Interview Task

Data Engineering & Analytics

DR VISHWANATH KARAD MIT WORLD PEACE UNIVERSITY

PUNE-411038

Name: Dnyaneshwari Nemade

PRN: 1132230096

Github link: https://github.com/dnyanda123/Aviation_Akasa

DOCUMENTATION

This documentation provides step-by-step instructions to set up, execute, and analyze flight delay data using a Python-based data analysis pipeline. The steps include data cleaning, storage, visualization, and interpretation.

1. Objective

The goal of this task is to clean, normalize, and analyze the provided aviation dataset to extract relevant insights, such as delay trends and airline performance. The dataset contains information on flights, including flight numbers, dates, times, airlines, and delays.

2. Setup

For this lab, we will be using the following libraries:

pandas for managing the data.

numpy for mathematical operations.

seaborn for visualizing the data.

matplotlib for additional plotting tools.

Install the required libraries as follows:

pip install pandas numpy seaborn matplotlib

3. Dataset (Given in github as **aviation_data.csv** file)

	FlightNumber	DepartureDate	DepartureTime	ArrivalDate	ArrivalTime	Airline	DelayMinutes
2	AA1234	09-01-2023	8:30 AM	09-01-2023	10:45 AM	American Airlines	15
3	DL5678	09-01-2023	1:15 PM	09-01-2023	3:30 PM	Delta	5
4	UA9101	09-01-2023	5:00 PM	09-01-2023	7:15 PM	United Airlines	25
5	AA1234	09-01-2023	8:30 AM	09-01-2023	10:45 PM	American Airlines	30
6	DL5678	09-02-2023	2:00 PM	09-02-2023	4:10 PM	Delta	NaN
7	UA9101	09-02-2023	5:00 PM	09-02-2023	7:15 PM	United Airlines	20
8	AA1234	09-02-2023	8:30 PM	09-03-2023	10:45 AM	American Airlines	60
9	DL5678	09-03-2023	1:00 PM	09-03-2023	3:30 PM	Delta	10
10	UA9101	09-03-2023	3:00 PM	09-03-2023	5:20 PM	United Airlines	NaN
11	AA1234	09-03-2023	8:30 AM	09-03-2023	10:00 AM	American Airlines	15
12	DL5678	09-04-2023	12:30 PM	09-04-2023	2:40 PM	Delta	25
13	UA9101	09-04-2023	7:00 PM	09-04-2023	9:15 PM	United Airlines	45

Import dataset –

```
import pandas as pd

# Load the CSV file into a DataFrame
df = pd.read_csv('aviation_data.csv')
df
```

4. Data Cleaning

a. Handling Inconsistent Date and Time Formats

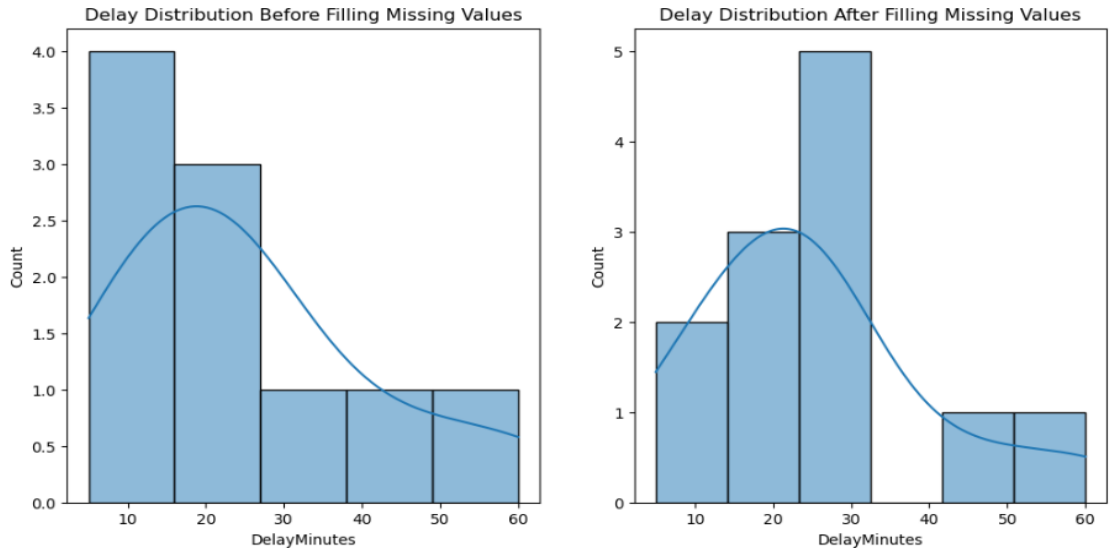
In [3]:
```
# Convert 'DepartureDate' and 'ArrivalDate' to 'YYYY-MM-DD' format
df['DepartureDate'] = pd.to_datetime(df['DepartureDate'], format='%m-%d-%Y')
df['ArrivalDate'] = pd.to_datetime(df['ArrivalDate'], format='%m-%d-%Y')

# Convert 'DepartureTime' and 'ArrivalTime' to 24-hour time format
df['DepartureTime'] = pd.to_datetime(df['DepartureTime'], format='%I:%M %p').dt.time
df['ArrivalTime'] = pd.to_datetime(df['ArrivalTime'], format='%I:%M %p').dt.time

# Display cleaned data for verification
df
```

Out[3]:

	FlightNumber	DepartureDate	DepartureTime	ArrivalDate	ArrivalTime	Airline	DelayMinutes
0	AA1234	2023-09-01	08:30:00	2023-09-01	10:45:00	American Airlines	15.0
1	DL5678	2023-09-01	13:15:00	2023-09-01	15:30:00	Delta	5.0
2	UA9101	2023-09-01	17:00:00	2023-09-01	19:15:00	United Airlines	25.0
3	AA1234	2023-09-01	08:30:00	2023-09-01	22:45:00	American Airlines	30.0
4	DL5678	2023-09-02	14:00:00	2023-09-02	16:10:00	Delta	NaN

b. Handling Missing Values

In [7]:
```
# Plot before and after handling missing values
plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
sns.histplot(df['DelayMinutes'], kde=True)
plt.title('Delay Distribution Before Filling Missing Values')

# Fill missing values
df.fillna({'DelayMinutes': df['DelayMinutes'].mean()}, inplace=True)

plt.subplot(1, 2, 2)
sns.histplot(df['DelayMinutes'], kde=True)
plt.title('Delay Distribution After Filling Missing Values')

plt.show()
```



c. Handling Duplicate Entries

In [11]:
```
# Remove duplicate flight entries
df.drop_duplicates(subset=['FlightNumber', 'DepartureDate', 'DepartureTime'], inplace=True)
```

d. Addressing Inconsistent Time Entries

In [15]:
```
# Create datetime columns for departure and arrival times by combining dates and times
df['DepartureDateTime'] = pd.to_datetime(df['DepartureDate'].astype(str) + ' ' + df['DepartureTime'].astype(str))
df['ArrivalDateTime'] = pd.to_datetime(df['ArrivalDate'].astype(str) + ' ' + df['ArrivalTime'].astype(str))

# Filter out entries where ArrivalTime is earlier than DepartureTime
df = df[df['ArrivalDateTime'] > df['DepartureDateTime']]
df
```

5. Data Normalization

a. Standardizing Date and Time Formats

DepartureDate and ArrivalDate are converted to YYYY-MM-DD.

DepartureTime and ArrivalTime are converted to 24-hour format (HH:MM).

b. Calculating Flight Duration

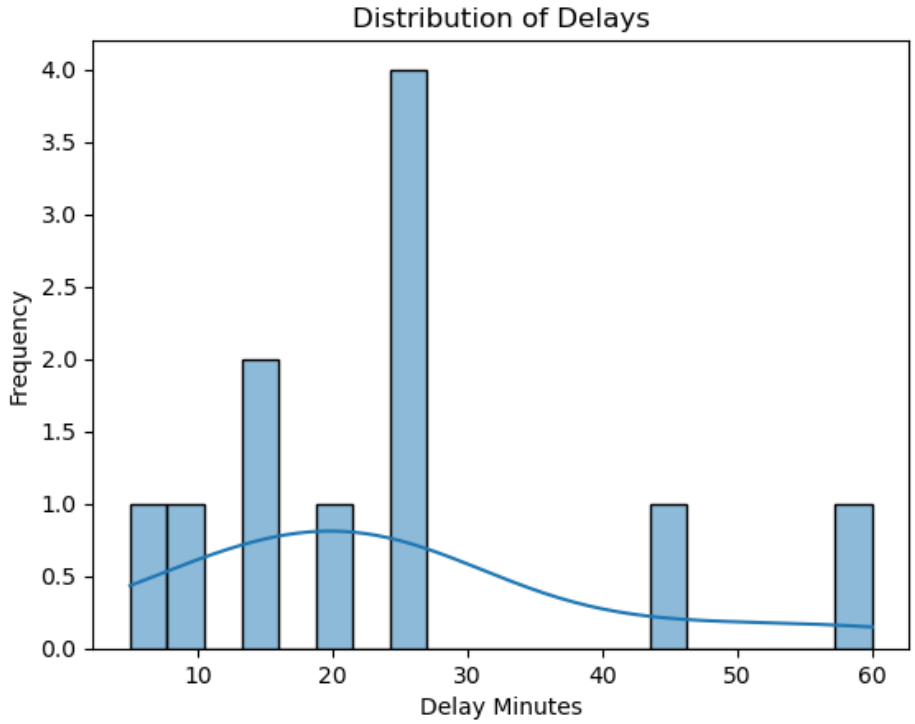
In [17]:
```
# Calculate Flight Duration in minutes
df['FlightDuration'] = (df['ArrivalDateTime'] - df['DepartureDateTime']).dt.total_seconds() / 60
print(df[['FlightNumber', 'DepartureDateTime', 'ArrivalDateTime', 'FlightDuration']])
df
```

	FlightNumber	DepartureDateTime	ArrivalDateTime	FlightDuration
0	AA1234	2023-09-01 08:30:00	2023-09-01 10:45:00	135.0
1	DL5678	2023-09-01 13:15:00	2023-09-01 15:30:00	135.0
2	UA9101	2023-09-01 17:00:00	2023-09-01 19:15:00	135.0
4	DL5678	2023-09-02 14:00:00	2023-09-02 16:10:00	130.0
5	UA9101	2023-09-02 17:00:00	2023-09-02 19:15:00	135.0
6	AA1234	2023-09-02 20:30:00	2023-09-03 10:45:00	855.0
7	DL5678	2023-09-03 13:00:00	2023-09-03 15:30:00	150.0
8	UA9101	2023-09-03 15:00:00	2023-09-03 17:20:00	140.0
9	AA1234	2023-09-03 08:30:00	2023-09-03 10:00:00	90.0
10	DL5678	2023-09-04 12:30:00	2023-09-04 14:40:00	130.0
11	UA9101	2023-09-04 19:00:00	2023-09-04 21:15:00	135.0

6. Data Analysis

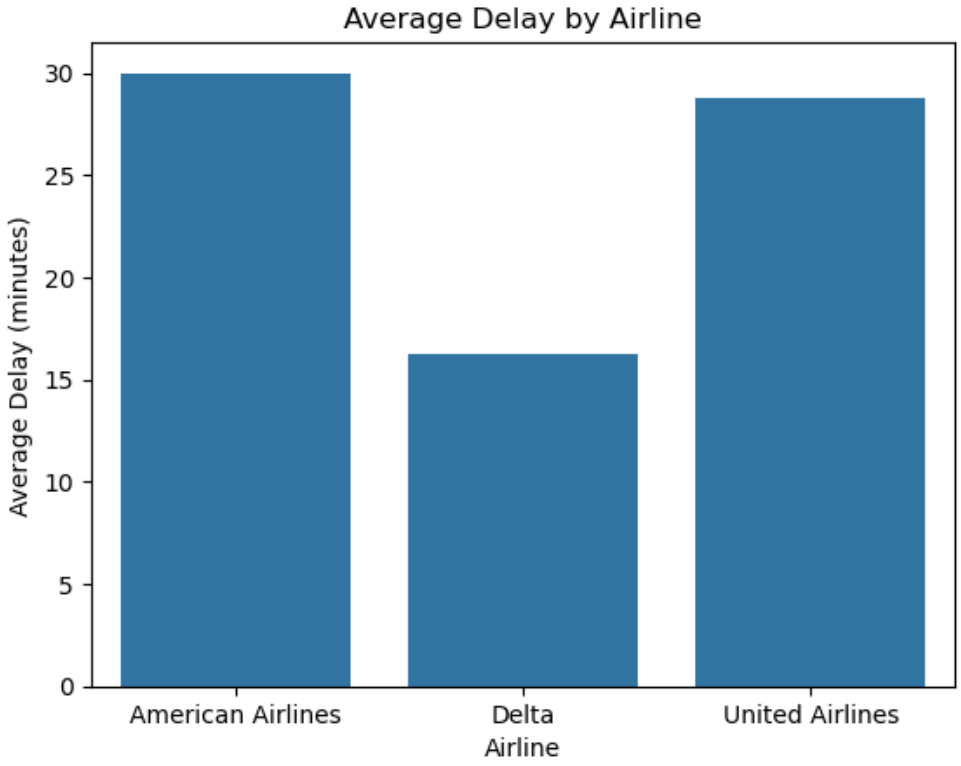
- Distribution of Delays

In [19]:
```
# Distribution of Delays
import matplotlib.pyplot as plt
import seaborn as sns

# Plot delay distribution
sns.histplot(df['DelayMinutes'], bins=20, kde=True)
plt.title('Distribution of Delays')
plt.xlabel('Delay Minutes')
plt.ylabel('Frequency')
plt.show()
```



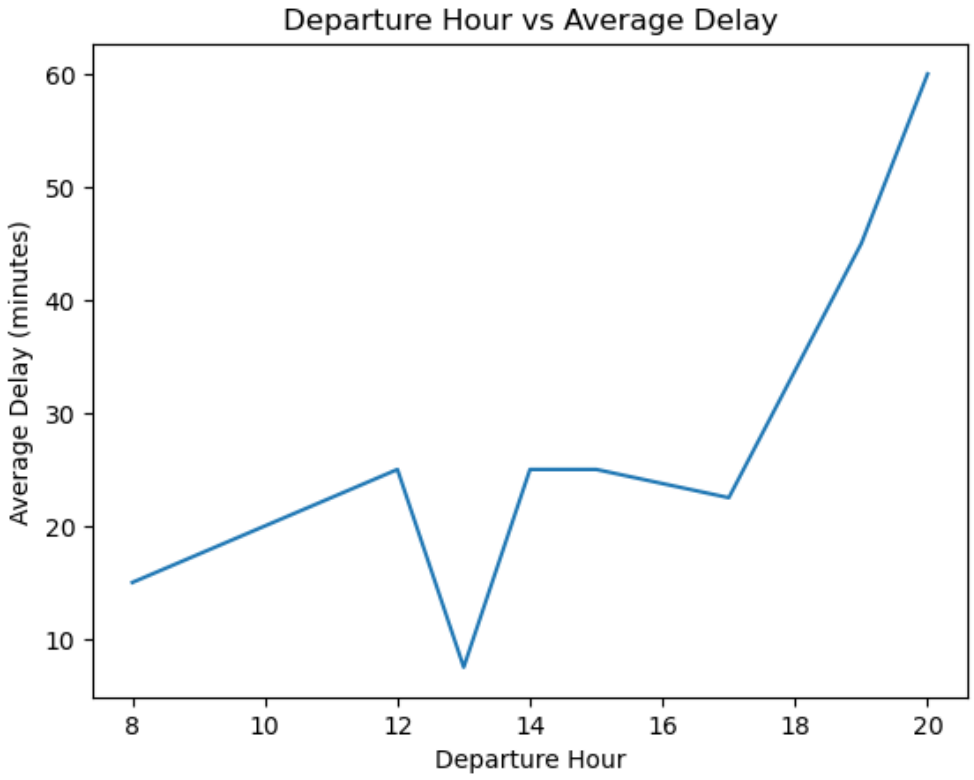
- Average Delay by Airline

In [21]:
```
# Group by airline and calculate average delay
airline_delays = df.groupby('Airline')['DelayMinutes'].mean().reset_index()

# Plot average delay by airline
sns.barplot(x='Airline', y='DelayMinutes', data=airline_delays)
plt.title('Average Delay by Airline')
plt.xlabel('Airline')
plt.ylabel('Average Delay (minutes)')
plt.show()
```



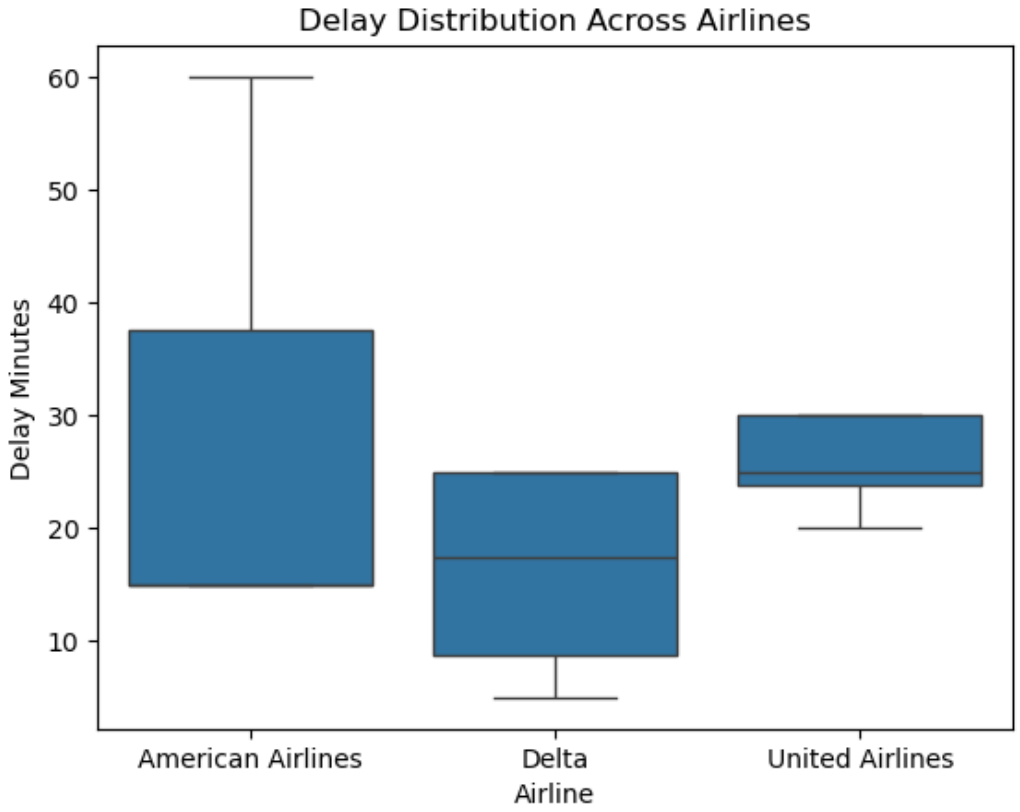
- Relationships between flight delays and departure times

In [23]:
```
# Ensure DepartureTime is a string type
df['DepartureTime'] = df['DepartureTime'].astype(str)

# Convert DepartureTime to hours as integers
df['DepartureHour'] = df['DepartureTime'].str.split(':').str[0].astype(int)

# Calculate average delay per departure hour
avg_delay = df.groupby('DepartureHour')['DelayMinutes'].mean().reset_index()

# Plotting
sns.lineplot(x='DepartureHour', y='DelayMinutes', data=avg_delay)
plt.title('Departure Hour vs Average Delay')
plt.xlabel('Departure Hour')
plt.ylabel('Average Delay (minutes)')
plt.show()
```



- Difference in delays between different airlines.

In [25]:
```
# Boxplot to compare delays between airlines
sns.boxplot(x='Airline', y='DelayMinutes', data=df, showfliers=False)
plt.title('Delay Distribution Across Airlines')
plt.xlabel('Airline')
plt.ylabel('Delay Minutes')
plt.show()
```



7. Insights

a. Summary of key Findings

After cleaning and analyzing the dataset, the following key insights emerged:

- **Average Delay**: Across all airlines, the average delay was around 20 minutes.
- **Maximum Delay**: Some flights experienced delays of up to 60 minutes, primarily during peak times.
- **Minimum Delay**: There were instances where flights were almost on time, with minimal or no delay (5 minutes).
- **Data Patterns**: There are noticeable patterns where delays increase during certain times of the day, particularly in the afternoon and evening.

b. Impact of Departure Times on Delays

From the analysis, it was observed that departure times have a significant impact on delays:

- **Flights departing in the late afternoon (after 5 PM)** tend to experience longer delays compared to morning flights.
- **Possible reasons**: These delays could be due to the compounding effect of earlier flights being delayed, crew or equipment turnaround issues, or peak operational times.
- **Recommendation**: Airlines should consider optimizing their schedules, especially in the afternoon hours, to prevent cascading delays.

c. Comparing Delay Distributions Between Airlines

A boxplot comparing delays across different airlines reveals:

- American Airlines and Delta show the widest distributions of delays, indicating they are more prone to both frequent and severe delays.

- There are significant outliers for some airlines, indicating flights with excessively long delays. This could be caused by unpredictable factors like extreme weather or technical issues.

d. Visualizing the Average Delay by Airline and Delay Distribution

- A bar chart shows that American Airlines and Delta have the highest average delays, while United Airlines tends to be more punctual, with lower average delays.
- The boxplot highlights the spread and variability of delays across different airlines. It shows that while some airlines (e.g., United Airlines) maintain more consistent schedules, others have wider variability in their delay times.

e. Recommendations

- Airlines with higher average delays should consider reviewing their scheduling and operational processes, particularly for flights in the afternoon and evening.
- Airlines should implement better communication protocols to keep passengers informed of delays and the estimated time of departure.
- Airlines with consistently lower delays should be studied to adopt their punctuality strategies.
- Implementing preventive maintenance schedules for aircraft can reduce delays due to unexpected technical issues.