

# **ACADEMIC TASK CA-2**

**Project Title:**

**Analyzing Sales Performance Using Spark SQL**

**Course Name:**

**Cluster Computing (INT315)**

**Submitted by: Submitted to:**

Name: Md Imlak Name: Dr. Saqib Ul Sabha

Roll No: 29 UID: 32521

Reg: 12104074 Assistant Professor

Section: K21AK LPU

**Date of Submission: 21/04/2025**

**School of Computer Science Engineering**

**Lovely Professional University**

**Punjab**

**DECLEARATION**

I am , Md Imlak a student of Bachelor of Technology under CSE discipline at Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own work and is genuine.

Your name : Md Imlak

Registration no : 12104074

**Project: Analyzing Sales Performance Using Spark SQL**

Context: A retail company wants to analyze its sales records to understand performance trends across products, regions, and time.

Tasks:

1. Load the sales dataset into Spark and convert it to a DataFrame.
2. Register the DataFrame as a temporary SQL table.
3. Write a SQL query to find total revenue generated by each product category.
4. Calculate monthly sales totals across different regions.
5. Identify the top 3 best-selling products based on quantity sold.
6. Visualize monthly sales performance using a line chart.
7. Calculate the average price of products in each category (using a subquery).
8. Find the region with the highest total revenue (using a subquery).
9. Visualize the total revenue by product category using a bar chart.
10. Find products with revenue greater than the average revenue.

Table Contents:

|  |  |  |
| --- | --- | --- |
| **Section** | **SubSection** | Pages |
| 1. Obejctive | Brief description of the project's aim | 4 |
| 1. Tools and Technologies used | Apache Spark, Scala, Dataset, Environment | 4 |
| 1. Dataset Description | Dataset Type, Rows/Columns, Key Column Explanation | 4 |
| 1. Implementation Steps | Task 1-10 (Load Data, SQL Queries, Visualizations) | 4-11 |
| 1. Final Visualizations | Screenshot of Charts, Visualization Explanations | 12-14 |
| 1. Challenges Faced | Errors and Challenges, Resolution | 14 |
| 1. Conclusion | Summary of Learning and Results | 14 |
| 1. References | List of Online Resources and Documentation | 14 |

1. **Objective:**

The objective of this project is to leverage Apache Spark SQL to analyze retail sales data, focusing on key performance indicators such as product category revenue, regional sales trends, monthly sales variations, average product prices, and individual product performance relative to average revenue. This analysis aims to provide actionable insights into sales performance.

1. **Tools and Technologies used:**

 Apache Spark (version 3.5.4)

 Scala (version 2.12.18)

 Dataset used: project\_sales\_data.csv

 Environment: Windows PowerShell

 Python (3.11.5) with Pandas and Matplotlib libraries.

1. **Dataset Description:**

 Dataset type: CSV (Comma Separated Values)

 Number of rows: 161

 Number of columns: 8

* **Explanation of key columns:**

 OrderID: Unique identifier for each order.

 Product: Name of the product sold.

 Category: Category of the product (e.g., Electronics, Clothing).

 Region: Geographical region of the sale (e.g., North, South, East, West).

 Quantity: Number of units sold.

 Price: Price per unit.

 OrderDate: Date of the order (DD-MM-YYYY).

 Revenue: Total revenue generated from the order (Quantity \* Price).

1. **Implementation Steps:**

**Task 1:** **Load the sales dataset into Spark and convert it to a DataFrame.**

**Explanation:** This code initializes a SparkSession and loads the CSV file into a DataFrame. printSchema() displays the DataFrame's schema, and show() displays sample data.

**Code:**

import org.apache.spark.sql.SparkSession

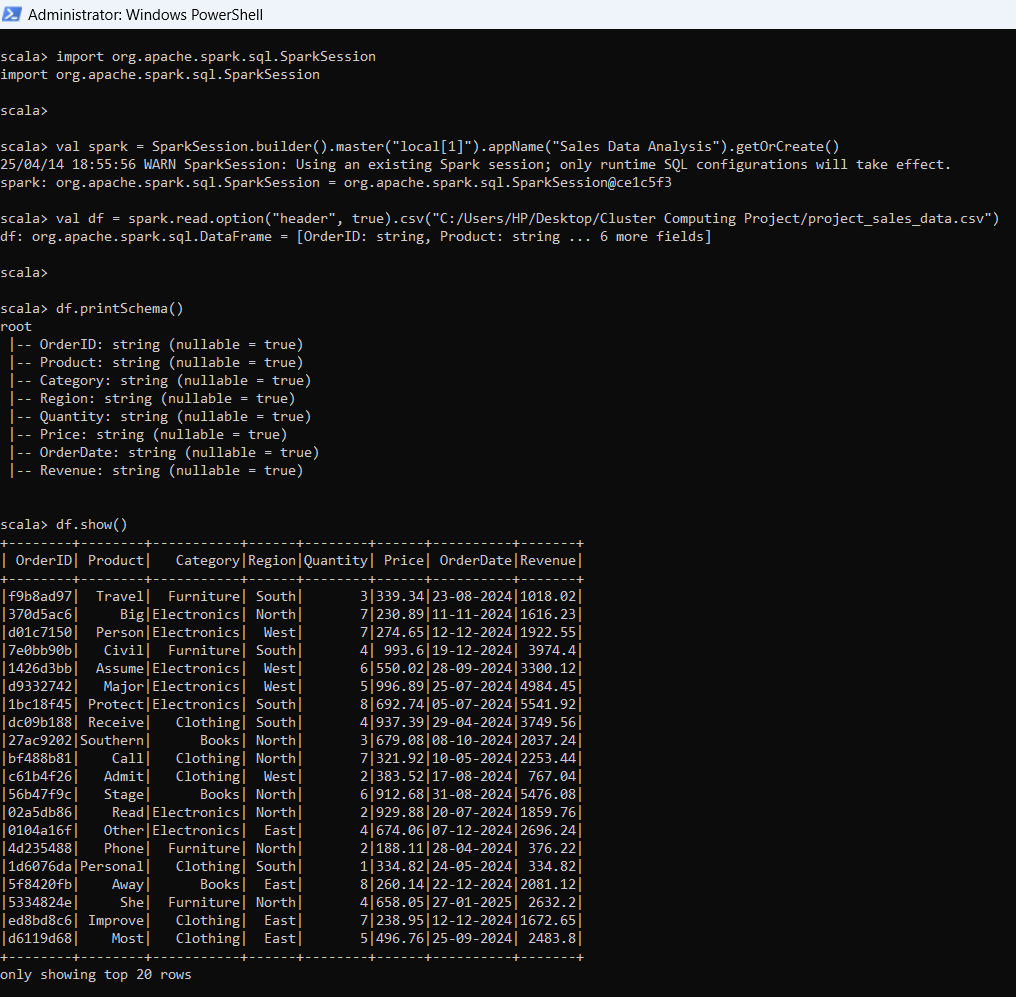
val spark = SparkSession.builder().master("local[1]").appName("Sales Data Analysis").getOrCreate()

val df = spark.read.option("header", true).csv("C:/Users/HP/Desktop/Cluster Computing Project/project\_sales\_data.csv")

df.printSchema()

df.show()

**Screenshot output:**



**Output:** The schema and sample data were displayed.

**Task 2:** **Register the DataFrame as a temporary SQL table**.

**Explanation:** The DataFrame was registered as a temporary SQL table named "sales" for SQL querying.

**Code:**

df.createOrReplaceTempView("sales")

**Output:** No visible output, but the table "sales" was created.

**Task 3:** **Write a SQL query to find total revenue generated by each product category.**

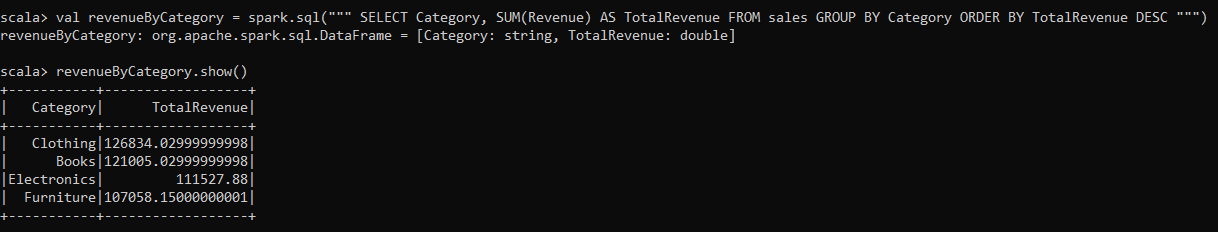
**Explanation:** A SQL query was used to calculate the sum of revenue per category.

**Code:**

val revenueByCategory = spark.sql(""" SELECT Category, SUM(Revenue) AS TotalRevenue FROM sales GROUP BY Category ORDER BY TotalRevenue DESC """)

revenueByCategory.show()

**Screenshot output:**

****

**Output: Displayed total revenue by category.**

**Task 4:** **Calculate monthly sales totals across different regions.**

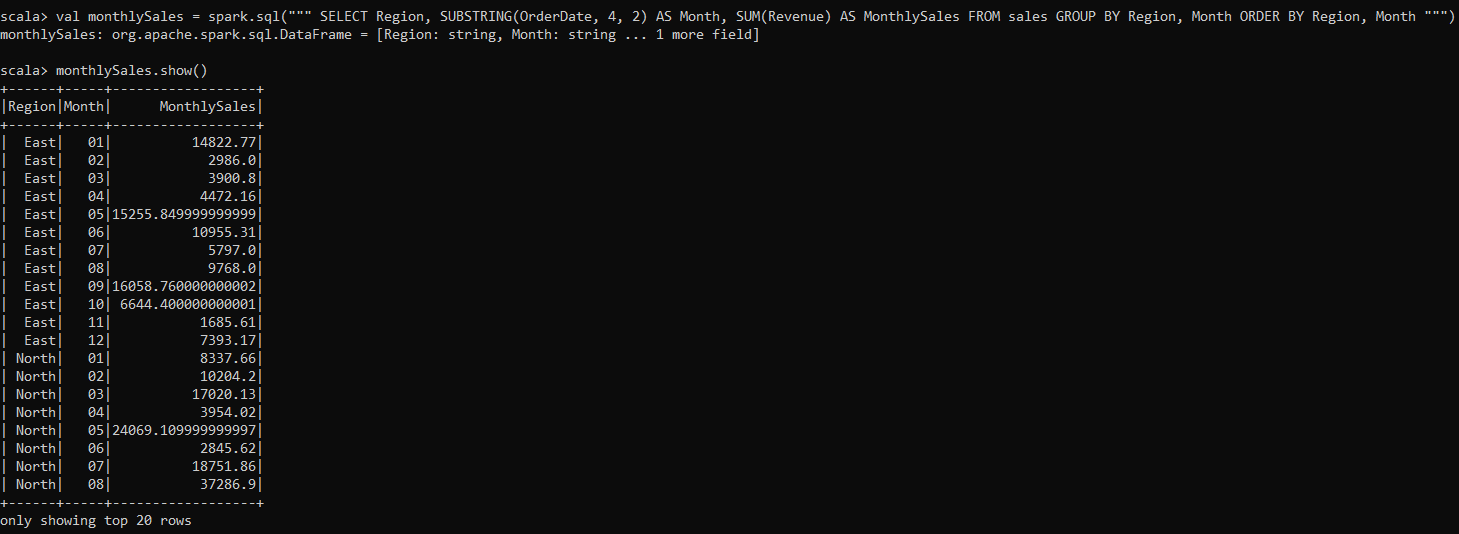
**Explanation:** A SQL query was used to calculate monthly sales for each region.

**Code:**

val monthlySales = spark.sql(""" SELECT Region, SUBSTRING(OrderDate, 4, 2) AS Month, SUM(Revenue) AS MonthlySales FROM sales GROUP BY Region, Month ORDER BY Region, Month """)

monthlySales.show()

**Screenshot output:**



**Output:** Displayed monthly sales by region.

**Task 5: Identify the top 3 best-selling products based on quantity sold.**

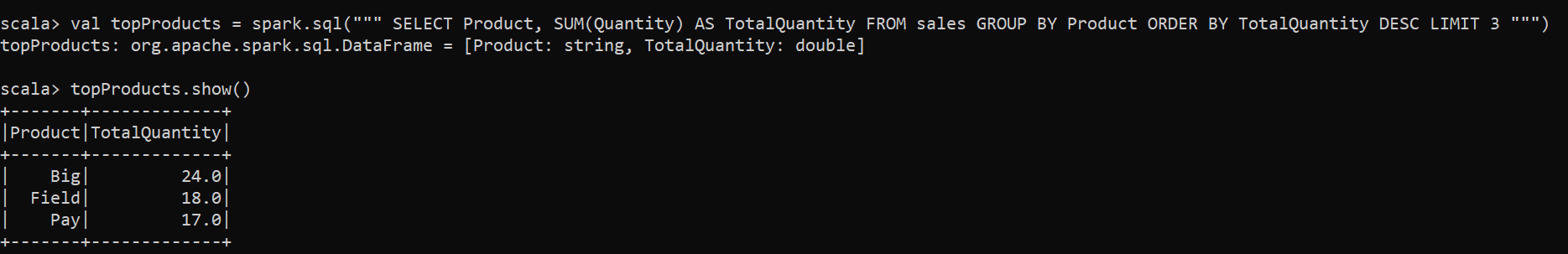
**Explanation:** A SQL query was used to find the top 3 products with the highest quantity sold.

**Code:**

val topProducts = spark.sql(""" SELECT Product, SUM(Quantity) AS TotalQuantity FROM sales GROUP BY Product ORDER BY TotalQuantity DESC LIMIT 3 """)

topProducts.show()

**Snapshot output:**



**Output:** Displayed the top 3 products.

**Task 6:** **Visualize monthly sales performance using a line chart.**

**Explanation:** Monthly sales data was visualized using a line chart in Python.

**Code (Scala and Python):**

* + **Scala (Saving to CSV):**

monthlySales.coalesce(1).write.option("header", "true").csv("C:/Users/HP/Desktop/Cluster Computing Project/monthly\_sales.csv")

**Python Code for Visualization:**

import pandas as pd

import glob

import matplotlib.pyplot as plt

file\_list = glob.glob("C:/Users/HP/Desktop/Cluster Computing Project/monthly\_sales.csv/part-\*.csv")

if file\_list:

df = pd.read\_csv(file\_list[0])

monthly\_totals = df.groupby("Month")["MonthlySales"].sum().reset\_index()

plt.figure(figsize=(10, 6))

plt.plot(monthly\_totals["Month"], monthly\_totals["MonthlySales"], marker="o")

plt.title("Total Monthly Sales Performance")

plt.xlabel("Month")

plt.ylabel("Total Monthly Sales")

plt.grid(True)

plt.show()

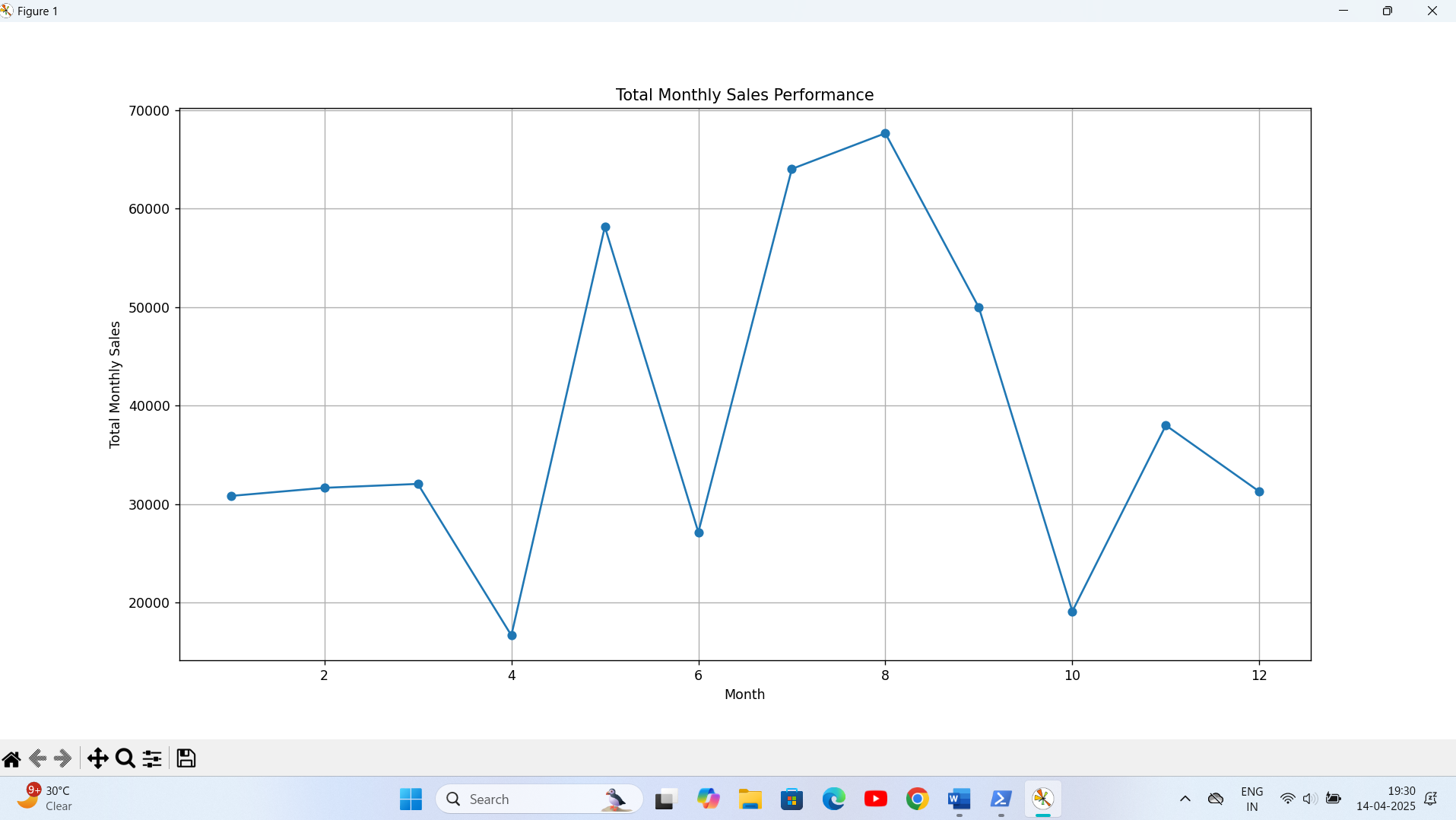
else:

print("No part files found!")

**Explanation:** This code saves the monthly sales data to a CSV and then uses Python's matplotlib to create a line chart.

**Output:** A line chart showing monthly sales Performance.

**Screenshot output:**



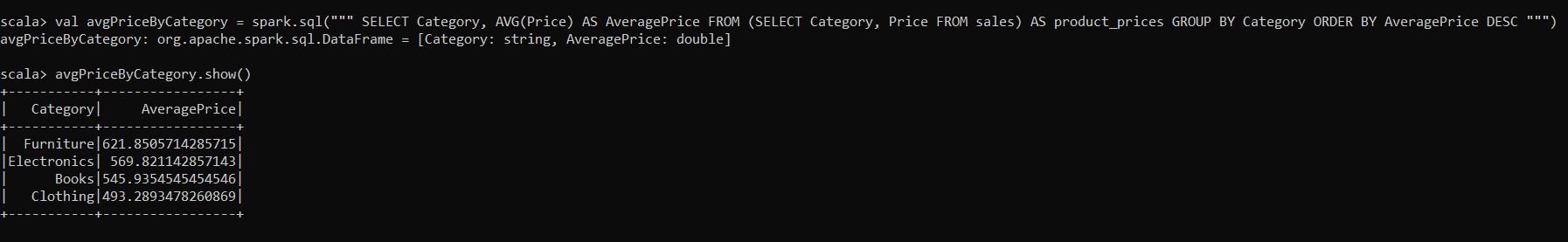
**Task 7:** **Calculate the average price of products in each category (using a subquery).**

**Explanation:** A subquery was used to select category and price, and then the average price per category was calculated.

**Code:**

val avgPriceByCategory = spark.sql(""" SELECT Category, AVG(Price) AS AveragePrice FROM (SELECT Category, Price FROM sales) AS product\_prices GROUP BY Category ORDER BY AveragePrice DESC """)

avgPriceByCategory.show()

**Snapshot output:** ****

**Output: Average price by category.**

**Task 8: Find the region with the highest total revenue (using a subquery).**

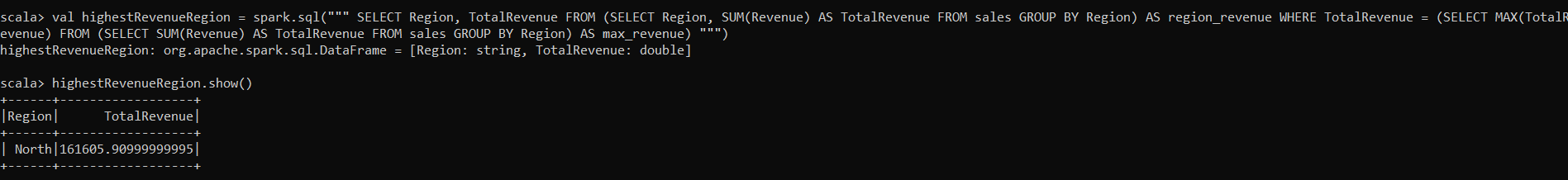
**Explanation:** Subqueries were used to find the maximum revenue region.

**Code:**

val highestRevenueRegion = spark.sql(""" SELECT Region, TotalRevenue FROM (SELECT Region, SUM(Revenue) AS TotalRevenue FROM sales GROUP BY Region) AS region\_revenue WHERE TotalRevenue = (SELECT MAX(TotalRevenue) FROM (SELECT SUM(Revenue) AS TotalRevenue FROM sales GROUP BY Region) AS max\_revenue) """)

highestRevenueRegion.show()

**Snapshot output:**

****

**Output:** Region with highest revenue.

**Task 9:** **Visualize the total revenue by product category using a bar chart.**

**Explanation:** Revenue by category was visualized using a bar chart in Python.

**Code (Scala and Python):**

* + **Scala (Saving to CSV):**

revenueByCategory.coalesce(1).write.option("header", "true").csv("C:/Users/HP/Desktop/Cluster Computing Project/revenue\_by\_category.csv")

**Python Code for Visualization:**

import pandas as pd

import glob

import matplotlib.pyplot as plt

file\_list = glob.glob("C:/Users/HP/Desktop/Cluster Computing Project/revenue\_by\_category.csv/part-\*.csv")

if file\_list:

df = pd.read\_csv(file\_list[0])

plt.figure(figsize=(10, 6))

plt.bar(df["Category"], df["TotalRevenue"])

plt.title("Total Revenue by Product Category")

plt.xlabel("Category")

plt.ylabel("Total Revenue")

plt.xticks(rotation=45)

plt.tight\_layout()

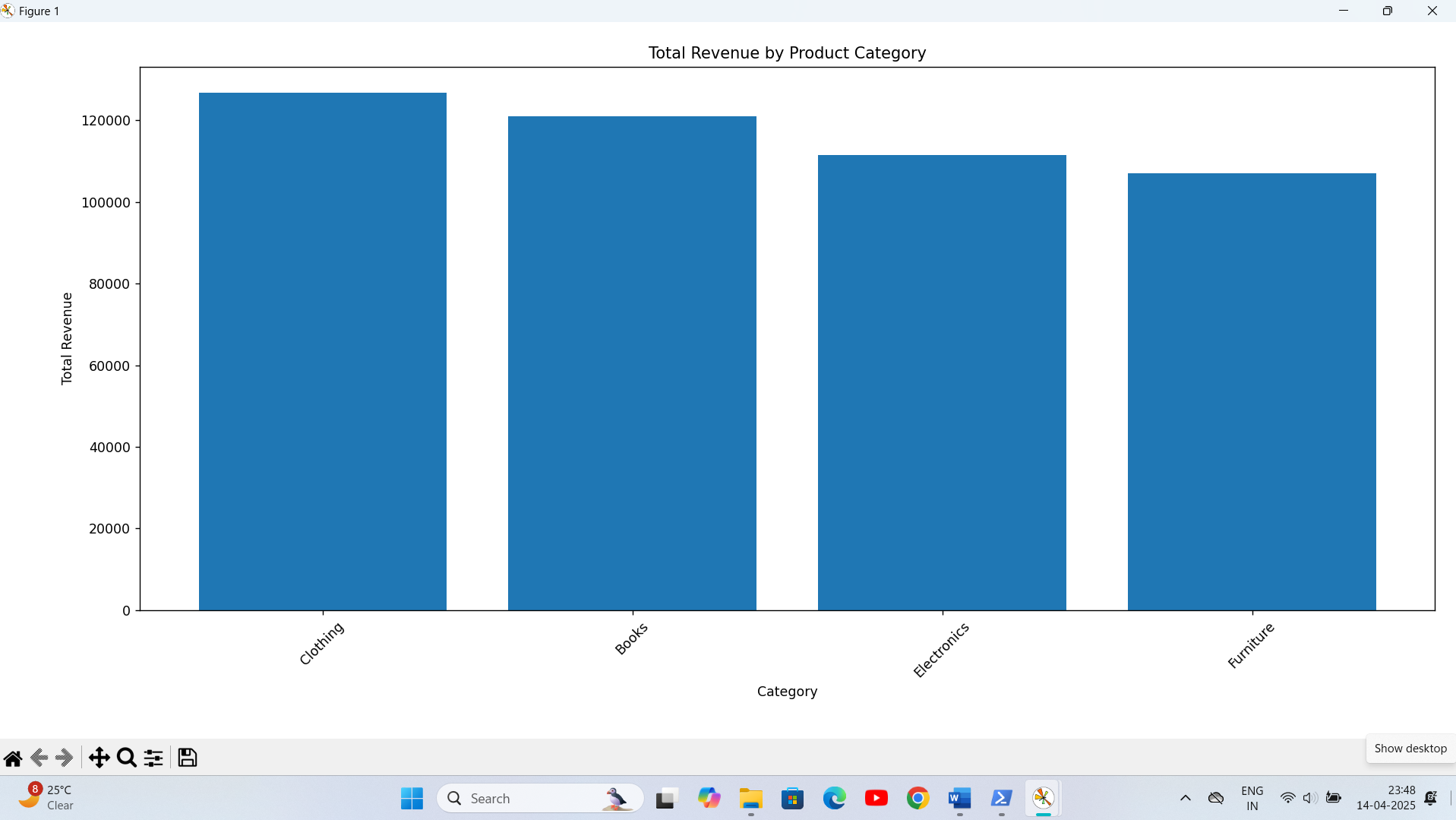
plt.show()

else:

print("No part files found!")

**Explanation:** This code generates a bar chart comparing total revenue across product categories.

**Output:** A bar chart showing revenue by category.

**Snapshot :** ****

**Task 10: Find products with revenue greater than the average revenue.**

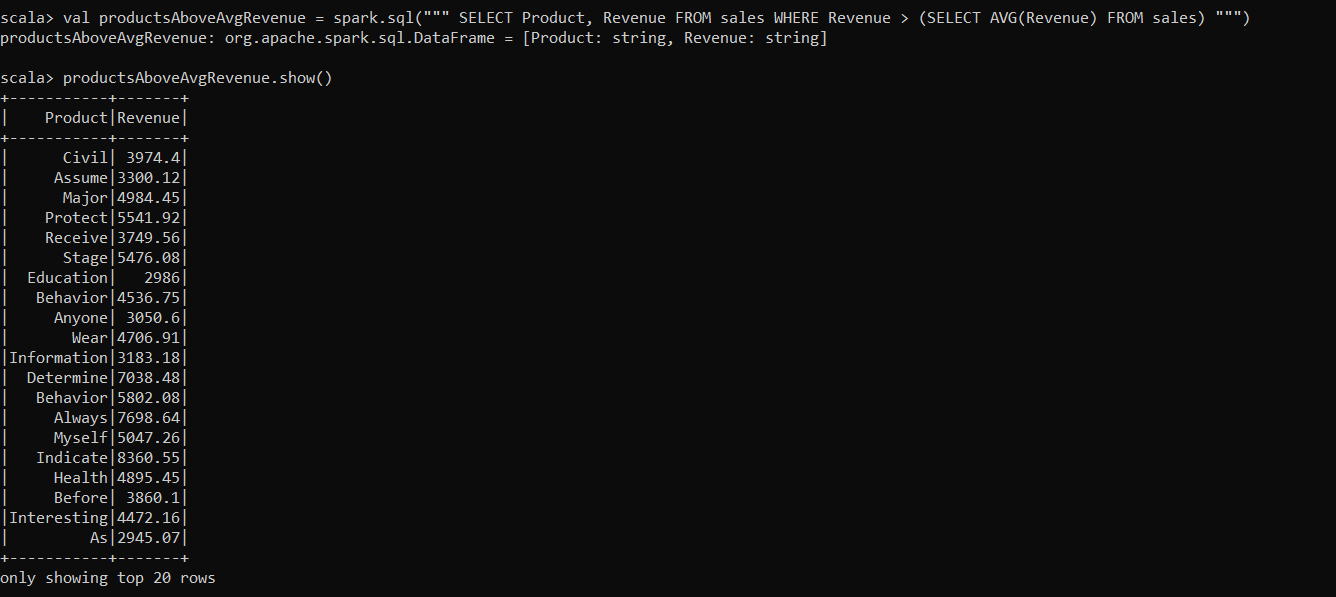
**Explanation:** A subquery was used to find the average revenue, and then products with revenue greater than that were displayed.

**Code:**

val productsAboveAvgRevenue = spark.sql(""" SELECT Product, Revenue FROM sales WHERE Revenue > (SELECT AVG(Revenue) FROM sales) """)

productsAboveAvgRevenue.show()

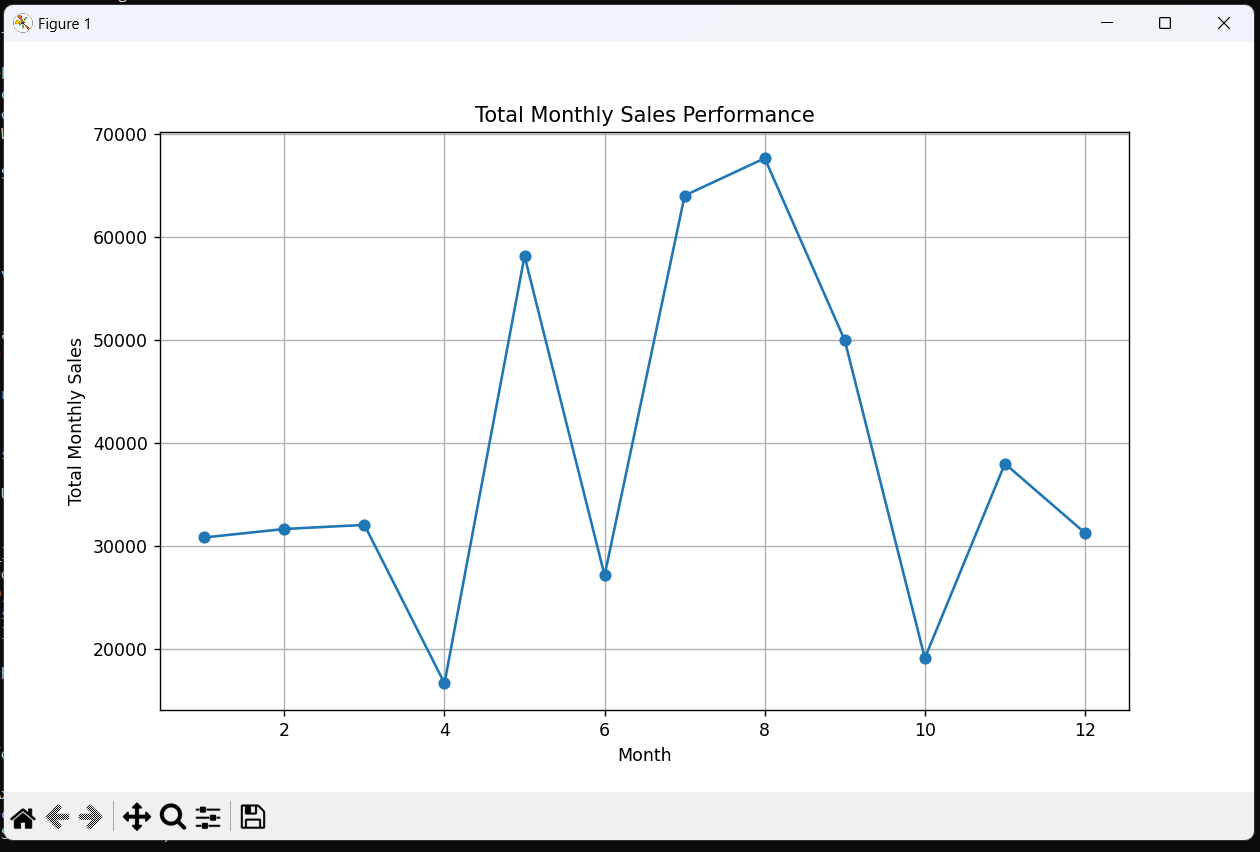
**Screenshot output:**

****

**Output:** Products with revenue above average.

1. **Final Visualizations:**

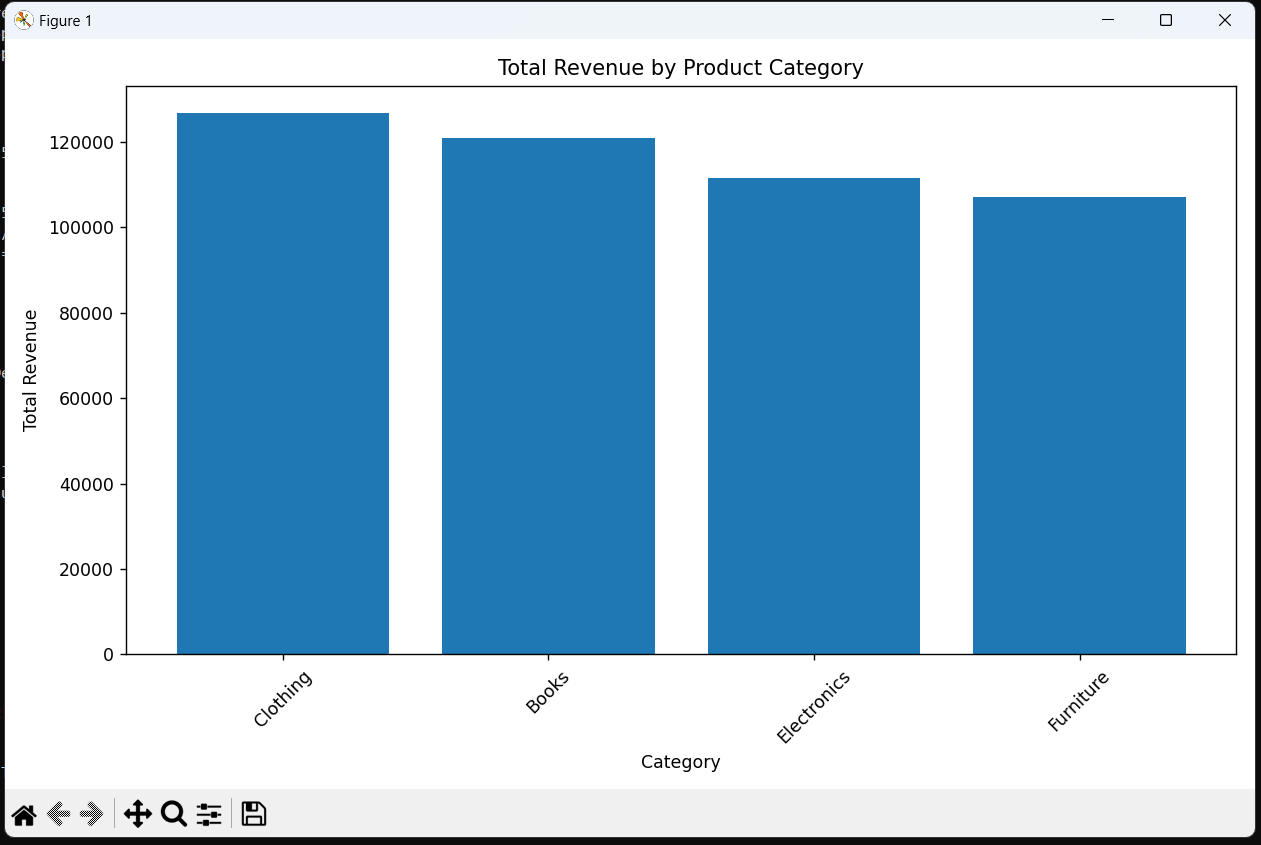
Screenshot: Line chart



Explanation of Visualizations:

**Line Chart: Total Monthly Sales Performance**.

* This line chart visually represents the trend of total monthly sales over the analyzed period. The x-axis displays the months, and the y-axis represents the corresponding total sales revenue.
* The chart effectively illustrates the fluctuations in sales performance across different months. Peaks and troughs in the line indicate periods of high and low sales, respectively. This visualization helps to identify seasonal trends, monthly variations, and overall sales patterns, which are crucial for strategic business decisions.
* For example, from the provided data, we can see the sales for month 5 is very high. This type of information is very valuable to the business.

Screenshot: Bar chart 

Explanation of Visualizations:

**Bar Chart: Total Revenue by Product Category .**

* This bar chart provides a comparative analysis of the total revenue generated by each product category. The x-axis represents the product categories (e.g., Clothing, Books, Electronics, Furniture), and the y-axis represents the total revenue for each category.
* The height of each bar corresponds to the total revenue earned from that particular category, allowing for a quick and easy comparison of the revenue performance of different product lines. This visualization helps to identify the most and least profitable product categories, enabling the company to focus on high-performing categories and address issues in underperforming ones.
* For example, we can see that the clothing category, and books category, creates the most revenue.

**Key Takeaways from the Visualizations:**

* **Trend Analysis:** The line chart helps in understanding the temporal patterns of sales, which can be correlated with marketing campaigns, seasonal changes, or economic factors.
* **Comparative Analysis:** The bar chart provides a direct comparison of category-wise revenue, which is essential for product portfolio management and resource allocation.
* **Decision Support:** Both charts offer valuable insights that can support data-driven decision-making in areas such as inventory management, sales forecasting, and marketing strategies.

1. **Challenges Faced:**

* **Challenge:** Issues with Python library installations in the PySpark environment.
* **Resolution:** Ensured libraries were installed correctly and accessible in the PySpark environment.
* **Challenge:** File path issues when reading and writing CSV files.
* **Resolution:** Used absolute paths and glob module to handle part file issues.

1. **Conclusion:**

This project demonstrated the use of Apache Spark SQL for advanced sales data analysis. Subqueries allowed for complex queries, and visualizations provided clear insights. The project enhanced understanding of Spark SQL and data visualization techniques.

1. **References:**

* Apache Spark Documentation: spark.apache.org/docs/latest/
* Pandas Documentation: pandas.pydata.org/docs/
* Matplotlib Documentation: matplotlib.org/stable/contents.html

