WASHINGTON UNIVERSITY IN ST. LOUIS

OLIN BUSINESS SCHOOL

FL2023.B69.DAT.560M

Final Project

Group 33

Tammy Zhong 523724

Morata Zhou 519566

Guangze Jin 513542

Xinge Han 517194

Yufan Lu 517303

Boyi Ren 523284

Analysis Report on Flight Delays in the 2021 US Airports

1. Executive Summary

This report presents an in-depth analysis of flight delays in US airports for 2021, using a structured dataset of 1.89 GB. The study focuses on factors such as airlines, months, departure locations, and shared partner flights, aiming to identify patterns in flight delays. Utilizing PySpark and Random Forest algorithms, the analysis highlights the variance in delay frequencies among different airlines and months, and the impact of departure locations and shared partner flights. In addition, this analysis uses random forest to predict the likelihood of delay given certain factors. The findings are intended to guide travelers and stakeholders in making informed decisions to minimize delays, thereby enhancing efficiency and reliability in air travel. The recommendations provided are based on data-driven insights, offering practical solutions to common travel dilemmas such as choosing between airlines for timely arrivals.

2. Introduction & Problem Statement

This report investigates the intricate patterns of flight delays across US airports in 2021, addressing a key concern for travelers and airline stakeholders: the likelihood of delays and their influencing factors.

Consider a scenario where you are arranging a business trip in July and are faced with a choice between two flights. Both flights are comparable in price and departure time. However, since you wish your flight have a less likelihood of a delay, the choice between the two airlines becomes crucial. This situation underscores the importance of analyzing the relationship between delay time and carrier.

Our research is inspired by this question that one can encounter on a regular basis. Apart from examining delay-and-carrier pattern, we also delve into relationship between delay and months, departure locations, and whether it is a shared partner flight. To do an in-depth analysis, we

unitized a dataset which encompasses comprehensive delay information from US airports. This dataset, a structured record of 1.89 GB, aligns with the 4V characteristics of big data: significant volume, high velocity, diverse variety, and strong commitment to veracity.

Our methodology incorporates the use of PySpark and Random Forest algorithms, focusing on unraveling the complex relationships between various factors such as airline, month, origin, and the presence of shared partner flights. These elements are crucial in understanding the dynamics of flight delays.

3. Main Analysis and Results

In this section, the total delay time is examined in relation to four indicators: the month of departure, the origin, the route, and whether it is a shared flight, exploring potential associations between flight delays and these factors. The section concludes by employing a random forest model, allowing users to predict flight delays using available flight information.

3.1. Flights delay rate with departure month

To initiate our analysis, we investigated the relationship between flight delays and the departure month. The distribution of total delay minutes over the 12 months is depicted in Figure 1. (Fig. 1.). Remarkably, a discernible trend emerges, with higher total delay minutes during the summer months, particularly in July. Conversely, the first quarter exhibits comparatively shorter delay minutes. However, it is essential to consider the influence of the total number of flights on this distribution. Recognizing the potential impact of the total number of flights on delay minutes, we collected data on flight numbers per month (Fig. 2.). The findings reveal a correlation between the total number of flights and delay minutes, with July registering the

highest number of flights and February the lowest.

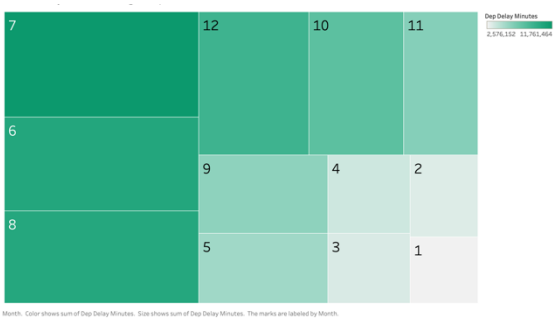


Fig. 1. Total delay minutes of different departure months.

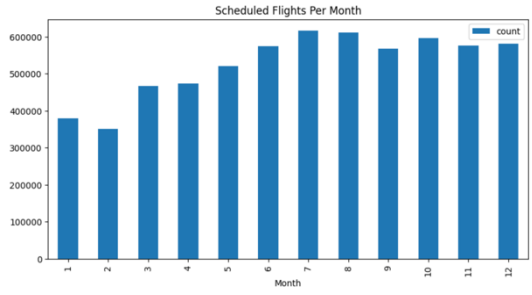


Fig. 2. Total flights number of different departure months.

To delve deeper into the analysis, PySpark was employed to differentiate flights based on their arrival status, and the resulting five flight status types were subjected to categorical statistics. Flights were categorized based on two metrics: minutes delayed and cancellation status. Among the non-canceled flights, those with 0 minutes of delay were labeled as "On-time," flights with delays up to 15 minutes were considered “Small_delay”, flights with delays ranging from 15 to 45 minutes were classified as “Medium_delay”, and flights with delays exceeding 45 minutes were categorized as “Large_delay”. Figure 3 provides a statistical overview of the number of flights for each type. (Fig. 3.)

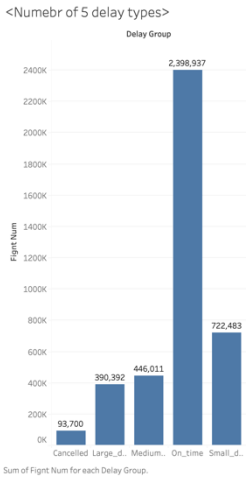


Fig. 3. Number of flights in different types.

Building upon the classification of flight types, we calculated the percentage of delays for each month (Fig. 4.). Notably, there is significant variation in the percentage of delays across the months, demonstrating that the departure month has a certain impact on the punctuality of flights. Higher percentages of abnormalities in a particular month indicate a higher likelihood of flight delays or cancellations, with July exhibiting the highest inaccurate percentage and January the lowest.

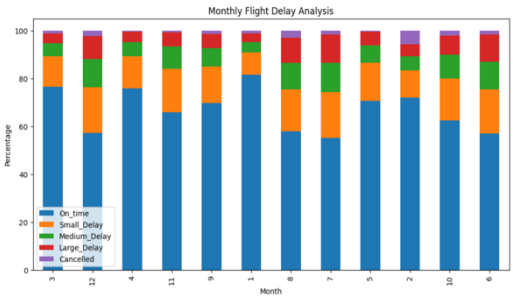


Fig. 4. Percentage of all types of flights of different departure months.

In summary, regardless of the influence of the total number of flights, July remains a peak period for flight delays, while the first quarter, especially January and February, boasts the highest punctuality rates.

3.2. Flights delay with origin airport

The efficiency of an airport is often mirrored in its flight delay statistics, and these can be heavily influenced by the volume of traffic the airport handles. In 2021, ATL, ORD, DFW, DEN, CLT, and LAX emerged as the busiest hubs, each catering to over 100,000 flights. Amongst these, DEN recorded the highest average delay at 19.6 minutes per flight, while LAX maintained a relatively swifter schedule with the shortest delay of 13.0 minutes.

When broadening the scope to include all airports, a pattern emerges: airports with the highest delay averages often have a smaller number of flights. This indicates a potential efficiency threshold, where beyond a certain number of flights, airports struggle to maintain punctuality.

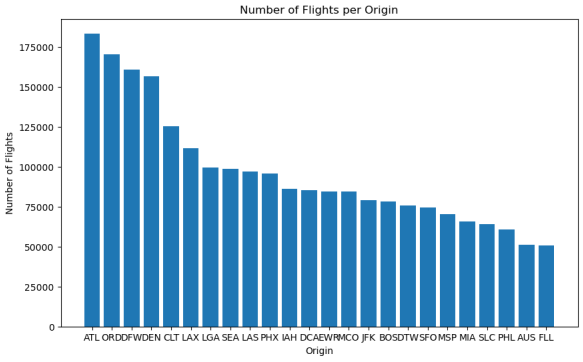


Fig. 5. Number of flights per origin airport

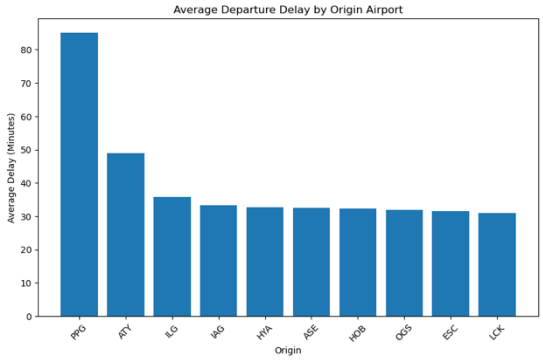


Fig. 6. Avg Delay by all airports

The classification of airport sizes is crucial to this analysis. By defining medium-sized airports

as those handling more than 0.25% of the annual flight traffic in the US, and large airports as those exceeding 1% of the traffic, we can scrutinize performance more effectively. In 2021, the thresholds for medium and large airports stood at 10,129 and 40,515 flights respectively, categorizing 41 airports as medium and 30 as large. PBI topped the above medium category with an average delay of 25.4 minutes, while EWR led the large category with 23.2 minutes.

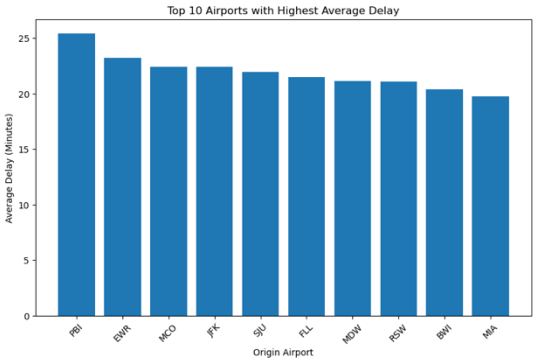


Fig. 7. Above medium size airports average delay time.

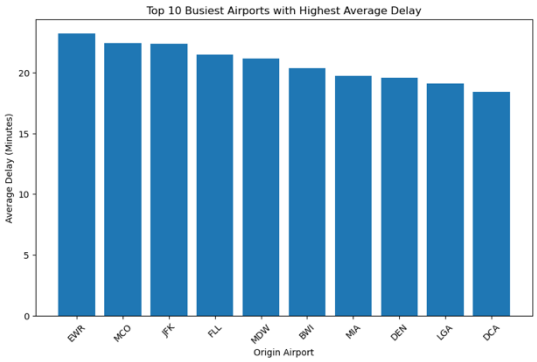


Fig. 8. Large size airports average delay time.

Delays were categorized as on-time (0 minutes), small-delay (up to 15 minutes), medium-delay (15 to 45 minutes), large-delay (over 45 minutes), and cancellations. Medium and large delays, along with cancellations, are particularly disruptive for travelers. LGA's cancellation rate was the highest among airports larger than medium size, standing at 5.92%. Furthermore, MDW had a combined delay rate of over 36%, suggesting that every third flight from this airport could face a medium or large delay.

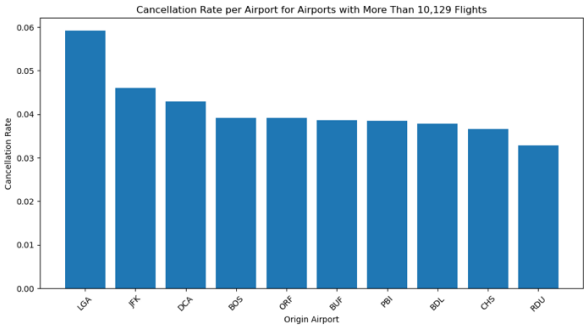


Fig. 9. Above medium size airports cancelled rate

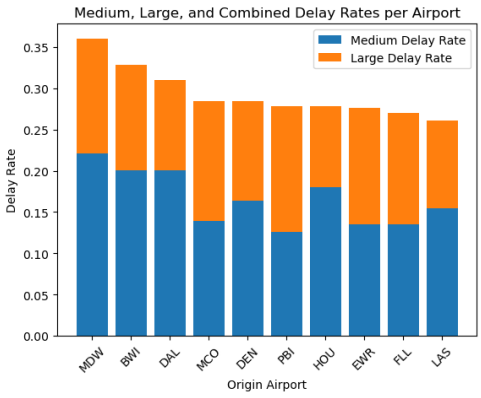


Fig. 10. Combined delay rate for above medium size

With this comprehensive analysis, stakeholders can better understand the dynamics of flight delays and the impact of airport traffic on travel punctuality.

3.3. Flights delay with airlines

In selecting flights, we often have options from various airlines, each with its own characteristics, including differing delay times. Therefore, we analyzed the flight delay situations of different airlines. First, we calculated the total delay time for each airline.

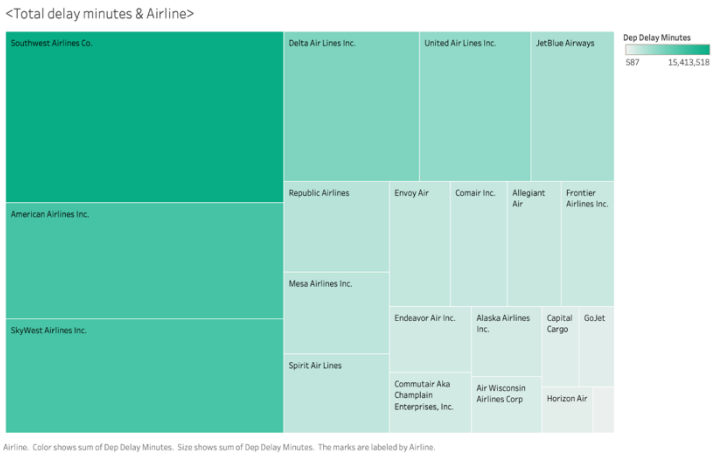


Fig. 11. Total delay minutes for different airlines

We found that Southwest Airlines, American Airlines, and SkyWest Airlines had the highest total delay times. However, the range of total delay times across different airlines was vast, from as little as 587 minutes to as much as 15,413,518 minutes. Thus, we wondered whether the long total delay time was due to the large number of flights or because the flights themselves typically have long delays.

Next, we analyzed the number of flights for different airlines.

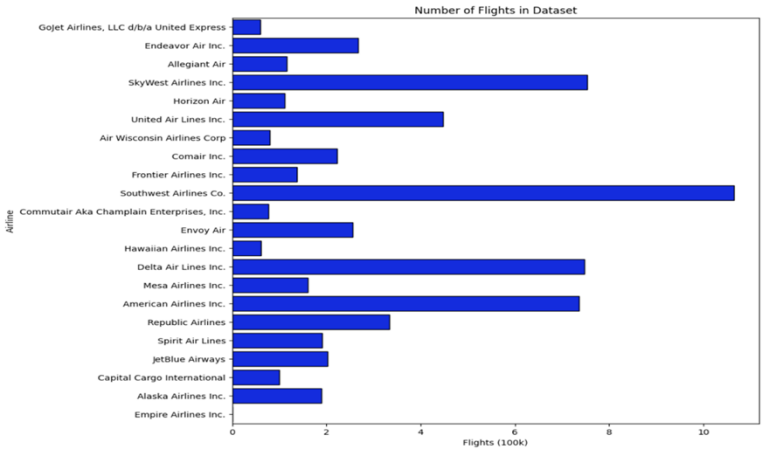


Fig. 12. The number of flights for different airlines

Interestingly, the airlines with the highest number of flights, such as SkyWest, Southwest, and American Airlines, also had the highest delay minutes. This correlation suggests that flight volume significantly impacts delay times.

Therefore, we believe it's more reasonable to represent the delay situation of different airlines in terms of delay rate.

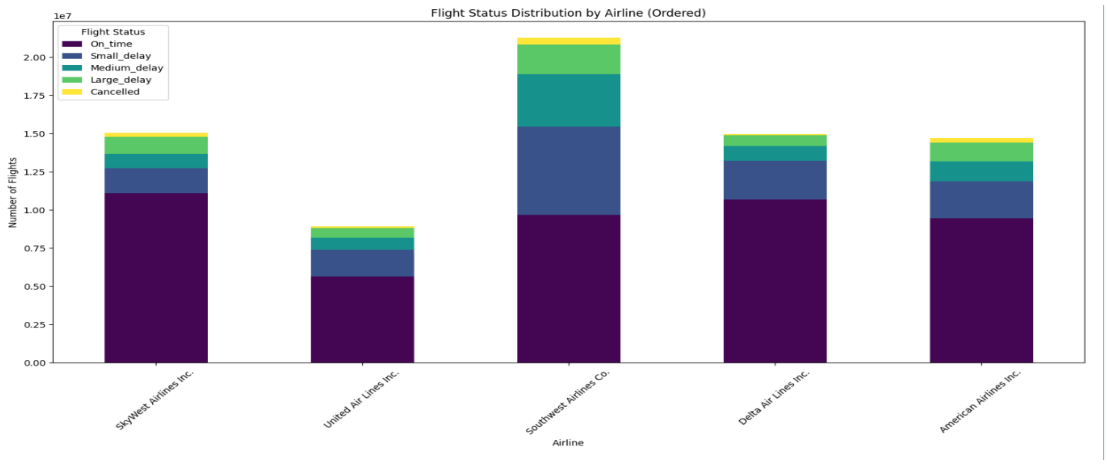


Fig. 13. The number of flights for different airlines

Based on the results from the last part, we chose five top airlines: SkyWest Airlines, United Airlines, Southwest Airlines, Delta Air Lines, and American Airlines. We found that SkyWest Airlines had the lowest delay rate among these top airlines, while Southwest Airlines had the highest delay rate, about 50 percent.

4. Results

Our comprehensive analysis of the dataset has yielded valuable insights that can significantly contribute to minimizing the occurrence of flight delays, thereby ensuring a more streamlined and predictable travel experience. This formal report outlines key recommendations derived from the data, offering actionable strategies for individuals seeking to optimize their air travel plans.

4.1 Seasonal Considerations:

The analysis indicates a strong correlation between the season and the likelihood of flight

delays. Travelers are advised to adjust their plans accordingly, with a notable emphasis on favoring winter months over summer. Specifically, our findings discourage flight departures in July, a period associated with higher delay rates. This strategic adjustment can enhance the overall reliability of travel plans.

4.2 Airport Selection:

The choice of departure airport emerges as a critical factor in minimizing delays. Optimal planning involves selecting airports with historically lower delay rates, thereby increasing the probability of punctual departures. It is imperative to exercise caution when considering airports with a documented history of higher delays, exemplified by airports mentioned in origin analysis. Prudent decision-making in this regard is pivotal to a smoother travel experience.

4.3 Airline Selection:

Our analysis underscores the significance of thoughtful airline selection. Specifically, travelers are advised to avoid carriers with high delay rates. Southwest Airlines, in particular, has exhibited a higher incidence of delays according to the data. As a result, individuals are encouraged to explore alternative airline options to mitigate the risk of disruptions to their travel schedules.

4.4 Preference for Shared Airlines:

The data suggests that opting for shared airlines over independent ones can contribute to a more efficient and reliable travel experience. The collaborative nature of shared airlines often translates to streamlined operations, potentially reducing the likelihood of delays. This recommendation aligns with a proactive approach to ensuring a smoother journey.

In conclusion, by incorporating these strategic recommendations into travel planning, individuals can proactively minimize the risk of flight delays and contribute to a more seamless and predictable travel experience. These insights serve as a valuable guide for informed

decision-making in the realm of air travel, aligning with the overarching goal of enhancing customer satisfaction and overall travel efficiency.

5. Random Forest Prediction

After exploratory data analysis, a model that can predict whether a flight will be delayed based on flight information is desired. The Random Forest Classifier, known for its robustness and efficacy in handling classification tasks, was chosen as the predictive model. The data was preprocessed to transform categorical variables into numerical representations suitable for machine learning algorithms. The feature vector was composed of the ‘Month’, ‘Distance’, ‘Shared_flights’(binary) and origin of the filght. The dataset was split into a training set (70%) and a test set (30%). Model training was executed on the training set, and predictions were made on the test set.

The Random Forest Classifier achieved a model accuracy of 83%, demonstrating a strong predictive capability. This suggests that the features selected for the model provide considerable part of information to predict delays effectively. However, it is important to recognize that accuracy alone may not fully represent the model's predictive power. Considering the imbalance that typically exists in delay datasets, where non-delayed flights are more common than delayed ones.

To fix this problem, another data set where the data points of ‘delayed’ and ‘not delayed’ are balanced, by duplicating the ‘delayed’ data points. Number of rows where 'DepDel15' is 5129191 and number of rows where 'DepDel15' is duplicated to 4297068. After fitting this dataset to the random forest model, the test MSE is still 0.17, showing that the prediction power of this model is robust.

Of course, if other important data like weather and engineering parameters are added to this model, presumably the accuracy of the model can be improved.

Flight Data Exploration

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

import pyspark
from pyspark.sql import SparkSession
from pyspark.sql import functions as F
spark = (SparkSession.builder.master("local[*]").appName("Final_Project").g
sc = spark.sparkContext
```

In [2]:
```
file_path = 'Combined_Flights_2021.csv'
df = spark.read.csv(file_path, header=True, inferSchema=True)
```

In [3]:
```
column_subset = [
    "FlightDate",
    "Airline",
    "Tail_Number",
    "Flight_Number_Marketing_Airline",
    "Origin",
    "Dest",
    "Cancelled",
    "Diverted",
    "CRSDepTime",
    "DepTime",
    "DepDelayMinutes",
    "OriginAirportID",
    "OriginCityName",
    "OriginStateName",
    "DestAirportID",
    "DestCityName",
    "DestStateName",
    "TaxiOut",
    "TaxiIn",
    "CRSArrTime",
    "ArrTime",
    "ArrDelayMinutes",
]

flight_data = df[column_subset]
type = flight_data.dtypes
describe = flight_data.describe().toPandas()
type,describe
```

Out[3]:
```
([('FlightDate', 'date'),
  ('Airline', 'string'),
  ('Tail_Number', 'string'),
  ('Flight_Number_Marketing_Airline', 'int'),
  ('Origin', 'string'),
  ('Dest', 'string'),
  ('Cancelled', 'boolean'),
  ('Diverted', 'boolean'),
  ('CRSDepTime', 'int'),
  ('DepTime', 'double'),
  ('DepDelayMinutes', 'double'),
  ('OriginAirportID', 'int'),
  ('OriginCityName', 'string'),
  ('OriginStateName', 'string'),
  ('DestAirportID', 'int'),
  ('DestCityName', 'string'),
  ('DestStateName', 'string'),
  ('TaxiOut', 'double'),
  ('TaxiIn', 'double'),
  ('CRSArrTime', 'int'),
  ('ArrTime', 'double'),
  ('ArrDelayMinutes', 'double')],
   summary                      Airline Tail_Number  \
 0   count                      6311871     6288568
 1    mean                         None        None
 2  stddev                         None        None
 3     min  Air Wisconsin Airlines Corp       206NV
 4     max        United Air Lines Inc.      n714FR

   Flight_Number_Marketing_Airline   Origin     Dest          CRSDepTime
\
 0                         6311871  6311871  6311871             6311871
 1             2795.5973735838393     None     None  1323.9526441525818
 2             1827.3750083881603     None     None  474.19715430143157
 3                               1      ABE      ABE                   1
 4                            9695      YUM      YUM                2359

               DepTime     DepDelayMinutes     OriginAirportID OriginCityN
ame  \
 0             6203546             6203458             6311871        6311
871
 1   1326.878043944544  12.761322636503705  12658.327866808431           N
one
 2  486.83304480956156  47.363188406446305  1532.7830658534601           N
one
 3                1.0                 0.0               10135   Aberdeen,
SD
 4             2400.0              3095.0               16869       Yuma,
AZ

   OriginStateName       DestAirportID  DestCityName DestStateName  \
 0         6311871             6311871       6311871       6311871
 1            None  12658.354210819582          None          None
 2            None  1532.8077801516893          None          None
 3         Alabama               10135  Aberdeen, SD       Alabama
 4         Wyoming               16869      Yuma, AZ       Wyoming

              TaxiOut             TaxiIn          CRSArrTime  \
 0            6201518            6199446             6311871
 1  16.19499838587907  7.683752870821038  1498.2723252740748
 2   8.58412054048919  6.375447172026258  494.65645535543257
```

```
3                1.0               1.0                   1
4              256.0             251.0                2400

               ArrTime    ArrDelayMinutes
0              6199463            6185870
1  1476.7584958568186  12.52927542932522
2   513.7708519497529  46.74769541798098
3                 1.0                0.0
4              2400.0             3089.0  )
```

In [4]:
```
flight_data.show(10)
```

```
+----------+--------------------+----------+--------------------------
--+------+----+---------+--------+----------+-------+--------------+-----
----------+--------------------+--------------+-------------+-----------
--------+-------------+-------+------+----------+-------+--------------+
|FlightDate|             Airline|Tail_Number|Flight_Number_Marketing_Airli
ne|Origin|Dest|Cancelled|Diverted|CRSDepTime|DepTime|DepDelayMinutes|Origi
nAirportID|      OriginCityName|OriginStateName|DestAirportID|        Dest
CityName|DestStateName|TaxiOut|TaxiIn|CRSArrTime|ArrTime|ArrDelayMinutes|
+----------+--------------------+----------+--------------------------
--+------+----+---------+--------+----------+-------+--------------+-----
----------+--------------------+--------------+-------------+-----------
--------+-------------+-------+------+----------+-------+--------------+
|2021-03-03|SkyWest Airlines ...|     N728SK|                          31
33|   SGU| PHX|    false|   false|       724|  714.0|           0.0|
14794|      St. George, UT|           Utah|        14107|         Phoenix,
AZ|      Arizona|   10.0|   5.0|       843|  818.0|           0.0|
|2021-03-03|SkyWest Airlines ...|     N752SK|                          31
34|   PHX| SGU|    false|   false|       922|  917.0|           0.0|
14107|         Phoenix, AZ|        Arizona|        14794|      St. George,
UT|         Utah|   23.0|   3.0|      1040| 1031.0|           0.0|
|2021-03-03|SkyWest Airlines ...|     N766SK|                          31
35|   MHT| ORD|    false|   false|      1330| 1321.0|           0.0|
13296|      Manchester, NH|  New Hampshire|        13930|         Chicago,
IL|     Illinois|   15.0|  16.0|      1530| 1501.0|           0.0|
|2021-03-03|SkyWest Airlines ...|     N752EV|                          31
36|   DFW| TRI|    false|   false|      1645| 1636.0|           0.0|
11298|Dallas/Fort Worth...|          Texas|        15323|Bristol/Johnson
C...|    Tennessee|   27.0|   7.0|      2010| 2002.0|           0.0|
|2021-03-03|SkyWest Airlines ...|     N715SK|                          31
37|   PHX| BFL|    false|   false|      1844| 1838.0|           0.0|
14107|         Phoenix, AZ|        Arizona|        10561|     Bakersfield,
CA|   California|   13.0|   3.0|      1925| 1903.0|           0.0|
|2021-03-03|SkyWest Airlines ...|     N744SK|                          31
39|   ORD| BNA|    false|   false|      1650| 1648.0|           0.0|
13930|         Chicago, IL|       Illinois|        10693|       Nashville,
TN|    Tennessee|   19.0|   4.0|      1834| 1808.0|           0.0|
|2021-03-03|SkyWest Airlines ...|     N727SK|                          31
40|   PSP| PHX|    false|   false|      1652| 1651.0|           0.0|
14262|    Palm Springs, CA|     California|        14107|         Phoenix,
AZ|      Arizona|   48.0|   5.0|      1902| 1929.0|          27.0|
|2021-03-03|SkyWest Airlines ...|     N771SK|                          31
41|   DFW| YUM|    false|   false|      1245| 1242.0|           0.0|
11298|Dallas/Fort Worth...|          Texas|        16218|            Yuma,
AZ|      Arizona|   32.0|   5.0|      1456| 1452.0|           0.0|
|2021-03-03|SkyWest Airlines ...|     N614SK|                          31
42|   LBB| PHX|    false|   false|       726|  717.0|           0.0|
12896|         Lubbock, TX|          Texas|        14107|         Phoenix,
AZ|      Arizona|   12.0|   8.0|       836|  821.0|           0.0|
|2021-03-03|SkyWest Airlines ...|     N773SK|                          31
44|   DFW| DRO|    false|   false|      2045| 2040.0|           0.0|
11298|Dallas/Fort Worth...|          Texas|        11413|         Durango,
CO|     Colorado|   25.0|   2.0|      2215| 2144.0|           0.0|
+----------+--------------------+----------+--------------------------
--+------+----+---------+--------+----------+-------+--------------+-----
----------+--------------------+--------------+-------------+-----------
--------+-------------+-------+------+----------+-------+--------------+
only showing top 10 rows
```

In [5]:
```
from pyspark.sql import SQLContext
sqlContext = SQLContext(sc)
flight_data.registerTempTable('flight')
```

```
C:\Users\jingu\anaconda3\envs\pyspark\Lib\site-packages\pyspark\sql\contex
t.py:112: FutureWarning: Deprecated in 3.0.0. Use SparkSession.builder.get
OrCreate() instead.
  warnings.warn(
C:\Users\jingu\anaconda3\envs\pyspark\Lib\site-packages\pyspark\sql\datafr
ame.py:330: FutureWarning: Deprecated in 2.0, use createOrReplaceTempView
instead.
  warnings.warn("Deprecated in 2.0, use createOrReplaceTempView instead.",
FutureWarning)
```

In [44]:
```
sqlContext.sql('''SELECT count(*) from flight
                ''').show()
```

```
+--------+
|count(1)|
+--------+
| 6311871|
+--------+
```

In [6]:
```
sqlContext.sql('''SELECT Airline, Tail_Number, DepDelayMinutes
               FROM flight
               WHERE DepDelayMinutes < 30
                ''').show()

#Distribution plot can be drawn by this.
```

```
+--------------------+-----------+---------------+
|             Airline|Tail_Number|DepDelayMinutes|
+--------------------+-----------+---------------+
|SkyWest Airlines ...|     N728SK|            0.0|
|SkyWest Airlines ...|     N752SK|            0.0|
|SkyWest Airlines ...|     N766SK|            0.0|
|SkyWest Airlines ...|     N752EV|            0.0|
|SkyWest Airlines ...|     N715SK|            0.0|
|SkyWest Airlines ...|     N744SK|            0.0|
|SkyWest Airlines ...|     N727SK|            0.0|
|SkyWest Airlines ...|     N771SK|            0.0|
|SkyWest Airlines ...|     N614SK|            0.0|
|SkyWest Airlines ...|     N773SK|            0.0|
|SkyWest Airlines ...|     N762SK|            2.0|
|SkyWest Airlines ...|     N730SK|            0.0|
|SkyWest Airlines ...|     N741EV|            8.0|
|SkyWest Airlines ...|     N716SK|            0.0|
|SkyWest Airlines ...|     N631SK|            0.0|
|SkyWest Airlines ...|     N772SK|            0.0|
|SkyWest Airlines ...|     N716SK|            0.0|
|SkyWest Airlines ...|     N732SK|            0.0|
|SkyWest Airlines ...|     N771SK|            0.0|
|SkyWest Airlines ...|     N709SK|            0.0|
+--------------------+-----------+---------------+
only showing top 20 rows
```

Delays are divided into three categories, namely "on time or small delay" (up to 15 minutes delay), "Medium delay" (15 – 45 minutes delay) and "Large delay" (45 minutes delay).

In [7]:
```
from pyspark.sql.functions import when

flight_data_with_group =  flight_data.withColumn(
    "Delay_Group",
    when(flight_data["DepDelayMinutes"] == 0, "On_time")
    .when(flight_data["DepDelayMinutes"] <= 15, "Small_delay")
    .when((flight_data["DepDelayMinutes"] > 15) & (flight_data["DepDelayMin
    .when(flight_data["DepDelayMinutes"] > 45, "Large_delay")
    .when(flight_data["Cancelled"] == True, "Cancelled")
)

flight_data_with_group.show(10)
```

```
+----------+--------------------+----------+----------------------------
--+------+----+---------+--------+----------+-------+--------------+-----
----------+--------------------+--------------+------------+-----------
--------+-------------+-------+------+----------+-------+--------------+-
----------+
|FlightDate|             Airline|Tail_Number|Flight_Number_Marketing_Airli
ne|Origin|Dest|Cancelled|Diverted|CRSDepTime|DepTime|DepDelayMinutes|Origi
nAirportID|      OriginCityName|OriginStateName|DestAirportID|        Dest
CityName|DestStateName|TaxiOut|TaxiIn|CRSArrTime|ArrTime|ArrDelayMinutes|D
elay_Group|
+----------+--------------------+----------+----------------------------
--+------+----+---------+--------+----------+-------+--------------+-----
----------+--------------------+--------------+------------+-----------
--------+-------------+-------+------+----------+-------+--------------+-
----------+
|2021-03-03|SkyWest Airlines ...|     N728SK|                          31
33|   SGU| PHX|    false|   false|       724|  714.0|           0.0|
14794|      St. George, UT|           Utah|        14107|         Phoenix,
AZ|      Arizona|   10.0|   5.0|       843|  818.0|           0.0|    On_
time|
|2021-03-03|SkyWest Airlines ...|     N752SK|                          31
34|   PHX| SGU|    false|   false|       922|  917.0|           0.0|
14107|         Phoenix, AZ|        Arizona|        14794|      St. George,
UT|         Utah|   23.0|   3.0|      1040| 1031.0|           0.0|    On_
time|
|2021-03-03|SkyWest Airlines ...|     N766SK|                          31
35|   MHT| ORD|    false|   false|      1330| 1321.0|           0.0|
13296|      Manchester, NH|  New Hampshire|        13930|         Chicago,
IL|     Illinois|   15.0|  16.0|      1530| 1501.0|           0.0|    On_
time|
|2021-03-03|SkyWest Airlines ...|     N752EV|                          31
36|   DFW| TRI|    false|   false|      1645| 1636.0|           0.0|
11298|Dallas/Fort Worth...|          Texas|        15323|Bristol/Johnson
C...|    Tennessee|   27.0|   7.0|      2010| 2002.0|           0.0|    O
n_time|
|2021-03-03|SkyWest Airlines ...|     N715SK|                          31
37|   PHX| BFL|    false|   false|      1844| 1838.0|           0.0|
14107|         Phoenix, AZ|        Arizona|        10561|     Bakersfield,
CA|   California|   13.0|   3.0|      1925| 1903.0|           0.0|    On_
time|
|2021-03-03|SkyWest Airlines ...|     N744SK|                          31
39|   ORD| BNA|    false|   false|      1650| 1648.0|           0.0|
13930|         Chicago, IL|       Illinois|        10693|       Nashville,
TN|    Tennessee|   19.0|   4.0|      1834| 1808.0|           0.0|    On_
time|
|2021-03-03|SkyWest Airlines ...|     N727SK|                          31
40|   PSP| PHX|    false|   false|      1652| 1651.0|           0.0|
14262|    Palm Springs, CA|     California|        14107|         Phoenix,
AZ|      Arizona|   48.0|   5.0|      1902| 1929.0|          27.0|    On_
time|
|2021-03-03|SkyWest Airlines ...|     N771SK|                          31
41|   DFW| YUM|    false|   false|      1245| 1242.0|           0.0|
11298|Dallas/Fort Worth...|          Texas|        16218|            Yuma,
AZ|      Arizona|   32.0|   5.0|      1456| 1452.0|           0.0|    On_
time|
|2021-03-03|SkyWest Airlines ...|     N614SK|                          31
42|   LBB| PHX|    false|   false|       726|  717.0|           0.0|
12896|         Lubbock, TX|          Texas|        14107|         Phoenix,
AZ|      Arizona|   12.0|   8.0|       836|  821.0|           0.0|    On_
time|
|2021-03-03|SkyWest Airlines ...|     N773SK|                          31
```

```
44|   DFW| DRO|    false|   false|      2045| 2040.0|            0.0|
11298|Dallas/Fort Worth...|          Texas|        11413|         Durango,
CO|     Colorado|   25.0|   2.0|      2215| 2144.0|            0.0|    On_
time|
+----------+--------------------+-----------+-----------------------------
--+------+----+---------+--------+----------+-------+---------------+-----
----------+--------------------+--------------+-------------+------------
--------+-------------+-------+------+----------+-------+---------------+-
----------+
only showing top 10 rows
```

In [8]:
```
flight_data_with_group.registerTempTable('flight_group')

sqlContext.sql('''SELECT COUNT(Tail_Number) AS Flight_Num, Delay_Group
               FROM flight_group
               GROUP BY Delay_Group
               ORDER BY Flight_Num DESC
                ''').show()

#The numbuer of flight in each delay group
```

```
+----------+------------+
|Flight_Num| Delay_Group|
+----------+------------+
|   4147768|     On_time|
|   1019416| Small_delay|
|    563238|Medium_delay|
|    473036| Large_delay|
|     85110|   Cancelled|
+----------+------------+
```

In [9]:
```
sqlContext.sql('''SELECT
               Delay_Group,
               COUNT(Tail_Number) AS Flight_Num,
               (COUNT(Tail_Number) / SUM(COUNT(Tail_Number)) OVER ()) * 100
               FROM flight_group
               GROUP BY Delay_Group
               ORDER BY Flight_Num DESC
                ''').show()

#Percentage of each group, can draw a pie chart based on that.
```

```
+------------+----------+------------------+
| Delay_Group|Flight_Num|        Percentage|
+------------+----------+------------------+
|     On_time|   4147768| 65.95727357961304|
| Small_delay|   1019416|16.210622195704968|
|Medium_delay|    563238| 8.956538277076753|
| Large_delay|    473036| 7.522157667691595|
|   Cancelled|     85110|1.3534082799136464|
+------------+----------+------------------+
```

In [10]:
```
sqlContext.sql('''SELECT Airline, count(Tail_Number) as Flight_Num
                FROM flight_group
                GROUP BY Airline
                ORDER BY Flight_Num DESC
                 ''').show()

#The numbuer of delayed flight in each airline.
#We can split it more specific with 'group by delay group'.
#See chart 'https://www.kaggle.com/code/robikscube/flight-delay-exploratory
```

```
+--------------------+----------+
|             Airline|Flight_Num|
+--------------------+----------+
|Southwest Airline...|   1061633|
|SkyWest Airlines ...|    753305|
|Delta Air Lines Inc.|    747859|
|American Airlines...|    735991|
|United Air Lines ...|    440872|
|   Republic Airlines|    332646|
|   Endeavor Air Inc.|    266862|
|           Envoy Air|    255533|
|         Comair Inc.|    222385|
|     JetBlue Airways|    202687|
|    Spirit Air Lines|    191361|
|Alaska Airlines Inc.|    188955|
|  Mesa Airlines Inc.|    159005|
|Frontier Airlines...|    135281|
|       Allegiant Air|    112764|
|         Horizon Air|    111628|
|Capital Cargo Int...|     99188|
|Air Wisconsin Air...|     77943|
|Commutair Aka Cha...|     74219|
|Hawaiian Airlines...|     60293|
+--------------------+----------+
only showing top 20 rows
```

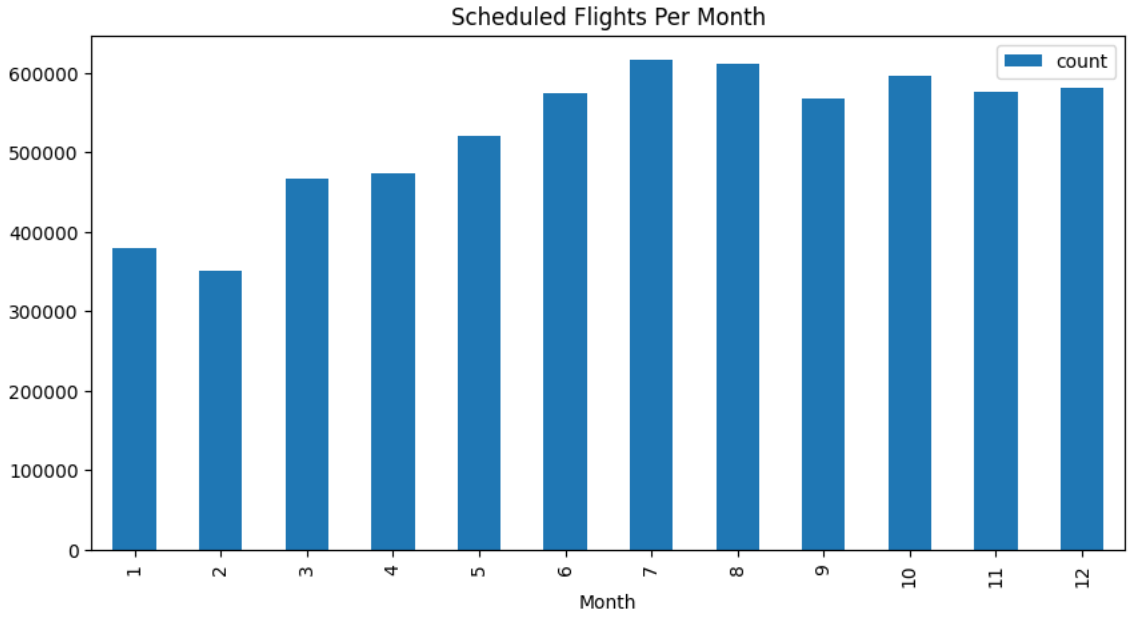
In [11]:
```
sqlContext.sql('''SELECT Airline, count(Tail_Number) as Flight_Cancelled
                  FROM flight_group
                  WHERE Cancelled = 'True'
                  GROUP BY Airline
                  ORDER BY Flight_Cancelled DESC
                  ''').show()
```

```
+--------------------+----------------+
|             Airline|Flight_Cancelled|
+--------------------+----------------+
|Southwest Airline...|           20498|
|American Airlines...|           16070|
|SkyWest Airlines ...|           13285|
|    Spirit Air Lines|            5661|
|           Envoy Air|            5468|
|   Republic Airlines|            5367|
|  Mesa Airlines Inc.|            3979|
|Delta Air Lines Inc.|            3492|
|     JetBlue Airways|            3284|
|Alaska Airlines Inc.|            3217|
|         Horizon Air|            2096|
|         Comair Inc.|            2037|
|Capital Cargo Int...|            1180|
|   Endeavor Air Inc.|             969|
|       Allegiant Air|             712|
|United Air Lines ...|             142|
|Air Wisconsin Air...|              96|
|Frontier Airlines...|              68|
|Commutair Aka Cha...|              42|
|GoJet Airlines, L...|              37|
+--------------------+----------------+
only showing top 20 rows
```

Scheduled Flights Per Month

In [12]:
```
df=flight_data
```

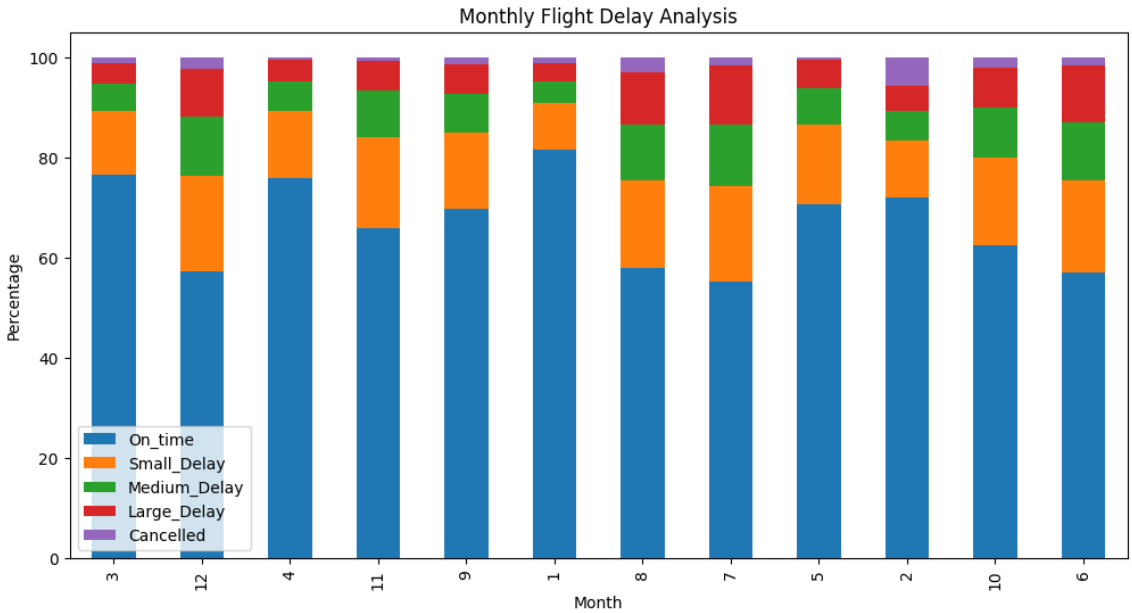
In [13]:
```
from pyspark.sql import SparkSession
from pyspark.sql.functions import month
import matplotlib.pyplot as plt

flight_data2 = flight_data.withColumn("Month", month("FlightDate"))

month_counts = flight_data2.groupBy("Month").count().orderBy("Month")

pdf = month_counts.toPandas()

pdf.plot(kind="bar", x='Month', y='count', figsize=(10, 5), title="Schedule
plt.show()
```



Monthly Flight Delay Analysis

In [14]:
```
from pyspark.sql import SparkSession
from pyspark.sql.functions import month, col, count, sum
from pyspark.sql import functions as F

flight_data_with_group2 = flight_data_with_group.withColumn("Month", month(

df_agg = flight_data_with_group2.groupBy("Month", "Delay_Group").count().wi

total_flights_per_month = flight_data_with_group2.groupBy("Month").count().

df_agg = df_agg.join(total_flights_per_month, "Month")

df_agg = df_agg.withColumn("Percentage", (col("group_count") / col("total_c

df_pivot = df_agg.groupBy("Month").pivot("Delay_Group").sum("Percentage")

col_order = ["On_time", "Small_Delay", "Medium_Delay", "Large_Delay", "Canc
df_final = df_pivot.select(["Month"] + col_order)
```

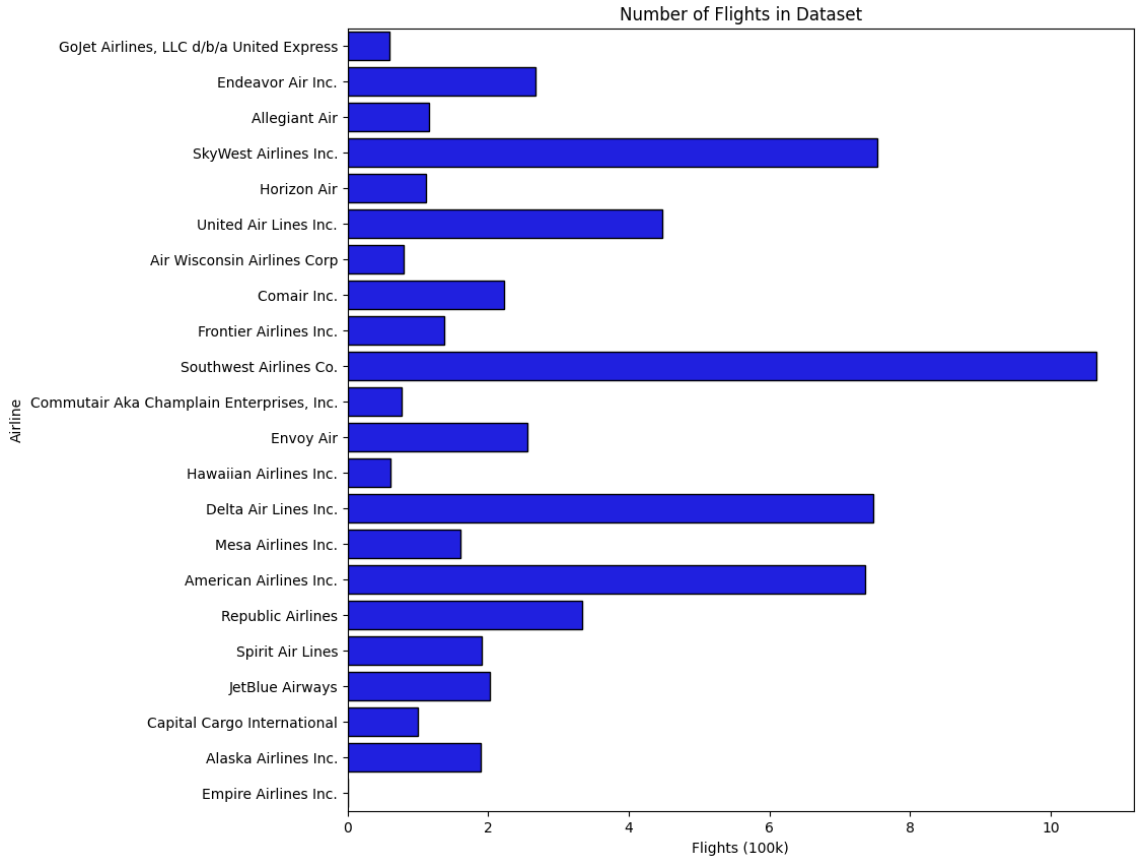
In [15]:
```
import matplotlib.pyplot as plt

df_pandas = df_final.toPandas()
df_pandas.plot(kind='bar', x='Month', stacked=True, figsize=(12, 6))
plt.title("Monthly Flight Delay Analysis")
plt.xlabel("Month")
plt.ylabel("Percentage")
plt.show()
```



Airline

Number of Flight in Dataset

In [17]:
```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col
import matplotlib.pyplot as plt
import seaborn as sns
airline_counts = flight_data_with_group2.groupBy("Airline").count()
airline_counts_pandas = airline_counts.toPandas()
airline_counts_pandas['count'] = airline_counts_pandas['count'] / 100_000
fig, ax = plt.subplots(figsize=(10, 10))
sns.barplot(x='count', y='Airline', data=airline_counts_pandas, ax=ax, colo
ax.set_title("Number of Flights in Dataset")
ax.set_xlabel("Flights (100k)")

plt.show()
```



Top Airlines Flight Status Distribution

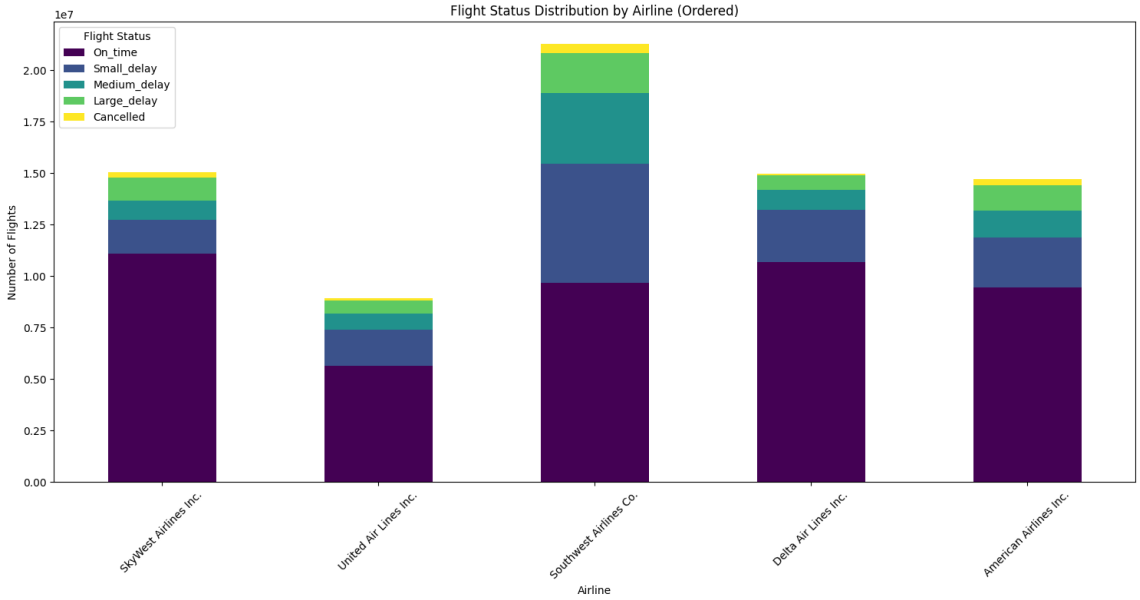
In [18]:
```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, when, count, lit
import matplotlib.pyplot as plt
import pandas as pd

airline_counts = flight_data_with_group2.groupBy("Airline").count()
top_airlines =['SkyWest Airlines Inc.','Southwest Airlines Co.','American A


df_top = flight_data_with_group2.filter(col("Airline").isin(top_airlines))


df_agg = df_top.groupBy("Airline").pivot("Delay_Group").count()

total_count = df_agg.select([count(lit(1)).alias('total')]).collect()[0]['t
for col_name in df_agg.columns[1:]:
    df_agg = df_agg.withColumn(col_name, (col(col_name) / total_count) * 10

df_agg_pandas = df_agg.toPandas().set_index("Airline")
df_agg_pandas
```

Out[18]:

Airline	Cancelled	Large_delay	Medium_delay	On_time	Small_delay
SkyWest Airlines Inc.	262400.0	1146640.0	937560.0	11107060.0	1613200.0
United Air Lines Inc.	119300.0	630560.0	782640.0	5656100.0	1748140.0
Southwest Airlines Co.	466780.0	1935680.0	3433040.0	9684820.0	5772480.0
Delta Air Lines Inc.	70500.0	684300.0	987440.0	10694460.0	2523260.0
American Airlines Inc.	322000.0	1235660.0	1282560.0	9452680.0	2435080.0

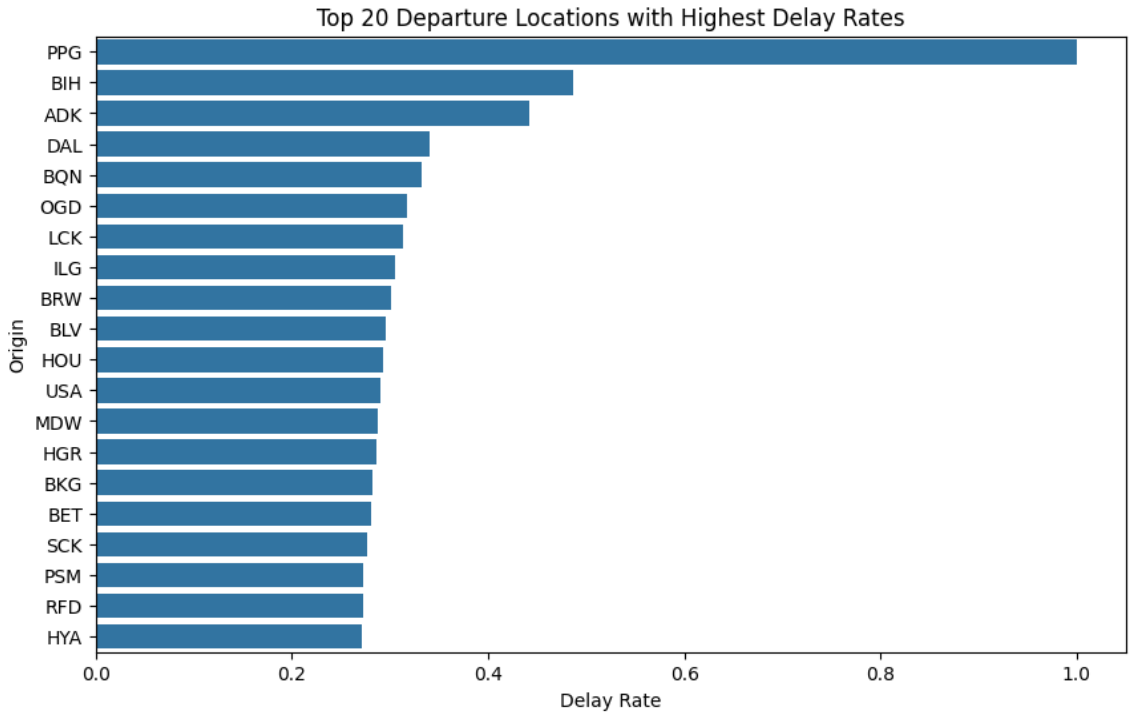
In [25]:

```
ordered_columns = ["On_time", "Small_delay", "Medium_delay", "Large_delay",
df_ordered = df_agg_pandas[ordered_columns]
ax = df_ordered.plot(kind='bar', stacked=True, figsize=(15, 8), colormap='v
plt.title('Flight Status Distribution by Airline (Ordered)')
plt.xlabel('Airline')
plt.ylabel('Number of Flights')
plt.xticks(rotation=45)
plt.legend(title='Flight Status', loc='upper left')
plt.tight_layout()

plt.show()
```



Departure Locations Analysis

In [20]:
```
delays = flight_data_with_group2.withColumn("IsDelayed", when(col("DepDelay
delay_counts = delays.groupBy("Origin").agg(
    count("FlightDate").alias("TotalFlights"),
    sum("IsDelayed").alias("DelayedFlights")
)

delay_rates = delay_counts.withColumn("DelayRate", col("DelayedFlights") /

delay_rates_pandas = delay_rates.toPandas()

delay_rates_pandas = delay_rates_pandas.sort_values(by="DelayRate", ascendi
plt.figure(figsize=(10, 6))
sns.barplot(x="DelayRate", y="Origin", data=delay_rates_pandas.head(20))
plt.title("Top 20 Departure Locations with Highest Delay Rates")
plt.xlabel("Delay Rate")
plt.ylabel("Origin")
plt.show()
```



In [33]:
```
delay_rates_pandas = delay_counts.toPandas()
```

In [40]:
```
delay_rates_pandas
```

Out[40]:

	Origin	TotalFlights	DelayedFlights
0	BGM	365	29
1	INL	642	67
2	DLG	480	93
3	MSY	37634	6506
4	GEG	19693	2260
...	...	...	...
375	ACK	2122	471
376	MVY	1011	260
377	LNY	28	4
378	MKK	28	3
379	GST	89	7

380 rows × 3 columns

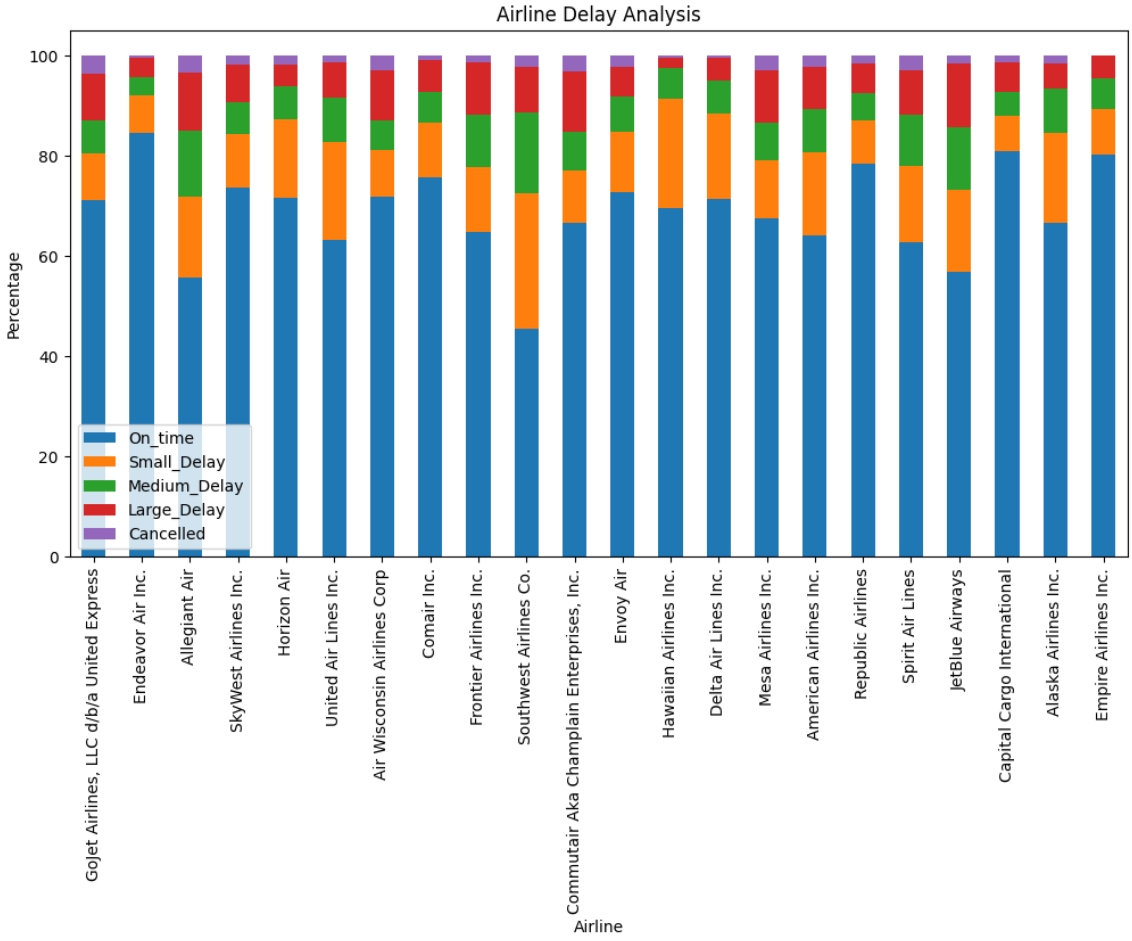
In [42]:
```
ppg_flights = delay_rates_pandas[delay_rates_pandas['Origin'] == 'PPG']
ppg_flights
```

Out[42]:

	Origin	TotalFlights	DelayedFlights
369	PPG	6	6

Airline Delay Analysis

In [21]:
```
df_agg2 = flight_data_with_group2.groupBy("Airline", "Delay_Group").count()

total_flights_per_airline = flight_data_with_group2.groupBy("Airline").coun

df_agg2 = df_agg2.join(total_flights_per_airline, "Airline")

df_agg2 = df_agg2.withColumn("Percentage_airline", (col("group_count") / co

df_pivot2 = df_agg2.groupBy("Airline").pivot("Delay_Group").sum("Percentage


df_final2 = df_pivot2.select(["Airline"] + col_order)


df_pandas = df_final2.toPandas()
df_pandas.plot(kind='bar', x='Airline', stacked=True, figsize=(12, 6))
plt.title("Airline Delay Analysis")
plt.xlabel("Airline")
plt.ylabel("Percentage")
plt.show()
```



In [ ]:
```
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Flight Origin Airport Exploration

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

import pyspark
from pyspark.sql import SparkSession
from pyspark.sql import functions as F
spark = (SparkSession.builder.master("local[*]").appName("Final_Project")
sc = spark.sparkContext
```

```
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setL
ogLevel(newLevel).
23/12/07 21:33:48 WARN NativeCodeLoader: Unable to load native-hadoop libr
ary for your platform... using builtin-java classes where applicable
23/12/07 21:33:49 WARN Utils: Service 'SparkUI' could not bind on port 404
0. Attempting port 4041.
```

In [ ]:
```
file_path = 'Combined_Flights_2022.csv'
df = spark.read.csv(file_path, header=True, inferSchema=True)
```

In [ ]:
```
column_subset = [
    "FlightDate",
    "Airline",
    "Tail_Number",
    "Flight_Number_Marketing_Airline",
    "Origin",
    "Dest",
    "Cancelled",
    "Diverted",
    "CRSDepTime",
    "DepTime",
    "DepDelayMinutes",
    "OriginAirportID",
    "OriginCityName",
    "OriginStateName",
    "DestAirportID",
    "DestCityName",
    "DestStateName",
    "TaxiOut",
    "TaxiIn",
    "CRSArrTime",
    "ArrTime",
    "ArrDelayMinutes",
]

flight_data = df[column_subset]
type = flight_data.dtypes
describe = flight_data.describe().toPandas()
type,describe
```

```
23/12/07 21:33:57 WARN package: Truncated the string representation of a p
lan since it was too large. This behavior can be adjusted by setting 'spar
k.sql.debug.maxToStringFields'.
```

Out[ ]:
```
([('FlightDate', 'date'),
  ('Airline', 'string'),
  ('Tail_Number', 'string'),
  ('Flight_Number_Marketing_Airline', 'int'),
  ('Origin', 'string'),
  ('Dest', 'string'),
  ('Cancelled', 'boolean'),
  ('Diverted', 'boolean'),
  ('CRSDepTime', 'int'),
  ('DepTime', 'double'),
  ('DepDelayMinutes', 'double'),
  ('OriginAirportID', 'int'),
  ('OriginCityName', 'string'),
  ('OriginStateName', 'string'),
  ('DestAirportID', 'int'),
  ('DestCityName', 'string'),
  ('DestStateName', 'string'),
  ('TaxiOut', 'double'),
  ('TaxiIn', 'double'),
  ('CRSArrTime', 'int'),
  ('ArrTime', 'double'),
  ('ArrDelayMinutes', 'double')],
   summary                      Airline Tail_Number  \
 0   count                      4078318     4051523
 1    mean                         None        None
 2  stddev                         None        None
 3     min  Air Wisconsin Airlines Corp       202NV
 4     max        United Air Lines Inc.       NT200

   Flight_Number_Marketing_Airline   Origin     Dest          CRSDepTime
\
 0                         4078318  4078318  4078318             4078318
 1              2562.1153048879464     None     None  1329.5866648946944
 2              1745.8262843514103     None     None   490.4801248134615
 3                               1      ABE      ABE                   1
 4                            9680      YUM      YUM                2359

                DepTime     DepDelayMinutes     OriginAirportID OriginCit
yName  \
 0              3957885             3957823             4078318        40
78318
 1  1334.3739312789533  16.014938010113134  12659.939662625622
None
 2   505.6219331420918  52.314976096088245  1522.7127357547113
None
 3                 1.0                 0.0               10135   Aberdee
n, SD
 4              2400.0              7223.0               16869       Yum
a, AZ

   OriginStateName       DestAirportID  DestCityName DestStateName  \
 0         4078318             4078318       4078318       4078318
```

```
1            None  12659.897625197447          None          None
2            None  1522.7183178838786          None          None
3         Alabama               10135  Aberdeen, SD       Alabama
4         Wyoming               16869      Yuma, AZ       Wyoming

              TaxiOut             TaxiIn         CRSArrTime  \
0             3955652            3954076            4078318
1  16.973750218674443   7.89438695664929  1486.057737528069
2    9.49540682631691  6.663118162768139  518.5077759377792
3                 1.0                1.0                  1
4               221.0              290.0               2359

              ArrTime     ArrDelayMinutes
0             3954079             3944916
1  1457.8860179576584  15.783071426615928
2   543.1841060487743  51.984235813139364
3                 1.0                 0.0
4              2400.0              7232.0  )
```

In [ ]:
```
flight_data.show(10)
```

```
+---------+-------------------+----------+--------------------------
--+------+----+---------+--------+---------+-------+--------------+-----
---------+-------------------+--------------+------------+-----------
-------+------------+-------+------+---------+-------+--------------+
|FlightDate|            Airline|Tail_Number|Flight_Number_Marketing_Airli
ne|Origin|Dest|Cancelled|Diverted|CRSDepTime|DepTime|DepDelayMinutes|Origi
nAirportID|      OriginCityName|OriginStateName|DestAirportID|       DestC
ityName|DestStateName|TaxiOut|TaxiIn|CRSArrTime|ArrTime|ArrDelayMinutes|
+---------+-------------------+----------+--------------------------
--+------+----+---------+--------+---------+-------+--------------+-----
---------+-------------------+--------------+------------+-----------
-------+------------+-------+------+---------+-------+--------------+
|2022-04-04|Commutair Aka Cha...|     N21144|                          43
01|   GJT| DEN|    false|   false|      1133| 1123.0|           0.0|
11921|  Grand Junction, CO|       Colorado|        11292|         Denver,
CO|     Colorado|   17.0|   8.0|      1245| 1228.0|           0.0|
|2022-04-04|Commutair Aka Cha...|     N16170|                          42
99|   HRL| IAH|    false|   false|       732|  728.0|           0.0|
12206|Harlingen/San Ben...|          Texas|        12266|        Houston,
TX|        Texas|   16.0|   9.0|       849|  848.0|           0.0|
|2022-04-04|Commutair Aka Cha...|     N21144|                          42
98|   DRO| DEN|    false|   false|      1529| 1514.0|           0.0|
11413|         Durango, CO|       Colorado|        11292|         Denver,
CO|     Colorado|   21.0|  14.0|      1639| 1636.0|           0.0|
|2022-04-04|Commutair Aka Cha...|     N11184|                          42
96|   IAH| GPT|    false|   false|      1435| 1430.0|           0.0|
12266|         Houston, TX|          Texas|        11973|Gulfport/Biloxi,
MS|  Mississippi|   16.0|   4.0|      1605| 1547.0|           0.0|
|2022-04-04|Commutair Aka Cha...|     N17146|                          42
95|   DRO| DEN|    false|   false|      1135| 1135.0|           0.0|
11413|         Durango, CO|       Colorado|        11292|         Denver,
CO|     Colorado|   19.0|   8.0|      1245| 1251.0|           6.0|
|2022-04-04|Commutair Aka Cha...|     N11191|                          42
94|   DEN| TUL|    false|   false|       955|  952.0|           0.0|
11292|          Denver, CO|       Colorado|        15370|          Tulsa,
OK|     Oklahoma|   25.0|   4.0|      1240| 1238.0|           0.0|
```

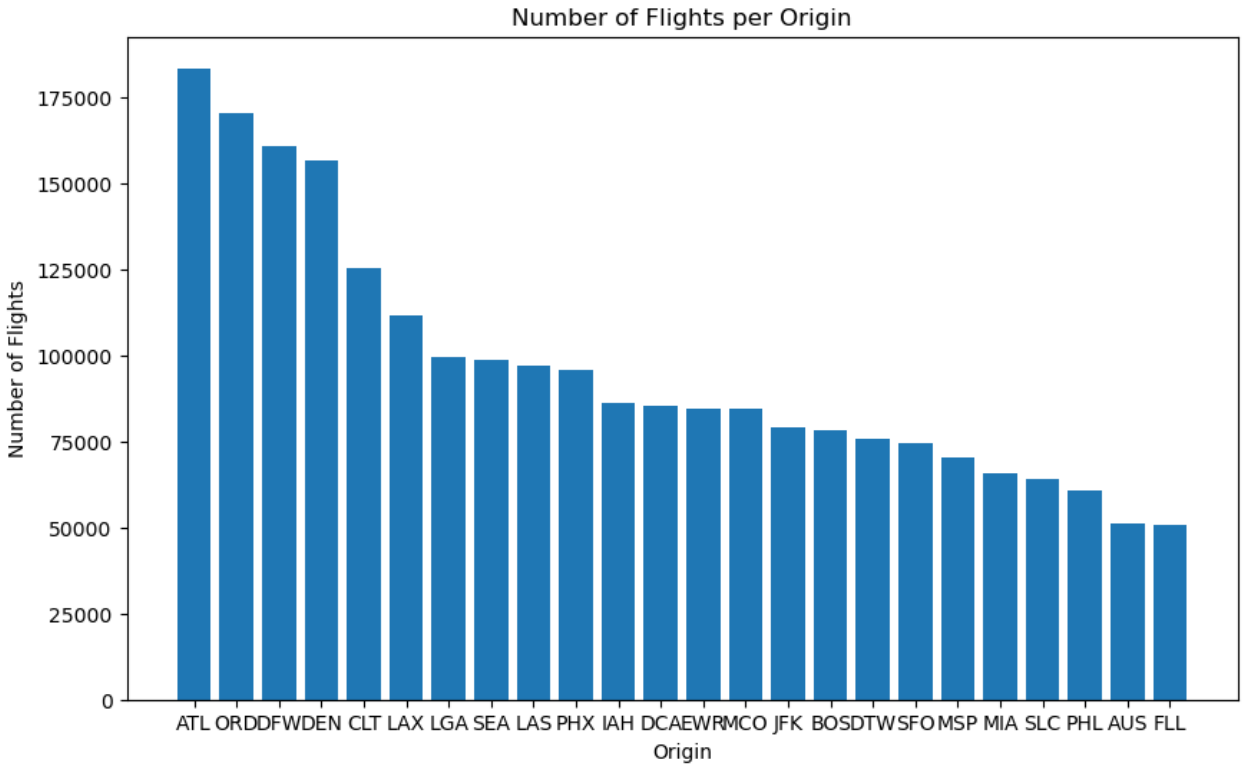
```
|2022-04-04|Commutair Aka Cha...|     N14143|                           42
93|   IAH| LCH|    false|   false|      2139| 2136.0|            0.0|
12266|         Houston, TX|          Texas|        12915|   Lake Charles,
LA|    Louisiana|   11.0|   5.0|      2231| 2218.0|            0.0|
|2022-04-04|Commutair Aka Cha...|     N13124|                           42
92|   TYS| IAH|    false|   false|      1129| 1117.0|            0.0|
15412|       Knoxville, TN|      Tennessee|        12266|        Houston,
TX|        Texas|   22.0|  16.0|      1306| 1311.0|            5.0|
|2022-04-04|Commutair Aka Cha...|     N33182|                           42
91|   IAH| AEX|    false|   false|      1424| 1414.0|            0.0|
12266|         Houston, TX|          Texas|        10185|     Alexandria,
LA|    Louisiana|   16.0|   6.0|      1524| 1513.0|            0.0|
|2022-04-04|Commutair Aka Cha...|     N21154|                           42
90|   IAH| MOB|    false|   false|       954|  947.0|            0.0|
12266|         Houston, TX|          Texas|        13422|         Mobile,
AL|      Alabama|   17.0|   6.0|      1121| 1110.0|            0.0|
+----------+--------------------+-----------+-----------------------------
--+------+----+---------+--------+----------+-------+--------------+-----
----------+--------------------+---------------+-------------+-----------
-------+-------------+-------+------+----------+-------+--------------+
only showing top 10 rows
```

In [ ]:
```
from pyspark.sql import SQLContext
sqlContext = SQLContext(sc)
flight_data.registerTempTable('flight')
```

```
/Users/evan/anaconda3/lib/python3.11/site-packages/pyspark/sql/context.py:
112: FutureWarning: Deprecated in 3.0.0. Use SparkSession.builder.getOrCre
ate() instead.
  warnings.warn(
/Users/evan/anaconda3/lib/python3.11/site-packages/pyspark/sql/dataframe.p
y:330: FutureWarning: Deprecated in 2.0, use createOrReplaceTempView inste
ad.
  warnings.warn("Deprecated in 2.0, use createOrReplaceTempView instead.",
FutureWarning)
```

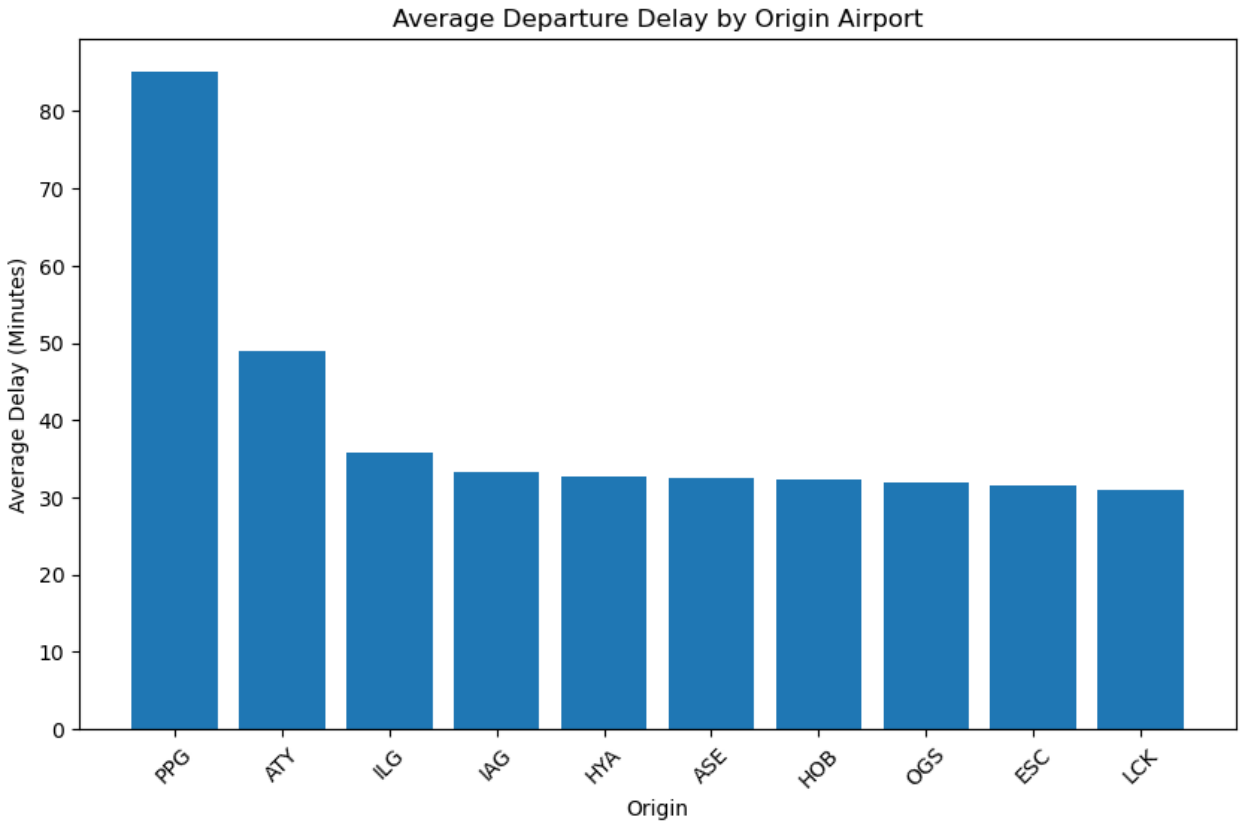
In [ ]:
```
result_1 = sqlContext.sql('''SELECT Origin, count(Tail_Number) as num, av
               FROM flight
               GROUP BY Origin
               having num > 50000
               ORDER BY num DESC
               ''')

result_1_pd = result_1.toPandas()
```

In [ ]:
```
plt.figure(figsize=(10,6))
plt.bar(result_1_pd['Origin'], result_1_pd['num'])
plt.xlabel('Origin')
plt.ylabel('Number of Flights')
plt.title('Number of Flights per Origin')
plt.show()
```



In [ ]:
```
result_2 = sqlContext.sql('''SELECT Origin, count(Tail_Number) as num, av
                FROM flight
                GROUP BY Origin
                ORDER BY avg DESC
                            LIMIT 10
                ''')
result_2_pd = result_2.toPandas()
```

In [ ]:
```
plt.figure(figsize=(10,6))
plt.bar(result_2_pd['Origin'], result_2_pd['avg'])
plt.xlabel('Origin')
plt.ylabel('Average Delay (Minutes)')
plt.title('Average Departure Delay by Origin Airport')
plt.xticks(rotation=45)
plt.show()
```



In [ ]:
```
sqlContext.sql('''SELECT count(Tail_Number) AS Total_Flight
                FROM flight
                 ''').show()
```

```
[Stage 17:=================================================>     (10 + 1)
/ 11]
+------------+
|Total_Flight|
+------------+
|     4051523|
+------------+
```

In [ ]:
```
sqlContext.sql('''SELECT count(Origin) as medium
                FROM (
                SELECT Origin, count(Tail_Number) as num
                FROM flight
                GROUP BY Origin
                having num >10129 and num <40515
                ) as medium
                 ''').show()
```

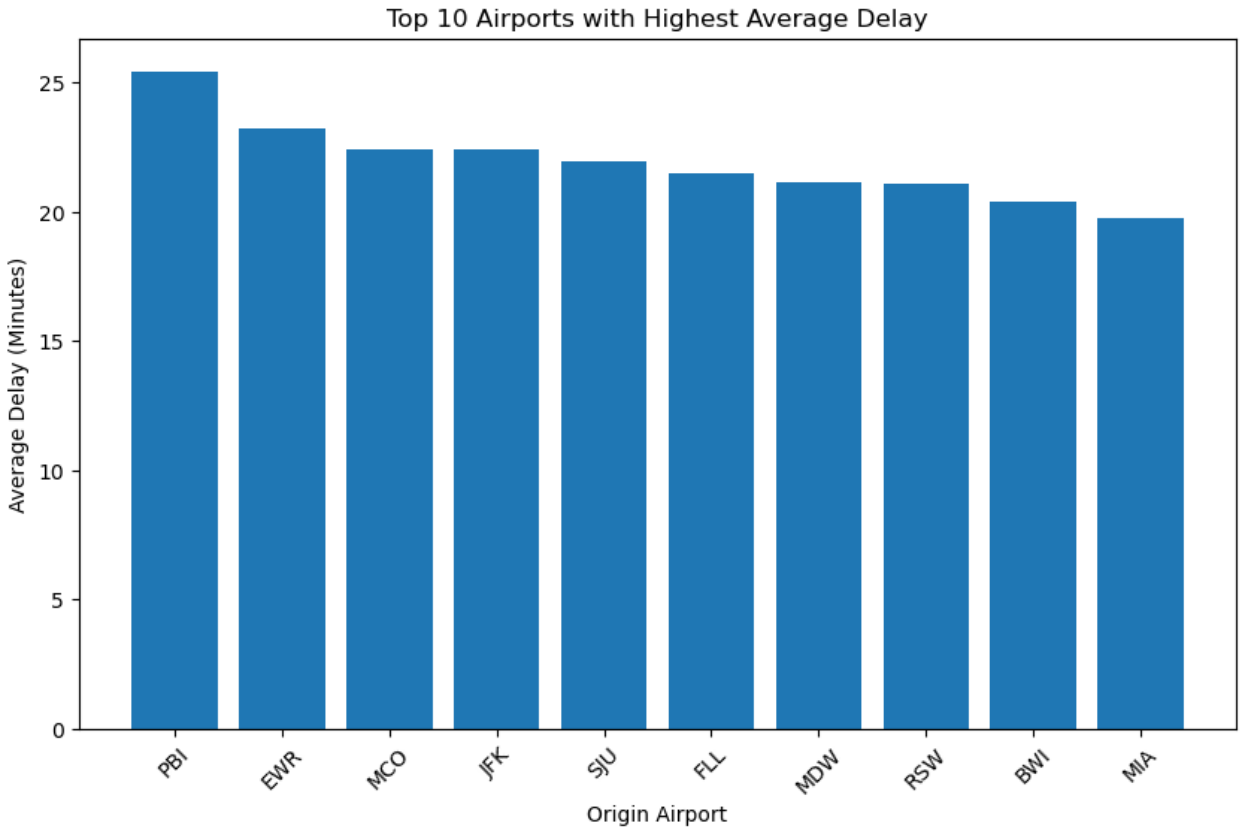
```
[Stage 20:=================================================>     (10 + 1)
/ 11]
+------+
|medium|
+------+
|    41|
+------+
```

In [ ]:
```
sqlContext.sql('''SELECT count(Origin) as large
                FROM (
                SELECT Origin, count(Tail_Number) as num
                FROM flight
                GROUP BY Origin
                having num > 40515
                ) as large
                 ''').show()
```

```
[Stage 26:=================================================>     (10 + 1)
/ 11]
+-----+
|large|
+-----+
|   30|
+-----+
```

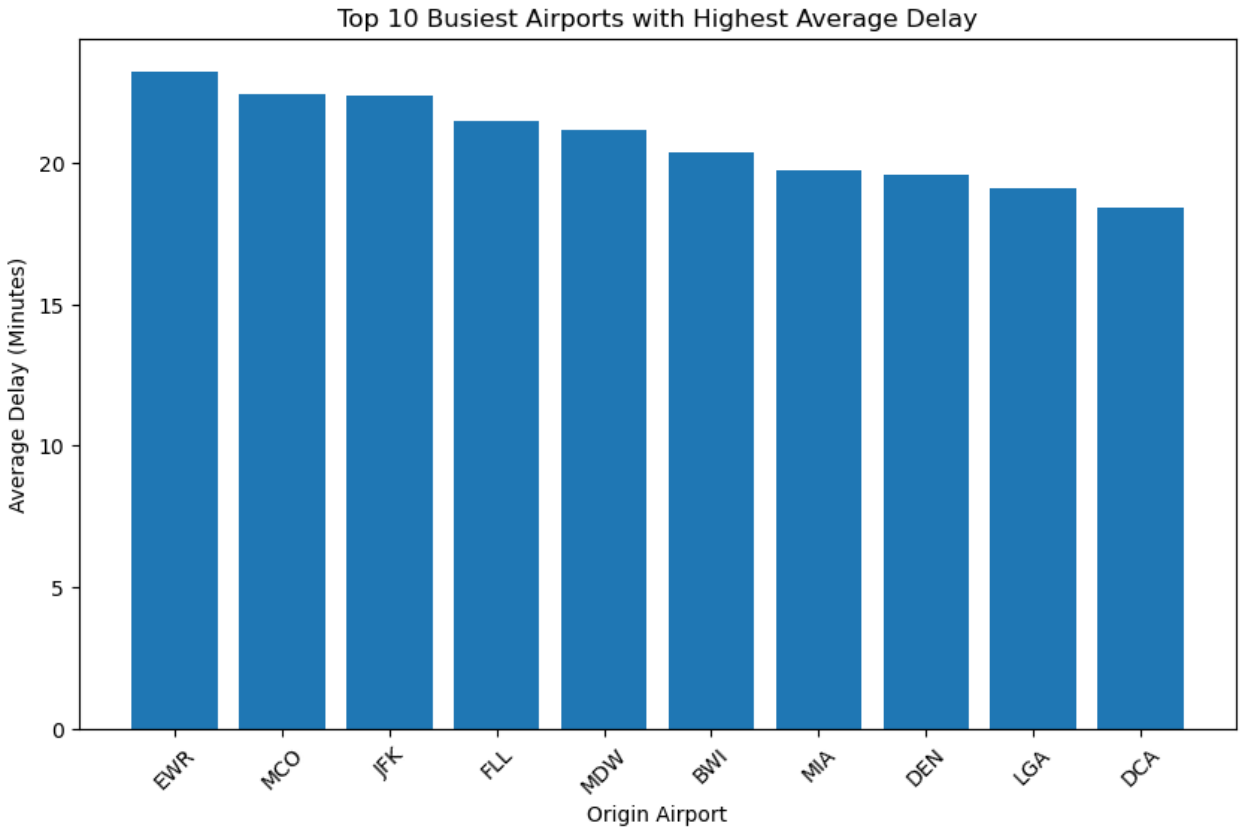
In [ ]:
```
result_3 = sqlContext.sql('''SELECT Origin, AVG_DELAY
                FROM (
                SELECT Origin, count(Tail_Number) as num, AVG(DepDelayMinu
                FROM flight
                GROUP BY Origin
                having num > 10129
                ORDER BY num DESC
                ) as table
                ORDER BY AVG_DELAY DESC
                LIMIT 10
                ''')

result_3_pd = result_3.toPandas()
```

In [ ]:
```
plt.figure(figsize=(10,6))
plt.bar(result_3_pd['Origin'], result_3_pd['AVG_DELAY'])
plt.xlabel('Origin Airport')
plt.ylabel('Average Delay (Minutes)')
plt.title('Top 10 Airports with Highest Average Delay')
plt.xticks(rotation=45)
plt.show()
```



In [ ]:
```
result_4 = sqlContext.sql('''SELECT Origin, AVG_DELAY
                FROM (
                SELECT Origin, count(Tail_Number) as num, AVG(DepDelayMinu
                FROM flight
                GROUP BY Origin
                having num > 40515
                ORDER BY num DESC
                ) as table
                ORDER BY AVG_DELAY DESC
                LIMIT 10
                ''')
result_4_pd = result_4.toPandas()
```

In [ ]:
```
plt.figure(figsize=(10,6))
plt.bar(result_4_pd['Origin'], result_4_pd['AVG_DELAY'])
plt.xlabel('Origin Airport')
plt.ylabel('Average Delay (Minutes)')
plt.title('Top 10 Busiest Airports with Highest Average Delay')
plt.xticks(rotation=45)
plt.show()
```



Delays are divided into three categories, namely "on time or small delay" (up to 15 minutes delay), "Medium delay" (15 – 45 minutes delay) and "Large delay" (45 minutes delay).

In [ ]:
```
from pyspark.sql.functions import when

flight_data_with_group =  flight_data.withColumn(
    "Delay_Group",
    when(flight_data["DepDelayMinutes"] == 0, "On_time")
    .when(flight_data["DepDelayMinutes"] <= 15, "Small_delay")
    .when((flight_data["DepDelayMinutes"] > 15) & (flight_data["DepDelayM
    .when(flight_data["DepDelayMinutes"] > 45, "Large_delay")
    .when(flight_data["Cancelled"] == True, "Cancelled")
)

flight_data_with_group.show(10)
```

```
+----------+------------------+-----------+-------------------------------
--+------+----+---------+--------+----------+-------+--------------+-----
----------+------------------+--------------+------------+-----------
-------+------------+-------+------+----------+-------+--------------+--
---------+
|FlightDate|           Airline|Tail_Number|Flight_Number_Marketing_Airli
ne|Origin|Dest|Cancelled|Diverted|CRSDepTime|DepTime|DepDelayMinutes|Origi
nAirportID|      OriginCityName|OriginStateName|DestAirportID|       DestC
ityName|DestStateName|TaxiOut|TaxiIn|CRSArrTime|ArrTime|ArrDelayMinutes|De
lay_Group|
+----------+------------------+-----------+-------------------------------
--+------+----+---------+--------+----------+-------+--------------+-----
----------+------------------+--------------+------------+-----------
-------+------------+-------+------+----------+-------+--------------+--
---------+
```

```
|2022-04-04|Commutair Aka Cha...|     N21144|                           43
01|   GJT| DEN|    false|   false|      1133| 1123.0|           0.0|
11921|  Grand Junction, CO|       Colorado|        11292|         Denver,
CO|     Colorado|   17.0|   8.0|      1245| 1228.0|           0.0|    On_
time|
|2022-04-04|Commutair Aka Cha...|     N16170|                           42
99|   HRL| IAH|    false|   false|       732|  728.0|           0.0|
12206|Harlingen/San Ben...|          Texas|        12266|        Houston,
TX|        Texas|   16.0|   9.0|       849|  848.0|           0.0|    On_
time|
|2022-04-04|Commutair Aka Cha...|     N21144|                           42
98|   DRO| DEN|    false|   false|      1529| 1514.0|           0.0|
11413|         Durango, CO|       Colorado|        11292|         Denver,
CO|     Colorado|   21.0|  14.0|      1639| 1636.0|           0.0|    On_
time|
|2022-04-04|Commutair Aka Cha...|     N11184|                           42
96|   IAH| GPT|    false|   false|      1435| 1430.0|           0.0|
12266|         Houston, TX|          Texas|        11973|Gulfport/Biloxi,
MS|  Mississippi|   16.0|   4.0|      1605| 1547.0|           0.0|    On_
time|
|2022-04-04|Commutair Aka Cha...|     N17146|                           42
95|   DRO| DEN|    false|   false|      1135| 1135.0|           0.0|
11413|         Durango, CO|       Colorado|        11292|         Denver,
CO|     Colorado|   19.0|   8.0|      1245| 1251.0|           6.0|    On_
time|
|2022-04-04|Commutair Aka Cha...|     N11191|                           42
94|   DEN| TUL|    false|   false|       955|  952.0|           0.0|
11292|          Denver, CO|       Colorado|        15370|          Tulsa,
OK|     Oklahoma|   25.0|   4.0|      1240| 1238.0|           0.0|    On_
time|
|2022-04-04|Commutair Aka Cha...|     N14143|                           42
93|   IAH| LCH|    false|   false|      2139| 2136.0|           0.0|
12266|         Houston, TX|          Texas|        12915|   Lake Charles,
LA|    Louisiana|   11.0|   5.0|      2231| 2218.0|           0.0|    On_
time|
|2022-04-04|Commutair Aka Cha...|     N13124|                           42
92|   TYS| IAH|    false|   false|      1129| 1117.0|           0.0|
15412|       Knoxville, TN|      Tennessee|        12266|        Houston,
TX|        Texas|   22.0|  16.0|      1306| 1311.0|           5.0|    On_
time|
|2022-04-04|Commutair Aka Cha...|     N33182|                           42
91|   IAH| AEX|    false|   false|      1424| 1414.0|           0.0|
12266|         Houston, TX|          Texas|        10185|     Alexandria,
LA|    Louisiana|   16.0|   6.0|      1524| 1513.0|           0.0|    On_
time|
|2022-04-04|Commutair Aka Cha...|     N21154|                           42
90|   IAH| MOB|    false|   false|       954|  947.0|           0.0|
12266|         Houston, TX|          Texas|        13422|         Mobile,
AL|      Alabama|   17.0|   6.0|      1121| 1110.0|           0.0|    On_
time|
+----------+--------------------+-----------+--------------------------
--+------+----+---------+--------+----------+-------+--------------+-----
----------+--------------------+---------------+-------------+-----------
-------+-------------+-------+------+----------+-------+--------------+--
---------+
only showing top 10 rows
```

In [ ]:
```
flight_data_with_group.registerTempTable('flight_group')

sqlContext.sql('''SELECT COUNT(Tail_Number) AS Flight_Num, Delay_Group
                FROM flight_group
                GROUP BY Delay_Group
                ORDER BY Flight_Num DESC
                 ''').show()

#The numbuer of flight in each delay group
```

```
/Users/evan/anaconda3/lib/python3.11/site-packages/pyspark/sql/dataframe.p
y:330: FutureWarning: Deprecated in 2.0, use createOrReplaceTempView inste
ad.
  warnings.warn("Deprecated in 2.0, use createOrReplaceTempView instead.",
FutureWarning)
[Stage 39:=================================================>     (10 + 1)
/ 11]
+----------+------------+
|Flight_Num| Delay_Group|
+----------+------------+
|   2398937|     On_time|
|    722483| Small_delay|
|    446011|Medium_delay|
|    390392| Large_delay|
|     93700|   Cancelled|
+----------+------------+
```

In [ ]:
```
sqlContext.sql('''SELECT
                Delay_Group,
                COUNT(Tail_Number) AS Flight_Num,
                (COUNT(Tail_Number) / SUM(COUNT(Tail_Number)) OVER ()) * 1
                FROM flight_group
                GROUP BY Delay_Group
                ORDER BY Flight_Num DESC
                 ''').show()

#Percentage of each group, can draw a pie chart based on that.
```

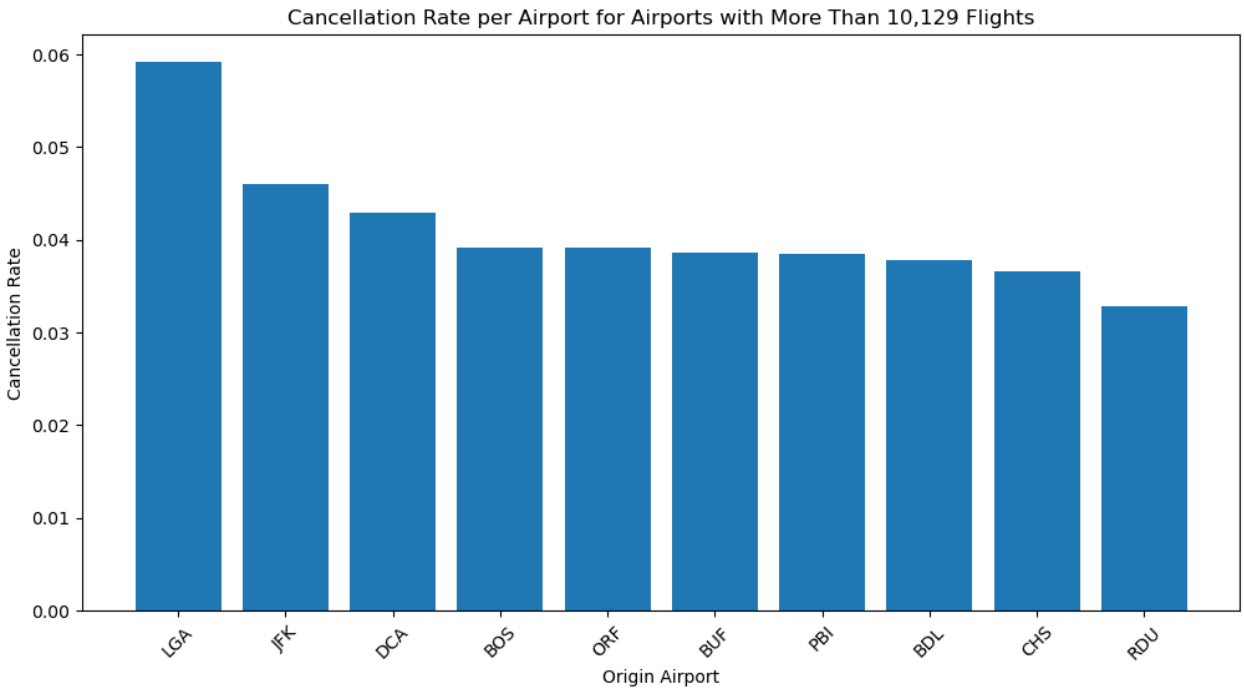
```
23/12/07 21:34:25 WARN WindowExec: No Partition Defined for Window operati
on! Moving all data to a single partition, this can cause serious performa
nce degradation.
23/12/07 21:34:25 WARN WindowExec: No Partition Defined for Window operati
on! Moving all data to a single partition, this can cause serious performa
nce degradation.
23/12/07 21:34:25 WARN WindowExec: No Partition Defined for Window operati
on! Moving all data to a single partition, this can cause serious performa
nce degradation.
[Stage 42:=================================================>     (10 + 1)
/ 11]
```

```
+------------+----------+------------------+
| Delay_Group|Flight_Num|        Percentage|
+------------+----------+------------------+
|     On_time|   2398937| 59.21074618112744|
| Small_delay|    722483| 17.83238056404962|
|Medium_delay|    446011|11.008477552762258|
| Large_delay|    390392|  9.63568514852316|
|   Cancelled|     93700|2.3127105535375216|
+------------+----------+------------------+
```

```
23/12/07 21:34:27 WARN WindowExec: No Partition Defined for Window operati
on! Moving all data to a single partition, this can cause serious performa
nce degradation.
23/12/07 21:34:27 WARN WindowExec: No Partition Defined for Window operati
on! Moving all data to a single partition, this can cause serious performa
nce degradation.
23/12/07 21:34:27 WARN WindowExec: No Partition Defined for Window operati
on! Moving all data to a single partition, this can cause serious performa
nce degradation.
23/12/07 21:34:27 WARN WindowExec: No Partition Defined for Window operati
on! Moving all data to a single partition, this can cause serious performa
nce degradation.
```

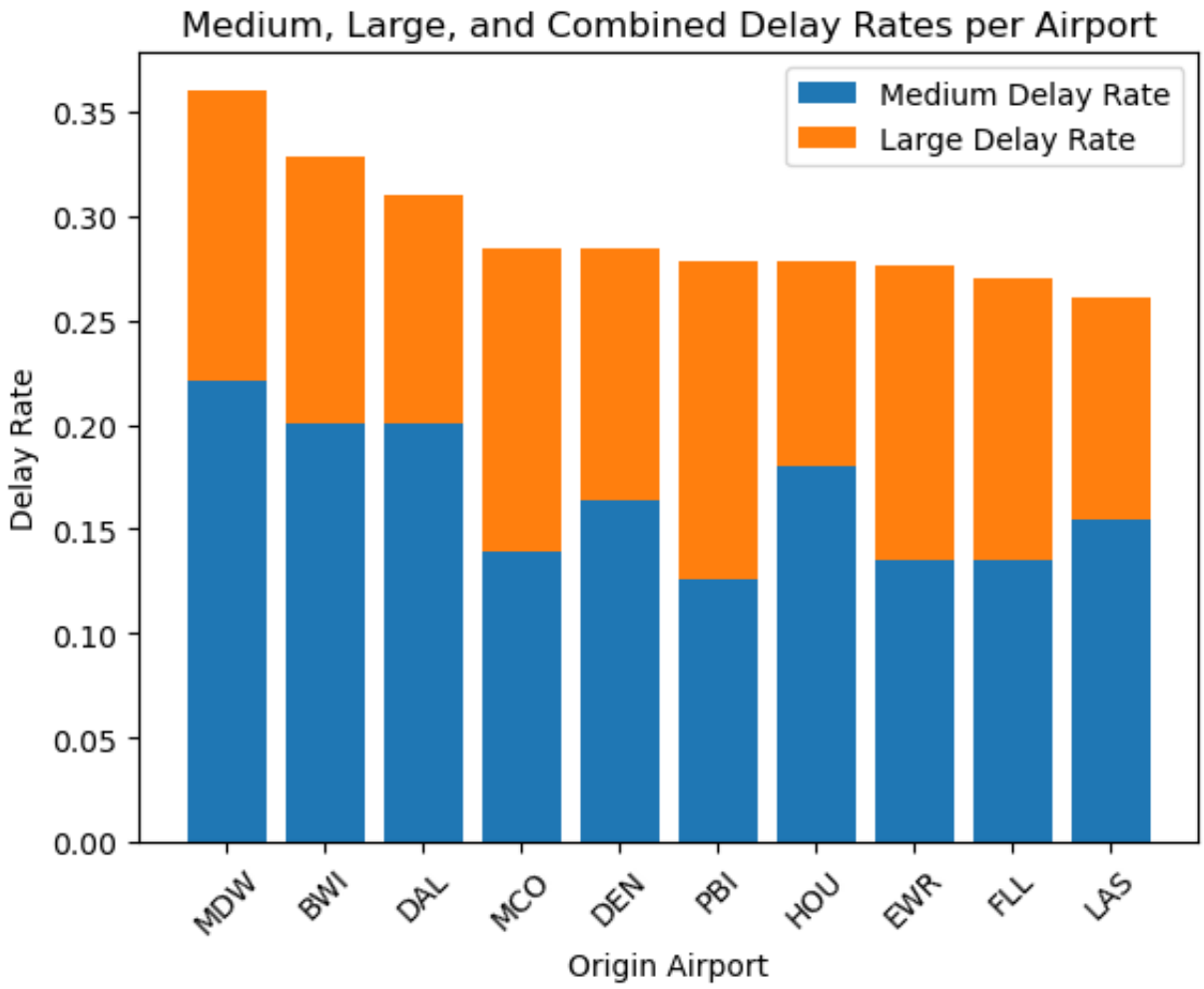
In [ ]:
```
result_5 = sqlContext.sql('''SELECT f.Origin, COUNT(f.Tail_Number) AS Can
               FROM flight_group f
               JOIN (SELECT Origin, COUNT(Tail_Number) AS Total_Count FRO
               GROUP BY Origin
               HAVING COUNT(Tail_Number) > 10129
               ) t ON f.Origin = t.Origin
               WHERE f.Cancelled = 'True'  -- or TRUE if it's a boolean f
               GROUP BY
               f.Origin, t.Total_Count
               ORDER BY Cancelled_RATE DESC
                         LIMIT 10
               ''')

result_5_pd = result_5.toPandas()
```

In [ ]:
```
plt.figure(figsize=(12, 6))
plt.bar(result_5_pd['Origin'], result_5_pd['Cancelled_Rate'])
plt.xlabel('Origin Airport')
plt.ylabel('Cancellation Rate')
plt.title('Cancellation Rate per Airport for Airports with More Than 10,1
plt.xticks(rotation=45)
plt.show()
```



In [ ]:
```
result_6 = sqlContext.sql('''SELECT f.Origin, SUM(CASE WHEN f.Delay_Group
                SUM(CASE WHEN f.Delay_Group = 'Large_delay' THEN 1 ELSE 0
                (SUM(CASE WHEN f.Delay_Group = 'Medium_delay' THEN 1 ELSE
                FROM flight_group f JOIN
                (SELECT Origin, COUNT(*) AS Total_Flights
                FROM flight_group
                GROUP BY Origin
                HAVING COUNT(*) > 10129
                ) t ON f.Origin = t.Origin
                GROUP BY
                f.Origin, t.Total_Flights
                ORDER BY Combined_Delay_Rate DESC
                          limit 10
                ''')
result_6_pd = result_6.toPandas()
```

In [ ]:
```
plt.bar(result_6_pd['Origin'], result_6_pd['Medium_Delay_Rate'], label='M
plt.bar(result_6_pd['Origin'], result_6_pd['Large_Delay_Rate'], bottom=re
plt.xlabel('Origin Airport')
plt.ylabel('Delay Rate')
plt.title('Medium, Large, and Combined Delay Rates per Airport')
plt.xticks(rotation=45)
plt.legend()
plt.show()
```


Medium, Large, and Combined Delay Rates per Airport
Medium Delay Rate
Large Delay Rate
Delay Rate
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
MDW
BWI
DAL
MCO
DEN
PBI
HOU
EWR
FLL
LAS
Origin Airport

Flight Data Exploration

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

import pyspark
from pyspark.sql import SparkSession
from pyspark.sql import functions as F
spark = (SparkSession.builder.master("local[*]").appName("Final_Project").getOrCrea
sc = spark.sparkContext
import pandas as pd
```

Random Forest

In [ ]:
```
df = spark.read.option('header','true').csv("2021.csv",inferSchema = True)
```

Data preparation

In [ ]:
```
df= df.select('DepDel15','Month','Distance','Operating_Airline','Operated_or_Brande
from pyspark.sql.functions import when, col
df = df.withColumn('Operated_or_Branded_Code_Share_Partners',
                            when(col('Operated_or_Branded_Code_Share_Partners').con
from pyspark.ml.feature import StringIndexer
indexer1=StringIndexer(inputCol='Operating_Airline',outputCol='Operating_Airline_ca
indexer2=StringIndexer(inputCol='OriginAirportID',outputCol='OriginAirportID_cat')
indexer3=StringIndexer(inputCol='DestAirportID',outputCol='DestAirportID_cat')
indexed=indexer1.fit(df).transform(df)
indexed=indexer2.fit(indexed).transform(indexed)
indexed=indexer3.fit(indexed).transform(indexed)


from pyspark.ml.feature import VectorAssembler
assembler = VectorAssembler(inputCols=["Month","Distance","Operated_or_Branded_Code

output = assembler.transform(indexed)
output = output.filter(~col('DepDel15').contains('NULL'))
```

Model fitting

In [ ]:
```
from pyspark.ml.classification import RandomForestClassifier
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

rf = RandomForestClassifier(featuresCol="features", labelCol="DepDel15", numTrees=1

train_data, test_data = output.randomSplit([0.7, 0.3])

rf_model = rf.fit(train_data)

predictions = rf_model.transform(test_data)

evaluator = MulticlassClassificationEvaluator(labelCol="DepDel15", predictionCol="p

accuracy = evaluator.evaluate(predictions)

print(f"Model accuracy: {accuracy:.2f}")
```

```
Model accuracy: 0.83
```

In [ ]:
```
from pyspark.ml.evaluation import RegressionEvaluator
evaluator = RegressionEvaluator(labelCol="DepDel15", predictionCol="prediction", me
mse = evaluator.evaluate(predictions)
mse
```

Out[ ]:
```
0.17355733310957025
```

In [ ]:
```
from pyspark.sql.functions import col
from pyspark.sql import DataFrame
from pyspark.sql.functions import col, explode, array, lit

major_df = output.filter(col('DepDel15') == 0)
minor_df = output.filter(col('DepDel15') == 1)
ratio = major_df.count() / minor_df.count()

oversampled_df = minor_df.withColumn("dummy", explode(array([lit(x) for x in range(
oversampled_df = oversampled_df.drop('dummy')

balanced_df = major_df.unionAll(oversampled_df)
train_data, test_data = balanced_df.randomSplit([0.7, 0.3])
count_class_0 = balanced_df.filter(col('DepDel15') == 0).count()
count_class_1 = balanced_df.filter(col('DepDel15') == 1).count()

print(f"Number of rows where 'DepDel15' is 0: {count_class_0}")
print(f"Number of rows where 'DepDel15' is 1: {count_class_1}")
```

```
Number of rows where 'DepDel15' is 0: 5129191
Number of rows where 'DepDel15' is 1: 4297068
```

In [ ]:
```
from pyspark.ml.evaluation import RegressionEvaluator


evaluator = RegressionEvaluator(labelCol="DepDel15", predictionCol="prediction", me

mse = evaluator.evaluate(predictions)
print("Mean Squared Error (MSE) on test data = %g" % mse)
```

```
Mean Squared Error (MSE) on test data = 0.173557
```