# demo

September 27, 2024

```
[10]: import pandas as pd
      import numpy as np
      import warnings

      warnings.filterwarnings("ignore")
```

```
[11]: df = pd.read_csv("avaitaion-data.csv")

      df
```

```
[11]:    FlightNumber DepartureDate DepartureTime ArrivalDate ArrivalTime  \
      0        AA1234    09/01/2023      08:30 AM  09/01/2023    10:45 AM
      1        DL5678    09/01/2023      01:15 PM  09/01/2023    03:30 PM
      2        UA9101    09/01/2023      05:00 PM  09/01/2023    07:15 PM
      3        AA1234    09/01/2023      08:30 AM  09/01/2023    10:45 PM
      4        DL5678    09/02/2023      02:00 PM  09/02/2023    04:10 PM
      5        UA9101    09/02/2023      05:00 PM  09/02/2023    07:15 PM
      6        AA1234    09/02/2023      08:30 PM  09/03/2023    10:45 AM
      7        DL5678    09/03/2023      01:00 PM  09/03/2023    03:30 PM
      8        UA9101    09/03/2023      03:00 PM  09/03/2023    05:20 PM
      9        AA1234    09/03/2023      08:30 AM  09/03/2023    10:00 AM
      10       DL5678    09/04/2023      12:30 PM  09/04/2023    02:40 PM
      11       UA9101    09/04/2023      07:00 PM  09/04/2023    09:15 PM

                    Airline  DelayMinutes
      0   American Airlines          15.0
      1               Delta           5.0
      2     United Airlines          25.0
      3   American Airlines          30.0
      4               Delta           NaN
      5     United Airlines          20.0
      6   American Airlines          60.0
      7               Delta          10.0
      8     United Airlines           NaN
      9   American Airlines          15.0
      10              Delta          25.0
      11    United Airlines          45.0
```

```
[ ]: # create a mysql database and store the data in the database and write a query␣
      ↪to get the data from the database and store in database.
```

```
[12]: # Step 1: Handle missing values by filling with the mean delay (for simplicity).
      mean_delay = df["DelayMinutes"].mean()
      df["DelayMinutes"].fillna(mean_delay, inplace=True)

      # Step 2: Remove duplicate entries based on FlightNumber, DepartureDate, and␣
       ↪DepartureTime.
      df_cleaned = df.drop_duplicates(
          subset=["FlightNumber", "DepartureDate", "DepartureTime"], keep="first"
      )

      # Step 3: Convert dates to standard YYYY-MM-DD format and times to 24-hour␣
       ↪format
      df_cleaned["DepartureDate"] = pd.to_datetime(
          df_cleaned["DepartureDate"], format="%m/%d/%Y"
      ).dt.strftime("%Y-%m-%d")
      df_cleaned["ArrivalDate"] = pd.to_datetime(
          df_cleaned["ArrivalDate"], format="%m/%d/%Y"
      ).dt.strftime("%Y-%m-%d")
      df_cleaned["DepartureTime"] = pd.to_datetime(
          df_cleaned["DepartureTime"], format="%I:%M %p"
      ).dt.strftime("%H:%M")
      df_cleaned["ArrivalTime"] = pd.to_datetime(
          df_cleaned["ArrivalTime"], format="%I:%M %p"
      ).dt.strftime("%H:%M")

      # Display the cleaned data
      df_cleaned
```

```
[12]:    FlightNumber DepartureDate DepartureTime ArrivalDate ArrivalTime  \
      0        AA1234    2023-09-01         08:30  2023-09-01       10:45
      1        DL5678    2023-09-01         13:15  2023-09-01       15:30
      2        UA9101    2023-09-01         17:00  2023-09-01       19:15
      4        DL5678    2023-09-02         14:00  2023-09-02       16:10
      5        UA9101    2023-09-02         17:00  2023-09-02       19:15
      6        AA1234    2023-09-02         20:30  2023-09-03       10:45
      7        DL5678    2023-09-03         13:00  2023-09-03       15:30
      8        UA9101    2023-09-03         15:00  2023-09-03       17:20
      9        AA1234    2023-09-03         08:30  2023-09-03       10:00
      10       DL5678    2023-09-04         12:30  2023-09-04       14:40
      11       UA9101    2023-09-04         19:00  2023-09-04       21:15

                   Airline  DelayMinutes
      0   American Airlines          15.0
      1               Delta           5.0
```

```
2     United Airlines          25.0
4               Delta          25.0
5     United Airlines          20.0
6   American Airlines          60.0
7               Delta          10.0
8     United Airlines          25.0
9   American Airlines          15.0
10              Delta          25.0
11    United Airlines          45.0
```

[13]:

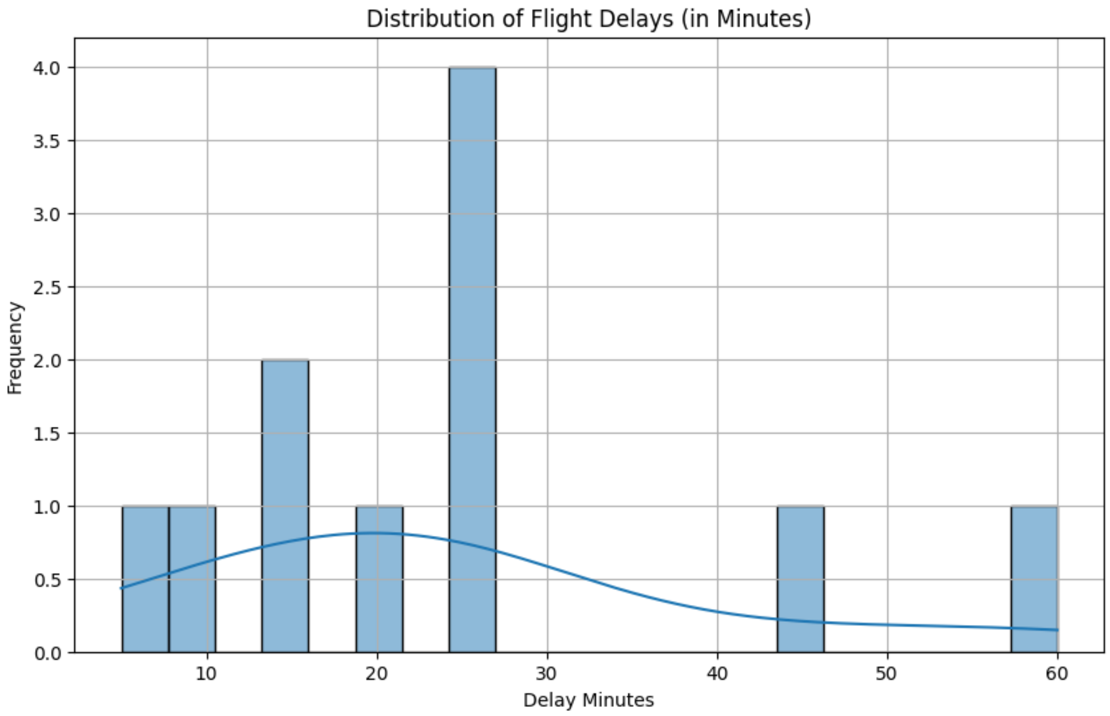
```
# Step 4: Handle inconsistent times by ensuring ArrivalTime is later than␣
↪DepartureTime.
# We will assume that any flight with an ArrivalTime earlier than DepartureTime␣
↪occurs the next day.

# Convert DepartureTime and ArrivalTime into timedelta for easy comparison
df_cleaned["DepartureDateTime"] = pd.to_datetime(
    df_cleaned["DepartureDate"] + " " + df_cleaned["DepartureTime"]
)
df_cleaned["ArrivalDateTime"] = pd.to_datetime(
    df_cleaned["ArrivalDate"] + " " + df_cleaned["ArrivalTime"]
)

# Adjust ArrivalDateTime if it occurs before DepartureDateTime (assume the␣
↪arrival is the next day)
df_cleaned.loc[
    df_cleaned["ArrivalDateTime"] < df_cleaned["DepartureDateTime"],␣
↪"ArrivalDateTime"
] += pd.Timedelta(days=1)

# Step 5: Create FlightDuration column by calculating the difference between␣
↪ArrivalDateTime and DepartureDateTime
df_cleaned["FlightDuration"] = (
    df_cleaned["ArrivalDateTime"] - df_cleaned["DepartureDateTime"]
)

# Display the updated dataset with FlightDuration
df_cleaned[
    [
        "FlightNumber",
        "DepartureDateTime",
        "ArrivalDateTime",
        "FlightDuration",
        "Airline",
        "DelayMinutes",
    ]
]
```

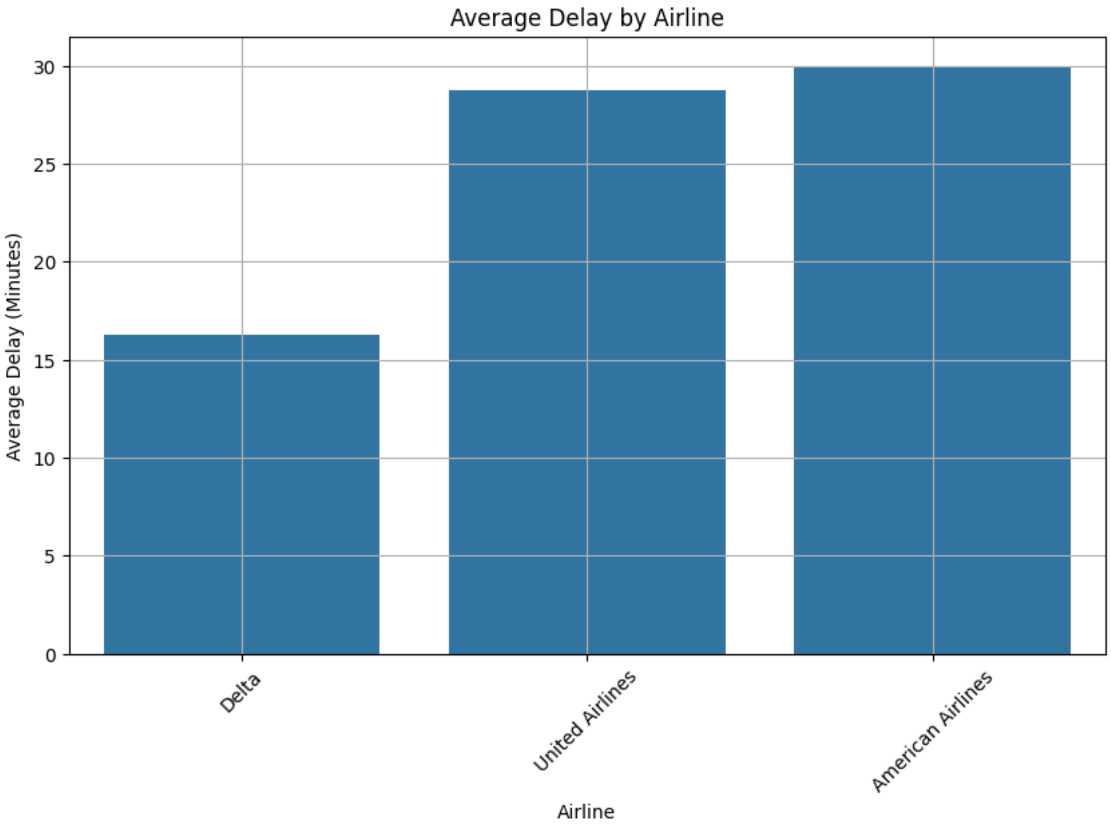
[13]:

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

              Airline  DelayMinutes
0   American Airlines          15.0
1               Delta           5.0
2     United Airlines          25.0
4               Delta          25.0
5     United Airlines          20.0
6   American Airlines          60.0
7               Delta          10.0
8     United Airlines          25.0
9   American Airlines          15.0
10              Delta          25.0
11    United Airlines          45.0
```

[14]:

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

# Step 1: Analyze the distribution of delays
plt.figure(figsize=(10, 6))
sns.histplot(df_cleaned["DelayMinutes"], bins=20, kde=True)
plt.title("Distribution of Flight Delays (in Minutes)")
plt.xlabel("Delay Minutes")
plt.ylabel("Frequency")
plt.grid(True)
plt.show()

# Step 2: Calculate the average delay per airline
avg_delay_per_airline = (
    df_cleaned.groupby("Airline")["DelayMinutes"].mean().sort_values()
)

# Step 3: Visualize average delay by airline
plt.figure(figsize=(10, 6))
sns.barplot(x=avg_delay_per_airline.index, y=avg_delay_per_airline.values)
plt.title("Average Delay by Airline")
plt.xlabel("Airline")
plt.ylabel("Average Delay (Minutes)")
plt.xticks(rotation=45)
plt.grid(True)
plt.show()

avg_delay_per_airline
```





```
[14]: Airline
      Delta                16.25
      United Airlines      28.75
      American Airlines    30.00
      Name: DelayMinutes, dtype: float64
```