Transaction Data Analysis Report

Executive Summary

This report presents the findings from extensive analysis of transaction data retrieved from the GCP bucket (`gs://implemention_engineer_interviews`). The analysis involved downloading, cleaning, processing, and visualizing transaction data to extract meaningful insights about transaction patterns, merchant activity, and potential anomalies.

1. Data Download & Summary

Methodology

- **Source**: GCP bucket (`gs://implemention_engineer_interviews`)
- **Authentication**: Created GCP project and authenticated using a service account key
- **Processing**: Developed a Python script using `google.cloud.storage` library

Key Features of the Download Process

- **Incremental Download**: Implemented system to download daily `transactions.csv` files incrementally
- **Resume Capability**: Created checkpoint system using `.status` files to resume interrupted downloads
- **Validation**: Implemented checks to verify if files were already downloaded to avoid duplication
- **Storage Structure**: Organized downloads in a structured `DOWNLOAD_DIR` hierarchy

Data Summary Reports

For each transaction file, automated summary reports were generated including:

- Total row count
- Column-wise data types
- Distinct vs. total value counts
- Top & bottom values (by count & lexicographical order)

2. Data Processing and Cleaning

Data Loading Optimization

- Utilized optimized `dtype_map` for memory efficiency during loading

Data Quality Management

The following data quality issues were addressed:

- **Missing Values**: Dropped rows with missing essential fields (`transaction_id`, `timestamp`, `amount`)
- **Standardization**: Applied standardization to categorical fields:
  - Lowercased text fields
  - Stripped whitespace
- **Duplicate Removal**: Eliminated duplicate transactions using `transaction_id`
- **Outlier Detection**: Applied IQR-based outlier detection methodology

Feature Engineering

New features were derived from the raw data:

- **Temporal Features**: Extracted `day_of_week` and `hour` from timestamp
- **User Behavior Metrics**: Calculated `recency_seconds` per user
- **Anomaly Detection**: Flagged anomalous transactions based on:
  - Unusual transaction values
  - Abnormally rapid transaction frequency

Storage

- Cleaned datasets were saved to a separate `CLEAN_DIR` with `_cleaned.csv` suffix
- Processing pipeline intelligently skipped re-processing of already cleaned files

3. Basic Analysis & Visualizations

Daily Transaction Analysis

- **Transaction Count**: Analysis revealed uniform daily transaction counts of approximately 1.7 million transactions per day
- This uniformity suggests the synthetic nature of the dataset

Top Merchant Analysis

- Identified and analyzed top 10 merchants by transaction count
- Key merchants include:
  - "super shop"
  - "learn school"
  - "quality solutions"
  - "exciting park"

Hourly Transaction Patterns

- Analyzed transaction distribution across 24 hours
- Identified peak transaction hours and low activity periods
- Notable peaks at hours 1, 6, and 23
- Lowest transaction volume observed at hour 15

Challenges & Solutions

Memory Management

- **Challenge**: Encountered `MemoryError` when attempting to load all files into one DataFrame
- **Solution**: Implemented file-by-file processing approach to manage memory efficiently

Dataset Characteristics

- **Observation**: Discovered uniform transaction count (1.7M per day) across all files
- **Conclusion**: Dataset is likely synthetic with predetermined daily transaction volumes

- **Approach**: Focused analysis on merchant patterns and anomaly detection logic where natural variation exists

Methodology & Approach

Technical Approach

- Prioritized memory efficiency due to 4GB+ total data size
- Designed reusable, restartable pipelines for production readiness
- Implemented structured logging for better traceability
- Utilized `defaultdict` for efficient aggregation to avoid memory-intensive joins

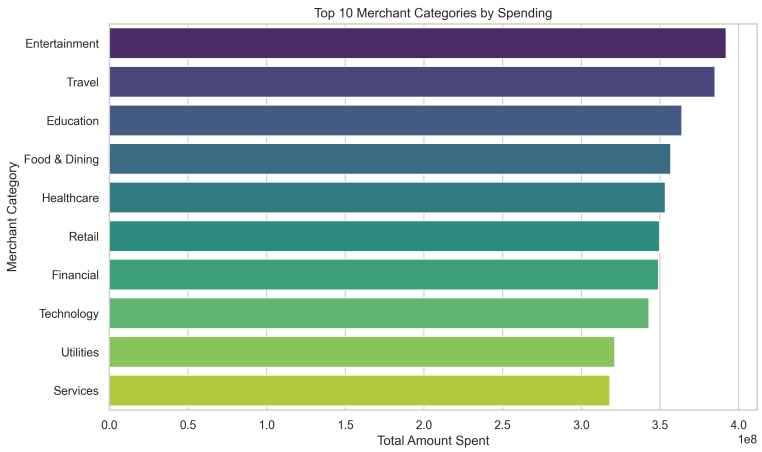
Data Processing Strategy

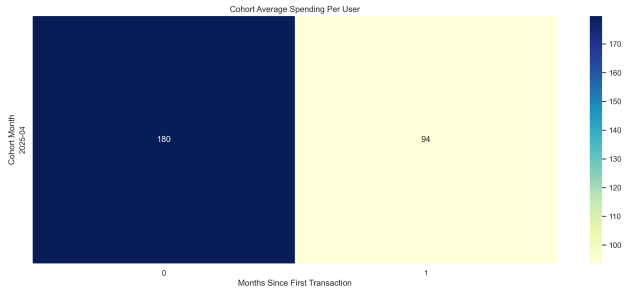
- Applied incremental processing to handle large dataset volumes
- Implemented validation at each stage of the pipeline
- Created diagnostic scripts to verify file structure consistency

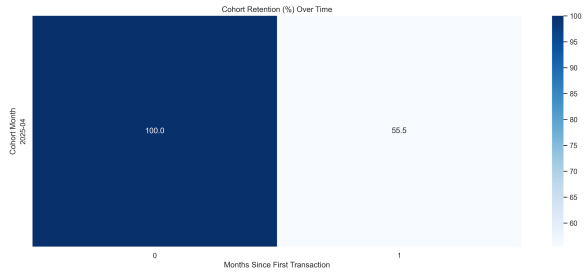
Conclusion

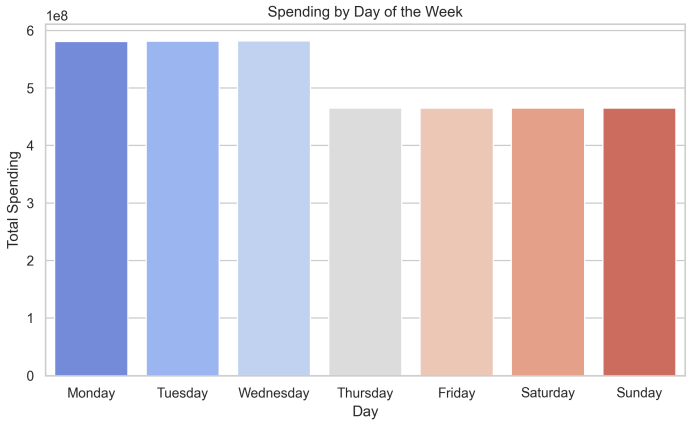
The analysis revealed that while the dataset appears to be synthetic with uniform daily transaction counts, there are meaningful patterns in merchant activity and hourly transaction distribution. The anomaly detection logic implemented provides a foundation for identifying unusual transaction patterns in this controlled dataset environment.

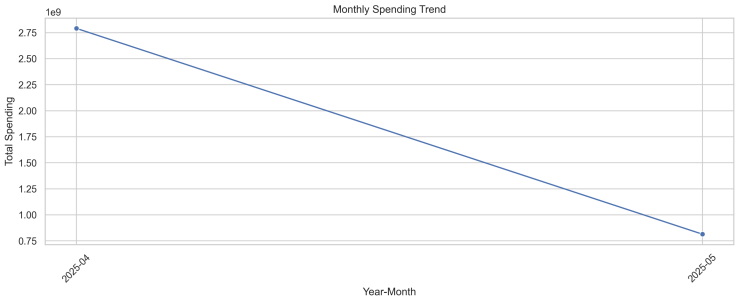
The data pipeline developed is robust, memory-efficient, and handles the full lifecycle from data retrieval to analysis and visualization.

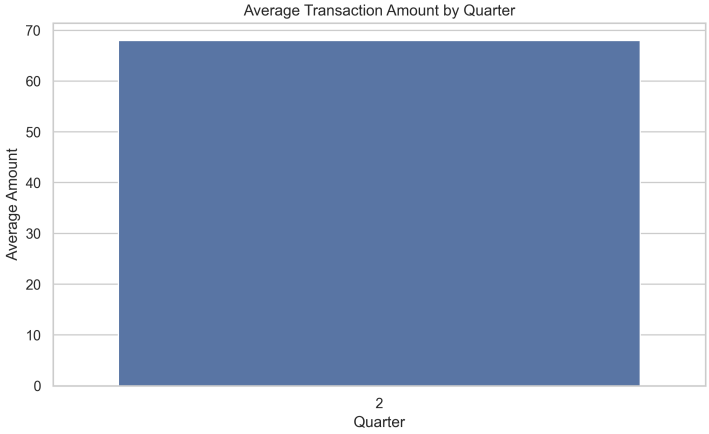


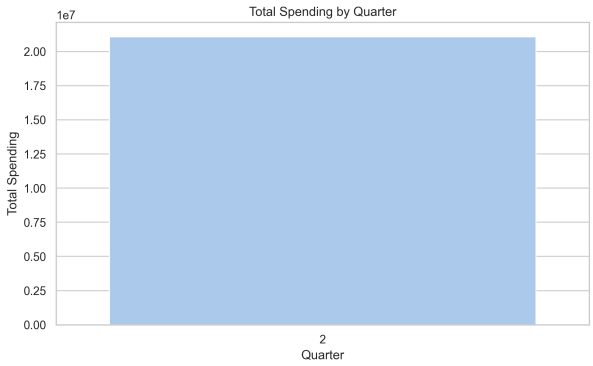

Cohort Average Spending Per User
Cohort Month
2025-04
180
94
0
1
Months Since First Transaction
170
160
150
140
130
120
110
100

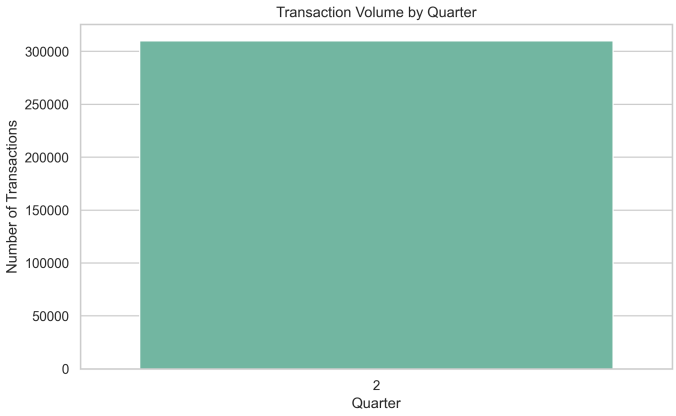

Cohort Retention (%) Over Time
Cohort Month
2025-04
100.0
55.5
0
1
Months Since First Transaction
100
95
90
85
80
75
70
65
60

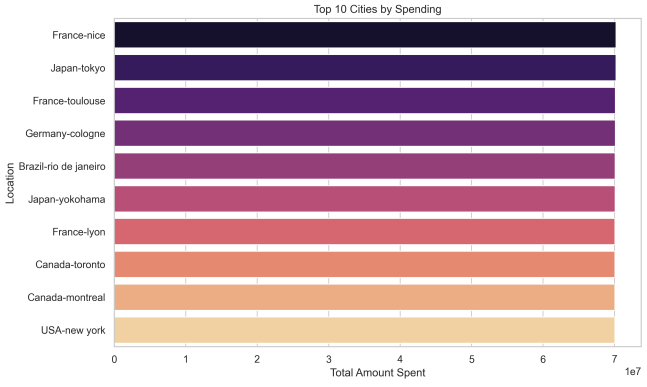

Spending by Day of the Week
1e8
Total Spending
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Day


Monthly Spending Trend
1e9
Total Spending
2025-04
2025-05
Year-Month


Average Transaction Amount by Quarter
Average Amount
2
Quarter


Total Spending by Quarter
1e7
Total Spending
2
Quarter


Transaction Volume by Quarter
Number of Transactions
2
Quarter


Top 10 Cities by Spending
France-nice
Japan-tokyo
France-toulouse
Germany-cologne
Brazil-rio de janeiro
Japan-yokohama
France-lyon
Canada-toronto
Canada-montreal
USA-new york
Location
Total Amount Spent
1e7