ENGM4620 Project #2:

Data Loading and Manipulation in Python

- Abdulla Sadoun B00900541
- Abdul Hameed Al Busaid B00832820
- Dataset used: Sri Harsha Eedala, Flight Delay Data, 2013-2023(August)
- https://www.kaggle.com/datasets/sriharshaeedala/airline-delay/data

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Q1) Dataset Selection and Loading

- Number of rows (samples) = 171666
- Number of columns (features) = 21
- The file format for the dataset used is a comma seperated value file where each coloumn is seperated by a comma "," and each row is in a new line "\n"
- The headers were included from the author as the first row and I havent included them as a row in the value above
- We have included the link where the dataset was obtained from (kaggle) and the file data was read using the read_csv method from pandas

Dataset Description

The dataset used provides detailed information on flight arrival delays for US airports from the beginning of the year 2013 to august of 2023 when it was last updated. The data focuses primarily on delays and arrivals of flights in the given period, and includes information like the date, carrier, airport, #of arriving flights , # of flights delayed by 15mins+, and counters like carrier_ct, weather_ct, nas_ct, security_ct, late_aircraft_ct(previous trip was delayed); these counters represent the amount of delays for that feature for eg. weather_ct is the counter for delays that occured due to the weather etc. The other features like weather_delay and all the other ones that have "_delay" subsequent to the label are for the amount of time it was delayed for that reason and there is also a flight cancelled counter as a feature.

```
# Reading the .csv which has been downloaded and uploaded to content
in colab

#df = pd.read_csv('/content/filtered_data_2020_to_2023.csv')
```

```python
#df = pd.read_csv('/filtered_data_2020_to_2023.csv')
#unfiltereddf = pd.read_csv('/Airline_Delay_Cause.csv')
df = pd.read_csv('/content/Airline_Delay_Cause.csv')

df.head()
```

{"type":"dataframe","variable_name":"df"}

The dataset I'm working with is too large and is causing issues in colab as it's occupying max memory. The data will be reduced to only delays from 2020 until 2022 since in 2023 the year is not complete and data only goes up to August which might be unfair due to missing out on the fall/winter months that may introduce weather delays.

```python
# filtering to get the range we're working with (2020-2022)
df = df[df['year'].between(2020, 2022)]

df.to_csv('filtered_data_2020_to_2023.csv', index=False)

df.head()
```

{"type":"dataframe","variable_name":"df"}

we will take a look at the type of data in the features (columns)

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 59158 entries, 12373 to 71530
Data columns (total 21 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   year                 59158 non-null  int64  
 1   month                59158 non-null  int64  
 2   carrier              59158 non-null  object 
 3   carrier_name         59158 non-null  object 
 4   airport              59158 non-null  object 
 5   airport_name         59158 non-null  object 
 6   arr_flights          59039 non-null  float64
 7   arr_del15            58857 non-null  float64
 8   carrier_ct           59039 non-null  float64
 9   weather_ct           59039 non-null  float64
 10  nas_ct               59039 non-null  float64
 11  security_ct          59039 non-null  float64
 12  late_aircraft_ct     59039 non-null  float64
 13  arr_cancelled        59039 non-null  float64
 14  arr_diverted         59039 non-null  float64
 15  arr_delay            59039 non-null  float64
 16  carrier_delay        59039 non-null  float64
 17  weather_delay        59039 non-null  float64
 18  nas_delay            59039 non-null  float64
```

```
 19  security_delay       59039 non-null  float64
 20  late_aircraft_delay  59039 non-null  float64
dtypes: float64(15), int64(2), object(4)
memory usage: 9.9+ MB
```

Now after visualizing the data we can see that we do not have NAN values in our features to remove but we can remove some unnecessary features that are taking up too much space in memory and are repeated like the carrier and airport_name.

```
# drop the carrier, airport name coloumn as they are not useful and repeated
df.drop(columns=['carrier','airport_name'],inplace=True)

df.head()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 59158,\n  \"fields\": [\n    {\n      \"column\": \"year\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 2020,\n        \"max\": 2022,\n        \"num_unique_values\": 3,\n        \"samples\": [\n          2022,\n          2021,\n          2020\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"month\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 3,\n        \"min\": 1,\n        \"max\": 12,\n        \"num_unique_values\": 12,\n        \"samples\": [\n          2,\n          3,\n          12\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"carrier_name\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 18,\n        \"samples\": [\n          \"Endeavor Air Inc.\",\n          \"American Airlines Inc.\",\n          \"Envoy Air\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"airport\",\n      \"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\": 379,\n        \"samples\": [\n          \"FOD\",\n          \"DRT\",\n          \"SPS\"\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"arr_flights\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 836.1357053413994,\n        \"min\": 1.0,\n        \"max\": 20669.0,\n        \"num_unique_values\": 3246,\n        \"samples\": [\n          3755.0,\n          5469.0,\n          1045.0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"arr_del15\",\n      \"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 143.11444389543945,\n        \"min\": 0.0,\n        \"max\": 3479.0,\n        \"num_unique_values\": 1094,\n        \"samples\": [\n          650.0,\n          83.0,\n          87.0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"carrier_ct\",\n
```

```
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
49.94125032708729,\n        \"min\": 0.0,\n        \"max\": 1147.0,\n
\"num_unique_values\": 7965,\n        \"samples\": [\n          9.66,\
n          111.9,\n          66.81\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"weather_ct\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
7.203639145591121,\n        \"min\": 0.0,\n        \"max\": 226.0,\n
\"num_unique_values\": 2194,\n        \"samples\": [\n          17.0,\
n          13.3,\n          3.7\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"nas_ct\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
42.66862358910194,\n        \"min\": 0.0,\n        \"max\": 1391.74,\n
\"num_unique_values\": 6627,\n        \"samples\": [\n
81.08,\n          25.85,\n          5.78\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"security_ct\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.872336661695932,\n        \"min\": 0.0,\n        \"max\": 58.69,\n
\"num_unique_values\": 684,\n        \"samples\": [\n          7.95,\n
2.8,\n          1.6\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"late_aircraft_ct\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 54.64512822749818,\n        \"min\":
0.0,\n        \"max\": 1537.66,\n        \"num_unique_values\": 7314,\
n        \"samples\": [\n          206.16,\n          32.73,\n
79.72\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"arr_cancelled\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 63.970616956375295,\n        \"min\":
0.0,\n        \"max\": 4951.0,\n        \"num_unique_values\": 485,\n
\"samples\": [\n          2527.0,\n          1926.0,\n          149.0\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"arr_diverted\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 2.9331446797475955,\n        \"min\":
0.0,\n        \"max\": 154.0,\n        \"num_unique_values\": 74,\n
\"samples\": [\n          7.0,\n          80.0,\n          16.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"arr_delay\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
10382.654894933989,\n        \"min\": 0.0,\n        \"max\":
323449.0,\n        \"num_unique_values\": 10503,\n        \"samples\":
[\n          1242.0,\n          25534.0,\n          37199.0\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"carrier_delay\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 4182.408351828419,\n        \"min\":
```

```
0.0,\n        \"max\": 119425.0,\n        \"num_unique_values\":
6895,\n        \"samples\": [\n          5663.0,\n          12581.0,\n
2050.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"weather_delay\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 819.7129080872531,\n        \"min\":
0.0,\n        \"max\": 27876.0,\n        \"num_unique_values\": 2597,\
n        \"samples\": [\n          230.0,\n          1061.0,\n
998.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"nas_delay\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 2147.3590144133286,\n        \"min\":
0.0,\n        \"max\": 84155.0,\n        \"num_unique_values\": 4495,\
n        \"samples\": [\n          3906.0,\n          2645.0,\n
13989.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"security_delay\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 49.06785179594043,\n        \"min\":
0.0,\n        \"max\": 3760.0,\n        \"num_unique_values\": 453,\n
\"samples\": [\n          171.0,\n          26.0,\n          857.0\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"late_aircraft_delay\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
4153.203036892378,\n        \"min\": 0.0,\n        \"max\": 158653.0,\
n        \"num_unique_values\": 6470,\n        \"samples\": [\n
1305.0,\n          1783.0,\n          12812.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```


- no NaN coloumns
- no constant value coloumns
- dropped 2023 rows as its incomplete (up to august)

Q2) Data Exploration

- In the first part, the mean, median, standard deviation and minimum values are calculated for the quantitative features that contain values
- Other features like names of airport/carrier, and date will be excluded from this part.

```
for i in range(4,19):
    print(f'---Feature {i} ({df.iloc[:,i].name}), Summary
Statistics---')
    print(f'mean = {np.mean(df.iloc[:,i])}, median =
{np.median(df.iloc[:,i])}')
    print(f'standard deviation = {np.std(df.iloc[:,i])}')
    print(f'min. value = {np.min(df.iloc[:,i])}, max. value =
{np.max(df.iloc[:,i])}\n')
```

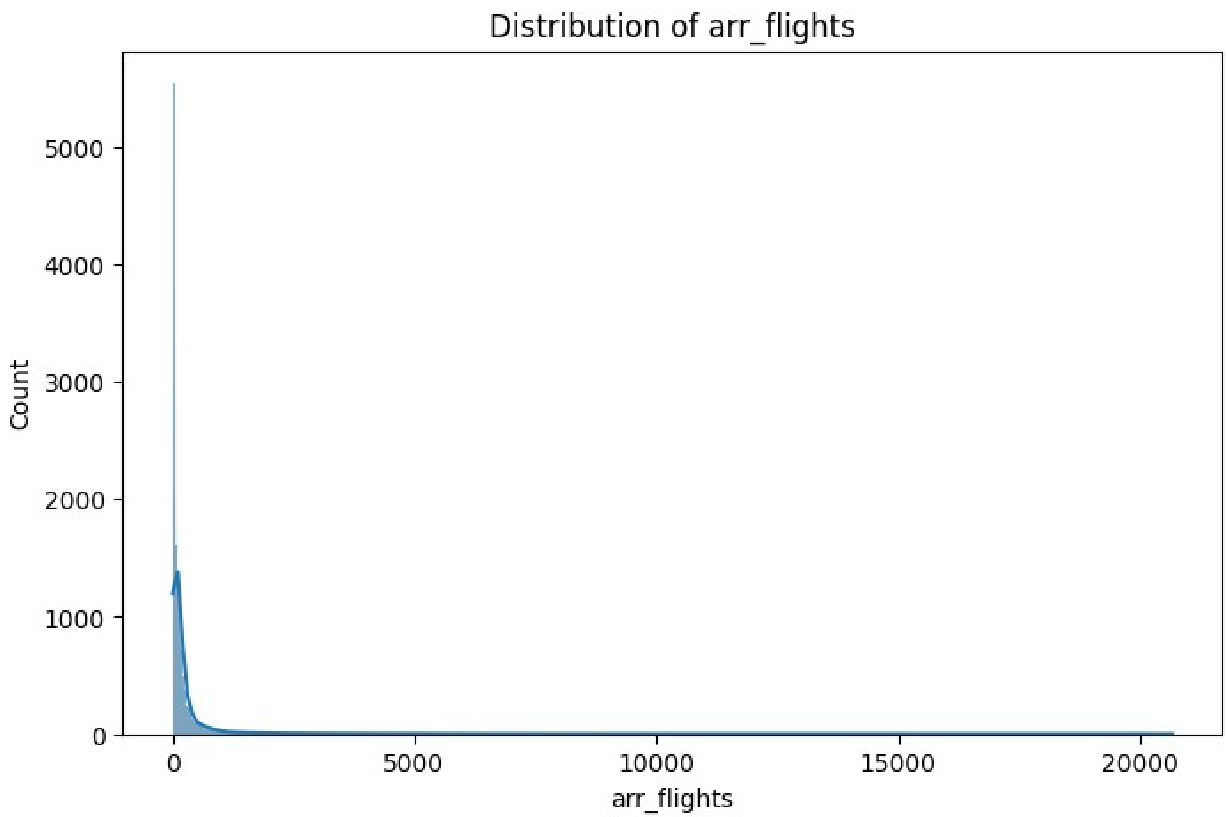
```
---Feature 4 (arr_flights), Summary Statistics---
mean = 294.93853215671, median = nan
standard deviation = 836.128624096409
min. value = 1.0, max. value = 20669.0

---Feature 5 (arr_del15), Summary Statistics---
mean = 47.89661382673259, median = nan
standard deviation = 143.11322810928152
min. value = 0.0, max. value = 3479.0

---Feature 6 (carrier_ct), Summary Statistics---
mean = 18.11036264164366, median = nan
standard deviation = 49.940827373939335
min. value = 0.0, max. value = 1147.0

---Feature 7 (weather_ct), Summary Statistics---
mean = 1.935674723487864, median = nan
standard deviation = 7.203578137870381
min. value = 0.0, max. value = 226.0

---Feature 8 (nas_ct), Summary Statistics---
mean = 12.224858991514083, median = nan
standard deviation = 42.668262227931606
min. value = 0.0, max. value = 1391.74

---Feature 9 (security_ct), Summary Statistics---
mean = 0.17714104236182865, median = nan
standard deviation = 0.8723292738645356
min. value = 0.0, max. value = 58.69

---Feature 10 (late_aircraft_ct), Summary Statistics---
mean = 15.3009698673758, median = nan
standard deviation = 54.64466543714246
min. value = 0.0, max. value = 1537.66

---Feature 11 (arr_cancelled), Summary Statistics---
mean = 9.576381713782416, median = nan
standard deviation = 63.970075188325005
min. value = 0.0, max. value = 4951.0

---Feature 12 (arr_diverted), Summary Statistics---
mean = 0.6383407578041633, median = nan
standard deviation = 2.9331198389042528
min. value = 0.0, max. value = 154.0

---Feature 13 (arr_delay), Summary Statistics---
mean = 3153.143345923881, median = nan
standard deviation = 10382.566964084375
min. value = 0.0, max. value = 323449.0

---Feature 14 (carrier_delay), Summary Statistics---
```

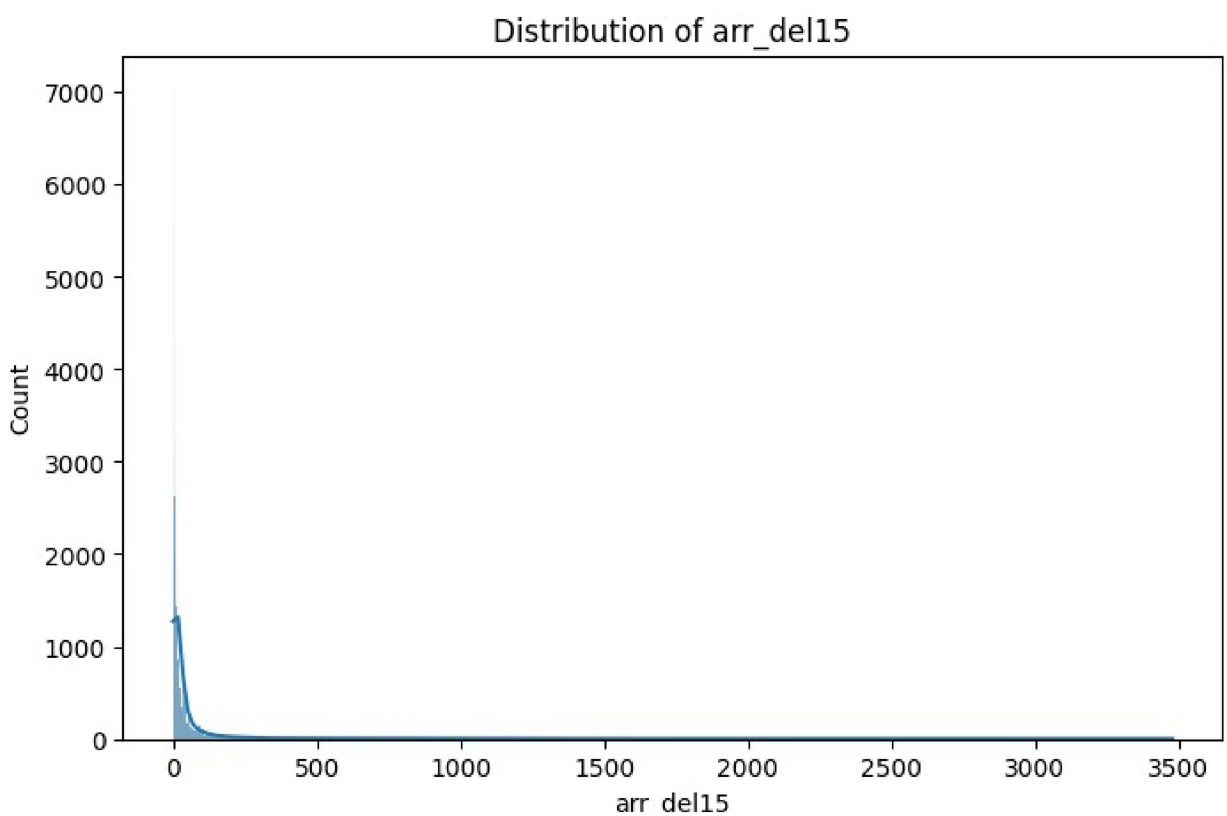
```
mean = 1275.6164738562645, median = nan
standard deviation = 4182.372930953554
min. value = 0.0, max. value = 119425.0

---Feature 15 (weather_delay), Summary Statistics---
mean = 196.78078897000287, median = nan
standard deviation = 819.7059659271625
min. value = 0.0, max. value = 27876.0

---Feature 16 (nas_delay), Summary Statistics---
mean = 550.9540134487373, median = nan
standard deviation = 2147.3408283997833
min. value = 0.0, max. value = 84155.0

---Feature 17 (security_delay), Summary Statistics---
mean = 8.220074865766696, median = nan
standard deviation = 49.06743623961685
min. value = 0.0, max. value = 3760.0

---Feature 18 (late_aircraft_delay), Summary Statistics---
mean = 1121.5630346042446, median = nan
standard deviation = 4153.167863357733
min. value = 0.0, max. value = 158653.0
```

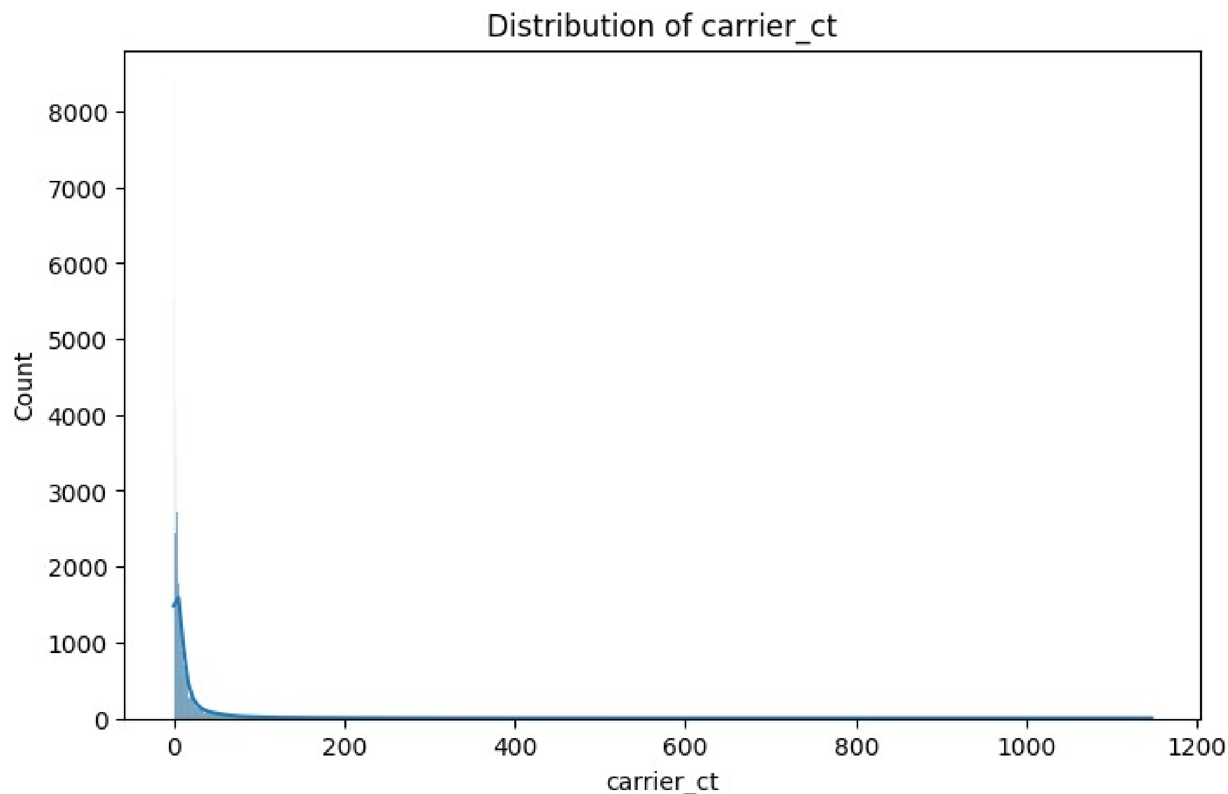
Q2 contd.

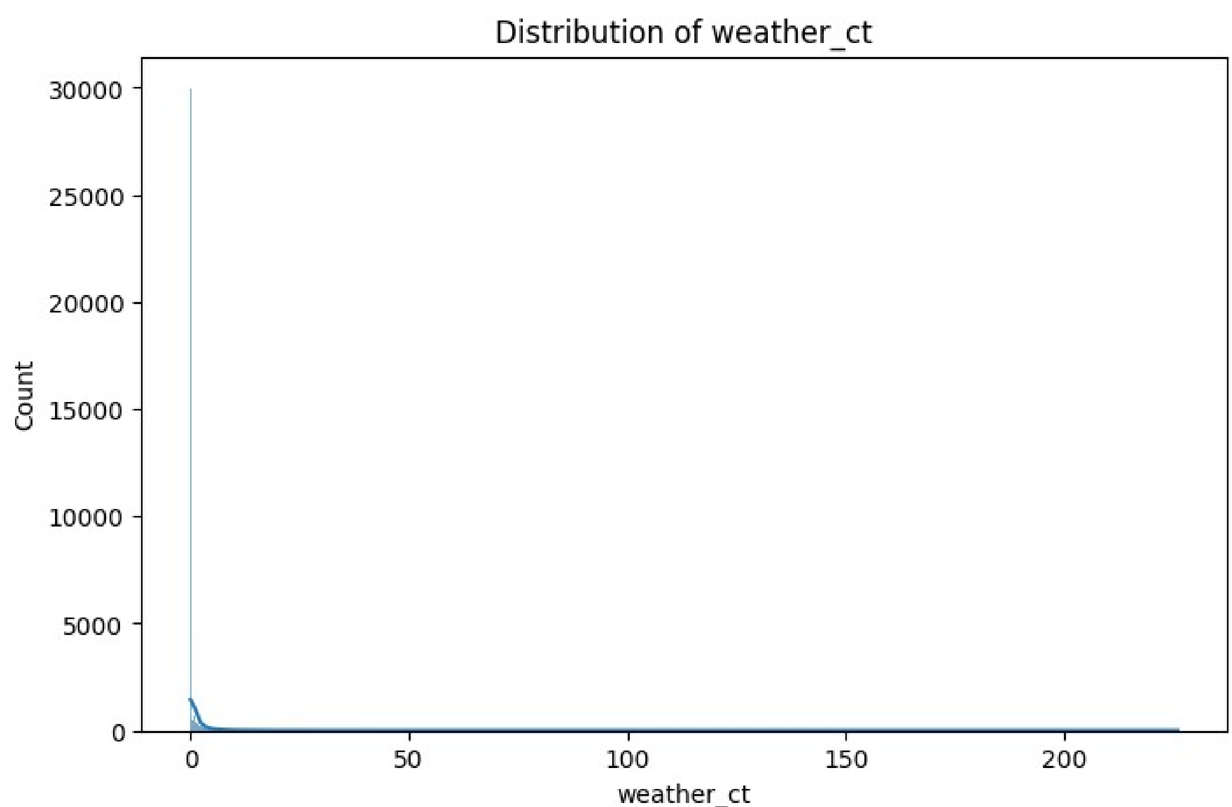
for this part of the notebook, we will attempt to look at the distribution of the data using a series of simple histogram.

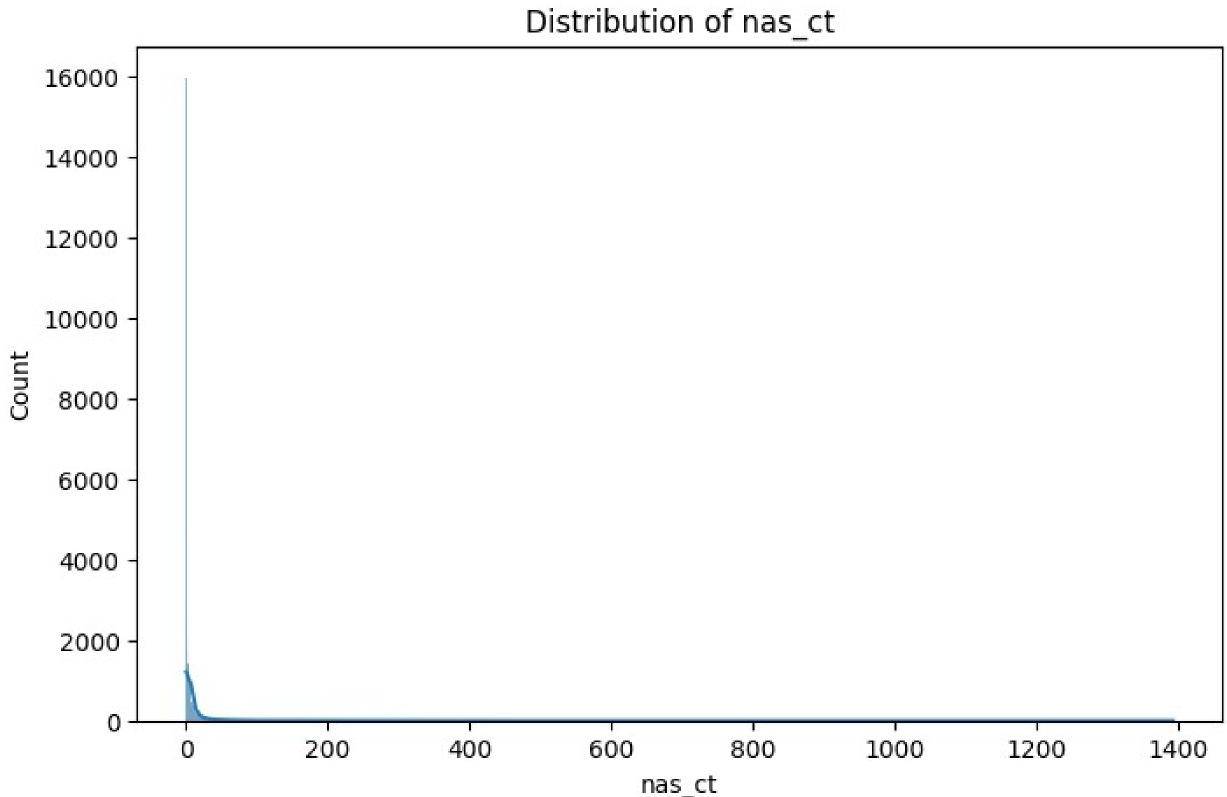
```
# Distribution of Numeric Features
quantitative_features = ['arr_flights', 'arr_del15', 'carrier_ct',
'weather_ct', 'nas_ct',
                    'security_ct', 'late_aircraft_ct',
'arr_cancelled', 'arr_diverted',
                    'arr_delay', 'carrier_delay', 'weather_delay',
'nas_delay',
                    'security_delay', 'late_aircraft_delay']

for feature in quantitative_features:
    plt.figure(figsize=(8, 5))
    sns.histplot(df[feature], kde=True)
    plt.title(f'Distribution of {feature}')
    plt.show()
```


Distribution of arr_flights
Count
0
1000
2000
3000
4000
5000
0
5000
10000
15000
20000
arr_flights

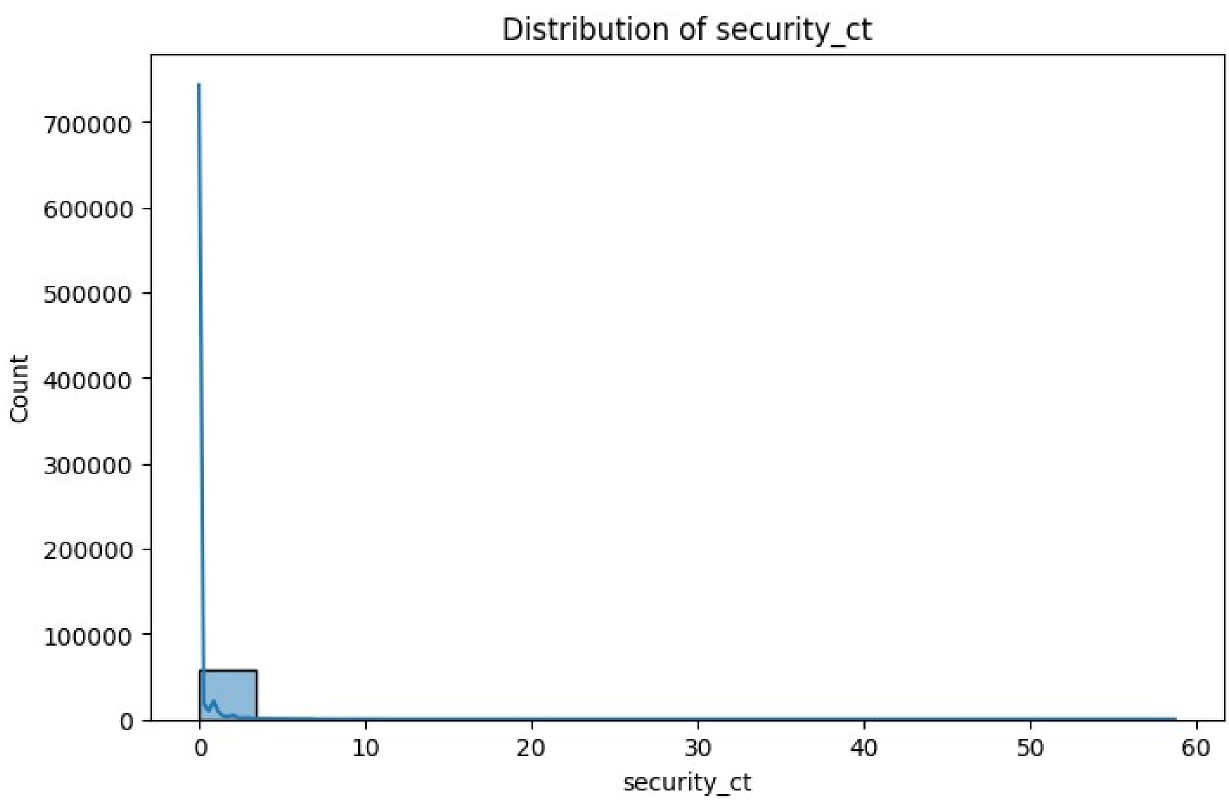

Distribution of arr_del15
Count
0
1000
2000
3000
4000
5000
6000
7000
0
500
1000
1500
2000
2500
3000
3500
arr_del15


Distribution of carrier_ct
Count
8000
7000
6000
5000
4000
3000
2000
1000
0
0
200
400
600
800
1000
1200
carrier_ct

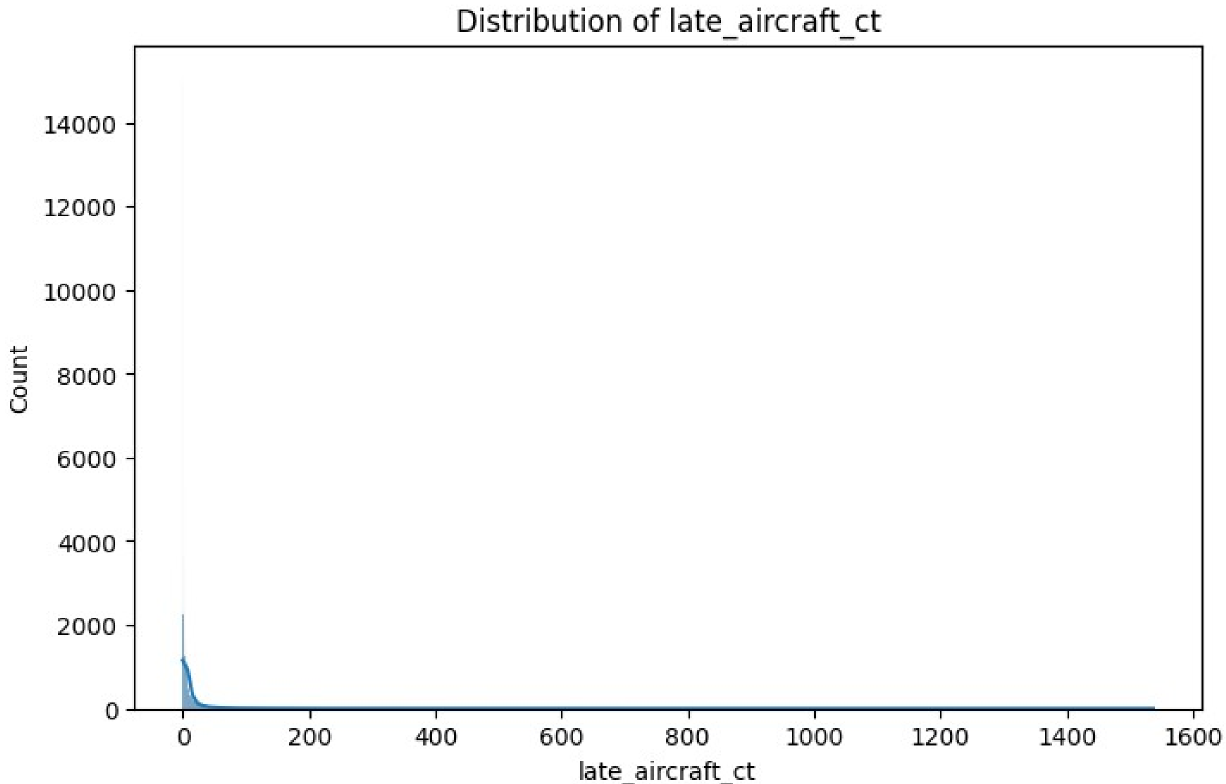

Distribution of weather_ct
Count
30000
25000
20000
15000
10000
5000
0
0
50
100
150
200
weather_ct

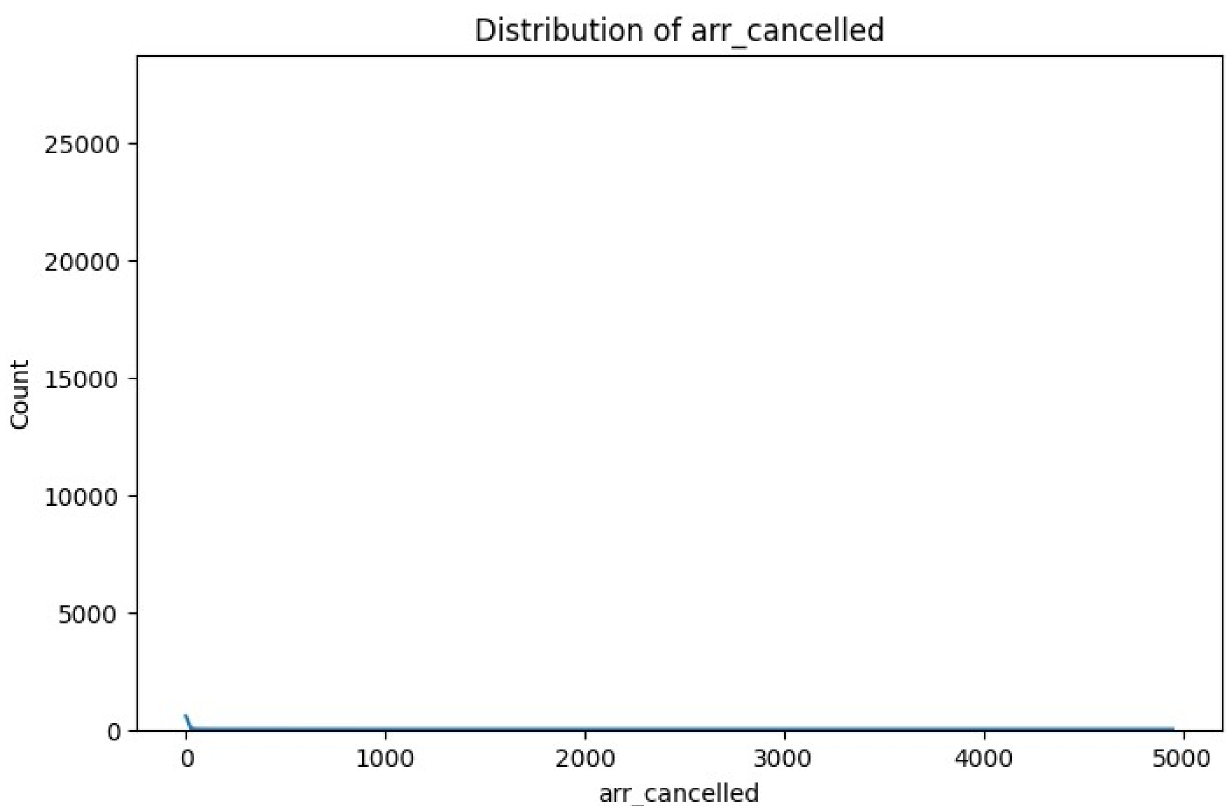


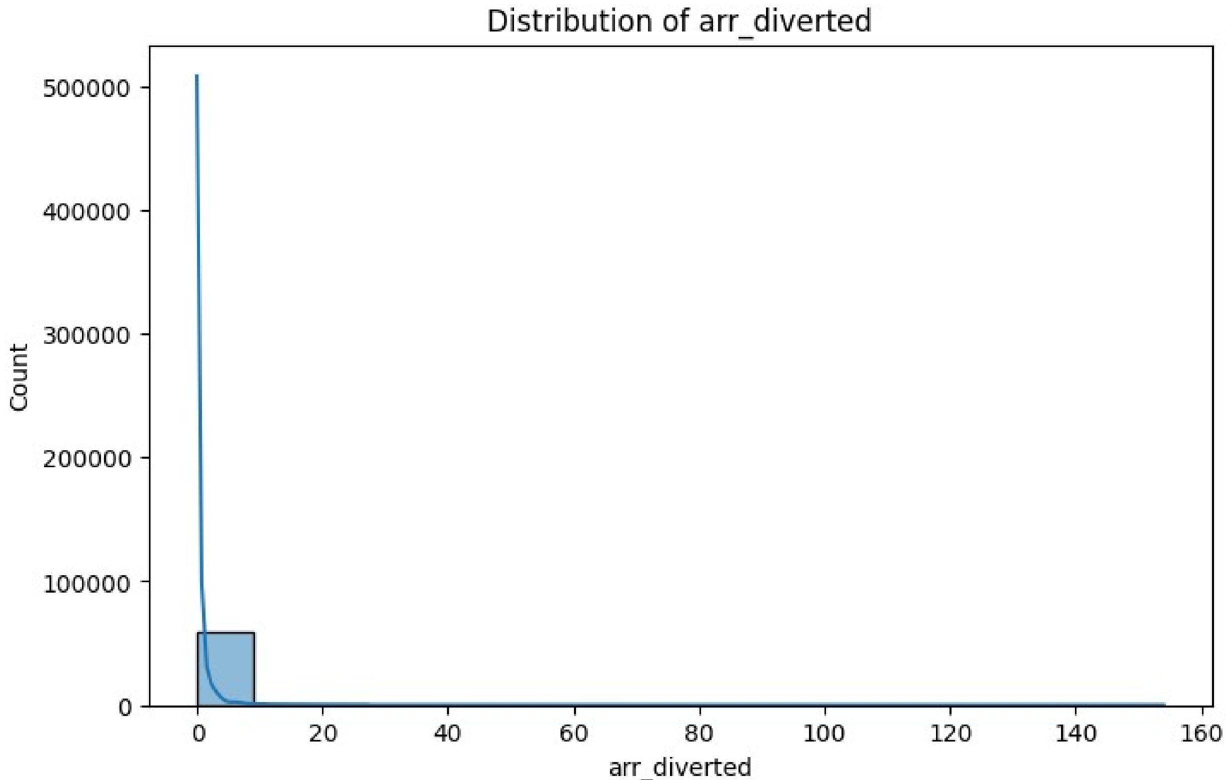
Distribution of nas_ct
Count
16000
14000
12000
10000
8000
6000
4000
2000
0
0
200
400
600
800
1000
1200
1400
nas_ct

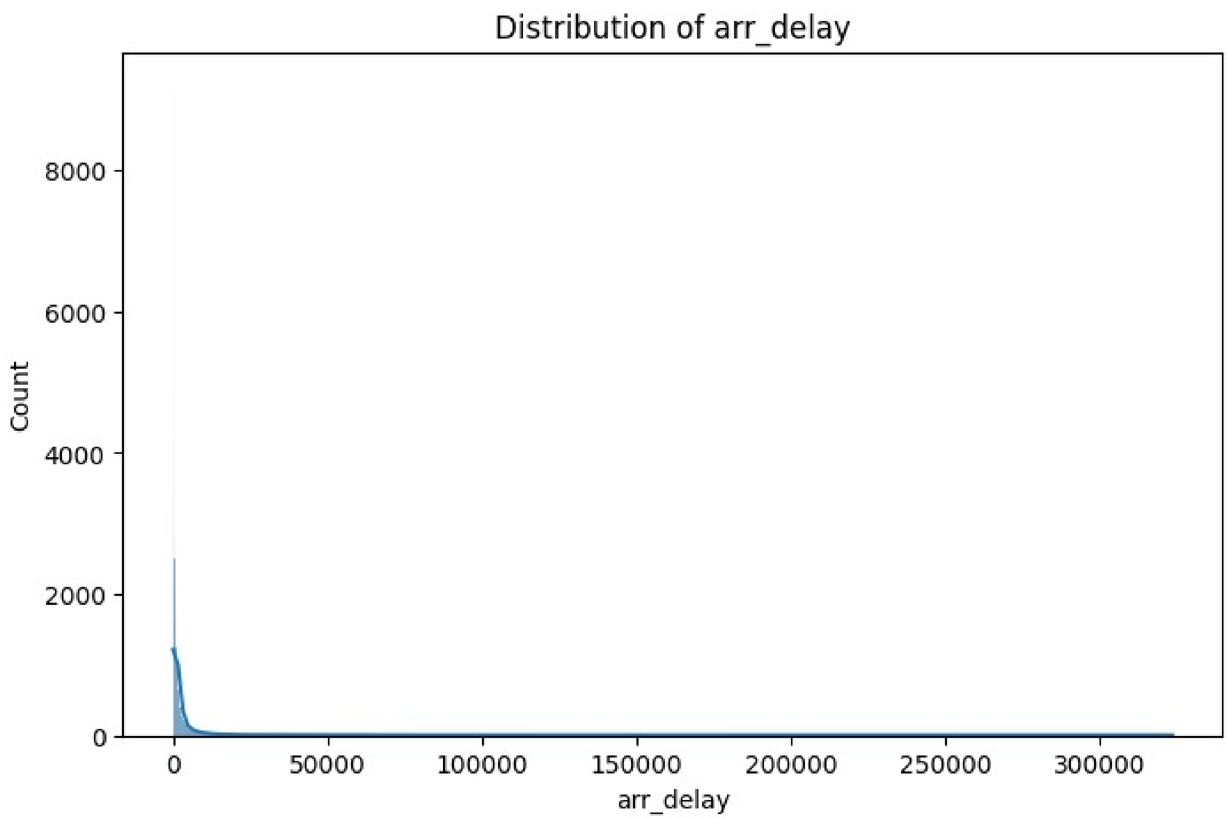


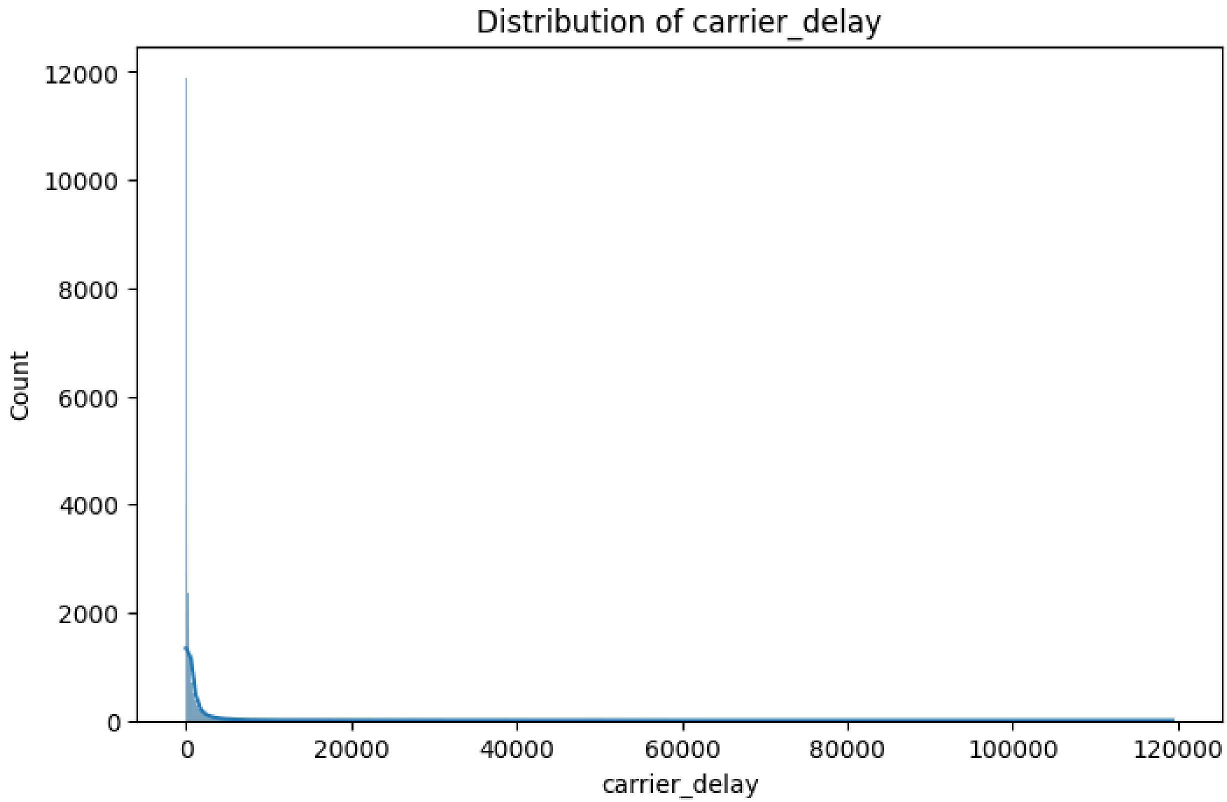
Distribution of security_ct
Count
700000
600000
500000
400000
300000
200000
100000
0
0
10
20
30
40
50
60
security_ct

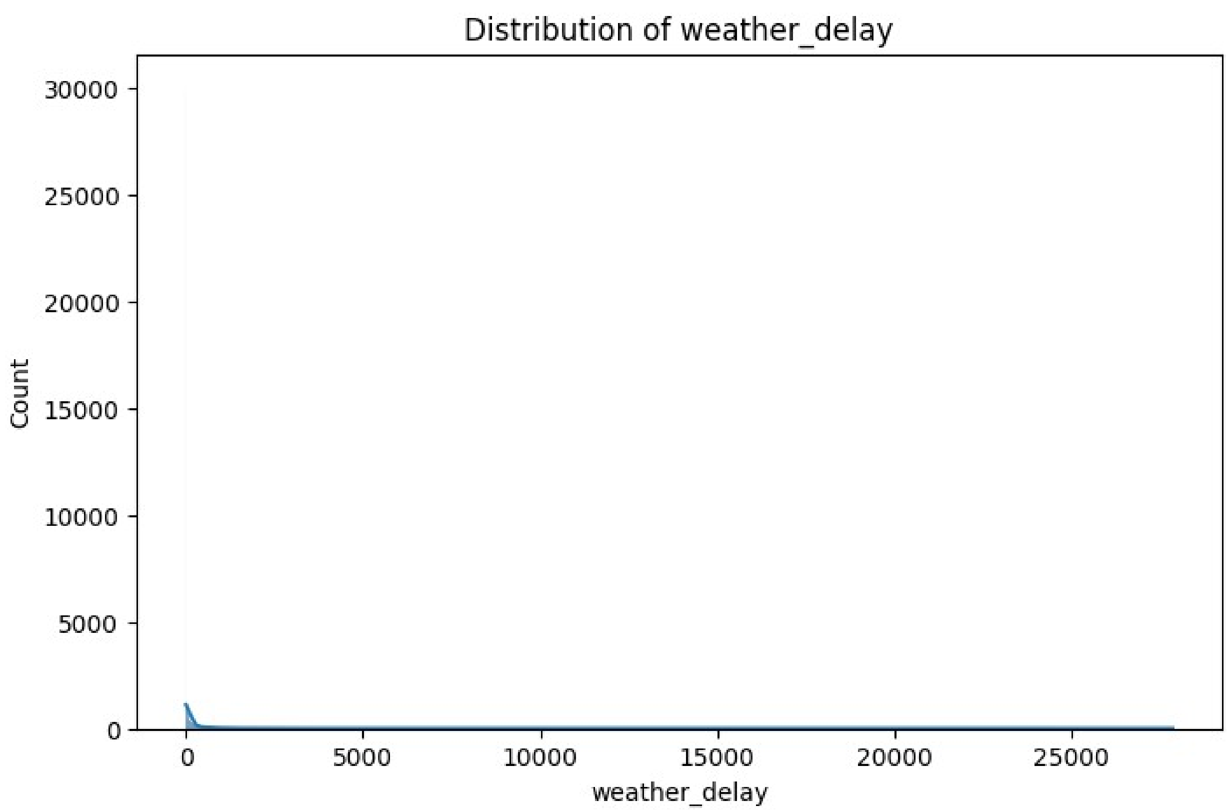

Distribution of late_aircraft_ct
Count
0
2000
4000
6000
8000
10000
12000
14000
0
200
400
600
800
1000
1200
1400
1600
late_aircraft_ct

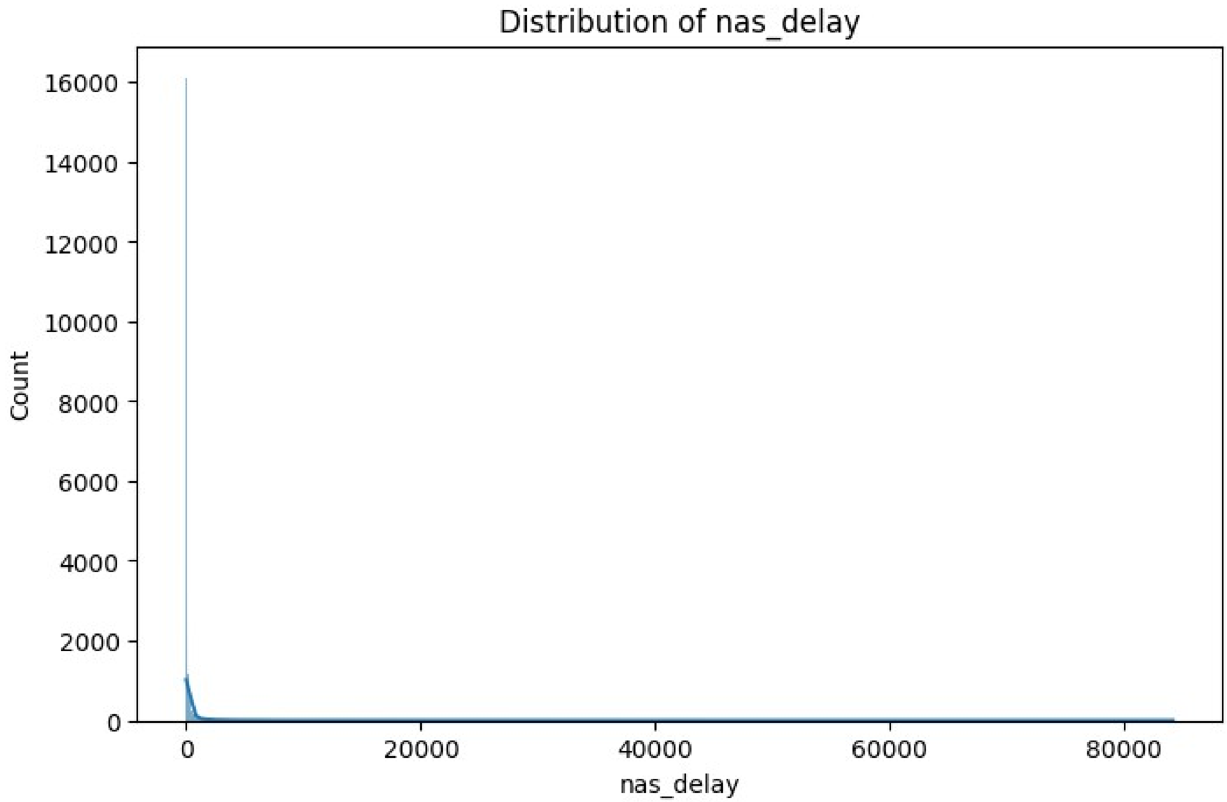

Distribution of arr_cancelled
Count
0
5000
10000
15000
20000
25000
0
1000
2000
3000
4000
5000
arr_cancelled

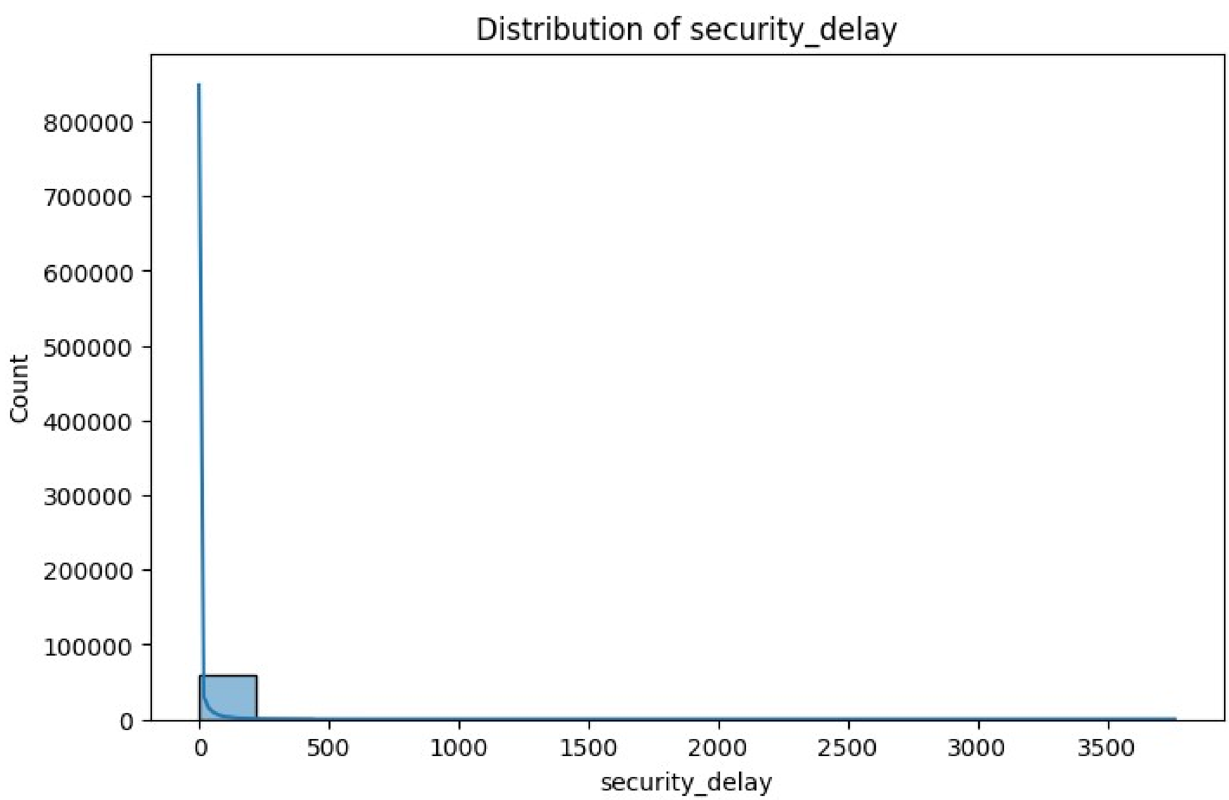

Distribution of arr_diverted
Count
arr_diverted
0
100000
200000
300000
400000
500000
20
40
60
80
100
120
140
160

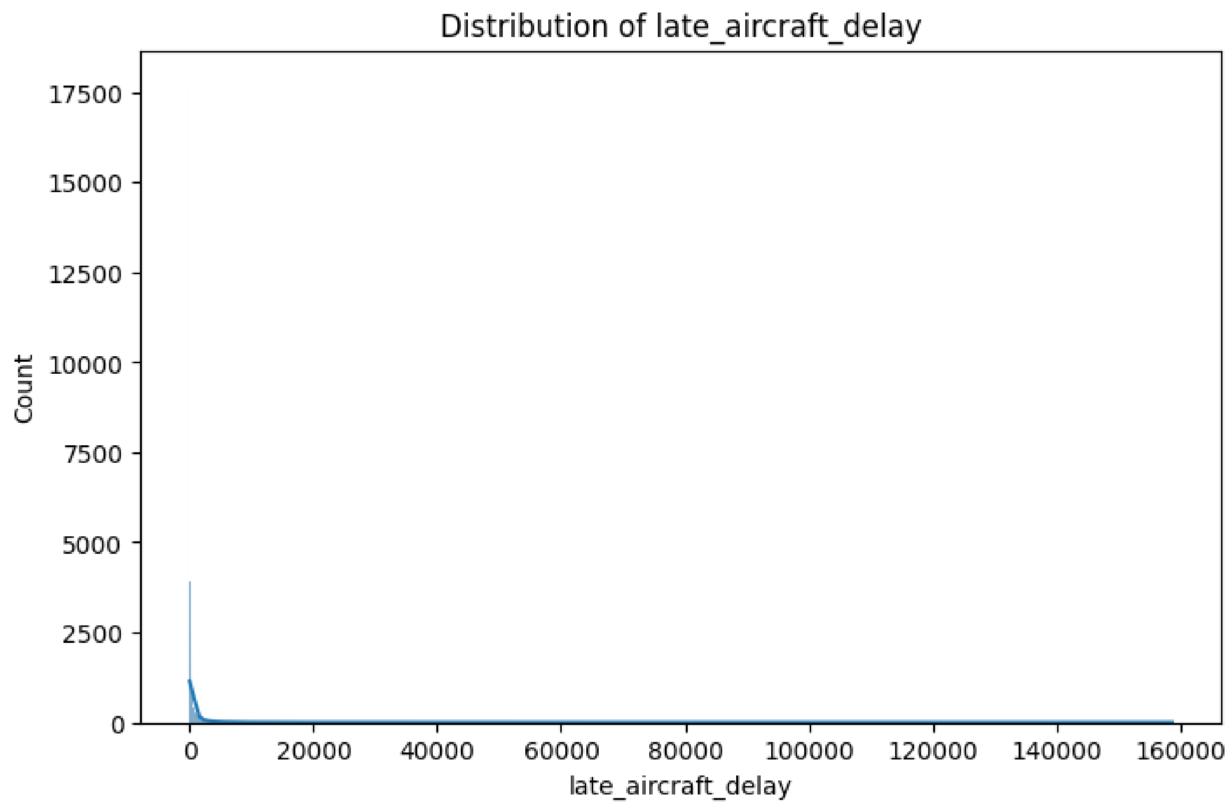

Distribution of arr_delay
Count
arr_delay
0
2000
4000
6000
8000
50000
100000
150000
200000
250000
300000


Distribution of carrier_delay
Count
carrier_delay
0
2000
4000
6000
8000
10000
12000
20000
40000
60000
80000
100000
120000


Distribution of weather_delay
Count
weather_delay
0
5000
10000
15000
20000
25000
30000


Distribution of nas_delay
Count
nas_delay
0
2000
4000
6000
8000
10000
12000
14000
16000
20000
40000
60000
80000

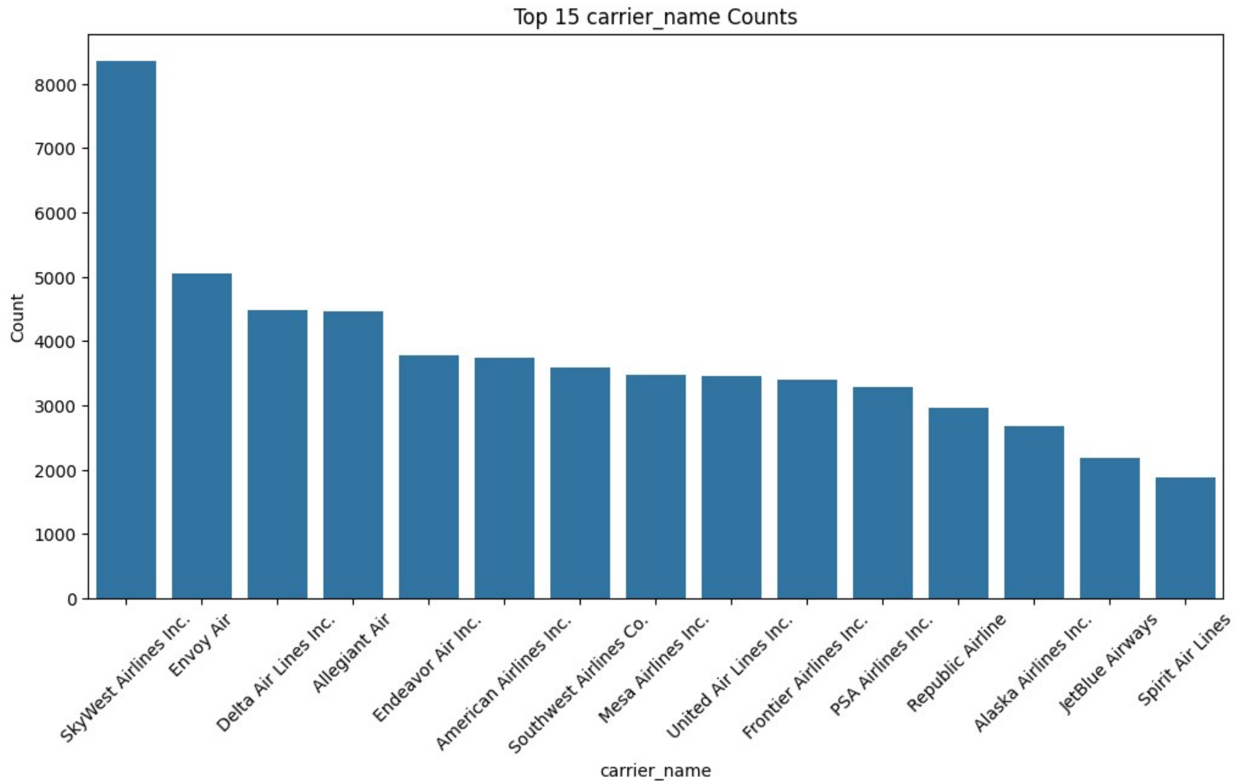

Distribution of security_delay
Count
security_delay
0
100000
200000
300000
400000
500000
600000
700000
800000
500
1000
1500
2000
2500
3000
3500

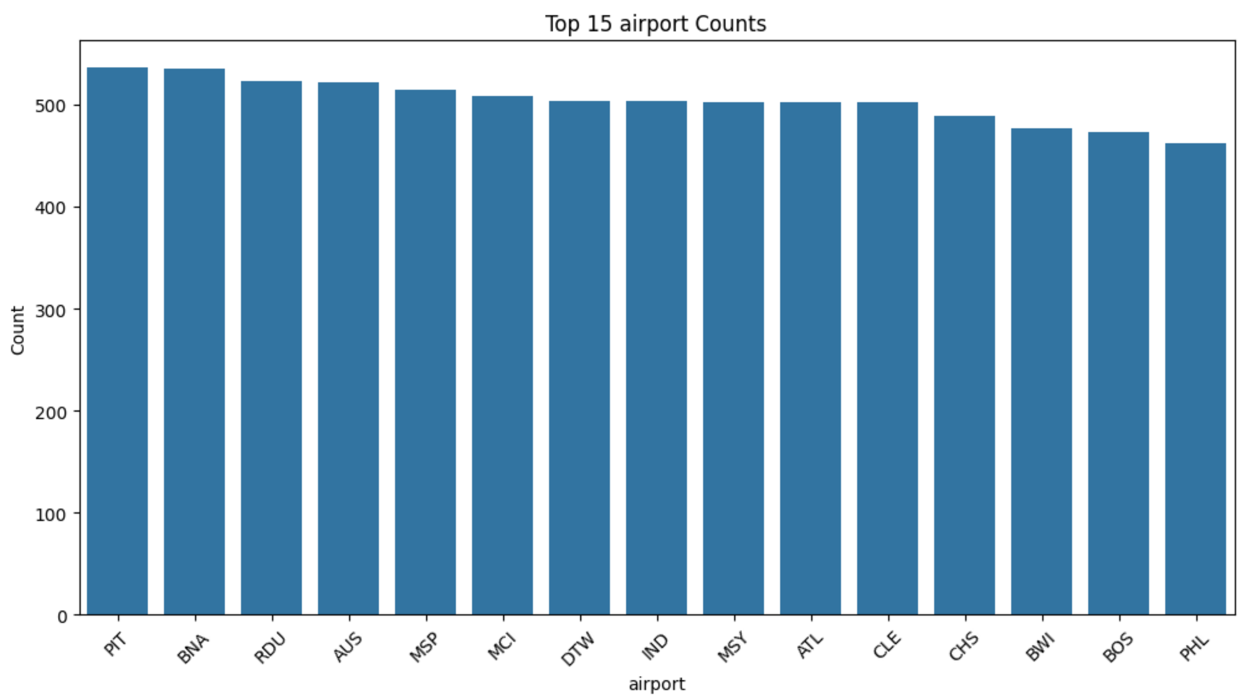


When looking at the data from the histograms created above we can see that they are mostly right skewed as the "tail" on the right is far smaller than the start on the left.

We will also try to visualize the distrubution of the qualitative features like the carrier_name and airport etc.

```
qualitative_features = ['carrier_name', 'airport']

for feature in qualitative_features:
    top_categories = df[feature].value_counts().nlargest(15)

    plt.figure(figsize=(12, 6))
    sns.barplot(x=top_categories.index, y=top_categories.values)
    plt.title(f'Top {15} {feature} Counts')
    plt.xlabel(feature)
    plt.ylabel('Count')
    plt.xticks(rotation=45)
    plt.show()
```





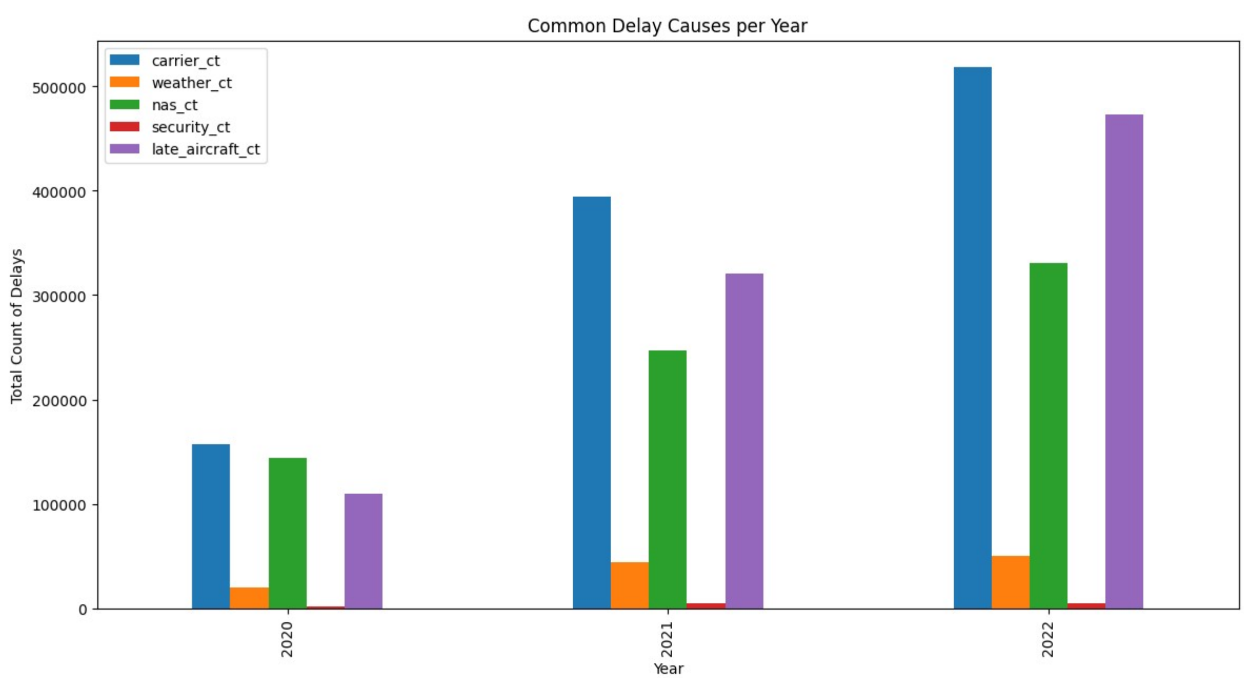
from those histograms we can get an idea of the rank of airports and carriers that have the majority of delays in the given period we are analyzing (2020-2022)

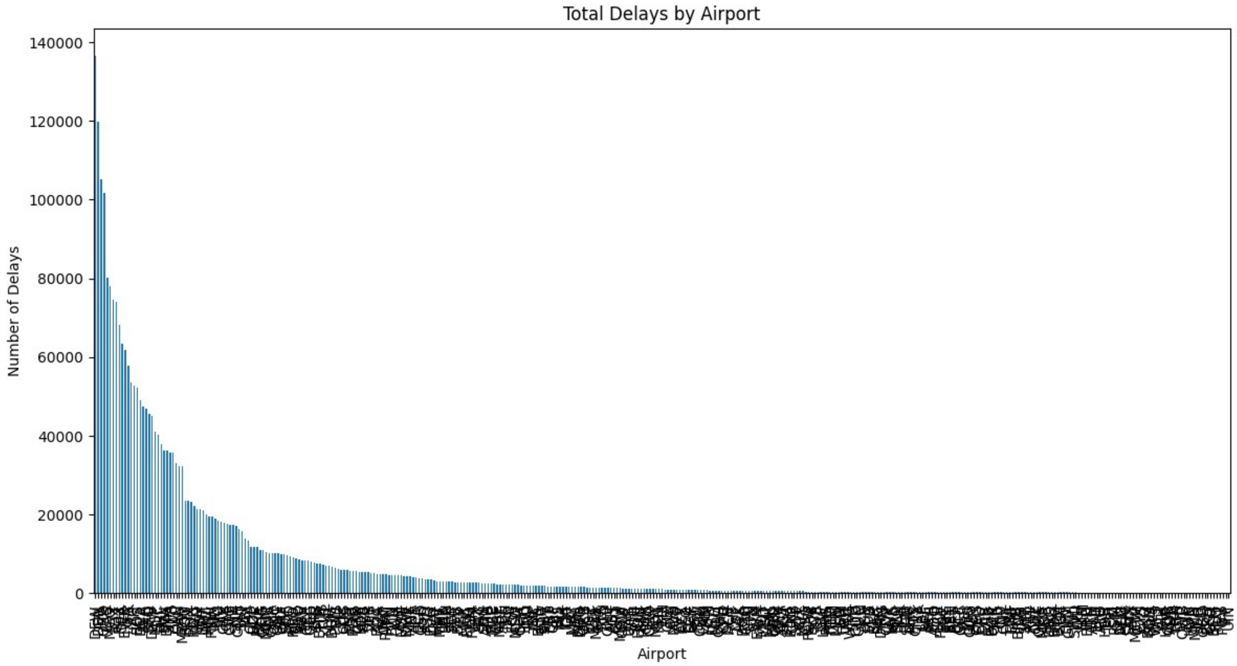
Q3 & Q4) Data Visualization and Manipulation

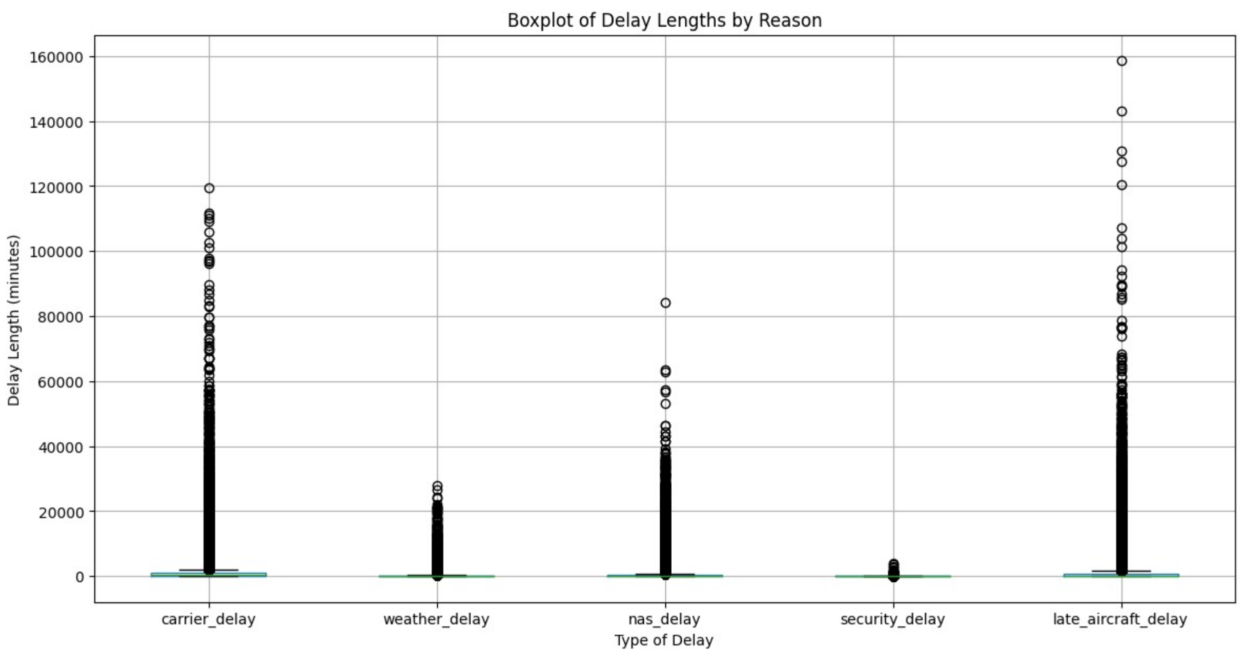
for this part we will select a series of features from the dataset and perform small analysis to find out more about the main reasons for the delays, when do they usually occur etc. We will use all the tools in our arsenal to do this including: bar charts (stacked, regular and sorted) as well as box plots

```
# delays by Month and Reason bar chart (stacked)
monthly_delays = df.groupby('month')[['carrier_delay',
'weather_delay', 'nas_delay', 'security_delay',
'late_aircraft_delay']].sum()
monthly_delays.plot(kind='bar', stacked=True, figsize=(14, 7))
plt.title('Monthly Delays by Reason')
plt.xlabel('Month')
plt.ylabel('Total Delay (minutes)')
plt.show()

# most/least common delay causes per year (bar charts)
annual_delay_causes = df.groupby('year')[['carrier_ct', 'weather_ct',
'nas_ct', 'security_ct', 'late_aircraft_ct']].sum()
annual_delay_causes.plot(kind='bar', figsize=(14, 7))
plt.title('Common Delay Causes per Year')
plt.xlabel('Year')
plt.ylabel('Total Count of Delays')
plt.show()

# most delays by airport bar chart (sorted)
airport_delays = df.groupby('airport')
['arr_del15'].sum().sort_values(ascending=False)
airport_delays.plot(kind='bar', figsize=(14, 7))
plt.title('Total Delays by Airport')
plt.xlabel('Airport')
plt.ylabel('Number of Delays')
plt.xticks(rotation=90)  # Rotate x-axis labels for better readability
plt.show()

# length of delays by Reason box plot
delay_types = ['carrier_delay', 'weather_delay', 'nas_delay',
'security_delay', 'late_aircraft_delay']
df.boxplot(column=delay_types, figsize=(14, 7))
plt.title('Boxplot of Delay Lengths by Reason')
plt.xlabel('Type of Delay')
plt.ylabel('Delay Length (minutes)')
plt.show()
```


Monthly Delays by Reason
1e7
carrier_delay
weather_delay
nas_delay
security_delay
late_aircraft_delay
Total Delay (minutes)
0.0
0.5
1.0
1.5
2.0
1
2
3
4
5
6
7
8
9
10
11
12
Month


Common Delay Causes per Year
carrier_ct
weather_ct
nas_ct
security_ct
late_aircraft_ct
Total Count of Delays
0
100000
200000
300000
400000
500000
2020
2021
2022
Year





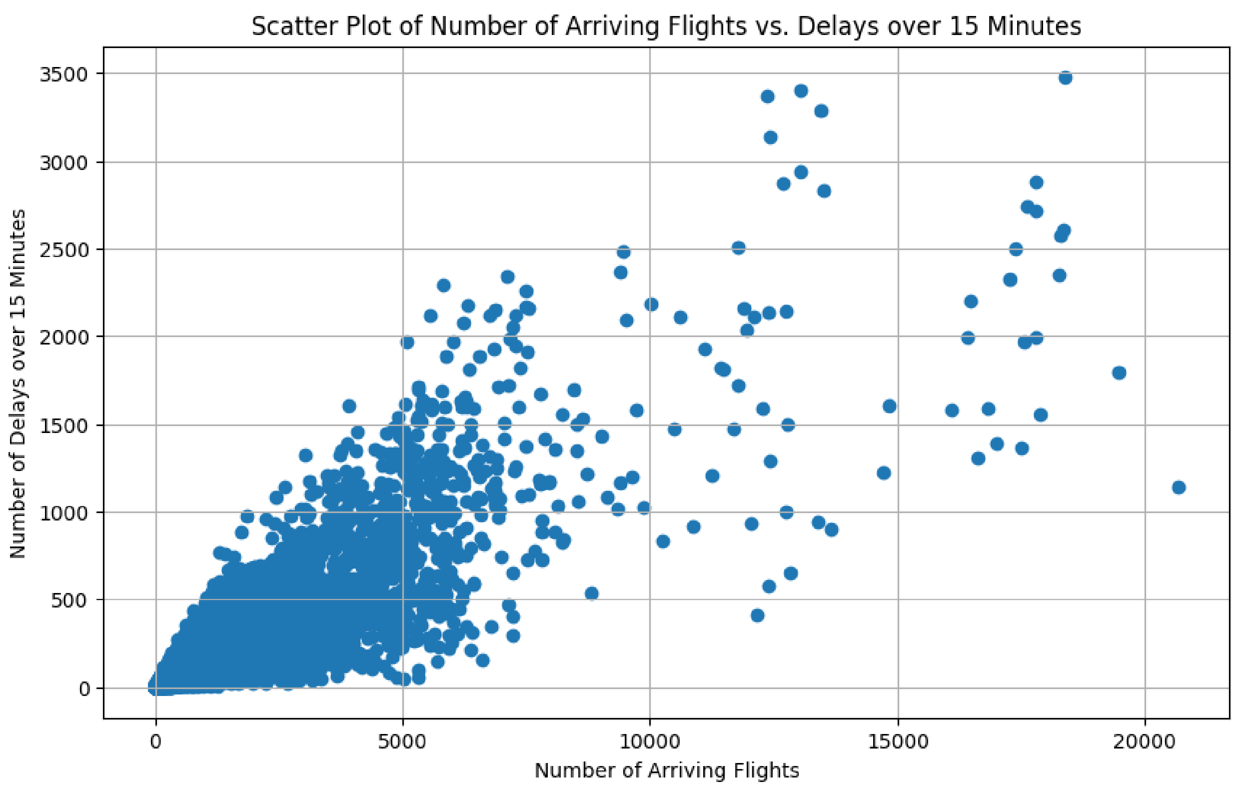
from the graphs above we can now see more useful data that we can label as facts and observe when studying this dataset.

- from the first graph (stacked bar chart) we can see the different reasons for delays and it seems that in the United States, the carriers are responsible for the majority of the delay as well as the late aircraft which usually occurs due to the previous carrier trip being late.
- from the second graph, we also see the same relation being emphasized and same reasons for the delays being dominant, note that this period is slightly after the covid-19 outbreak so this might be a significant reason why these delays are dominant, carriers

and airports had to ensure they are following safety procedures and health regulations prior to takeoff which is a significant factor for these delays.

- from the following graph (sorted bar chart) we can see the ranking of the airports that have the most delays but this time its cumalitive and not just top airports with 15mins+ delay.
- from the box plot we can see that the late aircraft, carrier, and nas delay seem to be the lengthiest delays in order, while the security and weahter delays tend to be the shortest delays.

```
#Q3 b
plt.figure(figsize=(10, 6))
plt.scatter(df['arr_flights'], df['arr_del15'])
plt.title('Scatter Plot of Number of Arriving Flights vs. Delays over
15 Minutes')
plt.xlabel('Number of Arriving Flights')
plt.ylabel('Number of Delays over 15 Minutes')
plt.grid(True)
plt.show()

correlation = df['arr_flights'].corr(df['arr_del15'])
print(correlation)
```




```
0.8871865735146494
```

From the scatter plot generated, we can see the relationship between the number of arriving flights and the number of delays over 15 minutes. From the distribution of the points, it appears there is a positive correlation of around 0.887 between these two variables: as the number of arriving flights increases, the number of delays over 15 minutes also tends to increase.

The scatter is more densely populated at the lower end of both axes, which indicates that smaller numbers of flights are more frequently associated with lower numbers of delays. As we move towards the higher end of the 'Number of Arriving Flights' axis, the points spread out more, suggesting variability increases with the larger number of flights. The relationship is not perfectly linear though due to noise and some outlier. Nevertheless, the positive correlation is still clearly observed

Overall, the scatter plot suggests a general trend where more arriving flights can lead to more delays, but with significant variability and some exceptions to the trend.

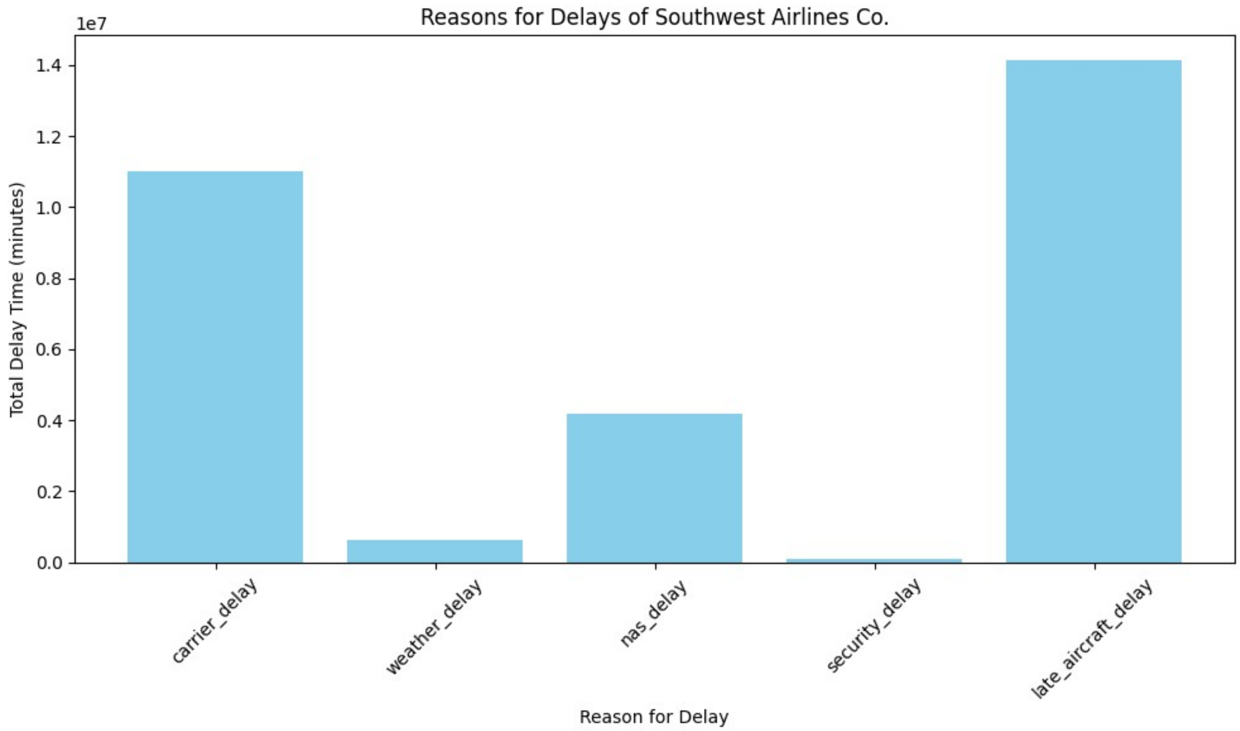
```
#Q4 airline
# grouping the data by carrier_name and sum the delays
carrier_delays = df.groupby('carrier_name').agg({
    'arr_del15': 'sum',
    'carrier_delay': 'sum',
    'weather_delay': 'sum',
    'nas_delay': 'sum',
    'security_delay': 'sum',
    'late_aircraft_delay': 'sum'
}).reset_index()

most_delayed_airline = 
carrier_delays.loc[carrier_delays['arr_del15'].idxmax()]
print(most_delayed_airline)
```

```
carrier_name           Southwest Airlines Co.
arr_del15                            593189.0
carrier_delay                      10997055.0
weather_delay                        633295.0
nas_delay                           4181903.0
security_delay                        89676.0
late_aircraft_delay                14134417.0
Name: 15, dtype: object
```

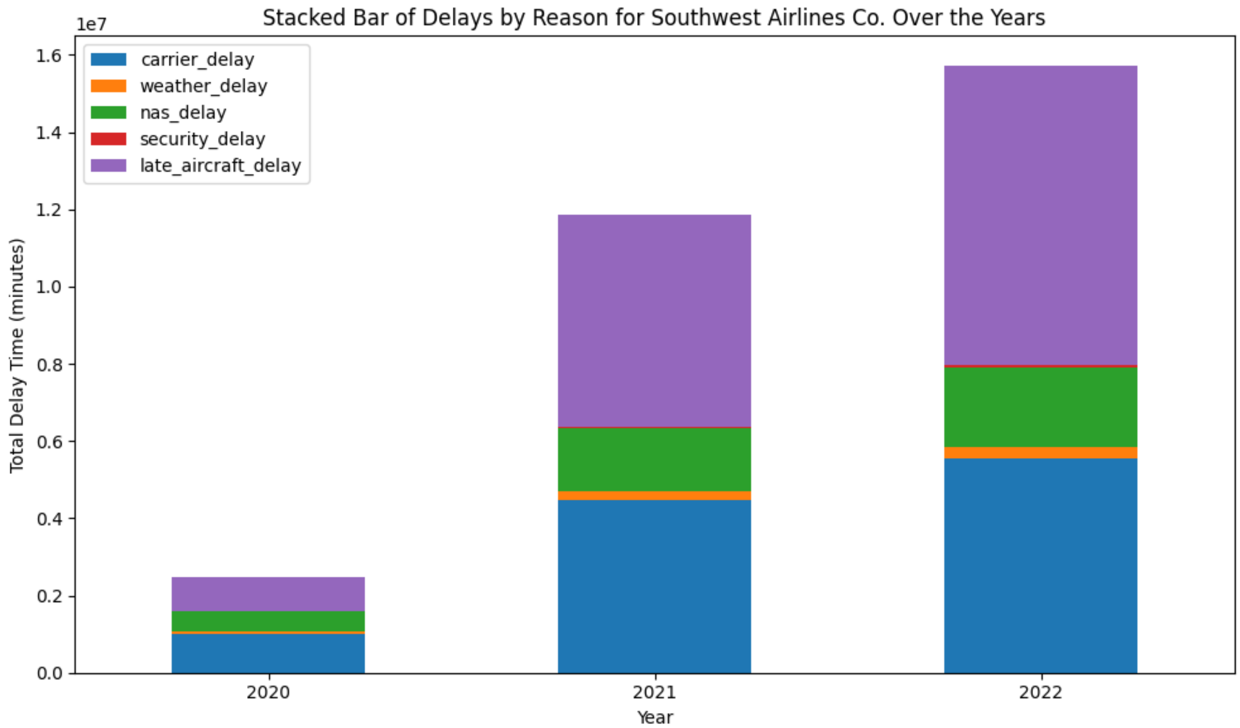
- In this part above, we identifed the airline with the most delays and we will analyze what is the leading cause fo delays for the carrier.
- The carrier that seems to have the most delays is Southwest Airlines. we will now visualize the reasons for the delays.

```
#Q4 airline 2

plt.figure(figsize=(10, 6))
reasons = ['carrier_delay', 'weather_delay', 'nas_delay', 
'security_delay', 'late_aircraft_delay']
delay_amounts = most_delayed_airline[reasons]
```

```
plt.bar(reasons, delay_amounts, color='skyblue')
plt.title(f'Reasons for Delays of
{most_delayed_airline["carrier_name"]}')
plt.xlabel('Reason for Delay')
plt.ylabel('Total Delay Time (minutes)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```




```
#Q4 airlines visuals
most_delayed_airline_data = df[df['carrier_name'] ==
most_delayed_airline['carrier_name']]
summary_most_delayed_airline =
most_delayed_airline_data.groupby('year')[reasons].sum()

summary_most_delayed_airline.plot(kind='bar', stacked=True,
figsize=(10, 6))
plt.title(f'Stacked Bar of Delays by Reason for
{most_delayed_airline["carrier_name"]} Over the Years')
plt.xlabel('Year')
plt.ylabel('Total Delay Time (minutes)')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```



Now we can clearly see that the leading reason for delay for Southwest is late aircraft delay. This is then followed by carrier delay and Nas delay which seems like it has carried over the trend from the total dataset.

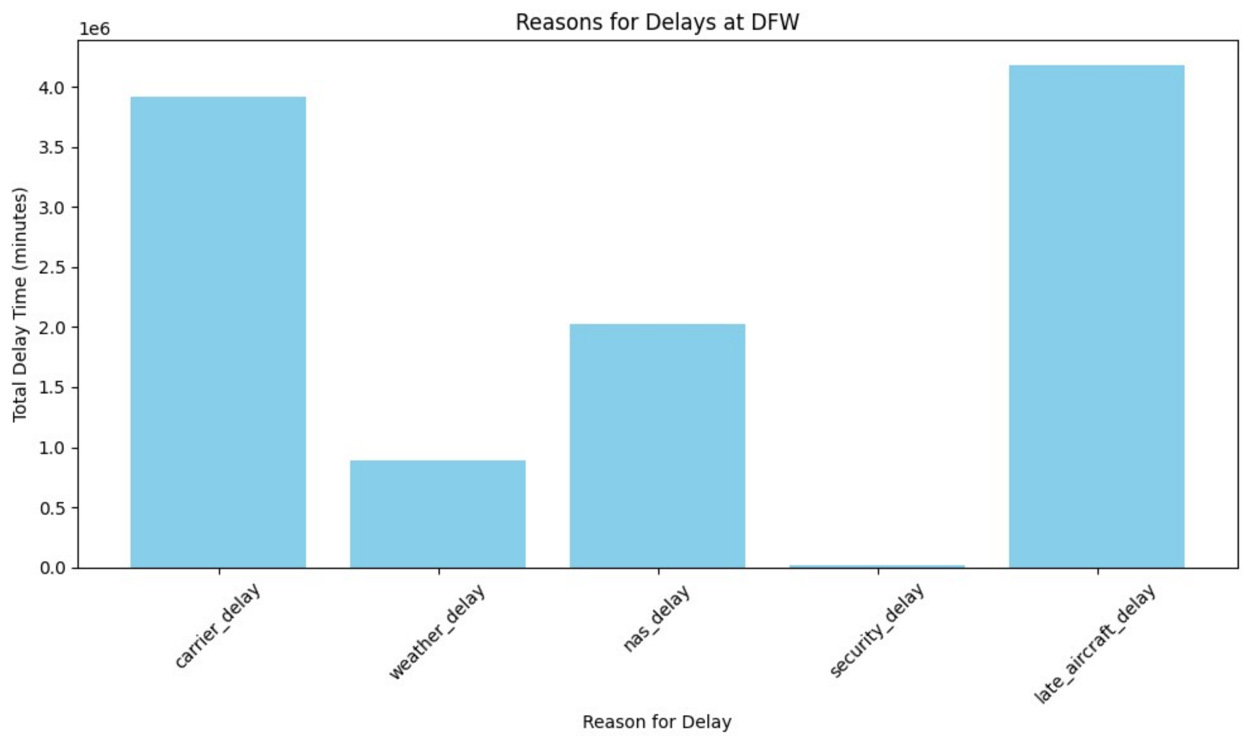
```
#Q4 airport
# group the data by airport_name and sum the delays
airport_delays = df.groupby('airport').agg({
    'arr_del15': 'sum',
    'carrier_delay': 'sum',
    'weather_delay': 'sum',
    'nas_delay': 'sum',
    'security_delay': 'sum',
    'late_aircraft_delay': 'sum'
}).reset_index()

# airport with the most number of delays
most_delayed_airport =
airport_delays.loc[airport_delays['arr_del15'].idxmax()]
print(most_delayed_airport)
```

```
airport                      DFW
arr_del15               136598.0
carrier_delay          3916607.0
weather_delay           886876.0
nas_delay              2021324.0
security_delay           21569.0
```

```
late_aircraft_delay    4179443.0
Name: 99, dtype: object
```

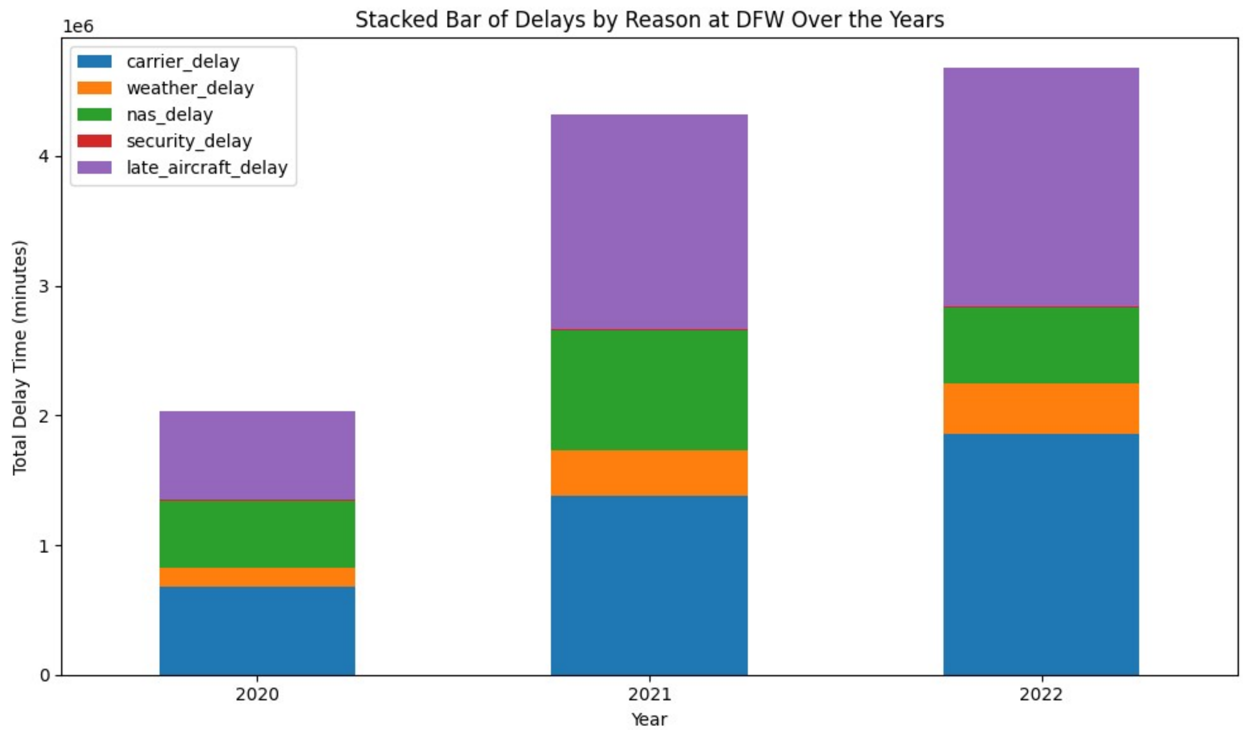
We have now repeated the same steps we have done to find the airline with the most delays to get the airport with the most delays. DFW or Dallas Fort Worth International Airport seems to be the airport with the most delays, this was also observed earlier in the sorted bar chart, lets dive into what are the causes for the delay in this airport and whether its the airports fault or more carrier reasons.

```
#Q4 airport 2
# Assuming you've already computed 'airport_delays' as in step 3
most_delayed_airport_name = most_delayed_airport['airport']

# Plot a bar chart to show delay reasons for the most delayed airport
plt.figure(figsize=(10, 6))
delay_amounts_airport = most_delayed_airport[reasons]
plt.bar(reasons, delay_amounts_airport, color='skyblue')
plt.title(f'Reasons for Delays at {most_delayed_airport_name}')
plt.xlabel('Reason for Delay')
plt.ylabel('Total Delay Time (minutes)')
plt.xticks(rotation=45)
plt.tight_layout()  # Adjusts plot parameters for better fit
plt.show()
```




```
#Q4 airports - visuals
most_delayed_airport_data = df[df['airport'] ==
```

```
most_delayed_airport_name]
summary_most_delayed_airport =
most_delayed_airport_data.groupby('year')[reasons].sum()

summary_most_delayed_airport.plot(kind='bar', stacked=True,
figsize=(10, 6))
plt.title(f'Stacked Bar of Delays by Reason at
{most_delayed_airport_name} Over the Years')
plt.xlabel('Year')
plt.ylabel('Total Delay Time (minutes)')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```



The general trend of the dataset seems to have carried over, the main cause for delay at DFW is also late aircrafts followed by carrier delay but the margin and difference between them seem to be alot lower than that observed in the most delayed airline. the ranking of the other reasons for delays is also identical to the airline, and the rest of the dataset.

```
#Q4 other relevant stats
# summary statistics for each airline
carrier_summary_stats = df.groupby('carrier_name')
[reasons].agg(['mean', 'median', 'std']).reset_index()

# summary statistics for each airport
airport_summary_stats = df.groupby('airport')[reasons].agg(['mean',
'median', 'std']).reset_index()
```

```
# airlines
print("Summary statistics for airlines:")
print(carrier_summary_stats)

# airports
print("Summary statistics for airports:")
print(airport_summary_stats)
```

```
Summary statistics for airlines:
               carrier_name carrier_delay
weather_delay  \
                                     mean  median          std
mean
0      Alaska Airlines Inc.    577.312360   152.0  1759.617354
69.793373
1             Allegiant Air    419.623234   128.0   910.339912
85.207670
2    American Airlines Inc.   2941.973241   784.0  8357.297448
387.253412
3      Delta Air Lines Inc.   2111.026760   493.0  6393.284104
222.327187
4         Endeavor Air Inc.    552.726020   170.0  1682.304692
131.704557
5                 Envoy Air    330.696238    93.0  1404.146740
161.607723
6   ExpressJet Airlines LLC    160.724913    22.5   535.854466
29.975779
7    Frontier Airlines Inc.    593.833038   107.0  1590.209172
47.165782
8    Hawaiian Airlines Inc.    998.207773   338.0  2370.568339
72.173393
9               Horizon Air    461.717373   181.0  1030.171927
69.364736
10          JetBlue Airways   2327.104443   681.0  5165.245998
160.916170
11        Mesa Airlines Inc.    680.684256   186.0  1847.948725
159.452015
12         PSA Airlines Inc.    638.572866   187.0  2300.236280
153.467073
13          Republic Airline    879.156823   194.0  1961.803914
164.793618
14    SkyWest Airlines Inc.   1707.006842   439.0  5520.824919
394.704477
15   Southwest Airlines Co.   3069.231091  1290.0  4910.626399
176.749930
16          Spirit Air Lines   1107.510378   452.0  1794.896226
175.195317
17    United Air Lines Inc.   1462.383028   344.0  3902.059137
```

```
248.386225

                                  nas_delay                    security_delay  \
   median          std         mean median          std              mean   median
0     0.0   325.947557   468.447133   93.0  1669.442236         14.242740   0.0
1     0.0   256.478185   246.160126   55.0   679.538632          5.568962   0.0
2    54.0  1320.963497  1157.330211  228.0  3665.880394         20.342521   0.0
3    17.0   791.349539   763.192264  127.0  2363.262582          9.088149   0.0
4     0.0   480.076506   337.946741   79.0  1565.712466          0.991521   0.0
5    19.0   874.383559   326.899208   92.0  1771.751191          2.397822   0.0
6     0.0   111.501923   244.544983   36.0  1178.838872          0.000000   0.0
7     0.0   188.317701   417.460177   63.0  1376.908004          0.000000   0.0
8     0.0   520.151564    37.055306    0.0   287.135103          7.396114   0.0
9     0.0   396.802174   217.866033   62.0   726.113279          4.078652   0.0
10    0.0   429.513681   911.768667  194.0  2378.683320         18.356390   0.0
11    0.0   741.650590   264.791534   53.0  1144.879409          1.870687   0.0
12    9.0   616.589241   341.609756  112.0  1266.868288          5.094512   0.0
13    0.0   467.427707   786.631704  151.0  2574.077575          3.610659   0.0
14   42.0  1411.968009   157.625015    0.0  1258.127198          4.950666   0.0
15   30.0   400.945694  1167.151270  285.0  2964.971478         25.028189   0.0
16   23.0   389.435141  1376.182544  459.0  2606.028447         38.304949   0.0
17   11.0   795.222943   950.569602  135.0  3711.582228          1.016565   0.0

                late_aircraft_delay
           std                 mean median          std
0   110.433140           568.422561   93.0  2245.791728
1    28.346410           566.176497   98.0  1561.423047
2    61.863425          2859.545625  629.0  9105.491500
3    41.352700          1019.011468  129.0  4337.814575
```

```
4     7.674600          463.616587    69.0  1612.747397
5    16.017183          463.512475   118.0  2039.297178
6     0.000000          115.828720     0.0   542.980768
7     0.000000          733.728909    90.0  2399.142975
8    23.608459          602.423019    35.0  2429.194327
9    24.853095          503.796889   133.0  1377.033869
10   72.837617         1806.940449   283.0  5585.780746
11   21.102503          595.755001    49.0  2402.068930
12   21.130959          945.383537   236.0  3741.251554
13   16.192186          950.418873   198.5  2214.840898
14   42.447531          638.780218    87.0  2593.393480
15   75.832031         3944.855428  1295.0  7504.404174
16  115.984907         1096.814795   353.0  2106.679442
17    9.582355         1714.381575   372.0  4717.580979
Summary statistics for airports:
    airport carrier_delay                     weather_delay         \
                     mean median          std          mean median
0       ABE    318.524390  158.0   434.397862     84.634146   14.5
1       ABI    384.729167  254.5   417.524097    172.583333   69.0
2       ABQ    573.367647  136.5  1010.230249     67.850490    0.0
3       ABR    345.472222  181.5   381.823406    133.472222   59.5
4       ABY    223.621622  149.0   256.532146     37.432432    0.0
..      ...           ...    ...          ...           ...    ...
374     XNA    273.457143  145.0   341.516243     61.834921    0.0
375     XWA    711.527778  524.0   669.009993    405.194444   18.0
376     YAK     58.444444   32.0    82.975881     19.388889    0.0
377     YKM    142.416667  111.5   104.216295     31.833333    0.0
378     YUM    308.984375  197.5   340.847892     58.546875    0.0

                   nas_delay                     security_delay
\
               std        mean median          std           mean median

0     164.702186  109.329268   58.5  174.304511       0.993902    0.0

1     245.738473  193.625000  120.0  190.680821       0.687500    0.0

2     190.121466  110.303922   43.0  162.600733       2.522059    0.0

3     220.192446    5.472222    0.0   16.608924       0.000000    0.0

4      73.446708   76.216216   53.0   78.448191       0.000000    0.0

..           ...         ...    ...         ...            ...    ...

374   142.460470  133.714286   49.0  235.041372       3.076190    0.0

375  1139.682884   59.916667    0.0  215.836760       0.000000    0.0

376    48.092933   90.500000   62.0   69.549366       3.527778    0.0
```

```
377   117.839119   26.958333   20.5   22.647544       0.000000    0.0

378   151.262825   27.921875    0.0   57.867474       0.093750    0.0


               late_aircraft_delay
          std                 mean median          std
0     5.260685          326.329268  120.0   638.365514
1     4.763140          286.812500  213.0   257.360717
2    11.273003          552.985294  109.5  1215.768415
3     0.000000            1.527778    0.0     9.166667
4     0.000000           73.081081   10.0   125.439534
..         ...                 ...    ...          ...
374  17.701000          305.866667   85.0   556.370026
375   0.000000          187.916667   77.0   265.898947
376  13.451471          257.722222  227.0   209.284442
377   0.000000          130.125000  100.0   116.001242
378   0.750000          465.281250  240.0   663.917318

[379 rows x 16 columns]
```

- Here is a summary of the statistics of all the airlines and Airports in the United States and their delays.
- The general trend of delays seem to continue for all the airports and all the airlines as observed before.

In conclusion we have observed the general trends when it comes to the reasoning behind trip delays in all US airports in the period (2022 to 2023). We have identified the main reasons and ranking of delays, we have found the airport and airlines with the most delays and further looked at their causes. We can conclude that the general delay reasons for that period of time applies to most airports and airline in the same ranking. We must also note that this trend can be observed due to breakout of the pandemic that occured very shortly prior to this period (around 2019). In this period, airports and airlines had to follow strict regulations and rules, along with recommendations from the world health organization (WHO) to operate safely and in a healthy manner.