My Projects

Project Name	About	Tools Used	Link
Airline Customer Ratings Visualization	Created a dashboard in Tableau to visualize customer ratings for an Airline company	Tableau (Visualizations, Parameters, Calculations)	
Stocks Portfolio Optimization	Generated optimal portfolios for a combination of 5 stocks, displayed as an Efficient Frontier	Python, Yahoo Finance	
Supermarket Analysis - SQL	Used SQL queries on the database of a supermarket to address various business questions	SQL	

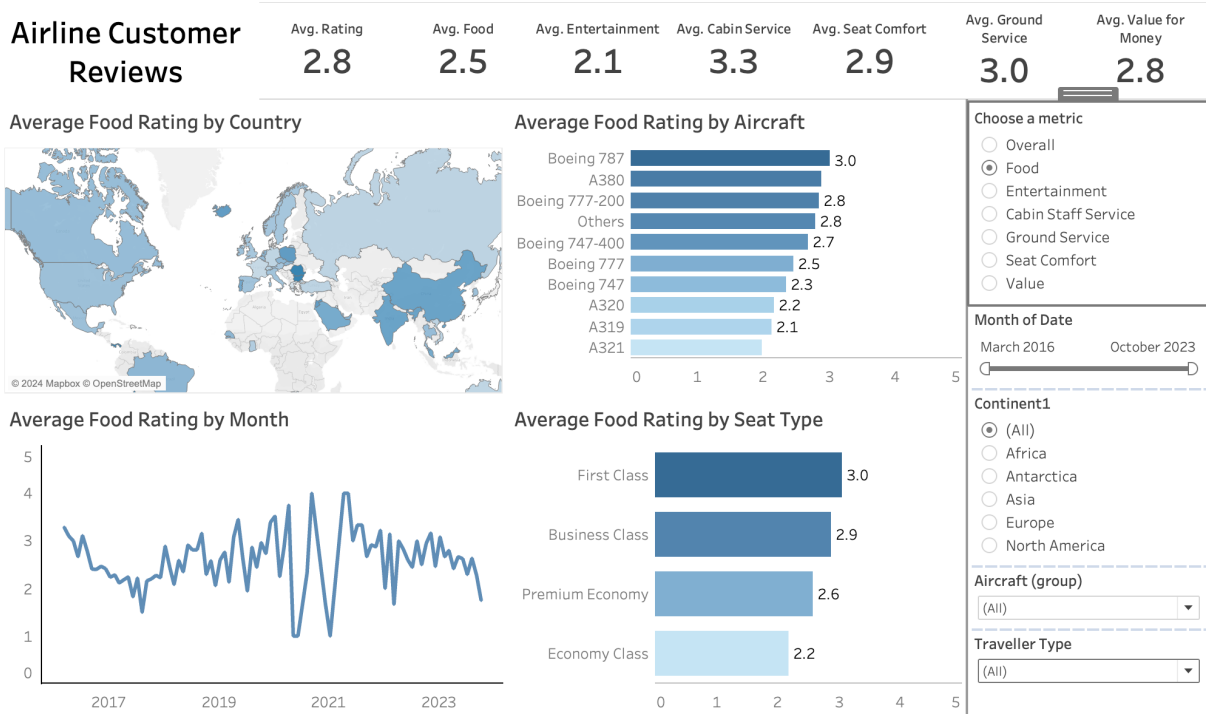
Airline Customer Reviews Visualization

In this project, I used Tableau to create a visualization dashboard of customer ratings data for an airline company. The dashboard is accessible by clicking on the project title above.

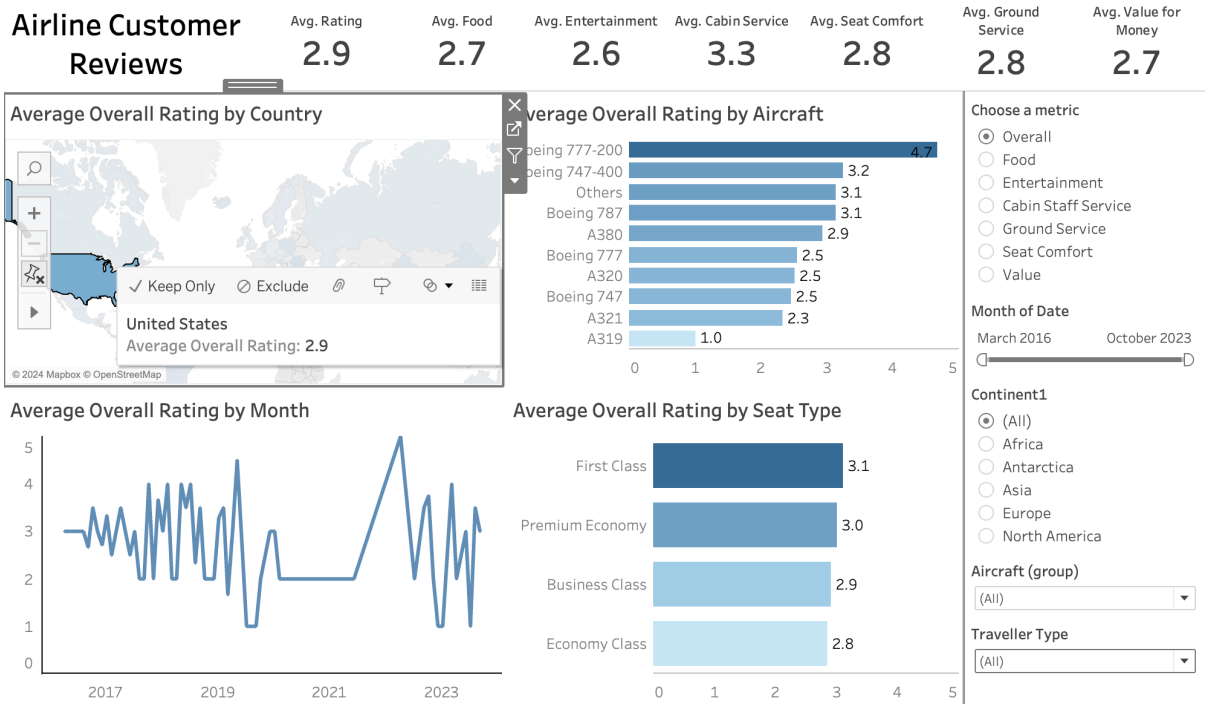
The dashboard allows the user to get a quick idea of the average customer ratings across a variety of important dimensions. A world map allows viewing how the ratings vary across regions. In addition, other charts allow analyzing the ratings over time and by aircraft type & seat type.

The dynamic nature of the dashboard allows an easy visualization of how all metrics change by selecting any specific criteria. Few screenshots given below.

- The user can select from a host of filters given in the pane on the right and the dashboard updates accordingly:



- The user can also select any criteria from within the visualizations and the dashboard updates accordingly e.g. selecting United States on the map:

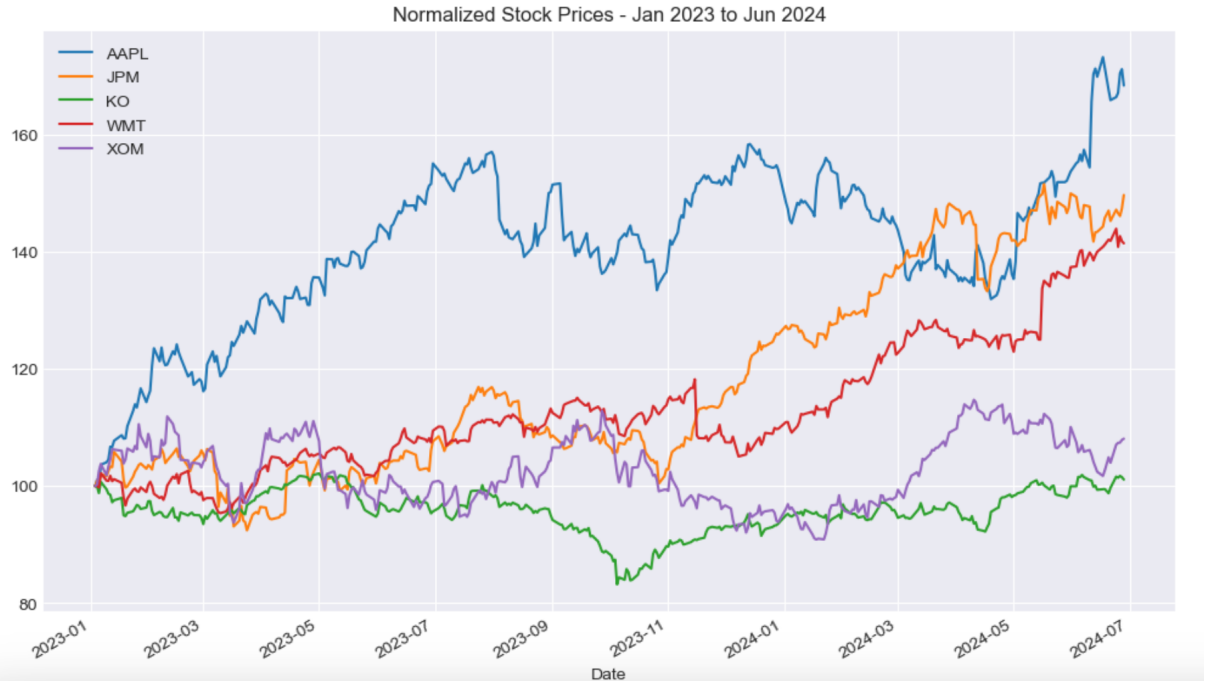


Stocks Portfolio Optimization

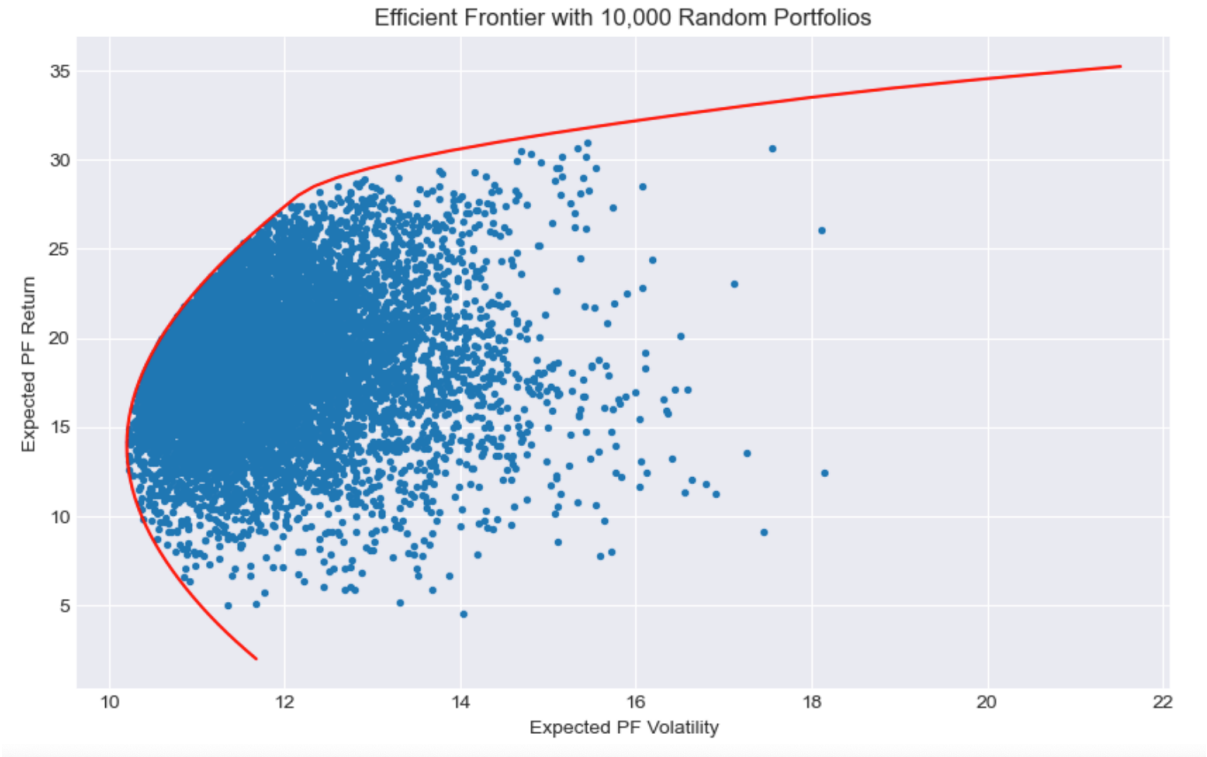
In this project, I used Python to perform portfolio optimization on a Portfolio of 5 stocks. I selected the stocks of the companies listed below, and extracted their

historical share price movement data from the Yahoo Finance library in Python fron Jan 2023 to June 2024.

- Apple Inc.
- Tesla Inc.
- Microsoft Corporation
- Amazon.com Inc.
- Alphabet Inc.



I generated 10,000 random portfolios and plotted them to visualize the Expected Return vs. Volatility (risk) of the random portfolios. I also generated the Efficient Frontier, representing the portfolios with the lowest level of risk for a given level of expected return. Lastly, the optimal portfolio allocation was also calculated for a target return of 25%. Optimal allocations can be calculated similarly for different target return values and other selections of companies.



Supermarket Analysis - SQL

In this project done as part of my Masters coursework, I used SQL queries on the database of a supermarket to answer different business questions.

Some of the questions addressed, the relevant SQL queries and their outputs are given in pictures below.

1) Total Quantity sold per product, ordered by the highest quantity

```sql
-- Quantity Sold per product, ordered by highest quantity
SELECT itemCode, SUM(quantitySold) AS total_quantity_sold
FROM invoices
GROUP BY itemCode
ORDER BY total_quantity_sold DESC
```

ItemCode	total_quantity_sold
102900011016701	28164.331
102900005116714	27537.228
102900005116899	27149.44
102900005115960	19187.218
102900005115779	15910.461
106949711300259	15596
102900011030059	14325
102900005116257	13602.001

2) Getting a list of all items sold on a specific date (e.g. 2021-09-20 in this case) along with their quantity sold and total amount.

```sql
-- List of all items sold on a specific date (e.g., '2021-09-20') with quantity and total amount
SELECT itemCode, SUM(quantitySold) AS quantity_sold, SUM(Amount) AS total_amount
FROM invoices
WHERE date = '2021-09-20'
GROUP BY itemCode
```

ItemCode	quantity_sold	total_amount
102900005115762	7.743	38.715
102900005115779	51.89	453.928
102900005115786	6.009	36.054
102900005115793	4.241	76.338
102900005115823	11.808	106.272
102900005115854	4.719	23.595
102900005115878	4.55	54.6

3) Getting a list of products which had no sales

```sql
-- Finding products with no sales by joining products with invoices
SELECT p.itemCode, p.itemName
FROM products p
LEFT JOIN invoices i ON p.itemCode = i.itemCode
WHERE i.itemCode IS NULL
```

ItemCode	ItemName
102900005116776	Local Spinach
102900005116042	Lotus Root
102900011023648	Wuhu Green Pepper (2)
102900011032145	Wuhu Green Pepper (Bag)
102900011011782	Chinese Caterpillar Fungus Flowers (Box) (1)

4) Getting the count of distinct products within each category

```sql
-- Count the number of distinct products in each category
SELECT c.categoryName, COUNT(p.itemCode) AS num_products
FROM categories c
LEFT JOIN products p ON c.categoryCode = p.CategoryCode
GROUP BY c.categoryName
```

CategoryName	num_products
Aquatic Tuberous Vegetables	19
Cabbage	5
Capsicum	45
Edible Mushroom	72
Flower/Leaf Vegetables	100
Solanum	10

5) Getting the total sales amount for each category

```sql
-- Obtain the total sales amount for each category
SELECT c.categoryName, SUM(i.Amount) AS total_sales
FROM invoices i
JOIN products p ON i.itemCode = p.itemCode
JOIN categories c ON p.CategoryCode = c.categoryCode
GROUP BY c.categoryName
```

CategoryName	total_sales
Aquatic Tuberous Vegetables	350089.9007
Cabbage	375751.705
Capsicum	754133.0167
Edible Mushroom	619597.7942
Flower/Leaf Vegetables	1079069.8016
Solanum	191124.2586