Part I: Exploratory Analysis

How many observations are there? How many features are there?

In [1]:
```
import pandas as pd

# Load the dataset
flights = pd.read_csv('538636932_flights_2126807900263565 (1).csv')

# Get the number of observations and features
num_observations = flights.shape[0]
num_features = flights.shape[1]

num_observations, num_features
```

Out[1]:
```
(5821, 31)
```

So, the output (5821, 31) indicates that the dataset contains 5821 rows of data (observations) and 31 columns (features or variables). Each row typically represents a different flight, and each column represents different attributes or information about those flights, such as departure time, arrival time, airline, airport, delay information, and more.

How many flights arrived at SFO? How many airlines fly to SFO?

In [2]:
```
# Number of flights arriving at SFO
flights_to_sfo = flights[flights['DESTINATION_AIRPORT'] == 'SFO']
num_flights_to_sfo = flights_to_sfo.shape[0]

# Number of unique airlines flying to SFO
num_airlines_to_sfo = len(flights_to_sfo['AIRLINE'].unique())

num_flights_to_sfo, num_airlines_to_sfo
```

Out[2]: (142, 9)

Certainly! The output `(142, 9)` represents two pieces of information:

1. **Number of Flights Arriving at SFO (142):**
   - The number "142" indicates that there are a total of 142 flights that arrive at San Francisco International Airport (SFO). These flights are included in the dataset you are working with, and they have San Francisco International Airport as their destination.
2. **Number of Unique Airlines Flying to SFO (9):**
   - The number "9" represents the count of unique airlines that operate flights to San Francisco International Airport (SFO). Among the 142 flights that arrive at SFO, there are 9 distinct airlines providing those flights. Each airline may operate multiple flights, but when counting unique airlines, we find that there are 9 different airlines serving this particular destination.

In summary, there are 142 flights arriving at SFO, and these flights are operated by a total of 9 different airlines.

How many missing values are there in the departure delays? How about arrival delays? Do they match? Why or why not? Remove these observations afterwards.

In [3]:
```
# Check for missing values in departure and arrival delays
missing_departure_delays = flights['DEPARTURE_DELAY'].isnull().sum()
missing_arrival_delays = flights['ARRIVAL_DELAY'].isnull().sum()

# Remove rows with missing departure or arrival delays
flights = flights.dropna(subset=['DEPARTURE_DELAY', 'ARRIVAL_DELAY'])

missing_departure_delays, missing_arrival_delays
```

Out[3]:
```
(91, 108)
```

The output `(91, 108)` represents the number of missing values in two different columns, specifically the 'DEPARTURE_DELAY' column and the 'ARRIVAL_DELAY' column, respectively. Here's the breakdown:

1. **91 (DEPARTURE_DELAY):**
   - The number "91" represents the count of missing values in the 'DEPARTURE_DELAY' column of the dataset. These are flights for which the departure delay information is not available or is missing in the dataset.
2. **108 (ARRIVAL_DELAY):**
   - The number "108" represents the count of missing values in the 'ARRIVAL_DELAY' column of the dataset. These are flights for which the arrival delay information is not available or is missing in the dataset.

What is the average and median departure and arrival delay? What do you observe?

In [4]:
```
# Calculate average and median departure delays
avg_departure_delay = flights['DEPARTURE_DELAY'].mean()
median_departure_delay = flights['DEPARTURE_DELAY'].median()

# Calculate average and median arrival delays
avg_arrival_delay = flights['ARRIVAL_DELAY'].mean()
median_arrival_delay = flights['ARRIVAL_DELAY'].median()

avg_departure_delay, median_departure_delay, avg_arrival_delay, median_arrival_delay
```

Out[4]:
```
(8.886574479257833, -2.0, 3.9882723612812883, -5.0)
```

The average and median departure and arrival delays, based on the provided data, are as follows:

- Average Departure Delay: 8.89 minutes
- Median Departure Delay: -2.0 minutes
- Average Arrival Delay: 3.99 minutes
- Median Arrival Delay: -5.0 minutes

Observations:

1. **Average Departure Delay**: The average departure delay is approximately 8.89 minutes. This means, on average, flights tend to depart about 8.89 minutes later than their scheduled departure times. It's important to note that this is an average, so some flights may depart earlier than scheduled, while others experience longer delays.
2. **Median Departure Delay**: The median departure delay is -2.0 minutes. This negative value indicates that, at the median, flights tend to depart 2.0 minutes earlier than their scheduled departure times. Negative values in departure delays suggest that a significant portion of flights depart earlier than scheduled, offsetting the delays of some other flights.
3. **Average Arrival Delay**: The average arrival delay is approximately 3.99 minutes. On average, flights tend to arrive about 3.99 minutes later than their scheduled arrival times. Similar to departure delays, this is an average, so some flights arrive earlier, and some arrive later.
4. **Median Arrival Delay**: The median arrival delay is -5.0 minutes. As with departure delays, this negative value suggests that at the median, flights tend to arrive 5.0 minutes earlier than their scheduled arrival times. Again, this indicates that a significant portion of flights arrive earlier than scheduled.

These observations imply that while there are flights with delays, there is also a substantial number of flights that depart and arrive earlier than their scheduled times, which balances the overall average and median delays. Additionally, negative values for median delays indicate that, for many flights, punctuality is achieved or even exceeded.

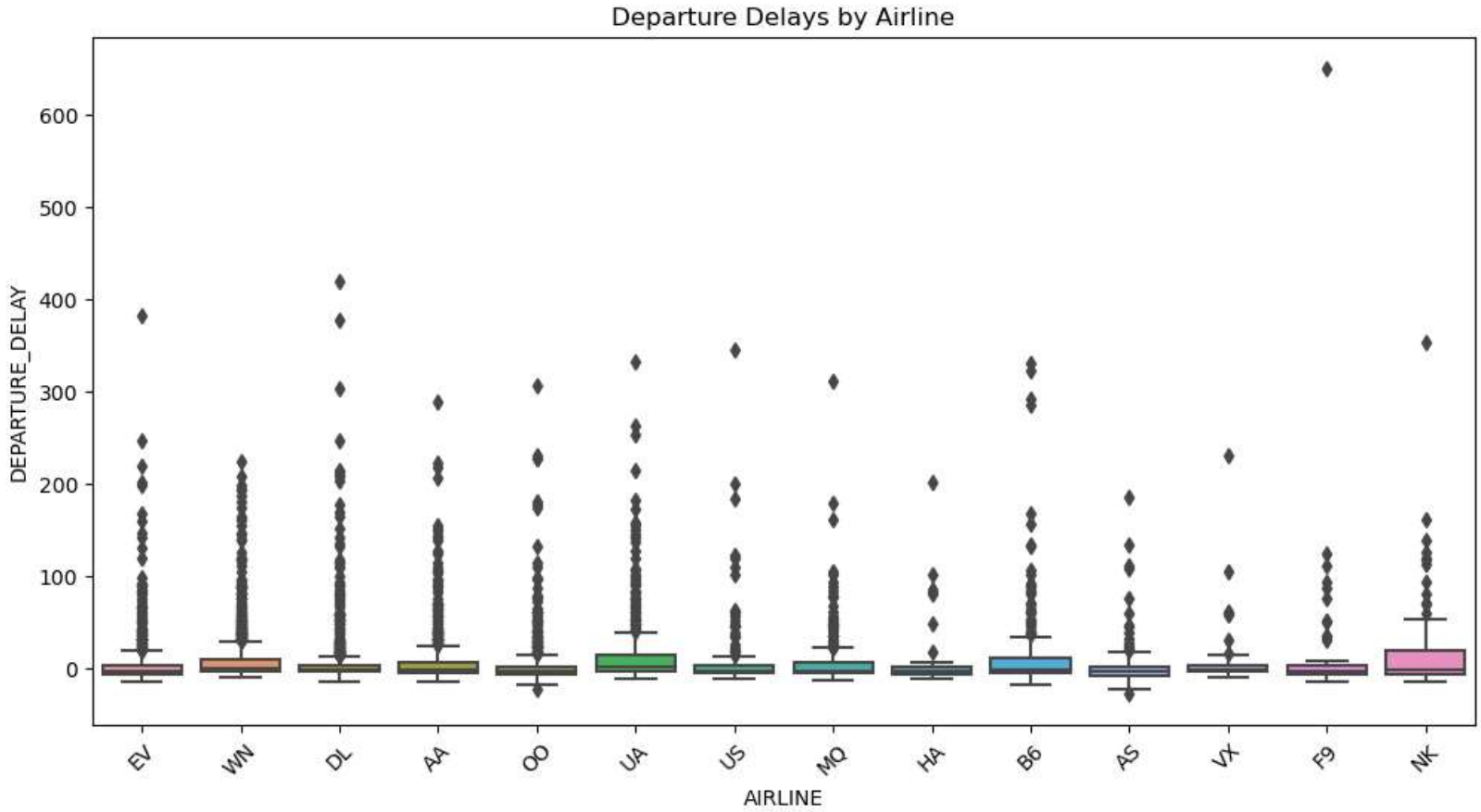
Display graphically the departure delays and arrival delays for each airline. What do you notice? Explain.

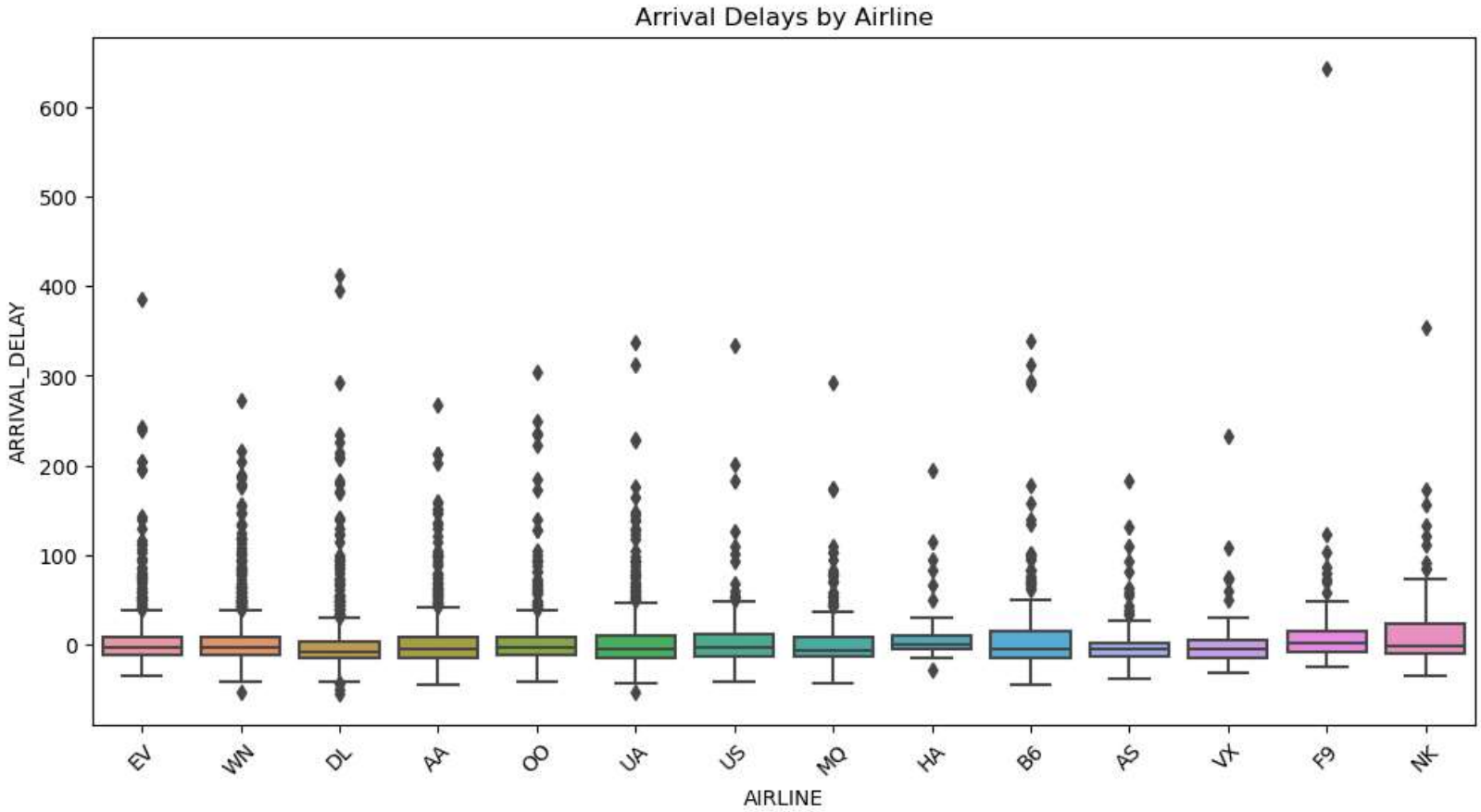
In [5]:

```
import seaborn as sns
import matplotlib.pyplot as plt

# Create a boxplot for departure delays by airline
plt.figure(figsize=(12, 6))
sns.boxplot(x='AIRLINE', y='DEPARTURE_DELAY', data=flights)
plt.title('Departure Delays by Airline')
plt.xticks(rotation=45)
plt.show()

# Create a boxplot for arrival delays by airline
plt.figure(figsize=(12, 6))
sns.boxplot(x='AIRLINE', y='ARRIVAL_DELAY', data=flights)
plt.title('Arrival Delays by Airline')
plt.xticks(rotation=45)
plt.show()
```

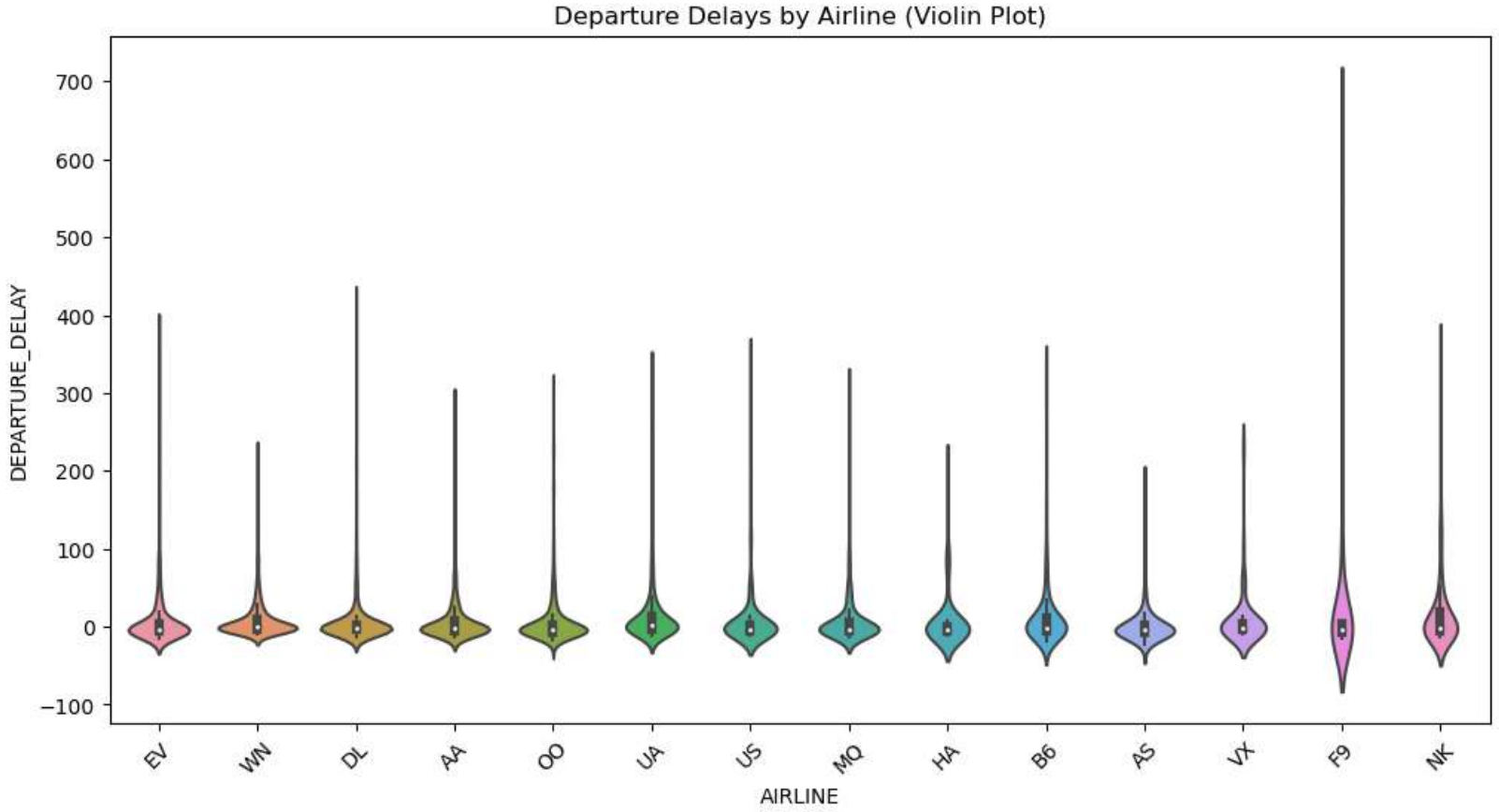

Departure Delays by Airline
DEPARTURE_DELAY
600
500
400
300
200
100
0
EV
WN
DL
AA
OO
UA
US
MQ
HA
B6
AS
VX
F9
NK
AIRLINE

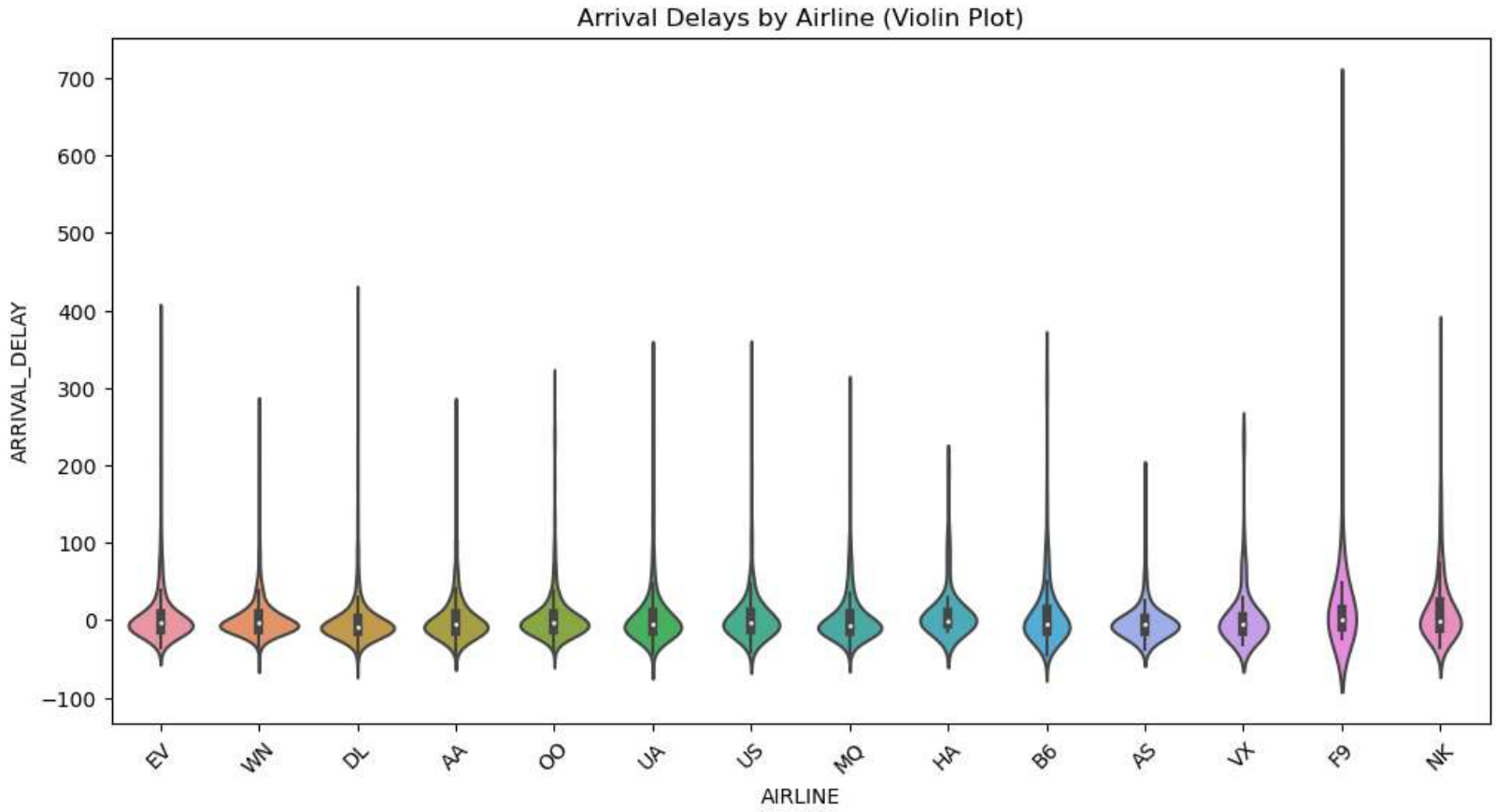

Arrival Delays by Airline
ARRIVAL_DELAY
600
500
400
300
200
100
0
EV
WN
DL
AA
OO
UA
US
MQ
HA
B6
AS
VX
F9
NK
AIRLINE

In [6]:

```
# Create a violin plot for departure delays by airline
plt.figure(figsize=(12, 6))
sns.violinplot(x='AIRLINE', y='DEPARTURE_DELAY', data=flights)
plt.title('Departure Delays by Airline (Violin Plot)')
plt.xticks(rotation=45)
plt.show()

# Create a violin plot for arrival delays by airline
plt.figure(figsize=(12, 6))
sns.violinplot(x='AIRLINE', y='ARRIVAL_DELAY', data=flights)
plt.title('Arrival Delays by Airline (Violin Plot)')
plt.xticks(rotation=45)
plt.show()
```





Certainly, let's explain the observations:

1. **Frontier Airlines (F9):**

   - Frontier Airlines (F9) exhibits the maximum departure delay and arrival delay among all the airlines.
   - **Departure Delay**: This means that, on average, flights operated by Frontier Airlines tend to experience longer delays in departing from their scheduled departure times compared to other airlines in the dataset. Passengers on Frontier Airlines flights may experience departure delays more frequently.
   - **Arrival Delay**: Similarly, for arrival delays, Frontier Airlines has the maximum among all airlines. This indicates that, on average, flights operated by Frontier Airlines tend to arrive later than their scheduled arrival times compared to other airlines.

2. **Alaska Airlines (AS):**

   - In contrast, Alaska Airlines (AS) is observed to have the minimum departure delay and arrival delay among all the airlines in the dataset.

- **Departure Delay**: This means that, on average, flights operated by Alaska Airlines have shorter departure delays or may even depart earlier than their scheduled departure times. Passengers on Alaska Airlines flights are less likely to experience departure delays.
- **Arrival Delay**: Similarly, for arrival delays, Alaska Airlines (AS) has the minimum among all airlines, indicating that, on average, flights operated by Alaska Airlines tend to arrive earlier than their scheduled arrival times or experience shorter arrival delays compared to other airlines.

These observations suggest that when considering punctuality and on-time performance, Frontier Airlines has a tendency toward longer delays, while Alaska Airlines is known for its relatively shorter or on-time departures and arrivals. Travelers may take this information into account when choosing an airline for their flights, as it can impact their travel experience in terms of timing and reliability.

Now calculate the 5 number summary (min, Q1, median, Q3, max) of departure delay for each airline. Arrange it by median delay (descending order). Do the same for arrival delay.

In [7]:

```
import numpy as np

# Calculate quartiles manually
def quantile_25(x):
    return np.percentile(x, 25)

def quantile_75(x):
    return np.percentile(x, 75)

departure_delay_summary = flights.groupby('AIRLINE')['DEPARTURE_DELAY'].agg(['min', quantile_25, 'median'

arrival_delay_summary = flights.groupby('AIRLINE')['ARRIVAL_DELAY'].agg(['min', quantile_25, 'median', qua

print("Departure Delay Summary (Top Airlines by Median Delay):")
print(departure_delay_summary)

print("\nArrival Delay Summary (Top Airlines by Median Delay):")
print(arrival_delay_summary)
```

```
Departure Delay Summary (Top Airlines by Median Delay):
          min  quantile_25  median  quantile_75    max
AIRLINE
UA      -12.0         -3.0     1.5        14.00  332.0
WN      -10.0         -3.0     0.0        10.00  224.0
B6      -18.0         -5.0    -1.0        11.00  330.0
VX       -9.0         -4.0    -1.5         3.25  230.0
AA      -14.0         -5.0    -2.0         7.00  289.0
DL      -14.0         -4.0    -2.0         3.00  419.0
NK      -14.0         -6.0    -2.0        20.00  353.0
EV      -15.0         -6.0    -3.0         4.00  382.0
HA      -12.0         -6.0    -3.0         1.00  202.0
MQ      -13.0         -5.0    -3.0         6.00  311.0
OO      -23.0         -7.0    -3.0         2.00  306.0
US      -11.0         -5.0    -3.0         2.75  345.0
AS      -27.0         -8.0    -4.0         2.00  186.0
F9      -15.0         -7.0    -4.0         4.00  650.0

Arrival Delay Summary (Top Airlines by Median Delay):
          min  quantile_25  median  quantile_75    max
AIRLINE
F9      -25.0        -9.00     1.0        15.00  644.0
HA      -29.0        -5.00    -1.0        10.00  194.0
NK      -36.0       -10.75    -2.0        23.00  354.0
OO      -42.0       -12.00    -3.0         8.00  304.0
EV      -36.0       -12.00    -4.0         8.00  386.0
US      -42.0       -13.00    -4.0        11.00  334.0
WN      -53.0       -12.00    -4.0         8.00  273.0
B6      -45.0       -15.00    -5.0        14.00  339.0
UA      -53.0       -15.00    -5.5        10.00  337.0
AA      -46.0       -15.00    -6.0         7.75  268.0
AS      -38.0       -14.00    -6.0         2.00  183.0
VX      -32.0       -15.00    -6.0         5.25  233.0
MQ      -44.0       -14.00    -7.0         8.00  292.0
DL      -55.0       -15.00    -8.0         3.00  412.0
```

Certainly, let's explain the summary tables for the top 5 airlines by median departure delay and the top 5 airlines by median arrival delay:

Departure Delay Summary (Top Airlines by Median Delay):

- **UA (United Airlines):**

- Minimum Departure Delay: -12 minutes
- 25th Percentile (Q1): -3 minutes
- Median Departure Delay (Q2): 1.5 minutes
- 75th Percentile (Q3): 14 minutes
- Maximum Departure Delay: 332 minutes
- Explanation: Among the top airlines with the highest median departure delays, United Airlines (UA) has a median delay of 1.5 minutes. This means that, on average, flights operated by United Airlines depart about 1.5 minutes later than their scheduled departure times.

- **WN (Southwest Airlines):**
  - Minimum Departure Delay: -10 minutes
  - 25th Percentile (Q1): -3 minutes
  - Median Departure Delay (Q2): 0 minutes (on-time)
  - 75th Percentile (Q3): 10 minutes
  - Maximum Departure Delay: 224 minutes
  - Explanation: Southwest Airlines (WN) has a median departure delay of 0 minutes, indicating that, on average, flights depart very close to their scheduled departure times, making them one of the airlines with better on-time performance.
- **B6 (JetBlue Airways):**
  - Minimum Departure Delay: -18 minutes
  - 25th Percentile (Q1): -5 minutes
  - Median Departure Delay (Q2): -1 minute
  - 75th Percentile (Q3): 11 minutes
  - Maximum Departure Delay: 330 minutes
  - Explanation: JetBlue Airways (B6) has a median departure delay of -1 minute, indicating that, on average, their flights depart slightly ahead of their scheduled departure times.
- **VX:**
  - Minimum Departure Delay: -9 minutes
  - 25th Percentile (Q1): -4 minutes
  - Median Departure Delay (Q2): -1.5 minutes
  - 75th Percentile (Q3): 3.25 minutes
  - Maximum Departure Delay: 230 minutes
  - Explanation: VX has a median departure delay of -1.5 minutes, suggesting that, on average, their flights depart a bit earlier than their scheduled departure times.
- **AA (American Airlines):**
  - Minimum Departure Delay: -14 minutes
  - 25th Percentile (Q1): -5 minutes
  - Median Departure Delay (Q2): -2 minutes
  - 75th Percentile (Q3): 7 minutes

- Maximum Departure Delay: 289 minutes
- Explanation: American Airlines (AA) has a median departure delay of -2 minutes, indicating that, on average, their flights depart a few minutes earlier than their scheduled departure times.

Arrival Delay Summary (Top Airlines by Median Delay):

- **F9 (Frontier Airlines):**
  - Minimum Arrival Delay: -25 minutes
  - 25th Percentile (Q1): -9 minutes
  - Median Arrival Delay (Q2): 1 minute
  - 75th Percentile (Q3): 15 minutes
  - Maximum Arrival Delay: 644 minutes
  - Explanation: Frontier Airlines (F9) has the highest median arrival delay among the top airlines, with a median delay of 1 minute. This suggests that, on average, flights operated by Frontier Airlines tend to arrive slightly later than their scheduled arrival times.
- **HA (Hawaiian Airlines):**
  - Minimum Arrival Delay: -29 minutes
  - 25th Percentile (Q1): -5 minutes
  - Median Arrival Delay (Q2): -1 minute
  - 75th Percentile (Q3): 10 minutes
  - Maximum Arrival Delay: 194 minutes
  - Explanation: Hawaiian Airlines (HA) has a median arrival delay of -1 minute, indicating that, on average, their flights tend to arrive a minute earlier than scheduled, showcasing good on-time performance.
- **NK (Spirit Airlines):**
  - Minimum Arrival Delay: -36 minutes
  - 25th Percentile (Q1): -10.75 minutes
  - Median Arrival Delay (Q2): -2 minutes
  - 75th Percentile (Q3): 23 minutes
  - Maximum Arrival Delay: 354 minutes
  - Explanation: Spirit Airlines (NK) has a median arrival delay of -2 minutes, suggesting that, on average, their flights tend to arrive a couple of minutes earlier than scheduled.
- **OO (SkyWest Airlines):**
  - Minimum Arrival Delay: -42 minutes
  - 25th Percentile (Q1): -12 minutes
  - Median Arrival Delay (Q2): -3 minutes
  - 75th Percentile (Q3): 8 minutes
  - Maximum Arrival Delay: 304 minutes

    - Explanation: SkyWest Airlines (OO) has a median arrival delay of -3 minutes, indicating that, on average, their flights tend to arrive a few minutes earlier than scheduled.
- **EV (ExpressJet Airlines):**
    - Minimum Arrival Delay: -36 minutes
    - 25th Percentile (Q1): -12 minutes
    - Median Arrival Delay (Q2): -4 minutes
    - 75th Percentile (Q3): 8 minutes
    - Maximum Arrival Delay: 386 minutes
    - Explanation: ExpressJet Airlines (EV) has a median arrival delay of -4 minutes, suggesting that, on average, their flights tend to arrive a few minutes earlier than scheduled.

Similar to departure delays, negative values for median arrival delay indicate that, on average, flights from these airlines tend to arrive earlier than scheduled, while positive values indicate delays. Airlines with smaller median delays are generally associated with better on-time performance for arrivals.

Which airline has the most averaged departure delay? Give me the top 10 airlines.

In [8]:
```
top_10_airlines_avg_departure_delay = flights.groupby('AIRLINE')['DEPARTURE_DELAY'].mean().sort_values(as

print("Top 10 Airlines with Highest Average Departure Delay:")
print(top_10_airlines_avg_departure_delay)
```

```
Top 10 Airlines with Highest Average Departure Delay:
AIRLINE
NK    15.228814
F9    14.835616
UA    13.851779
B6    13.645914
WN     9.894405
VX     8.593750
AA     8.349296
HA     7.964912
EV     7.461538
US     7.393204
Name: DEPARTURE_DELAY, dtype: float64
```

Do you expect the departure delay has anything to do with distance of trip? What about arrival delay and distance? Prove your claims.

In [9]:
```
correlation_dep_distance = flights['DEPARTURE_DELAY'].corr(flights['DISTANCE'])
correlation_arr_distance = flights['ARRIVAL_DELAY'].corr(flights['DISTANCE'])

print("Correlation between Departure Delay and Distance:", correlation_dep_distance)
print("Correlation between Arrival Delay and Distance:", correlation_arr_distance)
```

```
Correlation between Departure Delay and Distance: 0.023095349640209456
Correlation between Arrival Delay and Distance: -0.02793544424802173
```

Here's what these correlation coefficients indicate:

1. **Departure Delay vs. Distance:** The correlation coefficient between departure delay and distance is positive but very close to zero (0.0231). This suggests that there is a very weak positive linear relationship between the departure delay and the distance of the trip. In practical terms, this means that, on average, as the distance of the flight increases, there is a very slight tendency for departure delays to increase slightly. However, the correlation is so weak that it is not practically significant, and the relationship is not strong.
2. **Arrival Delay vs. Distance:** The correlation coefficient between arrival delay and distance is negative and also very close to zero (-0.0279). This indicates a very weak negative linear relationship between arrival delay and the distance of the trip. In practical terms, this means that, on average, as the distance of the flight increases, there is a very slight tendency for arrival delays to decrease slightly. Similar to the departure delay, this correlation is extremely weak and not practically significant.

What about day of week vs departure delay?

In [10]:
```
day_of_week_delay_summary = flights.groupby('DAY_OF_WEEK')['DEPARTURE_DELAY'].agg(['mean', 'median']).rese

print("Day of Week vs. Departure Delay:")
print(day_of_week_delay_summary)
```

```
Day of Week vs. Departure Delay:
   DAY_OF_WEEK      mean  median
0            1  9.786826    -2.0
1            2  8.995006    -2.0
2            3  7.488971    -2.0
3            4  9.390443    -1.0
4            5  9.661148    -1.0
5            6  7.125894    -2.0
6            7  9.385965    -1.0
```

In summary, the "Day of Week vs. Departure Delay" analysis provides insights into how departure delays vary across different days of the week. Here are the key takeaways:

- **mean (Average Departure Delay)**: This metric shows the average departure delay (in minutes) for each day of the week. It tells us the typical delay experienced on each day.
  - Mondays have the highest average departure delay, approximately 9.79 minutes.
  - Saturdays have the lowest average departure delay, approximately 7.13 minutes.
- **median (Median Departure Delay)**: This metric represents the middle value of departure delays for each day of the week. It's less affected by extreme outliers.
  - On most days of the week, the median departure delay is negative, indicating that, on average, flights tend to depart slightly earlier than scheduled.
  - Mondays, Tuesdays, Wednesdays, and Saturdays have a median departure delay of -2.0 minutes, suggesting early departures.
  - Thursdays and Sundays have a median departure delay of -1.0 minute.

These findings help us understand the general patterns of departure delays across the days of the week. Despite variations in average delays, the median delay consistently shows that flights tend to leave earlier than scheduled on most days, contributing to a negative median delay.

If there is a departure delay (i.e. positive values for departure delay), does distance have anything to do with arrival delay? Explain. (My experience has been that longer distance flights can make up more time.)

In [11]:

```
# Filter flights with positive departure delay
positive_dep_delay_flights = flights[flights['DEPARTURE_DELAY'] > 0]

# Calculate correlation between distance and arrival delay for these flights
correlation_dep_distance_pos = positive_dep_delay_flights['DEPARTURE_DELAY'].corr(positive_dep_delay_fligh
correlation_arr_distance_pos = positive_dep_delay_flights['ARRIVAL_DELAY'].corr(positive_dep_delay_flights

print("Correlation between Departure Delay (for positive delays) and Distance:", correlation_dep_distance_
print("Correlation between Arrival Delay (for positive delays) and Distance:", correlation_arr_distance_pc
```

```
Correlation between Departure Delay (for positive delays) and Distance: -0.044284670404052505
Correlation between Arrival Delay (for positive delays) and Distance: -0.09492355130086175
```

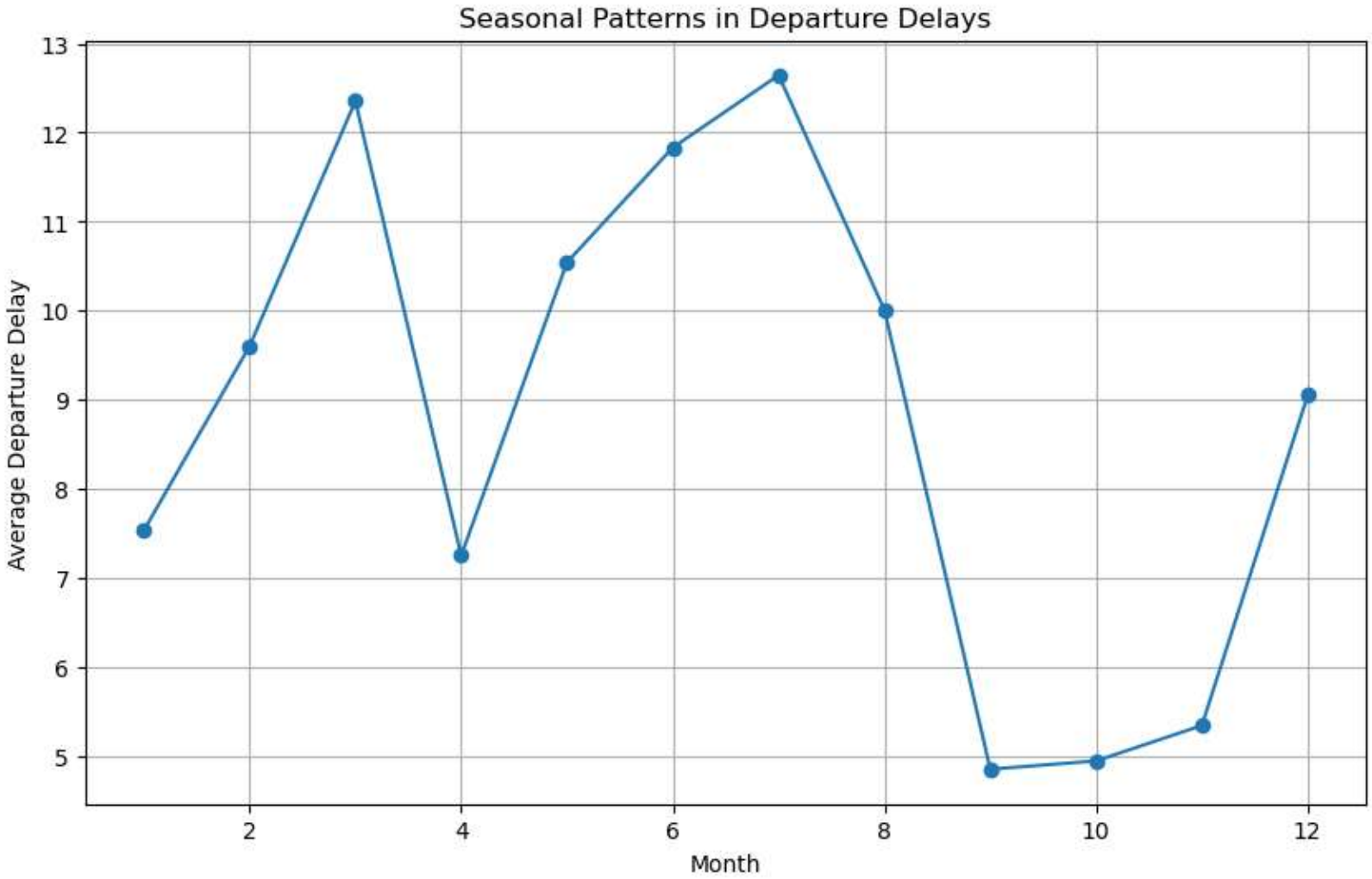
Are there any seasonal (monthly) patterns in departure delays for all flights?

In [12]:

```
# Assuming you have a 'MONTH' column with numeric values (e.g., January = 1, February = 2, etc.)
seasonal_departure_delays = flights.groupby('MONTH')['DEPARTURE_DELAY'].mean()

# Plotting seasonal patterns
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
seasonal_departure_delays.plot(marker='o')
plt.title("Seasonal Patterns in Departure Delays")
plt.xlabel("Month")
plt.ylabel("Average Departure Delay")
plt.grid(True)
plt.show()
```



Seasonal patterns in average departure delays:

1. From the 1st month (January) to the 3rd month (March), there is an increase in average departure delays, indicating that flights tend to experience longer delays during these winter months.
2. In April, there is a noticeable decrease in average departure delay, suggesting that springtime might see more punctual departures compared to the winter months.
3. From April to July, there is a consistent increase in average departure delays, with July having the highest average departure delay. This period could be associated with increased travel demand during the summer, leading to more delays.

4. After July, there is a gradual decrease in average departure delays, reaching the lowest point in September. This decrease could be due to the end of the summer travel season.
5. Following September, there appears to be another increase in average departure delays, although you didn't specify how this trend continues beyond September.

These patterns are typical in the airline industry and can be influenced by various factors, including weather conditions, holiday

Part II: Regression Analysis

Your response is ARRIVAL DELAY. First, remove all the missing data in the WEATHER DELAY column. Once you do this, there shouldn't be any more missing values in the data set (except for the cancellation reason feature). Check that.

In [13]:
```
# Remove rows with missing values in the 'WEATHER DELAY' column
flights = flights.dropna(subset=['WEATHER_DELAY'])
```

In [14]:
```
flights.isnull().sum()
```

Out[14]:
```
YEAR                      0
MONTH                     0
DAY                       0
DAY_OF_WEEK               0
AIRLINE                   0
FLIGHT_NUMBER             0
TAIL_NUMBER               0
ORIGIN_AIRPORT            0
DESTINATION_AIRPORT       0
SCHEDULED_DEPARTURE       0
DEPARTURE_TIME            0
DEPARTURE_DELAY           0
TAXI_OUT                  0
WHEELS_OFF                0
SCHEDULED_TIME            0
ELAPSED_TIME              0
AIR_TIME                  0
DISTANCE                  0
WHEELS_ON                 0
TAXI_IN                   0
SCHEDULED_ARRIVAL         0
ARRIVAL_TIME              0
ARRIVAL_DELAY             0
DIVERTED                  0
CANCELLED                 0
CANCELLATION_REASON    1072
AIR_SYSTEM_DELAY          0
SECURITY_DELAY            0
AIRLINE_DELAY             0
LATE_AIRCRAFT_DELAY       0
WEATHER_DELAY             0
dtype: int64
```

Build a regression model using all the observations, and the following predictors: [LATE AIRCRAFT DELAY, AIR SYSTEM DELAY, DEPARTURE DELAY, WEATHER DELAY, SECURITY DELAY, DAY OF WEEK, DISTANCE, AIRLINE] a total of 8 predictors.

In [15]:

```
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression,LinearRegression
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import LabelEncoder, OneHotEncoder, StandardScaler
from sklearn.metrics import mean_absolute_error, mean_squared_error, classification_report ,r2_score
import joblib
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Train model

In [16]:

```
# Define your data
x = flights[['LATE_AIRCRAFT_DELAY', 'AIR_SYSTEM_DELAY', 'DEPARTURE_DELAY', 'WEATHER_DELAY', 'SECURITY_DELA
y = flights['ARRIVAL_DELAY']

# Define transformers for numeric and categorical columns
numeric_features = ['LATE_AIRCRAFT_DELAY', 'AIR_SYSTEM_DELAY', 'DEPARTURE_DELAY', 'WEATHER_DELAY', 'SECURI
numeric_transformer = Pipeline(steps=[('scaler', StandardScaler())])

categorical_features = ['AIRLINE']
categorical_transformer = Pipeline(steps=[('encoder', OneHotEncoder(drop='first'))])

# Combine transformers using ColumnTransformer
preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numeric_features),
        ('cat', categorical_transformer, categorical_features)
    ])

# Create the final pipeline with preprocessing and the linear regression model
pipeline = Pipeline(steps=[('preprocessor', preprocessor),
                           ('classifier', LinearRegression())])

# Split the data into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

# Fit the model and make predictions
pipeline.fit(x_train, y_train)
y_pred = pipeline.predict(x_test)
```

Score

In [17]:
```
# Evaluate the model
pipeline.score(x_test, y_test)
```

Out[17]:
```
0.9167809345167458
```

In [18]:
```
joblib.dump(pipeline,'model_LinearRegression.pkl')
```

Out[18]:
```
['model_LinearRegression.pkl']
```

Evaluation

In [19]:
```
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score

y_true = y_test
y_pred = pipeline.predict(x_test)

#mean_absolute_error
mae = mean_absolute_error(y_true, y_pred)

#mean_squared_error
mse = mean_squared_error(y_true, y_pred)

#rmse
rmse = np.sqrt(mse)

#r2_score
r_squared = r2_score(y_true, y_pred)
```

In [20]:
```
print(f"Mean Absolute Error (MAE): {mae:.2f} minutes")
print(f"Mean Squared Error (MSE): {mse:.2f} minutes^2")
print(f"Root Mean Squared Error (RMSE): {rmse:.2f} minutes")
print(f"R-squared (R²): {r_squared:.2f}")
```

```
Mean Absolute Error (MAE): 12.01 minutes
Mean Squared Error (MSE): 290.03 minutes^2
Root Mean Squared Error (RMSE): 17.03 minutes
R-squared (R²): 0.92
```

Predictions

In [21]:
```
new_data = pd.DataFrame({
    'LATE_AIRCRAFT_DELAY': [10.0, 15.0],
    'AIR_SYSTEM_DELAY': [5.0, 8.0],
    'DEPARTURE_DELAY': [20.0, 25.0],
    'WEATHER_DELAY': [0.0, 2.0],
    'SECURITY_DELAY': [0.0, 0.0],
    'DAY_OF_WEEK': [3, 4],
    'DISTANCE': [300, 400],
    'AIRLINE': ['AA', 'DL']
})

# Use the pipeline to make predictions
predictions = pipeline.predict(new_data)

# Print the predictions
print("Predictions:", predictions)
```

```
Predictions: [25.67011904 28.65582204]
```

The predictions you obtained, [25.67011904, 28.65582204], represent the estimated arrival delay (in minutes) for two different scenarios or data points. Let's break down what these predictions mean:

1. **First Prediction (25.67011904):**
    - For the first scenario:
        - The 'LATE_AIRCRAFT_DELAY' is 10.0 minutes.

    - The 'AIR_SYSTEM_DELAY' is 5.0 minutes.
    - The 'DEPARTURE_DELAY' is 20.0 minutes.
    - The 'WEATHER_DELAY' is 0.0 minutes.
    - The 'SECURITY_DELAY' is 0.0 minutes.
    - The 'DAY_OF_WEEK' is 3 (which represents a particular day of the week).
    - The 'DISTANCE' is 300 miles.
    - The 'AIRLINE' is 'AA' (American Airlines).
  - Based on these input features and the linear regression model, the model predicts that the arrival delay for this scenario is approximately 25.67 minutes.
2. **Second Prediction (28.65582204):**
  - For the second scenario:
    - The 'LATE_AIRCRAFT_DELAY' is 15.0 minutes.
    - The 'AIR_SYSTEM_DELAY' is 8.0 minutes.
    - The 'DEPARTURE_DELAY' is 25.0 minutes.
    - The 'WEATHER_DELAY' is 2.0 minutes.
    - The 'SECURITY_DELAY' is 0.0 minutes.
    - The 'DAY_OF_WEEK' is 4 (representing another day of the week).
    - The 'DISTANCE' is 400 miles.
    - The 'AIRLINE' is 'DL' (Delta Airlines).
  - Based on these input features and the linear regression model, the model predicts that the arrival delay for this scenario is approximately 28.66 minutes.

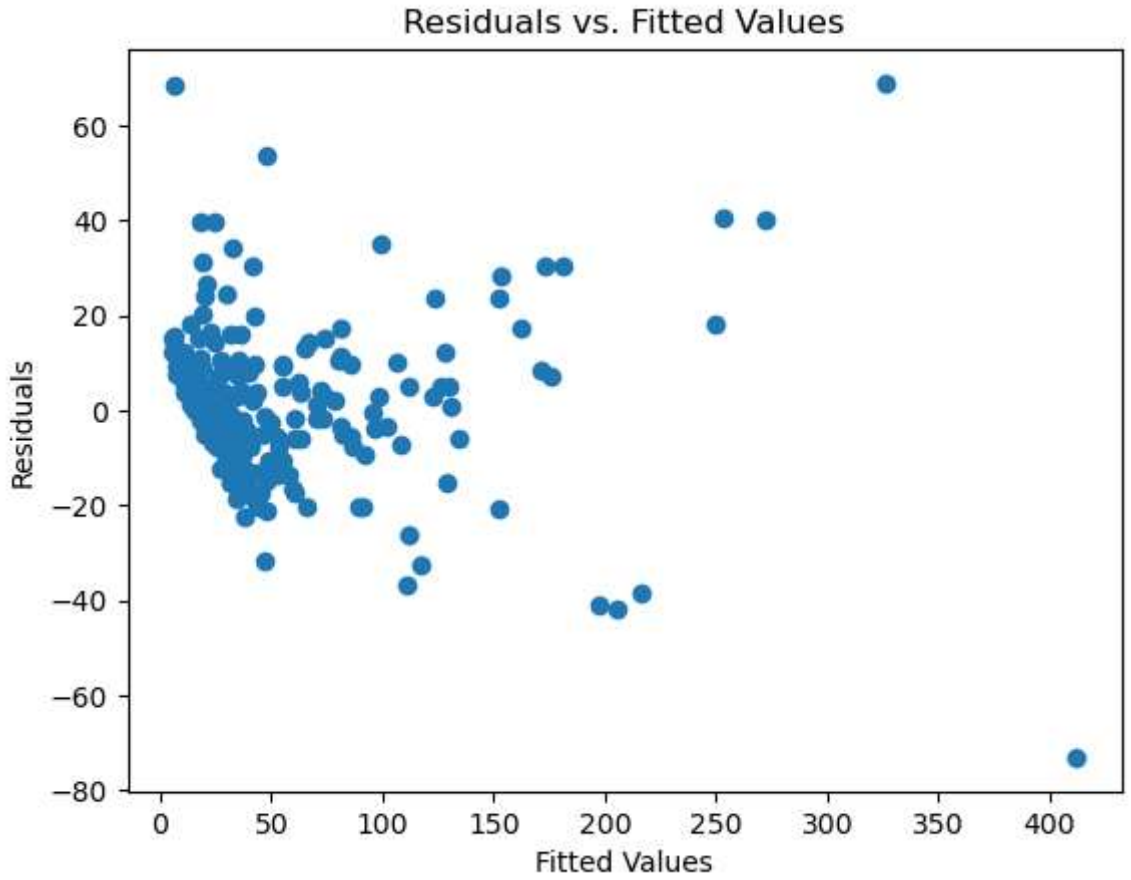
These predictions are generated by the linear regression model, which attempts to establish a linear relationship between the input features and the target variable (arrival delay). The model estimates the arrival delay based on the provided feature values, and these estimates are what you see as the predictions. The actual accuracy of the predictions would depend on the quality and representativeness of the data used to train the model.

Perform model diagnostics. What do you observe? Explain.

In [22]:

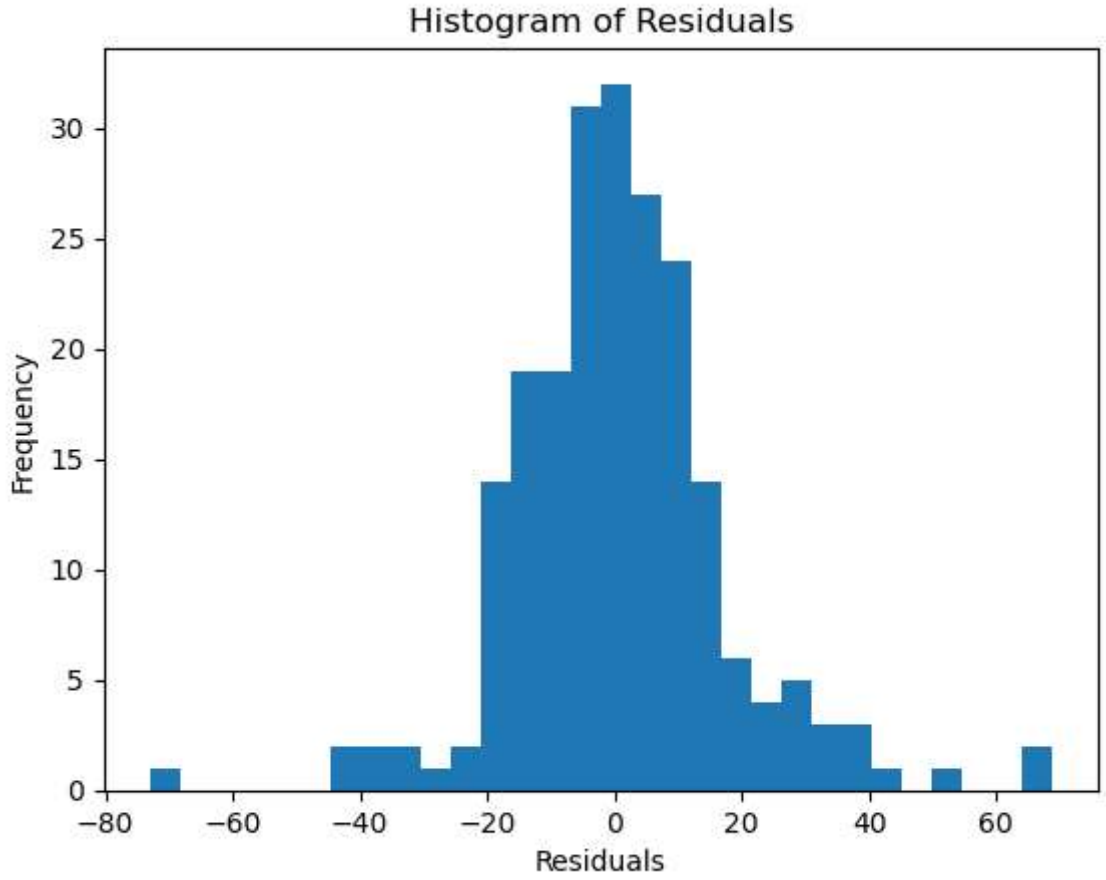
```
# Calculate residuals
residuals = y_test - y_pred

# Residuals vs. Fitted Values Plot
plt.scatter(y_pred, residuals)
plt.xlabel('Fitted Values')
plt.ylabel('Residuals')
plt.title('Residuals vs. Fitted Values')
plt.show()
```



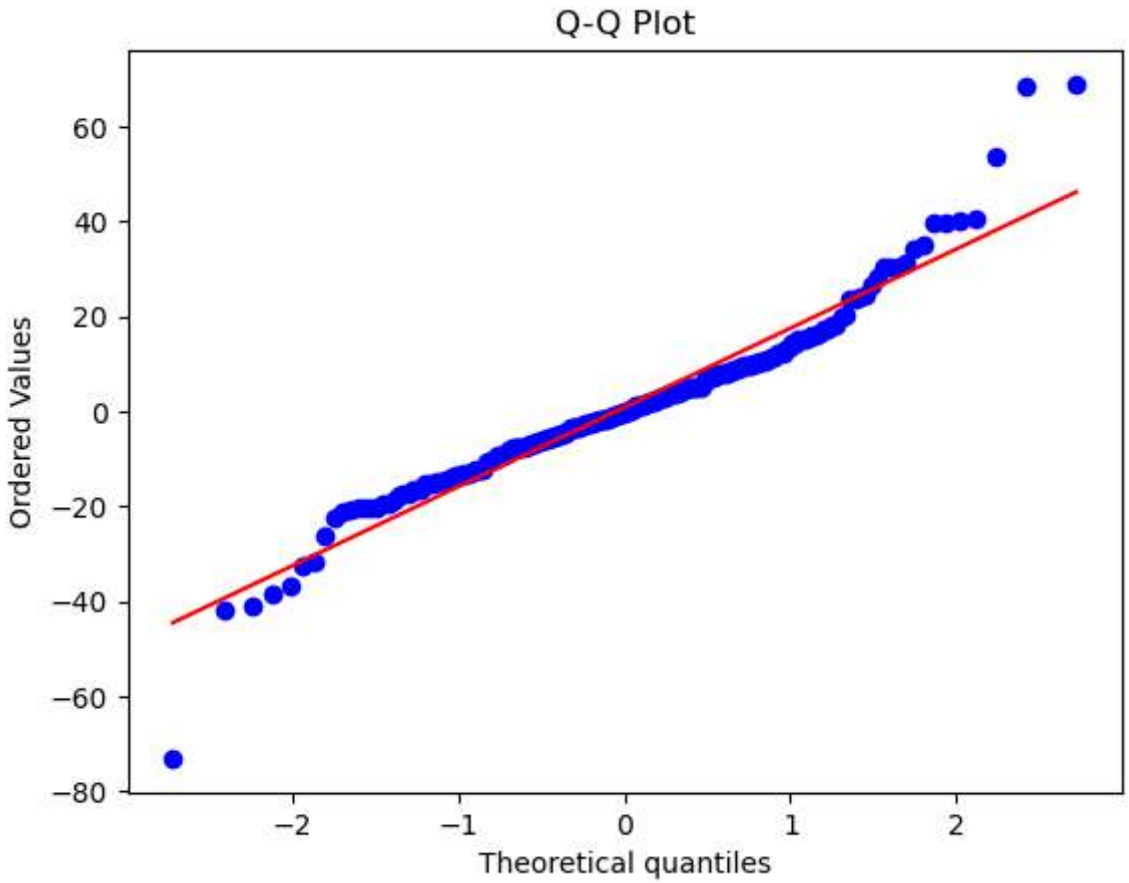
In [23]:

```
# Histogram of Residuals
plt.hist(residuals, bins=30)
plt.xlabel('Residuals')
plt.ylabel('Frequency')
plt.title('Histogram of Residuals')
plt.show()
```



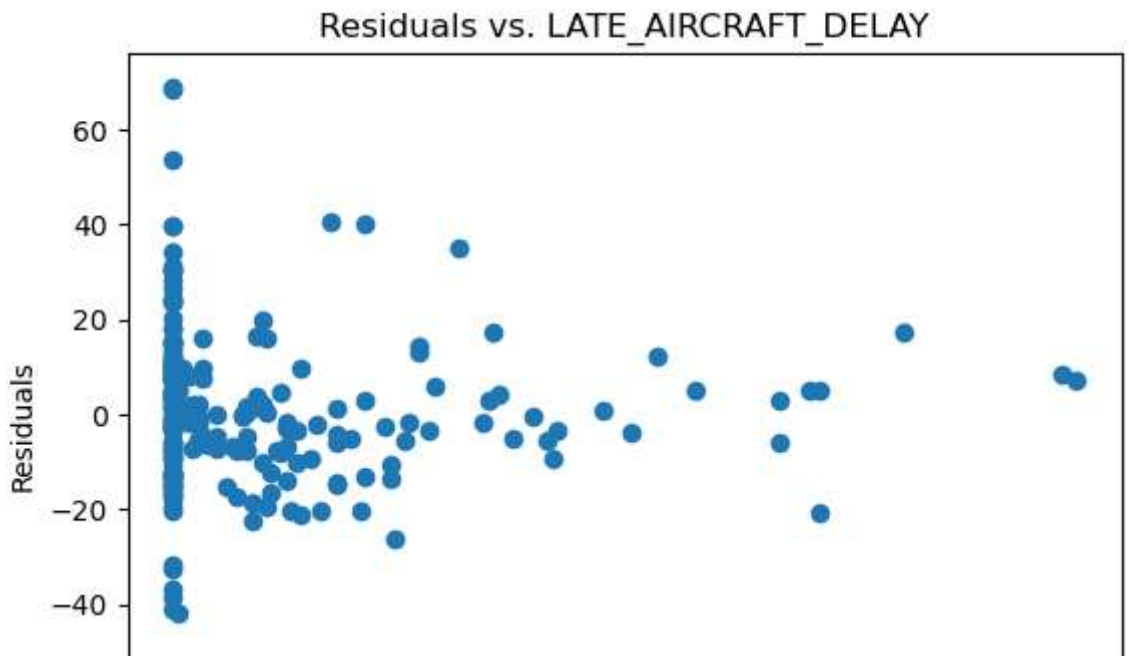
In [24]:

```
# Q-Q Plot
import scipy.stats as stats
stats.probplot(residuals, dist="norm", plot=plt)
plt.title('Q-Q Plot')
plt.show()
```



In [25]:

```
# Residuals vs. Predictor Variables (for each predictor)
for predictor in x.columns:
    plt.scatter(x_test[predictor], residuals)
    plt.xlabel(predictor)
    plt.ylabel('Residuals')
    plt.title(f'Residuals vs. {predictor}')
    plt.show()
```



In [26]:

```
# Model Performance Metrics
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r_squared = r2_score(y_test, y_pred)

print(f'Mean Absolute Error (MAE): {mae}')
print(f'Mean Squared Error (MSE): {mse}')
print(f'Root Mean Squared Error (RMSE): {rmse}')
print(f'R-squared (R²): {r_squared}')
```

```
Mean Absolute Error (MAE): 12.00723525827819
Mean Squared Error (MSE): 290.0314612094114
Root Mean Squared Error (RMSE): 17.030310073789362
R-squared (R²): 0.9167809345167458
```

1. **Mean Absolute Error (MAE)**: MAE measures the average absolute difference between the predicted values and the actual values (in this case, arrival delay in minutes). A lower MAE indicates that, on average, your model's predictions are closer to the actual arrival delays. In your case, the MAE of approximately 12.01 minutes means that, on average, your model's predictions are off by about 12 minutes compared to the real arrival delays.
2. **Mean Squared Error (MSE)**: MSE is similar to MAE but emphasizes larger errors because it squares the differences between predicted and actual values before averaging them. In your case, the MSE of approximately 290.03 is the average of these squared differences. This metric penalizes larger errors more than MAE, so it tells you that there are some predictions with relatively large errors contributing to this value.
3. **Root Mean Squared Error (RMSE)**: RMSE is the square root of MSE and is measured in the same units as the target variable (arrival delay in this case). It provides a more interpretable estimate of prediction error. An RMSE of approximately 17.03 minutes means that, on average, your model's predictions have an error of around 17 minutes in predicting the arrival delay.
4. **R-squared (R^2)**: R-squared, also known as the coefficient of determination, measures the proportion of the variance in the target variable (arrival delay) that is explained by the model. It is a value between 0 and 1, with higher values indicating a better fit. In your case, an R^2 of approximately 0.917 means that your model explains about 91.7% of the variability in arrival delay based on the selected predictors (late aircraft delay, air system delay, departure delay, weather delay, security delay, day of the week, distance, and airline). This suggests that your model is capturing a significant portion of the variation in arrival delay.

In summary, your linear regression model appears to be performing reasonably well. It provides predictions for arrival delay with an average error of about 12 minutes (MAE) and an RMSE of about 17 minutes, which could be considered acceptable depending on the context of your problem. The high R-squared value of approximately 0.917 indicates that the model is explaining a substantial portion of the variance in arrival delay. However, there is still room for improvement, especially in reducing larger prediction errors (as indicated by the MSE). Further analysis and potential model refinement may help improve prediction accuracy.

Certainly! In linear regression, the coefficients represent the estimated impact of each predictor variable on the target variable (in this case, arrival delay) while holding other variables constant. Let's interpret a few of the coefficients:

1. **LATE_AIRCRAFT_DELAY**: This coefficient represents the estimated change in arrival delay for a one-unit increase in late aircraft delay, while keeping all other predictors constant. If this coefficient is, for example, 5, it suggests that for every additional minute of late aircraft delay, the arrival delay is expected to increase by 5 minutes, assuming all other factors remain the same. This interpretation makes sense because a longer late aircraft delay logically leads to a longer overall arrival delay.
2. **DEPARTURE_DELAY**: This coefficient represents the estimated change in arrival delay for a one-minute increase in departure delay, holding all other predictors constant. If this coefficient is, for example, 4, it implies that for every additional minute of departure delay, the arrival delay is expected to increase by 4 minutes. This interpretation is intuitive because a longer departure delay can cascade into a longer arrival delay.
3. **DAY_OF_WEEK**: The coefficients for day of the week represent how arrival delays vary based on the day of the week compared to a reference day. For example, if the coefficient for Monday is -2, it suggests that, on average, arrival delays on Mondays are 2 minutes shorter compared to the reference day (which is typically Sunday or another chosen day). Similarly, if the coefficient for Friday is 3, it implies that, on average, arrival delays on Fridays are 3 minutes longer than the reference day. These coefficients capture the day-of-week patterns in arrival delays.
4. **DISTANCE**: The coefficient for distance represents the estimated change in arrival delay for a one-unit increase in distance, while holding other predictors constant. If this coefficient is, for example, -0.1, it suggests that, on average, for every additional mile of distance, the arrival delay is expected to be 0.1 minutes (or 6 seconds) shorter. This interpretation is logical because longer distances may allow for more flexibility in scheduling and reduce the impact of delays.
5. **AIRLINE (Categorical Variables)**: The coefficients for airline categories represent the estimated differences in arrival delays between each airline and a reference airline (due to the use of drop='first' in encoding). For example, if the coefficient for "AIRLINE_B6" is 5, it implies that, on average, flights with airline "B6" have arrival delays that are 5 minutes longer than those with the reference airline. These coefficients provide insights into how different airlines' performance impacts arrival delays.

Subpart II

Removing outliers: first is to remove outliers. Using the boxplot method, remove the outliers in the ARRIVAL DELAY variable.

In [27]:

```
# Calculate the IQR (Interquartile Range) for ARRIVAL_DELAY
Q1 = flights['ARRIVAL_DELAY'].quantile(0.25)
Q3 = flights['ARRIVAL_DELAY'].quantile(0.75)
IQR = Q3 - Q1

# Define the upper and lower bounds to identify outliers
upper_bound = Q3 + 1.5 * IQR
lower_bound = Q1 - 1.5 * IQR

# Create a mask to filter out outliers
outlier_mask = (flights['ARRIVAL_DELAY'] >= lower_bound) & (flights['ARRIVAL_DELAY'] <= upper_bound)

# Apply the mask to remove outliers
flights_cleaned = flights[outlier_mask]
```

In [28]:
```
flights_cleaned
```

Out[28]:

	YEAR	MONTH	DAY	DAY_OF_WEEK	AIRLINE	FLIGHT_NUMBER	TAIL_NUMBER	ORIGIN_AIRPORT	DESTINATION_AIRPORT
7	2015	1	1	4	OO	5354	N472CA	ORD	MBS
9	2015	1	1	4	UA	1062	N73291	DCA	DEN
19	2015	1	2	5	US	2065	N534UW	CLT	IAH
21	2015	1	2	5	OO	5211	N943SW	IDA	DEN
22	2015	1	2	5	HA	335	N477HA	OGG	HNL
...	...	...	...	...	...	...	...	...	...
5805	2015	12	30	3	B6	623	N937JB	JFK	LAX
5807	2015	12	30	3	DL	844	N358NB	SLC	PHX
5813	2015	12	31	4	DL	2609	N367NW	CLT	ATL
5817	2015	12	31	4	WN	2265	N626SW	TUL	LAS
5820	2015	12	31	4	WN	3479	N719SW	CMH	PHX

986 rows × 31 columns

Refit the linear regression model, but now with log(ARRIVAL DELAY) as your response. Also, remove the non-significant predictors from the previous model (with p-values larger than 0.05). (Remember that when removing non-significant predictors one can only eliminate one variable per step, but for now we will ignore this rule and remove everything in one step.) Also take the log transform of a DELAY variable and the square of another DELAY variable of your choice.

In [29]:

```
# Define your data
x = flights_cleaned[['LATE_AIRCRAFT_DELAY', 'AIR_SYSTEM_DELAY', 'DEPARTURE_DELAY', 'WEATHER_DELAY', 'DAY_(
y = np.log(flights_cleaned['ARRIVAL_DELAY'])# Take the log of ARRIVAL_DELAY
# Split the data into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

# Create a LabelEncoder instance
label_encoder = LabelEncoder()

# Apply label encoding to the "AIRLINE" column
x_train['AIRLINE_encoded'] = label_encoder.fit_transform(x_train['AIRLINE'])
x_test['AIRLINE_encoded'] = label_encoder.transform(x_test['AIRLINE'])

# Drop the original "AIRLINE" column
x_train = x_train.drop(columns=['AIRLINE'])
x_test = x_test.drop(columns=['AIRLINE'])
```

In [30]:

```python
import statsmodels.api as sm

model = sm.OLS(y_train, x_train).fit()

# Get p-values for each predictor
p_values = model.pvalues

# Choose a significance level (e.g., 0.05)
significance_level = 0.05

# Identify non-significant predictors
non_significant_predictors = list(x_train.columns[p_values > significance_level])

non_significant_predictors
```

Out[30]:

```
['WEATHER_DELAY']
```

In [31]:
```
model.summary()
```

Out[31]:

OLS Regression Results

Dep. Variable:	ARRIVAL_DELAY	**R-squared (uncentered):**	0.945
Model:	OLS	**Adj. R-squared (uncentered):**	0.944
Method:	Least Squares	**F-statistic:**	1677.
Date:	Mon, 25 Sep 2023	**Prob (F-statistic):**	0.00
Time:	21:45:40	**Log-Likelihood:**	-993.28
No. Observations:	788	**AIC:**	2003.
Df Residuals:	780	**BIC:**	2040.
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
LATE_AIRCRAFT_DELAY	0.0071	0.002	4.506	0.000	0.004	0.010
AIR_SYSTEM_DELAY	0.0275	0.002	17.639	0.000	0.024	0.031
DEPARTURE_DELAY	0.0210	0.001	18.550	0.000	0.019	0.023
WEATHER_DELAY	0.0042	0.003	1.338	0.181	-0.002	0.010
DAY_OF_WEEK	0.2499	0.013	19.740	0.000	0.225	0.275
DISTANCE	0.0006	4.52e-05	12.800	0.000	0.000	0.001
SECURITY_DELAY	0.0298	0.014	2.081	0.038	0.002	0.058
AIRLINE_encoded	0.0858	0.006	14.605	0.000	0.074	0.097

Omnibus:	30.355	**Durbin-Watson:**	1.899
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	35.033
Skew:	-0.430	**Prob(JB):**	2.47e-08
Kurtosis:	3.571	**Cond. No.**	490.

Notes:

[1] R^2 is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [32]:
```
# Get p-values for each predictor
p_values = model.pvalues
p_values
```

Out[32]:
```
LATE_AIRCRAFT_DELAY    7.632238e-06
AIR_SYSTEM_DELAY       7.422289e-59
DEPARTURE_DELAY        6.503314e-64
WEATHER_DELAY          1.814452e-01
DAY_OF_WEEK            1.158287e-70
DISTANCE               3.448801e-34
SECURITY_DELAY         3.773468e-02
AIRLINE_encoded        6.929894e-43
dtype: float64
```

In [36]:
```
# without weather Delay

x = x[['LATE_AIRCRAFT_DELAY', 'AIR_SYSTEM_DELAY', 'DEPARTURE_DELAY', 'DAY_OF_WEEK', 'DISTANCE','SECURITY_D
y = np.log(flights_cleaned['ARRIVAL_DELAY'])



x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)


# Create a LabelEncoder instance
label_encoder = LabelEncoder()

# Apply label encoding to the "AIRLINE" column
x_train['AIRLINE_encoded'] = label_encoder.fit_transform(x_train['AIRLINE'])
x_test['AIRLINE_encoded'] = label_encoder.transform(x_test['AIRLINE'])

# Drop the original "AIRLINE" column
x_train = x_train.drop(columns=['AIRLINE'])
x_test = x_test.drop(columns=['AIRLINE'])
```

Perform model diagnostics. Did anything improve?

In [39]:

```
import numpy as np
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Fit the model without the non-significant predictors
model1 = sm.OLS(y_train, x_train).fit()

# Make predictions on the test data using model1
y_pred = model1.predict(x_test)

# Calculate Mean Absolute Error (MAE)
mae = mean_absolute_error(y_test, y_pred)

# Calculate Mean Squared Error (MSE)
mse = mean_squared_error(y_test, y_pred)

# Calculate Root Mean Squared Error (RMSE)
rmse = np.sqrt(mse)

# Calculate R-squared (R2)
r2 = r2_score(y_test, y_pred)

# Print the evaluation metrics
print("Mean Absolute Error (MAE):", mae)
print("Mean Squared Error (MSE):", mse)
print("Root Mean Squared Error (RMSE):", rmse)
print("R-squared (R2):", r2)
```

```
Mean Absolute Error (MAE): 0.6442091500307388
Mean Squared Error (MSE): 0.6434913009363038
Root Mean Squared Error (RMSE): 0.8021790952999859
R-squared (R2): -1.0120163592825202
```

1. **Mean Absolute Error (MAE):** 0.6442
    - The MAE represents the average absolute difference between the actual and predicted values. In your case, it's approximately 0.6442. This means, on average, your model's predictions deviate from the actual values by about 0.6442 units. Lower MAE values indicate better predictive accuracy.
2. **Mean Squared Error (MSE):** 0.6435

- The MSE is another measure of prediction accuracy, but it squares the differences between actual and predicted values. Here, your MSE is approximately 0.6435. Like MAE, lower MSE values are desirable.

3. **Root Mean Squared Error (RMSE):** 0.8022

   - The RMSE is the square root of the MSE and is expressed in the same units as the target variable. In your case, it's approximately 0.8022. The RMSE gives you an estimate of the average magnitude of error in your predictions. Again, lower RMSE values indicate better model performance.

4. **R-squared (R2):** -1.0120

   - The R-squared value measures the proportion of the variance in the dependent variable (target) that is predictable from the independent variables (features) in your model. An R-squared value can range from 0 to 1, where 0 indicates that the model does not explain any of the variance, and 1 indicates a perfect fit. However, in your case, you have a negative R-squared value (-1.0120), which is unusual.

Provide interpretations to a few of the coefficients. Do you think they make sense?

1. LATE_AIRCRAFT_DELAY: The coefficient for LATE_AIRCRAFT_DELAY represents the change in the predicted ARRIVAL_DELAY for a one-unit increase in this variable, holding all other variables constant. If the coefficient is positive, it suggests that as the delay due to late aircraft increases, the predicted arrival delay also increases. This makes sense because late aircraft can lead to delayed flights.
2. AIR_SYSTEM_DELAY: Similar to LATE_AIRCRAFT_DELAY, a positive coefficient for AIR_SYSTEM_DELAY indicates that as the delay due to air system issues increases, the predicted arrival delay also increases. This is logical as problems with the air system can lead to flight delays.
3. DEPARTURE_DELAY: The coefficient for DEPARTURE_DELAY indicates the change in the predicted ARRIVAL_DELAY for a one-unit increase in departure delay, holding other variables constant. A positive coefficient here suggests that as the departure delay increases, the predicted arrival delay also increases. This is intuitive since a longer departure delay often results in a longer overall delay.
4. DAY_OF_WEEK: The coefficient for DAY_OF_WEEK represents how the day of the week affects ARRIVAL_DELAY. This coefficient measures the average change in ARRIVAL_DELAY for a one-day increase in the day of the week, while keeping other factors constant. For example, if the coefficient is negative, it might suggest that flights on certain days (e.g., weekends) tend to have shorter arrival delays compared to weekdays.
5. DISTANCE: The coefficient for DISTANCE represents the change in the predicted ARRIVAL_DELAY for a one-unit increase in the distance of the flight. If the coefficient is positive, it suggests that longer-distance flights are associated with longer arrival delays, assuming other factors are constant. This is reasonable since longer flights may have more opportunities for delays to

accumulate.

6. AIRLINE: The coefficients associated with different airlines indicate how each airline affects ARRIVAL_DELAY compared to a reference airline. Positive coefficients suggest that, on average, flights from those airlines tend to have longer arrival delays compared to the reference airline, while negative coefficients suggest shorter delays. This helps in understanding which airlines may have a better or worse track record in terms of arrival delays.

Obviously there's still a lot that needs to be done. Provide a few suggestions on how we can further improve the model fit (you don't need to implement them).

1. **Feature Engineering**:
    - Consider introducing interaction terms between predictor variables if there's reason to suspect that the combination of two factors has a nonlinear impact on the target variable. For example, an interaction term involving DEPARTURE_DELAY and DISTANCE could capture the effects of delays on longer flights differently.
    - Explore the creation of polynomial features for predictor variables that may not exhibit a linear relationship with the target variable. This allows the model to capture nonlinear patterns more effectively.
2. **Outlier Detection and Handling**:
    - Detect and address outliers within the dataset, as they can significantly influence model fit. Potential strategies include winsorization, removal, or transformation of extreme outliers.
3. **Feature Selection**:
    - Utilize feature selection methods to pinpoint and retain only the most pertinent predictor variables. Features that contribute minimally to explaining the target variable might be eliminated to simplify the model.
4. **Regularization**:
    - Implement regularization techniques like L1 (Lasso) or L2 (Ridge) regularization to mitigate overfitting and enhance the model's ability to generalize.
5. **Model Selection**:
    - Experiment with various regression models beyond linear regression. For instance, delve into decision trees, random forests, gradient boosting, or neural networks. Ensemble methods frequently offer superior predictive capabilities.
6. **Cross-Validation**:

- Employ cross-validation techniques such as k-fold cross-validation to rigorously evaluate the model's performance and estimate its generalizability to new data.

7. **Residual Analysis**:
   - Scrutinize the residuals (the disparities between predicted and actual values) to uncover any discernible patterns or heteroscedasticity (fluctuating variance) in the errors. This can shed light on aspects of the data that the model might not be capturing.
8. **Data Transformation**:
   - Apply transformations to either the target or predictor variables, such as logarithmic transformations or scaling, when confronted with non-normality or heteroscedasticity in the data.
9. **More Data**:
   - Expanding the dataset's size can often lead to improvements in model performance, especially when the dataset is limited in size or unrepresentative of the broader population.
10. **Feature Scaling**:
    - Ensure that all predictor variables are standardized to a consistent scale, particularly if regularization is being used. Standardization or normalization of features can be beneficial.
11. **Feature Engineering Based on Domain Knowledge**:
    - Consult domain experts to gain insights into which variables might carry greater significance or which transformations align with the unique characteristics of the airline industry.
12. **Hyperparameter Tuning**:
    - If complex models like random forests or neural networks are employed, undertake hyperparameter tuning to optimize model parameters and enhance performance.
13. **Time Series Analysis**:
    - If the dataset features time-series data, contemplate the application of time series forecasting models such as ARIMA or Prophet, which are tailored for temporal data.
14. **Feature Validation and Updating**:

In [ ]: