# *Airline Data Analysis*

Narasimha Royal Pola
Master’s in computer science
University of Houston
Houston, TX, USA
npola@cougarnet.uh.edu

***Abstract*—This project uses a sizable dataset made public by the U.S. Department of Transportation to study national flight traffic. From 1990 until the ending of 2024, the collection contains data on the flow of people and goods between American airports and foreign locations. The objective is to analyze airline performance, look at popular travel routes, and spot trends in air traffic volume over time. The project makes use of Tableau Public for data visualization and PySpark for processing massive amounts of data. To facilitate exploratory research, crucial procedures include data transformation, cleansing, and simple grouping. The study provides graphic insights into carrier-level information, route-based mobility, and annual passenger trends. This study shows how well basic big data methods may be used to glean insightful information from organized transportation records.**

***Keywords—Tableau, LSTM, Big Data Analytics, Apache Spark***

## I. Introduction

Air travel plays a critical role in global transportation, connecting people and economies across regions. Understanding trends in airline traffic can help stakeholders make informed decisions in areas such as operations, infrastructure, and policy. With the availability of large-scale open datasets, such as those provided by the U.S. Department of Transportation, it is now possible to perform detailed analysis on national air travel using modern data processing tools.

This project focuses on examining passenger and freight movements between U.S. airports and international destinations over the last three decades. The objective is to uncover patterns related to airline activity, seasonal shifts, and route popularity using PySpark for scalable data processing and Tableau Public for visual exploration. Unlike predictive modeling, the emphasis here is on summarizing trends and extracting insights through descriptive analytics.

## II. Background and Related Work

Airline data has long been used to study patterns in global travel, economic activity, and logistical efficiency. Researchers and analysts often explore flight records to evaluate demand trends, airport congestion, or the impact of external factors like holidays and fuel prices. While many academic papers take a forecasting approach, using statistical or machine learning models to predict future traffic, there is still significant value in analyzing historical trends without prediction.

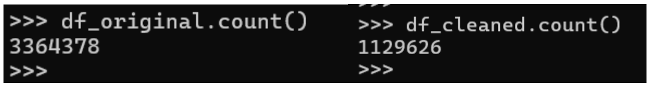
Several transportation studies have focused on delay prediction, pricing strategies, or network optimization, often requiring external data sources or labeled outcomes. In contrast, this project focuses purely on internal dataset patterns, looking at airline performance, seasonal behavior, and route popularity based on aggregated totals. While similar descriptive projects exist in industry, they are not always shared publicly or reproduced using open data tools.

This project builds on that idea by using openly available data and accessible tools like PySpark and Tableau Public. The goal was to perform a complete workflow, from raw dataset to insights, while keeping the process transparent and reproducible. Rather than replicating academic experiments, this work aims to show what a single person can uncover by carefully cleaning, structuring, and visualizing a large dataset.

## III. Dataset Description

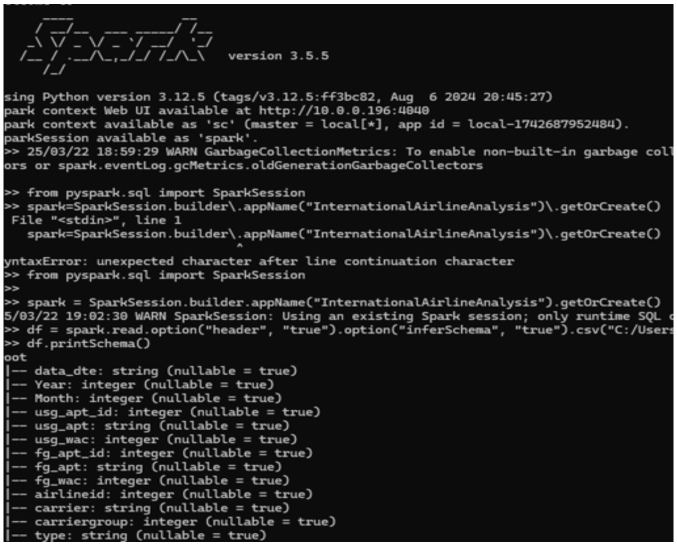
The system design for this project is kept minimal, focusing on efficient data handling and visualization. The workflow consists of two major stages: data processing and data visualization. PySpark is used as the primary tool for loading, cleaning, and aggregating the large-scale dataset, taking advantage of its distributed computing capabilities to handle over one million records. The cleaned and transformed dataset is exported to a CSV file, which serves as the input for visualization in Tableau Public.

The architecture follows a linear flow: data ingestion, preprocessing, export, and visualization. No complex machine learning pipeline or deployment infrastructure is involved, as the project's goal is centered around trend analysis and insights extraction rather than model training or prediction.



```
>>> df_original.count()      >>> df_cleaned.count()
3364378                      1129626
>>>                          >>>
```

Fig: Number of  rows before and after cleaning

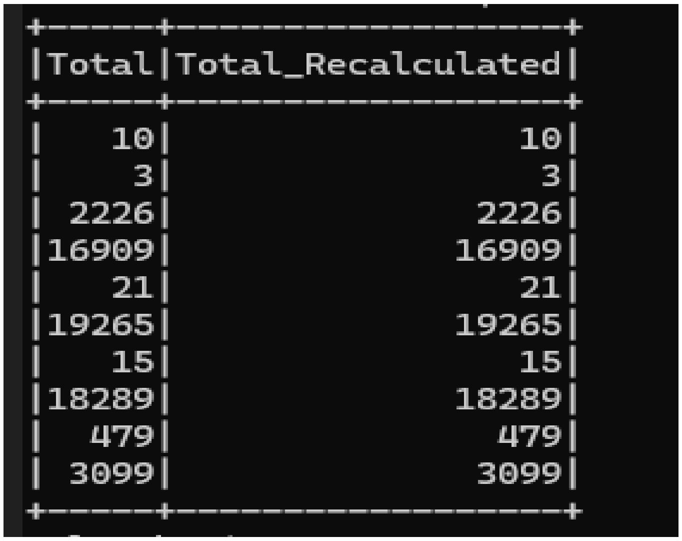


```
     ____              __
    / __/__  ___ _____/ /__
   _\ \/ _ \/ _ `/ __/  '_/
  /__ / .__/\_,_/_/ /_/\_\   version 3.5.5
     /_/

sing Python version 3.12.5 (tags/v3.12.5:ff3bc82, Aug  6 2024 20:45:27)
park context Web UI available at http://10.0.0.196:4040
park context available as 'sc' (master = local[*], app id = local-1742687952484).
parkSession available as 'spark'.
>> 25/03/22 18:59:29 WARN GarbageCollectionMetrics: To enable non-built-in garbage coll
ors or spark.eventLog.gcMetrics.oldGenerationGarbageCollectors

>> from pyspark.sql import SparkSession
>> spark=SparkSession.builder\.appName("InternationalAirlineAnalysis")\.getOrCreate()
 File "<stdin>", line 1
   spark=SparkSession.builder\.appName("InternationalAirlineAnalysis")\.getOrCreate()
                              ^
yntaxError: unexpected character after line continuation character
>> from pyspark.sql import SparkSession
>>
>> spark = SparkSession.builder.appName("InternationalAirlineAnalysis").getOrCreate()
5/03/22 19:02:30 WARN SparkSession: Using an existing Spark session; only runtime SQL c
>> df = spark.read.option("header", "true").option("inferSchema", "true").csv("C:/Users
>> df.printSchema()
oot
|-- data_dte: string (nullable = true)
|-- Year: integer (nullable = true)
|-- Month: integer (nullable = true)
|-- usg_apt_id: integer (nullable = true)
|-- usg_apt: string (nullable = true)
|-- usg_wac: integer (nullable = true)
|-- fg_apt_id: integer (nullable = true)
|-- fg_apt: string (nullable = true)
|-- fg_wac: integer (nullable = true)
|-- airlineid: integer (nullable = true)
|-- carrier: string (nullable = true)
|-- carriergroup: integer (nullable = true)
|-- type: string (nullable = true)
```

Fig: Apache Spark setup



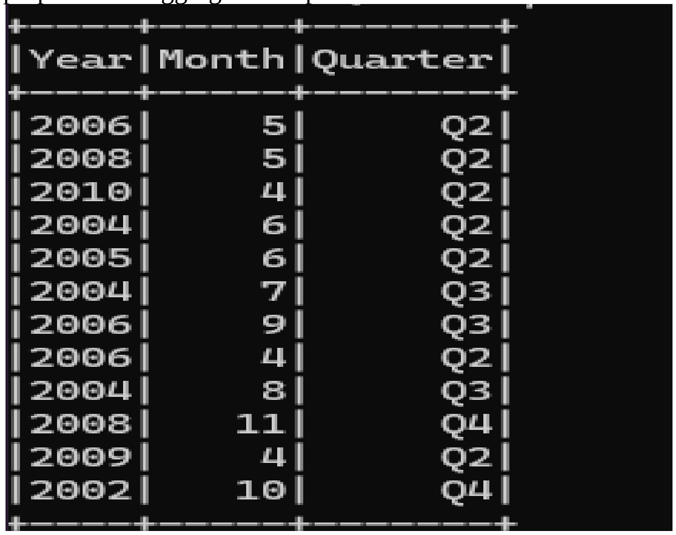
```
+-----+------------------+
|Total|Total_Recalculated|
+-----+------------------+
|   10|                10|
|    3|                 3|
| 2226|              2226|
|16909|             16909|
|   21|                21|
|19265|             19265|
|   15|                15|
|18289|             18289|
|  479|               479|
| 3099|              3099|
+-----+------------------+
```

Fig: Total_Recalculated

## IV. Data Preprocessing

The raw dataset, though comprehensive, required several cleaning and restructuring steps to ensure reliability for downstream analysis. Initially, we inspected the dataset for null or missing values, confirming their absence across all columns. To streamline the analysis, non-essential columns such as location identifiers and redundant metadata were dropped, keeping only relevant fields like year, month, airports, carrier, type, and total counts. A new column was derived to validate the 'Total' as the sum of 'Scheduled' and 'Charter' counts, ensuring internal consistency.

Furthermore, a 'Quarter' field was generated from the month values to facilitate seasonal analysis. To improve interpretability, airline carrier codes were mapped to their full names. These transformations were implemented using PySpark to efficiently handle the large dataset size and prepare it for aggregate computation and visualization.

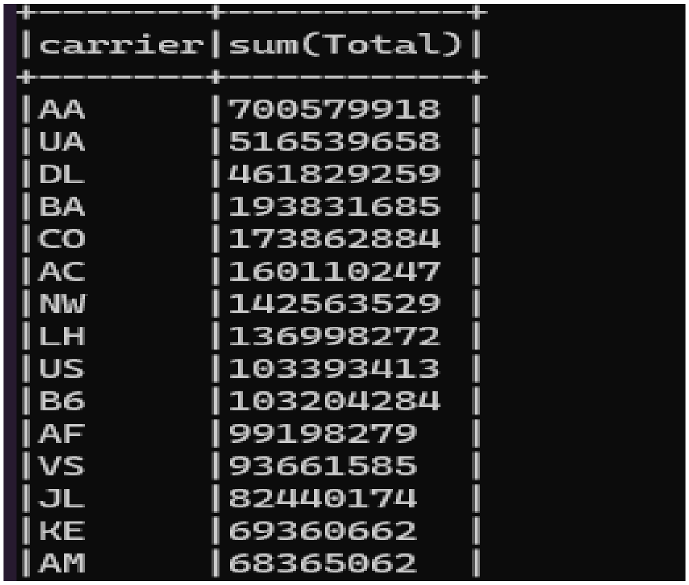


```
+----+-----+-------+
|Year|Month|Quarter|
+----+-----+-------+
|2006|    5|     Q2|
|2008|    5|     Q2|
|2010|    4|     Q2|
|2004|    6|     Q2|
|2005|    6|     Q2|
|2004|    7|     Q3|
|2006|    9|     Q3|
|2006|    4|     Q2|
|2004|    8|     Q3|
|2008|   11|     Q4|
|2009|    4|     Q2|
|2002|   10|     Q4|
+----+-----+-------+
```

Fig: Quarter Field

## V. Data Aggregation and Analysis

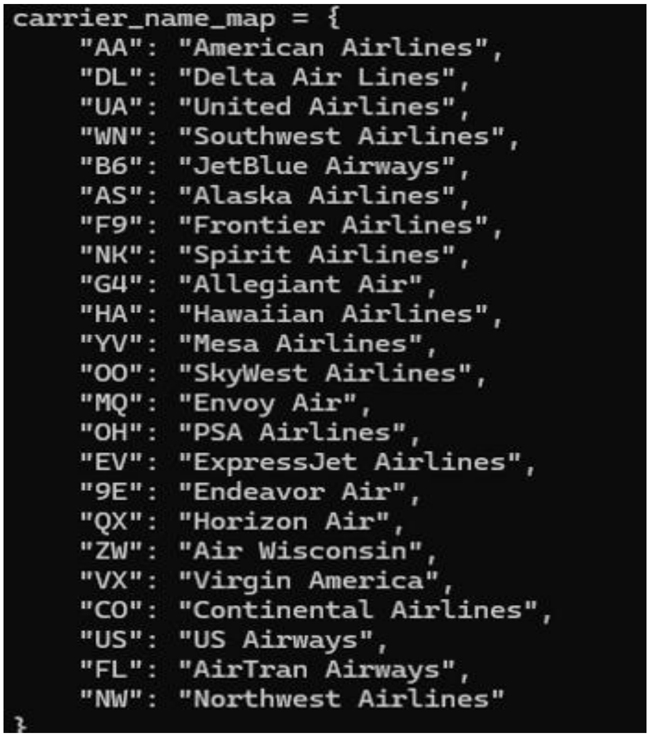
Once the dataset was cleaned and structured, the next step was to group and summarize the data to identify patterns. Using PySpark, I grouped records by year, airline, route, and quarter to track how passenger and freight volumes changed over time. Aggregations like total passengers per year, per carrier, and per route were done to simplify the large dataset and make the trends easier to analyze. This step helped turn raw rows into meaningful summaries, like how many passengers United Airlines carried from 1995 to 2010 or what the top five busiest international routes were over the last decade.



```
+-------+----------+
|carrier|sum(Total)|
+-------+----------+
|AA     |700579918 |
|UA     |516539658 |
|DL     |461829259 |
|BA     |193831685 |
|CO     |173862884 |
|AC     |160110247 |
|NW     |142563529 |
|LH     |136998272 |
|US     |103393413 |
|B6     |103204284 |
|AF     |99198279  |
|VS     |93661585  |
|JL     |82440174  |
|KE     |69360662  |
|AM     |68365062  |
```

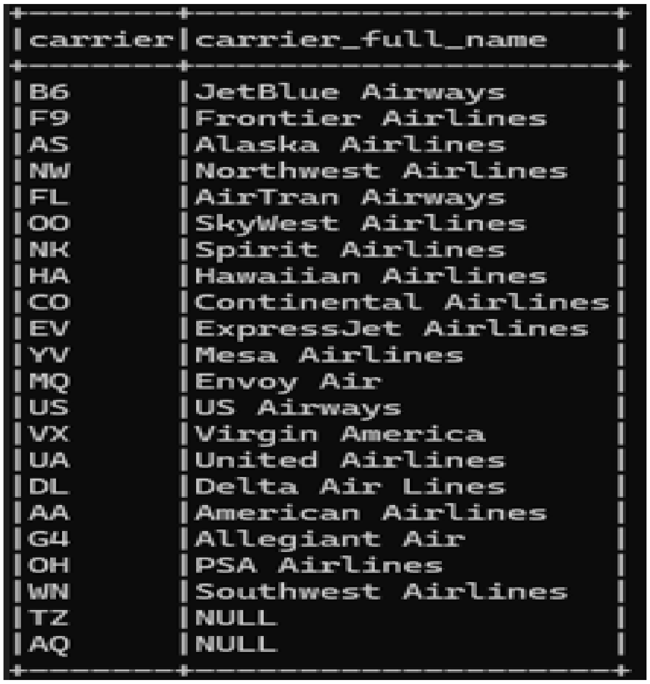
Fig: SUM Field

During analysis, I found that grouping by both time and carrier gave more specific insights compared to just looking at yearly totals. For example, instead of saying 2005 had high traffic, I could say which airlines or airports were responsible for the spike. I also calculated totals by quarter to see how seasons affected travel. These grouped outputs were later used in Tableau for visualization, but even within PySpark, they helped me filter down the noise and focus on patterns that made sense for presentation. No machine learning was used — just basic grouping, filtering, and sorting to get a clear picture of the trends.



```
carrier_name_map = {
    "AA": "American Airlines",
    "DL": "Delta Air Lines",
    "UA": "United Airlines",
    "WN": "Southwest Airlines",
    "B6": "JetBlue Airways",
    "AS": "Alaska Airlines",
    "F9": "Frontier Airlines",
    "NK": "Spirit Airlines",
    "G4": "Allegiant Air",
    "HA": "Hawaiian Airlines",
    "YV": "Mesa Airlines",
    "OO": "SkyWest Airlines",
    "MQ": "Envoy Air",
    "OH": "PSA Airlines",
    "EV": "ExpressJet Airlines",
    "9E": "Endeavor Air",
    "QX": "Horizon Air",
    "ZW": "Air Wisconsin",
    "VX": "Virgin America",
    "CO": "Continental Airlines",
    "US": "US Airways",
    "FL": "AirTran Airways",
    "NW": "Northwest Airlines"
}
```

Fig: Carrier names

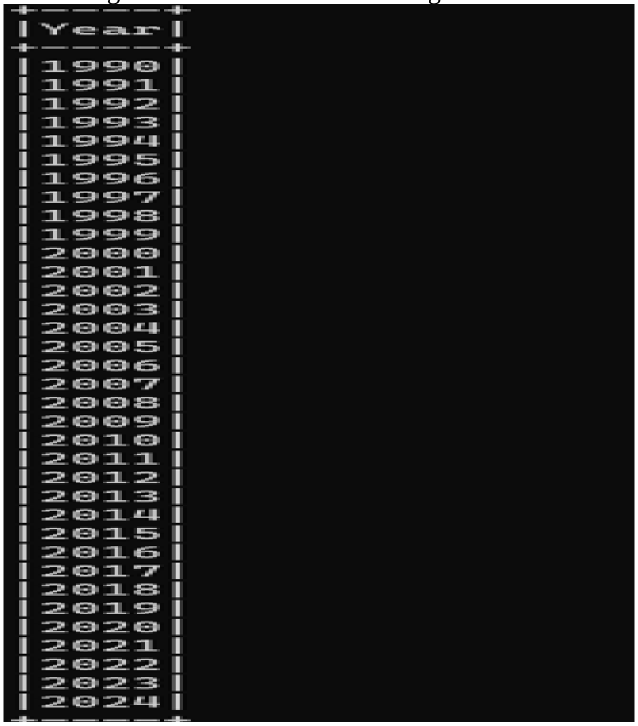


```
+-------+--------------------+
|carrier|carrier_full_name   |
+-------+--------------------+
|B6     |JetBlue Airways     |
|F9     |Frontier Airlines   |
|AS     |Alaska Airlines     |
|NW     |Northwest Airlines  |
|FL     |AirTran Airways     |
|OO     |SkyWest Airlines    |
|NK     |Spirit Airlines     |
|HA     |Hawaiian Airlines   |
|CO     |Continental Airlines|
|EV     |ExpressJet Airlines |
|YV     |Mesa Airlines       |
|MQ     |Envoy Air           |
|US     |US Airways          |
|VX     |Virgin America      |
|UA     |United Airlines     |
|DL     |Delta Air Lines     |
|AA     |American Airlines   |
|G4     |Allegiant Air       |
|OH     |PSA Airlines        |
|WN     |Southwest Airlines  |
|TZ     |NULL                |
|AQ     |NULL                |
+-------+--------------------+
```

Fig: Carrier_full_names

## VI. Visualization using Tableau

After aggregating the cleaned dataset using PySpark, I exported the results into CSV format to prepare for visual exploration. Tableau Public was used as the main tool to create interactive visualizations from these summary tables. Since Tableau doesn’t directly connect to PySpark, the exported CSV served as a bridge between the processing and visualization phases. I chose Tableau because it allows fast drag-and-drop chart creation, which was helpful for testing different angles of the data without writing additional code.



```
+----+
|Year|
+----+
|1990|
|1991|
|1992|
|1993|
|1994|
|1995|
|1996|
|1997|
|1998|
|1999|
|2000|
|2001|
|2002|
|2003|
|2004|
|2005|
|2006|
|2007|
|2008|
|2009|
|2010|
|2011|
|2012|
|2013|
|2014|
|2015|
|2016|
|2017|
|2018|
|2019|
|2020|
|2021|
|2022|
|2023|
|2024|
+----+
```

Fig: Year span

The visualizations included time-series plots for passenger trends over the years, bar charts comparing airline performance, and ranked lists of the busiest international flight routes. One of the most useful visuals was the yearly trend chart that clearly showed the steep drop in 2020 due to the pandemic, followed by a gradual recovery. I also created route-specific visuals by combining origin and destination airport codes into a single “route” field. This made it easier to identify which international connections had the highest traffic. Using Tableau helped make these insights more readable and presentation-friendly, especially for non-technical viewers.

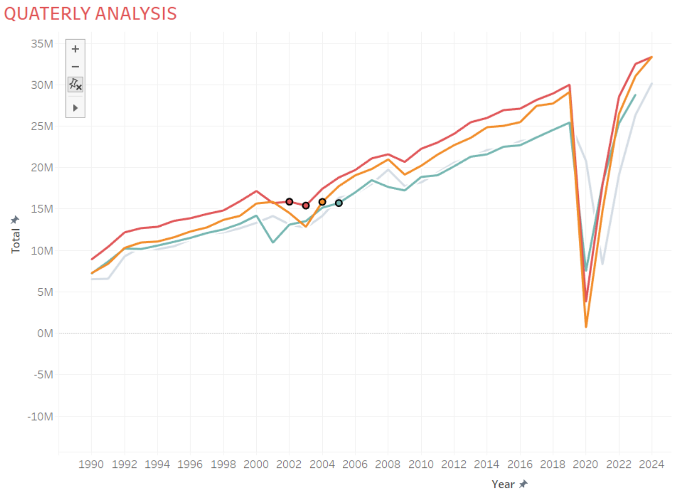


Fig: Quarterly Analysis

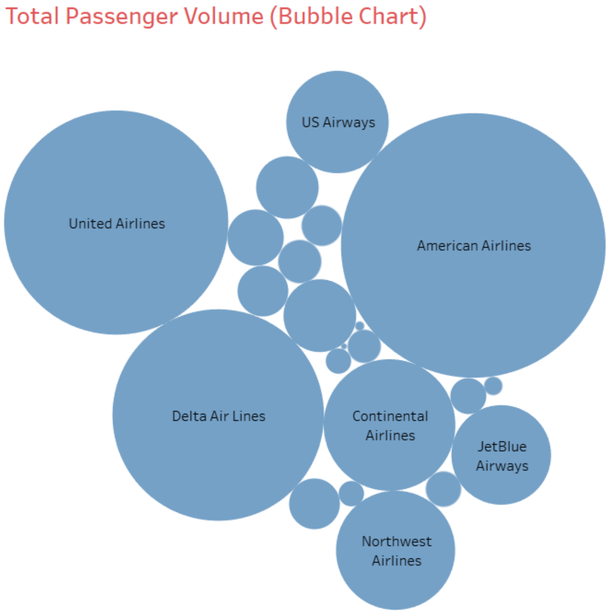


Fig: Total Passenger Volume per airline(Bubble Chart)

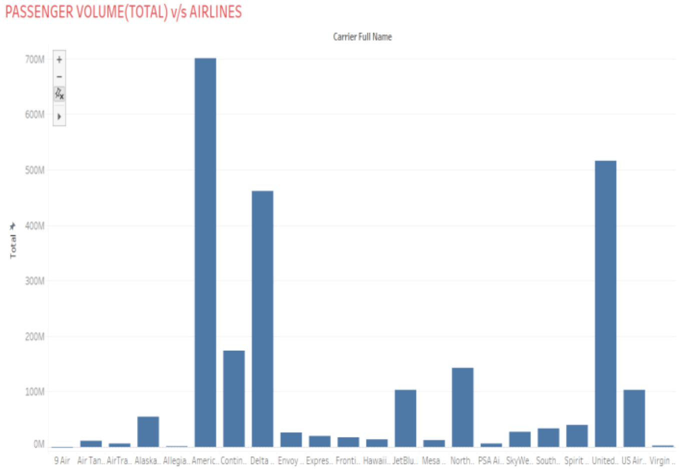


Fig: Passenger Volume/Airlines (Bar chart)

## VII. Challenges and Solutions

Working with a large dataset for the first time came with a few learning curves. One of the main challenges was dealing with the size of the file. Traditional tools like Excel or pandas couldn't handle the full dataset smoothly, which is why I moved to PySpark early in the project. However, setting up Spark locally and getting it to work with my system variables took time. I faced multiple issues with Java paths and environment variables and had to troubleshoot several times before it worked properly.

Another issue was understanding the data itself. Some of the columns, like airport or carrier codes, weren't explained clearly in the dataset, so I had to manually look them up and create a reference for airline names. There was also confusion around totals, some rows had mismatches between the scheduled, charter, and total values, so I had to re-calculate totals and filter out incorrect rows. Exporting data from PySpark to CSV for use in Tableau also caused some permission errors at first, especially with Windows file paths, but those were resolved after adjusting file permissions and using absolute paths. Each of these problems slowed me down but solving them gave me a clearer understanding of how data tools work under the hood.

## VIII. Results and Evaluation

The analysis revealed some clear trends in how international air travel has changed over time. From 1990 to around 2019, passenger numbers steadily increased each year, showing the growth of global connectivity. However, in 2020 there was a sharp drop in passenger traffic across all airlines, which clearly reflects the impact of the COVID-19 pandemic. In the following years, passenger numbers began to recover, but the data showed that different airlines bounced back at different rates. For example, United Airlines and Delta showed strong recovery by 2023, while some smaller carriers had slower growth.

One of the most interesting insights came from route analysis. By combining origin and destination airport codes into a single “route” field, I was able to rank the busiest international corridors. Routes like JFK to NRT (Tokyo Narita), ORD to LHR (London Heathrow), and IAH to CUN (Cancún) consistently appeared among the top. When analyzing seasonal patterns, I found that the third quarter, especially July to September had the highest passenger traffic across most years, likely due to summer travel. These insights match real-world trends and give a clearer picture of how different factors like location, season, and global events affect air travel.

## IX. Conclusion

This project started with the goal of understanding international airline travel patterns using a large, real-world dataset. By using PySpark for data handling and Tableau for visualization, I was able to explore over three decades of data and extract meaningful insights. The focus wasn’t on building complex models, but instead on preparing and analyzing the data in a structured way to highlight trends that would otherwise be buried in millions of rows.

The approach was entirely based on descriptive analysis. Starting from data cleaning, fields were restructured, airline codes were mapped, and totals were validated to make the dataset more readable and consistent. After processing, aggregations were done based on year, quarter, carrier, and route, which helped uncover seasonality, major drops in travel during the pandemic, and recovery patterns in the years that followed.

Overall, this project showed how big data tools can be used in a simple and direct way to explore large datasets without needing machine learning or complex models. Tools like PySpark and Tableau made it easier to handle scale and to present the insights visually. The process also helped reinforce how important data preparation is when trying to find real-world trends.

## References

[1] Dataset: https://www.bts.gov/

[2] https://www.academia.edu/44787344/Airline_Data_Analysis

[3] https://www.researchgate.net/publication/357661734_Data_Analytics_for_Air_Travel_Data_A_Survey_and_New_Perspectives

[4] https://public.tableau.com/app/discover

[5] https://spark.apache.org/