# Automotive Sales Analytics Platform

## Project Idea:

The project's main goal is to derive useful insights from an automotive company's sales data by utilizing exploratory data analysis (EDA) and recency-frequency-monetary (RFM) analysis methodologies. In order to provide practical suggestions for improving customer happiness, inventory control, and sales tactics, ASAP looks for patterns, trends, and customer behavior.

To gain a more in-depth understanding of the dataset, we'll employ the big data idea of the 5Vs.

## Volume:

The dataset has 18 rows and 18 properties, resulting in a large amount of data. As we investigate order fulfillment in an automobile firm, the amount of the dataset is critical to understanding patterns, trends, and optimizing the order fulfillment process.

## Velocity:

In the context of order fulfillment, velocity, or the rate at which data is generated, is very important. The steady intake of sales data from diverse clients and areas helps to the dataset's ongoing growth. This high data velocity provides useful insights for the automaker to modify and streamline their order fulfillment tactics in real-time.

## Variety:

The dataset's properties are numerous, including order numbers, quantity ordered, pricing, customer details, and more. This structured data with various categories and established formats emphasizes the variation inherent in a car company's order fulfillment procedure.

## Veracity:

Veracity, which emphasizes the dependability and quality of data, is crucial for making educated judgments in order fulfillment. The dataset is thoroughly preprocessed to ensure correctness and reliability. By resolving any discrepancies or inaccuracies, we want to improve the data's authenticity, resulting in more trustworthy insights.

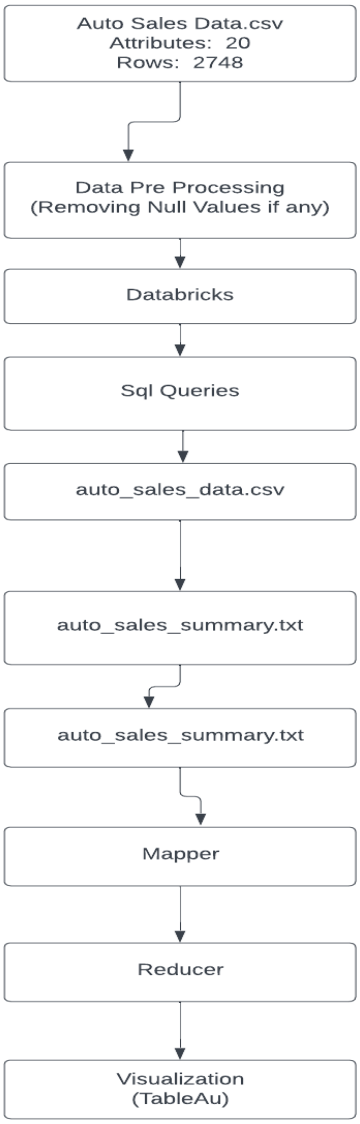
## Value:

The ultimate objective is to maximize order fulfillment by taking advantage of analytics to extract value from the dataset. The car manufacturer is able to increase customer value, increase efficiency, and make well-informed decisions thanks to thorough analysis and insights from the data.

## Tools and Technologies Used:

- **Microsoft Excel:** Microsoft Excel supports the storage of a wide range of data types within spreadsheet cells or ranges.
- **Databricks:** It is a tool for transforming complex data into formats that can be reused.
- **MySQL:** MySQL is a relational database management system (RDBMS) that makes it easier to extract useful information from a variety of queries.
- **Hadoop:** Based on Java, Hadoop uses simple programming techniques to distribute large datasets among machine clusters.
- **Map& Reduce:** Mapper tasks in Hadoop are in charge of breaking down jobs into smaller subtasks and handling intermediate data, whereas Reducers are in charge of organizing and condensing this intermediate data into more manageable units during the data processing pipeline.

## Architecture Diagram:




## Architecture Summary:

The auto sales dataset includes 2748 records that document a variety of sales information. It includes entries such as ORDERNUMBER, QUANTITYORDERED, PRICEEACH, and more. Our goal is to use SQL queries to import and extract important insights by utilizing Databricks. The output file, called "auto_sales.csv" and saved as "auto_sales.txt," will serve as a primary source for our research projects involving cars. By putting the "Map Reduce" technique into practice, we can further hone our analysis and ensure accuracy and precision when assessing auto sales trends and patterns.

## Goals:

The following are the five objectives and their corresponding goals.

1. To calculate evaluate the average days since the last order in different cities.
2. To understand the distribution of sales across different deal sizes. To find out the total average amount with respective Transfers.
3. To Identify the top customers based on the quantity of products purchased.
4. To Examine the trend in sales over different months and years.
5. To Analyze the average price of products sold in different countries.
6. To Understand the contribution of each product line to the total sales.

## Project Description:

### 1)Project Setup:

**Setting up the Project Environment:**

**1. Installing VS Code and Java Extensions:**

Installing VS Code as well as necessary Java modules. For coding, debugging, and project management, VS Code will be your primary Integrated Development Environment (IDE). To facilitate easy Java development, ensure that Java-related capabilities are smoothly integrated within VS Code.

**2. Setting up the Java Environment:**

Check the accuracy of your Java installation on your machine. Confirm the presence and proper configuration of Java. Check that the system environment variables are correctly configured and that the Java Development Kit (JDK) is properly added and available.

## Using the Automotive Sales Analytics Platform:

1. **Dataset's Availability:** I have confirmed the dataset's availability from the designated source, ensuring proper access permissions and repository storage.
2. **Dataset Structure Evaluation:** The dataset structure has been assessed, with columns checked for accuracy, data types verified, missing values handled, clear column labels ensured, and unique IDs and linkages between tables identified.
3. **Data Analysis:** We have successfully loaded the dataset, produced basic statistics, and conducted preliminary investigations for data analysis. To evaluate the dataset's structure and acquire insights into its content and properties, data quality checks, visualizations, and descriptive analysis were performed.

## Data Retrieval:

1. We have used VS Code for Java Programming.
2. Initialize a new Java file within Visual Studio Code (VS Code) for programming purposes.

3. **Map Reduce Compatibility:** After extracting data with Java, structure it into ordered key-value pairs to ensure interoperability with Hadoop MapReduce. To ease smooth use of MapReduce tools, save the processed files in a standard, readable format such as CSV.

## Data Preprocessing:

Inspect for missing data and plan how to handle it, taking into account imputation or removal. Remove duplicate entries to ensure data consistency. Filters should be used to extract relevant subsets that are aligned with project objectives.

## Hadoop MapReduce:

Make Java MapReduce applications to work with preprocessed data. Create a Mapper function to extract key-value pairs. For data processing and aggregation, use a Reducer function. Ensure that the output formatting is optimum for analysis and visualization.

## Data post-processing and Analysis:

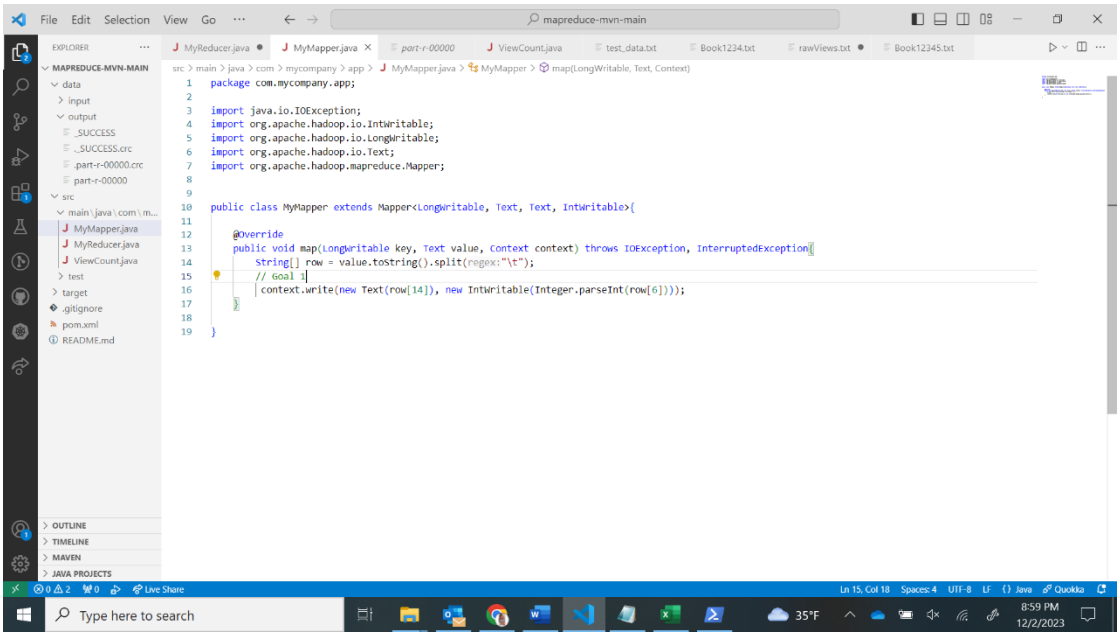
Perform a thorough examination of Map.Reduce outputs to reveal patterns and useful insights. Recognize key data patterns and trends in the processed data. Perform further computations and filtering to ensure compliance with project-specific standards. Learn about intricate patterns and trends to make more informed decisions. Conduct further analyses to get useful information from the improved data.

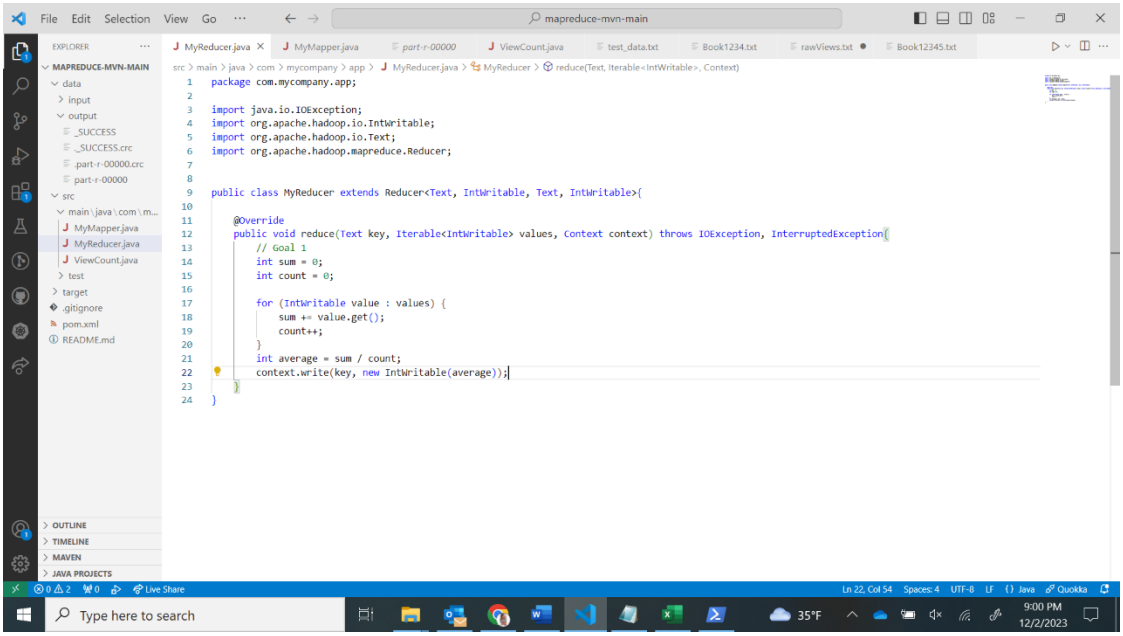
## Visual Presentation:

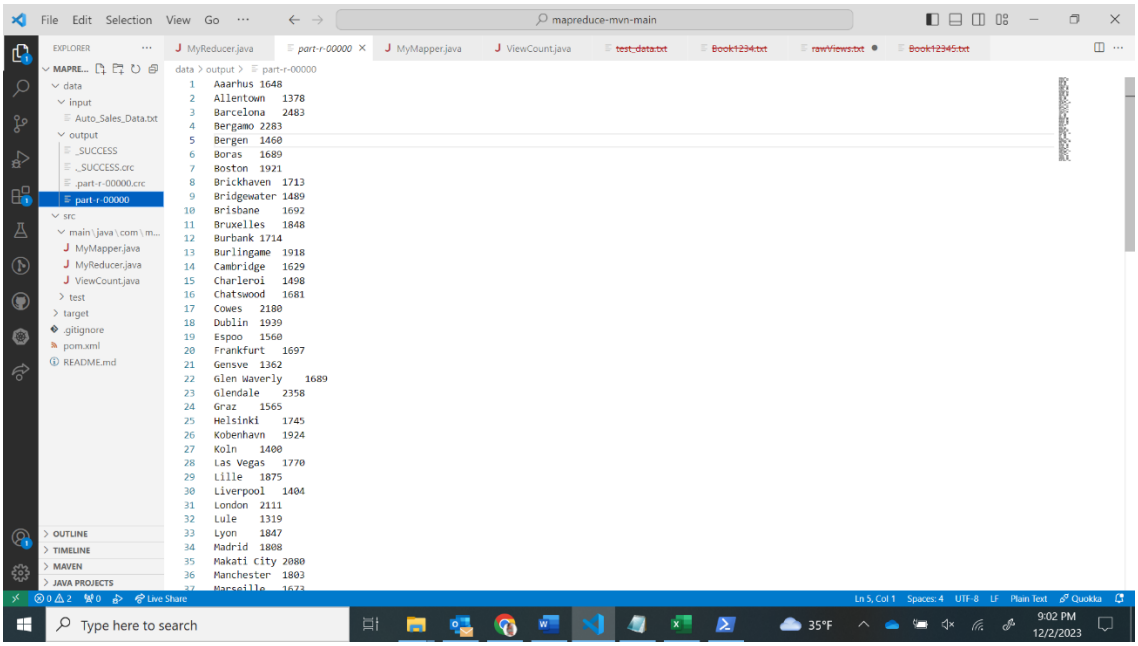
For Tableau compatibility, save processed data in CSV format. Import into Tableau for interactive visual representation. Create bespoke visuals that are in line with project goals. Create charts that compare bridge condition, age, traffic impact, type categorization, and load. Use Tableau for detailed data visualization.

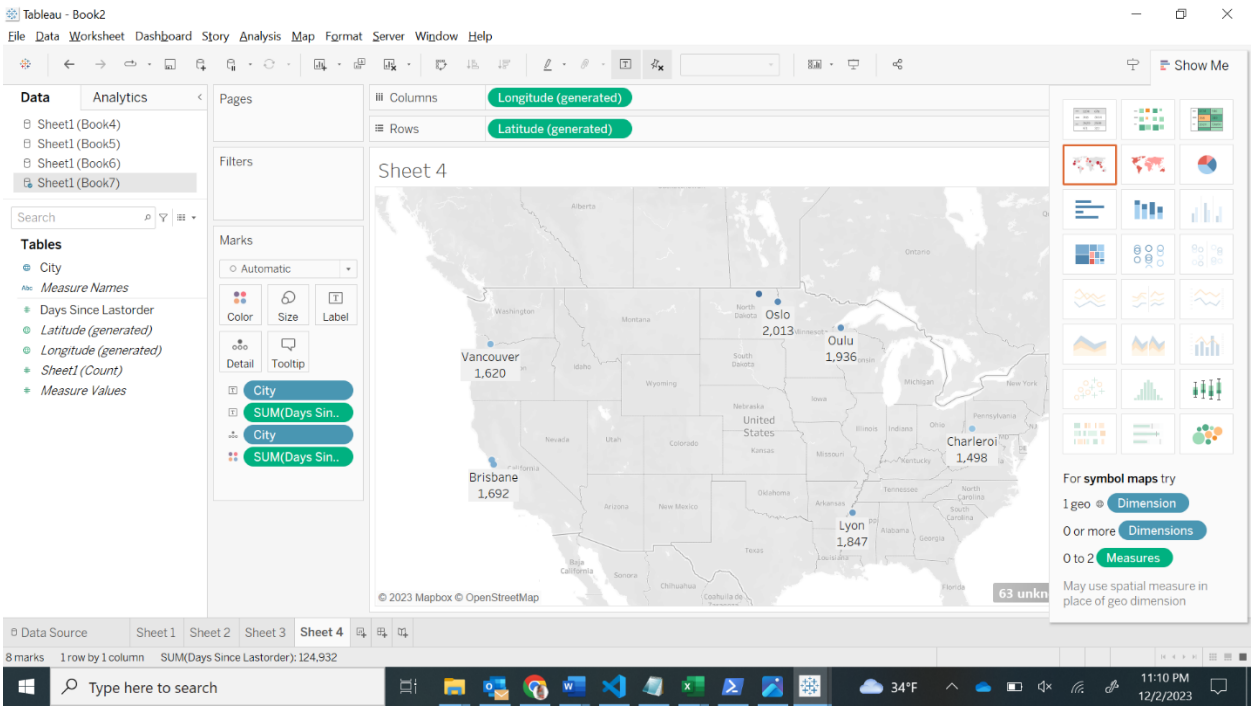
## Result Summary:

**Goal 1:** To calculate evaluate the average days since the last order in different cities.


File Edit Selection View Go
mapreduce-mvn-main
MyReducer.java
MyMapper.java
part-r-00000
ViewCount.java
test_data.txt
Book1234.txt
rawViews.txt
Book12345.txt
src > main > java > com > mycompany > app > MyMapper.java > MyMapper > map(LongWritable, Text, Context)
MAPREDUCE-MVN-MAIN
data
input
output
_SUCCESS
_SUCCESS.crc
.part-r-00000.crc
part-r-00000
src
main\java\com\m...
MyMapper.java
MyReducer.java
ViewCount.java
test
target
.gitignore
pom.xml
README.md
OUTLINE
TIMELINE
MAVEN
JAVA PROJECTS
package com.mycompany.app;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
public class MyMapper extends Mapper<LongWritable, Text, Text, IntWritable>{
@Override
public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException{
String[] row = value.toString().split(regex:"\t");
// Goal 1
context.write(new Text(row[14]), new IntWritable(Integer.parseInt(row[6])));
}
}
Ln 15, Col 18 Spaces: 4 UTF-8 LF Java Quokka
Live Share
Type here to search
35°F
8:59 PM
12/2/2023

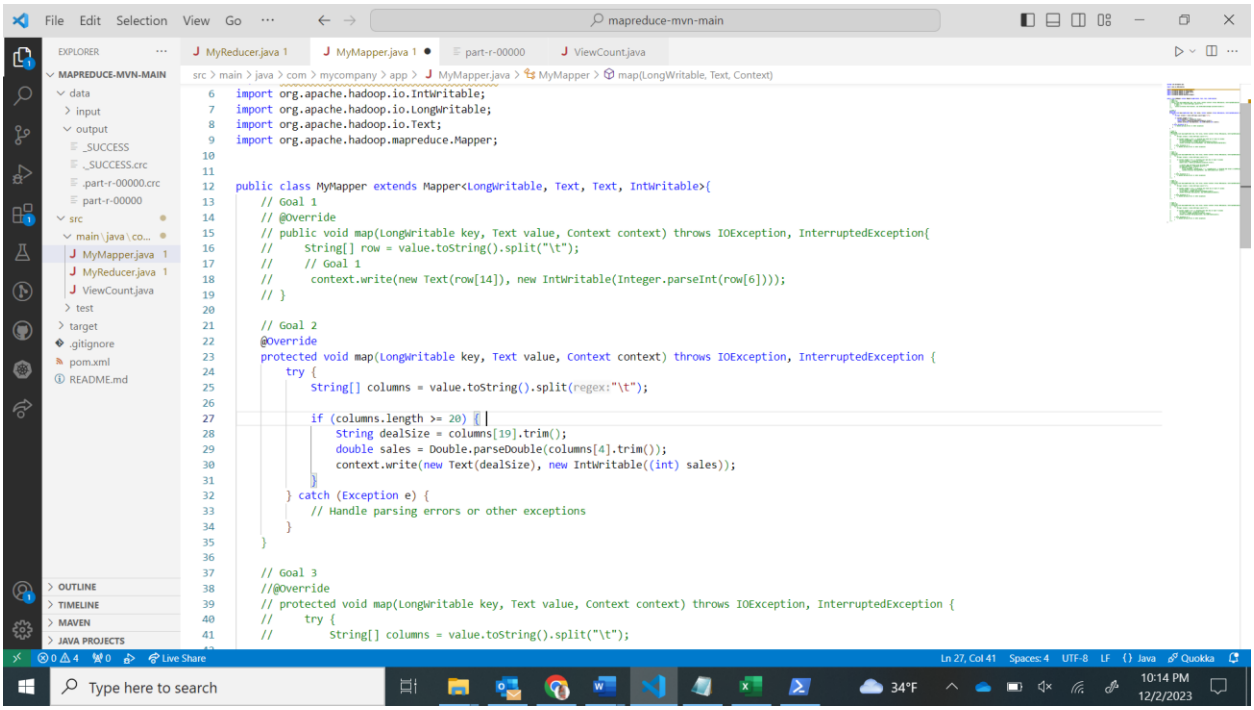

File Edit Selection View Go
mapreduce-mvn-main
MyReducer.java
MyMapper.java
part-r-00000
ViewCount.java
test_data.txt
Book1234.txt
rawViews.txt
Book12345.txt
src > main > java > com > mycompany > app > MyReducer.java > MyReducer > reduce(Text, Iterable<IntWritable>, Context)
MAPREDUCE-MVN-MAIN
data
input
output
_SUCCESS
_SUCCESS.crc
.part-r-00000.crc
part-r-00000
src
main\java\com\m...
MyMapper.java
MyReducer.java
ViewCount.java
test
target
.gitignore
pom.xml
README.md
OUTLINE
TIMELINE
MAVEN
JAVA PROJECTS
package com.mycompany.app;
import java.io.IOException;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;
public class MyReducer extends Reducer<Text, IntWritable, Text, IntWritable>{
@Override
public void reduce(Text key, Iterable<IntWritable> values, Context context) throws IOException, InterruptedException{
// Goal 1
int sum = 0;
int count = 0;
for (IntWritable value : values) {
sum += value.get();
count++;
}
int average = sum / count;
context.write(key, new IntWritable(average));
}
}
Ln 22, Col 54 Spaces: 4 UTF-8 LF Java Quokka
Live Share
Type here to search
35°F
9:00 PM
12/2/2023



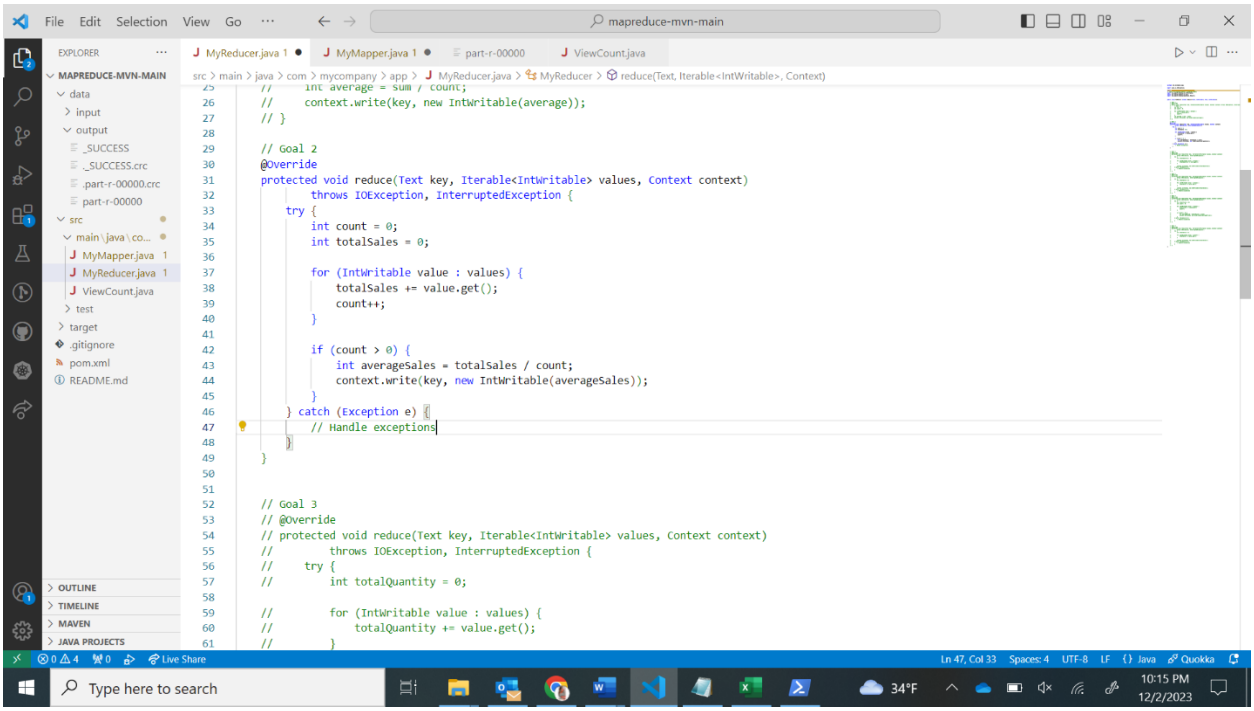




**Story:** Analyze the dataset to get the average number of days since the last order in each city. Investigate patterns to identify any geographic tendencies in client ordering behavior. Create an approach for calculating and evaluating the average number of days since the last order. Use city-level computations to gain insight about ordering frequency across multiple sites. Inform decision-making and improve operational efficiency by presenting findings.

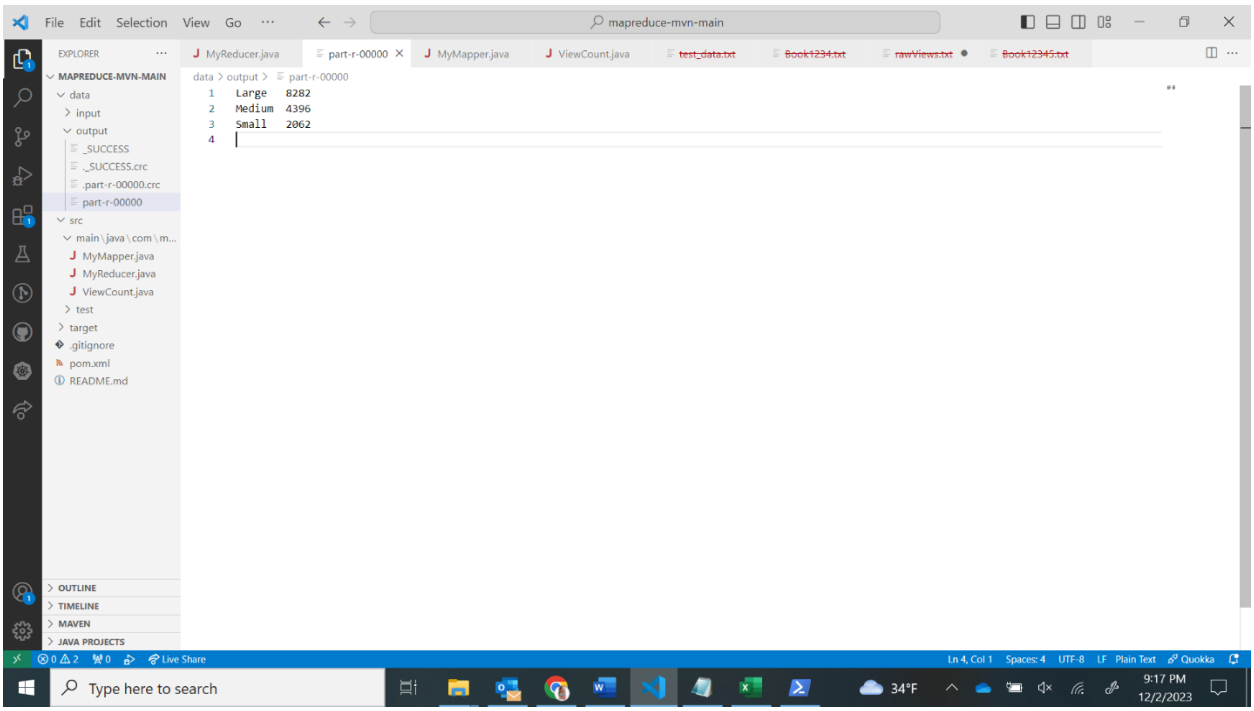
**Goal 2:** To understand the distribution of sales across different deal sizes. To find out the total average amount with respective Transfers.

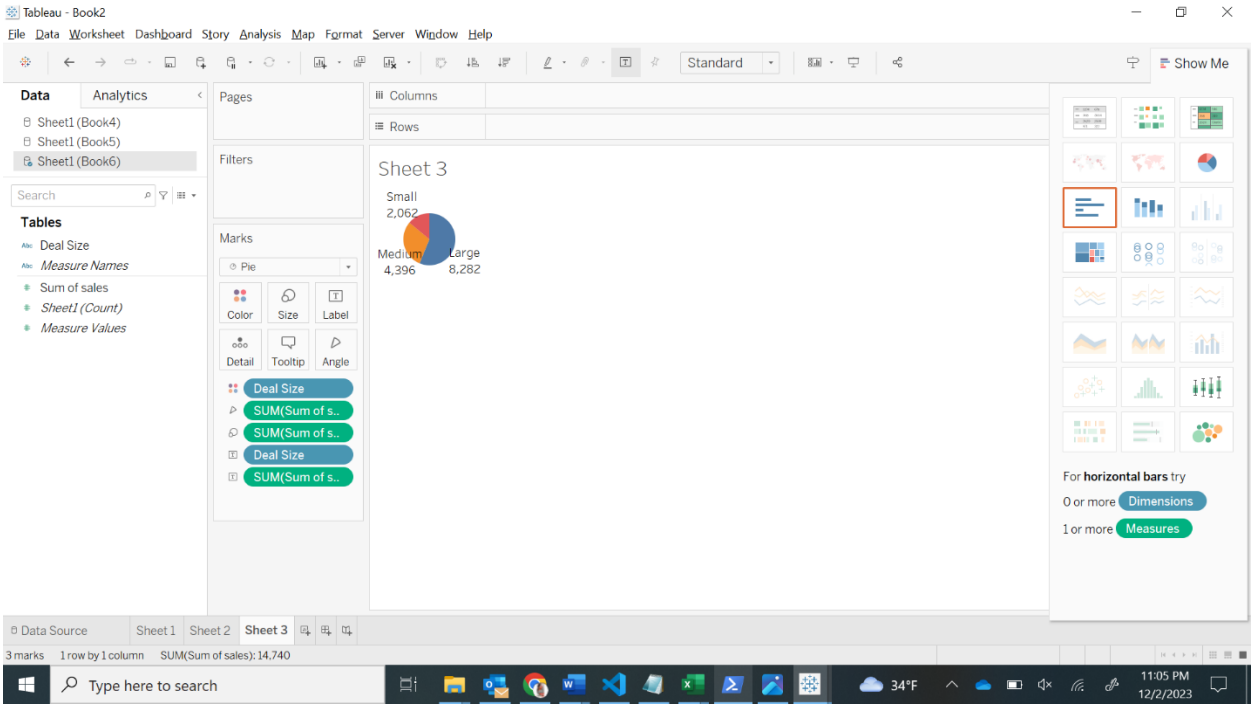


```java
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;


public class MyMapper extends Mapper<LongWritable, Text, Text, IntWritable>{
    // Goal 1
    // @Override
    // public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException{
    //     String[] row = value.toString().split("\t");
    //     // Goal 1
    //      context.write(new Text(row[14]), new IntWritable(Integer.parseInt(row[6])));
    // }

    // Goal 2
    @Override
    protected void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
        try {
            String[] columns = value.toString().split(regex:"\t");

            if (columns.length >= 20) {
                String dealSize = columns[19].trim();
                double sales = Double.parseDouble(columns[4].trim());
                context.write(new Text(dealSize), new IntWritable((int) sales));
            }
        } catch (Exception e) {
            // Handle parsing errors or other exceptions
        }
    }

    // Goal 3
    //@Override
    // protected void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
    //     try {
    //         String[] columns = value.toString().split("\t");
```



```java
    //     int average = sum / count;
    //     context.write(key, new IntWritable(average));
    // }

    // Goal 2
    @Override
    protected void reduce(Text key, Iterable<IntWritable> values, Context context)
            throws IOException, InterruptedException {
        try {
            int count = 0;
            int totalSales = 0;

            for (IntWritable value : values) {
                totalSales += value.get();
                count++;
            }

            if (count > 0) {
                int averageSales = totalSales / count;
                context.write(key, new IntWritable(averageSales));
            }
        } catch (Exception e) {
            // Handle exceptions
        }
    }


    // Goal 3
    // @Override
    // protected void reduce(Text key, Iterable<IntWritable> values, Context context)
    //         throws IOException, InterruptedException {
    //     try {
    //         int totalQuantity = 0;

    //         for (IntWritable value : values) {
    //             totalQuantity += value.get();
    //         }
```

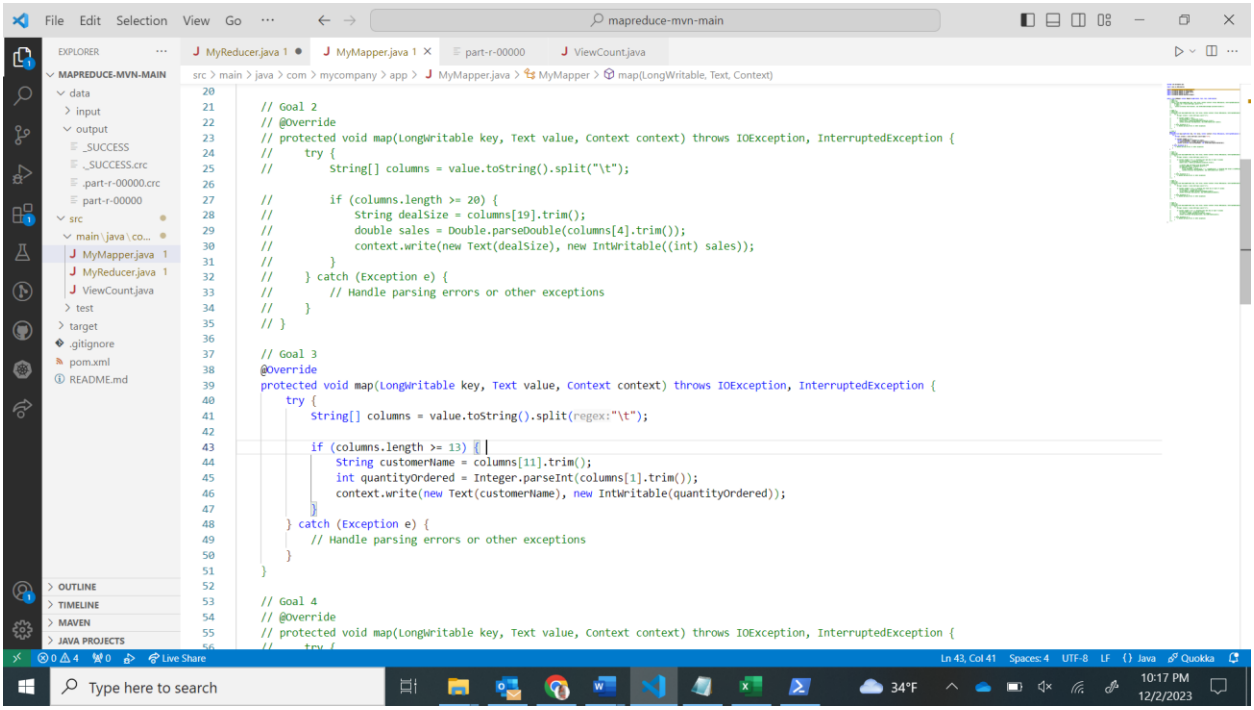


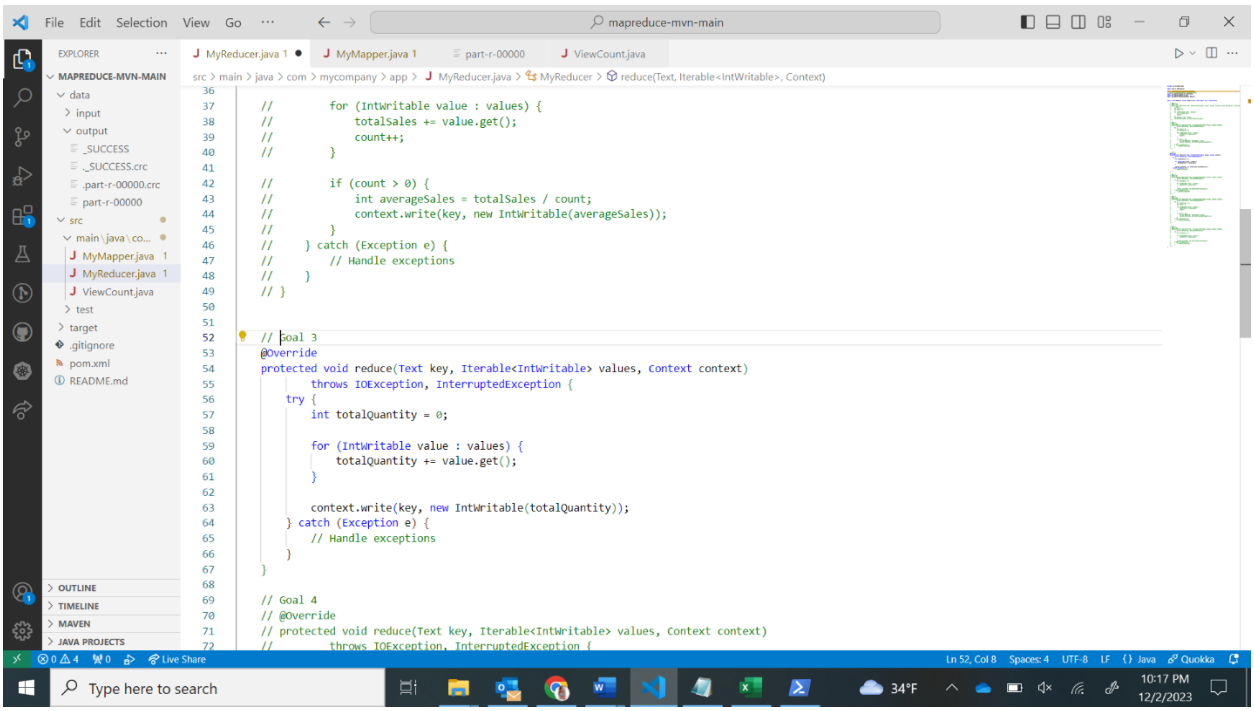



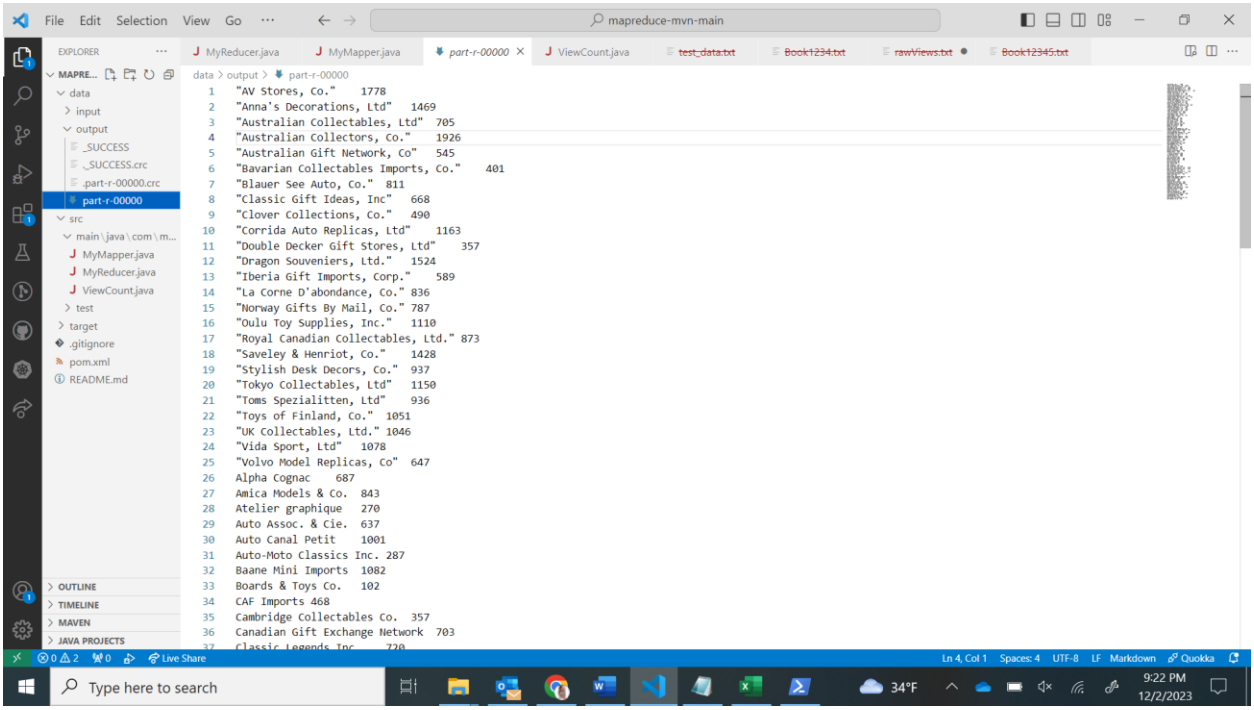

**Story:** Investigate the dataset to learn about the distribution of sales across various transaction sizes. Analyze transactional data to estimate the total average amount connected with various deal sizes and subsequent transfers. Develop insights into the financial landscape, identifying significant transaction categories and their impact on overall sales. Use this data to improve

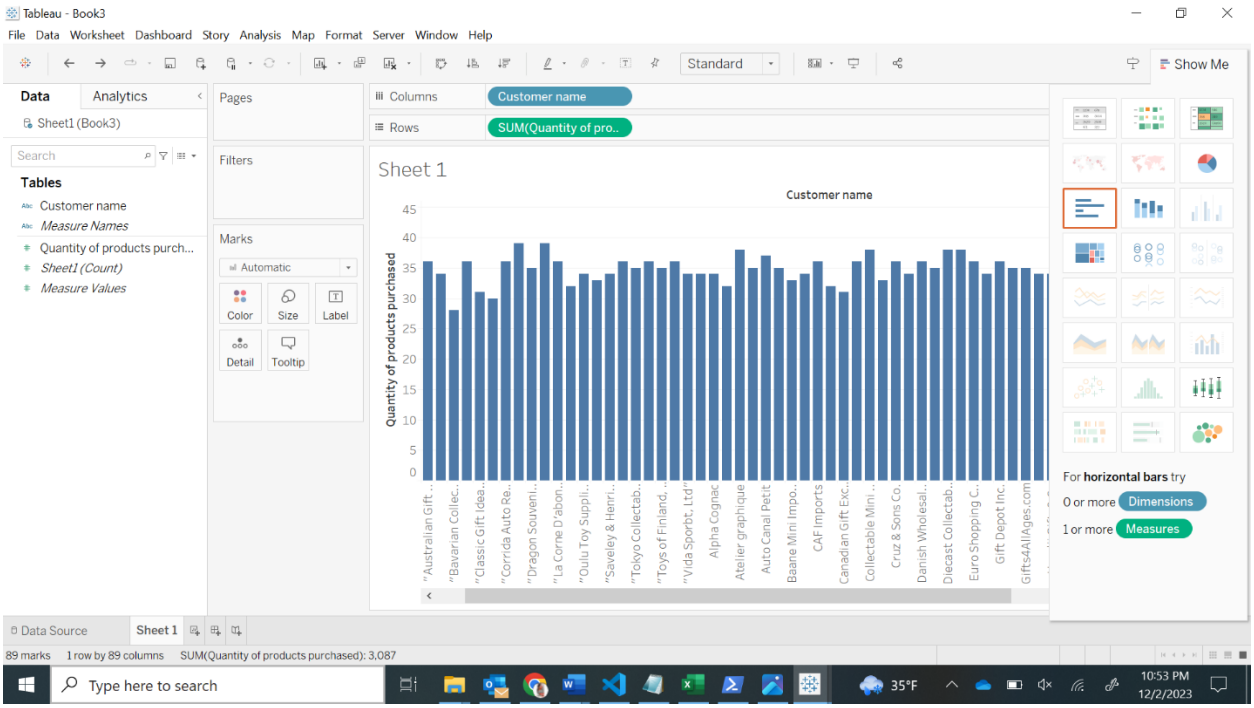
transaction tactics, financial planning, and overall business performance. Present insights for informed decision-making and strategic financial management.

**Goal 3**: To Identify the top customers based on the quantity of products purchased.



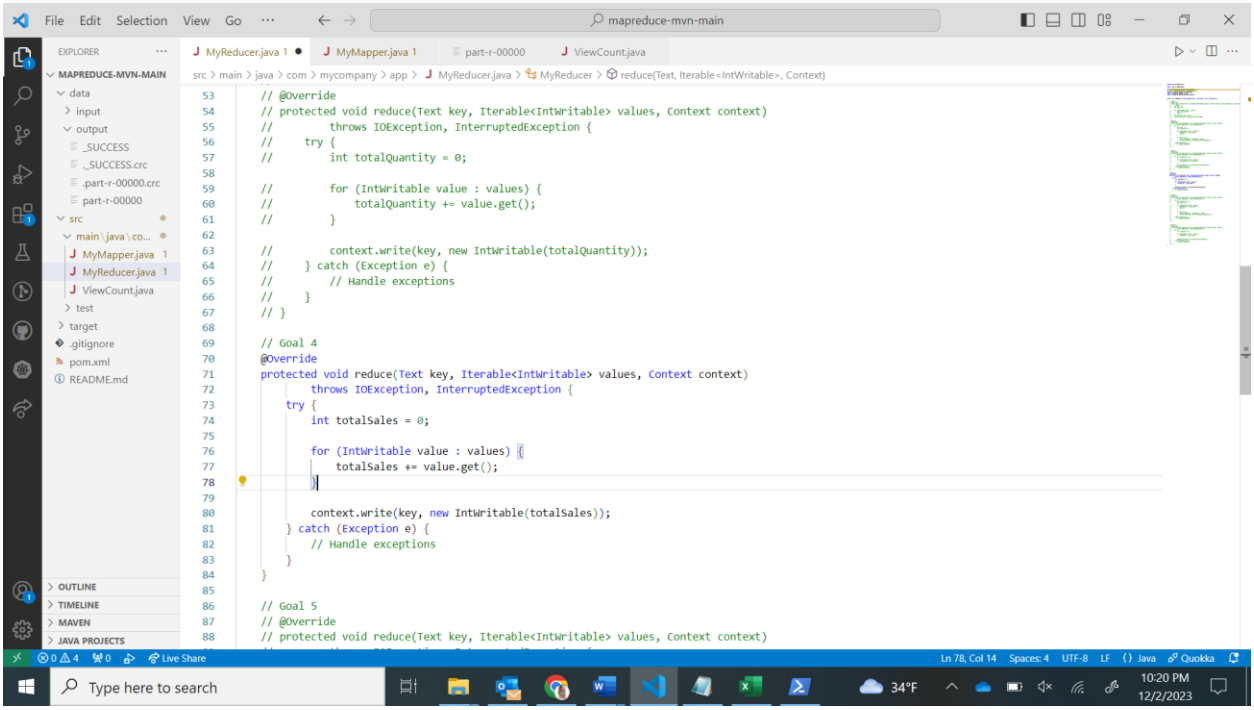

File Edit Selection View Go
mapreduce-mvn-main
EXPLORER
MAPREDUCE-MVN-MAIN
MyReducer.java 1
MyMapper.java 1
part-r-00000
ViewCount.java
src > main > java > com > mycompany > app > MyReducer.java > MyReducer > reduce(Text, Iterable<IntWritable>, Context)
// Goal 3
@Override
protected void reduce(Text key, Iterable<IntWritable> values, Context context)
throws IOException, InterruptedException {
try {
int totalQuantity = 0;
for (IntWritable value : values) {
totalQuantity += value.get();
}
context.write(key, new IntWritable(totalQuantity));
} catch (Exception e) {
// Handle exceptions
}
}
// Goal 4
// @Override
// protected void reduce(Text key, Iterable<IntWritable> values, Context context)
OUTLINE
TIMELINE
MAVEN
JAVA PROJECTS
Live Share
Ln 52, Col 8 Spaces: 4 UTF-8 LF Java Quokka
Type here to search
34°F
10:17 PM
12/2/2023

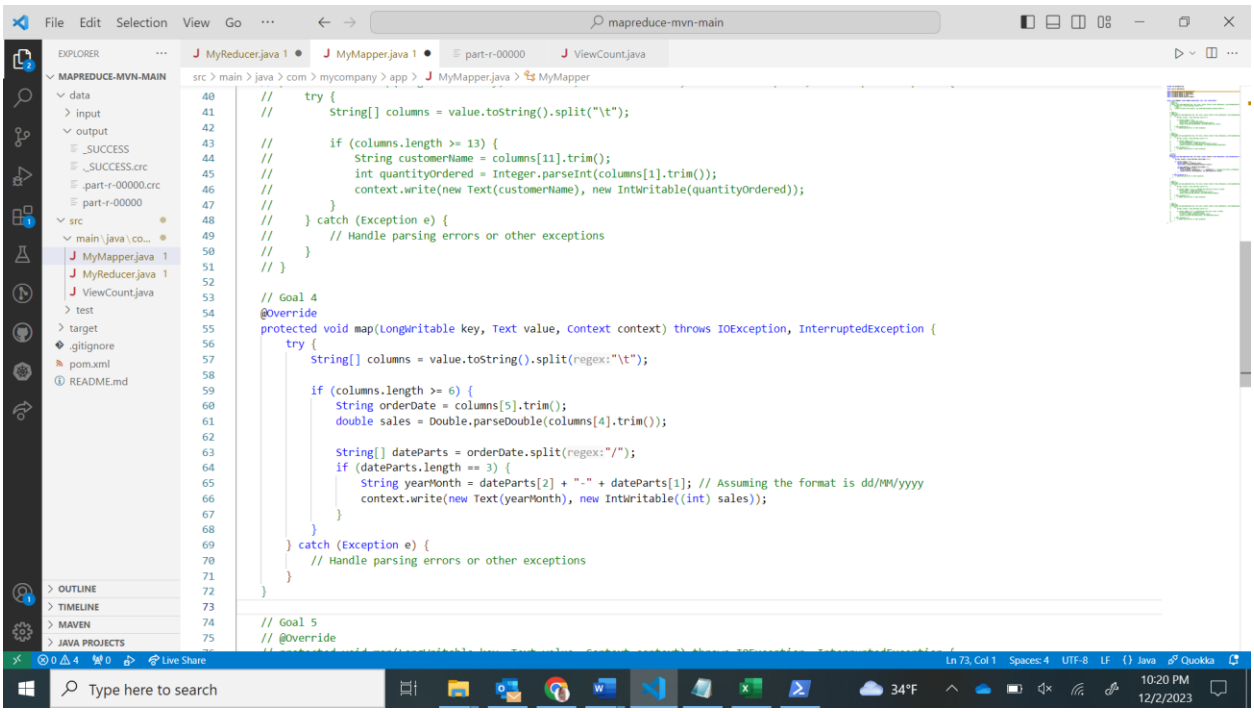

File Edit Selection View Go
mapreduce-mvn-main
EXPLORER
MyReducer.java
MyMapper.java
part-r-00000
ViewCount.java
test_data.txt
Book1234.txt
rawViews.txt
Book12345.txt
data > output > part-r-00000
"AV Stores, Co." 1778
"Anna's Decorations, Ltd" 1469
"Australian Collectables, Ltd" 705
"Australian Collectors, Co." 1926
"Australian Gift Network, Co" 545
"Bavarian Collectables Imports, Co." 401
"Blauer See Auto, Co." 811
"Classic Gift Ideas, Inc" 668
"Clover Collections, Co." 490
"Corrida Auto Replicas, Ltd" 1163
"Double Decker Gift Stores, Ltd" 357
"Dragon Souveniers, Ltd." 1524
"Iberia Gift Imports, Corp." 589
"La Corne D'abondance, Co." 836
"Norway Gifts By Mail, Co." 787
"Oulu Toy Supplies, Inc." 1110
"Royal Canadian Collectables, Ltd." 873
"Saveley & Henriot, Co." 1428
"Stylish Desk Decors, Co." 937
"Tokyo Collectables, Ltd" 1150
"Toms Spezialitten, Ltd" 936
"Toys of Finland, Co." 1051
"UK Collectables, Ltd." 1046
"Vida Sport, Ltd" 1078
"Volvo Model Replicas, Co" 647
Alpha Cognac 687
Amica Models & Co. 843
Atelier graphique 270
Auto Assoc. & Cie. 637
Auto Canal Petit 1001
Auto-Moto Classics Inc. 287
Baane Mini Imports 1082
Boards & Toys Co. 102
CAF Imports 468
Cambridge Collectables Co. 357
Canadian Gift Exchange Network 703
OUTLINE
TIMELINE
MAVEN
JAVA PROJECTS
Live Share
Ln 4, Col 1 Spaces: 4 UTF-8 LF Markdown Quokka
Type here to search
34°F
9:22 PM
12/2/2023

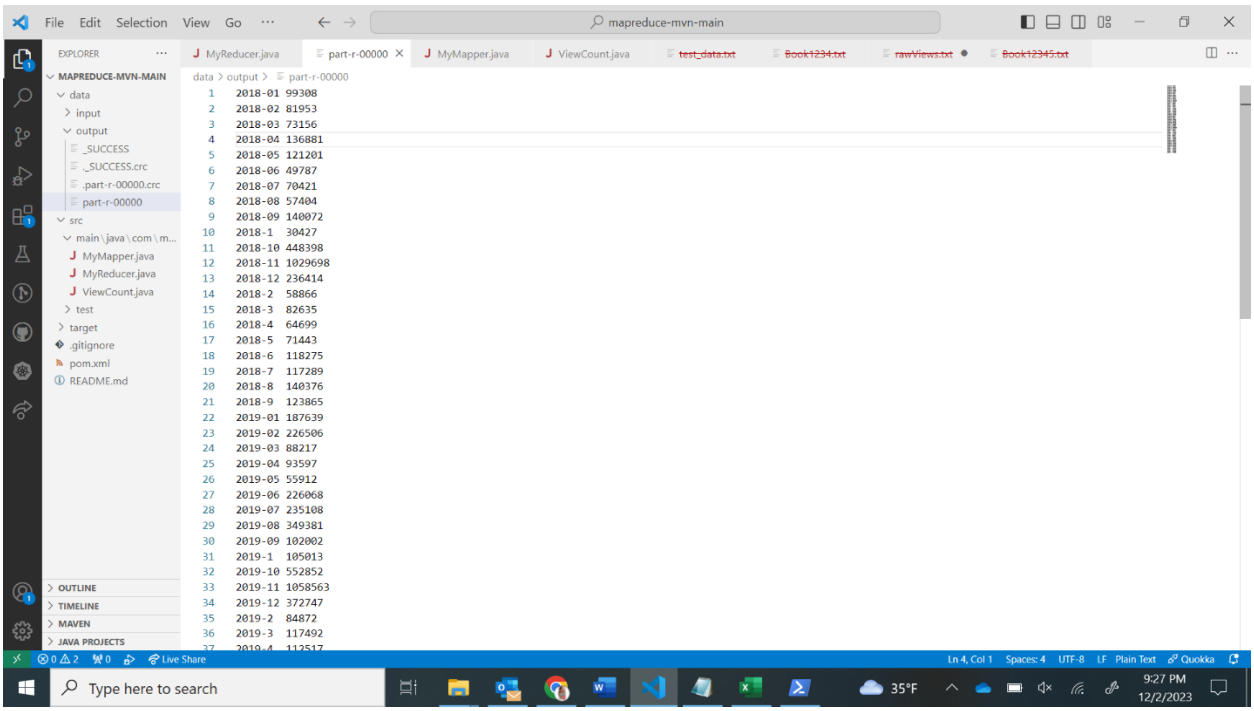


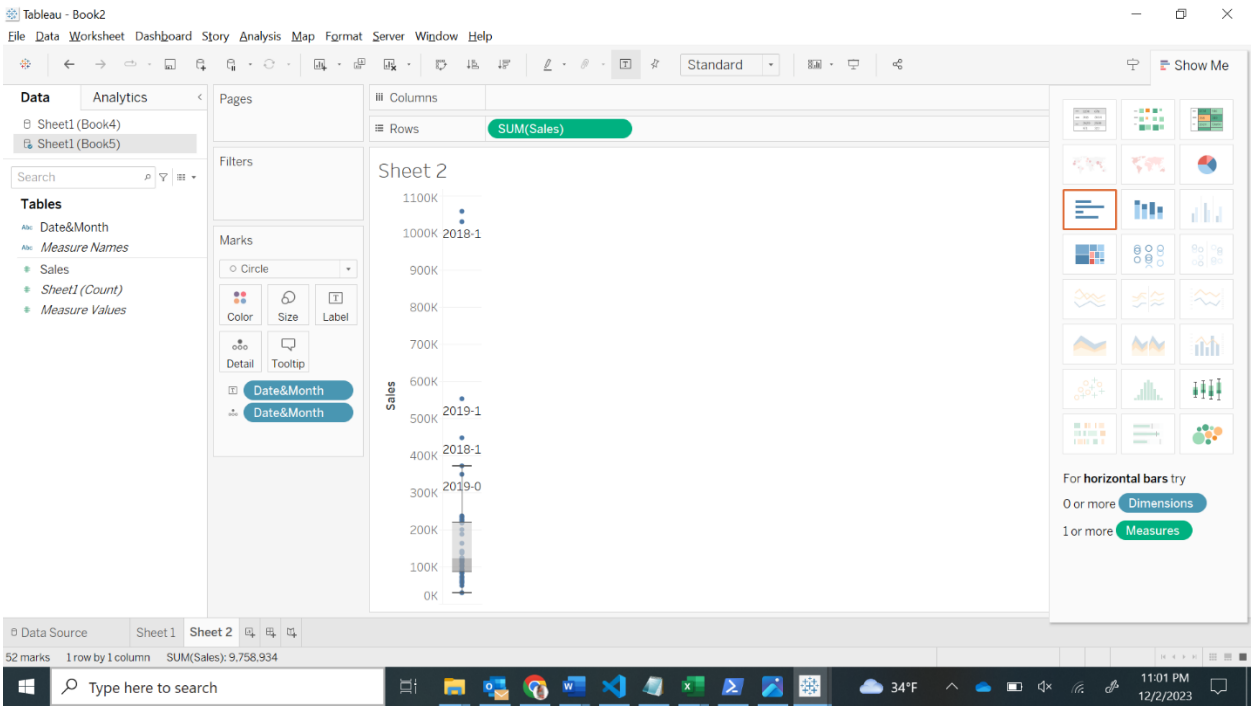

**Story:** Discover the most valued consumers by recognizing individuals who have purchased the most products. Analyze consumer transaction data to identify top buyers, allowing for more personalized engagement and targeted marketing initiatives. By recognizing and prioritizing high-impact clients based on their purchase history, this effort attempts to improve customer connections, drive loyalty, and optimize business outcomes.

**Goal4:** To Examine the trend in sales over different months and years.



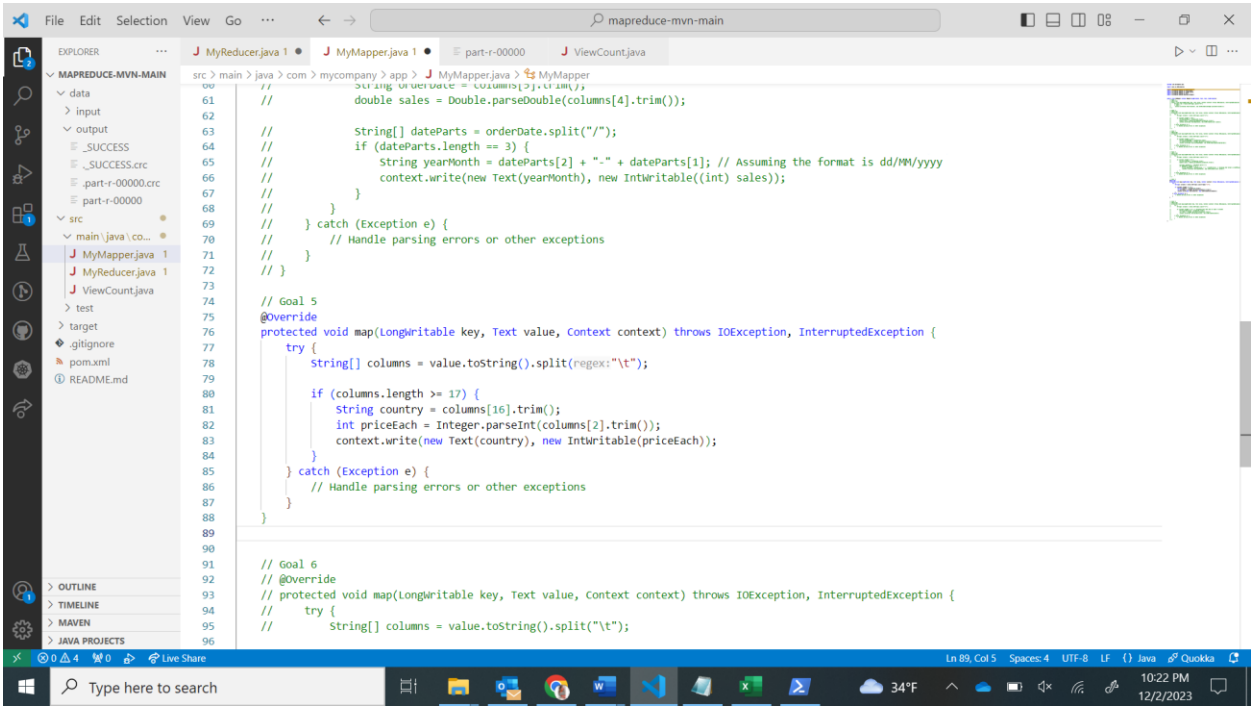





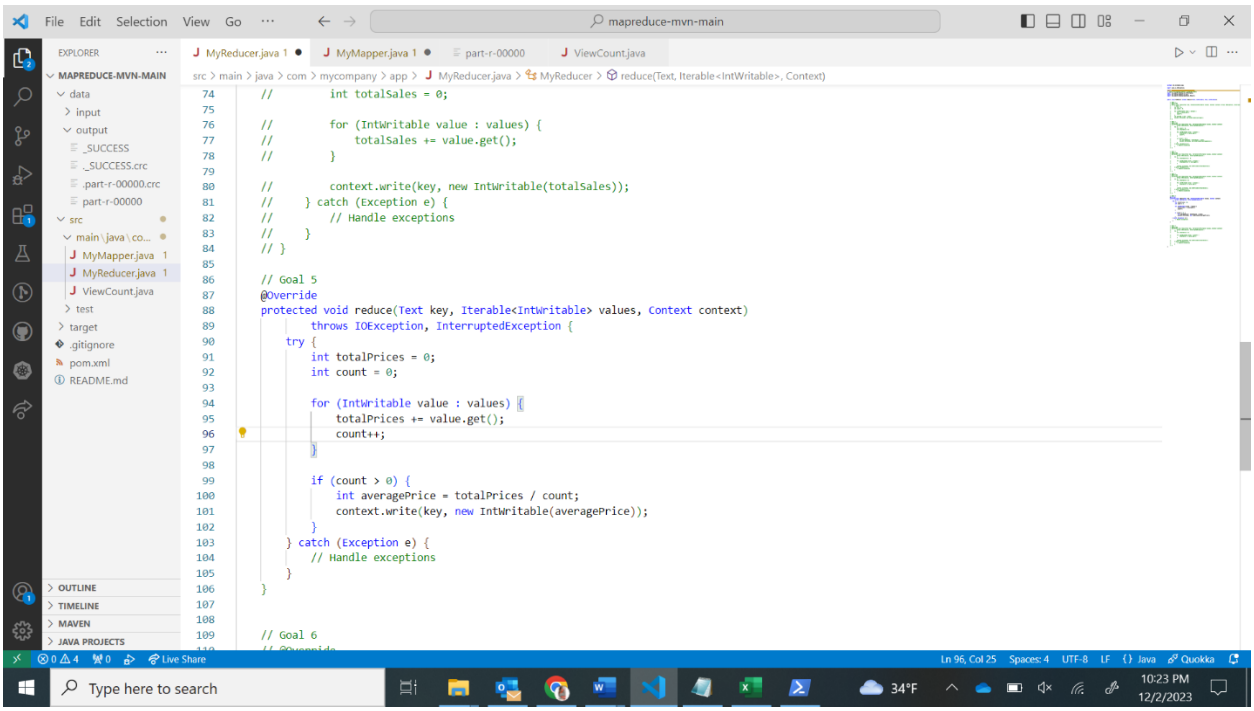




**Story:** Examining the dataset to track and analyze sales trends over time. Investigate trends and changes in sales data to get insights into seasonal variances and long-term performance. This investigation seeks to comprehend the dynamics of sales over time in order to facilitate educated decision-making and strategic planning for future business ventures.

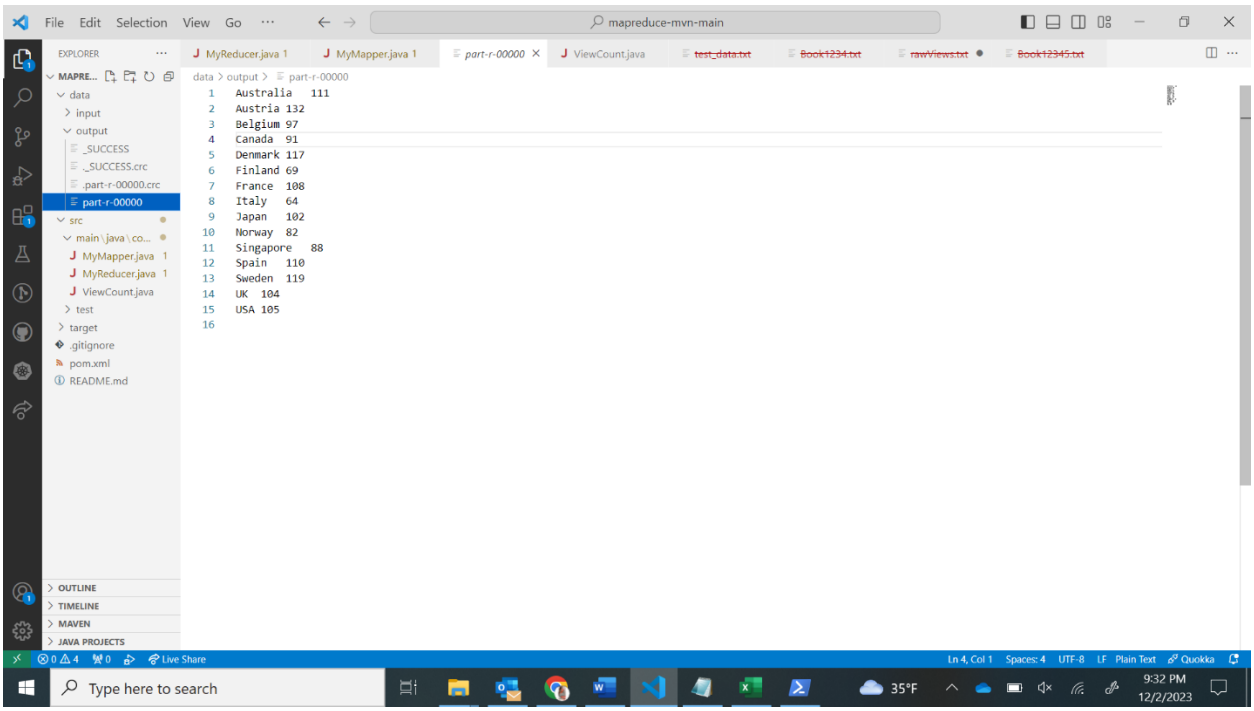
**Goal 5:** To Analyze the average price of products sold in different countries.

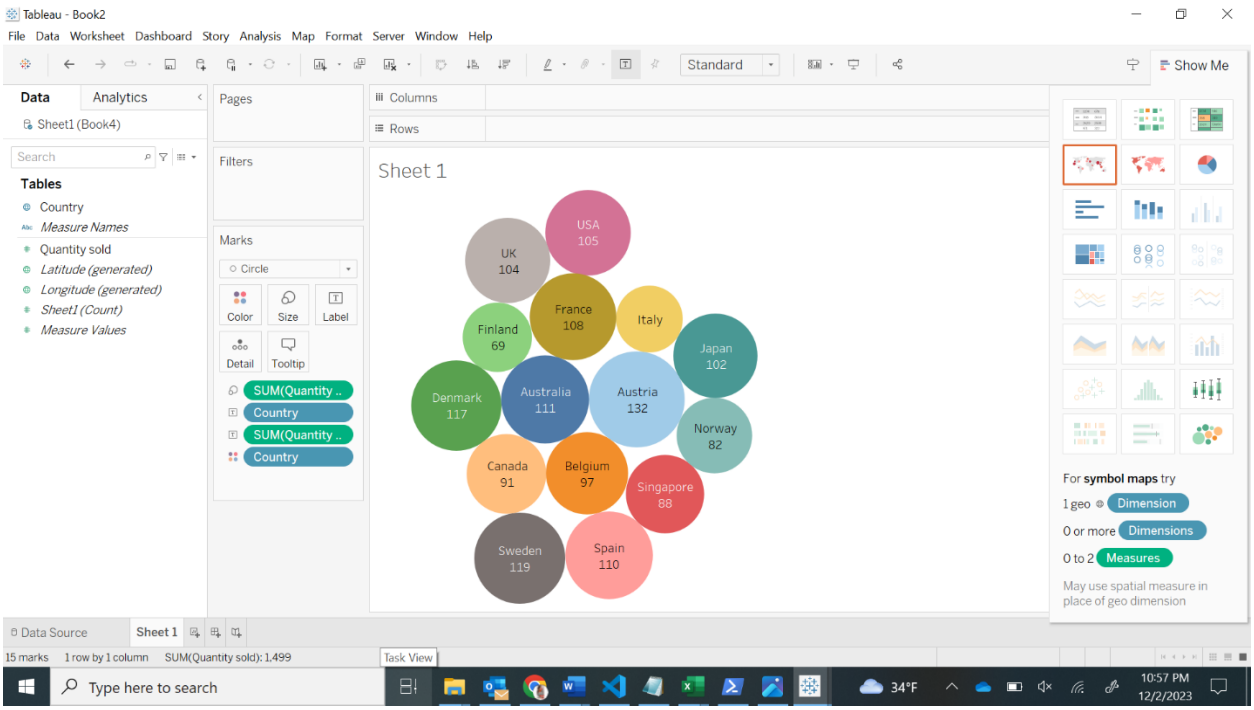


```
// Goal 5
@Override
protected void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
    try {
        String[] columns = value.toString().split("\t");

        if (columns.length >= 17) {
            String country = columns[16].trim();
            int priceEach = Integer.parseInt(columns[2].trim());
            context.write(new Text(country), new IntWritable(priceEach));
        }
    } catch (Exception e) {
        // Handle parsing errors or other exceptions
    }
}
```



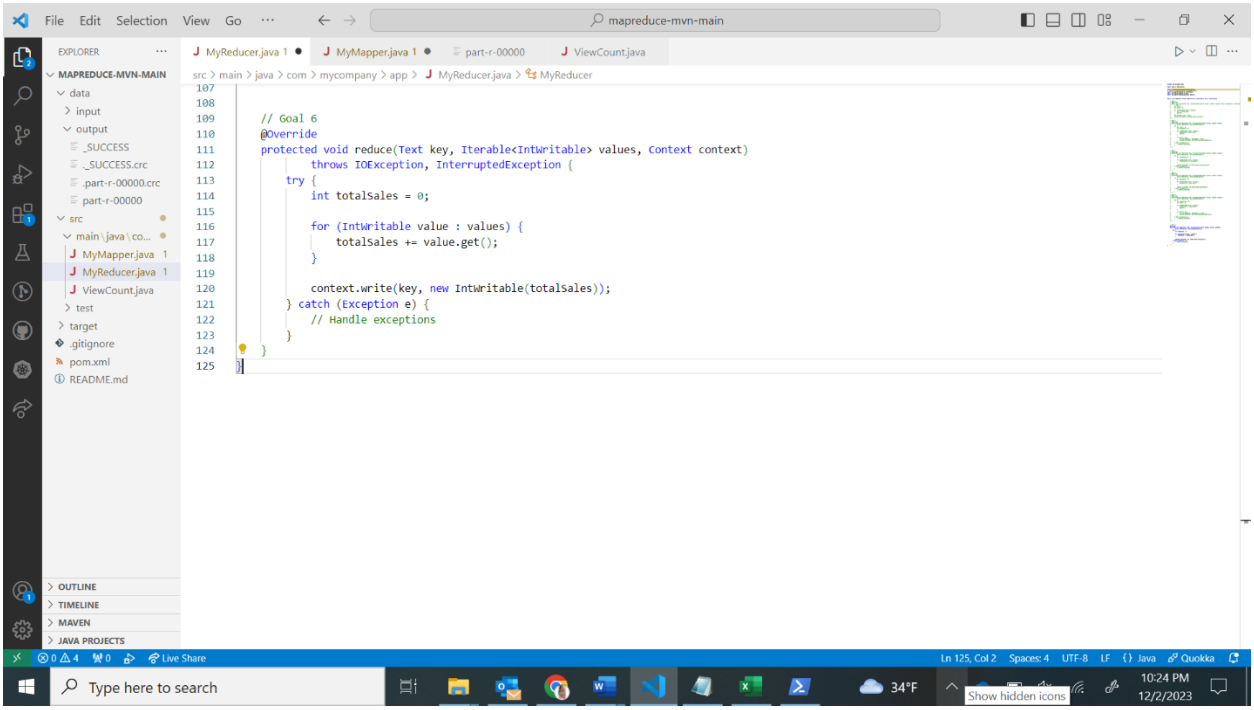
```
// Goal 5
@Override
protected void reduce(Text key, Iterable<IntWritable> values, Context context)
        throws IOException, InterruptedException {
    try {
        int totalPrices = 0;
        int count = 0;

        for (IntWritable value : values) {
            totalPrices += value.get();
            count++;
        }

        if (count > 0) {
            int averagePrice = totalPrices / count;
            context.write(key, new IntWritable(averagePrice));
        }
    } catch (Exception e) {
        // Handle exceptions
    }
}
```

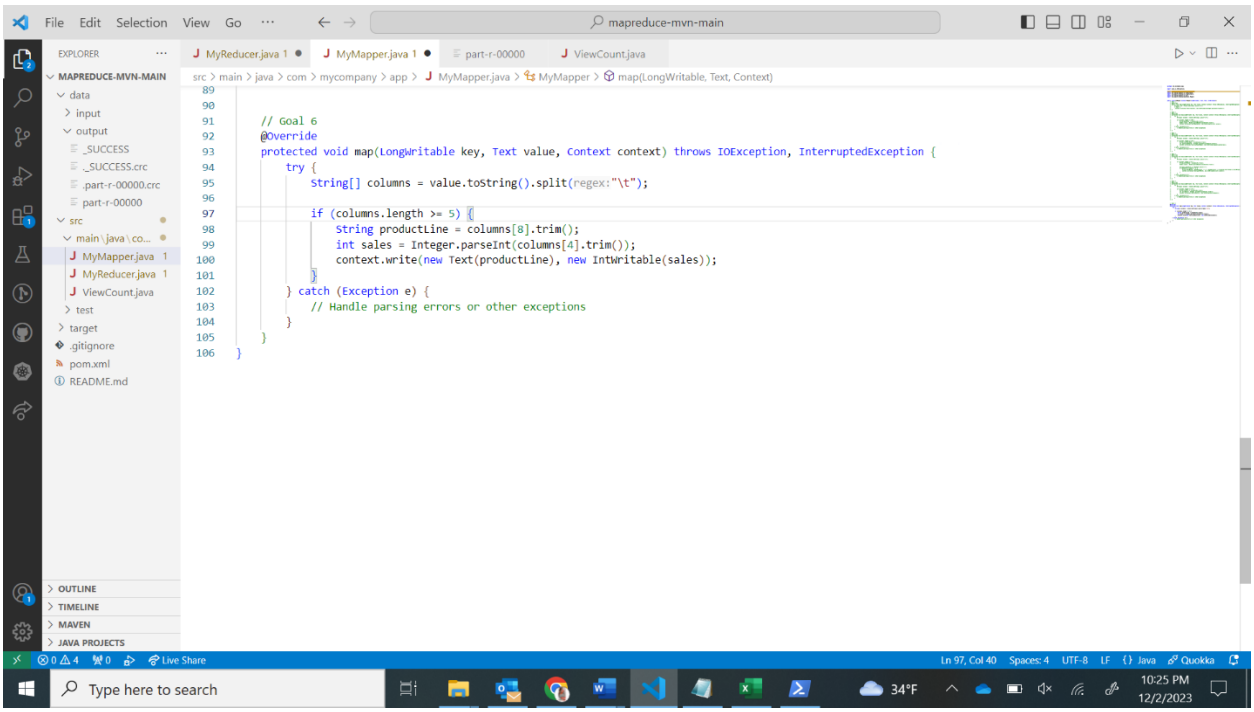


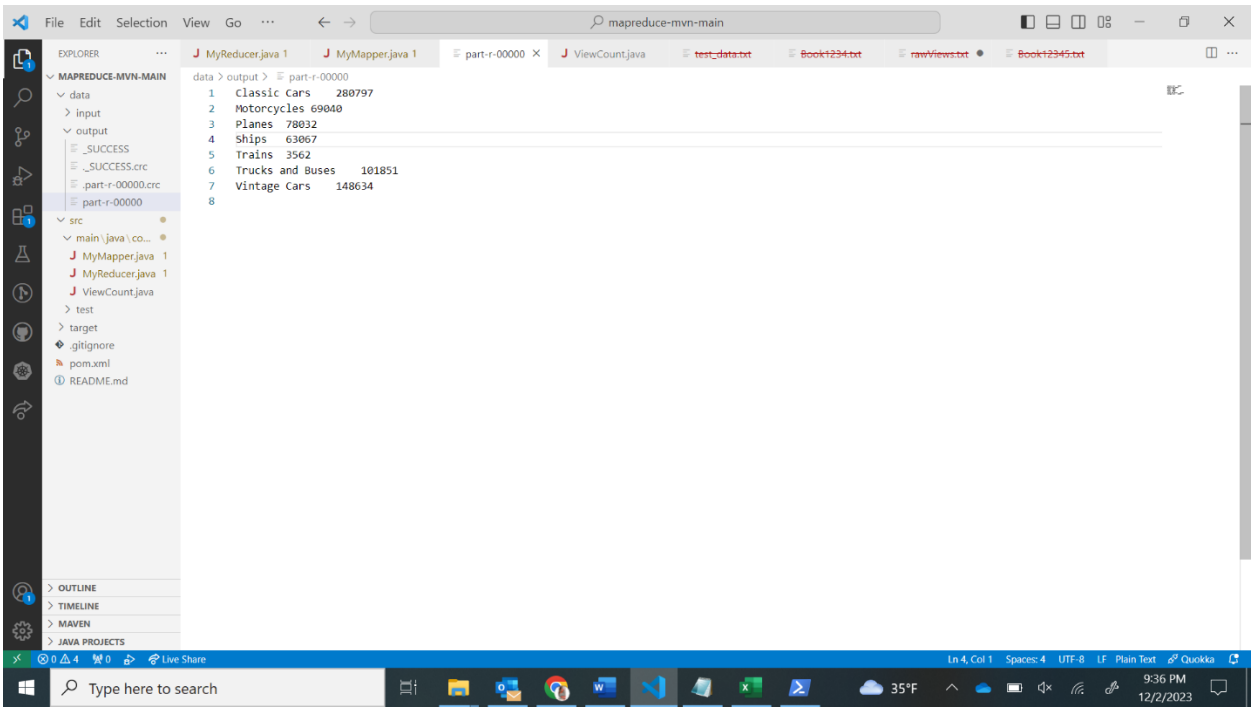



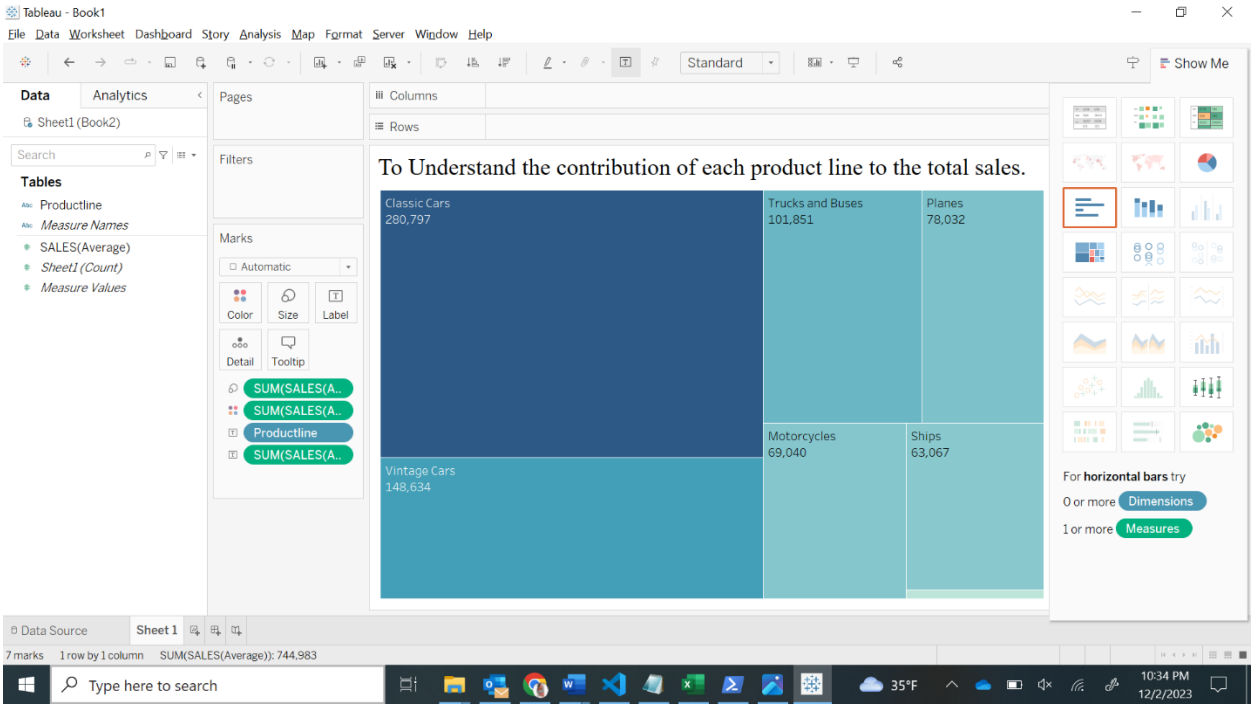

**Story:** Investigate and evaluate the average pricing of things sold, noting differences between countries. The purpose is to get insights into pricing dynamics, geographical preferences, and market trends by evaluating this data. This study aids in informed decision-making by enabling for smart pricing adjustments and targeted marketing tactics adapted to the economic landscape of each country.

**Goal 6:** To Understand the contribution of each product line to the total sales.






**Story:** Investigate the statistics to learn how each product line contributes to overall sales. Analyze sales data to determine the relative impact of different product lines on total revenue. This investigation intends to provide insights into the performance of various product categories, allowing strategic decisions for optimizing sales tactics and resource allocation. Ultimately, the

goal is to comprehend and improve each product line's contribution to the overall performance of the company.

## Conclusion:

The analysis of several goals generated practical information. Understanding customer behavior, sales dispersion, and top customers improves targeted strategies. Examining sales trends over time and examining pricing dynamics by area provide significant market knowledge. Finally, understanding each product line's contribution improves inventory and marketing efforts, enabling educated decision-making for long-term business success.

**GitHub:** https://github.com/Achyuthareddy-gantla/ByteBrilliance

**Citations:**


1. https://www.factorywarrantylist.com/car-sales-by-country.html
2. https://www.statista.com/topics/1487/automotive-industry/#topicOverview
3. https://www.oica.net/category/sales-statistics/