Airline Data Challenge Documentation

Table of Contents

1 Introduction

This document outlines the approach and methodology used to analyze airline flight data for 1Q2019. The goal is to help a new airline identify the top 5 round-trip domestic U.S. routes to invest in, ensuring profitability and alignment with the 'On-time, for you' punctuality standards.

2 Problem Statement

The company plans to: Launch 5 round-trip routes between medium and large U.S. airports.

- Purchase 5 airplanes ($90M each).
- Analyze flight data for 1Q2019 to:
- Identify busiest routes.
- Calculate the most profitable routes.
- Recommend 5 routes to invest in.
- Determine breakeven points for each.
- Define KPIs for ongoing success tracking.

3 Technologies & Tools Used

- **Python Libraries**:
    - pandas and NumPy for data wrangling and manipulation.
    - matplotlib for visualizing outliers and distribution patterns.
- **Tableau**:
    - Used for visual exploration and to create a final reusable visualization dashboard for key findings.
- **Code Structure**:
    - A separate custom module, Functions.py, was developed to house reusable utility functions (data wrangling and cost calculations).
    - The main workflow and analytical logic were implemented in airline_analysis.py within a Jupyter Notebook environment.

4 Data Preparation:

We were provided with three primary data files along with one metadata file to support the Capital One Airline Data Challenge. The datasets include:

- Flights.csv – Contains flight-level operational data such as routes, timings, and occupancy rates.
- Airport_codes.csv – Maps airport codes to their respective sizes and locations.
- Tickets.csv – Provides additional ticket-level revenue information.

Upon importing the datasets, a series of **data integrity assessments and preprocessing steps** were carried out to ensure quality and consistency for downstream analysis.

4.1 Load Datasets

- Flights.csv
- Tickets.csv
- Airport_codes.csv

Use pandas.read_csv() to load each into a Data Frame.

Initial Data Filtering Steps

To ensure the analysis is aligned with the business objectives and relevant only to the U.S. domestic market, the following **filtering steps** were applied:

- Flight Dataset: Filtered to include only non-canceled flights.
- Tickets Dataset: Restricted to only round-trip ticket entries.
- Airports Dataset: Limited to U.S.-based medium and large airports only.

To understand the structure and quality of each dataset before performing in-depth analysis, the summary_statistics () function was applied to the Flights, Tickets, and Airports data. This function returns a high-level overview including:

- **Non-null value counts** for each column
- **Data types** of each field
- **Descriptive statistics** (mean, std, min, max, quartiles) for numerical columns
- **Overall dataset shape** (number of rows and columns)
-

This diagnostic step was critical to identify missing data, confirm column formats, and validate the readiness of each dataset for merging and analysis. It helped ensure consistency across the data pipeline and reduced potential issues during calculations related to cost, revenue, and route performance.

4.2 Removing duplicates:

Removes duplicates from Flights, Tickets, and Airports using find_and_remove_duplicates() function. The process logs initial and post-cleaning row counts, eliminates duplicates, and returns structured datasets. This ensures data consistency while maintaining transparency through clear metrics.

```
Flights - Total Rows Before Cleaning: 1864272
Flights - Duplicate Rows Found: 4410
Flights - Total Rows After Removing Duplicates: 1859862
Tickets - Total Rows Before Cleaning: 708600
Tickets - Duplicate Rows Found: 47564
Tickets - Total Rows After Removing Duplicates: 661036
Airports - Total Rows Before Cleaning: 858
Airports - Duplicate Rows Found: 0
Airports - Total Rows After Removing Duplicates: 858
```

4.3 Eliminate unnecessary columns:

To get the best results from our analysis, we need to keep only valuable information and get rid of unnecessary columns. For example, I removed the *reporting_carrier* column because we're not trying to compare different airlines. I also took out the *year* and *quarter* columns since the data only covers one quarter, so those aren't useful. This helps us focus on what really matters and makes the analysis clearer.

Flights	OP_CARRIER','OP_CARRIER_FL_NUM','ORIGIN_AIRPORT_ID', 'DEST_AIRPORT_ID','CANCELLED','AIR_TIME'
Tickets	ITIN_ID','YEAR','QUARTER','ORIGIN_COUNTRY','ORIGIN_STATE_ABR', 'REPORTING_CARRIER','ORIGIN_STATE_NM'
Airport_codes	'CONTINENT','ISO_COUNTRY','MUNICIPALITY','ELEVATION_FT'

4.4 Data Type Formatting

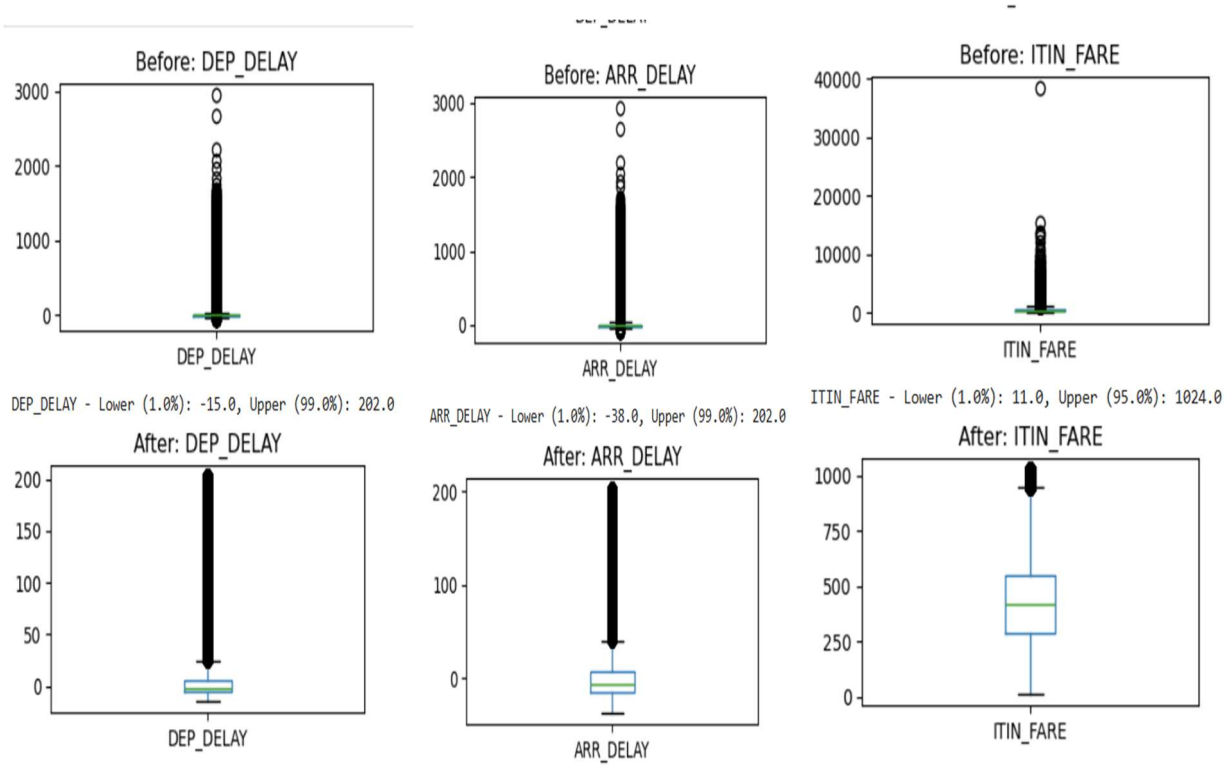
This step focuses on converting object-type columns into appropriate numerical and datetime formats for accurate analysis.

- The DISTANCE column in the flights dataset and the ITIN_FARE column in the tickets dataset were initially stored as text, often with unwanted characters. These were cleaned and converted into numbers using a custom function.
- The FL_DATE column was converted from a string to a datetime format to ensure proper handling of date-related operations later in the analysis.

4.5 Handling Outliers:

In this step, I identified extreme values in key numeric columns DEP_DELAY, ARR_DELAY, and ITIN_FARE using quantile thresholds. Values falling outside the defined lower and upper bounds

were considered outliers and replaced with NaN. These were then filled with the column's median to maintain data consistency without heavily skewing the results. Visual boxplots were used before and after the cleanup to verify the effectiveness of the process.



4.6 Handling Missing Values

This function cleans the dataset by addressing missing values based on column type. For object-type columns, it removes any rows containing nulls to avoid incomplete categorical data. For numeric columns, it fills missing values with the column's median to preserve the overall distribution. The function returns the cleaned dataset.

```
Filled 268 nulls in numeric column 'DISTANCE' with median value 612.0.
Filled 31 nulls in numeric column 'OCCUPANCY_RATE' with median value 0.65.
Filled 960 nulls in numeric column 'PASSENGERS' with median value 1.0.
Dropped 37 rows due to nulls in object column 'IATA_CODE'.
```

5 Data Transformation:

5.1 Merging and Transforming datasets:

Route Standardization:

- Tickets Data: Created a consistent ROUTE identifier by sorting origin/destination IATA codes alphabetically (e.g., JFK-LAX for both JFK→LAX and LAX→JFK).
- Flights Data: Applied the same sorting logic to ensure bidirectional routes are treated identically.

Ticket Fare Aggregation:

- Calculated median fares per route, accounting for ticket frequency to better represent revenue potential.

Airport Metadata Merge:

- Enriched flight data with airport types (medium_airport/large_airport) by merging:
- Once for origins (added as ORIGIN_TYPE)
- Once for destinations (added as DESTINATION_TYPE)

Final Merge:

- Combined flight data with aggregated ticket statistics using the standardized ROUTE key.
- Used a left join to preserve all flights even if ticket data was missing (though filtered later for round-trip pairs).

Purpose:

- Ensured bidirectional routes (e.g., JFK-LAX and LAX-JFK) are analyzed as one entity.
- Incorporate operational costs (via airport types) and revenue estimates (via ticket fares) into flight records.
- Supports accurate profitability calculations for round-trip analysis.

```
[12]: #consistent route identifiers for tickets
tickets['ROUTE'] = tickets.apply(lambda x: '-'.join(sorted([x['ORIGIN'], x['DESTINATION']])), axis=1)

[13]: ticket_stats =tickets.groupby('ROUTE').agg({'ITIN_FARE': np.median}).reset_index().\
                                                    rename(columns={'ITIN_FARE': 'ITIN_FARE_MEDIAN'})

[14]: airports = airports[['IATA_CODE', 'TYPE', 'COORDINATES']]
flights = flights.merge(airports, left_on='ORIGIN', right_on='IATA_CODE', how='inner', suffixes=('', '_origin')).rename(columns={\
    'TYPE': 'ORIGIN_TYPE', 'COORDINATES': 'ORIGIN_COORDINATES'})

flights = flights.merge(airports, left_on='DESTINATION', right_on='IATA_CODE', how='inner', suffixes=('', '_destination')).rename(columns={\
    'TYPE': 'DESTINATION_TYPE', 'COORDINATES': 'DESTINATION_COORDINATES'})

flights = flights.drop(columns=['IATA_CODE','TAIL_NUM', 'IATA_CODE_destination'], errors='ignore')

[15]: #Create consistent route identifiers for flights
flights['ROUTE'] = flights.apply(lambda x: '-'.join(sorted([x['ORIGIN'], x['DESTINATION']])), axis=1)

[16]: flights = flights.merge(ticket_stats, on='ROUTE', how='left')
```

6 Solutions and Data Visualization

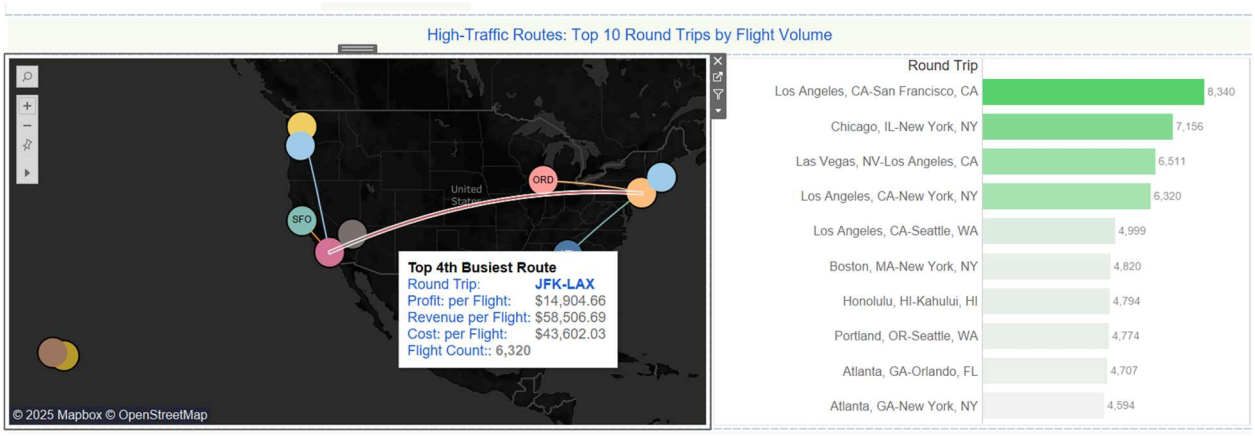
6.1 Question 1: Top 10 busiest round-trip routes

1. The 10 busiest round-trip routes in terms of number of round-trip flights in the quarter.

Exclude canceled flights when performing the calculation.

This analysis identifies the ten most frequently flown round-trip routes during the first quarter of 2019, based on actual (non-canceled) flight counts. By grouping flights by city pairs and summing their occurrences, we gain insights into the highest-demand travel corridors. These high-traffic routes are crucial for understanding passenger flow and guiding operational focus.

```
Top 10 Busiest Round Trip Routes in 1st quarter:
         ROUTE  FLIGHT_COUNT
2140  LAX-SFO          8340
2180  LGA-ORD          7156
2045  LAS-LAX          6511
1989  JFK-LAX          6320
2139  LAX-SEA          4999
542   BOS-LGA          4820
1751  HNL-OGG          4794
2590  PDX-SEA          4774
189   ATL-MCO          4707
185   ATL-LGA          4594
```



6.2 Question 2: 10 most profitable round-trip routes:

2. The 10 most profitable round-trip routes (without considering the upfront airplane cost) in the quarter. Along with the profit, show total revenue, total cost, summary values of other key components and total round-trip flights in the quarter for the top 10 most profitable routes. Exclude canceled flights from these calculations.

This analysis ranks the top 10 round-trip routes by total profit during the first quarter of 2019, excluding canceled flights and upfront airplane costs. Profit is calculated by subtracting total operational costs (e.g., fuel, delays, airport fees, depreciation) from total revenue (ticket sales and baggage fees). The output includes each route's **total revenue, total cost, total profit, and number of completed flights**.

This helps highlight the most financially efficient routes and supports strategic decision-making around route optimization and resource allocation.

Top 10 Most Profitable Round Trip Routes in 1st quarter:

	ROUTE	TOTAL_PROFIT	TOTAL_REVENUE	TOTAL_COSTS	FLIGHT_COUNT
1117	DCA-ORD	152808294.80	252095571.00	99287276.20	3695
131	ATL-CLT	135134828.32	204783444.00	69648615.68	3076
1101	DCA-LGA	127816979.32	207374552.00	79557572.68	3359
838	CLT-GSP	126039000.50	159176160.00	33137159.50	1547
2140	LAX-SFO	120960920.60	330178970.00	209218049.40	8340
542	BOS-LGA	120096223.60	234032882.00	113936658.40	4820
2406	MSP-ORD	119294676.20	203440104.00	84145427.80	3440
2180	LGA-ORD	115301051.36	321442996.00	206141944.64	7156
140	ATL-DCA	113291083.52	202820886.00	89529802.48	3488
1290	DFW-IAH	111553331.40	180050832.00	68497500.60	2955

High Profitable Round Trip Routes Summary:

These routes showed the highest total profits in Q1 and should be prioritized for continued investment and optimization:

- DCA-ORD – $152.8Mprofitacross3,695flights - ATL - CLT$–135.1M profit with strong ROI
- DCA-LGA – $127.8Mprofit; consistentlystrongperformer - CLT - GSP$–126M profit and also a recommended investment route

6.3 Question 3: Recommend 5 Best Routes to Invest In

3. The 5 round trip routes that you recommend investing in based on any factors that you choose.

To identify the most recommended round-trip flight routes for investment, a multi-factor evaluation was conducted using Return on Investment (ROI), profitability, Flight volume and, and Delay Impacts.

- Return on Investment (ROI): Measures how efficiently each route converts operating costs into profit. ROI = (Total Profit/Total Cost) *100
- Total Profit: Captures overall earnings from each route in the quarter.
- Flight Volume: Indicates demand and scalability potential.
- Average Delays: Reflects punctuality, a critical factor aligned with the airline's "**On time, for you**" brand promise.

This method ensures investment recommendations are grounded in operational performance, not just capital expenditure.

These routes demonstrate:

- Exceptionally high operational ROI (400%)

- Strong profitability (up to $95M in a quarter)
- Low average delays (many with early arrivals)
-

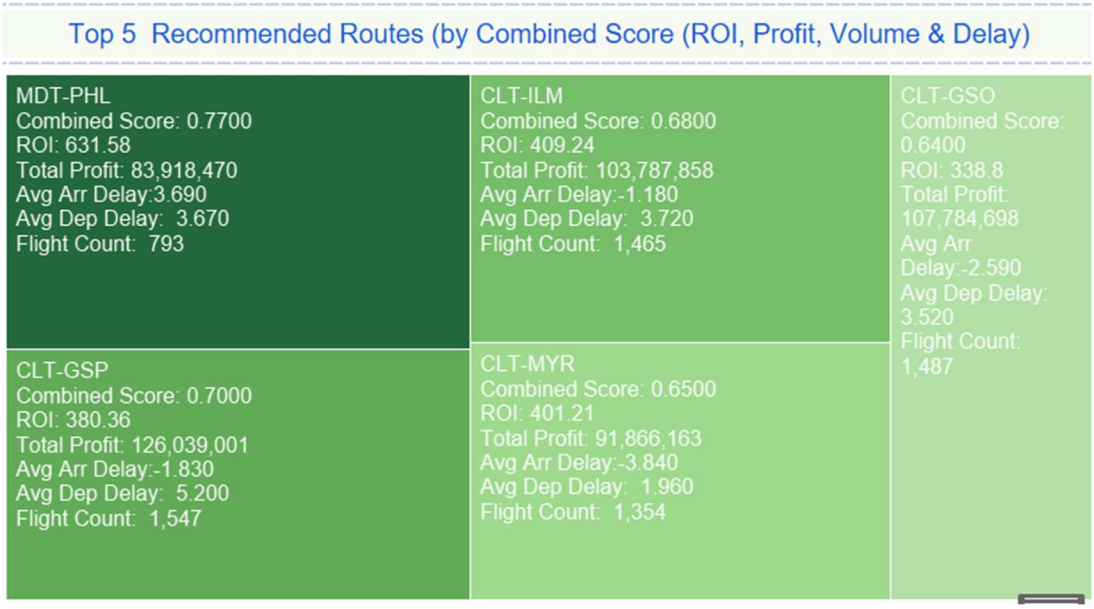
```
Top 5 Recommended Routes to Invest In:
       ROUTE  COMBINED_SCORE    ROI  TOTAL_PROFIT  FLIGHT_COUNT  \
2294  MDT-PHL           0.77 631.58   83918469.84           793
838   CLT-GSP           0.70 380.36  126039000.50          1547
846   CLT-ILM           0.68 409.24  103787857.50          1465
873   CLT-MYR           0.65 401.21   91866162.96          1354
837   CLT-GSO           0.64 338.80  107784698.22          1487

      AVG_DEP_DELAY  AVG_ARR_DELAY
2294           3.67           3.69
838            5.20          -1.83
846            3.72          -1.18
873            1.96          -3.84
837            3.52          -2.59
```

Top 5 Recommended Routes Summary:

These routes show strong ROI, high profitability per flight, and relatively low average delays:

- MDT-PHL – Highest ROI (631.58%), low delays, breakeven best case at just 708 flights
- CLT-GSP – High combined score (0.70), profitable, already among top 10 routes
- CLT-ILM – Strong ROI (409.24%), good profit with minimal delay
- CLT-MYR – Excellent ROI and very low arrival delays (-3.84 mins avg)
- CLT-GSO – Over $100M in profit with reliable on-time performance

Tree maps visual:



6.4 Question 4: round trip flights to breakeven on the upfront airplane cost

4. The number of round-trip flights will take to breakeven on the upfront airplane cost for each of the 5 round trip routes that you recommend. Print key summary components for these routes.

This analysis calculates the number of flights required for 5 recommended round-trip airline routes to breakeven on an upfront airplane cost $90M. It considers:

- Average profit per flight
- Total profit earned.
- Flight count completed.

Actionable Takeaways:

- Profitable Routes: Invest more in Chicago-Washington, DC and New York-Washington, DC, as they breakeven fastest and generate the most profit.
- Optimize High-Volume Routes: For Boston-New York, investigate ways to increase per-flight profit (ex: higher ticket prices, additional revenue).

Breakeven Analysis for Recommended Routes:

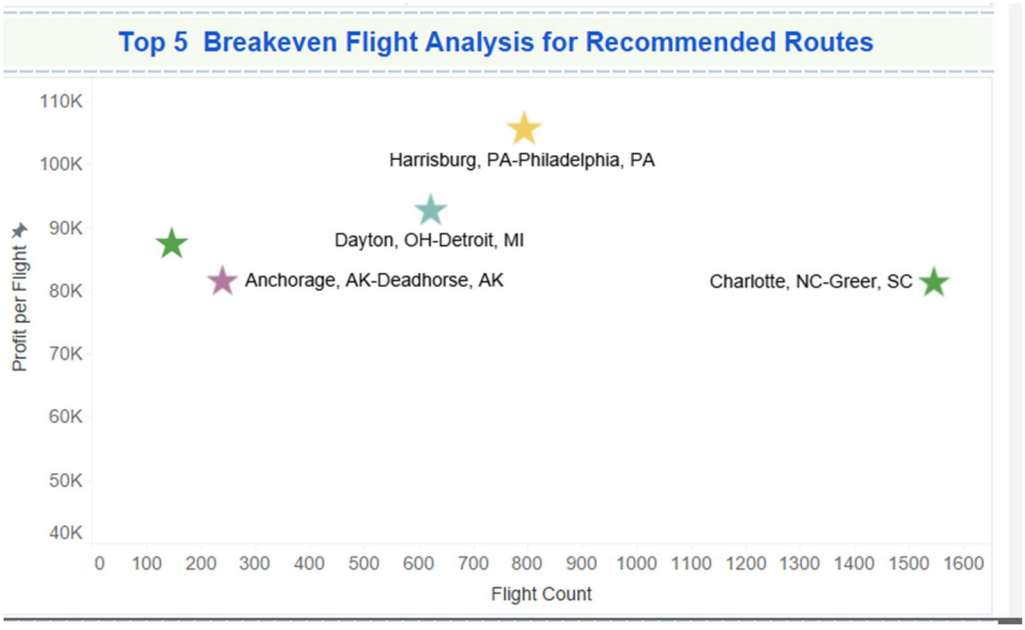
	ROUTE	PROFIT_PER_FLIGHT	BREAKEVEN_FLIGHTS	BREAKEVEN_BEST_CASE	BREAKEVEN_WORST_CASE
2294	MDT-PHL	105824.05	850.47	708.72	1063.09
1067	DAY-DTW	92799.87	969.83	808.19	1212.29
1251	DEN-SUN	87271.25	1031.27	859.39	1289.08
838	CLT-GSP	81473.17	1104.66	920.55	1380.82
1962	ISP-PHL	80871.52	1112.88	927.40	1391.10

Breakeven Analysis for Recommended Routes Summary:

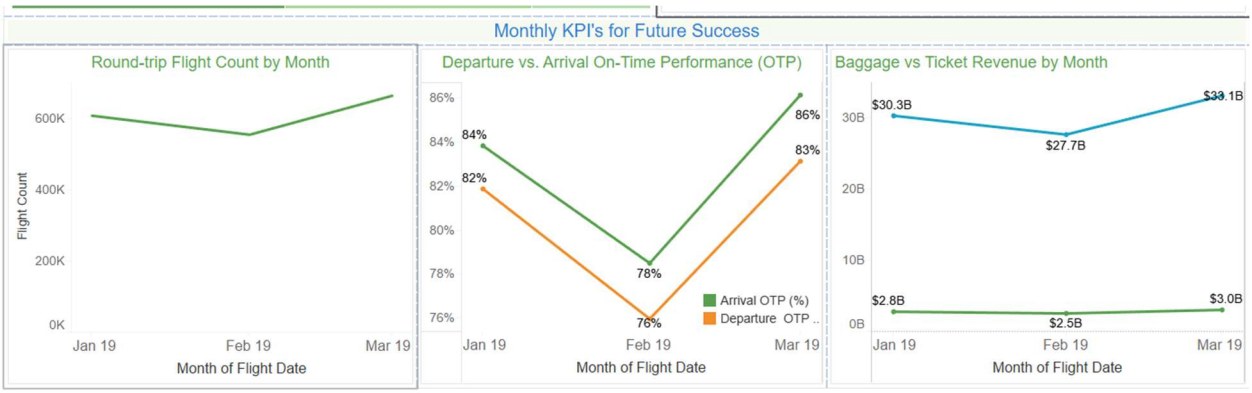
The following routes show favorable breakeven flight volumes, indicating high feasibility for reaching profitability quickly:

- MDT-PHL – Breaks even in as few as 708 flights (best case)
- CLT-GSP – Breaks even in 920 to 1,380 flights
- DEN-SUN, DAY-DTW, and ISP-PHL – All show high per-flight profitability and manageable breakeven thresholds

Scatter plot:



6.5 Question 5: Key Performance Indicators (KPI's)




```
print("Average Load Factor:", route_summary['LOAD_FACTOR'].mean(), "%")
```

```
Average Load Factor: 65.00432666783888 %
```

```
print("Average RASM (Revenue per Available Seat Mile):", route_summary['RASM'].mean())
```

```
Average RASM (Revenue per Available Seat Mile): 4.4743746668083615
```

```
print("Average CASM (Cost per Available Seat Mile):", route_summary['CASM'].mean())
```

```
Average CASM (Cost per Available Seat Mile): 0.7904507207321467
```

```
#On-time Performance (Departure and Arrival Delays)
on_time_departure = (flights['DEP_DELAY'] <= 15).mean() * 100
on_time_arrival = (flights['ARR_DELAY'] <= 15).mean() * 100

print(f"On-Time Departure Rate: {on_time_departure:.2f}%")
print(f"On-Time Arrival Rate: {on_time_arrival:.2f}%")
```

```
On-Time Departure Rate: 82.96%
On-Time Arrival Rate: 81.98%
```

Recommendations:

- Invest in top-scoring routes with strong ROI and fast breakeven potential, especially MDT-PHL and CLT regional routes.
- Maintain and expand service on high-profit legacy routes, such as DCA-ORD and ATL-CLT, to secure continued revenue growth.

- Leverage CLT as a strategic hub, focusing on regional connectivity to maximize profitability and route efficiency.

Conclusion:

Investing in data-driven, high-performing routes with strong profitability and operational efficiency will strengthen financial returns. Focused expansion around proven hubs like CLT, alongside disciplined performance tracking, will ensure sustainable and scalable route growth.

Key Metrics for Success

Monitoring effectively Key Performance Indicators (KPIs): Tableau Market Analysis Dashboard

- Total Revenue per Route: Measures the overall income generated by each route.
- Total Costs per Route: Tracks all operational expenses associated with each route.
- Profit Margin per Route: (Total Revenue - Total Costs) / Total Revenue, indicating the profitability of each route.
- Return on Investment (ROI) per Route: (Total Profit / Total Costs) * 100, assessing the efficiency of capital utilization.
- On-Time Performance (Departure & Arrival): Percentage of flights departing and arriving within 15 minutes of the scheduled time.

By consistently monitoring these KPIs, the airline can gain valuable insights into the financial viability, operational efficiency, and customer perception of each route. This data-driven approach will enable informed decisions regarding route adjustments, marketing efforts, and overall strategic direction to ensure sustainable and profitable growth in the US domestic market.