Personal Finance & Spending Pattern Analysis

Course: Python Programming End-To-End Project **Date:** 23rd February 2026

Made By: Sanyam Chanana

Data Description

This dataset comprises a comprehensive collection of personal financial transaction records, capturing both income and expenditure activities across an extended time period. Each record is characterized by multiple attributes — including the transaction date, spending category (such as Food, Fuel, Shopping, Rent, Entertainment, and more), payment mode (UPI, Card, Cash, or Bank Transfer), transaction amount, and transaction type (Income or Expense).

The richness of the dataset — spanning both numerical variables (transaction amounts, dates) and categorical variables (categories, payment modes, transaction types) — makes it well-suited for in-depth exploratory analysis. It facilitates the examination of spending behavior, monthly financial trends, savings accumulation, and overall fiscal discipline, while also enabling robust data visualization and statistical interpretation.

Problem Statement

Personal financial management is a critical life skill, yet spending patterns often go unexamined without the aid of structured data analysis. This project aims to bridge that gap by applying data analytics techniques to real-world personal transaction data.

The core objectives of this study are to:

1. Analyze Monthly Income and Expenses — Track and compare income against expenditure on a month-by-month basis to reveal periods of surplus or deficit.

2. Identify Major Spending Categories — Determine which categories (e.g., Food, Rent, Shopping) consume the largest share of total expenditure, enabling targeted financial awareness.

3. Evaluate Savings Trends — Assess how savings evolve over time and identify patterns that reflect sound or concerning financial behavior.

4. Understand Payment Behavior — Examine the distribution of payment modes to understand preferences and their implications for financial tracking and security.

5. Provide Actionable Insights for Financial Management — Translate analytical findings into practical, data-backed recommendations for improved budgeting and financial decision-making.

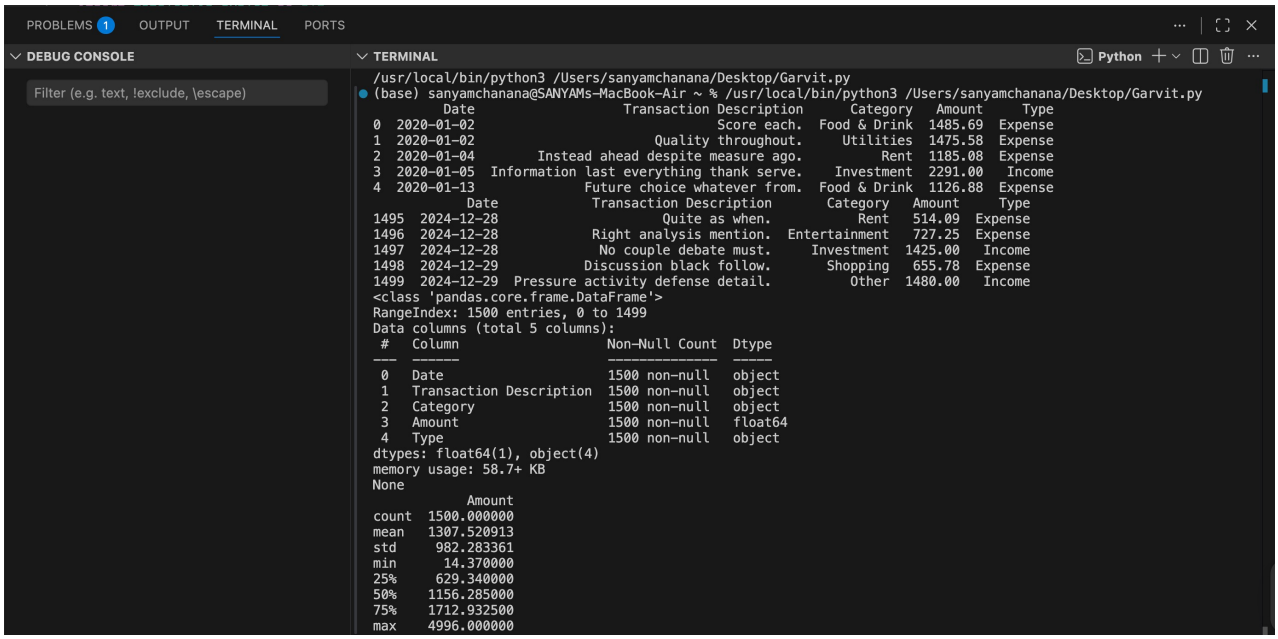
Through systematic Exploratory Data Analysis (EDA) and meaningful visualization, this project demonstrates how Python-driven data analytics can serve as a powerful tool for personal financial planning and long-term fiscal health.

1 Import Libraries & Load Dataset

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load dataset (update filename if needed)
data = pd.read_csv("/Users/sanyamchanana/Desktop/personal-finance.csv")

# Display dataset structure
print(data.head())
print(data.tail())
print(data.info())
print(data.describe())
```



2 Data Preprocessing

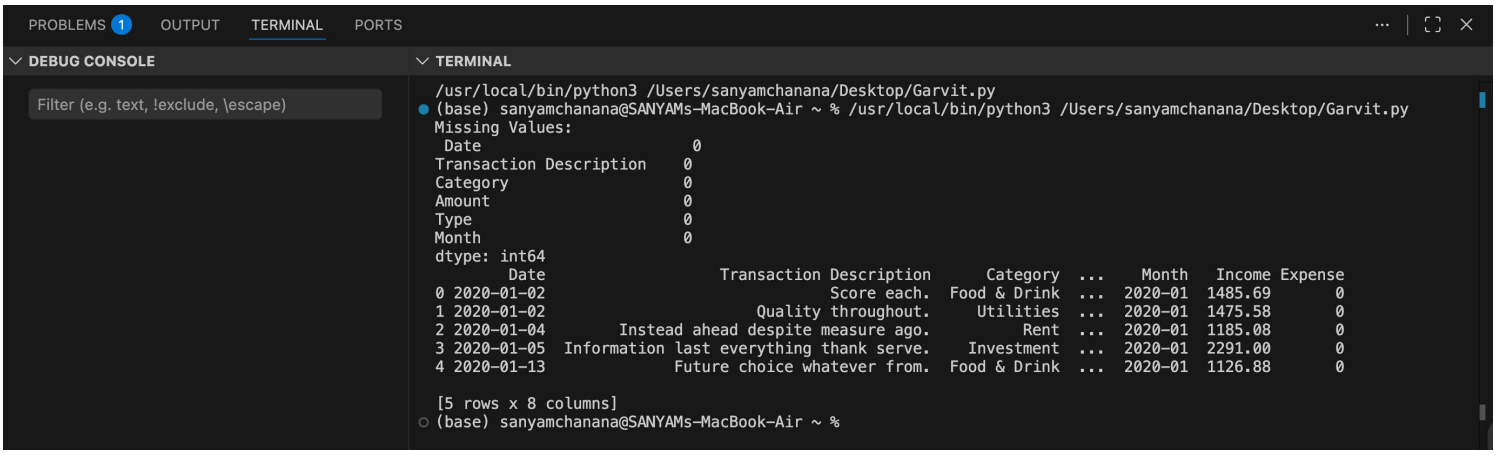
```
# Convert Transaction_Date column to datetime
data['Transaction_Date'] = pd.to_datetime(data['Transaction_Date'])

# Create Month column
data['Month'] = data['Transaction_Date'].dt.to_period('M')

# Create Income column
data['Income'] = data.apply(
```

```
    lambda row: row['Amount'] if row['Transaction_Type'] ==
'Income' else 0,
    axis=1
)

# Create Expense column
data['Expense'] = data.apply(
    lambda row: row['Amount'] if row['Transaction_Type'] ==
'Expense' else 0,
    axis=1
)
```



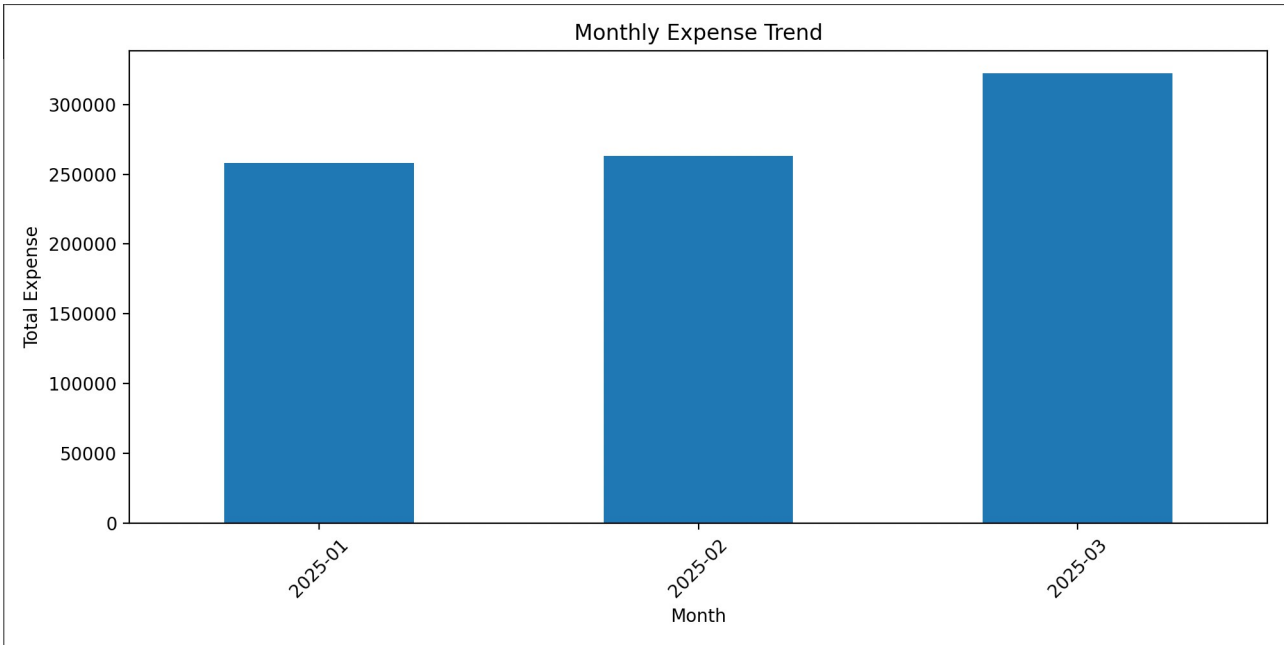
3 Monthly Expense Trend

```
monthly_expense = data.groupby('Month')['Expense'].sum()

plt.figure(figsize=(10,5))
monthly_expense.plot(kind='bar')

plt.title("Monthly Expense Trend")
plt.xlabel("Month")
plt.ylabel("Total Expense")
plt.xticks(rotation=45)

plt.tight_layout()
plt.show()
```



4 Category-wise Expense Analysis

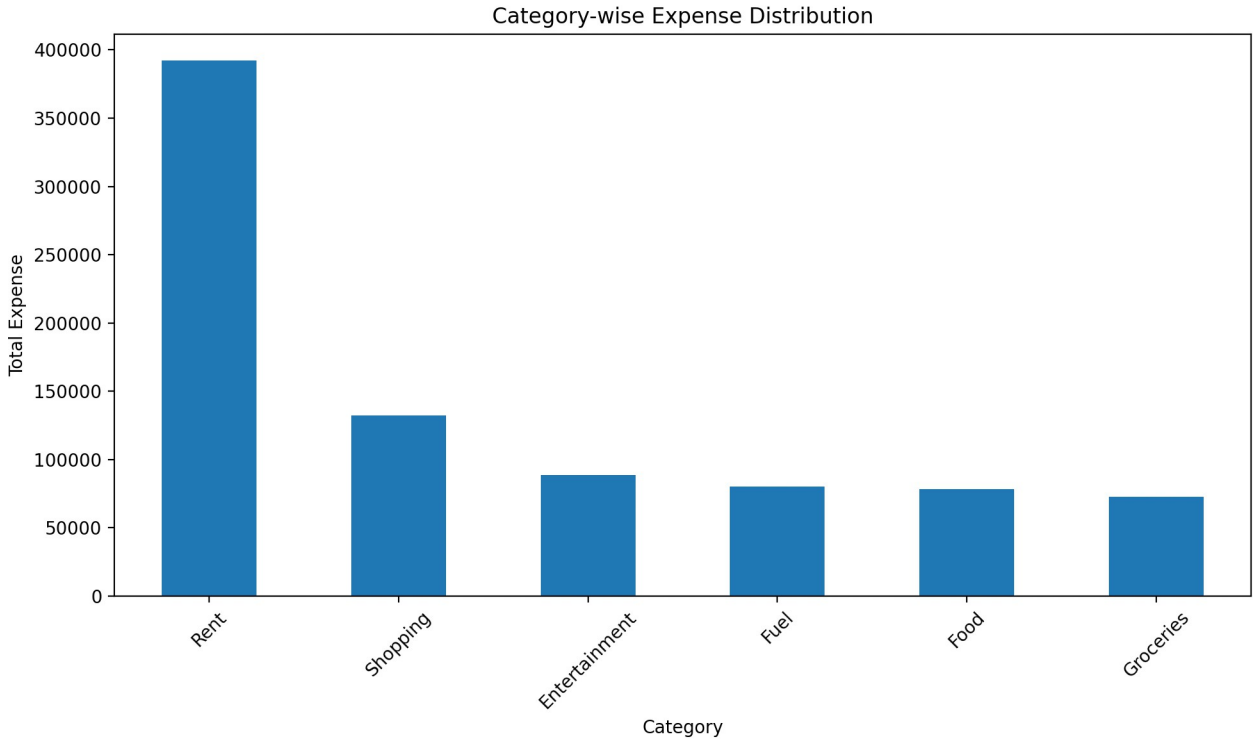
```
# Filter only Expense transactions
expense_data = data[data['Transaction_Type'] == 'Expense']

# Group by Category
category_expense = expense_data.groupby('Category')
['Amount'].sum().sort_values(ascending=False)

plt.figure(figsize=(10,6))
category_expense.plot(kind='bar')

plt.title("Category-wise Expense Distribution")
plt.xlabel("Category")
plt.ylabel("Total Expense")
plt.xticks(rotation=45)

plt.tight_layout()
plt.show()
```



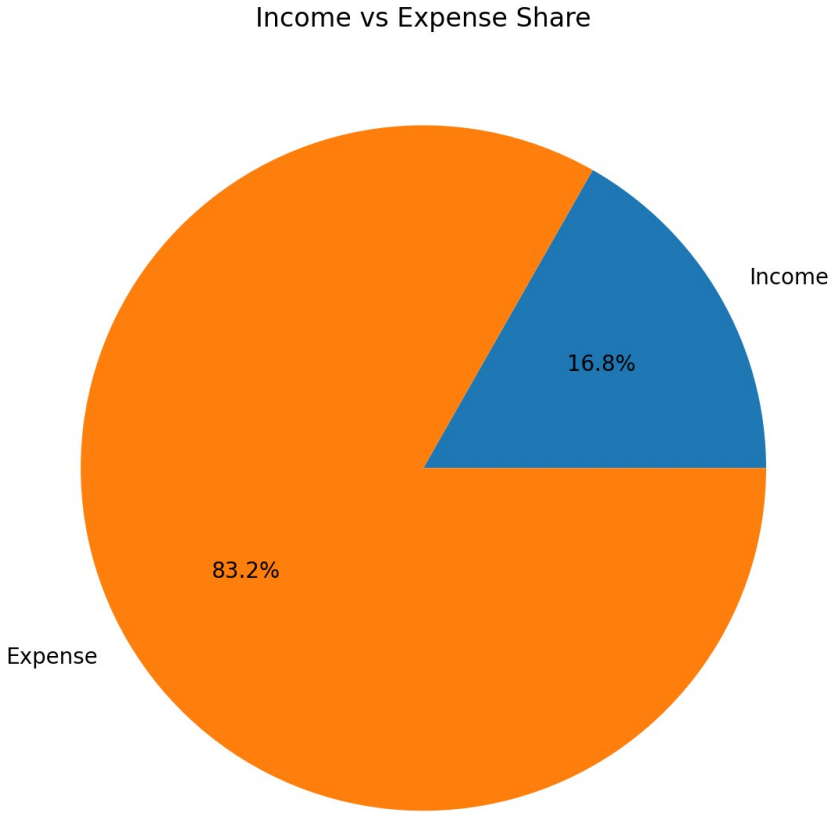
5 Income vs Expense Comparison

```
# Calculate totals correctly
total_income = data[data['Transaction_Type'] == 'Income']
['Amount'].sum()
total_expense = data[data['Transaction_Type'] == 'Expense']
['Amount'].sum()

plt.figure(figsize=(6,6))
plt.pie(
    [total_income, total_expense],
    labels=['Income', 'Expense'],
    autopct='%1.1f%%'
)

plt.title("Income vs Expense Share")

plt.tight_layout()
plt.show()
```



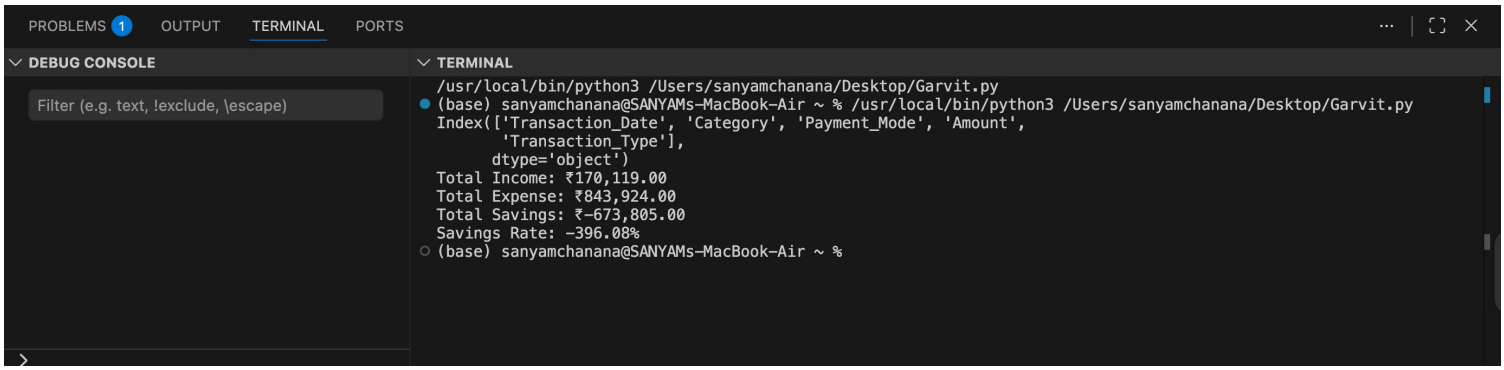
6 Savings Calculation

```
# Calculate totals properly
total_income = data.loc[data['Transaction_Type'] == 'Income',
'Amount'].sum()
total_expense = data.loc[data['Transaction_Type'] == 'Expense',
'Amount'].sum()

# Calculate savings
savings = total_income - total_expense

# Print formatted results
print(f"Total Income: ₹{total_income:,.2f}")
print(f"Total Expense: ₹{total_expense:,.2f}")
print(f"Total Savings: ₹{savings:,.2f}")

# Calculate savings rate safely
if total_income > 0:
    savings_rate = (savings / total_income) * 100
    print(f"Savings Rate: {savings_rate:.2f}%")
else:
    print("Savings Rate: Cannot calculate (Income is 0)")
```



7 Monthly Savings Trend

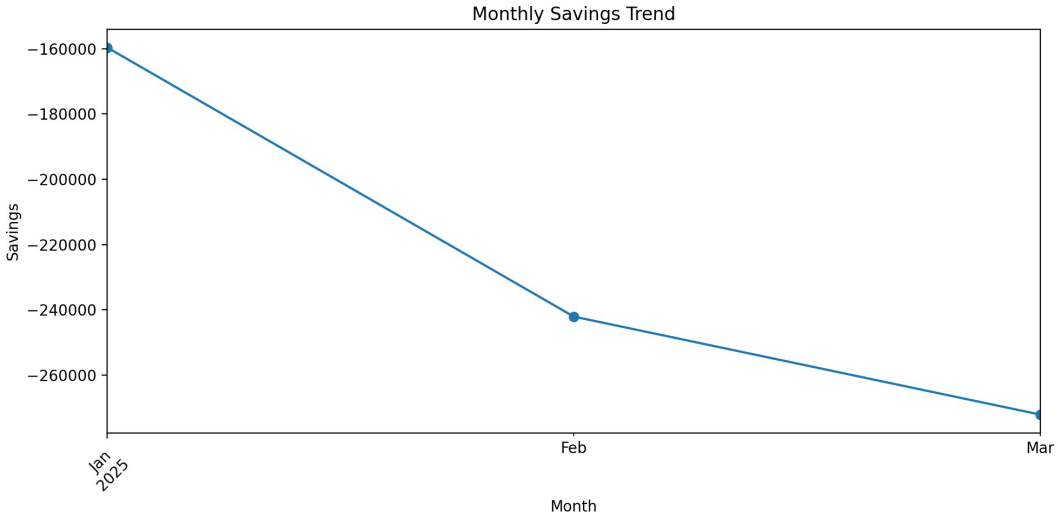
```
# Create monthly summary table
monthly_summary = (
    data.groupby(['Month', 'Transaction_Type'])['Amount']
    .sum()
    .unstack(fill_value=0)
)

# Create Savings column
monthly_summary['Savings'] = monthly_summary['Income'] -
monthly_summary['Expense']

# Plot Savings Trend
plt.figure(figsize=(10,5))
monthly_summary['Savings'].plot(marker='o')

plt.title("Monthly Savings Trend")
plt.xlabel("Month")
plt.ylabel("Savings")
plt.xticks(rotation=45)

plt.tight_layout()
plt.show()
```



8 Top 5 Highest Expenses

```
# Filter only expense transactions
expense_data = data[data['Transaction_Type'] == 'Expense']

# Sort by Amount (highest first)
top_expenses = expense_data.sort_values(by='Amount',
ascending=False).head(5)

print("Top 5 Highest Expenses:")
print(top_expenses[['Transaction_Date', 'Category', 'Amount']])
```

```
Savings Rate: -396.08%
Top 5 Highest Expenses:
    Transaction_Date Category  Amount
175       2025-03-19     Rent   14939
83        2025-02-05     Rent   14435
186       2025-03-24     Rent   14228
153       2025-03-09     Rent   14161
143       2025-03-06     Rent   13571
(base) sanyamchanana@SANYAMs-MacBook-Air ~ %
```

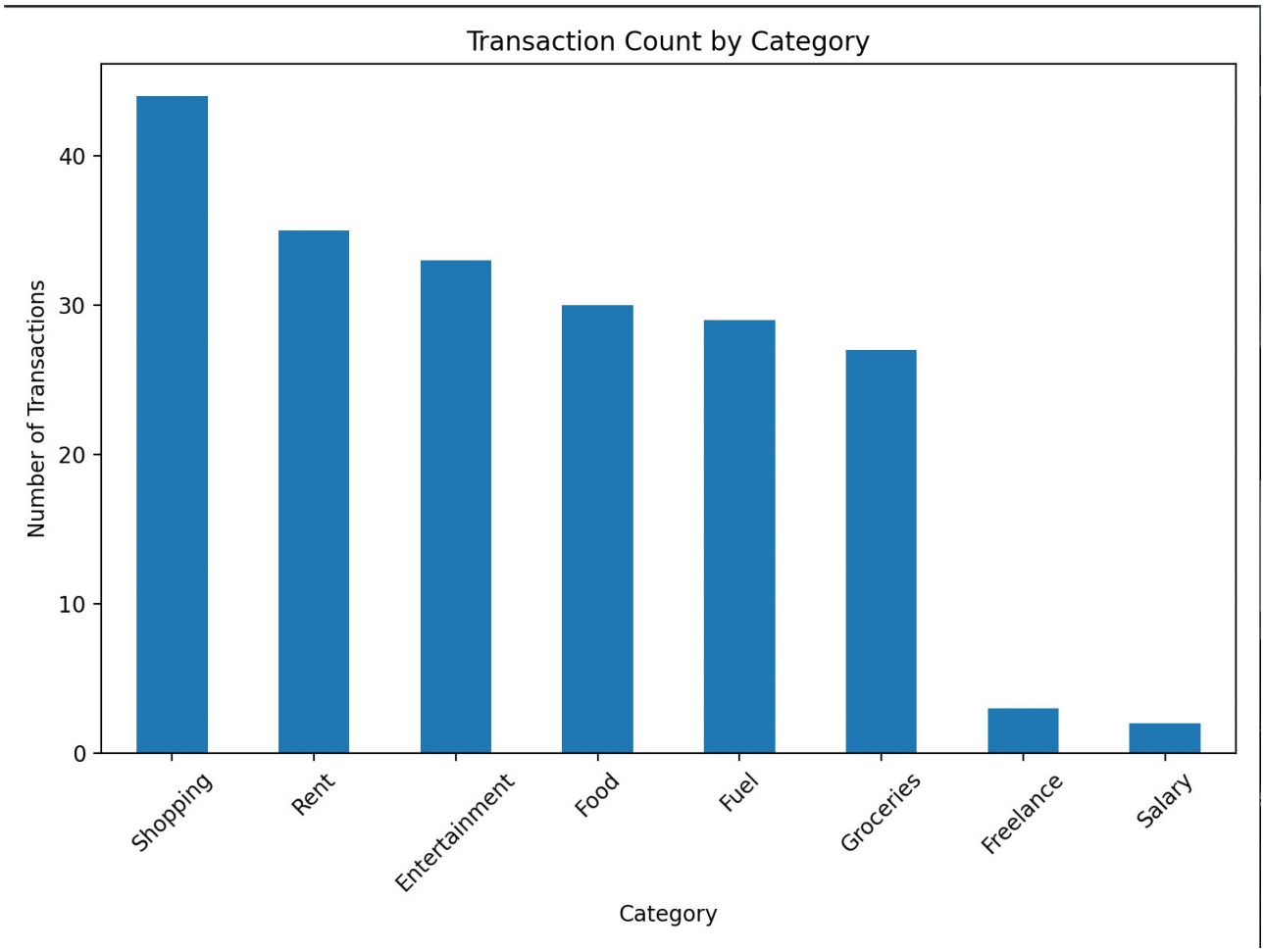
9 Transaction Count by Category

```
transaction_count = data['Category'].value_counts()

plt.figure(figsize=(8,6))
transaction_count.plot(kind='bar')
```

```
plt.title("Transaction Count by Category")
plt.xlabel("Category")
plt.ylabel("Number of Transactions")
plt.xticks(rotation=45)

plt.tight_layout()
plt.show()
```



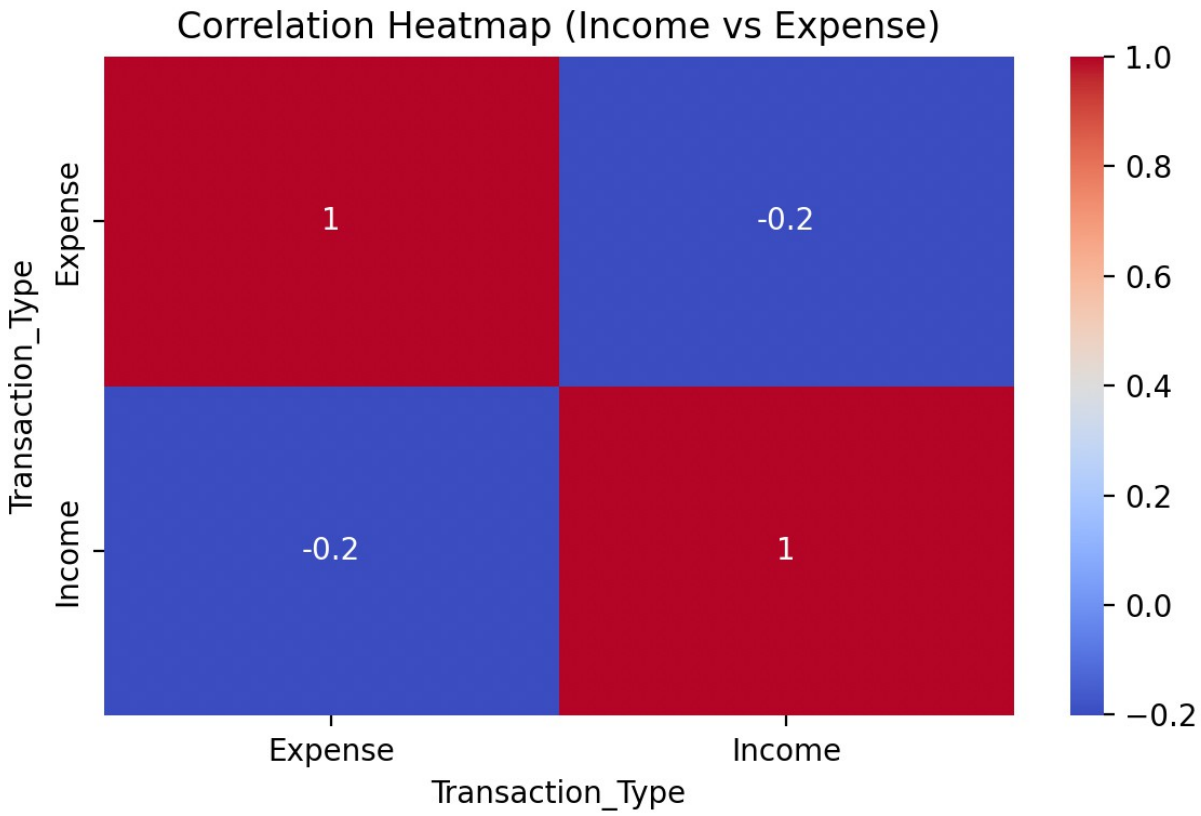
🔟 Correlation Heatmap

```
# Create monthly summary
monthly_summary = (
    data.groupby(['Month', 'Transaction_Type'])['Amount']
    .sum()
    .unstack(fill_value=0)
)

plt.figure(figsize=(6,4))
sns.heatmap(monthly_summary.corr(), annot=True, cmap='coolwarm')

plt.title("Correlation Heatmap (Income vs Expense)")
```

```
plt.tight_layout()
plt.show()
```



Conclusion

This project undertook a systematic analysis of personal financial transaction data with the goal of uncovering meaningful patterns in income, expenditure, and savings behavior. The findings yielded several significant insights that collectively paint a comprehensive picture of personal financial health.

Monthly expense trend analysis revealed notable fluctuations in spending behavior across different time periods, suggesting the influence of seasonal factors, irregular purchases, or varying lifestyle demands. These variations underscored the importance of continuous financial monitoring rather than relying on static, one-time assessments.

Category-wise expenditure analysis successfully pinpointed the primary drivers of total spending, highlighting which areas — such as Food, Rent, Shopping, or Fuel — command the greatest financial attention. This granular breakdown enables more targeted budgeting strategies and helps identify categories where spending optimisation is most feasible.

The comparative examination of income against expenses served as a reliable indicator of financial discipline and budget adherence. Periods of surplus and deficit were identified, offering a realistic assessment of overall fiscal management and the effectiveness of day-to-day financial decisions.

Savings and savings rate analysis further reinforced the evaluation of long-term financial stability, revealing whether income growth translated into proportionate savings accumulation or was offset by rising expenditure. This metric stands as one of the most telling indicators of sustainable financial behaviour.

In summation, this project affirms that data analytics is not merely a technical exercise — it is a practical and powerful instrument for personal financial empowerment. By transforming raw transaction records into structured, visual, and interpretable insights, this analysis demonstrates how Python-driven exploratory data analysis can meaningfully support smarter budgeting, disciplined spending, and well-informed financial decision-making. The methodologies applied here can be readily extended to larger datasets or integrated into automated personal finance dashboards for ongoing, real-time financial oversight.