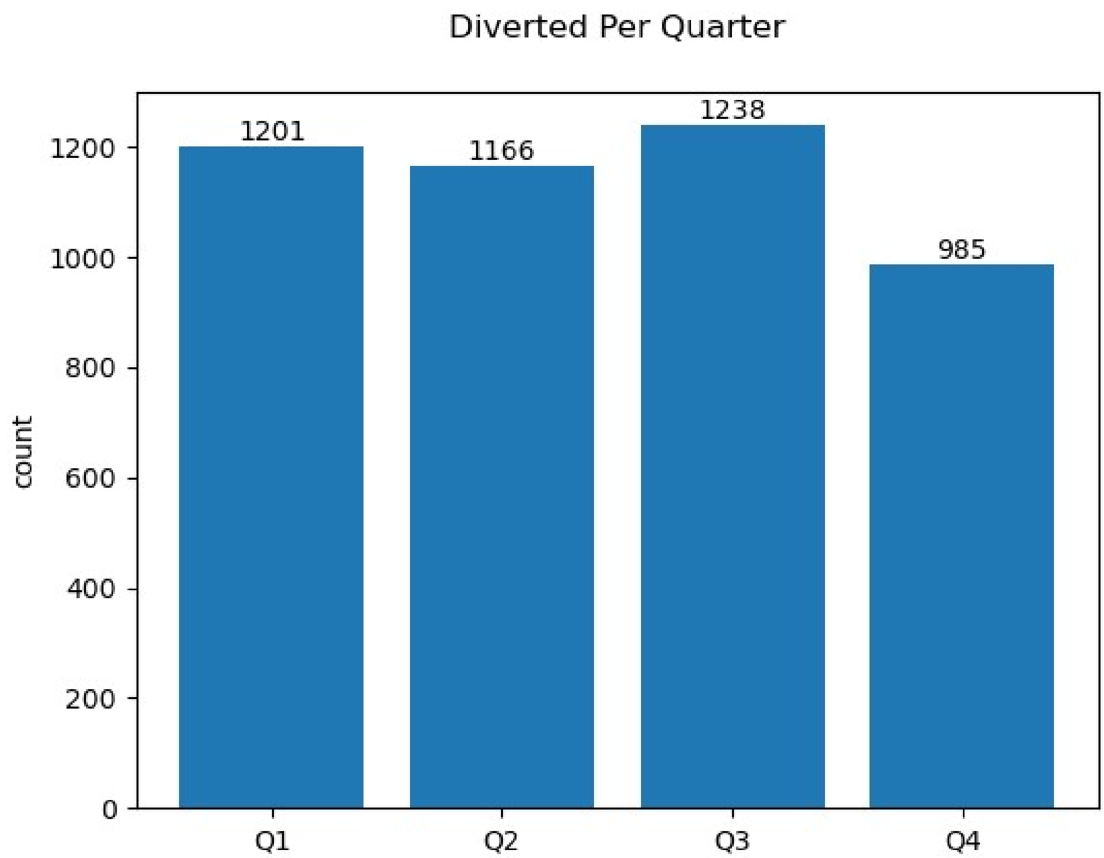
Diverted Per Quarter

In this section, we examine the pattern of diverted flights across different Quarters of the year. However, no significant pattern emerges from the data.

```
diverted_airline_df['Quarter_Desc'].unique()
```

```
array(['Q4', 'Q3', 'Q1', 'Q2'], dtype=object)
```

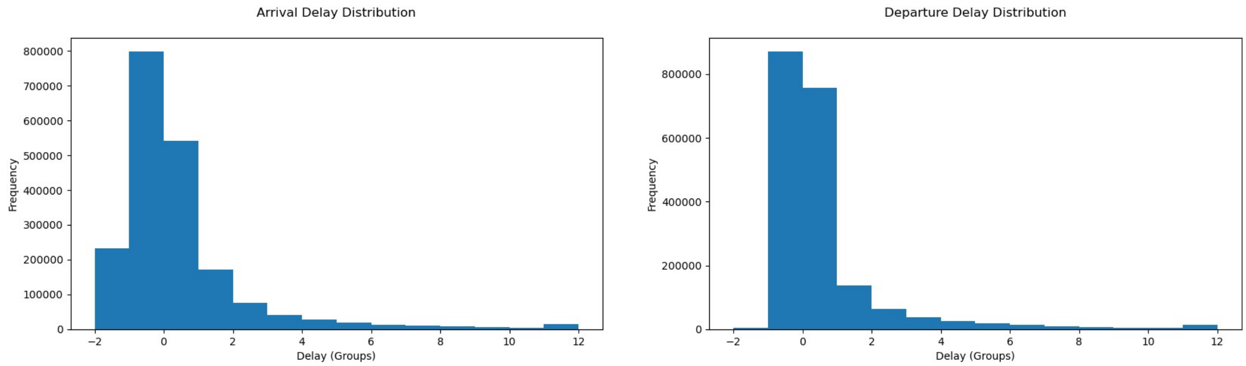
```
bar_chart(diverted_airline_df.sort_values(['Quarter']),
        'Quarter_Desc',
        'Diverted Per Quarter',)
```



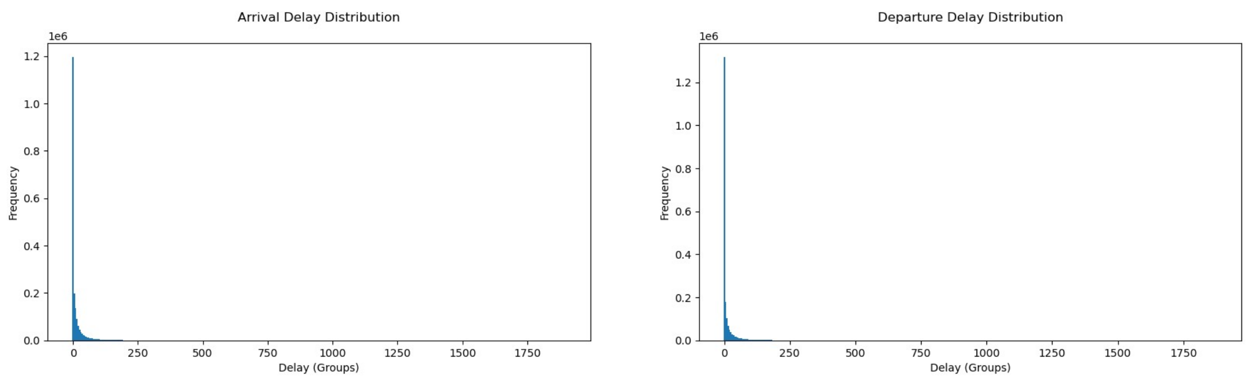
Distribution Of Arrival Delay Groups and DepartureDelayGroups

The histogram below illustrates the distribution of arrival and departure delays. It shows that most data points are concentrated at the lower end of the range, with only a small number extending into the higher end. This distribution is highly skewed to the right, indicating that while most flights experience minimal or no delays, there are a few significant delays that create a long tail on the right side of the chart

```
two_hist_chart(cleaned_airline_df,
               ['ArrivalDelayGroups', 'DepartureDelayGroups'],
               ['Arrival Delay Distribution', 'Departure Delay
Distribution'],
                ['Delay (Groups)', 'Delay (Groups)'])
```




```
two_hist_chart(cleaned_airline_df,
               ['ArrDelayMinutes', 'DepDelayMinutes'],
               ['Arrival Delay Distribution', 'Departure Delay
Distribution'],
                ['Delay (Groups)', 'Delay (Groups)'], 5)
```



Based on the above there exists outliers so we have to remove them

```
cleaned_airline_df['ArrDelayMinutes'].describe()

count    1.958922e+06
mean     1.179442e+01
std      3.197121e+01
min      0.000000e+00
25%      0.000000e+00
50%      0.000000e+00
75%      1.000000e+01
max      1.898000e+03
Name: ArrDelayMinutes, dtype: float64
```

From the above:

1. The average arrival delay in minutes across all flights is ~ 11 min
2. The minimum delay in minutes is ~ 0 min
3. 25% of the flights had no arrival delay - 25th Percentile (25%)
4. 50% of the flights had no arrival delay - 50th Percentile (50%)

5. 75% of the flights had no arrival delay - 70th Percentile (70%)
6. The maximum delay in minutes is ~ 1898 minutes (about 31.6 hours)

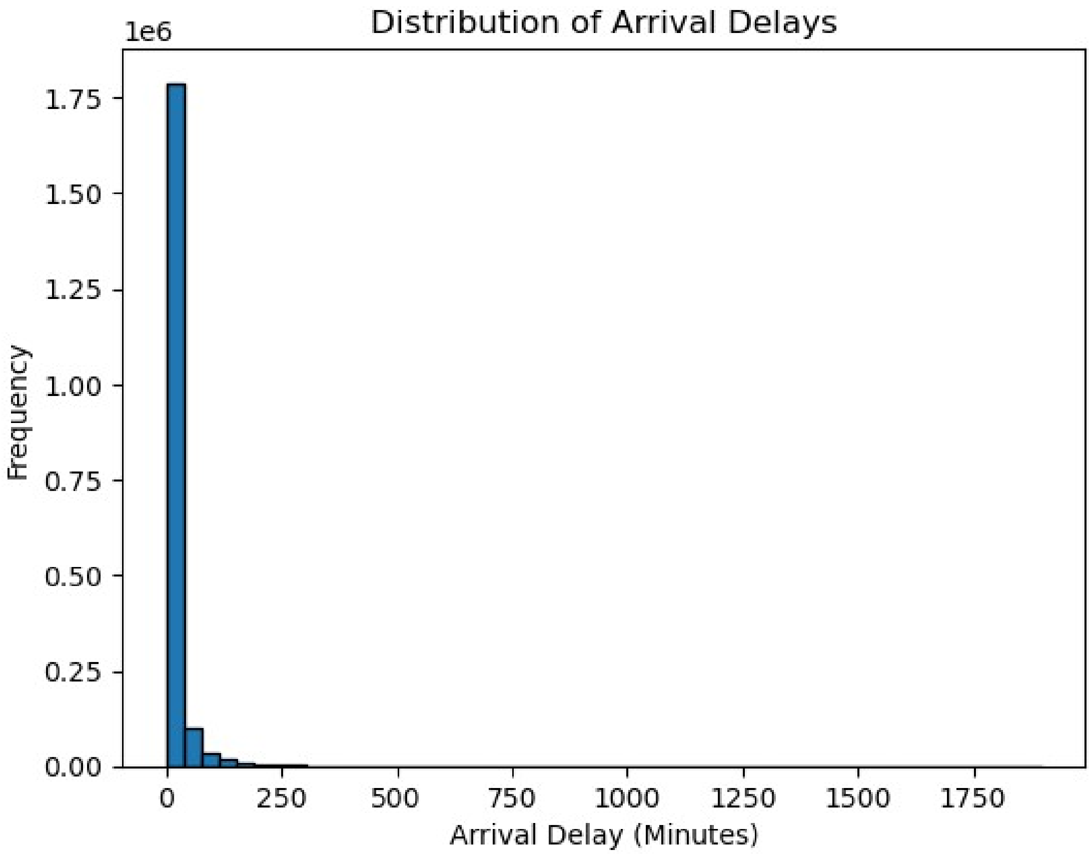
```
cleaned_airline_df['DepDelayMinutes'].describe()
```

```
count    1.963932e+06
mean     1.049667e+01
std      3.196467e+01
min      0.000000e+00
25%      0.000000e+00
50%      0.000000e+00
75%      7.000000e+00
max      1.878000e+03
Name: DepDelayMinutes, dtype: float64
```

From the above:

1. The average departure delay in minutes across all flights is ~ 10 min
2. The minimum delay in minutes is ~ 0 min
3. 25% of the flights had no departure delay - 25th Percentile (25%)
4. 50% of the flights had no departure delay - 50th Percentile (50%)
5. 75% of the flights had 7 minutes departure delay - 70th Percentile (70%)
6. The maximum delay in minutes is ~ 1878 minutes (about 31.3 hours)

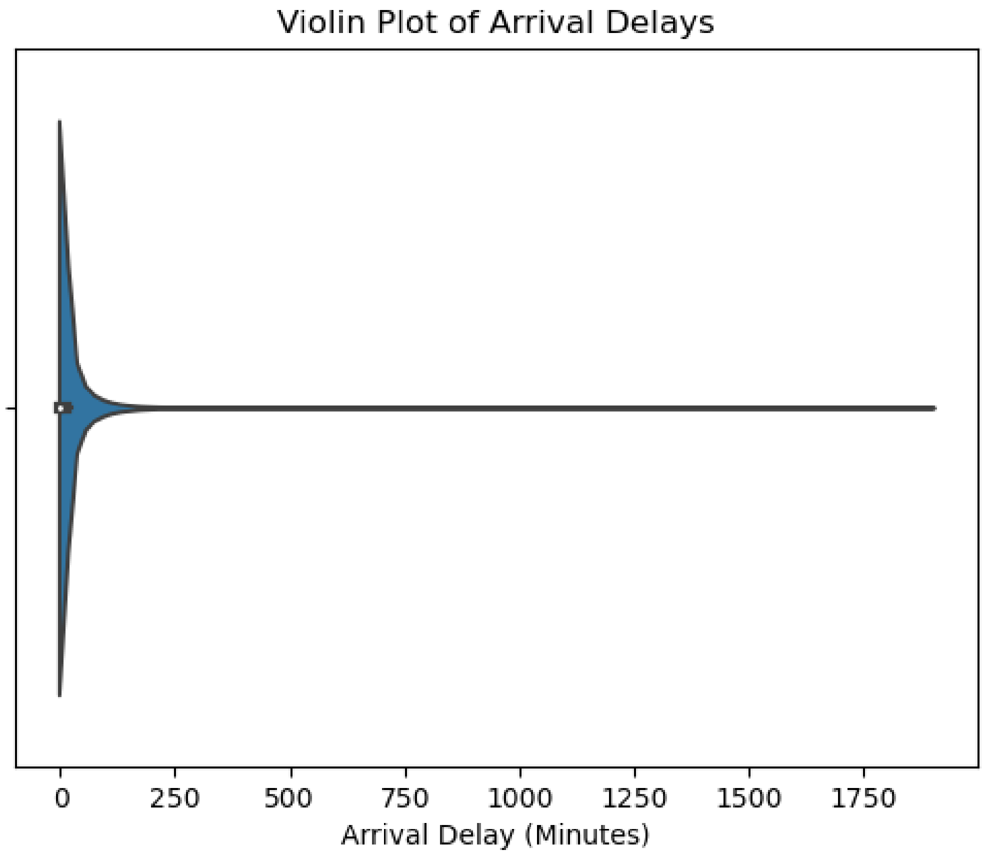
```
plt.hist(cleaned_airline_df['ArrDelayMinutes'].dropna(), bins=50,
edgecolor='k')
plt.xlabel('Arrival Delay (Minutes)')
plt.ylabel('Frequency')
plt.title('Distribution of Arrival Delays')
plt.show()
```



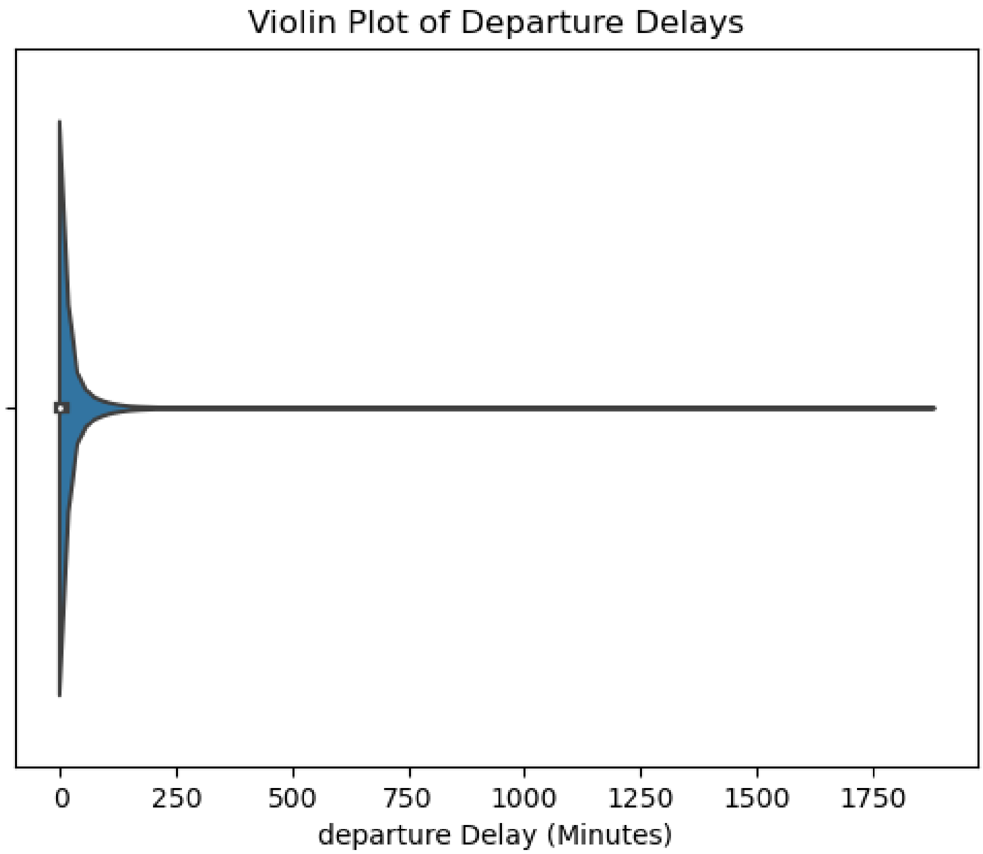
Violin plots for arrival and departure delay

The violin plots for arrival and departure delays clearly demonstrate a highly skewed distribution, with most data concentrated around lower values. Delays up to 30 minutes fall predominantly within the first two delay groups, with a few extreme outliers. The overall delay data is categorized into 12 groups, with approximately 70% of the occurrences concentrated in the first three groups, as depicted in the pie chart below

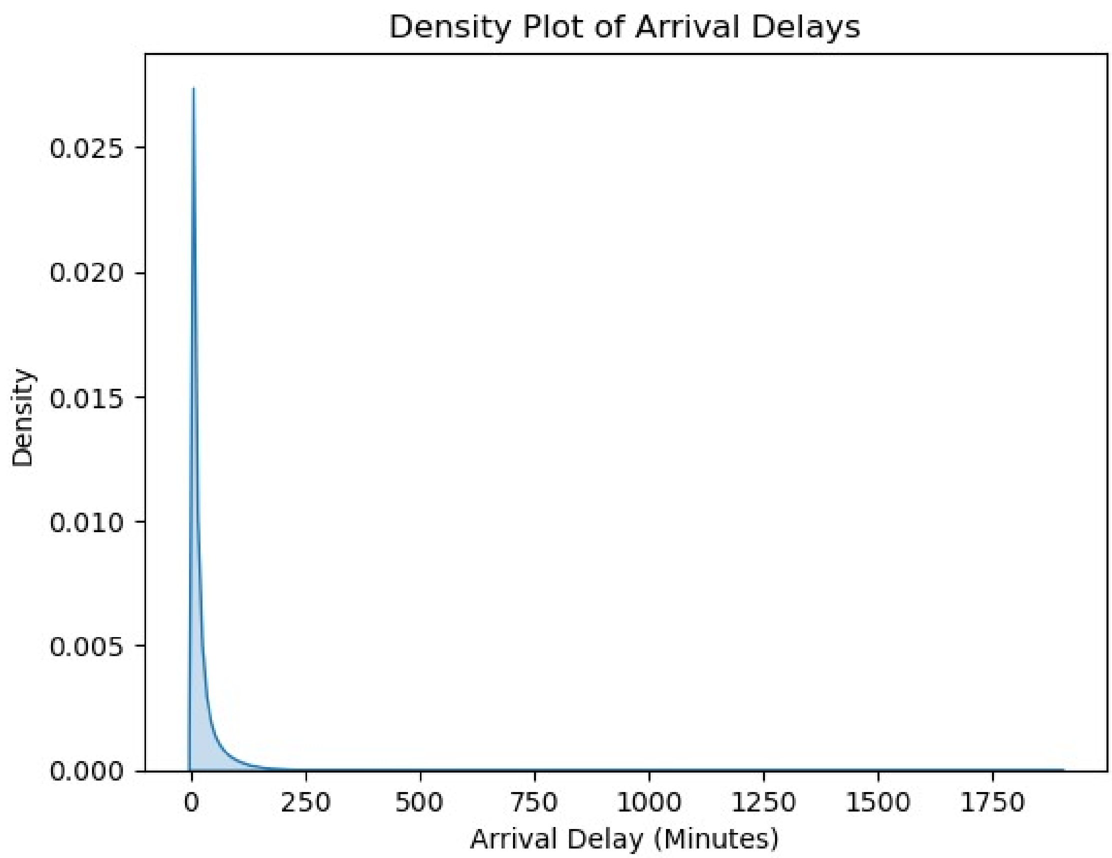
```
sns.violinplot(x=df['ArrDelayMinutes'].dropna())
plt.xlabel('Arrival Delay (Minutes)')
plt.title('Violin Plot of Arrival Delays')
plt.show()
```



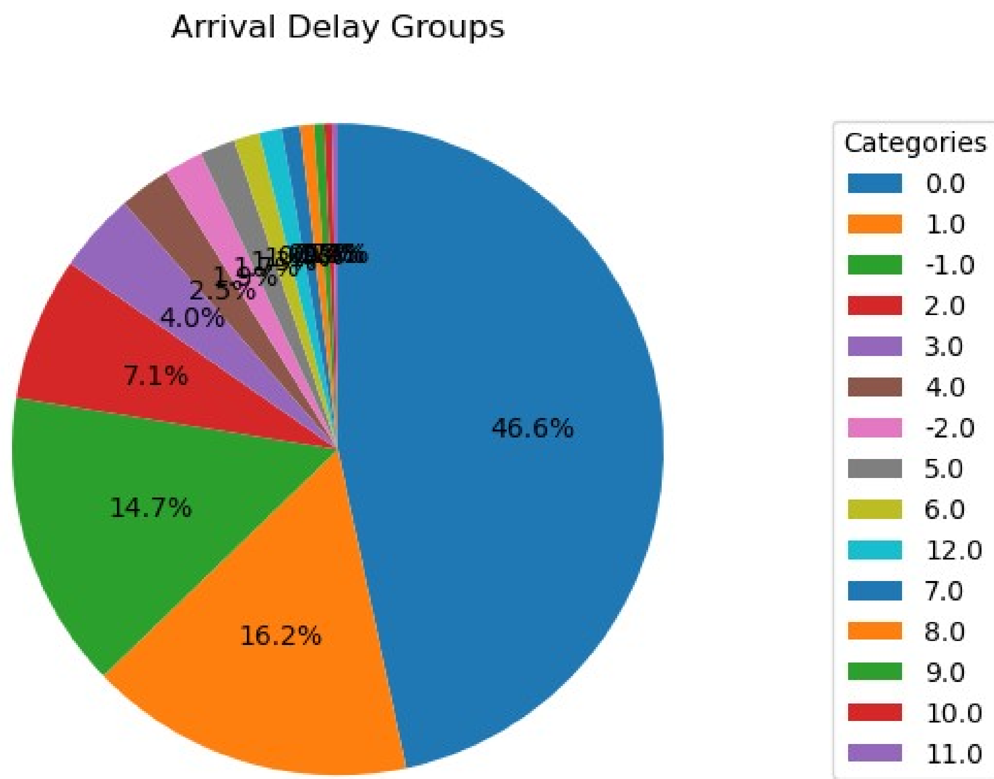

```
sns.violinplot(x=df['DepDelayMinutes'].dropna())
plt.xlabel('departure Delay (Minutes)')
plt.title('Violin Plot of Departure Delays')
plt.show()
```




```
sns.kdeplot(df['ArrDelayMinutes'].dropna(), fill=True)
plt.xlabel('Arrival Delay (Minutes)')
plt.title('Density Plot of Arrival Delays')
plt.show()
```




```
pie_chart(delay_airline_df, 'ArrivalDelayGroups','Arrival Delay
Groups')
```



Outliers

Check for Departure Outliers

In this section, we are identifying outliers.

```
Check_for_Outliers(cleaned_airline_df, 'DepDelayMinutes',
'DepDelayMinutes')
```

```
upper_bound = 42.461333684338044 | lower_bound = -21.46799990292402
Count of outlier more than upper_bound = 137727
percentage  of outlier more than upper_bound = 6.89%
```

```
Check_for_Outliers(cleaned_airline_df, 'DepDelayMinutes',
'DepDelayMinutes',250)
```

```
upper_bound = 250 | lower_bound = -21.46799990292402
Count of outlier more than upper_bound = 4252
percentage  of outlier more than upper_bound = 0.21%
```

```
Check_for_Outliers(cleaned_airline_df, 'DepDelayMinutes',
'DepDelayMinutes',500)
```

```
upper_bound = 500 | lower_bound = -21.46799990292402
Count of outlier more than upper_bound = 591
percentage  of outlier more than upper_bound = 0.03%
```

```python
Check_for_Outliers(cleaned_airline_df, 'DepDelayMinutes', 
'DepDelayMinutes',1000)
```

```
upper_bound = 1000 | lower_bound = -21.46799990292402
Count of outlier more than upper_bound = 123
percentage  of outlier more than upper_bound = 0.01%
```

Remove Outliers

A new dataset will be created by removing DepDelayMinutes outliers that exceed 250 minutes, representing 0.21% of the total data

```python
delay_airline_without_outliers_df = 
cleaned_airline_df[cleaned_airline_df['DepDelayMinutes']<250]
delay_airline_without_outliers_df.shape
```

```
(1959628, 87)
```

Check for Arrival Outliers

```python
delay_airline_without_outliers_df['ArrDelayMinutes'].describe()
```

```
count    1.954667e+06
mean     1.103441e+01
std      2.614815e+01
min      0.000000e+00
25%      0.000000e+00
50%      0.000000e+00
75%      1.000000e+01
max      1.430000e+03
Name: ArrDelayMinutes, dtype: float64
```

```python
Check_for_Outliers(delay_airline_without_outliers_df, 
'ArrDelayMinutes', 'ArrDelayMinutes')
```

```
upper_bound = 37.18256539942617 | lower_bound = -15.113741400248816
Count of outlier more than upper_bound = 166527
percentage  of outlier more than upper_bound = 8.5%
```

```python
Check_for_Outliers(delay_airline_without_outliers_df, 
'ArrDelayMinutes', 'ArrDelayMinutes',250)
```

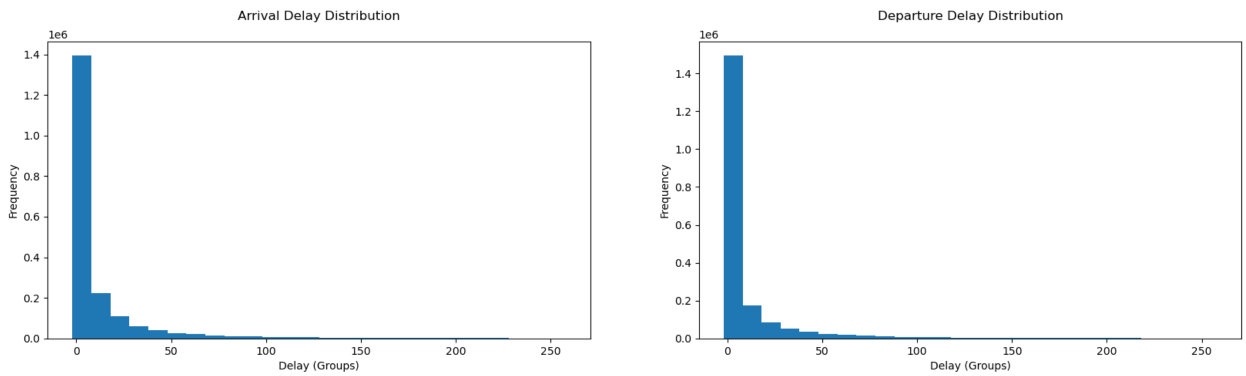
```
upper_bound = 250 | lower_bound = -15.113741400248816
Count of outlier more than upper_bound = 635
percentage  of outlier more than upper_bound = 0.03%
```

Remove Arrival Outliers

```python
# remove outliers
delay_airline_without_outliers_df = delay_airline_without_outliers_df[
    delay_airline_without_outliers_df['ArrDelayMinutes']<250]
delay_airline_without_outliers_df.shape
```

```
(1954010, 87)
```

```
two_hist_chart(delay_airline_without_outliers_df,
               ['ArrDelayMinutes', 'DepDelayMinutes'],
               ['Arrival Delay Distribution', 'Departure Delay
Distribution'],
                ['Delay (Groups)', 'Delay (Groups)'], 10)
```



Bivariate Exploration

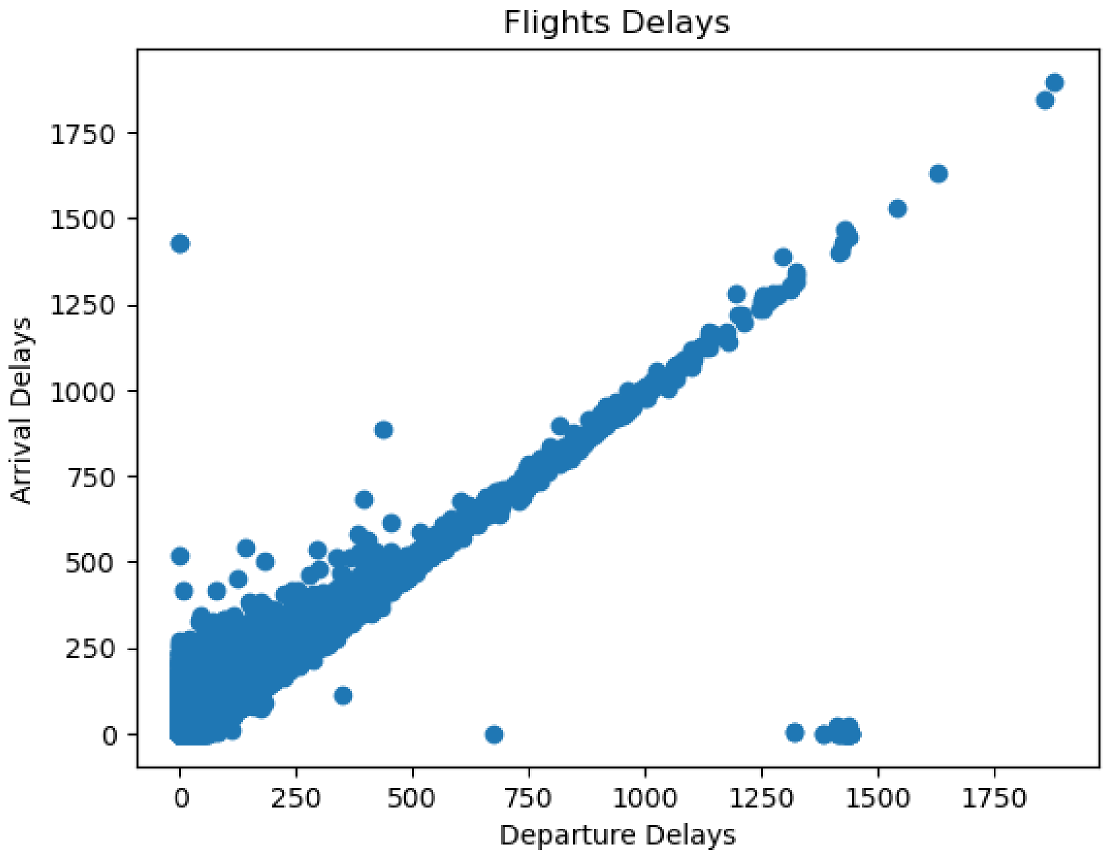
This section involves analyzing the relationship between two variables to understand how they interact with each other. This analysis helps to identify patterns, correlations, and potential causal relationships between pairs of variables.

Relation Between Arrival Delay & Departure Delay

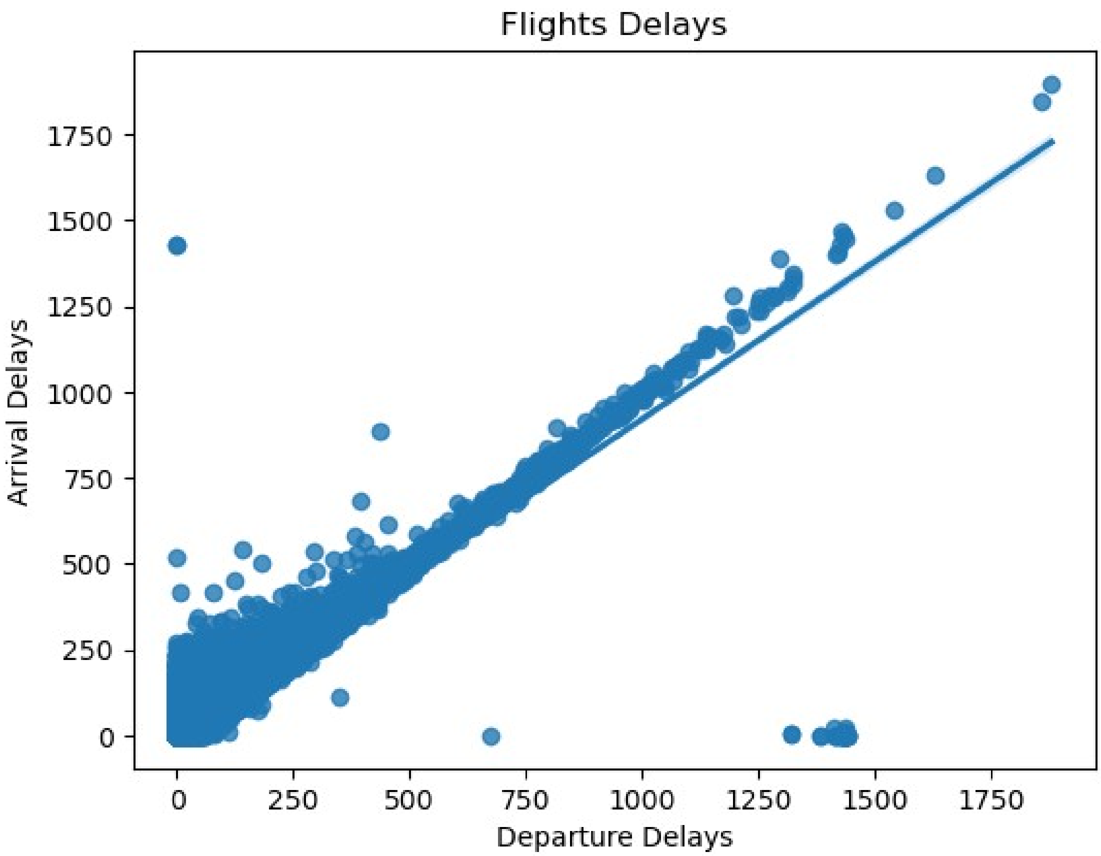
A new DataFrame is generated to illustrate the relationship between arrival and departure delays, where delay minutes are un-pivoted. In this DataFrame, 'DelayMinutes' represents the value from either 'ArrDelayMinutes' or 'DepDelayMinutes', and 'DelayType' indicates 0 for departure delay and 1 for arrival delay.

Note: in this relationship the outliers is not removed to reflect the full image

```
Scatter_plot(delay_airline_df,
             ['DepDelayMinutes', 'ArrDelayMinutes'],
             'Flights Delays',
             ['Departure Delays','Arrival Delays'])
```




```
regression_scatter_plot(delay_airline_df,
              ['DepDelayMinutes', 'ArrDelayMinutes'],
              'Flights Delays',
              ['Departure Delays','Arrival Delays'])
```

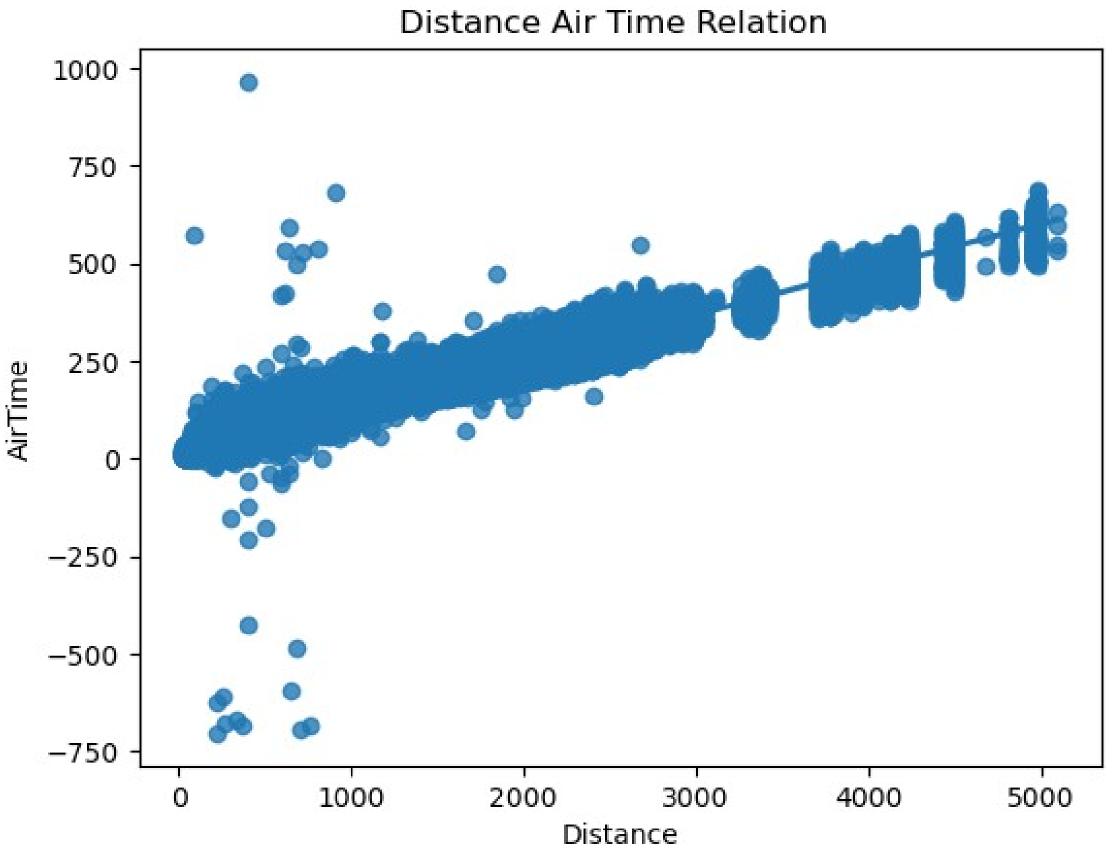


There is a clear linear relationship between arrival and departure delays, indicating that as departure delay increases, arrival delay also tends to rise proportionally. Consequently, the longer a flight is delayed at departure, the more likely it is to be delayed upon arrival. This direct correlation underscores the importance of minimizing departure delays, as they can have a cascading effect on arrival times, potentially disrupting schedules and affecting subsequent flights.

Distance Air Time Relation

he scatter plot of Distance versus AirTime reveals a linear trend, meaning that as the distance of the flight increases, the air time also increases.

```
regression_scatter_plot(cleaned_airline_df,
             ['Distance', 'AirTime'],
             'Distance Air Time Relation ',
             ['Distance','AirTime'])
```




```
cleaned_airline_df[cleaned_airline_df['AirTime']< 0].shape
```

```
(29, 87)
```

```
cleaned_airline_df[(cleaned_airline_df['AirTime']< 0) &
(cleaned_airline_df['Cancelled'] == 0)]
```

	Year	Quarter	Month	DayofMonth	DayOfWeek	FlightDate \
18428	2004	4	10	26	2	2004-10-26
72219	2004	3	9	26	7	2004-09-26
122559	2004	2	6	1	2	2004-06-01
158307	2004	1	1	14	3	2004-01-14
222240	2004	2	6	4	5	2004-06-04
383878	2004	1	3	2	2	2004-03-02
422021	2004	3	8	15	7	2004-08-15
433704	2004	1	1	3	6	2004-01-03
471541	2004	4	12	22	3	2004-12-22
523068	2004	4	12	22	3	2004-12-22
629659	2004	4	11	9	2	2004-11-09
664461	2004	2	6	13	7	2004-06-13
758073	2004	4	11	10	3	2004-11-10
889683	2004	2	6	23	3	2004-06-23
1146997	2004	2	5	16	7	2004-05-16
1177930	2004	2	5	22	6	2004-05-22

```
1203935  2004        4     12           19          7  2004-12-19
1261054  2004        1      1           29          4  2004-01-29
1368740  2004        4     10           19          2  2004-10-19
1399006  2004        2      6            1          2  2004-06-01
1470063  2004        1      2           23          1  2004-02-23
1567413  2004        3      7            9          5  2004-07-09
1631924  2004        1      1           19          1  2004-01-19
1636313  2004        1      2           20          5  2004-02-20
1688441  2004        2      5            3          1  2004-05-03
1772659  2004        1      2           12          4  2004-02-12
1775702  2003        1      3            7          5  2003-03-07
1785141  2004        1      2            2          1  2004-02-02
1792912  2004        2      5            9          7  2004-05-09

        Reporting_Airline  DOT_ID_Reporting_Airline  \
18428                  OH                     20417
72219                  OH                     20417
122559                 OO                     20304
158307                 OH                     20417
222240                 OH                     20417
383878                 OH                     20417
422021                 OH                     20417
433704                 OH                     20417
471541                 OH                     20417
523068                 OO                     20304
629659                 OH                     20417
664461                 OH                     20417
758073                 OH                     20417
889683                 OH                     20417
1146997                OH                     20417
1177930                OH                     20417
1203935                OH                     20417
1261054                OH                     20417
1368740                OH                     20417
1399006                OH                     20417
1470063                OH                     20417
1567413                OH                     20417
1631924                OH                     20417
1636313                OH                     20417
1688441                OH                     20417
1772659                OH                     20417
1775702                OO                     20304
1785141                OH                     20417
1792912                OO                     20304

        IATA_CODE_Reporting_Airline Tail_Number  ...  Div2Airport  \
18428                            OH      N995CA  ...          NaN
72219                            OH      N712CA  ...          NaN
122559                           OO      N298SW  ...          NaN
```

```
158307                               OH     N34CA  ...          NaN
222240                               OH    N965CA  ...          NaN
383878                               OH    N498CA  ...          NaN
422021                               OH    N999CA  ...          NaN
433704                               OH    N378CA  ...          NaN
471541                               OH    N447CA  ...          NaN
523068                               OO    N443SW  ...          NaN
629659                               OH    N416CA  ...          NaN
664461                               OH    N995CA  ...          NaN
758073                               OH    n408ca  ...          NaN
889683                               OH    N998CA  ...          NaN
1146997                              OH    N779CA  ...          NaN
1177930                              OH    N374CA  ...          NaN
1203935                              OH    N812CA  ...          NaN
1261054                              OH    N999CA  ...          NaN
1368740                              OH    N470CA  ...          NaN
1399006                              OH    N811CA  ...          NaN
1470063                              OH    N420CA  ...          NaN
1567413                              OH    N956CA  ...          NaN
1631924                              OH    N523CA  ...          NaN
1636313                              OH    N523CA  ...          NaN
1688441                              OH    N954CA  ...          NaN
1772659                              OH    N981CA  ...          NaN
1775702                              OO    N582SW  ...          NaN
1785141                              OH    N920CA  ...          NaN
1792912                              OO    N58733  ...          NaN

         Div2AirportID  Div2AirportSeqID  Div2WheelsOn Div2TotalGTime  \
18428              NaN               NaN           NaN            NaN

72219              NaN               NaN           NaN            NaN

122559             NaN               NaN           NaN            NaN

158307             NaN               NaN           NaN            NaN

222240             NaN               NaN           NaN            NaN

383878             NaN               NaN           NaN            NaN

422021             NaN               NaN           NaN            NaN

433704             NaN               NaN           NaN            NaN

471541             NaN               NaN           NaN            NaN

523068             NaN               NaN           NaN            NaN

629659             NaN               NaN           NaN            NaN
```

```
664461                 NaN              NaN           NaN            NaN
758073                 NaN              NaN           NaN            NaN
889683                 NaN              NaN           NaN            NaN
1146997                NaN              NaN           NaN            NaN
1177930                NaN              NaN           NaN            NaN
1203935                NaN              NaN           NaN            NaN
1261054                NaN              NaN           NaN            NaN
1368740                NaN              NaN           NaN            NaN
1399006                NaN              NaN           NaN            NaN
1470063                NaN              NaN           NaN            NaN
1567413                NaN              NaN           NaN            NaN
1631924                NaN              NaN           NaN            NaN
1636313                NaN              NaN           NaN            NaN
1688441                NaN              NaN           NaN            NaN
1772659                NaN              NaN           NaN            NaN
1775702                NaN              NaN           NaN            NaN
1785141                NaN              NaN           NaN            NaN
1792912                NaN              NaN           NaN            NaN

        Div2LongestGTime Div2WheelsOff  Div2TailNum DayOfWeek_Desc  \
18428                NaN           NaN          NaN        Tuesday
72219                NaN           NaN          NaN         Sunday
122559               NaN           NaN          NaN        Tuesday
158307               NaN           NaN          NaN      Wednesday
222240               NaN           NaN          NaN         Friday
383878               NaN           NaN          NaN        Tuesday
422021               NaN           NaN          NaN         Sunday
433704               NaN           NaN          NaN       Saturday
471541               NaN           NaN          NaN      Wednesday
523068               NaN           NaN          NaN      Wednesday
629659               NaN           NaN          NaN        Tuesday
664461               NaN           NaN          NaN         Sunday
```

758073	NaN	NaN	NaN	Wednesday
889683	NaN	NaN	NaN	Wednesday
1146997	NaN	NaN	NaN	Sunday
1177930	NaN	NaN	NaN	Saturday
1203935	NaN	NaN	NaN	Sunday
1261054	NaN	NaN	NaN	Thursday
1368740	NaN	NaN	NaN	Tuesday
1399006	NaN	NaN	NaN	Tuesday
1470063	NaN	NaN	NaN	Monday
1567413	NaN	NaN	NaN	Friday
1631924	NaN	NaN	NaN	Monday
1636313	NaN	NaN	NaN	Friday
1688441	NaN	NaN	NaN	Monday
1772659	NaN	NaN	NaN	Thursday
1775702	NaN	NaN	NaN	Friday
1785141	NaN	NaN	NaN	Monday
1792912	NaN	NaN	NaN	Sunday

	Quarter_Desc
18428	Q4
72219	Q3
122559	Q2
158307	Q1
222240	Q2
383878	Q1
422021	Q3
433704	Q1
471541	Q4
523068	Q4
629659	Q4
664461	Q2
758073	Q4
889683	Q2
1146997	Q2
1177930	Q2
1203935	Q4
1261054	Q1
1368740	Q4
1399006	Q2
1470063	Q1
1567413	Q3
1631924	Q1
1636313	Q1
1688441	Q2
1772659	Q1
1775702	Q1
1785141	Q1
1792912	Q2

```
[29 rows x 87 columns]
```

There exists 29 records where the flight time is less than zero and distance is greater than zero and the trip is not cancelled which can be marked as data issue

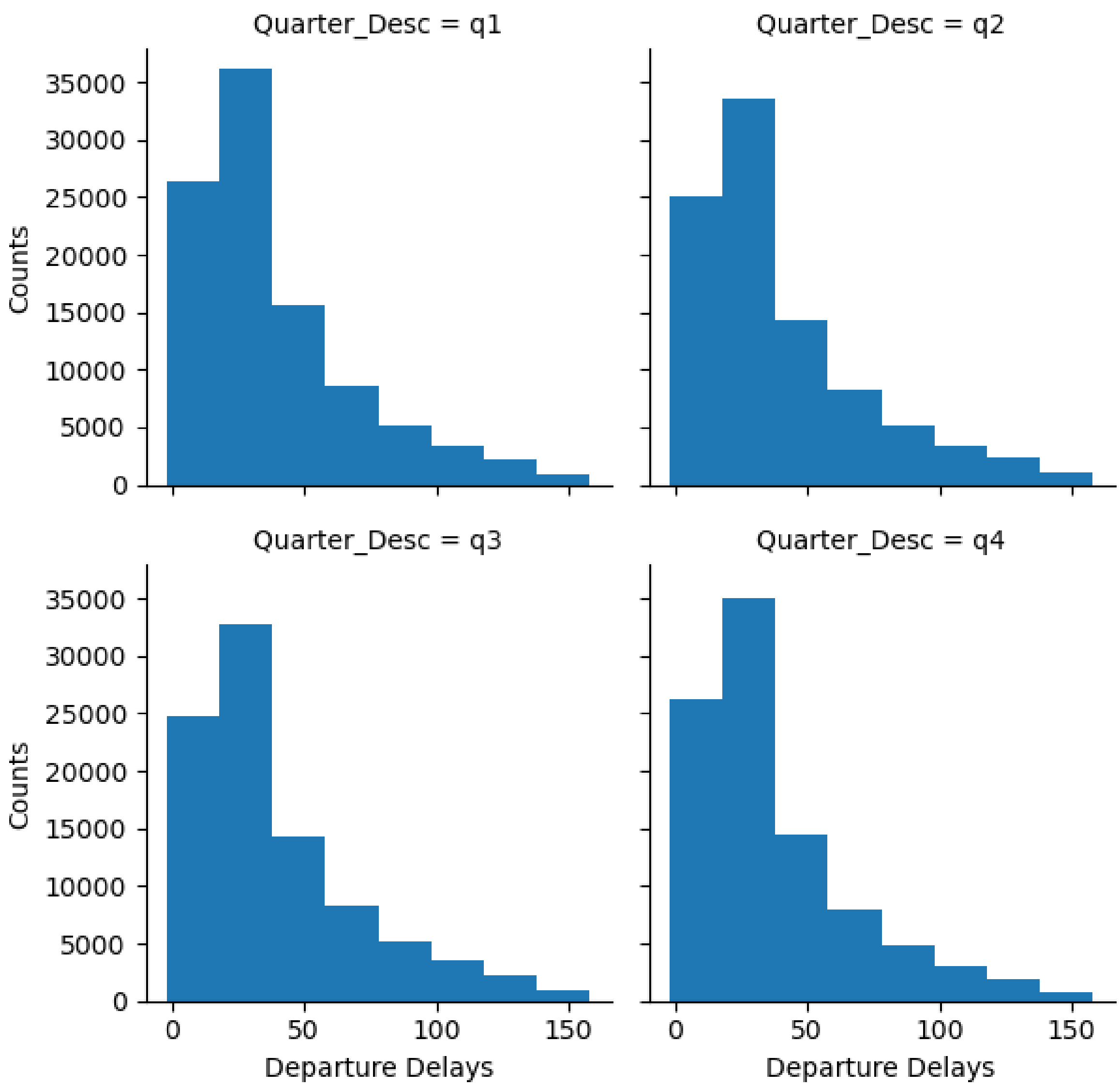
Delay Distribution

```
delay_airline_df_copy = delay_airline_df.copy()

quarter_mapping = {1: 'q1', 2: 'q2', 3: 'q3', 4: 'q4'}
delay_airline_df_copy['Quarter_Desc'] =
delay_airline_df_copy['Quarter'].map(quarter_mapping)

day_of_week_mapping = {
    1: 'Monday',
    2: 'Tuesday',
    3: 'Wednesday',
    4: 'Thursday',
    5: 'Friday',
    6: 'Saturday',
    7: 'Sunday'
}

# Apply the mapping to the 'DayOfWeek' column
delay_airline_df_copy['DayOfWeek_Desc'] =
delay_airline_df_copy['DayOfWeek'].map(day_of_week_mapping)

delay_airline_df_copy['DayOfWeek_Desc']

0             Friday
2           Saturday
4             Sunday
5          Wednesday
6             Monday
             ...
1999987       Monday
1999988       Friday
1999993     Thursday
1999995       Sunday
1999998      Tuesday
Name: DayOfWeek_Desc, Length: 1060475, dtype: object
```

Departure Delay Distribution by Quarter

```
FacetGrid(delay_airline_df_copy[(delay_airline_df_copy['DepDelayMinute
s']>10) & (delay_airline_df_copy['DepDelayMinutes']<150)],
           value_column='DepDelayMinutes',
           class_column='Quarter_Desc',
           bin_size=20,
```

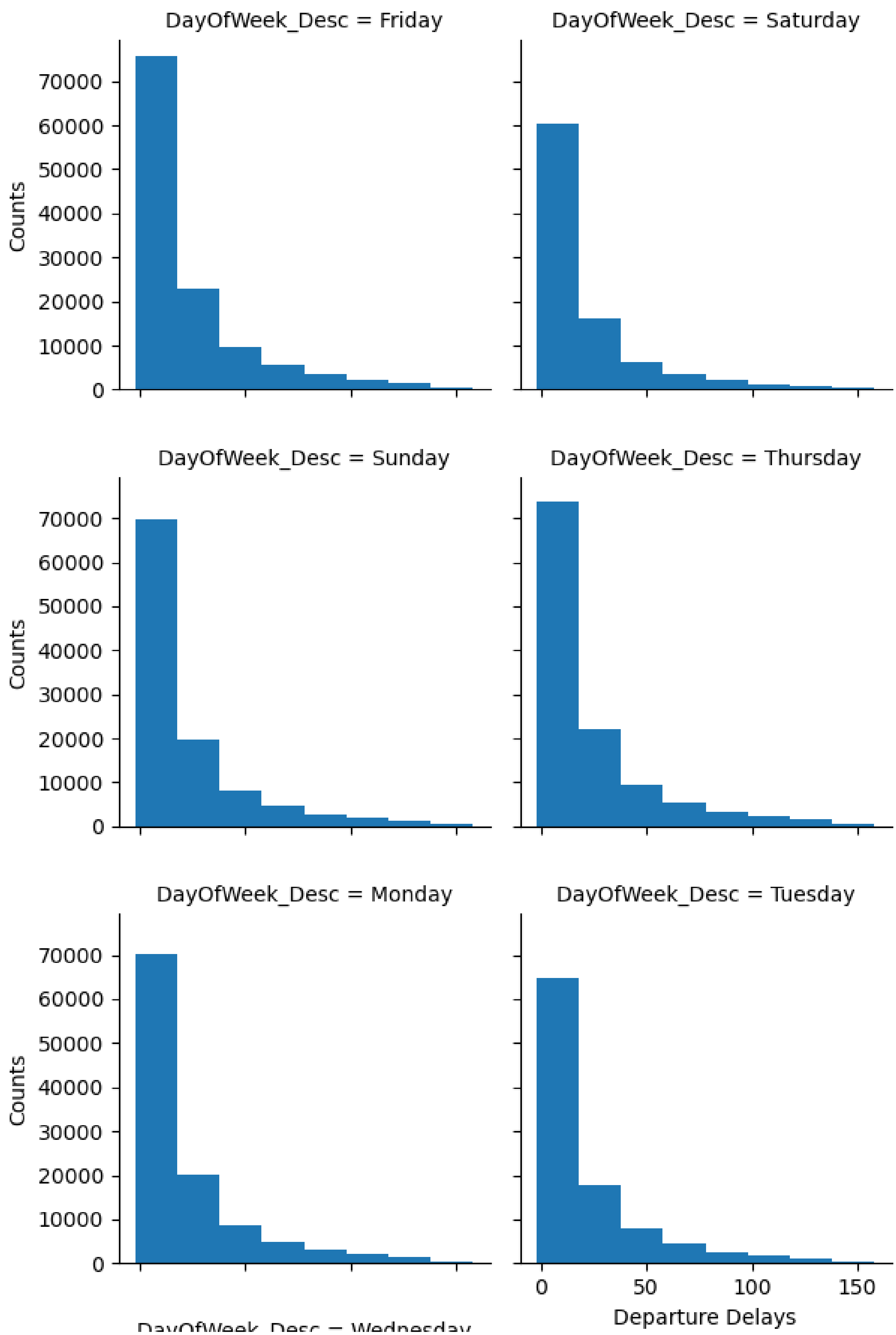
```
            title='Flights Delays',
            xyLabels=['Departure Delays', 'Counts'])
```



there is no large differences in delay distribution per Quarter

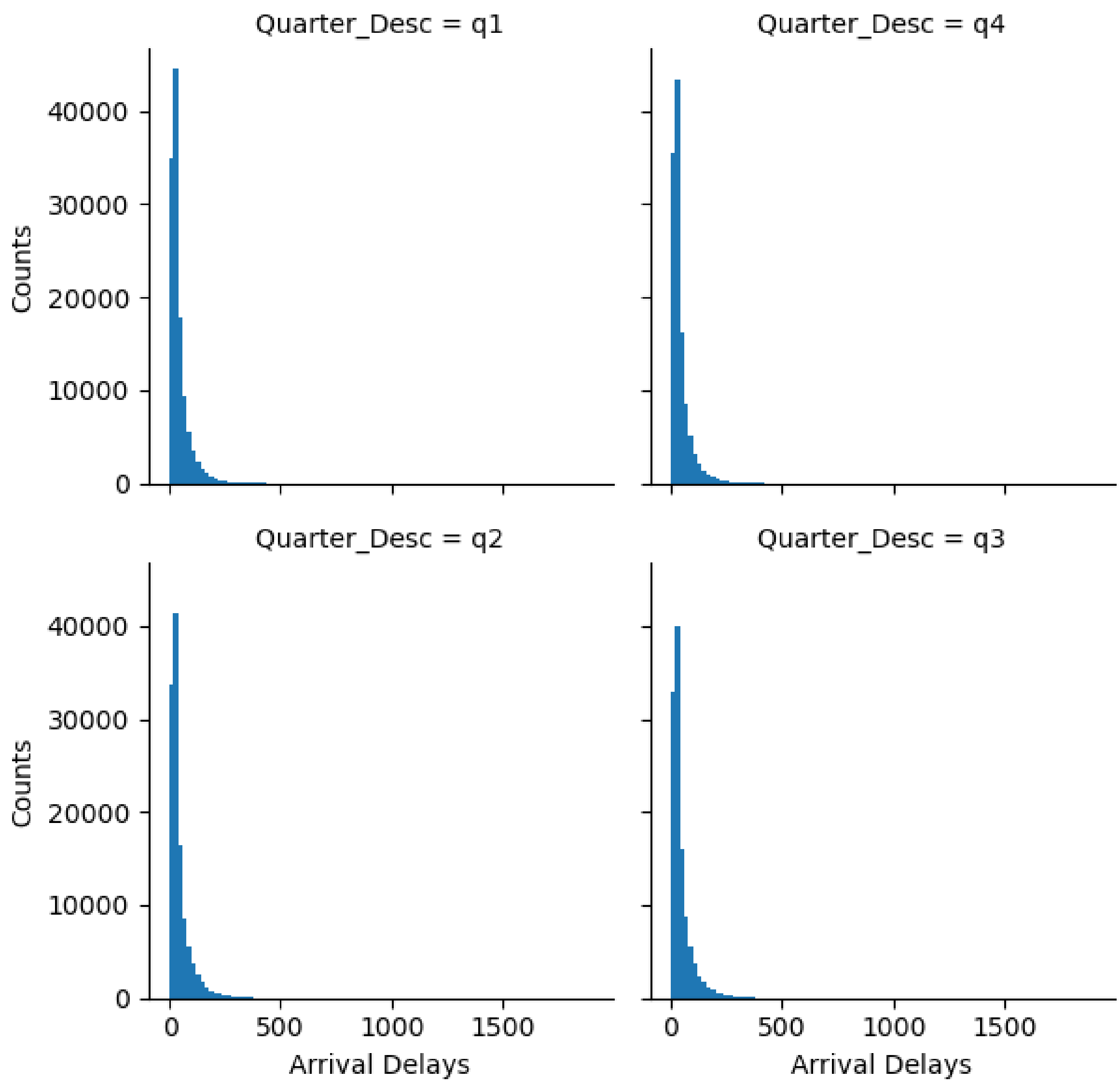
Departure Delay Distribution by Day Of Week

```
FacetGrid(delay_airline_df_copy[(delay_airline_df_copy['DepDelayMinute
s']>0) & (delay_airline_df_copy['DepDelayMinutes']<150)],
            value_column='DepDelayMinutes',
            class_column='DayOfWeek_Desc',
            bin_size=20,
            title='Flights Delays',
            xyLabels=['Departure Delays', 'Counts'])
```


DayOfWeek_Desc = Friday
DayOfWeek_Desc = Saturday
DayOfWeek_Desc = Sunday
DayOfWeek_Desc = Thursday
DayOfWeek_Desc = Monday
DayOfWeek_Desc = Tuesday
Counts
0
10000
20000
30000
40000
50000
60000
70000
0
50
100
150
Departure Delays

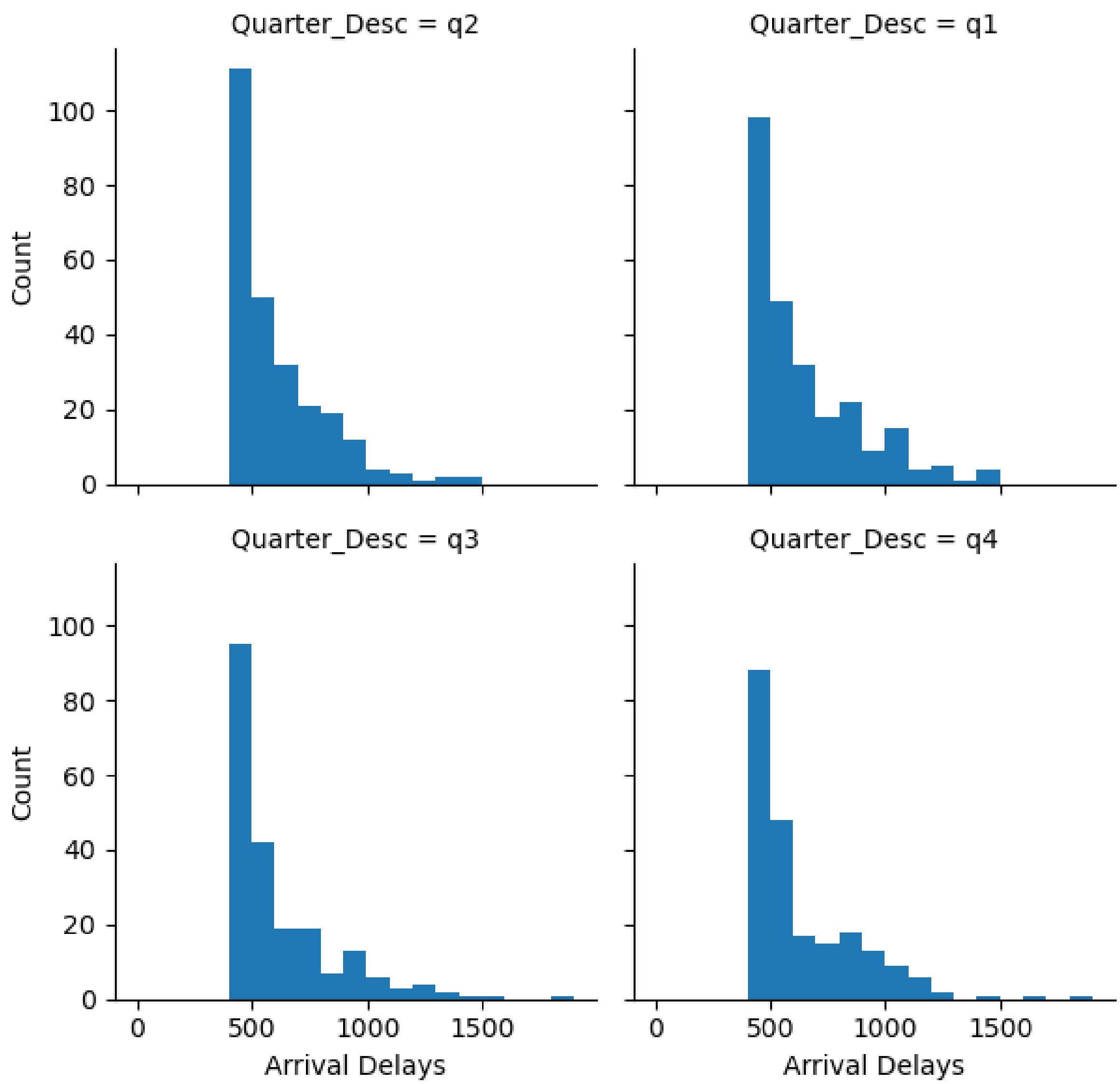
** Delay Distribution on saturday is less than the rest of the other week days

```
FacetGrid(delay_airline_df_copy[(delay_airline_df_copy['ArrDelayMinute
s']>10) ],
          value_column='ArrDelayMinutes',
          class_column='Quarter_Desc',
          bin_size=20,
          title='Flights Delays',
          xyLabels=['Arrival Delays', 'Counts'])
```

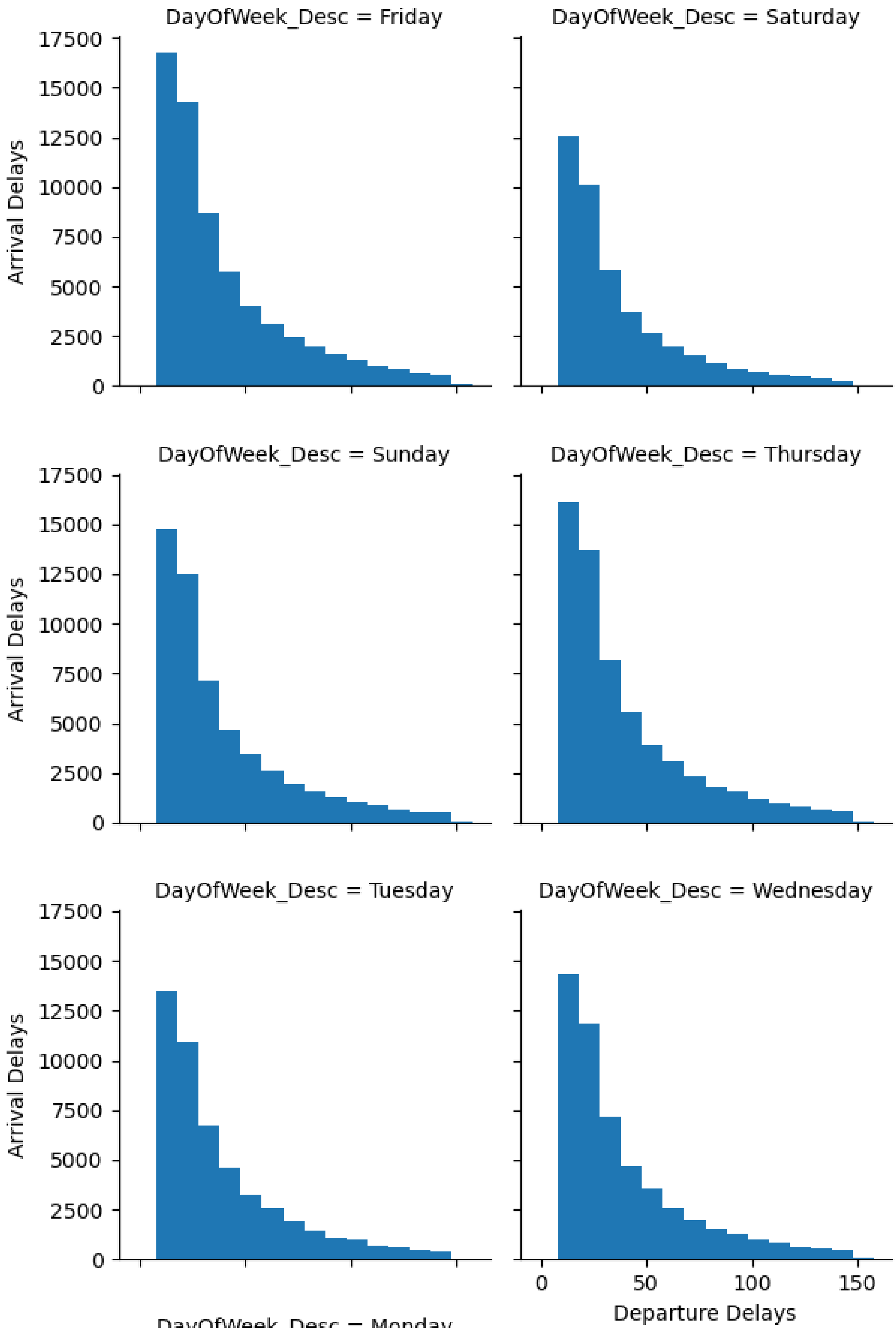



```
FacetGrid(delay_airline_df_copy[(delay_airline_df_copy['ArrDelayMinute
s']>400) ],
          value_column='ArrDelayMinutes',
          class_column='Quarter_Desc',
          bin_size=100,
```

```
            title='Flights Delays',
            xyLabels=['Arrival Delays', 'Count'])
```

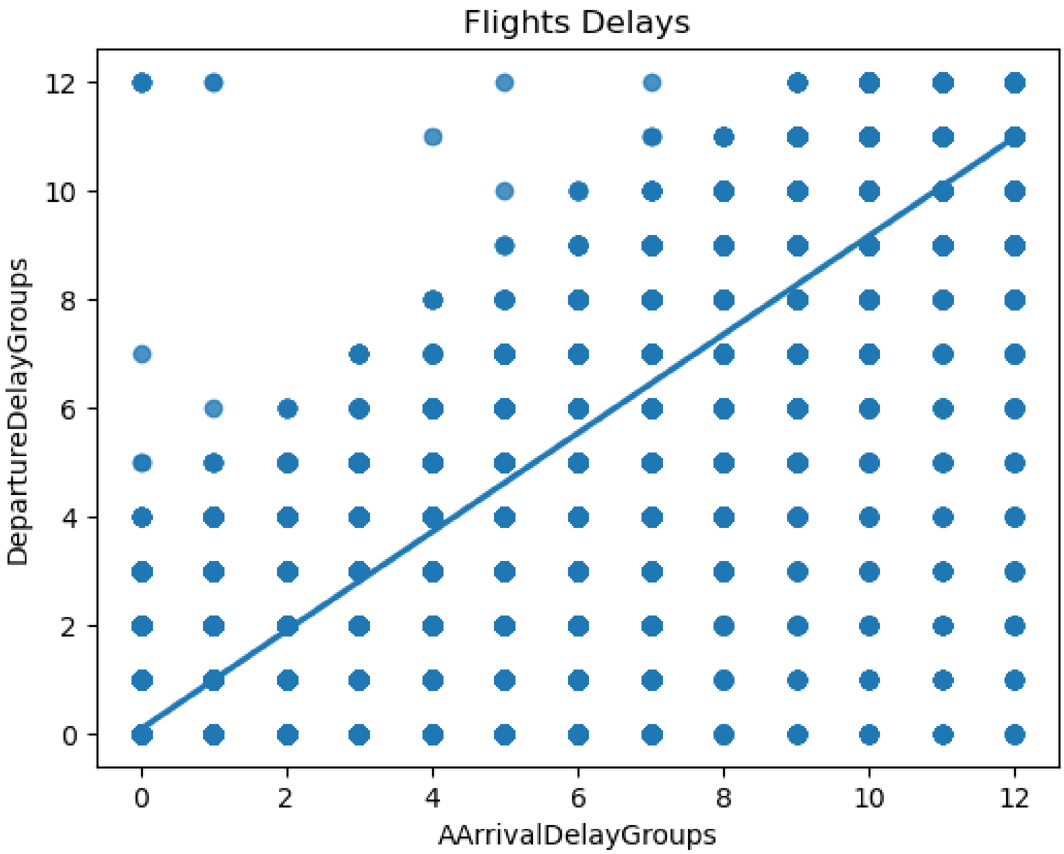



```
FacetGrid(delay_airline_df_copy[(delay_airline_df_copy['DepDelayMinute
s']>10) & (delay_airline_df_copy['DepDelayMinutes']<150)],
           'DepDelayMinutes',
           'DayOfWeek_Desc',
            10,
           'Flights Delays',
           ['Departure Delays','Arrival Delays'])
```

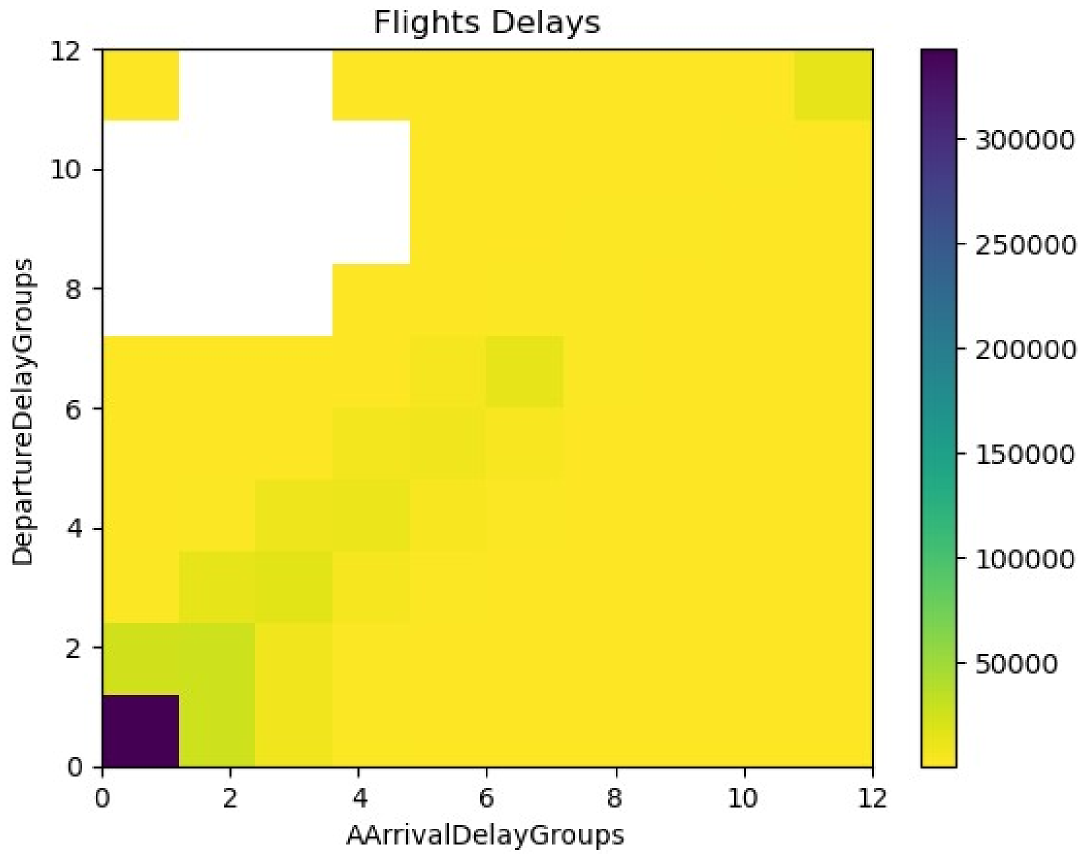


DayOfWeek_Desc = Friday
DayOfWeek_Desc = Saturday
DayOfWeek_Desc = Sunday
DayOfWeek_Desc = Thursday
DayOfWeek_Desc = Tuesday
DayOfWeek_Desc = Wednesday
Arrival Delays
0
2500
5000
7500
10000
12500
15000
17500
0
50
100
150
Departure Delays
DayOfWeek_Desc = Monday

```
regression_scatter_plot( delay_airline_df[(delay_airline_df['ArrDelayM
inutes']>0) & (delay_airline_df['DepDelayMinutes']>0)],
    ['ArrivalDelayGroups', 'DepartureDelayGroups'],
    'Flights Delays',
    ['AArrivalDelayGroups','DepartureDelayGroups'])
```

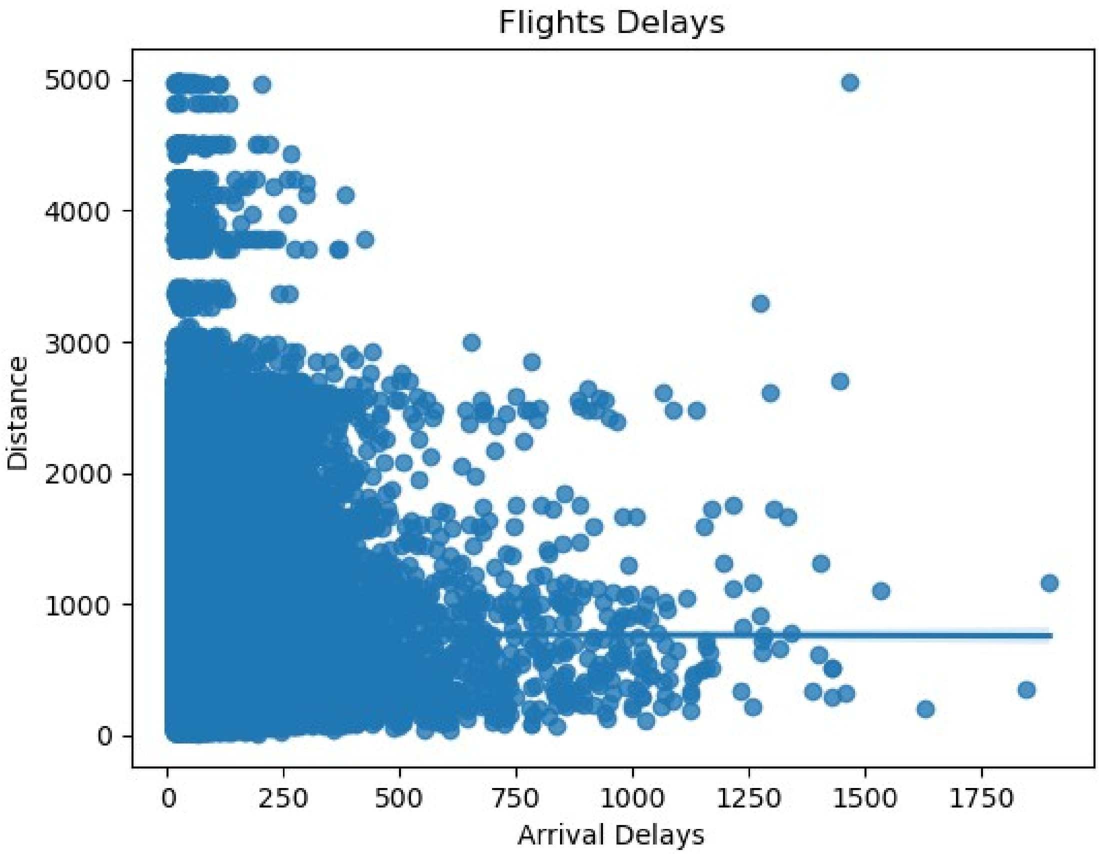



```
heat_map(
    delay_airline_df[(delay_airline_df['ArrDelayMinutes']>0) &
(delay_airline_df['DepDelayMinutes']>0)],
    ['ArrivalDelayGroups', 'DepartureDelayGroups'],
    'Flights Delays',
    ['AArrivalDelayGroups','DepartureDelayGroups'])
```



Is there a relation between delay more than 15 min with flight destance?

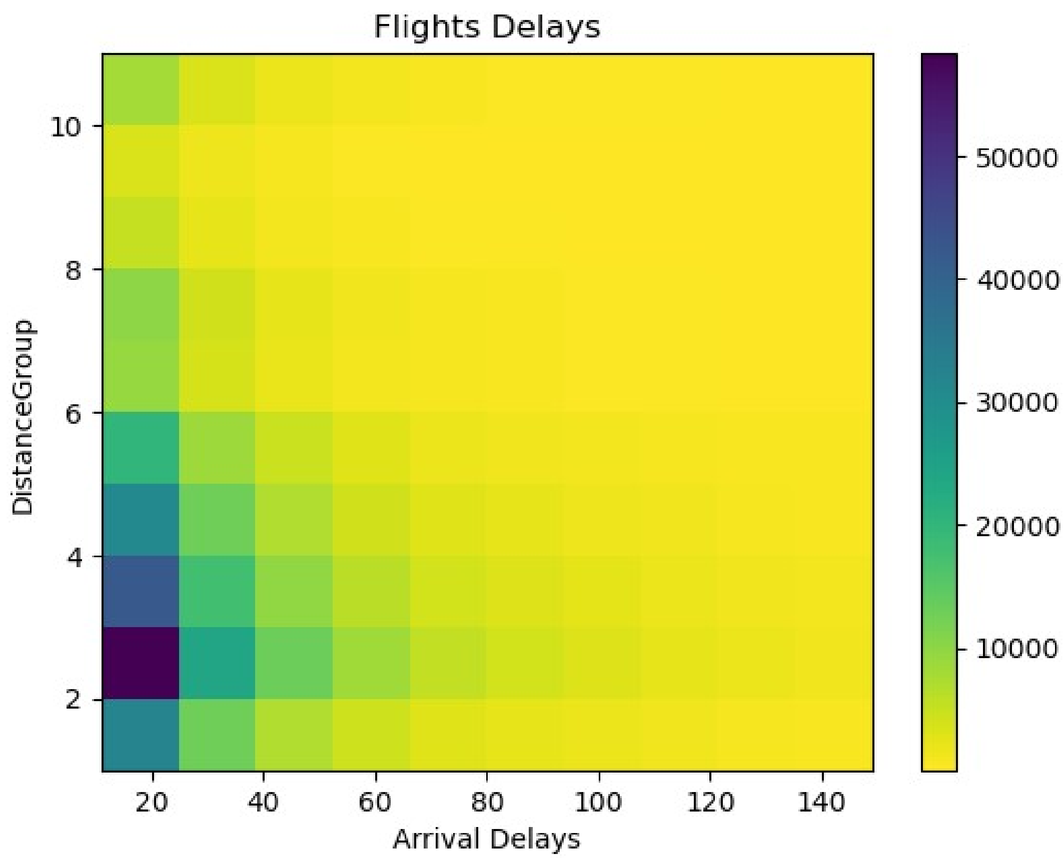
```
regression_scatter_plot(delay_airline_df[delay_airline_df['ArrDelayMin
utes']>15],
             ['ArrDelayMinutes', 'Distance'],
             'Flights Delays',
             ['Arrival Delays','Distance'])
```



conclusion: there is no relationship between distance and Arrival delay

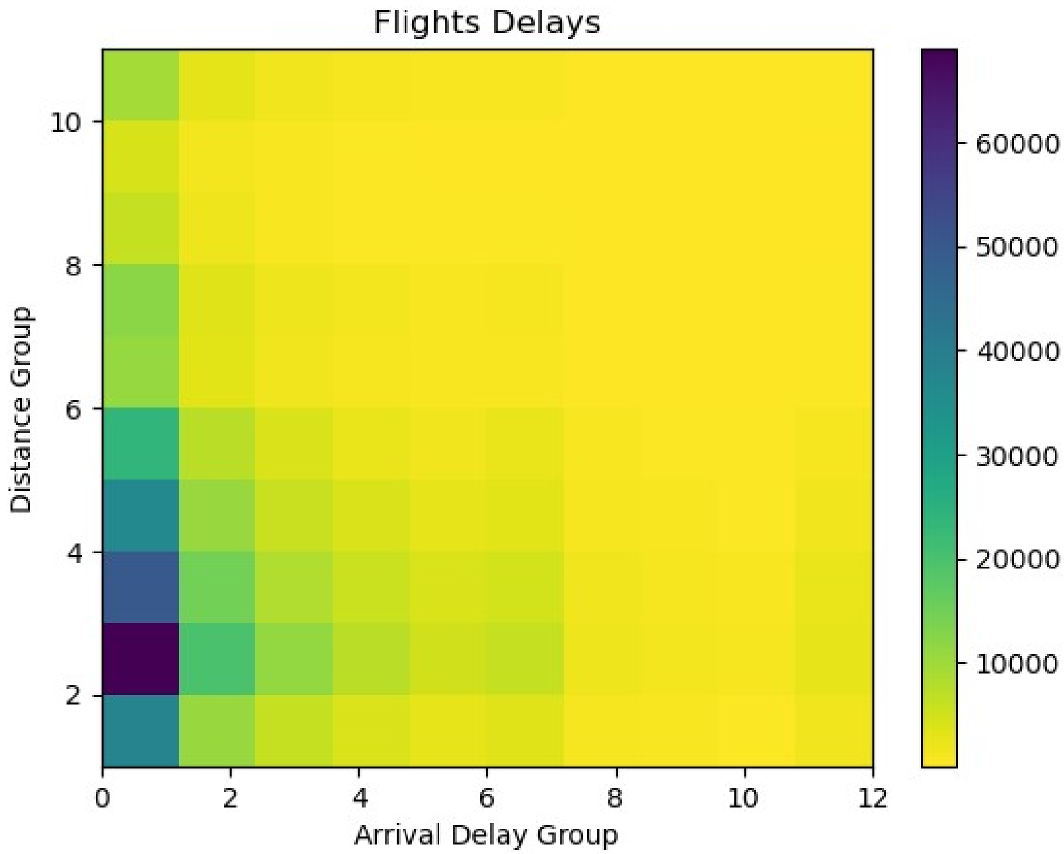
By analyzing the heatmap below, it is evident that the values are predominantly concentrated in the arrival delay of 20 minutes within Distance Group 2. This is followed by Distance Group 4, then Group 1, and Group 5.

```
heat_map(
    delay_airline_df[(delay_airline_df['ArrDelayMinutes']>10) &
(delay_airline_df['ArrDelayMinutes']<150)],
    ['ArrDelayMinutes', 'DistanceGroup'],
    'Flights Delays',
    ['Arrival Delays','DistanceGroup'])
```



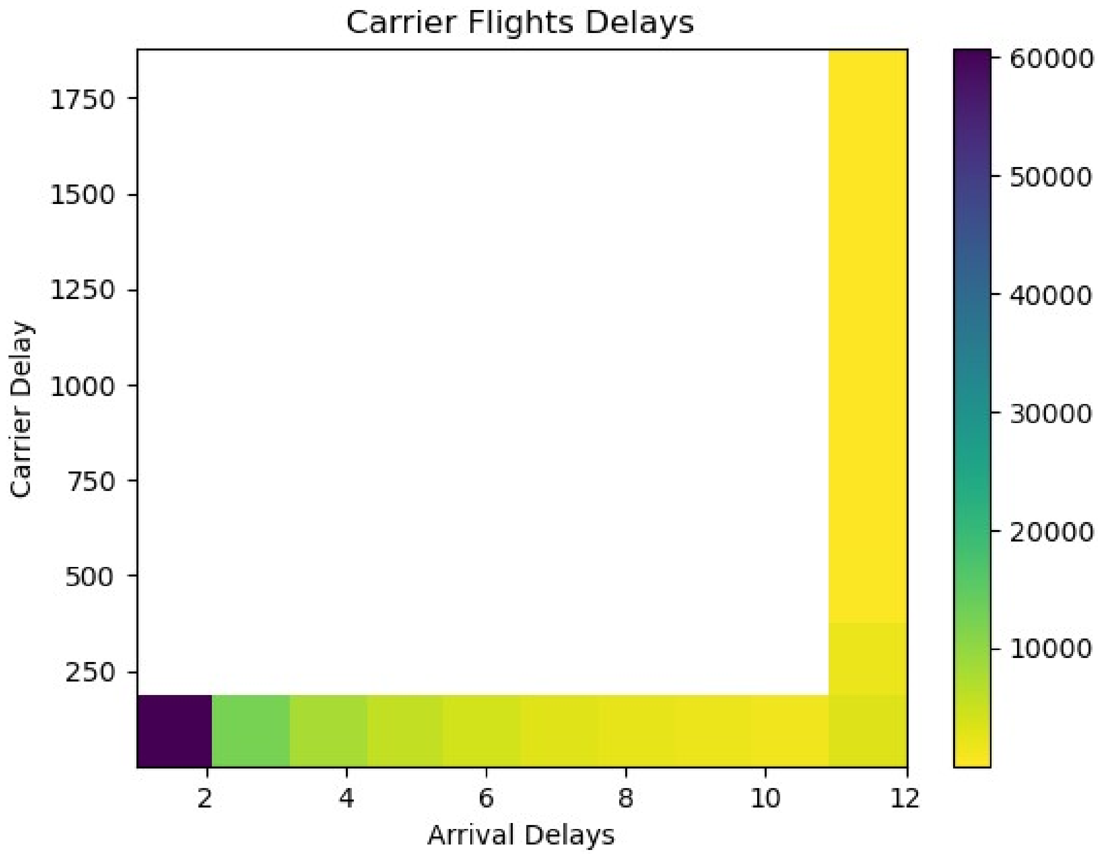
In the heatmap below, the feature used is the arrival delay group rather than minutes, yet it yields the same result.

```
heat_map(
    delay_airline_df[(delay_airline_df['ArrDelayMinutes']>10) &
(delay_airline_df['ArrDelayMinutes']<250)],
    ['ArrivalDelayGroups', 'DistanceGroup'],
    'Flights Delays',
    ['Arrival Delay Group','Distance Group'])
```



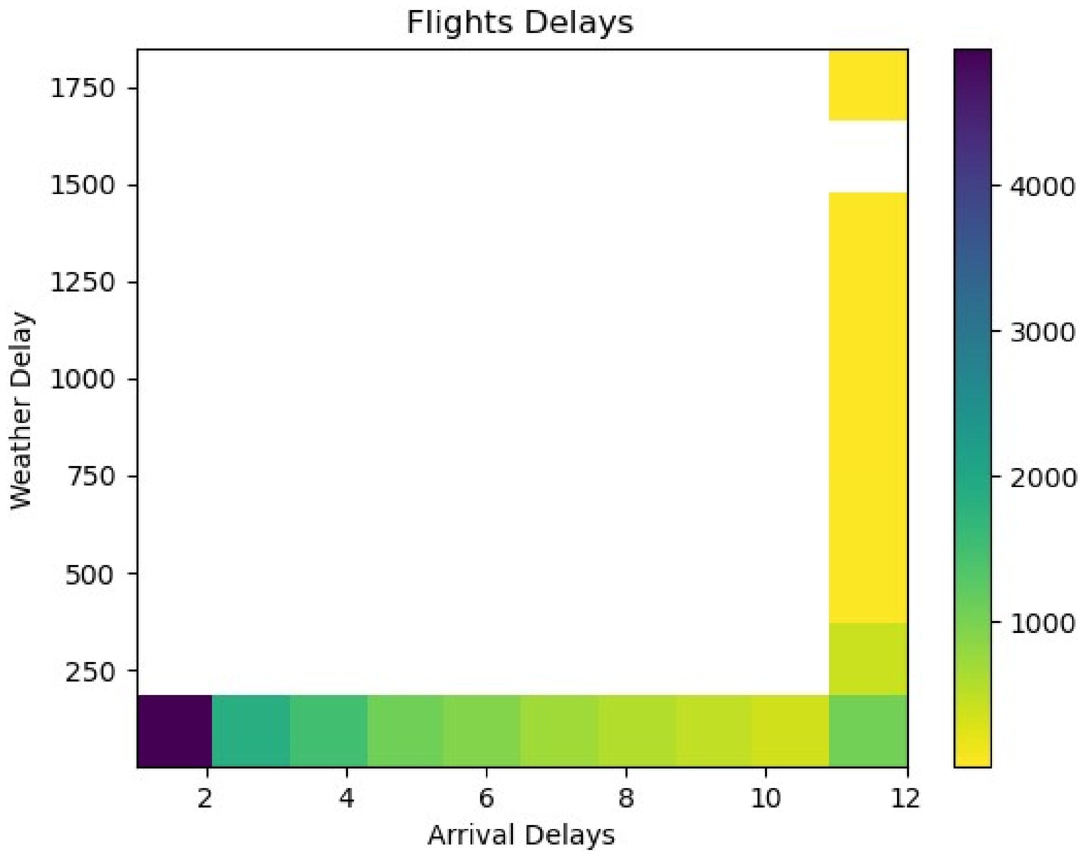
Observe the distribution pattern of points for CarrierDelay with Arrival Delays

```
heat_map(
    delay_airline_df[delay_airline_df['CarrierDelay']>0],
    ['ArrivalDelayGroups', 'CarrierDelay'],
    'Carrier Flights Delays',
    ['Arrival Delays','Carrier Delay'])
```



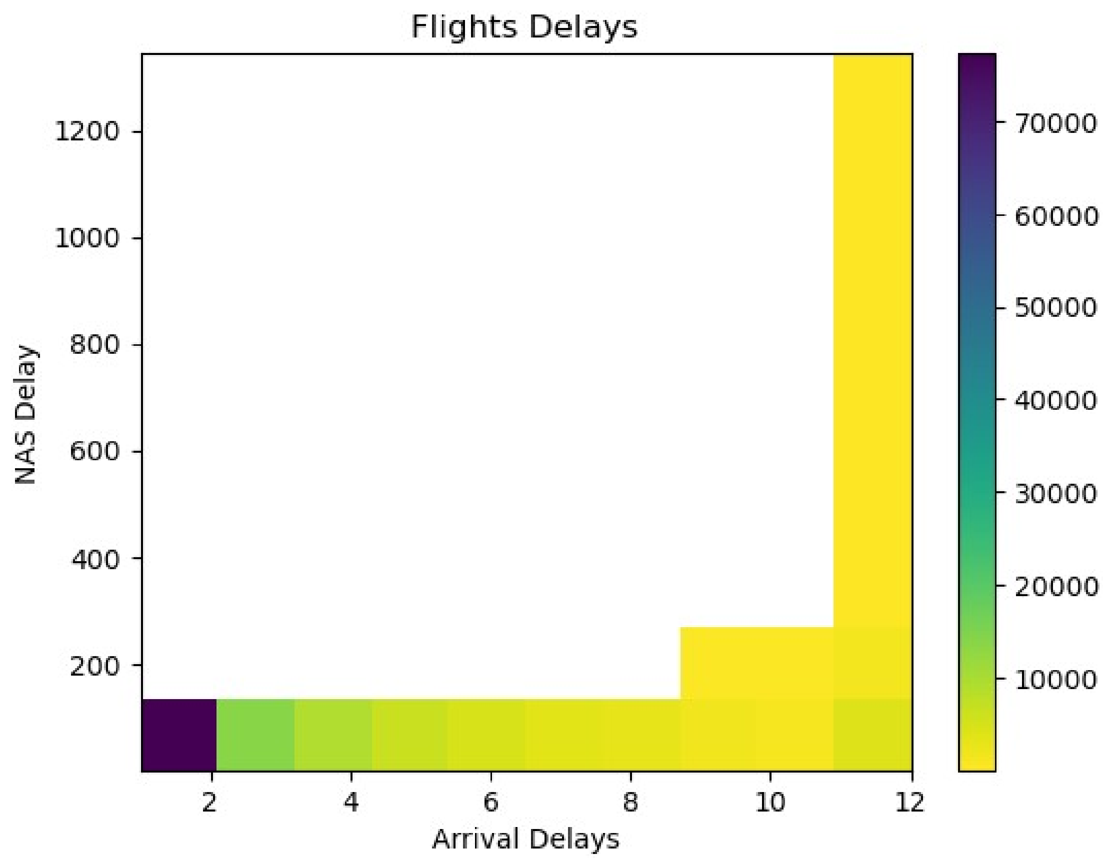
Observe the distribution pattern of points for WeatherDelay with Arrival Delays

```
heat_map(
    delay_airline_df[delay_airline_df['WeatherDelay']>0],
    ['ArrivalDelayGroups', 'WeatherDelay'],
    'Flights Delays',
    ['Arrival Delays','Weather Delay'])
```



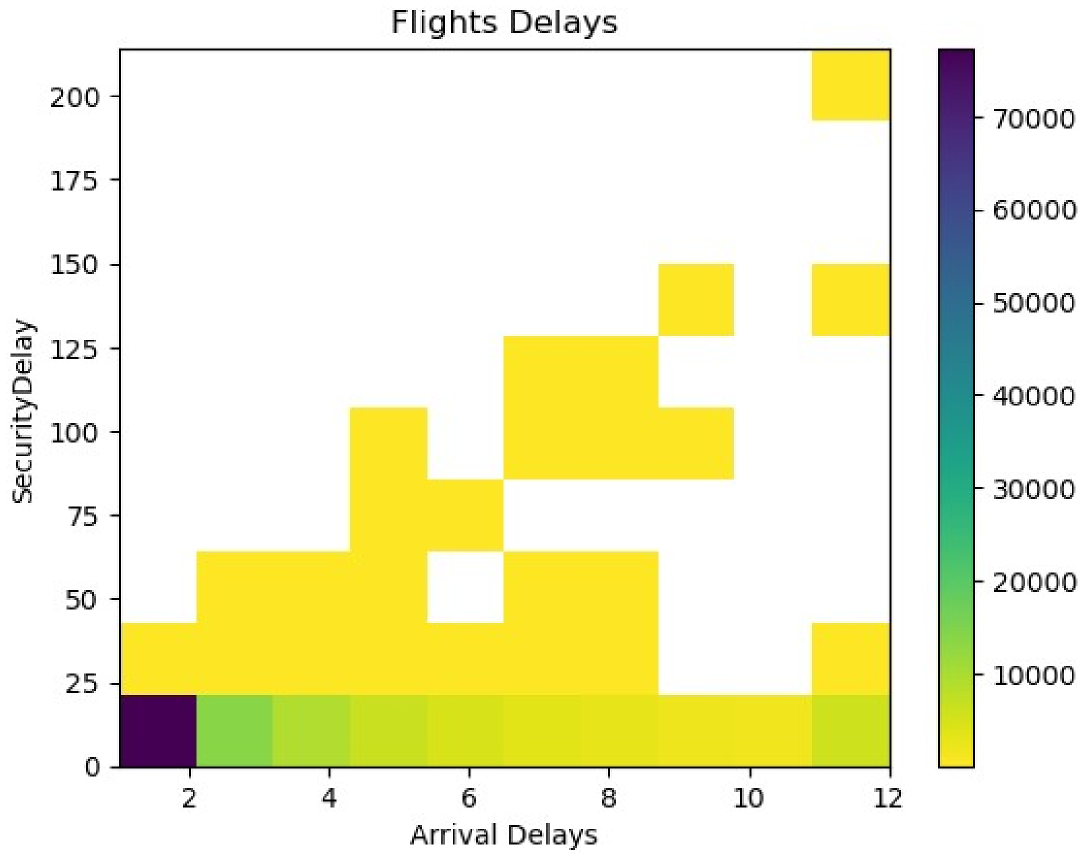
Observe the distribution pattern of points for NAS Delay with Arrival Delays

```
heat_map(
    delay_airline_df[delay_airline_df['NASDelay']>0],
    ['ArrivalDelayGroups', 'NASDelay'],
    'Flights Delays',
    ['Arrival Delays','NAS Delay'])
```



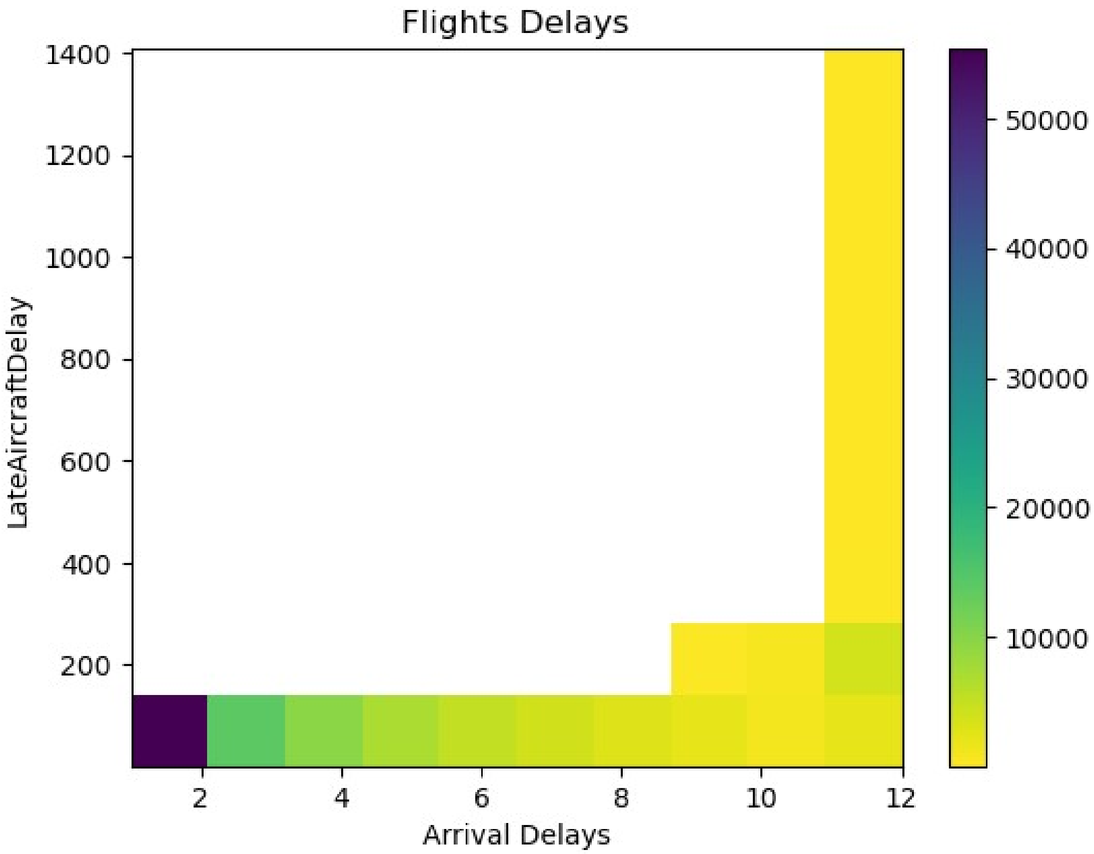
Observe the distribution pattern of points for SecurityDelay with Arrival Delays

```
heat_map(
    delay_airline_df[delay_airline_df['NASDelay']>0],
    ['ArrivalDelayGroups', 'SecurityDelay'],
    'Flights Delays',
    ['Arrival Delays','SecurityDelay'])
```



Observe the distribution pattern of points for LateAircraftDelay with Arrival Delays

```
heat_map(
    delay_airline_df[delay_airline_df['LateAircraftDelay']>0],
    ['ArrivalDelayGroups', 'LateAircraftDelay'],
    'Flights Delays',
    ['Arrival Delays','LateAircraftDelay'])
```




```
unpivoted_delay_type_df = pd.melt(
    delay_airline_df,  # Pass the DataFrame directly
    value_vars=['DepDelayMinutes', 'ArrDelayMinutes'],  # Specify the
columns to unpivot
    var_name='DelayType',  # Name for the new variable column
    value_name='DelayMinutes'  # Name for the new value column
)

# Map 'DelayType' to 0 for 'DepDelayMinutes' and 1 for
'ArrDelayMinutes'
unpivoted_delay_type_df['DelayType'] =
unpivoted_delay_type_df['DelayType'].map({
    'DepDelayMinutes': 0,
    'ArrDelayMinutes': 1
})

# View the new DataFrame
unpivoted_delay_type_df.head()
```

	DelayType	DelayMinutes
0	0	19.0
1	0	14.0
2	0	51.0

```
3          0          0.0
4          0          0.0
```

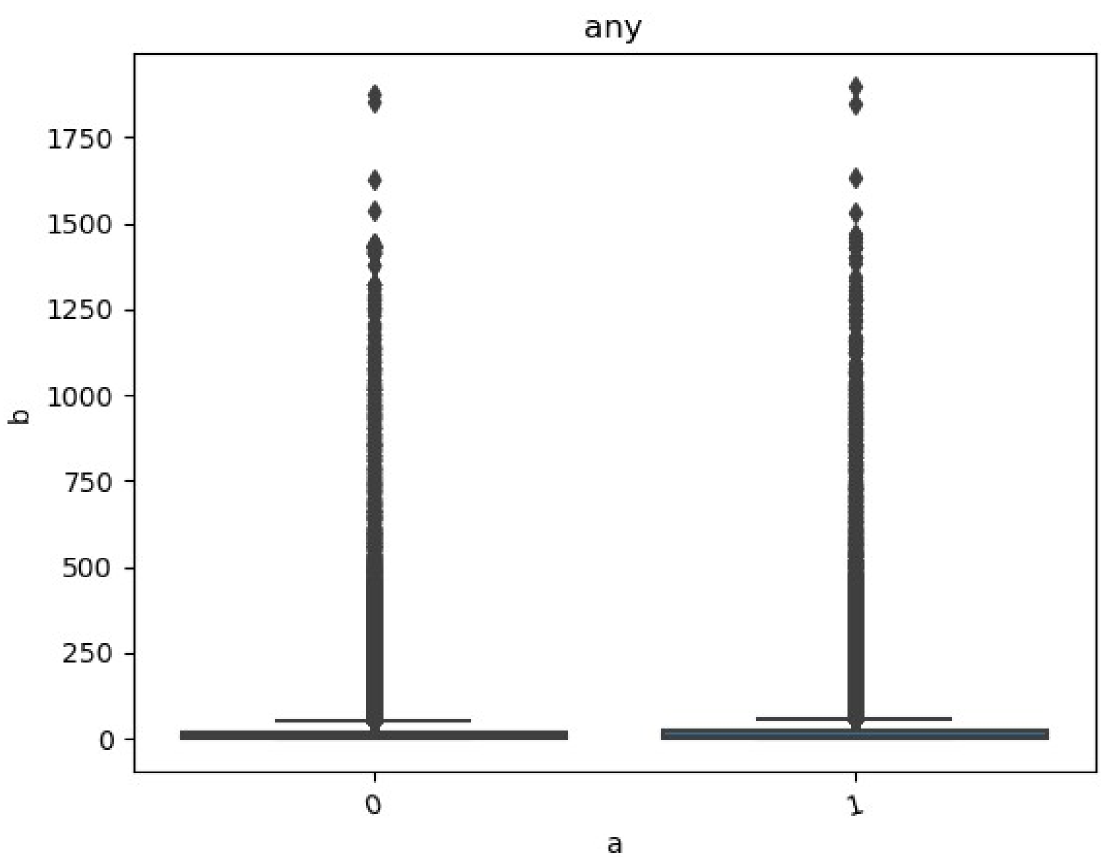
```
unpivoted_delay_type_df.tail()
```

	DelayType	DelayMinutes
2120945	1	0.0
2120946	1	1.0
2120947	1	4.0
2120948	1	0.0
2120949	1	0.0

```
unpivoted_delay_type_df.shape
```

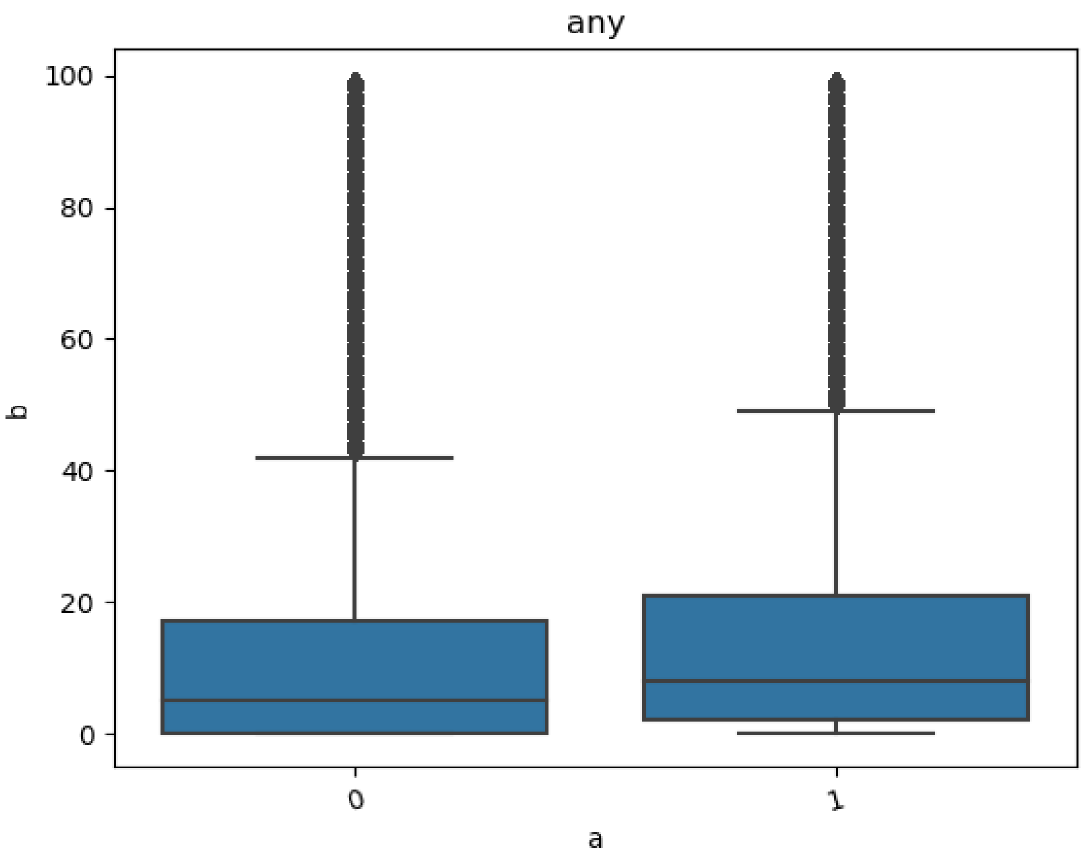
```
(2120950, 2)
```

```
# show scatter plot for Arrival delay
box_plot(unpivoted_delay_type_df,'DelayType',['dep',
'arr'],'DelayMinutes','any',['a','b'])
```




```
# show scatter plot for Arrival delay
box_plot(unpivoted_delay_type_df[unpivoted_delay_type_df['DelayMinutes
']<100]
```

```
         ,'DelayType',
         ['dep', 'arr'],
         'DelayMinutes',
         'any',
         ['a','b'])
```




```
Reporting_Airline_count = df['Reporting_Airline'].value_counts()
print(Reporting_Airline_count)

WN          306238
DL          264455
AA          234730
UA          194294
US          172532
NW          109336
OO          107153
CO           90931
MQ           78113
EV           67600
AS           49656
TW           38531
B6           37638
```

```
HP          37630
XE          35645
FL          25954
OH          24786
YV          22555
9E          19666
F9          14320
HA          11683
EA           9375
PI           8954
NK           8647
YX           7569
DH           7165
VX           3994
PA (1)       3168
G4           2306
TZ           2163
KH           1576
PS            867
ML (1)        770
Name: Reporting_Airline, dtype: int64
```

```python
year_count = df['Year'].value_counts()
print(year_count)
```

```
2019    76616
2007    76529
2018    74175
2006    73814
2004    73200
2005    73152
2008    72133
2003    66456
2010    66425
2009    66360
2013    65957
2012    62813
2011    62672
2001    61551
2015    59876
2014    59668
2000    58587
2017    58361
2016    57729
1999    56772
1997    55506
1998    55380
1996    54976
1990    54709
1995    54653
```

```
2002    54031
1988    53333
1994    53325
1993    52438
1992    52360
1989    52028
1991    52006
2020    18905
1987    13504
Name: Year, dtype: int64
```

Multivariate Exploration

This section involves analyzing the relationships between multiple variables simultaneously to uncover insights that might not be apparent when examining individual variables in isolation. In our analysis of airline flight performance, we focused on Correlation Analysis for many delay feature

Correlation between delay Features, distance and flight time

Based on the correlation matrix, we can draw the following conclusions:

1. Departure Delay and Arrival Delay: There is a very strong positive correlation between DepDelayMinutes and ArrDelayMinutes. This indicates that flights with longer departure delays tend to experience longer arrival delays as well.

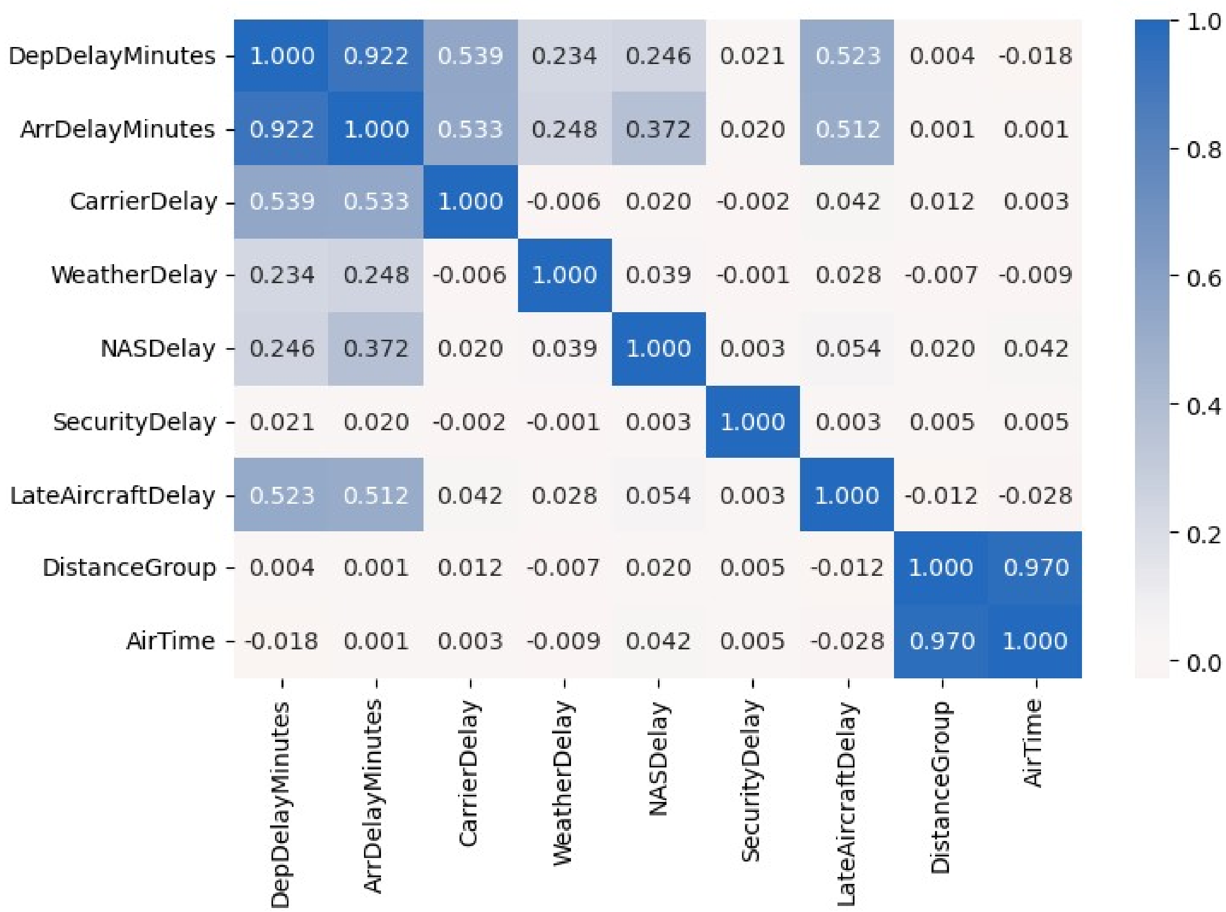
2. Arrival Delay and Late Aircraft Delay: A moderate positive correlation exists between ArrDelayMinutes and LateAircraftDelay. This suggests that significant delays caused by late aircraft are associated with increased arrival delays, though this relationship is less pronounced than with departure delays.

3. Departure Delay and Late Aircraft Delay: There is a moderate positive correlation between DepDelayMinutes and LateAircraftDelay. This implies that flights with longer departure delays are somewhat likely to experience delays due to late aircraft.

4. Distance Group and AirTime: A very strong positive correlation is observed between DistanceGroup and AirTime. This indicates that longer distances are strongly associated with longer flight durations, which aligns with the expectation that flights covering greater distances require more time in the air.

```
correlation_columns = [
'DepDelayMinutes',
'ArrDelayMinutes',
'CarrierDelay',
'WeatherDelay',
'NASDelay',
'SecurityDelay',
```

```
'LateAircraftDelay',
'DistanceGroup',
'AirTime',
]

# correlation plot

plt.figure(figsize = [8, 5])
sb.heatmap(delay_airline_df[correlation_columns].corr(), annot = True,
fmt = '.3f',
            cmap = 'vlag_r', center = 0)
plt.show()
```



Cancellation based On Cancellation Code

Examining the causes of cancellations, it is notably that in Quarter 1, cause B is the predominant reason for cancellations, with a markedly higher count than in other quarters. Cause D is exclusively observed in Quarter 1. In contrast, cause A is the primary reason for cancellations in both Quarter 2 and Quarter 3.

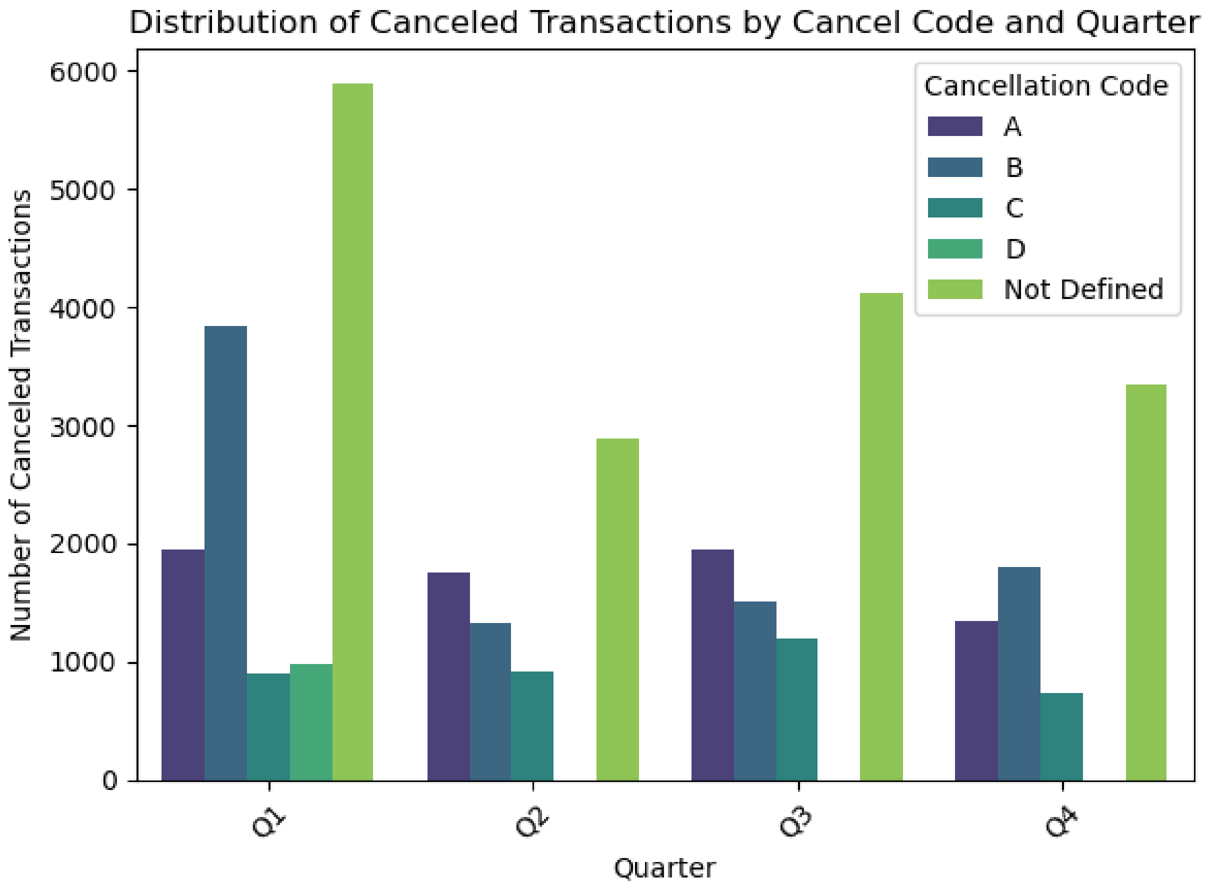
```python
# Aggregate the data to get counts for each combination of
'Quarter_Desc' and 'CancellationCode'
canceled_airline_df_agg = canceled_airline_df.groupby(['Quarter_Desc',
'CancellationCode']).size().reset_index(name='Count')
canceled_airline_df_agg.head(10)
```

	Quarter_Desc	CancellationCode	Count
0	Q1	A	1953
1	Q1	B	3842
2	Q1	C	906
3	Q1	D	988
4	Q1	Not Defined	5885
5	Q2	A	1757
6	Q2	B	1321
7	Q2	C	915
8	Q2	D	3
9	Q2	Not Defined	2892

```python
canceled_airline_df_agg.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 3 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Quarter_Desc      20 non-null     object
 1   CancellationCode  20 non-null     object
 2   Count             20 non-null     int64
dtypes: int64(1), object(2)
memory usage: 608.0+ bytes
```

```python
# Plotting
sns.barplot(data=canceled_airline_df_agg, x='Quarter_Desc', y='Count',
hue='CancellationCode', palette='viridis')

# Title and labels
plt.title('Distribution of Canceled Transactions by Cancel Code and
Quarter')
plt.xlabel('Quarter')
plt.ylabel('Number of Canceled Transactions')


plt.legend(title='Cancellation Code')
plt.xticks(rotation=45)
plt.tight_layout()
```


Distribution of Canceled Transactions by Cancel Code and Quarter
Number of Canceled Transactions
0
1000
2000
3000
4000
5000
6000
Q1
Q2
Q3
Q4
Quarter
Cancellation Code
A
B
C
D
Not Defined