Walmart Inc. Black Friday Purchase Behavior Analysis

1. Business Problem

The Management team at Walmart Inc. wants to analyze the customer purchase behavior (specifically, purchase amount) against the customer's gender and the various other factors to help the business make better decisions. They want to understand if the spending habits differ between male and female customers: Do women spend more on Black Friday than men?

2. Data Description

The company collected the transactional data of customers who purchased products from Walmart Stores during Black Friday. The dataset in walmart_data.csv has the following features:

User_ID - (A unique identification number assigned to each customer)
Product_ID - (A unique identification number for each item sold)
Gender - (The sex of the customer (usually 'M' for Male and 'F' for Female)
Age - (Age of the customer, usually grouped into "bins" (e.g., 0-17, 18-25, 26-35)
Occupation - (A numerical code representing the customer's profession)
City_Category - (The category of the city where the customer lives (A, B, or C))
Stay_In_Current_City_Years - (number of years a customer stays in their current city)
Marital_Status - (A binary value (0 or 1) representing if the user is married or single)
Product_Category - (Categories/Groups of products (e.g., Electronics, Clothing, Home)
Purchase - (The total amount spent on a particular transaction)

Dataset

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category	Purchase
0	1000001	P00069042	F	0-17	10	A	2	0	3	8370
1	1000001	P00248942	F	0-17	10	A	2	0	1	15200
2	1000001	P00087842	F	0-17	10	A	2	0	12	1422
3	1000001	P00085442	F	0-17	10	A	2	0	12	1057
4	1000002	P00285442	M	55+	16	C	4+	0	8	7969
5	1000003	P00193542	M	26-35	15	A	3	0	1	15227
6	1000004	P00184942	M	46-50	7	B	2	1	1	19215
7	1000004	P00346142	M	46-50	7	B	2	1	1	15854
8	1000004	P0097242	M	46-50	7	B	2	1	1	15686
9	1000005	P00274942	M	26-35	20	A	1	1	8	7871
10	1000005	P00251242	M	26-35	20	A	1	1	5	5254
11	1000005	P00014542	M	26-35	20	A	1	1	8	3957
12	1000005	P00031342	M	26-35	20	A	1	1	8	6073
13	1000005	P00145042	M	26-35	20	A	1	1	1	15665
14	1000006	P00231342	F	51-55	9	A	1	0	5	5378
15	1000006	P00190242	F	51-55	9	A	1	0	4	2079
16	1000006	P0096642	F	51-55	9	A	1	0	2	13055
17	1000006	P00058442	F	51-55	9	A	1	0	5	8851
18	1000007	P00036842	M	36-45	1	B	1	1	1	11788
19	1000008	P00249542	M	26-35	12	C	4+	1	1	19614

Automated Data Ingestion Pipeline Summary

1. Purpose

The script in previous slide automates the process of reading raw retail data and storing it in a structured SQLite database for further analysis.

2. Core Components

- Libraries Used: Utilizes pandas for data manipulation, sqlalchemy for database connection, and logging for process tracking.
- Database Engine: Creates a connection to a local SQLite database named Purchase.db.
- Logging System: Records every step of the process (start time, file reading, and errors) into a log file located at loggings2/ingestion_db.log.

3. Workflow Execution

- File Handling: The script locates a specific CSV file (walmart_data.csv) from the directory.
- Data Transfer: It reads the CSV into a DataFrame and uses the to_sql function to "replace" the existing table in the database with fresh data.
- Performance Monitoring: It calculates the total time taken for ingestion—successfully processing the data in approximately 7.86 seconds.

4. Error Handling

- Includes validation checks to ensure the source file exists before attempting to read it, preventing script crashes.

[1]:
```python
# import necessaries library
import numpy as np
import pandas as pd
from sqlalchemy import create_engine
import logging
import time
import os
```

[2]:
```python
# logging
logging.getLogger().handler = []
logging.basicConfig(
    filename = 'loggings2/ingestion_db.log',
    level = logging.DEBUG,
    format = '%(asctime)s - %(levelname)s - %(message)s',
    filemode = 'a'
)
```

```python
# Database engine
engine = create_engine('sqlite:///Purchase.db')
def ingest_db(df,table_name,engine):
    ''' This function will ingest the dataframe into database table'''
    df.to_sql(table_name,con=engine,if_exists = 'replace',index=False)

def load_raw_data():
    ''' This function will load a single CSV as dataframe and ingest into db '''
    start = time.time()

    # Write your file name path
    file_path = 'Data2/walmart_data.csv'

    if os.path.exists(file_path):
        # chaning file name from walmart_data.csv to walmart_data (e.g., 'sales.csv' -> 'sales')
        table_name = os.path.basename(file_path).replace('walmart_data.csv', 'walmart_data')

        logging.info(f'Reading file: {file_path}')
        df = pd.read_csv(file_path)

        logging.info(f'Ingesting {table_name} into Purchase.db')
        ingest_db(df, table_name, engine)

        end = time.time()
        total_time = (end - start)
        logging.info(f'Ingestion Complete. Total Time: {total_time:.2f} seconds')
        print(f"Data ingested successfully into table '{table_name}' in {total_time:.2f} seconds.")
    else:
        logging.warning(f"File not found at: {file_path}")
        print("Error: file not found. check your path.")

if __name__ == '__main__':
    load_raw_data()
```

```
Data ingested successfully into table 'walmart_data' in 7.14 seconds.
```

3. Exploratory Data Analysis using Python

1. Data Ingestion

The script connects to the Purchase.db SQLite database using the sqlite3 library and loads the walmart_data table into a Pandas DataFrame.

2. Initial Data Inspection

It examines the dataset's structure using df.info(), revealing 550,068 rows and 10 columns, while checking for null values and summarizing statistical properties using df.describe().

```
# check data type
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550068 entries, 0 to 550067
Data columns (total 10 columns):
 #   Column                      Non-Null Count   Dtype
---  ------                      --------------   -----
 0   User_ID                     550068 non-null  int64
 1   Product_ID                  550068 non-null  object
 2   Gender                      550068 non-null  object
 3   Age                         550068 non-null  object
 4   Occupation                  550068 non-null  int64
 5   City_Category               550068 non-null  object
 6   Stay_In_Current_City_Years  550068 non-null  object
 7   Marital_Status              550068 non-null  int64
 8   Product_Category            550068 non-null  int64
 9   Purchase                    550068 non-null  int64
dtypes: int64(5), object(5)
memory usage: 42.0+ MB
```

```
# summarize data
df.describe(include='all').T
```

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
User_ID	550068.0	NaN	NaN	NaN	1003028.842401	1727.591586	1000001.0	1001516.0	1003077.0	1004478.0	1006040.0
Product_ID	550068	3631	P00265242	1880	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Gender	550068	2	M	414259	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Age	550068	7	26-35	219587	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Occupation	550068.0	NaN	NaN	NaN	8.076707	6.52266	0.0	2.0	7.0	14.0	20.0
City_Category	550068	3	B	231173	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Stay_In_Current_City_Years	550068	5	1	193821	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Marital_Status	550068.0	NaN	NaN	NaN	0.409653	0.49177	0.0	0.0	0.0	1.0	1.0
Product_Category	550068.0	NaN	NaN	NaN	5.40427	3.936211	1.0	1.0	5.0	8.0	20.0
Purchase	550068.0	NaN	NaN	NaN	9263.968713	5023.065394	12.0	5823.0	8047.0	12054.0	23961.0

4. Check for Missing and Null values:

used df.isnull() to check for missing and null values.

```
df.isnull().sum()/len(df)*100
```

```
User_ID                       0.0
Product_ID                    0.0
Gender                        0.0
Age                           0.0
Occupation                    0.0
City_Category                 0.0
Stay_In_Current_City_Years    0.0
Marital_Status                0.0
Product_Category              0.0
Purchase                      0.0
dtype: float64
```

There are no missing values found

5. **Feature Engineering & Data Transformation Summary:**

- Categorizing Gender Labels: Used the .replace() function to transform numeric and short-code values into descriptive labels, mapping 'M' and '1' to 'Male', and 'F' and '0' to 'Female' for better readability.
- Cleaning Numerical Strings: Applied .str.split('+').str[0] to the 'Stay_In_Current_City_Years' column to remove special characters and convert it into a clean numerical format.
- Creating 'Age_Category' Feature: Used a custom mapping dictionary with the .map() function to create a new high-level feature called age_category.
- Age-Based Classification: Categorized specific age ranges into broader groups, such as labeling '0-17' as 'Minor' and '26-35' as 'Young Adult' to simplify demographic analysis.
- Feature Renaming: Utilized the df.rename() function to change the long column name 'Stay_In_Current_City_Years' to the more concise 'Years'.
- Standardizing Headers: Applied df.columns.str.lower() to ensure all feature names follow a consistent lowercase format, making them easier to reference in code.

[17]:
```
for col in df[['City_Category','Stay_In_Current_City_Years','Marital_Status','Product_Category']]:
    unique_values = df[col].unique()
    print('unique_values in: ',col,unique_values)
```

```
unique_values in:  City_Category ['A' 'C' 'B']
unique_values in:  Stay_In_Current_City_Years ['2' '4+' '3' '1' '0']
unique_values in:  Marital_Status [0 1]
unique_values in:  Product_Category [ 3  1 12  8  5  4  2  6 14 11 13 15  7 16 18 10 17  9 20 19]
```

[18]:
```
df['Stay_In_Current_City_Years']=df['Stay_In_Current_City_Years'].str.split('+').str[0].astype('int64')
```

[19]:
```
df['Gender'] = df['Gender'].replace({'M':'Male','F':'Female'})
```

[20]:
```
df['Marital_Status'] = df['Marital_Status'].replace({0:'Married',1:'Unmarried'})
```

[21]:
```
# 'Old_Name' ko 'New_Name' se badalne ke liye
df = df.rename(columns={'Stay_In_Current_City_Years': 'Years'})
```

[22]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550068 entries, 0 to 550067
Data columns (total 10 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   User_ID           550068 non-null  int64
 1   Product_ID        550068 non-null  object
 2   Gender            550068 non-null  object
 3   Age               550068 non-null  object
 4   Occupation        550068 non-null  int64
 5   City_Category     550068 non-null  object
 6   Years             550068 non-null  int64
 7   Marital_Status    550068 non-null  object
 8   Product_Category  550068 non-null  int64
 9   Purchase          550068 non-null  int64
dtypes: int64(5), object(5)
memory usage: 42.0+ MB
```

[24]:
```
age_mapping = {
    '0-17': 'Minor',
    '18-25': 'Young Adult',
    '26-35': 'Young Adult',
    '36-45': 'Adult',
    '46-50': 'Adult',
    '51-55': 'Senior',
    '55+': 'Senior'
}

# Naya column banayein
df['Age_Category'] = df['Age'].map(age_mapping)

# Ab iska count dekhein
print(df['Age_Category'].value_counts())
```

```
Age_Category
Young Adult    319247
Adult          155714
Senior          60005
Minor           15102
Name: count, dtype: int64
```

[26]:
```
# converting columns name into lower cases with underscore
df.columns = df.columns.str.lower()
df.columns = df.columns.str.replace(' ','_')
```

Transformed Data

	user_id	product_id	gender	occupation	city_category	years	marital_status	product_category	purchase	age_category
0	1000001	P00069042	Female	10	A	2	Married	3	8370	Minor
1	1000001	P00248942	Female	10	A	2	Married	1	15200	Minor
2	1000001	P00087842	Female	10	A	2	Married	12	1422	Minor
3	1000001	P00085442	Female	10	A	2	Married	12	1057	Minor
4	1000002	P00285442	Male	16	C	4	Married	8	7969	Senior
5	1000003	P00193542	Male	15	A	3	Married	1	15227	Young Adult
6	1000004	P00184942	Male	7	B	2	Unmarried	1	19215	Adult
7	1000004	P00346142	Male	7	B	2	Unmarried	1	15854	Adult
8	1000004	P0097242	Male	7	B	2	Unmarried	1	15686	Adult
9	1000005	P00274942	Male	20	A	1	Unmarried	8	7871	Young Adult
10	1000005	P00251242	Male	20	A	1	Unmarried	5	5254	Young Adult
11	1000005	P00014542	Male	20	A	1	Unmarried	8	3957	Young Adult
12	1000005	P00031342	Male	20	A	1	Unmarried	8	6073	Young Adult
13	1000005	P00145042	Male	20	A	1	Unmarried	1	15665	Young Adult
14	1000006	P00231342	Female	9	A	1	Married	5	5378	Senior
15	1000006	P00190242	Female	9	A	1	Married	4	2079	Senior
16	1000006	P0096642	Female	9	A	1	Married	2	13055	Senior
17	1000006	P00058442	Female	9	A	1	Married	5	8851	Senior
18	1000007	P00036842	Male	1	B	1	Unmarried	1	11788	Adult
19	1000008	P00249542	Male	12	C	4	Unmarried	1	19614	Young Adult

4. Data Analysis using SQL

Performed structured Analysis in python using sqlite3 to answer key business questions:

- **Database Connection:** Uses sqlite3.connect('Purchase.db') to establish a secure link with the stored data.
- **Data Retrieval via SQL:** Employs pd.read_sql_query() to execute SQL commands and load the results directly into a Pandas DataFrame for analysis.

```
📊 Total Records: 550,068
📋 Columns: 10

🔍 Data Overview:
```

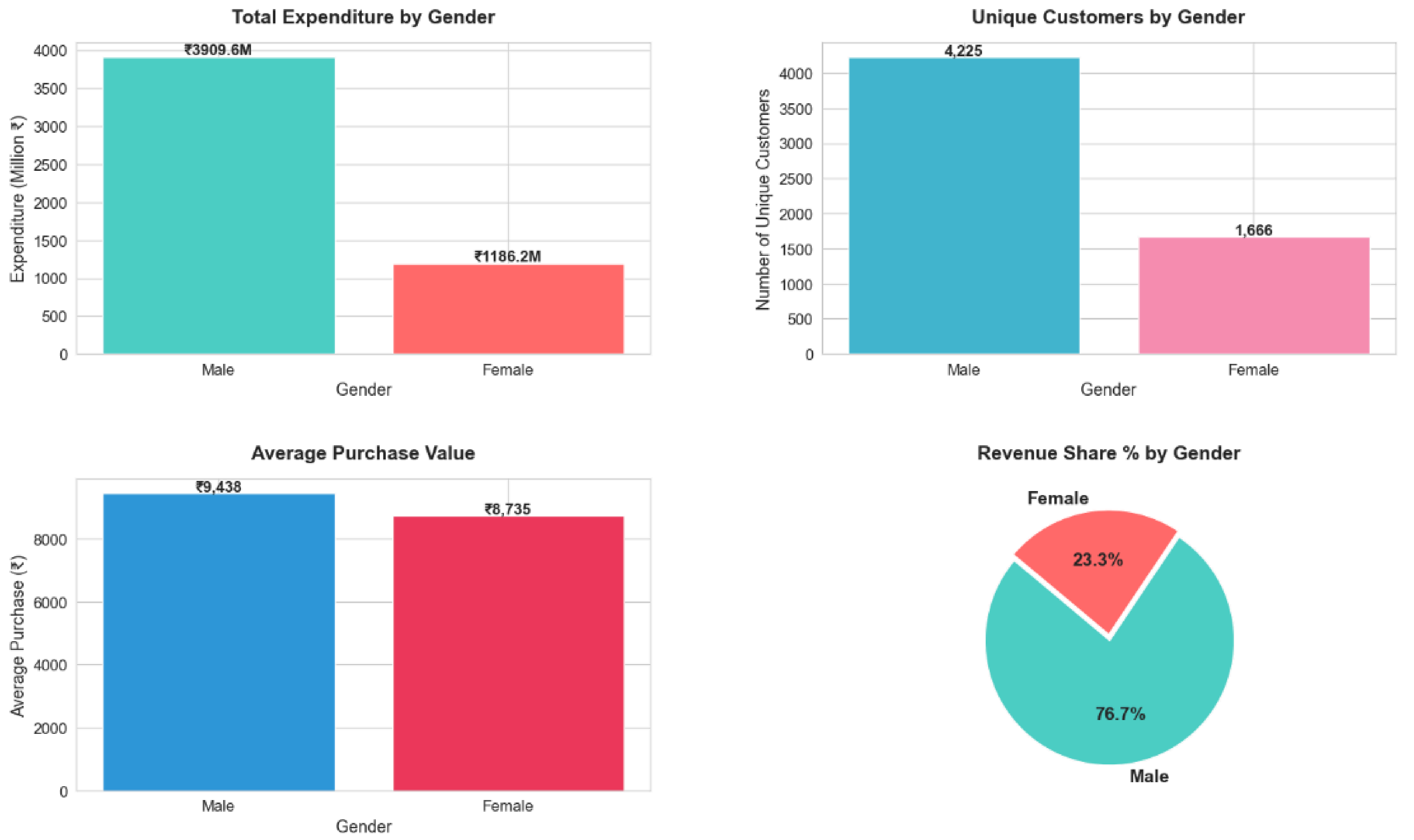
	user_id	product_id	gender	occupation	city_category	years	marital_status	product_category	purchase	age_category
0	1000001	P00069042	Female	10	A	2	Married	3	8370	Minor
1	1000001	P00248942	Female	10	A	2	Married	1	15200	Minor
2	1000001	P00087842	Female	10	A	2	Married	12	1422	Minor
3	1000001	P00085442	Female	10	A	2	Married	12	1057	Minor
4	1000002	P00285442	Male	16	C	4	Married	8	7969	Senior

📊 Analysis 1: Gender-wise Purchase Behavior

Business Question: What are the purchase patterns between Male vs Female customers?

👨👩 GENDER-WISE PURCHASE ANALYSIS
==

	gender	Total_Transactions	Unique_Customers	Total_Expenditure	Avg_Purchase	Min_Purchase	Max_Purchase
0	Male	414259	4225	3909580100	9437.53	12.0	23961.0
1	Female	135809	1666	1186232642	8734.57	12.0	23959.0



💡 KEY INSIGHTS:

- Male: 76.72% of total revenue
  - Average Purchase: ₹9,437.53
  - Transactions per Customer: 98.05
- Female: 23.28% of total revenue
  - Average Purchase: ₹8,734.57
  - Transactions per Customer: 81.52

==
💡 GENDER ANALYSIS - KEY BUSINESS INSIGHTS
==

🎯 MAIN QUESTION ANSWER:
Males spend MORE on average per transaction.
Difference: ₹702.96 per transaction

👥 CUSTOMER BASE:
- Male customers: 4,225 (76.7% of revenue)
- Female customers: 1,666 (23.3% of revenue)
- Male:Female ratio is approximately 3:1

💰 SPENDING BEHAVIOR:
- Male avg purchase: ₹9,437.53
- Female avg purchase: ₹8,734.57
- Males make 98.0 transactions per customer
- Females make 81.5 transactions per customer

📊 REVENUE SPLIT:
- Males contribute 76.72% of total revenue
- Females contribute 23.28% of total revenue

📊 Analysis 2: Age Group Purchase Patterns

Business Question: Which age group has the highest purchasing power?

👥 AGE & GENDER-WISE PURCHASE ANALYSIS
==

	age_category	gender	Transactions	Customers	Total_Expenditure_Million	Avg_Purchase
0	Adult	Female	40369	515	360.0	8921.35
1	Adult	Male	115345	1183	1087.0	9426.22
2	Minor	Female	5083	78	42.0	8338.77
3	Minor	Male	10019	140	92.0	9235.17
4	Senior	Female	14977	241	135.0	9030.43
5	Senior	Male	45028	612	432.0	9607.76
6	Young Adult	Female	75380	832	648.0	8602.44
7	Young Adult	Male	243867	2290	2297.0	9419.75

💰 TOP REVENUE GENERATING AGE GROUPS:

	age_category	Total_Transactions	Total_Revenue_Million	Avg_Purchase
0	Young Adult	319247	2945.0	9226.77
1	Adult	155714	1447.0	9295.33
2	Senior	60005	567.0	9463.66

==
💡 AGE GROUP ANALYSIS - KEY BUSINESS INSIGHTS
==

🏆 TOP REVENUE AGE GROUP:
- Young Adult generates maximum revenue: ₹2945.0M
- This group has 319,247 total transactions

📊 AGE GROUP RANKINGS BY REVENUE:
1. Young Adult: ₹2945.0M (Avg: ₹9,227)
2. Adult: ₹1447.0M (Avg: ₹9,295)
3. Senior: ₹567.0M (Avg: ₹9,464)
4. Minor: ₹134.0M (Avg: ₹8,933)

💰 AVERAGE PURCHASE INSIGHTS:
- Highest avg purchase: Senior → ₹9,463.66
- Lowest avg purchase: Minor → ₹8,933.46

==

📊 Analysis 3: City Category Impact

Business Question: In which city category do customers spend the most?

```
🏙 CITY CATEGORY ANALYSIS
======================================================================
```

	city_category	gender	Transactions	Unique_Customers	Revenue_Million	Avg_Purchase	Revenue_Per_Customer
0	A	Female	35704	295	306.0	8579.71	1038406.49
1	A	Male	112016	750	1010.0	9017.83	1346855.66
2	B	Female	57796	503	493.0	8540.68	981345.94
3	B	Male	173377	1204	1621.0	9354.85	1347106.81
4	C	Female	42309	868	386.0	9130.11	445029.63
5	C	Male	128866	2271	1277.0	9913.57	562537.10

```
🏆 CITY CATEGORY RANKING BY REVENUE:
```

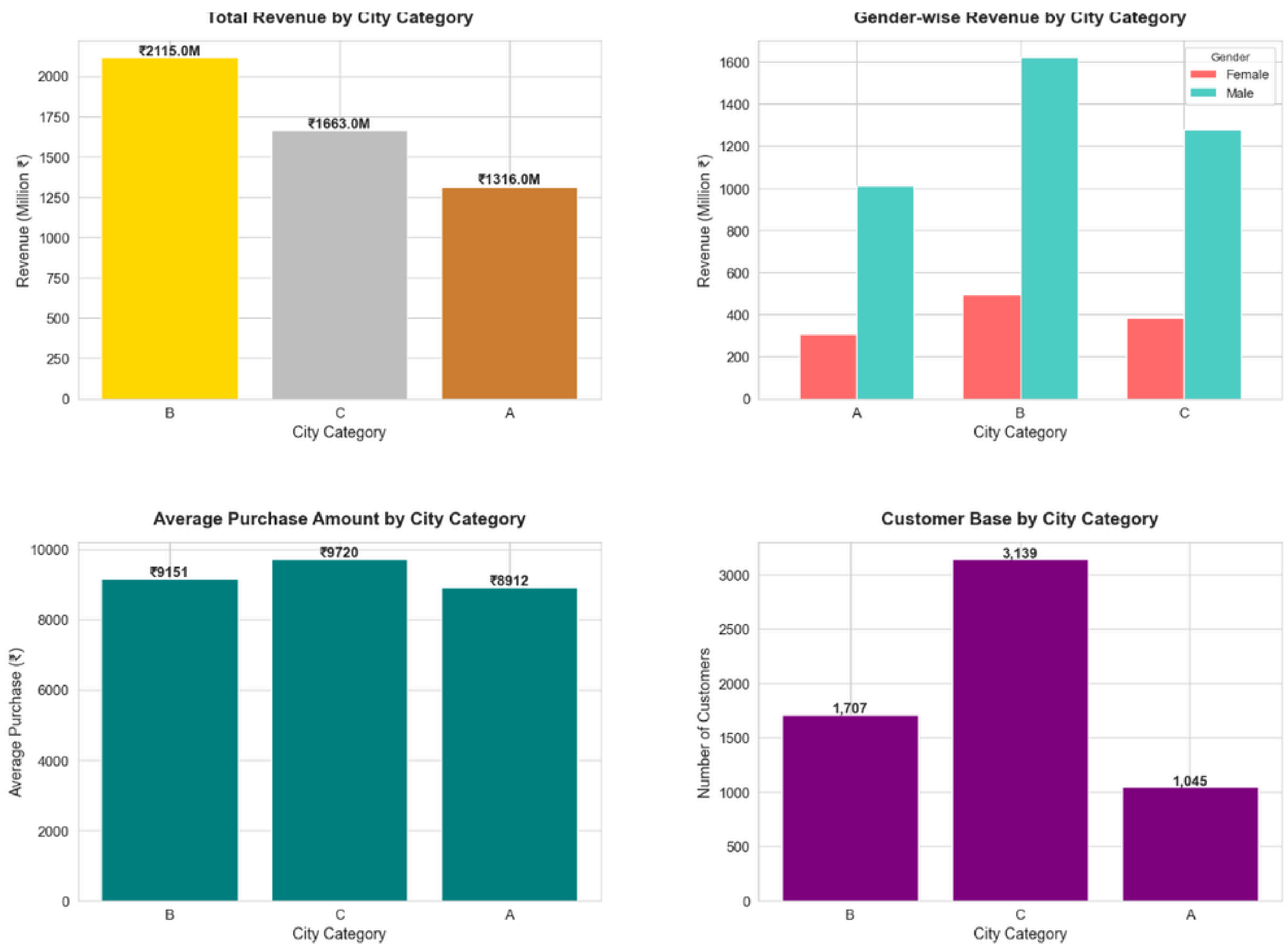
	city_category	Total_Transactions	Total_Revenue_Million	Avg_Purchase	Total_Customers
0	B	231173	2115.0	9151.30	1707
1	C	171175	1663.0	9719.92	3139
2	A	147720	1316.0	8911.94	1045

```
======================================================================
💡 CITY CATEGORY ANALYSIS - KEY BUSINESS INSIGHTS
======================================================================

🏙 CITY REVENUE RANKING:
   City B: ₹2115.0M revenue | 1,707 customers
   City C: ₹1663.0M revenue | 3,139 customers
   City A: ₹1316.0M revenue | 1,045 customers

🏆 KEY FINDINGS:
  • Top Revenue City:    City B → ₹2115.0M
  • Highest Avg Purchase: City C → ₹9,719.92
  • Most Customers:      City C → 3,139 customers

💰 AVERAGE PURCHASE COMPARISON:
   City B: ₹9,151.30 per transaction
   City C: ₹9,719.92 per transaction
   City A: ₹8,911.94 per transaction
```



📊 Analysis 4: Marital Status Impact

Business Question: What is the difference in spending patterns between Married vs Unmarried customers?

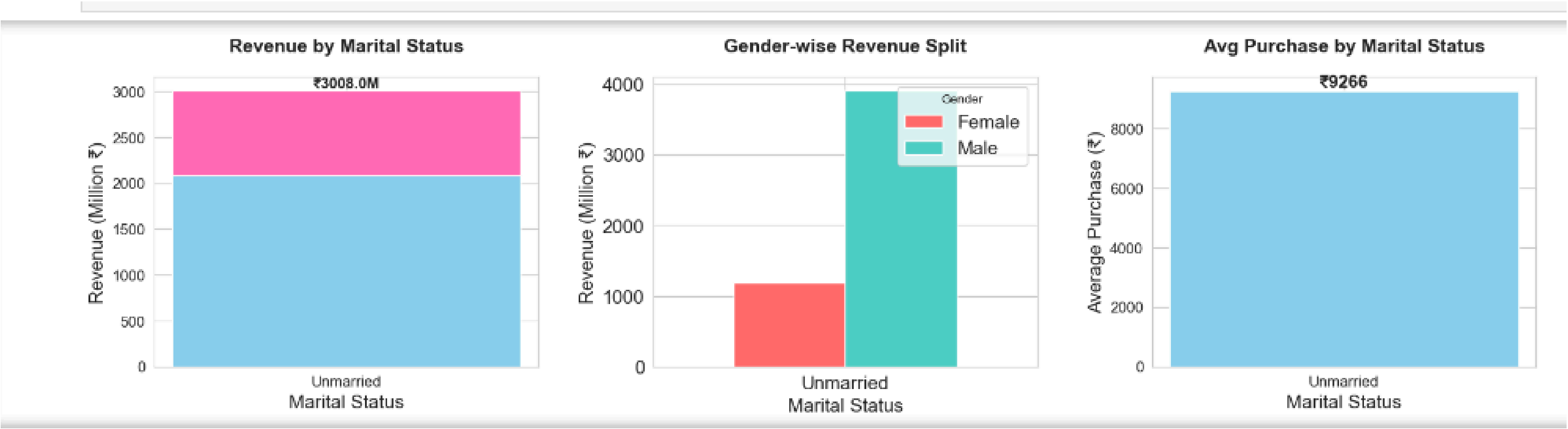
💍 MARITAL STATUS ANALYSIS

==

	Marital_Status	gender	Transactions	Customers	Revenue_Million	Avg_Purchase
0	Unmarried	Male	245910	2470	2324.0	9453.76
1	Unmarried	Male	168349	1755	1584.0	9413.82
2	Unmarried	Female	78821	947	684.0	8679.85
3	Unmarried	Female	56988	719	502.0	8810.25

📊 OVERALL MARITAL STATUS COMPARISON:

	Marital_Status	Total_Transactions	Revenue_Million	Avg_Purchase
0	Unmarried	324731	3008.0	9265.91
1	Unmarried	225337	2086.0	9261.17



💡 MARITAL STATUS ANALYSIS - KEY BUSINESS INSIGHTS

==

💍 REVENUE COMPARISON:

- Unmarried: ₹3008.0M revenue
- Unmarried: ₹2086.0M revenue
- Revenue Difference: ₹922.00M

💰 AVERAGE PURCHASE COMPARISON:

- Unmarried avg purchase: ₹9,265.91
- Unmarried avg purchase: ₹9,261.17
- Difference per transaction: ₹4.74

🏆 KEY FINDINGS:

- Unmarried customers generate MORE revenue
- Transaction volume is higher for Unmarried customers
- Both groups have similar average purchase amounts
- Marital status alone is NOT a strong differentiator

Analysis 5: Product Category Performance

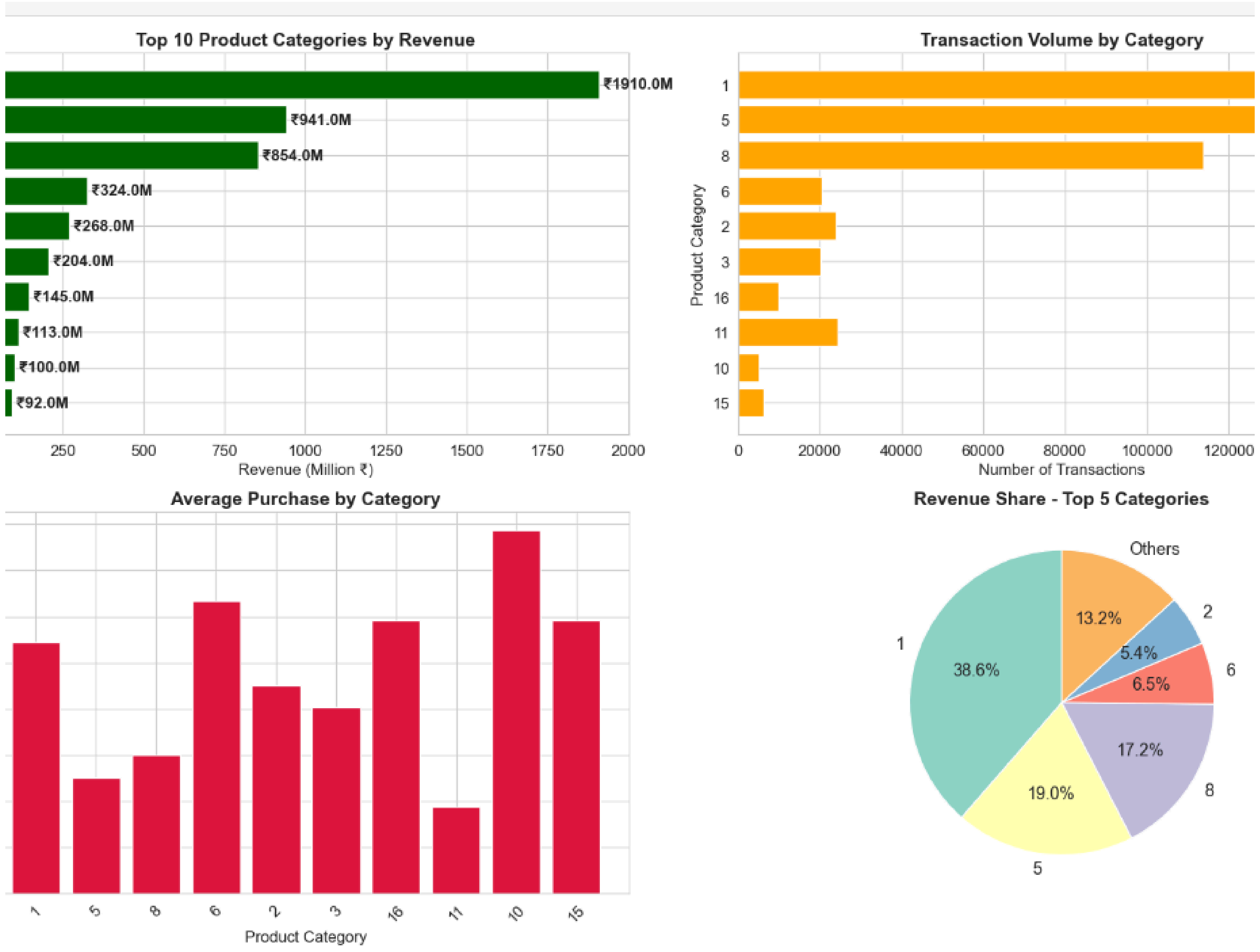
Business Question: Which product categories generate the most revenue?

TOP 10 PRODUCT CATEGORIES BY REVENUE
==

	Category	Transactions	Customers	Revenue_Million	Avg_Purchase
0	1	140378	5767	1910.0	13606.22
1	5	150933	5751	941.0	6240.09
2	8	113925	5659	854.0	7498.96
3	6	20466	4085	324.0	15838.48
4	2	23864	4296	268.0	11251.94
5	3	20213	3838	204.0	10096.71
6	16	9828	3130	145.0	14766.04
7	11	24287	3583	113.0	4685.27
8	10	5125	2328	100.0	19675.57
9	15	6290	2440	92.0	14780.45

GENDER PREFERENCE IN TOP CATEGORIES:

	Category	gender	Transactions	Revenue_Million
0	1	Male	115547	1572.0
1	5	Male	108972	677.0
2	8	Male	80367	602.0
3	1	Female	24831	337.0
4	5	Female	41961	264.0
5	6	Male	15907	253.0
6	8	Female	33558	251.0
7	2	Male	18206	203.0
8	3	Male	14207	142.0
9	16	Male	7426	109.0



```
-----------------------------------------------------------------
💡 PRODUCT CATEGORY ANALYSIS - KEY BUSINESS INSIGHTS
=================================================================

🏆 TOP 5 REVENUE CATEGORIES:
   #1 Category 1.0: ₹1910.0M | Avg: ₹13,606
   #2 Category 5.0: ₹941.0M | Avg: ₹6,240
   #3 Category 8.0: ₹854.0M | Avg: ₹7,499
   #4 Category 6.0: ₹324.0M | Avg: ₹15,838
   #5 Category 2.0: ₹268.0M | Avg: ₹11,252

📊 KEY METRICS:
   • Revenue Leader: Category 1.0 → ₹1910.0M
   • Highest Avg Purchase: Category 10.0 → ₹19,675.57
   • Top 3 categories contribute: 74.8% of total revenue
   • Total transactions in top 10: 515,309

💰 PARETO OBSERVATION:
   Top 3 categories = 75% of revenue (Pareto Principle applies)
```

📊 Analysis 6: Stay Duration Impact

Business Question: How does the number of years residing in current city impact purchase behavior?

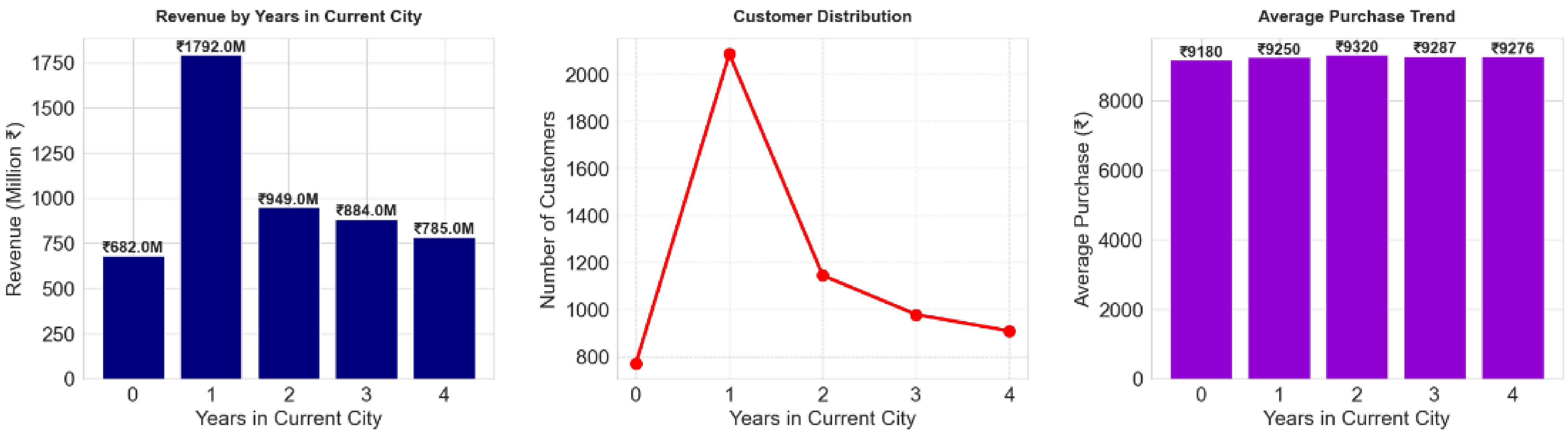
📍 STAY DURATION ANALYSIS

==

	Stay_Duration	Transactions	Customers	Revenue_Million	Avg_Purchase
0	0	74398	772	682.0	9180.08
1	1	193821	2086	1792.0	9250.15
2	2	101838	1145	949.0	9320.43
3	3	95285	979	884.0	9286.90
4	4	84726	909	785.0	9275.60

🏙 STAY DURATION vs CITY CATEGORY:

	Stay_Duration	city_category	Revenue_Million	Avg_Purchase
0	1	B	765.0	9178.99
1	1	C	589.0	9647.04
2	1	A	437.0	8878.65
3	3	B	392.0	9189.99
4	2	B	384.0	9193.01
5	2	C	321.0	9769.95
6	4	B	318.0	9192.62
7	3	C	271.0	9767.69
8	4	C	270.0	9736.61
9	0	B	255.0	8902.62




```
==================================================================
💡 STAY DURATION ANALYSIS - KEY BUSINESS INSIGHTS
==================================================================

📍 REVENUE BY YEARS IN CITY:
   0.0 years: ₹682.0M | 772.0 customers | Avg: ₹9,180
   1.0 years: ₹1792.0M | 2,086.0 customers | Avg: ₹9,250
   2.0 years: ₹949.0M | 1,145.0 customers | Avg: ₹9,320
   3.0 years: ₹884.0M | 979.0 customers | Avg: ₹9,287
   4.0 years: ₹785.0M | 909.0 customers | Avg: ₹9,276

🏆 KEY FINDINGS:
   • Highest Revenue:       1.0 years → ₹1792.0M
   • Highest Avg Purchase:  2.0 years → ₹9,320.43
   • Most Customers:        1.0 years → 2,086.0

📊 LOYALTY INSIGHT:
   • New residents (1 year) tend to shop more — exploring their city
   • Long-stay residents (4+ years) are established buyers
   • Stay duration reflects city familiarity and brand loyalty
```

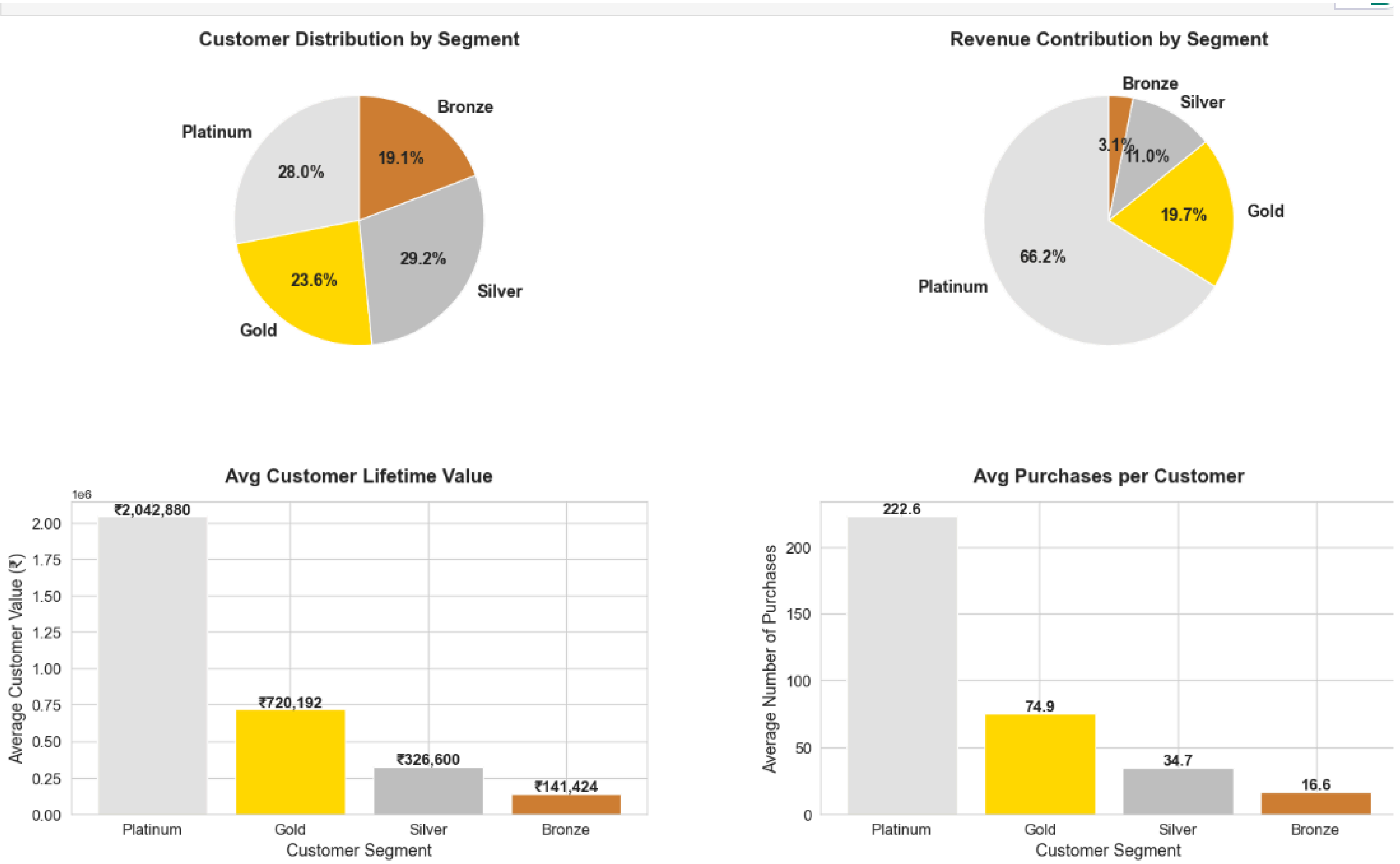
📊 Analysis 7: High-Value Customer Segments

Business Question: Who are the premium customers who spend the most?

💎 CUSTOMER SEGMENTATION ANALYSIS
==
Segment Criteria:
• Platinum: Total Spent >= ₹10,00,000
• Gold: ₹5,00,000 <= Total Spent < ₹10,00,000
• Silver: ₹2,00,000 <= Total Spent < ₹5,00,000
• Bronze: Total Spent < ₹2,00,000

	Customer_Segment	Customer_Count	Total_Revenue_Million	Avg_Customer_Value	Avg_Purchases_Per_Custom
0	Platinum	1650	3370.0	2042880.38	222.
1	Gold	1393	1003.0	720191.91	74.
2	Silver	1723	562.0	326599.69	34.
3	Bronze	1125	159.0	141423.57	16.

📊 REVENUE CONTRIBUTION BY SEGMENT:
• Platinum: 66.16% of total revenue
(1,650 customers generating ₹3370.0M)
• Gold: 19.69% of total revenue
(1,393 customers generating ₹1003.0M)
• Silver: 11.03% of total revenue
(1,723 customers generating ₹562.0M)
• Bronze: 3.12% of total revenue
(1,125 customers generating ₹159.0M)



```
======================================================================
💡 CUSTOMER SEGMENTATION - KEY BUSINESS INSIGHTS
======================================================================

💎 SEGMENT BREAKDOWN:
   🔷 Platinum: 1,650 customers → ₹3370.0M (66.16% revenue)
   🥇 Gold:     1,393 customers → ₹1003.0M (19.69% revenue)
   🥈 Silver:   1,723 customers → ₹562.0M (11.03% revenue)
   🥉 Bronze:   1,125 customers → ₹159.0M (3.12% revenue)

🏆 PARETO ANALYSIS (Premium Segment):
   • Platinum + Gold = 51.7% of customers
   • But they generate 85.8% of revenue
   • This confirms the 80-20 rule in customer value

💰 AVERAGE CUSTOMER VALUE:
   • Platinum CLV: ₹2,042,880
   • Gold CLV:     ₹720,192
   • Silver CLV:   ₹326,600
   • Bronze CLV:   ₹141,424
```

📊 **Analysis 8: Combined Demographic Insights**

Business Question: What is the best customer profile based on Age, Gender, and City combination?

🎯 TOP 20 CUSTOMER PROFILES BY REVENUE

==

	gender	age_category	city_category	Transactions	Customers	Revenue_Million	Avg_Purchase
0	Male	Young Adult	B	70147	468	654.0	9326.54
1	Male	Young Adult	A	56254	338	508.0	9030.55
2	Male	Young Adult	C	42434	702	426.0	10052.35
3	Male	Adult	B	36488	237	336.0	9215.27
4	Male	Young Adult	B	31561	237	296.0	9404.65
5	Male	Adult	C	26843	474	269.0	10040.67
6	Male	Young Adult	C	22205	387	219.0	9881.57
7	Male	Young Adult	A	21266	158	192.0	9034.72
8	Female	Young Adult	B	21437	184	183.0	8568.88
9	Male	Adult	A	19512	123	177.0	9089.92
10	Female	Young Adult	A	17491	123	152.0	8701.51
11	Male	Adult	B	14002	88	133.0	9535.18
12	Male	Senior	B	13498	99	128.0	9486.94
13	Male	Adult	C	12143	226	118.0	9720.60
14	Female	Young Adult	C	11824	238	107.0	9056.76
15	Male	Senior	C	10788	194	107.0	9925.19
16	Female	Adult	B	11110	98	97.0	8755.27
17	Female	Young Adult	B	11686	94	93.0	8024.47
18	Male	Senior	C	9401	191	89.0	9530.31
19	Female	Adult	C	8955	182	84.0	9406.43

🏆 HIGHEST REVENUE PROFILE:

Gender: Male

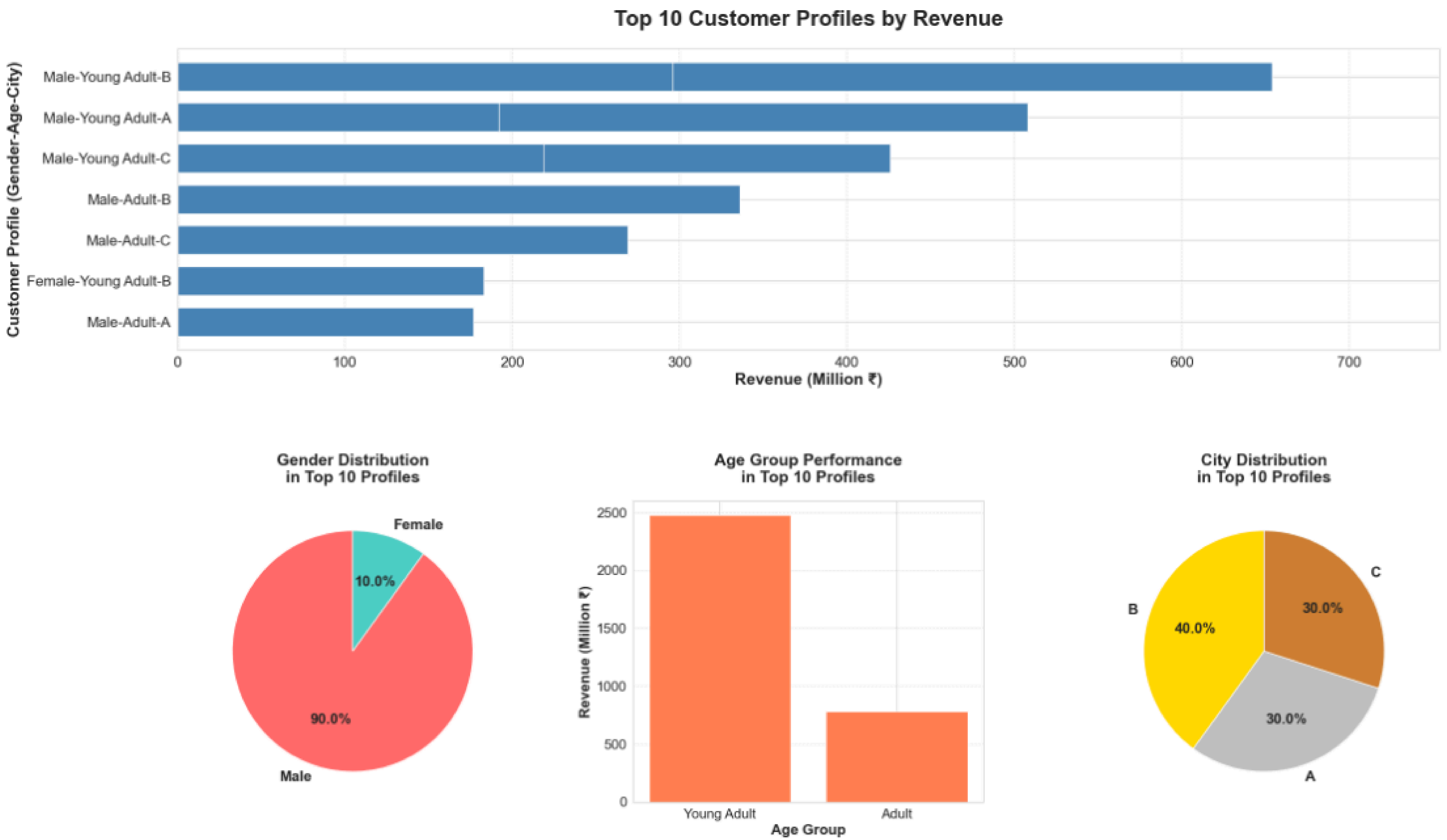
Age: Young Adult

City: B

Revenue: ₹654.0M

Customers: 468

Avg Purchase: ₹9,326.54



Top 10 Customer Profiles by Revenue
Customer Profile (Gender-Age-City)
Male-Young Adult-B
Male-Young Adult-A
Male-Young Adult-C
Male-Adult-B
Male-Adult-C
Female-Young Adult-B
Male-Adult-A
0
100
200
300
400
500
600
700
Revenue (Million ₹)
Gender Distribution in Top 10 Profiles
Female
10.0%
90.0%
Male
Age Group Performance in Top 10 Profiles
Revenue (Million ₹)
2500
2000
1500
1000
500
0
Young Adult
Adult
Age Group
City Distribution in Top 10 Profiles
C
30.0%
B
40.0%
30.0%
A

```
================================================================
💡 COMBINED DEMOGRAPHICS - KEY BUSINESS INSIGHTS
================================================================

🏆 #1 BEST CUSTOMER PROFILE:
   • Gender:   Male
   • Age:      Young Adult
   • City:     B
   • Revenue:  ₹654.0M
   • Customers:468
   • Avg Buy:  ₹9,326.54

📊 TOP 10 PROFILES SUMMARY:
   • Dominant Gender: Male (9 out of 10 profiles)
   • Top Age Group:   Young Adult (highest revenue in top 10)
   • Top City:        B (most profiles in top 10)

🎯 IDEAL TARGET CUSTOMER:
   Gender Male + Age Young Adult + City B
   This combination delivers maximum revenue potential
```

5. Statistical Analysis Using Hypothesis Testing

BUSINESS QUESTION: Do women spend more on Black Friday than men?

```python
import scipy.stats as stats

# 1. Data Preparation
# Extracting purchase amounts for Male and Female customers from the dataframe
male_purchase = df[df['gender'] == 'Male']['purchase']
female_purchase = df[df['gender'] == 'Female']['purchase']

# 2. Defining Hypothesis
# Null Hypothesis (H0): There is no significant difference between the mean spending of Men and Women.
# Alternative Hypothesis (H1): There is a significant difference in the mean spending between Men and Women.

print("--- Statistical Analysis: Independent Two-Sample T-Test ---")

# 3. Performing T-test
# (Used because sample sizes and variances for Men and Women might be different)
t_stat, p_value = stats.ttest_ind(male_purchase, female_purchase, equal_var=False)

print(f"T-statistic: {t_stat:.4f}")
print(f"P-value: {p_value:.4e}")

# 4. Result Interpretation
alpha = 0.05  # 5% Significance Level

if p_value < alpha:
    print("\nResult: Reject the Null Hypothesis (H0).")
    print("Insight: The difference in spending between Men and Women is Statistically Significant.")
else:
    print("\nResult: Fail to reject the Null Hypothesis (H0).")
    print("Insight: There is no strong statistical evidence of a difference in spending.")

# 5. Final Conclusion based on Means
male_mean = male_purchase.mean()
female_mean = female_purchase.mean()

print("-" * 60)
print(f"Average Spend (Male):   ₹{male_mean:.2f}")
print(f"Average Spend (Female): ₹{female_mean:.2f}")

if female_mean > male_mean and p_value < alpha:
    print("\nConclusion: Yes, women spend significantly more than men.")
elif male_mean > female_mean and p_value < alpha:
    print("\nConclusion: No, men actually spend significantly more than women.")
else:
    print("\nConclusion: There is no significant difference in spending based on gender.")
```

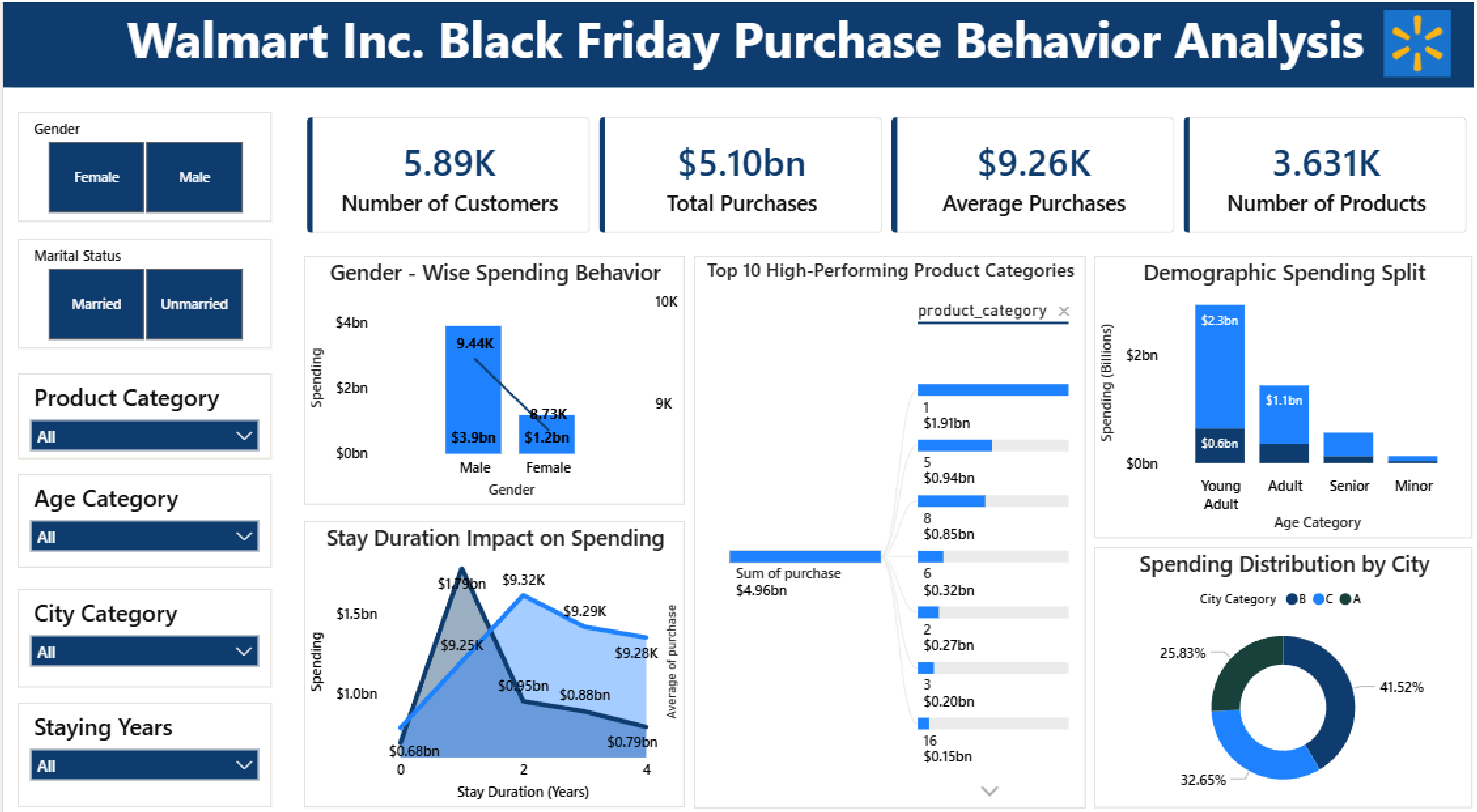
--- Statistical Analysis: Independent Two-Sample T-Test ---
T-statistic: 46.3582
P-value: 0.0000e+00

Result: Reject the Null Hypothesis (H0).
Insight: The difference in spending between Men and Women is Statistically Significant.
--
Average Spend (Male):   ₹9437.53
Average Spend (Female): ₹8734.57

Conclusion: No, men actually spend significantly more than women.

6. Dashboard in Power BI

Finally, I Built an interactive Dashboard in Power BI



7. Business Recommendations

- **High-Value Targeting:** Male customers dominate revenue, so focus on high-ticket campaigns for them. Simultaneously, launch gender-specific loyalty programs to increase female customer acquisition and transaction frequency to balance the revenue risk.
- **Strategic Focus:** Focus the marketing budget and loyalty programs on the "Young Adult" (26-35) segment to maximize revenue, while creating age-specific product bundles. Simultaneously, implement entry-level offers for younger customers (18-25) and target the Senior demographic with premium, high-value product offerings.
- **Segmented Targeting:** We should promote lifestyle, fashion, and tech gadgets to unmarried professionals, while focusing on home and family-oriented products for married customers.
- **Inventory Optimization:** Increase stock for Category 1.0 — revenue leader. Prioritize Category 10.0 for premium positioning. Focus promotions on top 3 categories for max ROI. Bundle top categories with lower-performing ones. Seasonal offers on high-volume transaction.
- **Customer Lifecycle Strategy:** We should target new residents (1 year) with welcome discounts via real estate partnerships, while retaining long-term residents (4+ years) through exclusive loyalty programs.
- **Engagement & Growth:** For mid-stay customers (2-3 years), we should focus on upselling and location-based promotions to increase their lifetime value and brand commitment.
- **Reward Top Shoppers:** We should spend 60% of our budget on Platinum and Gold customers by giving them VIP rewards and early sale access to keep them loyal.
- **Grow Small Spenders:** For Silver and Bronze customers, we should use special discounts and "win-back" offers to encourage them to spend more and move up to higher tiers.

- **Grow Small Spenders:** For Silver and Bronze customers, we should use special discounts and "win-back" offers to encourage them to spend more and move up to higher tiers.

Combined Demographics Analysis

1. Primary Target: Male customers aged Young Adult in City B
2. Design hyper-targeted campaigns for top profile combination
3. Use this profile for look-alike audience marketing
4. Personalized product recommendations for top segments
5. Allocate 50% of ad budget to top 3 profile combinations
6. Expand customer base in underperforming profile combinations

THANK YOU