In [34]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.stattools import adfuller
from scipy.stats import norm

# Read the flights data
df= pd.read_csv("C:\DOWNLOADSSS\DA project\FlightDelay.csv",low_memory=False)

df['YEAR'].replace({2015:2023,},inplace=True)
df['AIRLINE'].replace({
    'AS':'Air Asia',
    'AA':'Air India',
    'US':'Endeavor Air',
    'DL':'Air Bus',
    'NK':'Go Air',
    'UA':'Jet Airways',
    'HA':'SpiceJet',
    'B6':'Emirates',
    'OO':'Sahara',
    'EV':'Boeing',
    'MQ':'Deccan Airlines',
    'F9':'Indigo',
    'WN':'Virgin America',
    'VX':'Vistara',
},inplace=True)

#function that assigns season based on month. May be useful for visualization
def season_cat(x):
    if x in [12,1,2]:
        return 'winter'
    elif x in [3,4,5]:
        return 'spring'
    elif x in [6,7,8]:
        return 'summer'
    return 'autumn'
df.head()
```

Out[34]:

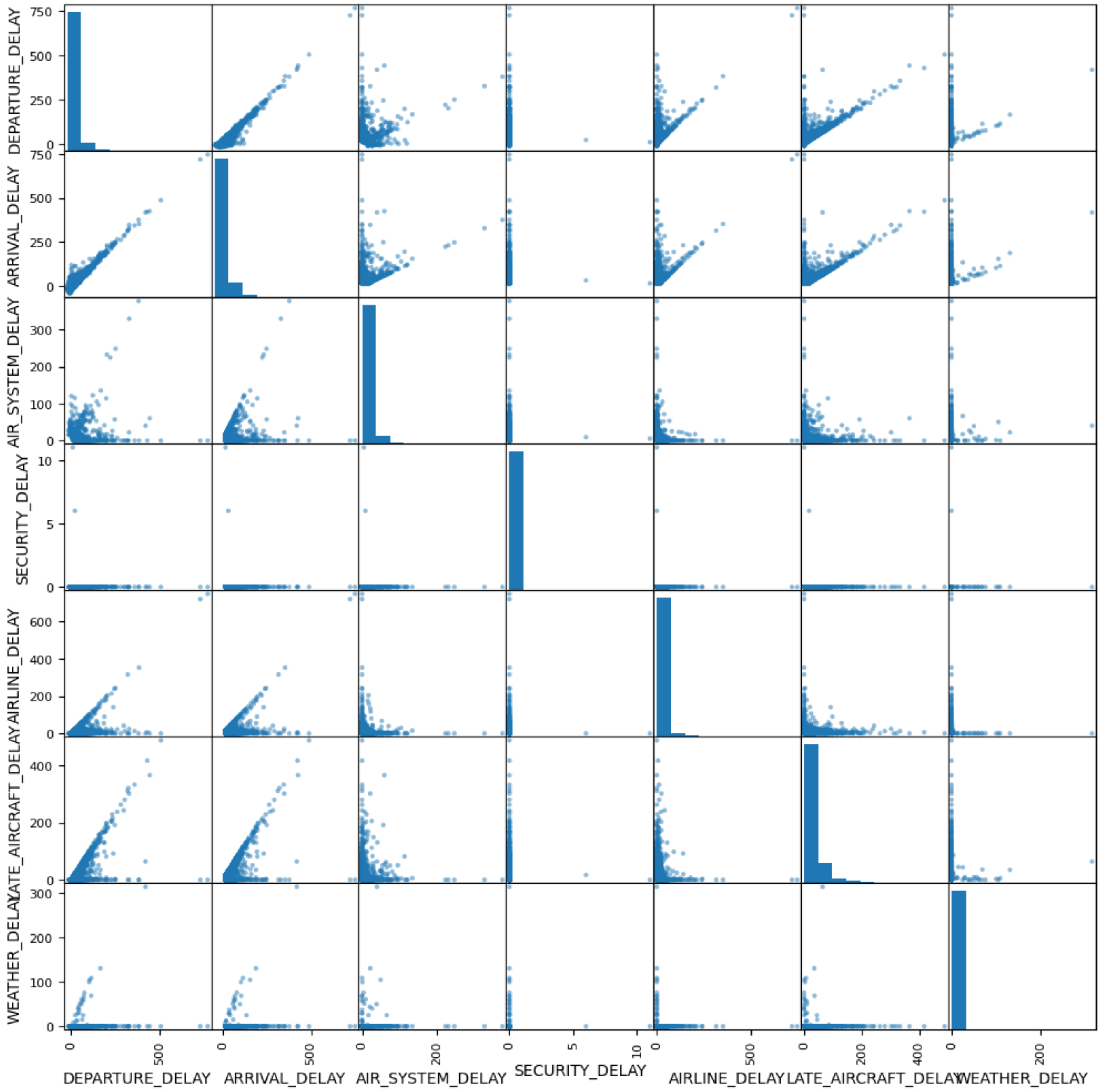
	YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTIN
0	2023	1	1	4	Air Asia	98	N407AS	ANC	
1	2023	1	1	4	Air India	2336	N3KUAA	LAX	
2	2023	1	1	4	Endeavor Air	840	N171US	SFO	
3	2023	1	1	4	Air India	258	N3HYAA	LAX	
4	2023	1	1	4	Air Asia	135	N527AS	SEA	

In [35]:

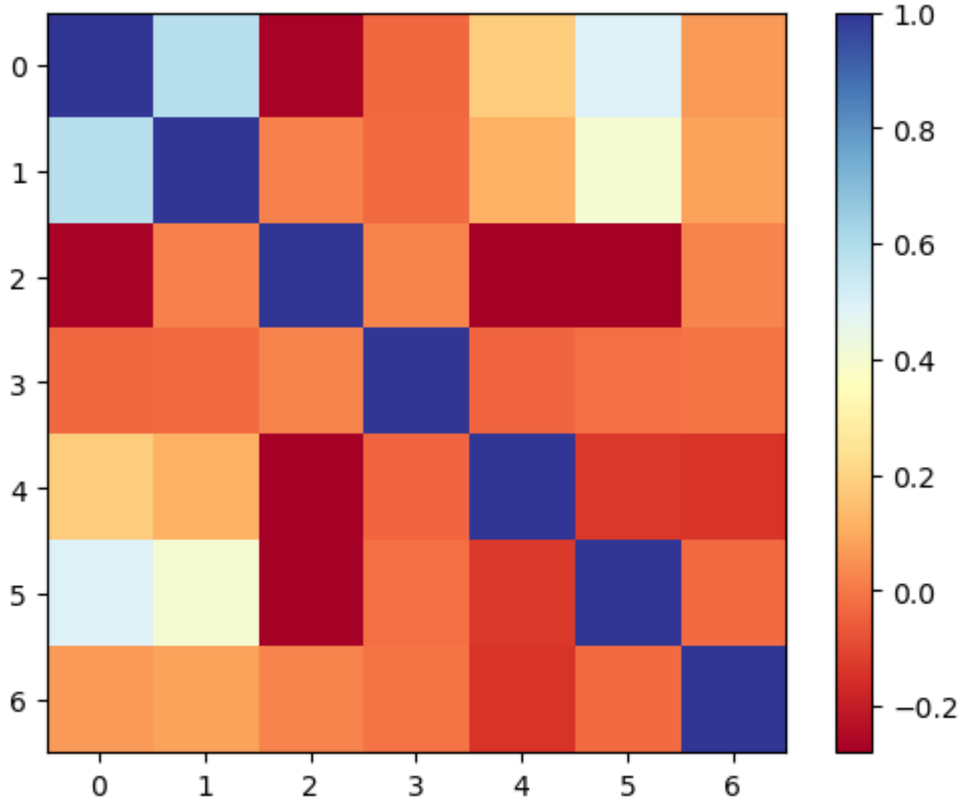
```
# Remove missing values
df.dropna(subset=['ORIGIN_AIRPORT', 'DESTINATION_AIRPORT'], inplace=True)

# Ask user for origin and destination airports
origin_airport = input("Enter origin airport code (e.g., LAX): ")
destination_airport = input("Enter destination airport code (e.g., SEA): ")
# Filter data for selected origin and destination airports
flights_selected = flights_data[(flights_data['ORIGIN_AIRPORT'] == origin_airport) &
                                 (flights_data['DESTINATION_AIRPORT'] == destination_airport)
# Feature engineering
num_cols = ['DEPARTURE_DELAY', 'ARRIVAL_DELAY', 'AIR_SYSTEM_DELAY', 'SECURITY_DELAY', 'AIRLIN
            'LATE_AIRCRAFT_DELAY', 'WEATHER_DELAY']

# Print summary statistics
print(flights_selected[num_cols].describe())

df.info()

# Correlation matrix
print(flights_selected[num_cols].corr())
# Covariance matrix
print(flights_selected[num_cols].cov())
# Scatterplot matrix
pd.plotting.scatter_matrix(flights_selected[num_cols], figsize=(12, 12))
plt.show()

R = flights_selected[num_cols].corr(method='kendall')
print(R)
plt.imshow(R, cmap='RdYlBu', interpolation='nearest')
plt.colorbar()
plt.show()

# Select relevant columns for time series analysis
flights_delay_sub_new = df[['YEAR', 'MONTH', 'DAY', 'DAY_OF_WEEK', 'AIRLINE', 'ORIGIN_AIRPORT
                                      'DESTINATION_AIRPORT', 'DEPARTURE_DELAY', 'ARRIVAL_DELA
# Remove missing values
flights_delay_sub_new = flights_delay_sub_new.dropna()

# Create a date column
flights_selected['DATE'] = pd.to_datetime(flights_selected[['YEAR', 'MONTH', 'DAY']])

# Density plot of arrival delay
plt.figure(figsize=(12, 8))
sns.kdeplot(data=flights_selected, x='ARRIVAL_DELAY')
plt.xlabel('Arrival Delay in minutes')
plt.ylabel('Density of Arrival Delay')
plt.title('Density chart of arrival delay by Airline for 2023 flights from - {} to {}'.
          format(origin_airport,destination_airport))
plt.show()

# Boxplot of arrival delay by airline
plt.figure(figsize=(12, 8))
sns.boxplot(data=flights_selected, x='AIRLINE', y='ARRIVAL_DELAY')
plt.xlabel('Airline')
plt.ylabel('Arrival Delay in minutes')
plt.title('Density chart of arrival delay by Airline for 2023 flights from - {} to {}'.
          format(origin_airport,destination_airport))
plt.show()

# Violin plot of arrival delay by airline
plt.figure(figsize=(12, 8))
sns.violinplot(data=flights_selected, x='AIRLINE', y='ARRIVAL_DELAY')
plt.xlabel('Airline ')
plt.ylabel('Arrival Delay in minutes')
plt.title('Density chart of arrival delay by Airline for 2023 flights from - {} to {}'.
          format(origin_airport,destination_airport))
plt.show()

# Scatter plot of departure delay vs arrival delay with airline
```

```python
# Scatter plot of departure delay vs arrival delay with airline
plt.figure(figsize=(12, 8))
sns.scatterplot(data=flights_selected, x='DEPARTURE_DELAY', y='ARRIVAL_DELAY', hue='AIRLINE',
                size='ARRIVAL_DELAY', alpha=0.5)
plt.xlabel('Departure delay in minutes')
plt.ylabel('Arrival delay in minutes')
plt.title('Relationship between Departure delay and Arrival delay with Airline shown')
plt.show()

# Scatter plot of day of week vs arrival delay with airline
plt.figure(figsize=(12, 8))
sns.scatterplot(data=flights_selected, x='DAY_OF_WEEK', y='ARRIVAL_DELAY', hue='AIRLINE',
                style='AIRLINE', size='ARRIVAL_DELAY', alpha=0.5)
plt.xlabel('Day of week in numbers (MON=1...SUN=7)')
plt.ylabel('Arrival delay in minutes')
plt.title('Relationship between Day of week and Arrival delay with Airline shown')
plt.show()

# Distribution and ACF of arrival delay
arrival_delay_ts = flights_selected.set_index('DATE')['ARRIVAL_DELAY']
plt.figure(figsize=(12, 4))
plt.hist(arrival_delay_ts, bins=40, density=True, color='skyblue', alpha=0.7)
plt.title('Distribution of Arrival Delay')
plt.xlabel('Arrival Delay')
plt.ylabel('Density')
plt.show()
#plot_acf(arrival_delay_ts, lags=40)
#plt.show()


# Create a bar chart using seaborn
airline_counts = df['AIRLINE'].value_counts(14).sort_values(ascending=False)
plt.figure(figsize=(6, 4))
sns.color_palette("colorblind")
plt.title("Count of Flights by Airline")
sns.barplot(x=airline_counts.values, y=airline_counts.index,palette="colorblind")
plt.xlabel("Count")
plt.ylabel("Airline")
plt.show()

# Boxplot of arrival delay by airline for individual months
plt.figure(figsize=(12, 8))
sns.boxplot(data=flights_selected, x='MONTH', y='ARRIVAL_DELAY', hue='AIRLINE')
plt.xlabel('Month')
plt.ylabel('Arrival Delay in minutes')
plt.title('Density chart of arrival delay by Airline for 2023 flights from - {} to {}'.
          format(origin_airport,destination_airport))
plt.legend(title='Airline', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()

# Histogram of arrival delay by airline for individual months
plt.figure(figsize=(12, 8))
sns.histplot(data=flights_selected, x='ARRIVAL_DELAY', hue='MONTH', multiple='stack')
plt.xlabel('Arrival Delay in minutes')
plt.ylabel('Frequency')
plt.title('Histogram of Arrival delay by Month')
plt.legend(title='Month', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()

# Strip plot of arrival delay by airline for individual months
plt.figure(figsize=(12, 8))
sns.stripplot(data=flights_selected, x='MONTH', y='ARRIVAL_DELAY', hue='AIRLINE', jitter=True
plt.xlabel('Month')
plt.ylabel('Arrival Delay in minutes')
plt.title('Density chart of arrival delay by Airline for 2023 flights from - {} to {}'.
          format(origin_airport,destination_airport))
plt.legend(title='Airline', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
plt.figure(figsize=(25, 12)).subplots_adjust(hspace = 0.5)
plt.subplot(2, 2 ,1)
df.groupby('MONTH').ARRIVAL_DELAY.sum().plot.bar().set_title('ARRIVAL delays by month')
plt.title('ARRIVAL delays by month', fontsize=16)
plt.ylabel('Hours', fontsize=14)
plt.xlabel('Month of the year', fontsize=14)
plt.subplot(2, 2 ,2)
df.groupby('MONTH').DEPARTURE_DELAY.sum().plot.bar()
plt.title('DEPARTURE delays by month', fontsize=16)
plt.ylabel('Hours', fontsize=14)
plt.xlabel('Month of the year', fontsize=14)
plt.show()
```

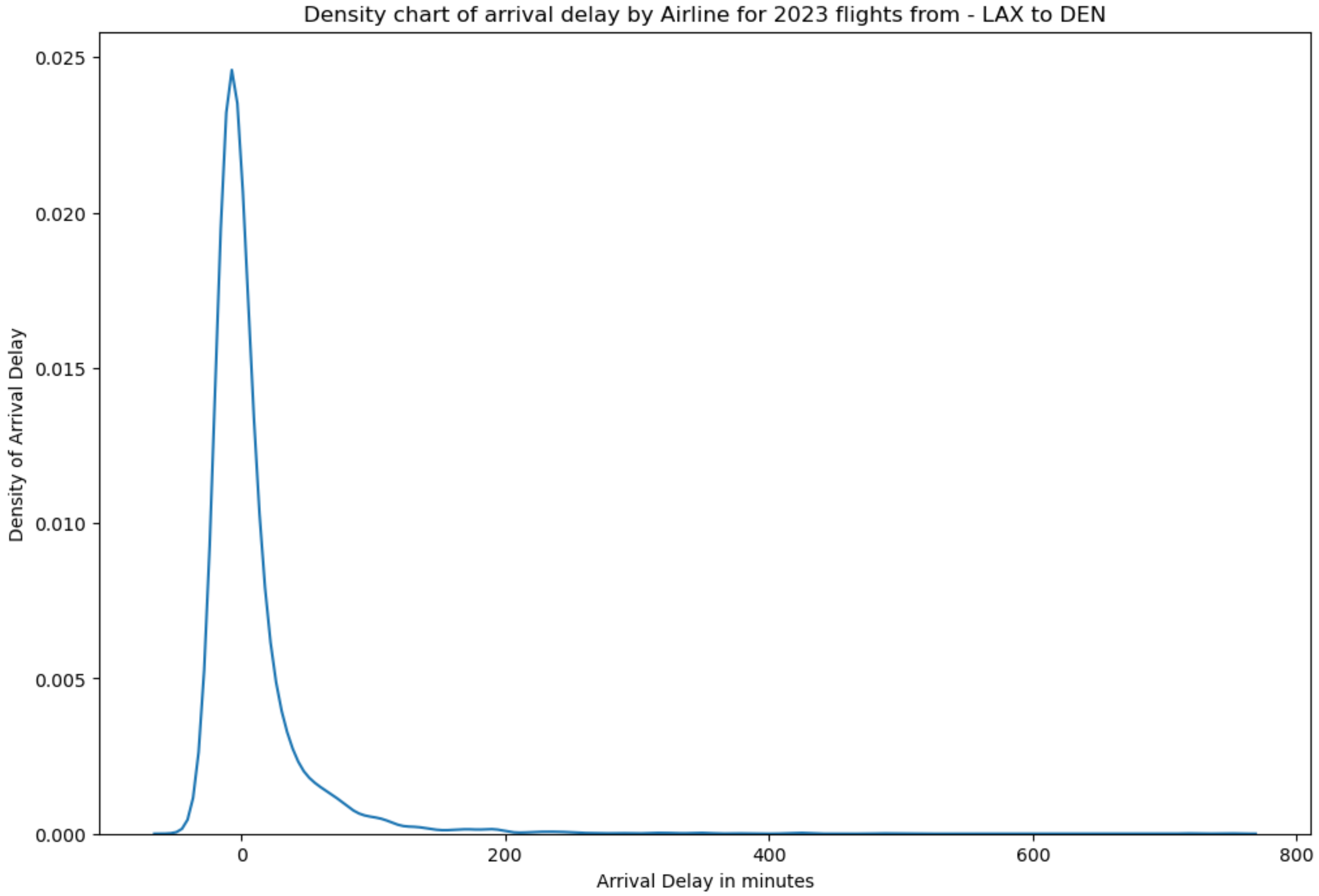
```
Enter origin airport code (e.g., LAX): LAX
Enter destination airport code (e.g., SEA): DEN
       DEPARTURE_DELAY  ARRIVAL_DELAY  AIR_SYSTEM_DELAY  SECURITY_DELAY  \
count      5707.000000    5680.000000       1265.000000     1265.000000   
mean         13.041178       7.369718         10.817391        0.013439   
std          38.141139      39.291111         25.035494        0.352176   
min         -20.000000     -45.000000          0.000000        0.000000   
25%          -3.000000     -11.000000          0.000000        0.000000   
50%           1.000000      -2.000000          0.000000        0.000000   
75%          14.000000      11.000000         14.000000        0.000000   
max         767.000000     748.000000        377.000000       11.000000   

       AIRLINE_DELAY  LATE_AIRCRAFT_DELAY  WEATHER_DELAY  
count    1265.000000          1265.000000    1265.000000  
mean       15.750988            27.672727       1.213439  
std        40.876537            46.098774      12.152264  
min         0.000000             0.000000       0.000000  
25%         0.000000             0.000000       0.000000  
50%         6.000000            13.000000       0.000000  
75%        17.000000            34.000000       0.000000  
max       748.000000           488.000000     315.000000  
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5819079 entries, 0 to 5819078
Data columns (total 31 columns):
 #   Column               Dtype  
---  ------               -----  
 0   YEAR                 int64  
 1   MONTH                int64  
 2   DAY                  int64  
 3   DAY_OF_WEEK          int64  
 4   AIRLINE              object 
 5   FLIGHT_NUMBER        int64  
 6   TAIL_NUMBER          object 
 7   ORIGIN_AIRPORT       object 
 8   DESTINATION_AIRPORT  object 
 9   SCHEDULED_DEPARTURE  int64  
 10  DEPARTURE_TIME       float64
 11  DEPARTURE_DELAY      float64
 12  TAXI_OUT             float64
 13  WHEELS_OFF           float64
 14  SCHEDULED_TIME       float64
 15  ELAPSED_TIME         float64
 16  AIR_TIME             float64
 17  DISTANCE             int64  
 18  WHEELS_ON            float64
 19  TAXI_IN              float64
 20  SCHEDULED_ARRIVAL    int64  
 21  ARRIVAL_TIME         float64
 22  ARRIVAL_DELAY        float64
 23  DIVERTED             int64  
 24  CANCELLED            int64  
 25  CANCELLATION_REASON  object 
 26  AIR_SYSTEM_DELAY     float64
 27  SECURITY_DELAY       float64
 28  AIRLINE_DELAY        float64
 29  LATE_AIRCRAFT_DELAY  float64
 30  WEATHER_DELAY        float64
dtypes: float64(16), int64(10), object(5)
memory usage: 1.3+ GB
                     DEPARTURE_DELAY  ARRIVAL_DELAY  AIR_SYSTEM_DELAY  \
DEPARTURE_DELAY             1.000000       0.960389          0.140549
ARRIVAL_DELAY               0.960389       1.000000          0.251219
AIR_SYSTEM_DELAY            0.140549       0.251219          1.000000
SECURITY_DELAY             -0.024322      -0.021503         -0.006720
AIRLINE_DELAY               0.548404       0.543689         -0.109653
LATE_AIRCRAFT_DELAY         0.658706       0.637870         -0.126413
WEATHER_DELAY               0.157232       0.177358          0.030147

                     SECURITY_DELAY  AIRLINE_DELAY  LATE_AIRCRAFT_DELAY  \
DEPARTURE_DELAY           -0.024322       0.548404             0.658706
ARRIVAL_DELAY             -0.021503       0.543689             0.637870
AIR_SYSTEM_DELAY          -0.006720      -0.109653            -0.126413
SECURITY_DELAY             1.000000      -0.014716            -0.017954
AIRLINE_DELAY             -0.014716       1.000000            -0.108480
LATE_AIRCRAFT_DELAY       -0.017954      -0.108480             1.000000
WEATHER_DELAY             -0.003813      -0.038507            -0.014716

                     WEATHER_DELAY
DEPARTURE_DELAY           0.157232
ARRIVAL_DELAY             0.177358
AIR_SYSTEM_DELAY          0.030147
SECURITY_DELAY           -0.003813
AIRLINE_DELAY            -0.038507
LATE_AIRCRAFT_DELAY      -0.014716
WEATHER_DELAY             1.000000
                     DEPARTURE_DELAY  ARRIVAL_DELAY  AIR_SYSTEM_DELAY  \
DEPARTURE_DELAY          1454.746464    1427.553031        220.158365
ARRIVAL_DELAY            1427.553031    1543.791441        377.780146
AIR_SYSTEM_DELAY          220.158365     377.780146        626.775963
SECURITY_DELAY             -0.535926      -0.454870         -0.059253
AIRLINE_DELAY            1402.577647    1334.921214       -112.214812
LATE_AIRCRAFT_DELAY      1899.908516    1766.252172       -145.893671
WEATHER_DELAY             119.550308     129.460953          9.171918

                     SECURITY_DELAY  AIRLINE_DELAY  LATE_AIRCRAFT_DELAY  \
DEPARTURE_DELAY           -0.535926    1402.577647          1899.908516
ARRIVAL_DELAY             -0.454870    1334.921214          1766.252172
AIR_SYSTEM_DELAY          -0.059253    -112.214812          -145.893671
SECURITY_DELAY             0.124028      -0.211841            -0.291484
AIRLINE_DELAY             -0.211841    1670.891267          -204.415420
LATE_AIRCRAFT_DELAY       -0.291484    -204.415420          2125.096922
WEATHER_DELAY             -0.016320     -19.127980            -8.244174

                     WEATHER_DELAY
DEPARTURE_DELAY         119.550308
ARRIVAL_DELAY           129.460953
AIR_SYSTEM_DELAY          9.171918
SECURITY_DELAY           -0.016320
AIRLINE_DELAY           -19.127980
LATE_AIRCRAFT_DELAY      -8.244174
WEATHER_DELAY           147.677509
```

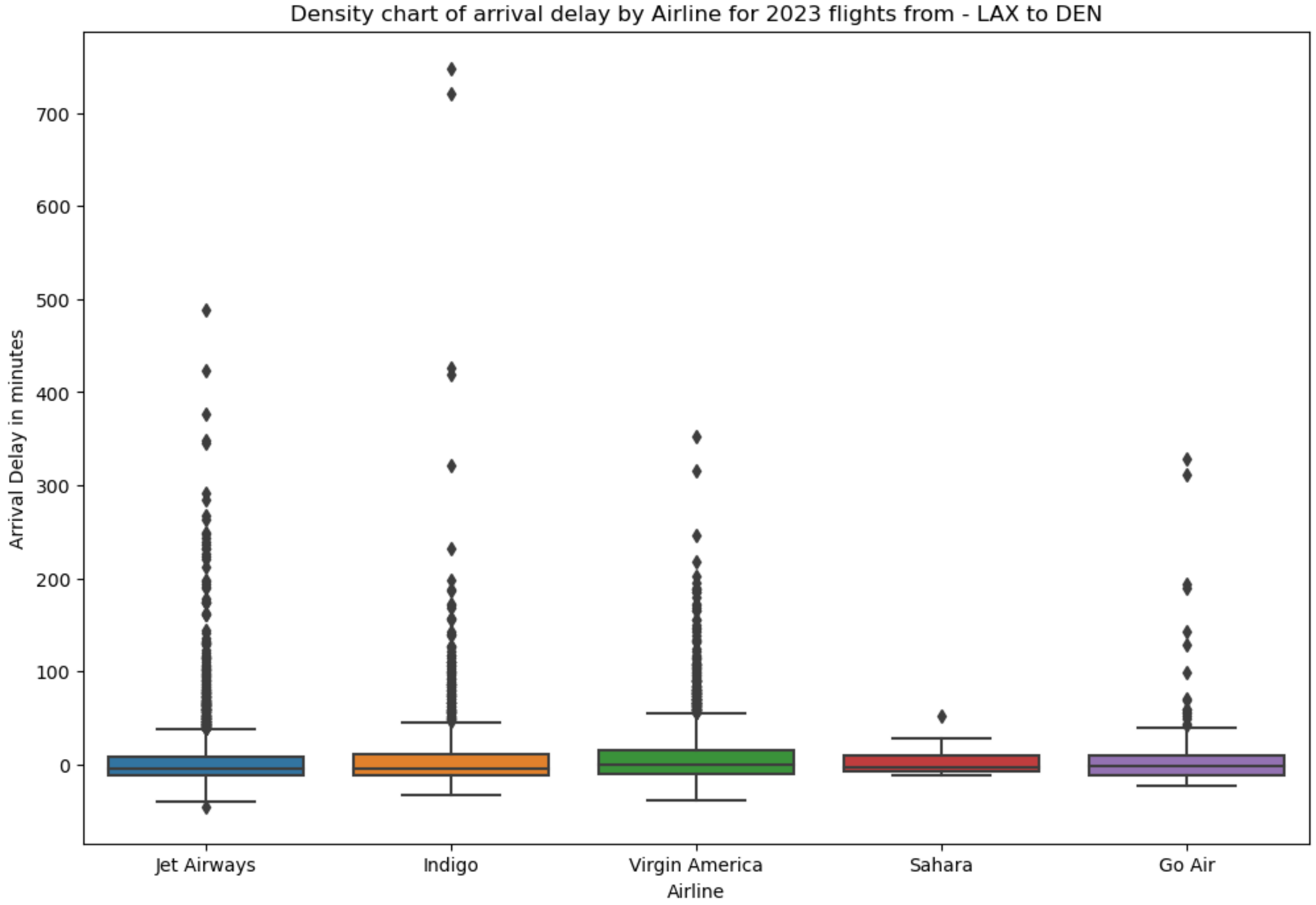


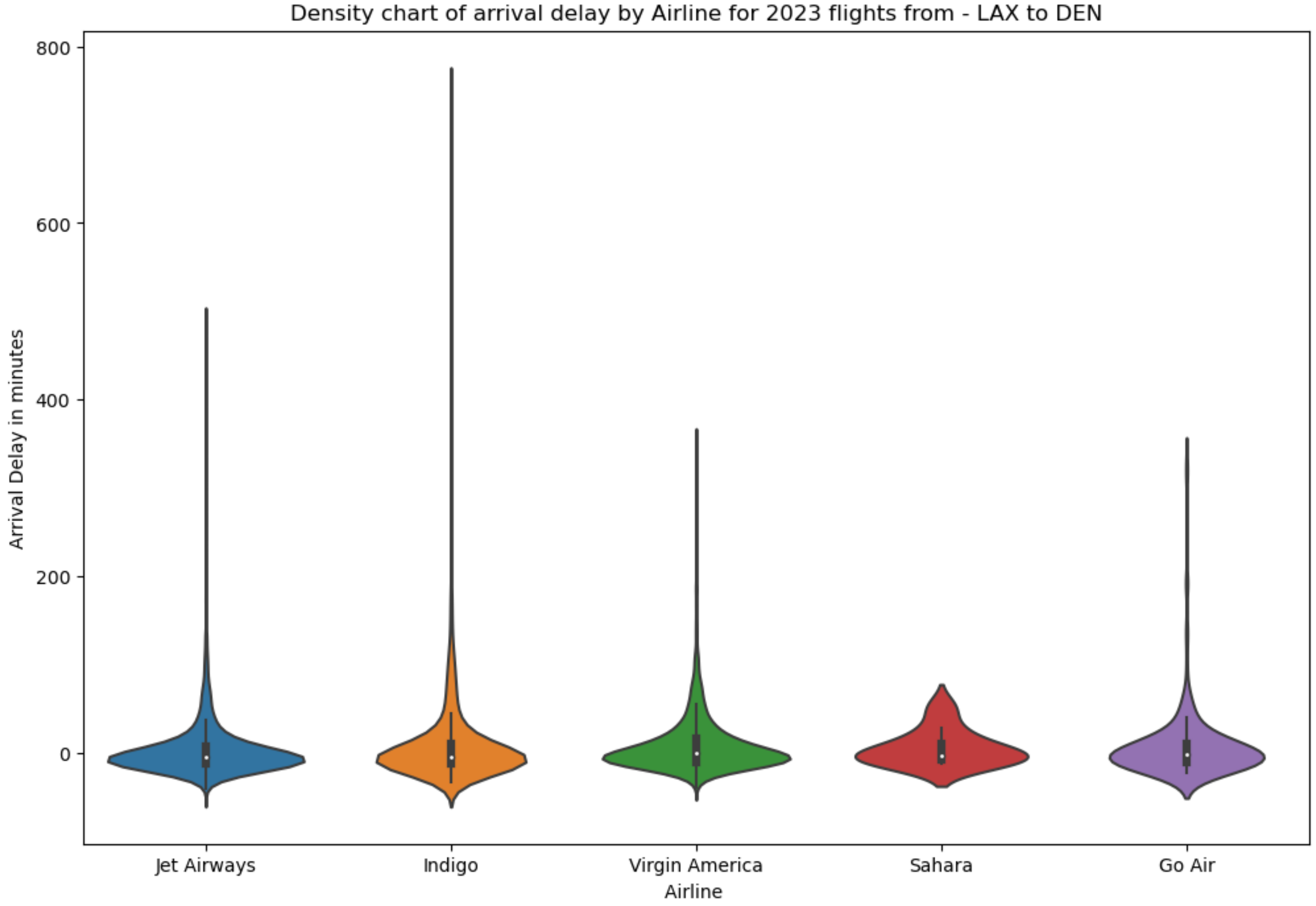
```
                     DEPARTURE_DELAY  ARRIVAL_DELAY  AIR_SYSTEM_DELAY  \
DEPARTURE_DELAY             1.000000       0.582413         -0.267057
ARRIVAL_DELAY               0.582413       1.000000          0.019359
AIR_SYSTEM_DELAY           -0.267057       0.019359          1.000000
SECURITY_DELAY             -0.034792      -0.028102          0.020944
AIRLINE_DELAY               0.183422       0.117696         -0.280298
LATE_AIRCRAFT_DELAY         0.491324       0.400168         -0.280164
WEATHER_DELAY               0.068028       0.081336          0.024554

                     SECURITY_DELAY  AIRLINE_DELAY  LATE_AIRCRAFT_DELAY  \
DEPARTURE_DELAY           -0.034792       0.183422             0.491324
ARRIVAL_DELAY             -0.028102       0.117696             0.400168
AIR_SYSTEM_DELAY           0.020944      -0.280298            -0.280164
SECURITY_DELAY             1.000000      -0.038768            -0.015409
AIRLINE_DELAY             -0.038768       1.000000            -0.133429
LATE_AIRCRAFT_DELAY       -0.015409      -0.133429             1.000000
WEATHER_DELAY             -0.005735      -0.140452            -0.029472

                     WEATHER_DELAY
DEPARTURE_DELAY           0.068028
ARRIVAL_DELAY             0.081336
AIR_SYSTEM_DELAY          0.024554
SECURITY_DELAY           -0.005735
AIRLINE_DELAY            -0.140452
LATE_AIRCRAFT_DELAY      -0.029472
WEATHER_DELAY             1.000000
```

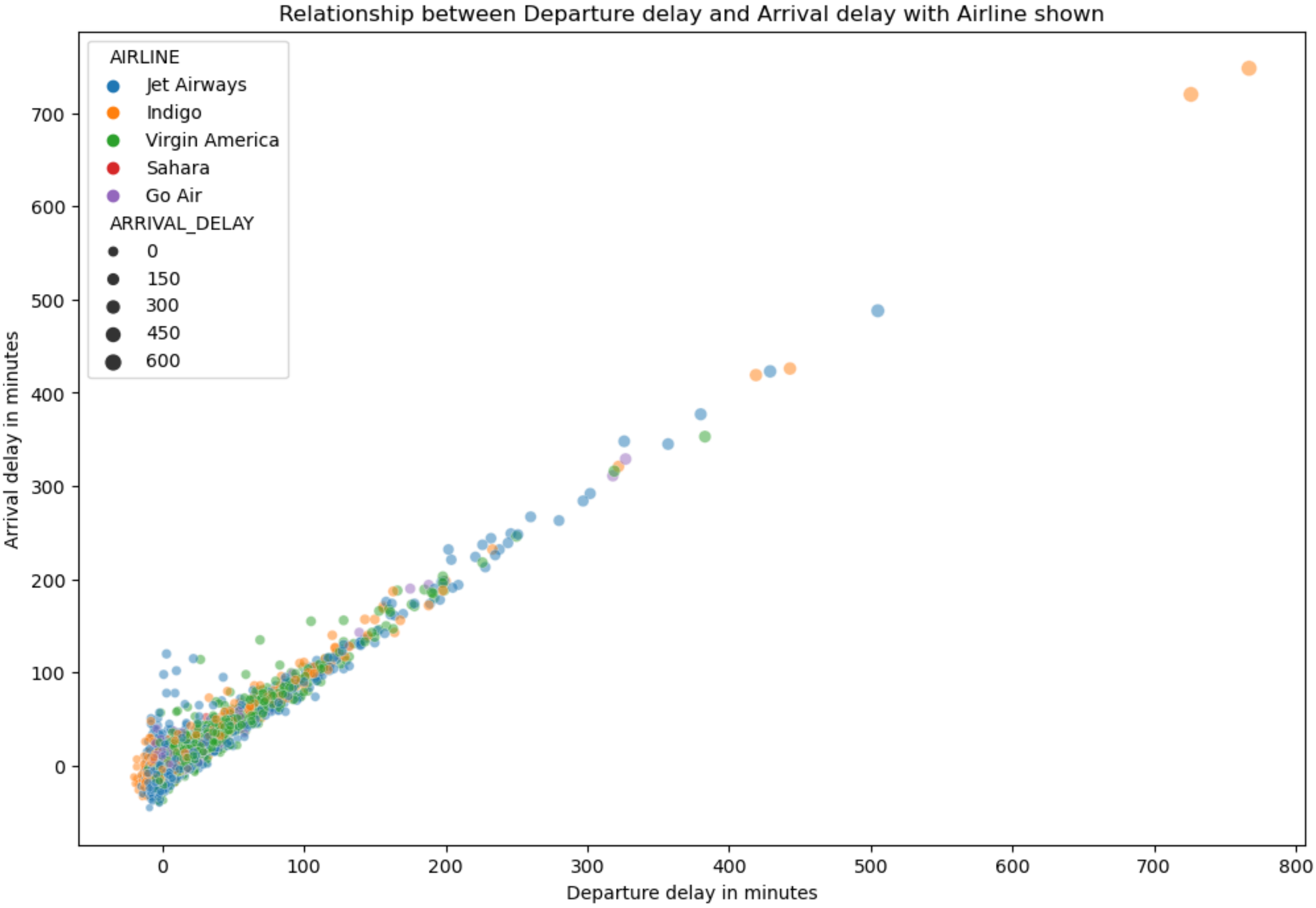


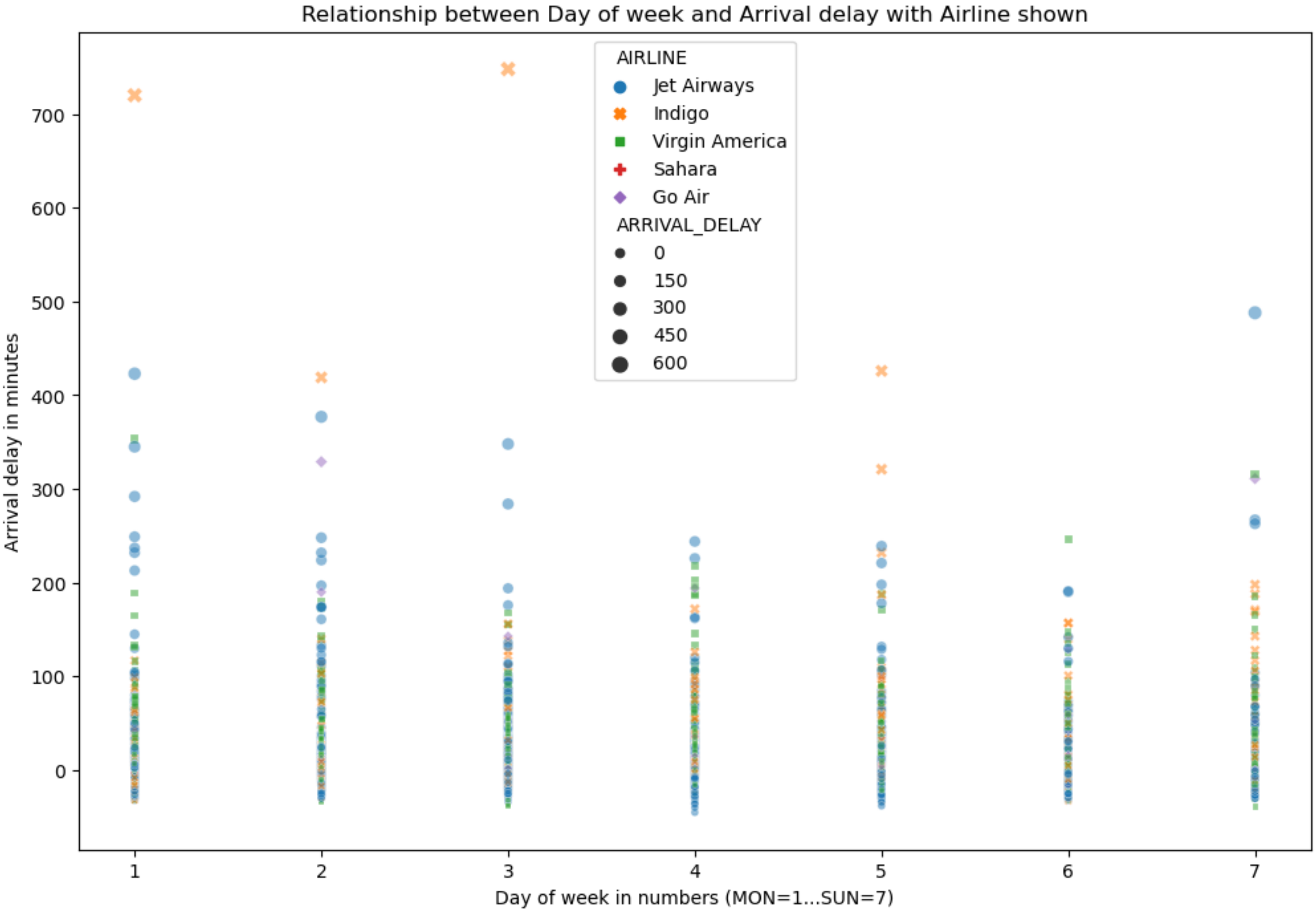
```
C:\Users\Shashank\AppData\Local\Temp\ipykernel_13872\1441118347.py:40: SettingWithCopyWarning
:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guid
e/indexing.html#returning-a-view-versus-a-copy
  flights_selected['DATE'] = pd.to_datetime(flights_selected[['YEAR', 'MONTH', 'DAY']])
```

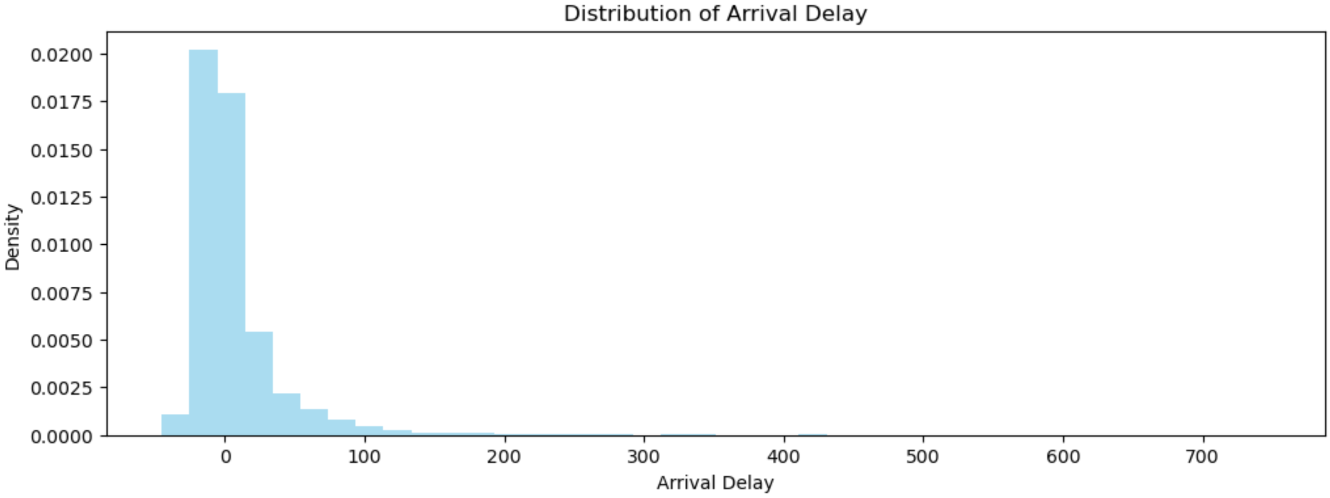


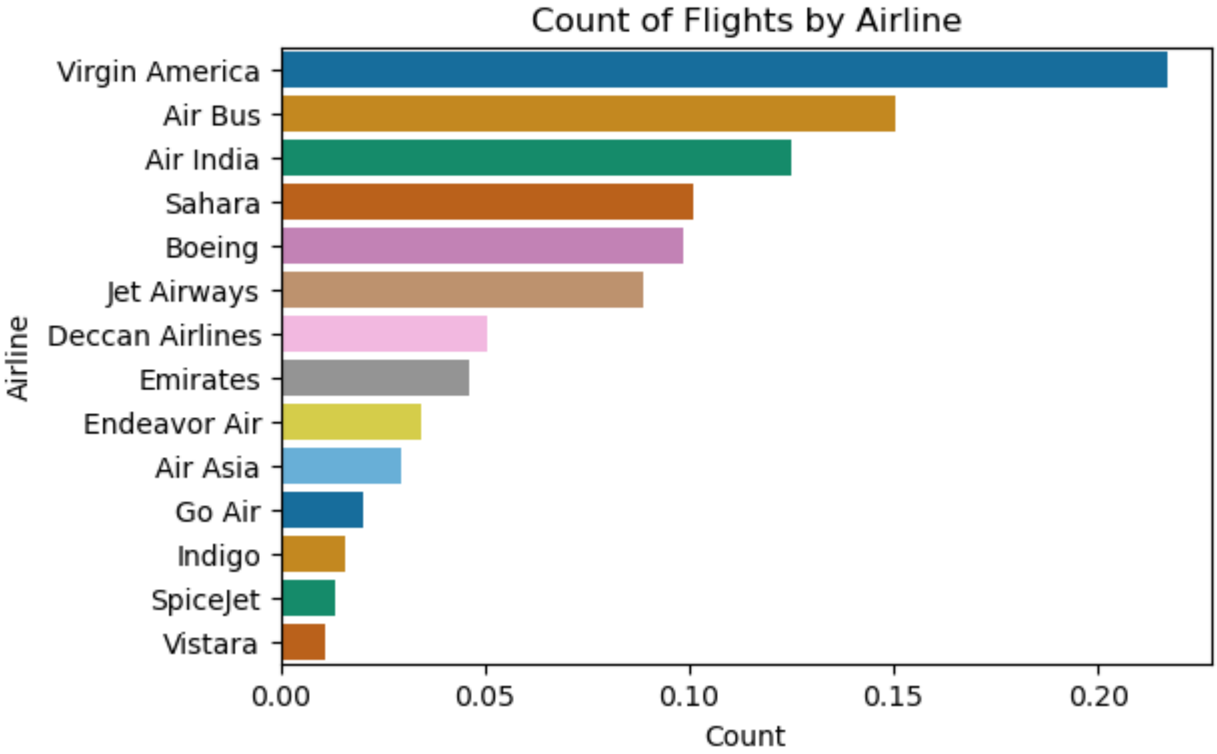

Density chart of arrival delay by Airline for 2023 flights from - LAX to DEN
Arrival Delay in minutes
0
100
200
300
400
500
600
700
Jet Airways
Indigo
Virgin America
Sahara
Go Air
Airline

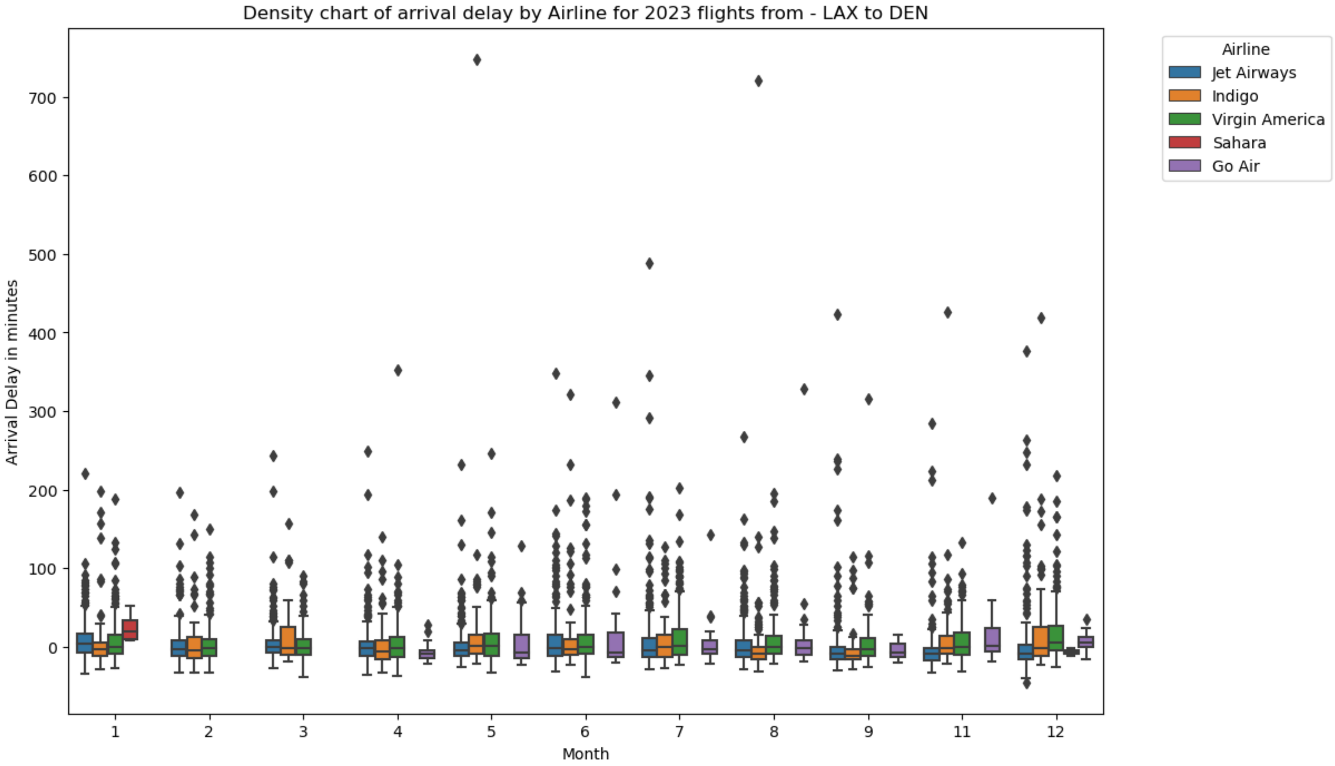

Density chart of arrival delay by Airline for 2023 flights from - LAX to DEN
Arrival Delay in minutes
0
200
400
600
800
Jet Airways
Indigo
Virgin America
Sahara
Go Air
Airline


Relationship between Departure delay and Arrival delay with Airline shown
AIRLINE
Jet Airways
Indigo
Virgin America
Sahara
Go Air
ARRIVAL_DELAY
0
150
300
450
600
Arrival delay in minutes
700
600
500
400
300
200
100
0
Departure delay in minutes
0
100
200
300
400
500
600
700
800

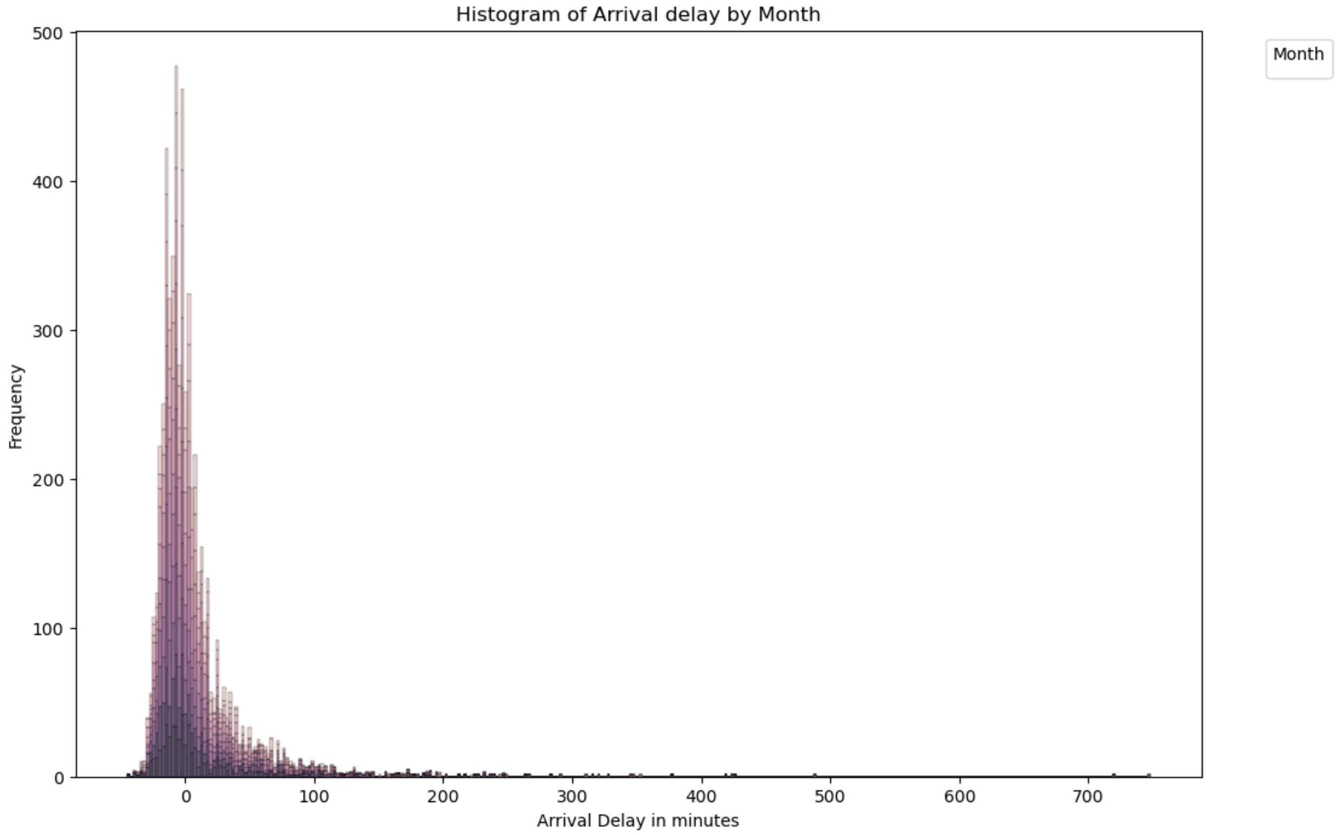

Relationship between Day of week and Arrival delay with Airline shown
AIRLINE
Jet Airways
Indigo
Virgin America
Sahara
Go Air
ARRIVAL_DELAY
0
150
300
450
600
Arrival delay in minutes
700
600
500
400
300
200
100
0
Day of week in numbers (MON=1...SUN=7)
1
2
3
4
5
6
7

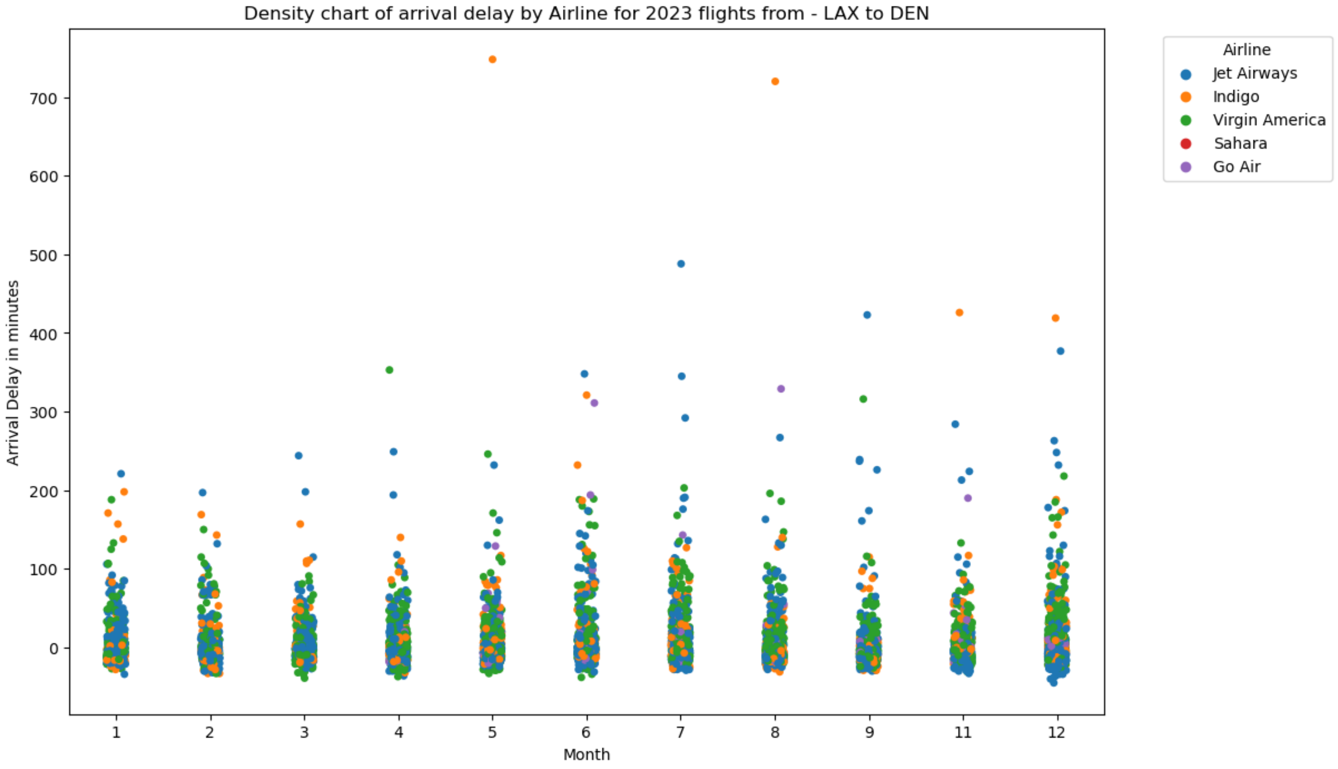


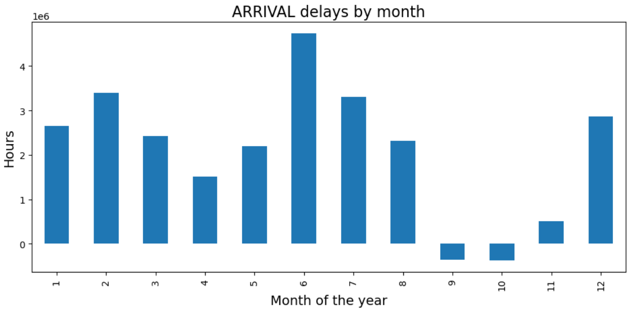


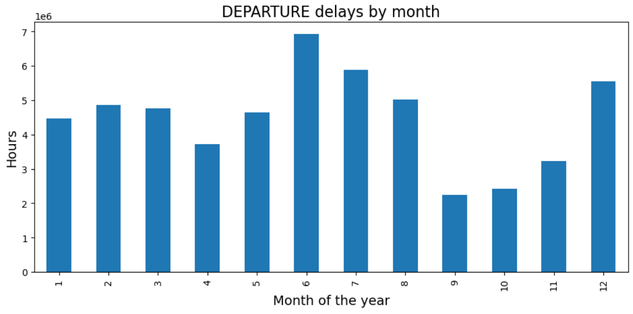


```
No artists with labels found to put in legend.  Note that artists whose label start with an u
nderscore are ignored when legend() is called with no argument.
```


Histogram of Arrival delay by Month
Frequency
0
100
200
300
400
500
Arrival Delay in minutes
0
100
200
300
400
500
600
700
Month

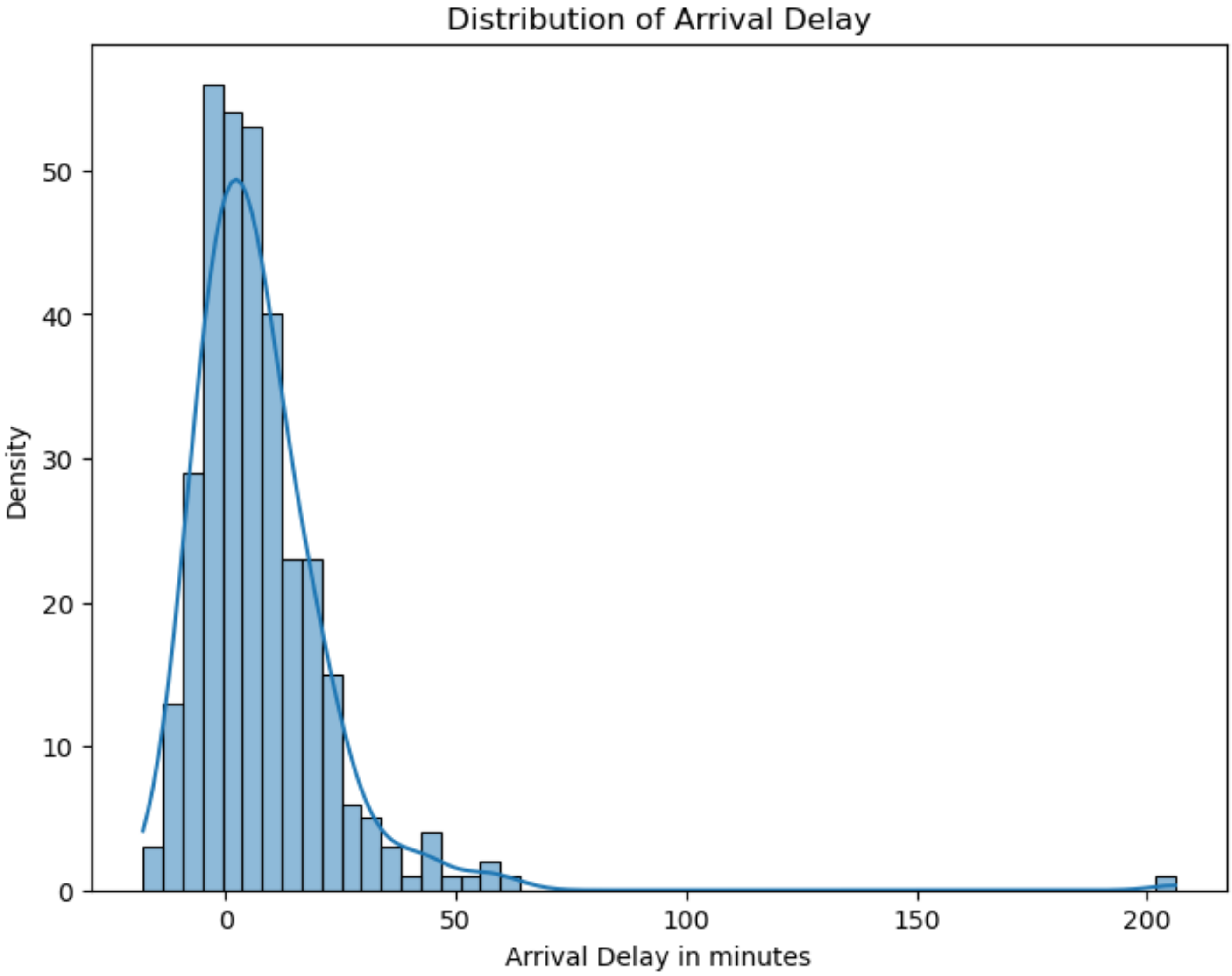

Density chart of arrival delay by Airline for 2023 flights from - LAX to DEN
Arrival Delay in minutes
0
100
200
300
400
500
600
700
Month
1
2
3
4
5
6
7
8
9
11
12
Airline
Jet Airways
Indigo
Virgin America
Sahara
Go Air

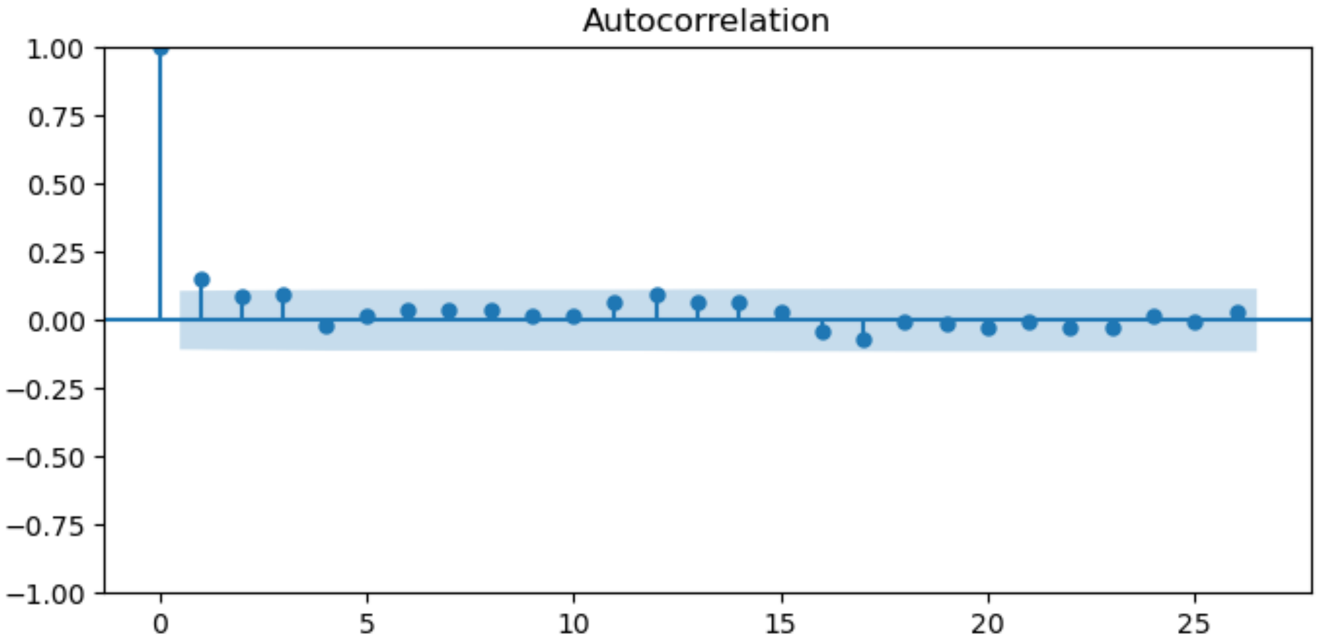

ARRIVAL delays by month
1e6
Hours
0
1
2
3
4
Month of the year
1
2
3
4
5
6
7
8
9
10
11
12

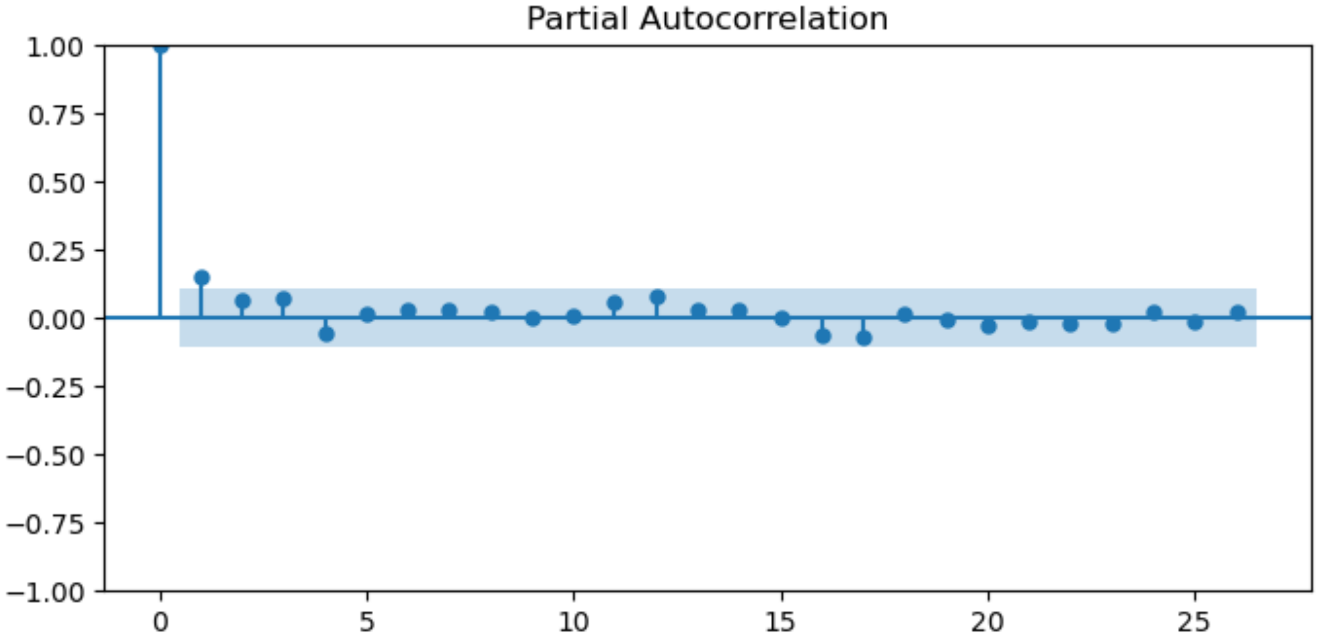

DEPARTURE delays by month
1e6
Hours
0
1
2
3
4
5
6
7
Month of the year
1
2
3
4
5
6
7
8
9
10
11
12

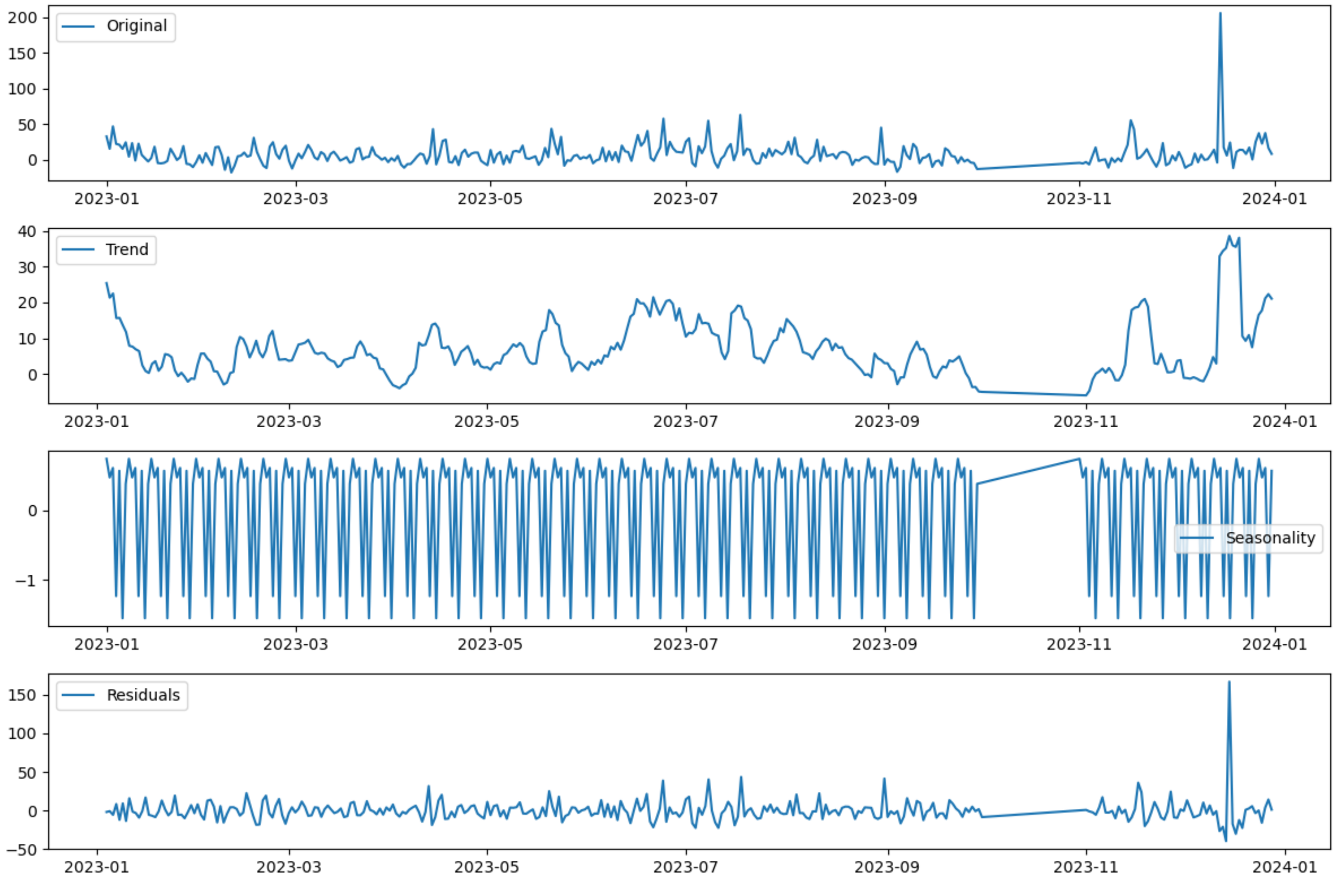
In [36]:

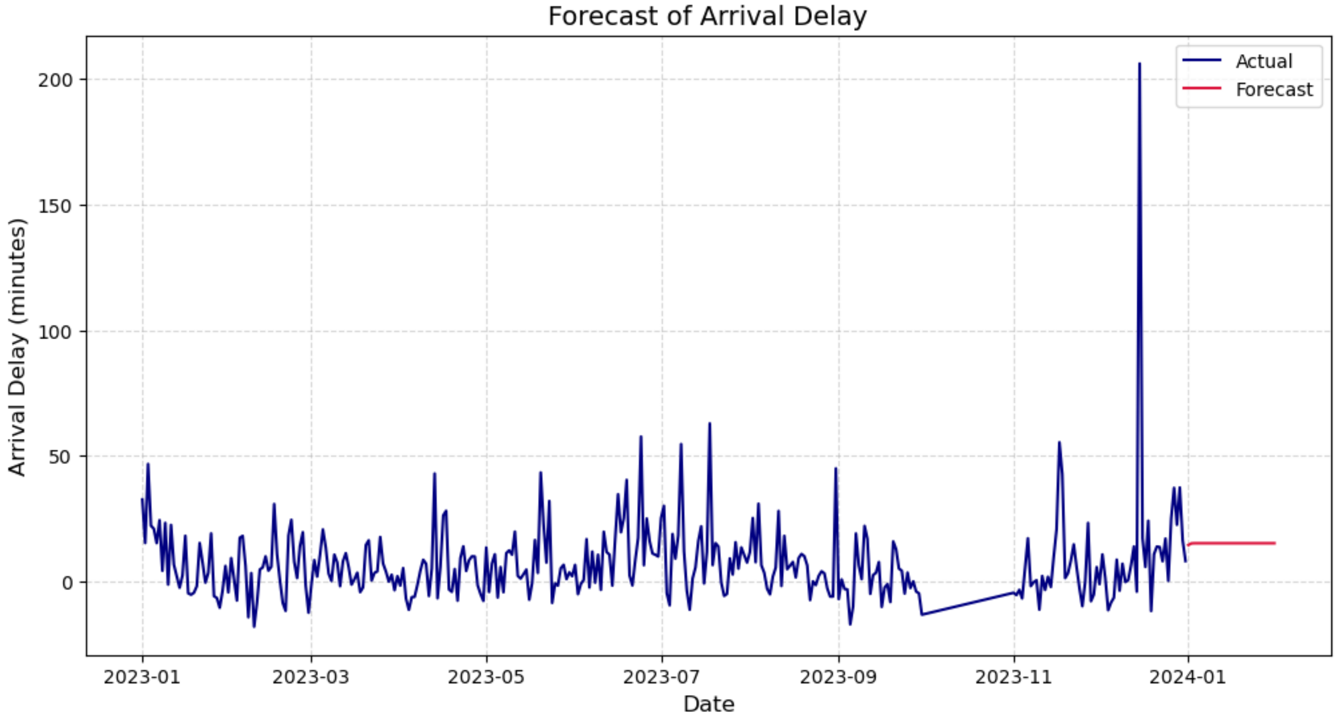
```
# Calculate average arrival delay per day
flights_arrival_delay_daily_avg = flights_selected.groupby('DATE')['ARRIVAL_DELAY'].mean().re

# Visualize distribution of arrival delay
plt.figure(figsize=(8, 6))
sns.histplot(flights_arrival_delay_daily_avg['ARRIVAL_DELAY'], kde=True)
plt.xlabel('Arrival Delay in minutes')
plt.ylabel('Density')
plt.title('Distribution of Arrival Delay')
plt.show()

# Plot ACF and PACF
fig, ax = plt.subplots(2, 1, figsize=(8, 8))
plot_acf(flights_arrival_delay_daily_avg['ARRIVAL_DELAY'], ax=ax[0])
plot_pacf(flights_arrival_delay_daily_avg['ARRIVAL_DELAY'], ax=ax[1])
plt.show()

# Time series decomposition
decomposition = seasonal_decompose(flights_arrival_delay_daily_avg['ARRIVAL_DELAY'], model='a
trend = decomposition.trend
seasonal = decomposition.seasonal
residual = decomposition.resid

# Plot decomposition
plt.figure(figsize=(12, 8))
plt.subplot(411)
plt.plot(flights_arrival_delay_daily_avg['DATE'], flights_arrival_delay_daily_avg['ARRIVAL_DE
plt.legend(loc='best')
plt.subplot(412)
plt.plot(flights_arrival_delay_daily_avg['DATE'], trend, label='Trend')
plt.legend(loc='best')
plt.subplot(413)
plt.plot(flights_arrival_delay_daily_avg['DATE'], seasonal, label='Seasonality')
plt.legend(loc='best')
plt.subplot(414)
plt.plot(flights_arrival_delay_daily_avg['DATE'], residual, label='Residuals')
plt.legend(loc='best')
plt.tight_layout()
plt.show()

# Fit ARIMA model
model = ARIMA(flights_arrival_delay_daily_avg['ARRIVAL_DELAY'], order=(1, 1, 1))
model_fit = model.fit()

# Forecast
forecast = model_fit.forecast(steps=31)

# Plot forecast
plt.figure(figsize=(12, 6))
plt.plot(flights_arrival_delay_daily_avg['DATE'], flights_arrival_delay_daily_avg['ARRIVAL_DE
plt.plot(flights_arrival_delay_daily_avg['DATE'].iloc[-1] + pd.to_timedelta(np.arange(1, 32),
plt.xlabel('Date', fontsize=12)
plt.ylabel('Arrival Delay (minutes)', fontsize=12)
plt.title('Forecast of Arrival Delay', fontsize=14)
plt.legend()
plt.grid(True, linestyle='--', alpha=0.5)
plt.show()
```


Distribution of Arrival Delay
Density
0
10
20
30
40
50
0
50
100
150
200
Arrival Delay in minutes


Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
25


Partial Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
25


Original
200
150
100
50
0
2023-01
2023-03
2023-05
2023-07
2023-09
2023-11
2024-01
Trend
40
30
20
10
0
2023-01
2023-03
2023-05
2023-07
2023-09
2023-11
2024-01
Seasonality
0
−1
2023-01
2023-03
2023-05
2023-07
2023-09
2023-11
2024-01
Residuals
150
100
50
0
−50
2023-01
2023-03
2023-05
2023-07
2023-09
2023-11
2024-01


Forecast of Arrival Delay
Actual
Forecast
Arrival Delay (minutes)
200
150
100
50
0
2023-01
2023-03
2023-05
2023-07
2023-09
2023-11
2024-01
Date

```
In [ ]:
```