CodeBasic Resume Challenge 8

Provide Insights to the Product Strategy Team in the Banking Domain

---

Problem Statement

Mitron Bank is a legacy financial institution headquartered in Hyderabad. They want to introduce a new line of credit cards, aiming to broaden its product offerings and reach in the financial market.

AtliQ Data Services came to know about this through an internal link and approached Mitron Bank with a proposal to implement this project. However, strategy director of Mitron Bank, Mr.Bashnir Rover is skeptical and asked them to do a pilot project with the sample data before handing them the full project. They provided a sample dataset of 4000 customers across five cities on their online spend and other details.

Peter Pandey is a data analyst at AtliQ Data Services and asked by his manager to take over this project. His role is to analyse the provided sample data and report key findings to the strategy team of Mitron Bank. This analysis is expected to guide them in tailoring the credit cards to customer needs and market trends.

The successful acquisition of this project depends on Peter's ability to provide actionable, data-driven recommendations and impress Mr. Bashnir Rover & his team. Peter requested support from his manager Tony Sharma, and he provided him with some ideas to generate insights based on the data provided.

This file contains all the meta information regarding the columns described in the CSV files. We have provided 2 CSV files:

1. dim_customers
2. fact_spends

Column Description for dim_customers:

- customer_id: This column represents the Unique ID assigned to each customer.
- gender: This column represents the gender of the customer. (Male, Female)
- age_group: This column categorizes the customer into different age groups. (21-24, 25-34, 35-45, 45+)
- marital_status: This column indicates the marital status of the customer (single, married).
- city: This column represents the city of residence for the customer. (Mumbai, Delhi-NCR, Chennai, Hyderabad, Bengaluru)
- occupation: This column denotes the occupation or profession of the customer. (Salaried IT Employees, Salaried Other Employees, Business Owners, Freelancers, Government Employees)
- average_income: This column indicates the monthly average income of the customer, in INR currency.

Column Description for fact_spends:

- customer_id: This column represents the Unique ID of each customer, linking to the dim_customer table.
- month: This column indicates the month in which the spending was recorded. (May, June, July, August, September, October)
- category: This column describes the category of spending (Entertainment, Apparel, Electronics, etc).
- payment_type: This column specifies the type of payment used by the customer (Debit Card, Credit Card, UPI, Net Banking).
- spends: This column shows the total amount spent by the customer in the specified month, category and payment_type.

Data Loading and Merging:

Load the dim_customers and fact_spends CSV files into a data analysis tool (e.g., Python with pandas).

Merge the two datasets using the "customer_id" column as the common identifier

In [1]:
```
import pandas as pd

# Load data
dim_customers = pd.read_csv(r"C:\Users\HP\Desktop\Codebasic\challenge 8\dim_customers.csv")
fact_spends = pd.read_csv(r"C:\Users\HP\Desktop\Codebasic\challenge 8\fact_spends.csv")

# Merge datasets
merged_data = pd.merge(fact_spends, dim_customers, on="customer_id")
```

In [2]:
```
merged_data.head()
```

Out[2]:

	customer_id	month	category	payment_type	spend	age_group	city	occupation	gender	marital status	avg_income
0	ATQCUS1371	July	Health & Wellness	Credit Card	1114	35-45	Chennai	Salaried IT Employees	Male	Married	64696
1	ATQCUS1371	May	Bills	UPI	787	35-45	Chennai	Salaried IT Employees	Male	Married	64696
2	ATQCUS1371	September	Bills	Net Banking	1022	35-45	Chennai	Salaried IT Employees	Male	Married	64696
3	ATQCUS1371	September	Bills	UPI	1942	35-45	Chennai	Salaried IT Employees	Male	Married	64696
4	ATQCUS1371	August	Electronics	Credit Card	1865	35-45	Chennai	Salaried IT Employees	Male	Married	64696

In [3]:
```
merged_data.isnull()
```

Out[3]:

	customer_id	month	category	payment_type	spend	age_group	city	occupation	gender	marital status	avg_income
0	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...
863995	False	False	False	False	False	False	False	False	False	False	False
863996	False	False	False	False	False	False	False	False	False	False	False
863997	False	False	False	False	False	False	False	False	False	False	False
863998	False	False	False	False	False	False	False	False	False	False	False
863999	False	False	False	False	False	False	False	False	False	False	False

864000 rows × 11 columns

In [4]:
```
merged_data.isnull().sum()
```

Out[4]:
```
customer_id       0
month             0
category          0
payment_type      0
spend             0
age_group         0
city              0
occupation        0
gender            0
marital status    0
avg_income        0
dtype: int64
```

In [5]:
```
print(dim_customers)
```

```
     customer_id age_group       city                occupation  gender  \
0      ATQCUS1825       45+  Bengaluru     Salaried IT Employees    Male
1      ATQCUS0809     25-34  Hyderabad  Salaried Other Employees    Male
2      ATQCUS0663     25-34    Chennai  Salaried Other Employees    Male
3      ATQCUS0452     25-34  Delhi NCR      Government Employees    Male
4      ATQCUS3350     21-24  Bengaluru                Freelancers    Male
...           ...       ...        ...                       ...     ...
3995   ATQCUS3035       45+  Delhi NCR           Business Owners  Female
3996   ATQCUS2585     35-45     Mumbai  Salaried Other Employees  Female
3997   ATQCUS1229     35-45  Bengaluru     Salaried IT Employees    Male
3998   ATQCUS0581     25-34  Bengaluru      Government Employees    Male
3999   ATQCUS3477     25-34     Mumbai           Business Owners    Male

     marital status  avg_income
0           Married       73523
1           Married       39922
2           Married       37702
3           Married       54090
4            Single       28376
...             ...         ...
3995        Married       72805
3996        Married       41343
3997        Married       65948
3998        Married       52589
3999         Single       73541

[4000 rows x 7 columns]
```

In [6]:
```
print(fact_spends)
```

```
       customer_id      month           category payment_type  spend
0       ATQCUS1371       July  Health & Wellness  Credit Card   1114
1       ATQCUS0368    October          Groceries  Credit Card   1466
2       ATQCUS0595        May  Health & Wellness  Credit Card    387
3       ATQCUS0667    October        Electronics  Credit Card   1137
4       ATQCUS3477  September              Bills          UPI   2102
...            ...        ...                ...          ...    ...
863995  ATQCUS1993       June              Bills   Debit Card    897
863996  ATQCUS1063  September              Bills  Credit Card   2680
863997  ATQCUS0416     August             Others  Credit Card    270
863998  ATQCUS3361  September              Bills          UPI    446
863999  ATQCUS1736  September            Apparel          UPI    242

[864000 rows x 5 columns]
```

In [7]:
```
print(merged_data)
```

```
       customer_id      month           category payment_type  spend  \
0       ATQCUS1371       July  Health & Wellness  Credit Card   1114
1       ATQCUS1371        May              Bills          UPI    787
2       ATQCUS1371  September              Bills  Net Banking   1022
3       ATQCUS1371  September              Bills          UPI   1942
4       ATQCUS1371     August        Electronics  Credit Card   1865
...            ...        ...                ...          ...    ...
863995  ATQCUS0890    October          Groceries  Net Banking    764
863996  ATQCUS0890  September  Health & Wellness   Debit Card   1188
863997  ATQCUS0890  September        Electronics          UPI   2786
863998  ATQCUS0890     August          Groceries   Debit Card   2593
863999  ATQCUS0890     August          Groceries          UPI   3074

       age_group     city             occupation gender marital status  \
0          35-45  Chennai  Salaried IT Employees   Male        Married
1          35-45  Chennai  Salaried IT Employees   Male        Married
2          35-45  Chennai  Salaried IT Employees   Male        Married
3          35-45  Chennai  Salaried IT Employees   Male        Married
4          35-45  Chennai  Salaried IT Employees   Male        Married
...          ...      ...                    ...    ...           ...
863995     35-45   Mumbai  Salaried IT Employees   Male        Married
863996     35-45   Mumbai  Salaried IT Employees   Male        Married
863997     35-45   Mumbai  Salaried IT Employees   Male        Married
863998     35-45   Mumbai  Salaried IT Employees   Male        Married
863999     35-45   Mumbai  Salaried IT Employees   Male        Married

        avg_income
0            64696
1            64696
2            64696
3            64696
4            64696
...            ...
863995       65187
863996       65187
863997       65187
863998       65187
863999       65187

[864000 rows x 11 columns]
```

Data Cleaning:

Check for missing values and handle them appropriately.

Look for any outliers or anomalies in the data.

In [8]:
```
# Check for missing values
print(merged_data.isnull().sum())
```

```
customer_id       0
month             0
category          0
payment_type      0
spend             0
age_group         0
city              0
occupation        0
gender            0
marital status    0
avg_income        0
dtype: int64
```

In [9]:
```
# Handle missing values if needed
merged_data = merged_data.dropna()

# Check for outliers
# Handle outliers if needed
```

In [10]:
```
merged_data
```

Out[10]:

	customer_id	month	category	payment_type	spend	age_group	city	occupation	gender	marital status	avg_income
0	ATQCUS1371	July	Health & Wellness	Credit Card	1114	35-45	Chennai	Salaried IT Employees	Male	Married	64696
1	ATQCUS1371	May	Bills	UPI	787	35-45	Chennai	Salaried IT Employees	Male	Married	64696
2	ATQCUS1371	September	Bills	Net Banking	1022	35-45	Chennai	Salaried IT Employees	Male	Married	64696
3	ATQCUS1371	September	Bills	UPI	1942	35-45	Chennai	Salaried IT Employees	Male	Married	64696
4	ATQCUS1371	August	Electronics	Credit Card	1865	35-45	Chennai	Salaried IT Employees	Male	Married	64696
...	...	...	...	...	...	...	...	...	...	...	...
863995	ATQCUS0890	October	Groceries	Net Banking	764	35-45	Mumbai	Salaried IT Employees	Male	Married	65187
863996	ATQCUS0890	September	Health & Wellness	Debit Card	1188	35-45	Mumbai	Salaried IT Employees	Male	Married	65187
863997	ATQCUS0890	September	Electronics	UPI	2786	35-45	Mumbai	Salaried IT Employees	Male	Married	65187
863998	ATQCUS0890	August	Groceries	Debit Card	2593	35-45	Mumbai	Salaried IT Employees	Male	Married	65187
863999	ATQCUS0890	August	Groceries	UPI	3074	35-45	Mumbai	Salaried IT Employees	Male	Married	65187

864000 rows × 11 columns

Exploratory Data Analysis (EDA):

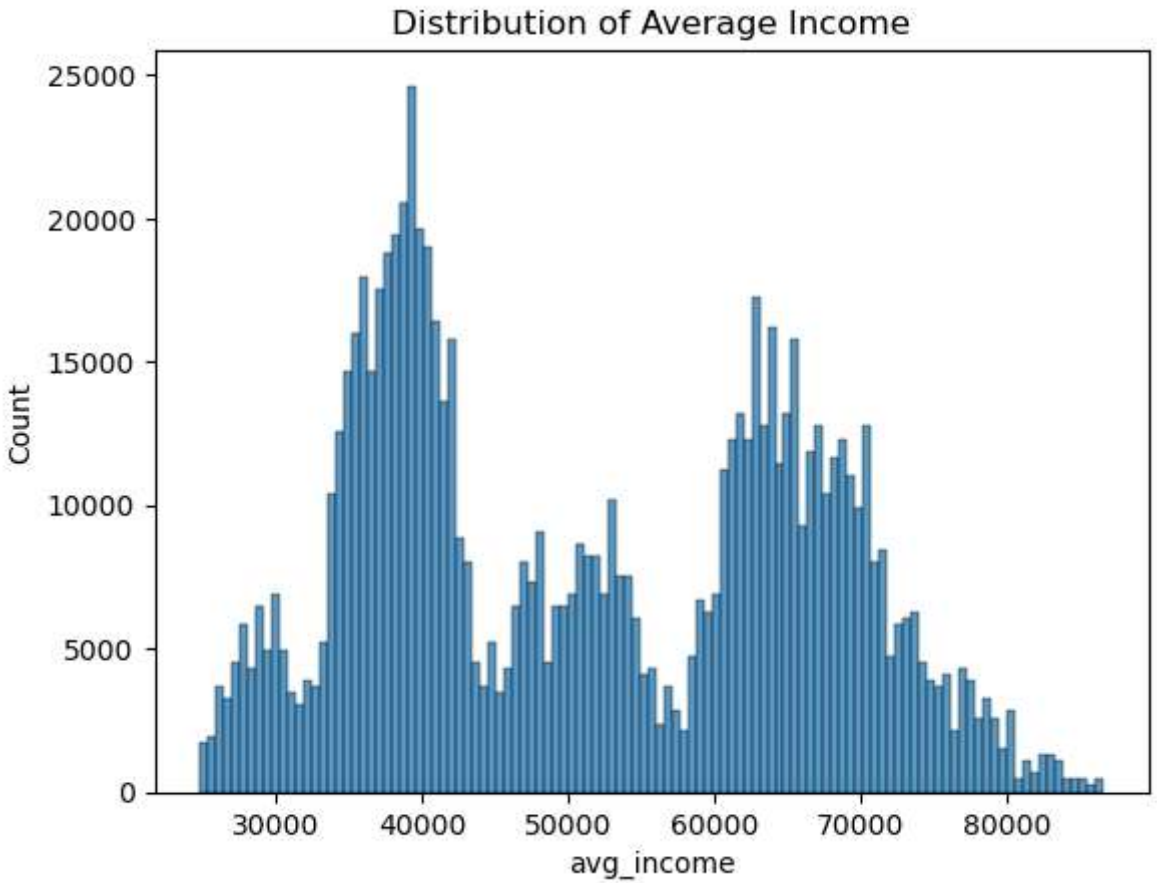
Explore basic statistics and visualizations to understand the distribution of key variables

In [11]:
```
# Basic statistics
print(merged_data.describe())
```

```
              spend     avg_income
count  864000.000000  864000.000000
mean      614.464994   51657.032250
std       661.571676   14688.312762
min         6.000000   24816.000000
25%       191.000000   38701.000000
50%       395.000000   50422.000000
75%       793.000000   64773.250000
max     10313.000000   86600.000000
```

In [12]:

```
# Visualizations
# Example: Histogram of average_income
import matplotlib.pyplot as plt
import seaborn as sns

sns.histplot(merged_data['avg_income'])
plt.title('Distribution of Average Income')
plt.show()
```



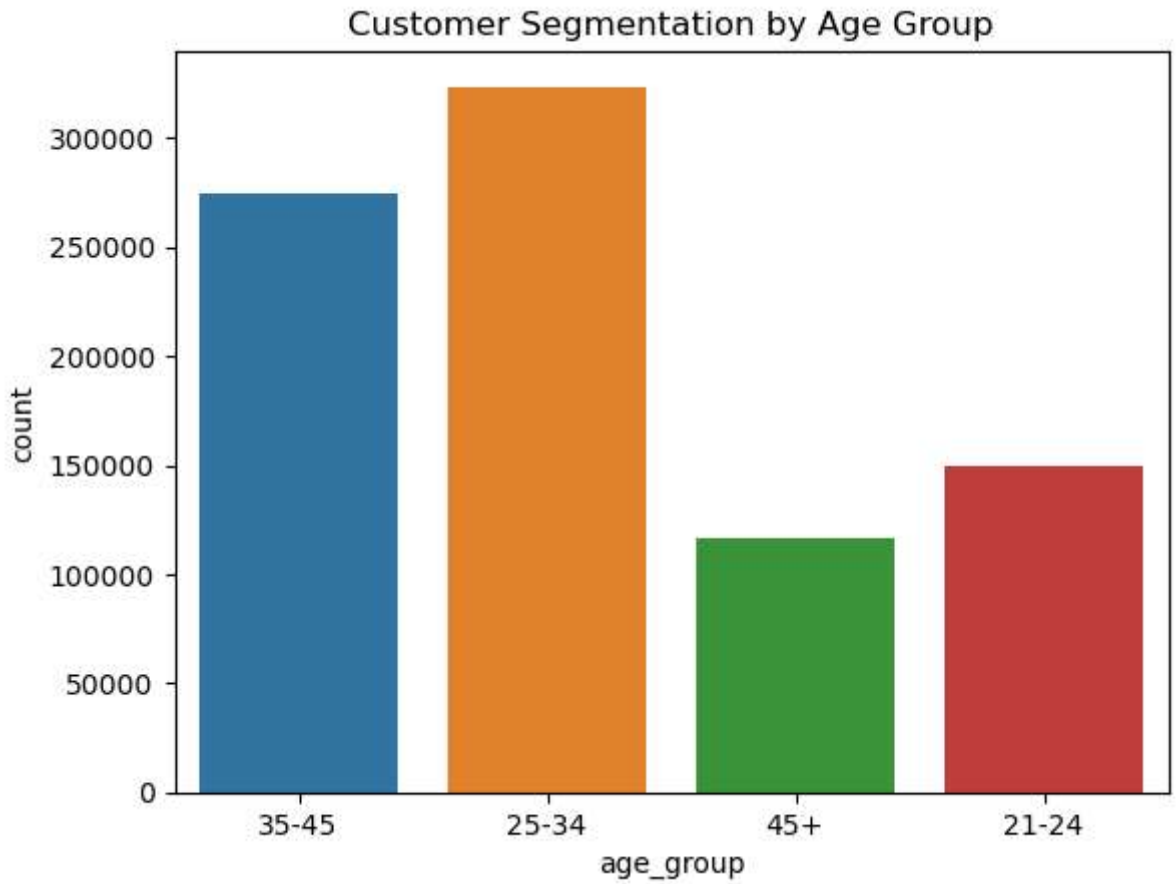
Customer Segmentation:

Analyze customer segments based on demographics.

In [13]:
```
# Example: Customer segmentation by age group
age_group_counts = merged_data['age_group'].value_counts()
print(age_group_counts)
```

```
25-34    323568
35-45    274968
21-24    149256
45+      116208
Name: age_group, dtype: int64
```

In [14]:
```
# Visualize the distribution of age groups
sns.countplot(x='age_group', data=merged_data)
plt.title('Customer Segmentation by Age Group')
plt.show()
```



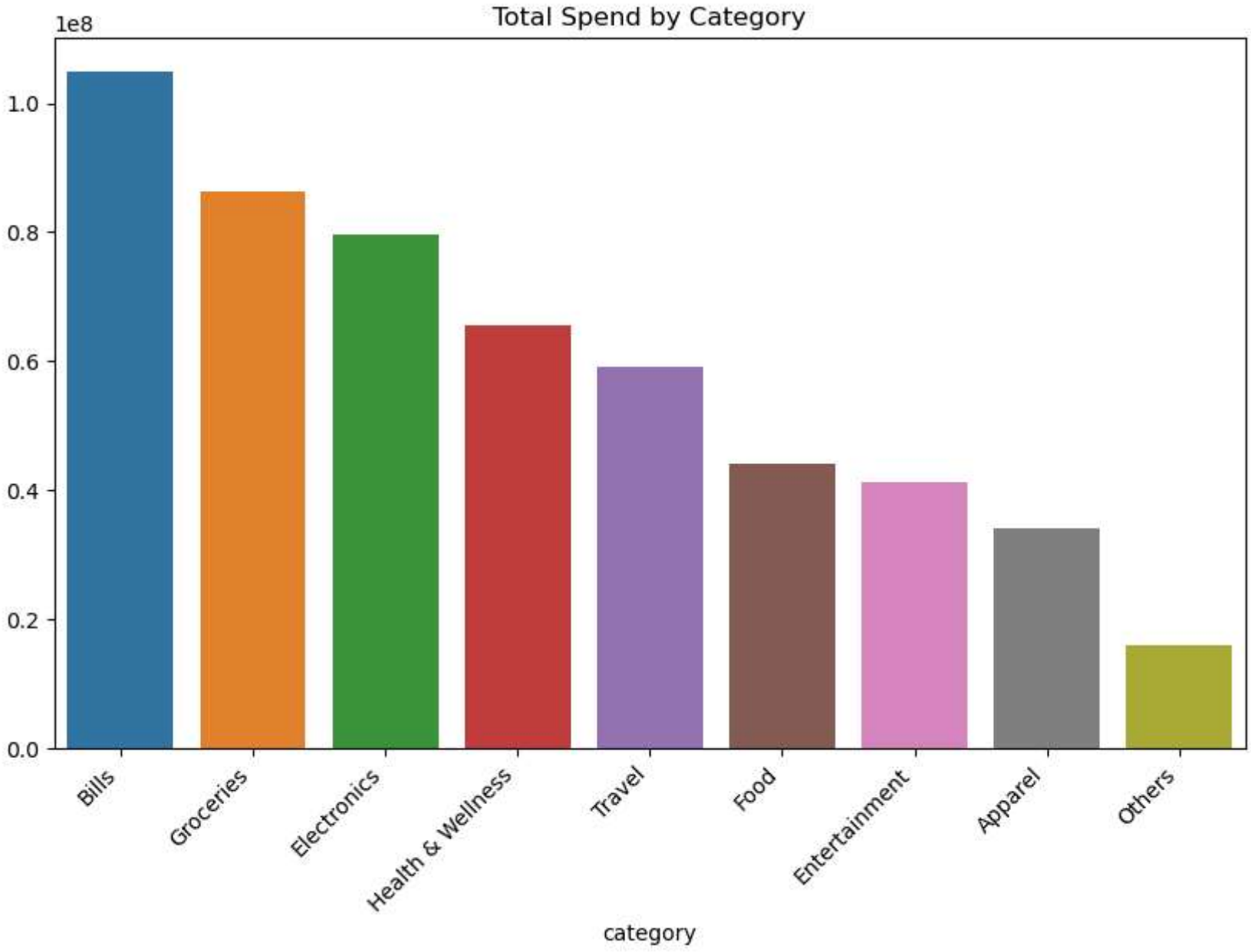
Online Spend Analysis:

Explore spending patterns, popular categories, and payment types.

In [15]:
```
# Example: Total spend by category
category_spend = merged_data.groupby('category')['spend'].sum().sort_values(ascending=False)
print(category_spend)
```

```
category
Bills                104912768
Groceries             86303761
Electronics           79562220
Health & Wellness     65599867
Travel                59223324
Food                  44013470
Entertainment         41289162
Apparel               34036001
Others                15957182
Name: spend, dtype: int64
```

In [16]:
```
# Visualize spending by category
plt.figure(figsize=(10, 6))
sns.barplot(x=category_spend.index, y=category_spend.values)
plt.title('Total Spend by Category')
plt.xticks(rotation=45, ha='right')
plt.show()
```



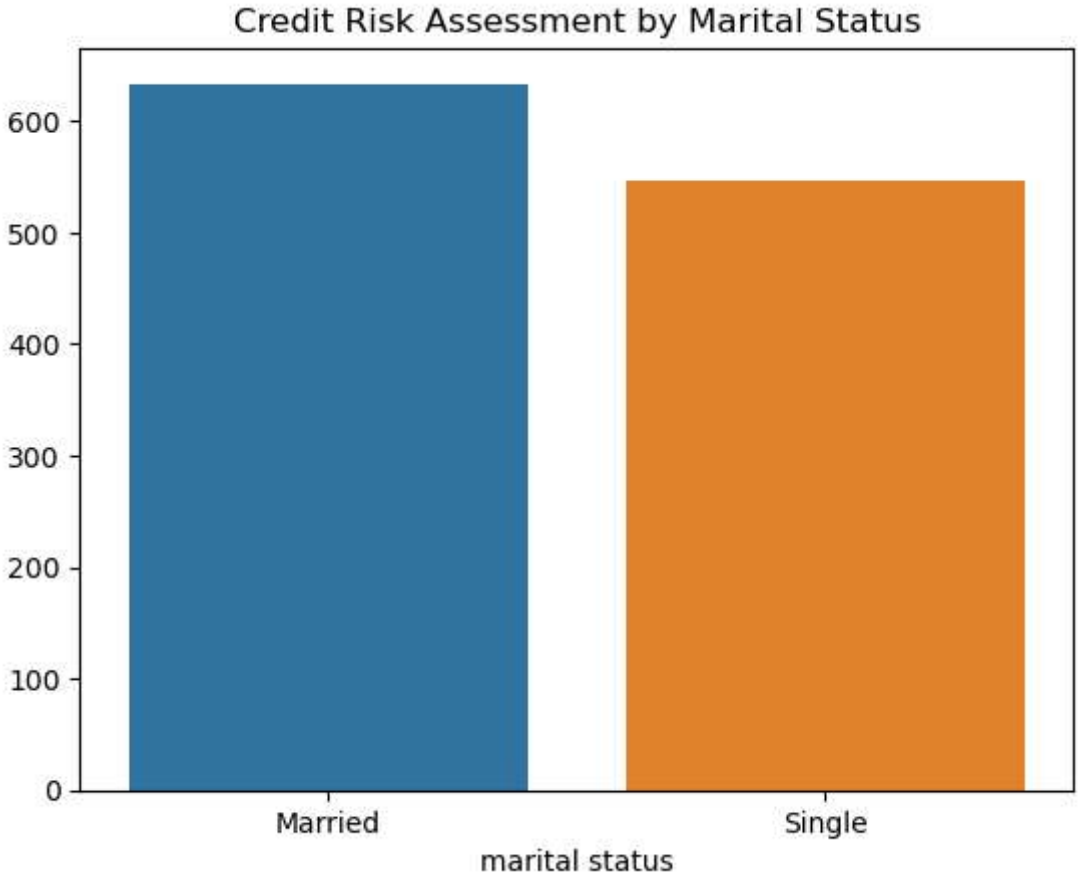
Credit Risk Assessment:

Analyze spending patterns and financial behavior to assess credit risk.

In [17]:
```
# Example: Credit risk assessment
credit_risk_analysis = merged_data.groupby('marital status')['spend'].mean()
print(credit_risk_analysis)
```

```
marital status
Married    633.369905
Single     545.847169
Name: spend, dtype: float64
```

In [18]:
```
# Visualize credit risk by marital status
sns.barplot(x=credit_risk_analysis.index, y=credit_risk_analysis.values)
plt.title('Credit Risk Assessment by Marital Status')
plt.show()
```



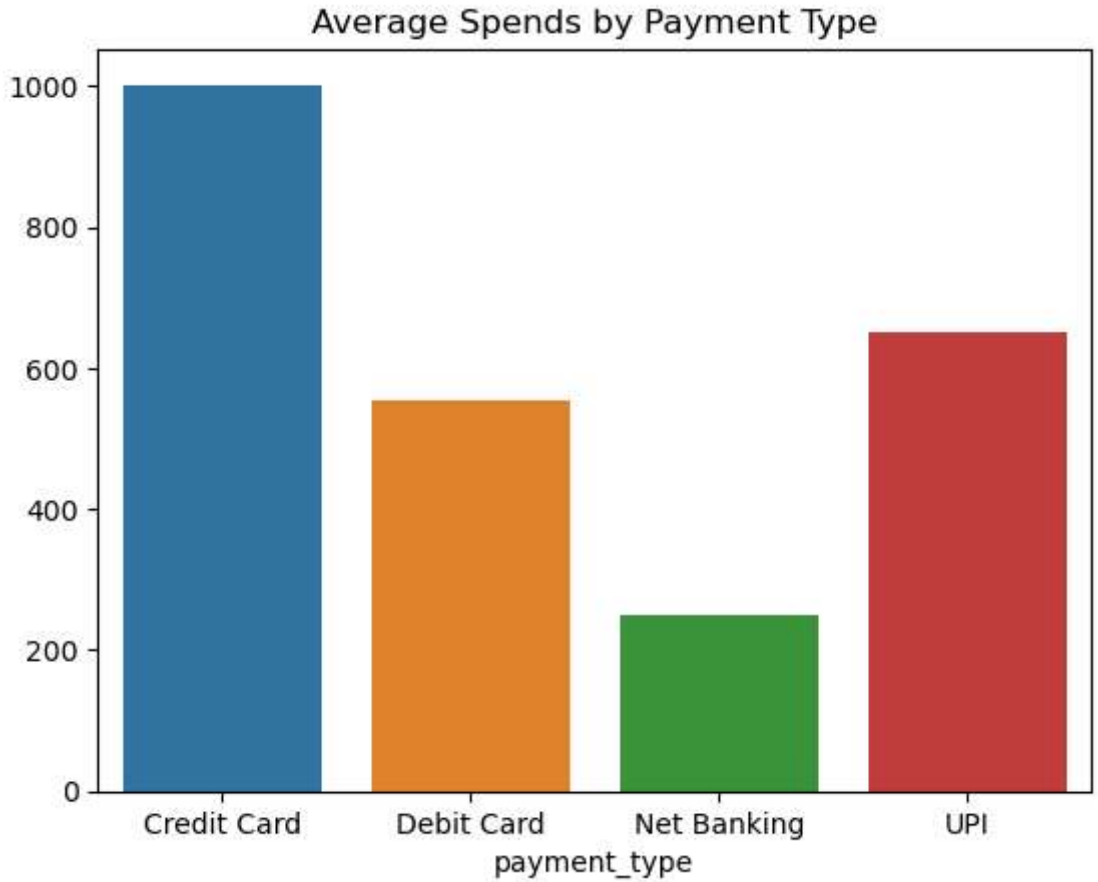
Fraud Detection:

Implement fraud detection analysis to identify irregularities in spending patterns.

In [19]:
```
# Example: Fraud detection analysis
fraud_detection = merged_data.groupby('payment_type')['spend'].mean()
print(fraud_detection)
```

```
payment_type
Credit Card    1001.429968
Debit Card      553.505690
Net Banking     250.964954
UPI             651.959366
Name: spend, dtype: float64
```

In [20]:
```
# Visualize average spends by payment type
sns.barplot(x=fraud_detection.index, y=fraud_detection.values)
plt.title('Average Spends by Payment Type')
plt.show()
```



Demographic Classification:

Classify customers based on demographics such as age group, gender, occupation, etc.

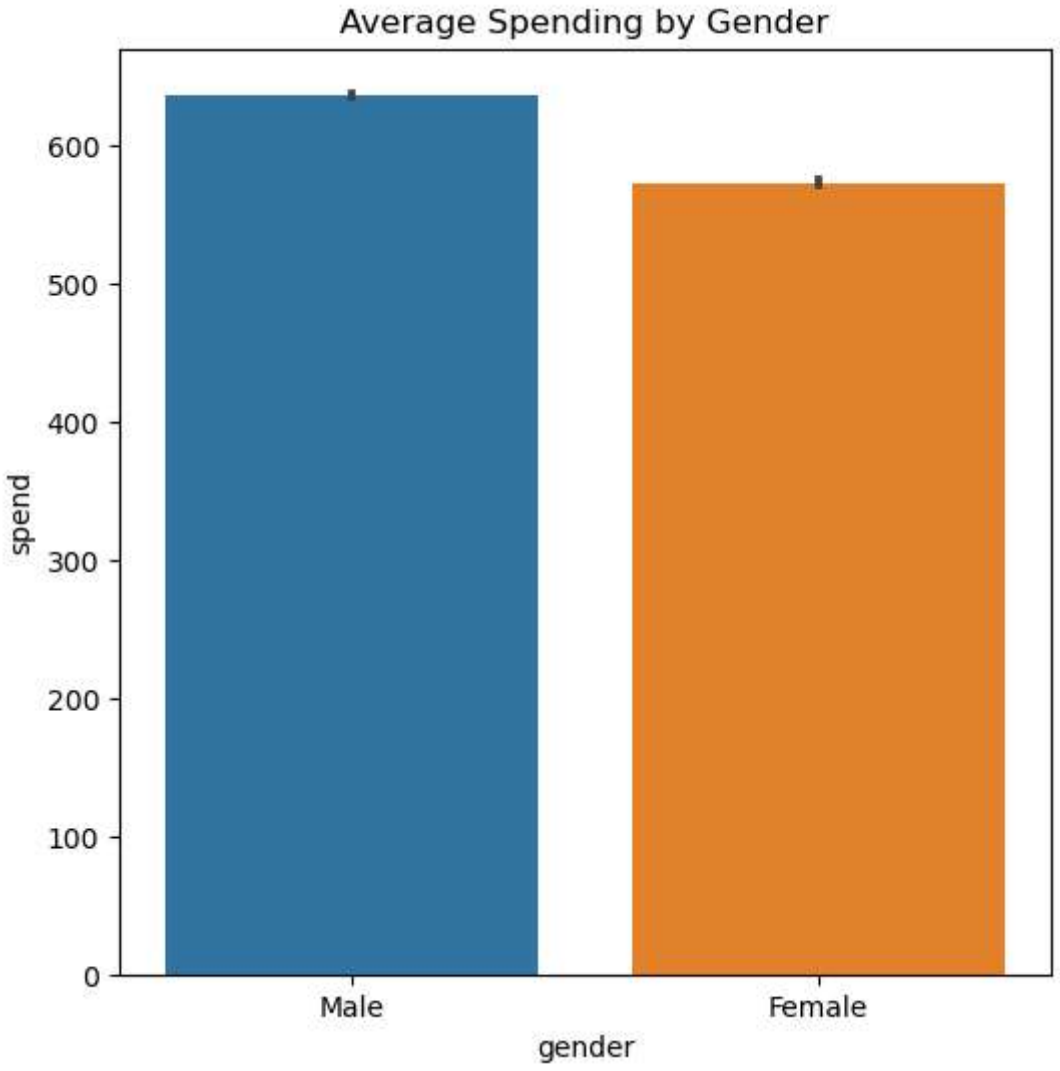
Provide insights based on these classifications.

In [21]:
```
# Demographic classification
demographic_classification = merged_data.groupby(['age_group', 'gender', 'occupation'])['spend'].mean()
print(demographic_classification)

# Visualize average spends by demographics
# Add appropriate visualizations based on your preference
```

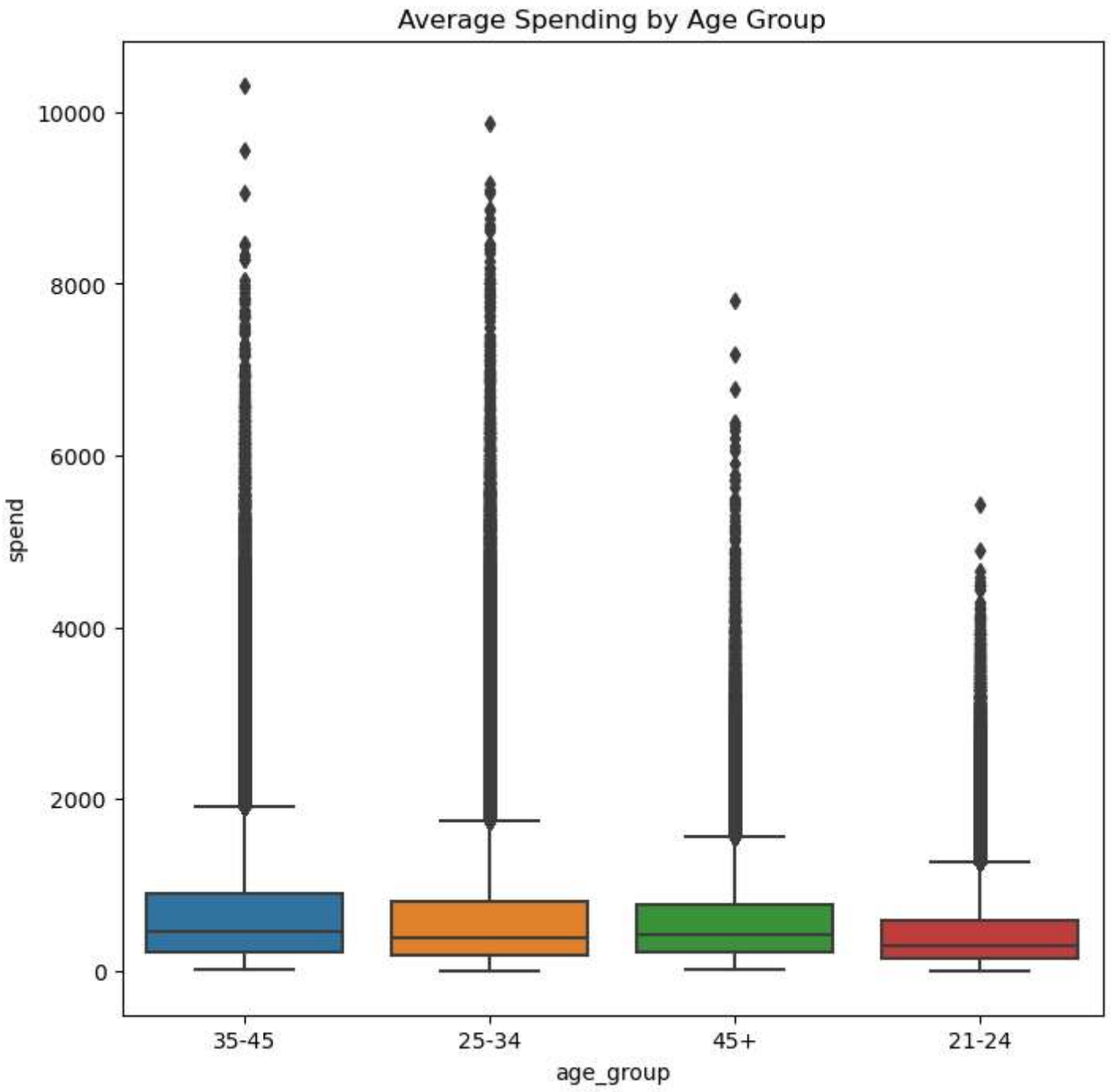
```
age_group  gender  occupation
21-24      Female  Business Owners             415.789760
                   Freelancers                 286.132440
                   Government Employees        267.299576
                   Salaried IT Employees       588.093630
                   Salaried Other Employees    297.618754
           Male    Business Owners             481.056056
                   Freelancers                 331.443496
                   Government Employees        318.295635
                   Salaried IT Employees       672.423004
                   Salaried Other Employees    342.462126
25-34      Female  Business Owners             576.653935
                   Freelancers                 408.448567
                   Government Employees        391.239888
                   Salaried IT Employees       824.007663
                   Salaried Other Employees    424.176522
           Male    Business Owners             691.181040
                   Freelancers                 469.893429
                   Government Employees        470.321288
                   Salaried IT Employees       972.085963
                   Salaried Other Employees    499.457321
35-45      Female  Business Owners             641.752170
                   Freelancers                 435.527778
                   Government Employees        412.422743
                   Salaried IT Employees       910.533200
                   Salaried Other Employees    460.729911
           Male    Business Owners             767.384033
                   Freelancers                 509.197259
                   Government Employees        499.760665
                   Salaried IT Employees      1070.949597
                   Salaried Other Employees    543.605771
45+        Female  Business Owners             538.420833
                   Freelancers                 375.298246
                   Government Employees        380.106019
                   Salaried IT Employees       740.472295
                   Salaried Other Employees    380.583995
           Male    Business Owners             643.397705
                   Freelancers                 440.517620
                   Government Employees        433.271655
                   Salaried IT Employees       889.976852
                   Salaried Other Employees    463.760732
Name: spend, dtype: float64
```

In [22]:
```
# Assuming merged_data is the DataFrame containing the relevant data
```

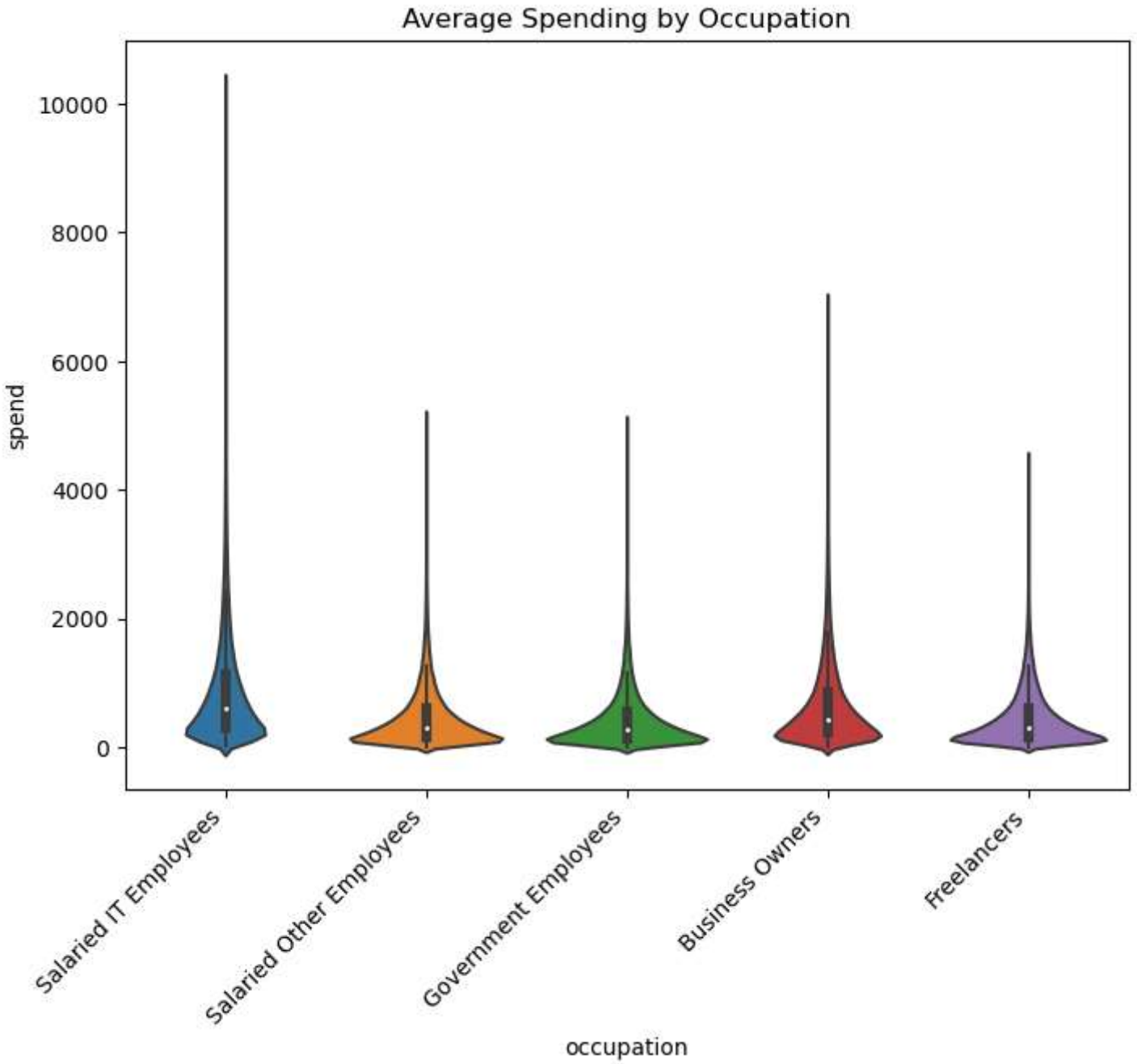
```
# Bar plot for average spending by gender
plt.figure(figsize=(6, 6))
sns.barplot(x='gender', y='spend', data=merged_data, estimator='mean')
plt.title('Average Spending by Gender')
plt.show()
```



In [23]:
```
# Box plot for average spending by age group
plt.figure(figsize=(8, 8))
sns.boxplot(x='age_group', y='spend', data=merged_data)
plt.title('Average Spending by Age Group')
plt.show()
```

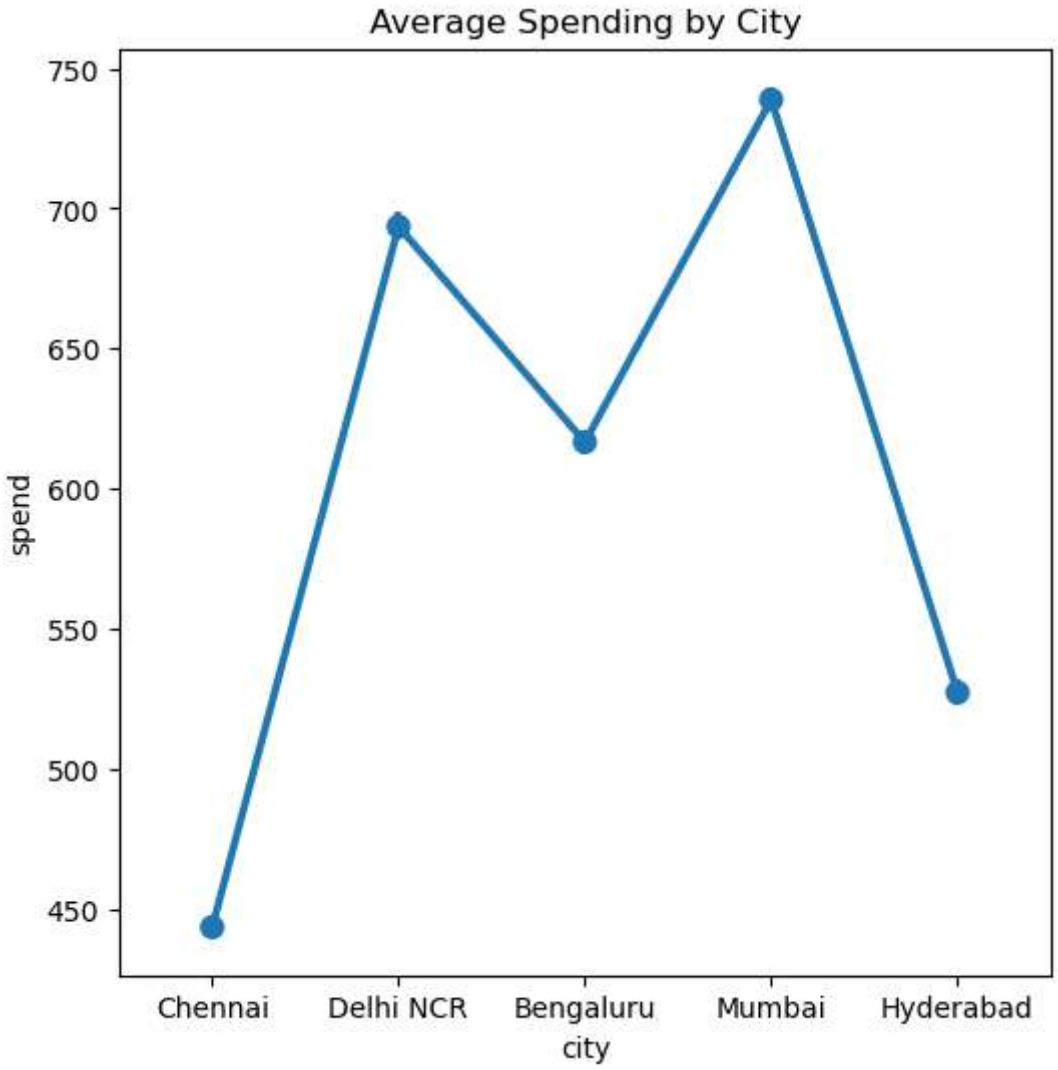

Average Spending by Age Group
10000
8000
6000
4000
2000
0
spend
35-45
25-34
45+
21-24
age_group

In [24]:
```
# Violin plot for average spending by occupation
plt.figure(figsize=(8,6))
sns.violinplot(x='occupation', y='spend', data=merged_data)
plt.title('Average Spending by Occupation')
plt.xticks(rotation=45, ha='right')
plt.show()
```



In [25]:
```
# Point plot for average spending by city
plt.figure(figsize=(6, 6))
sns.pointplot(x='city', y='spend', data=merged_data)
```

```
plt.title('Average Spending by City')
plt.show()
```



Avg Income Utilisation %:

Calculate the average income utilization percentage of customers (avg_spends/avg_income).

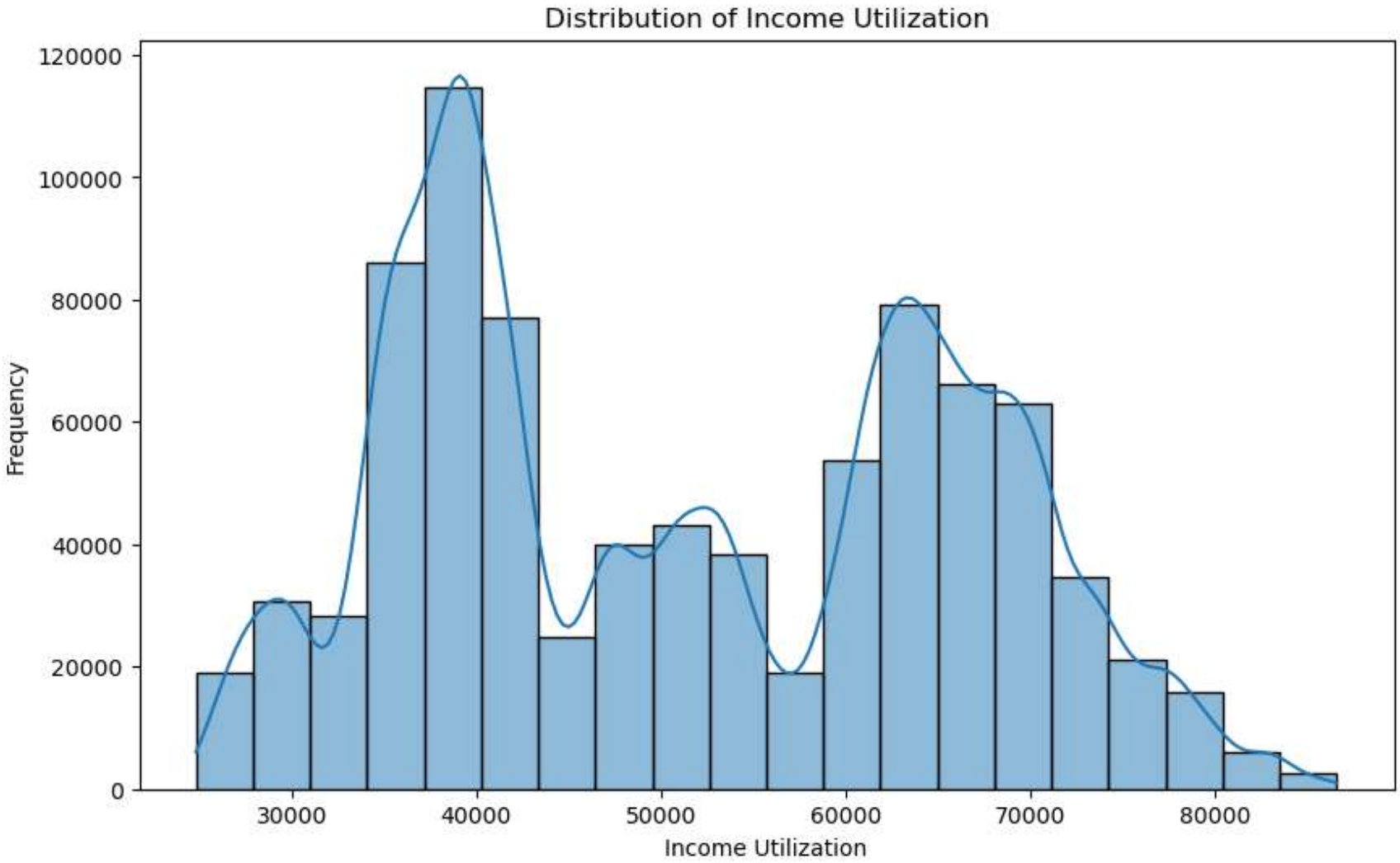
Use this metric as a key indicator of likelihood to use credit cards.

In [26]:
```
# Calculate avg income utilization %
merged_data['income_utilization'] = merged_data['spend'] / merged_data['avg_income']
```

In [27]:
```
# Average income utilization %
avg_income_utilization = merged_data.groupby('customer_id')['income_utilization'].mean()
print(avg_income_utilization)

# Visualize the distribution of income utilization
# Add appropriate visualizations based on your preference
```

```
customer_id
ATQCUS0001    0.015865
ATQCUS0002    0.016481
ATQCUS0003    0.014930
ATQCUS0004    0.016577
ATQCUS0005    0.015583
                ...
ATQCUS3996    0.006780
ATQCUS3997    0.007817
ATQCUS3998    0.008299
ATQCUS3999    0.008300
ATQCUS4000    0.004870
Name: income_utilization, Length: 4000, dtype: float64
```

In [28]:
```
# import matplotlib.pyplot as plt
# import seaborn as sns

# Assuming 'income_utilization' is a column in your dataset
plt.figure(figsize=(10, 6))
sns.histplot(merged_data['avg_income'], bins=20, kde=True)
plt.title('Distribution of Income Utilization')
plt.xlabel('Income Utilization')
plt.ylabel('Frequency')
plt.show()
```



Spending Insights:

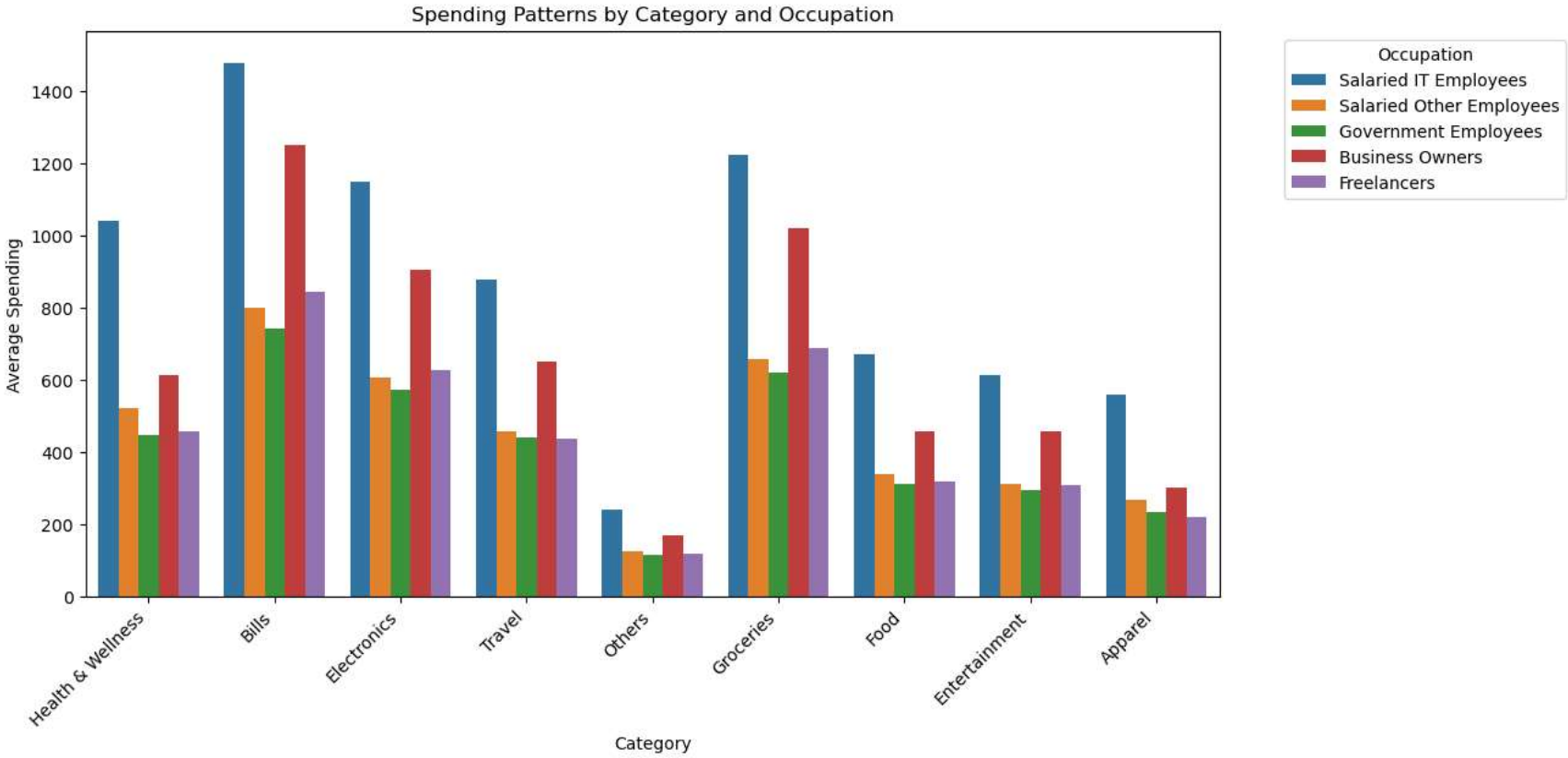
Analyze where people spend money the most and its correlation with occupation, gender, city, and age.

In [29]:
```
# Spending insights
spending_insights = merged_data.groupby(['category', 'occupation', 'gender', 'city', 'age_group'])['spend'].mean()
print(spending_insights)
```

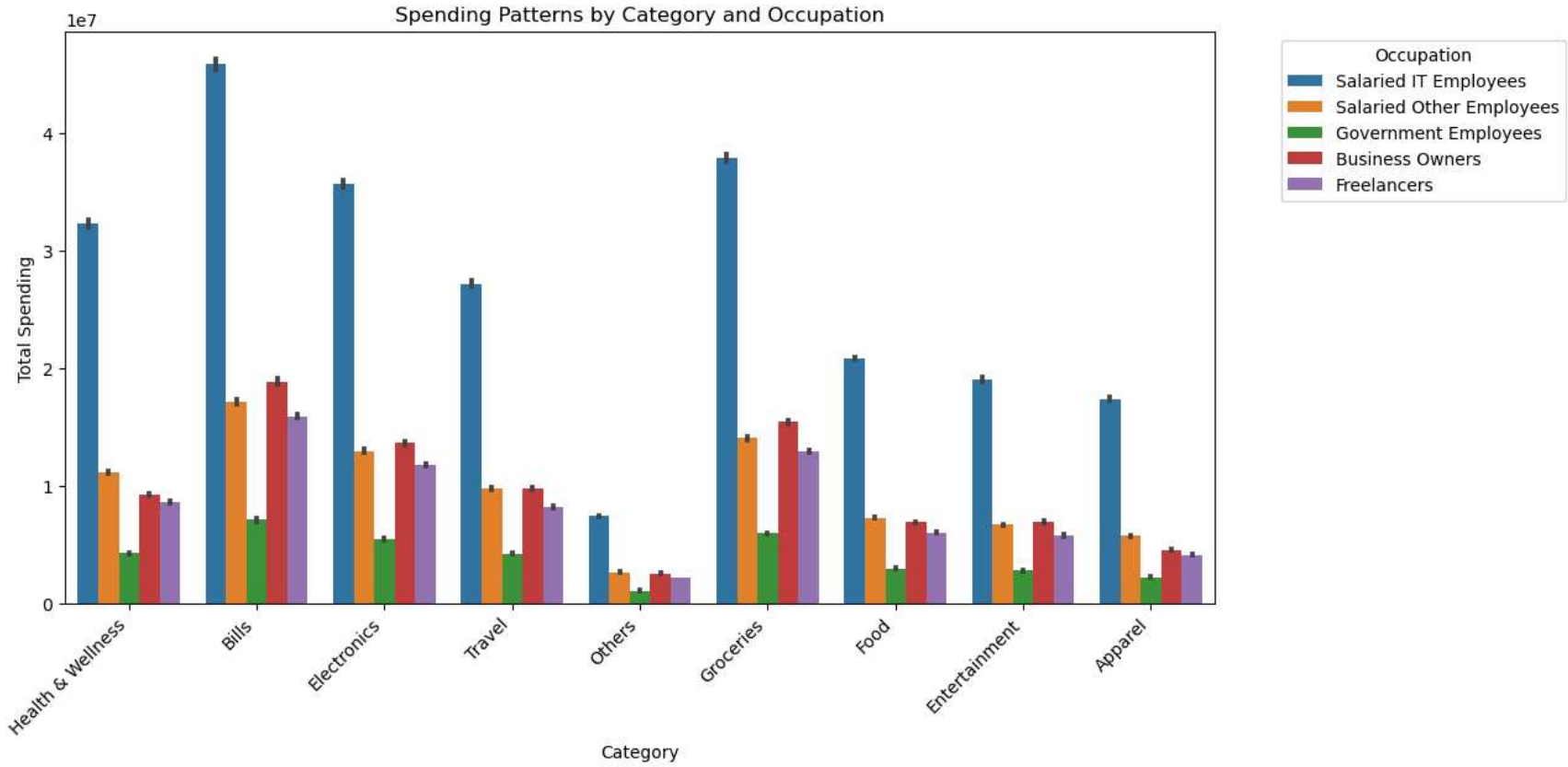
```
# Visualize spending patterns by different factors
# Add appropriate visualizations based on your preference
```

```
category  occupation                gender  city       age_group
Apparel   Business Owners           Female  Bengaluru  21-24        937.708333
                                              25-34        423.240741
                                              35-45        527.812500
                                              45+          411.104167
                                    Chennai   21-24        652.041667
                                                              ...
Travel    Salaried Other Employees  Male    Hyderabad  45+          445.435185
                                    Mumbai    21-24        200.928571
                                              25-34        664.596354
                                              35-45        561.977273
                                              45+          644.705882
Name: spend, Length: 1800, dtype: float64
```

In [30]:
```
# import matplotlib.pyplot as plt
# import seaborn as sns

# Assuming 'category' and 'occupation' are columns in your dataset
plt.figure(figsize=(12, 6))
sns.barplot(x='category', y='spend', hue='occupation', data=merged_data, estimator='mean', errwidth=0)
plt.title('Spending Patterns by Category and Occupation')
plt.xlabel('Category')
plt.ylabel('Average Spending')
plt.xticks(rotation=45, ha='right')
plt.legend(title='Occupation', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



In [31]:
```
# import matplotlib.pyplot as plt
# import seaborn as sns

# Assuming 'category' and 'occupation' are columns in your dataset
plt.figure(figsize=(12, 6))
sns.barplot(x='category', y='spend', hue='occupation', data=merged_data, estimator='sum')
plt.title('Spending Patterns by Category and Occupation')
plt.xlabel('Category')
plt.ylabel('Total Spending')
plt.xticks(rotation=45, ha='right')
plt.legend(title='Occupation', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



Key Customer Segments:

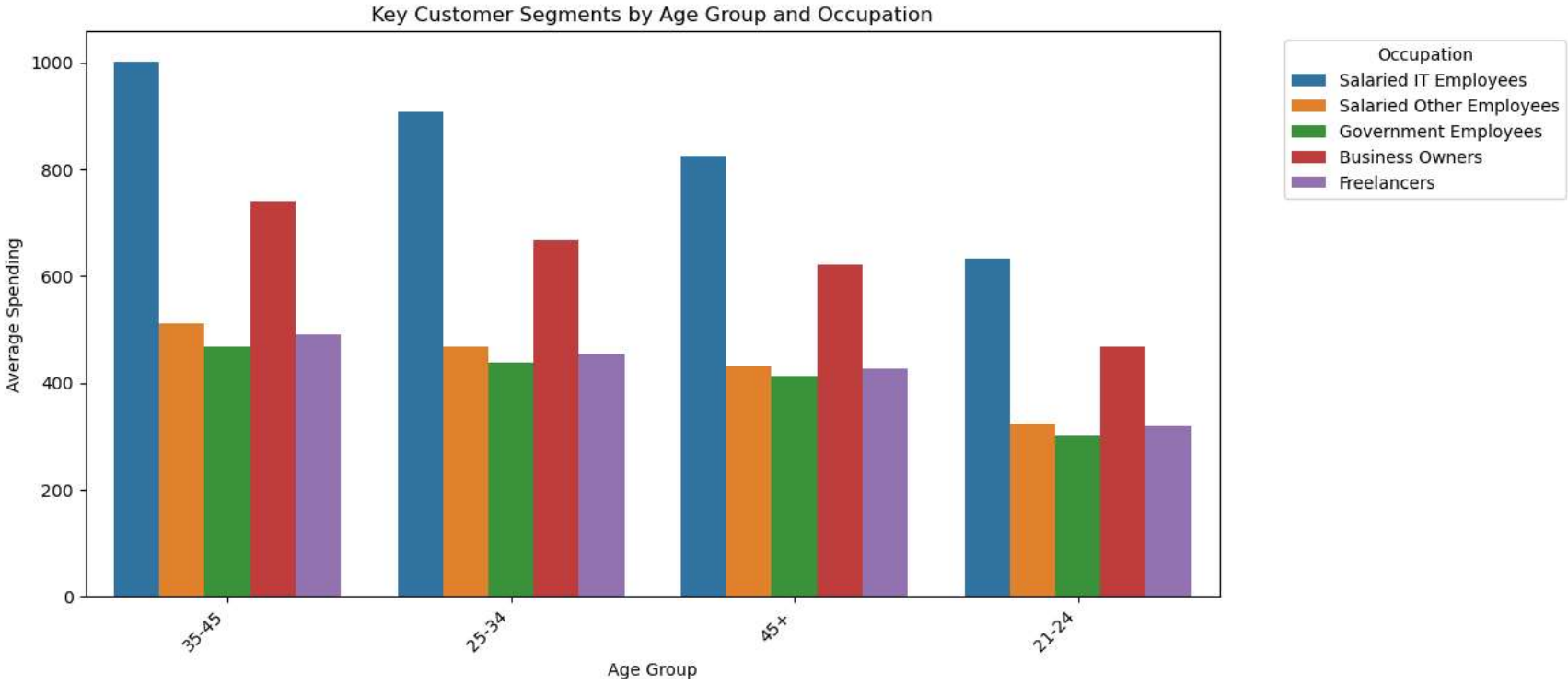
Identify and profile key customer segments based on demographics, spending behaviors, and financial preferences.

In [32]:
```
# Identify key customer segments
key_segments = merged_data.groupby(['age_group', 'occupation'])['spend'].mean()
print(key_segments)

# Visualize key customer segments
# Add appropriate visualizations based on your preference
```

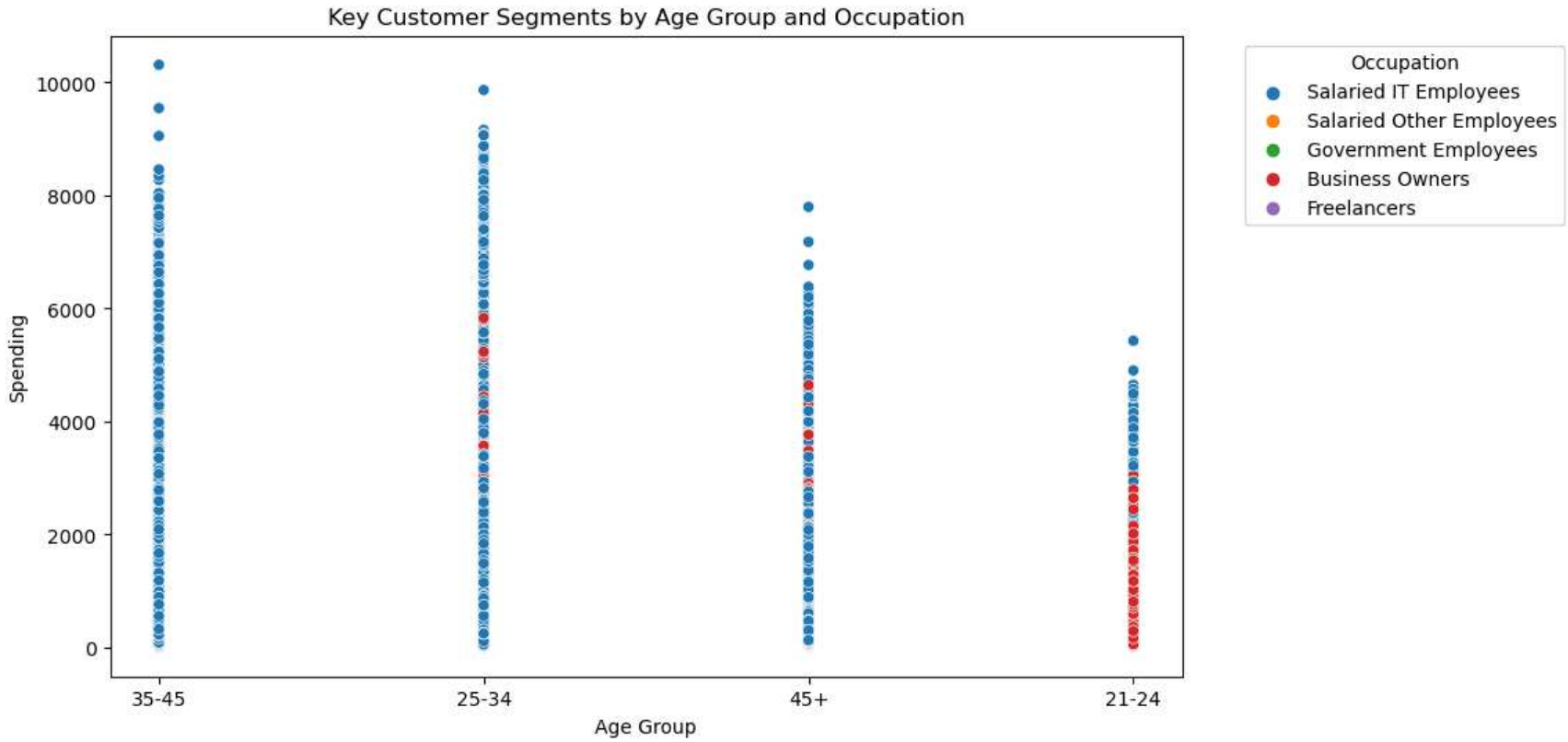
```
age_group  occupation
21-24      Business Owners              468.863451
           Freelancers                  320.013680
           Government Employees         299.751613
           Salaried IT Employees        632.794538
           Salaried Other Employees     324.467652
25-34      Business Owners              668.179781
           Freelancers                  453.930885
           Government Employees         439.340946
           Salaried IT Employees        906.738366
           Salaried Other Employees     468.787366
35-45      Business Owners              741.447133
           Freelancers                  490.178014
           Government Employees         468.001420
           Salaried IT Employees       1001.817769
           Salaried Other Employees     512.254706
45+        Business Owners              621.678352
           Freelancers                  426.108689
           Government Employees         412.422386
           Salaried IT Employees        825.326232
           Salaried Other Employees     431.414223
Name: spend, dtype: float64
```

In [33]:

```
# import matplotlib.pyplot as plt
# import seaborn as sns

# Assuming 'age_group' and 'occupation' are columns in your dataset
plt.figure(figsize=(12, 6))
sns.barplot(x='age_group', y='spend', hue='occupation', data=merged_data, estimator='mean', errwidth=0)
plt.title('Key Customer Segments by Age Group and Occupation')
plt.xlabel('Age Group')
plt.ylabel('Average Spending')
plt.xticks(rotation=45, ha='right')
plt.legend(title='Occupation', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



In [34]:
```
# import matplotlib.pyplot as plt
# import seaborn as sns

# Assuming 'age_group' and 'occupation' are columns in your dataset
plt.figure(figsize=(10, 6))
sns.scatterplot(x='age_group', y='spend', hue='occupation', data=merged_data)
plt.title('Key Customer Segments by Age Group and Occupation')
plt.xlabel('Age Group')
plt.ylabel('Spending')
plt.legend(title='Occupation', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



Credit Card Feature Recommendations:

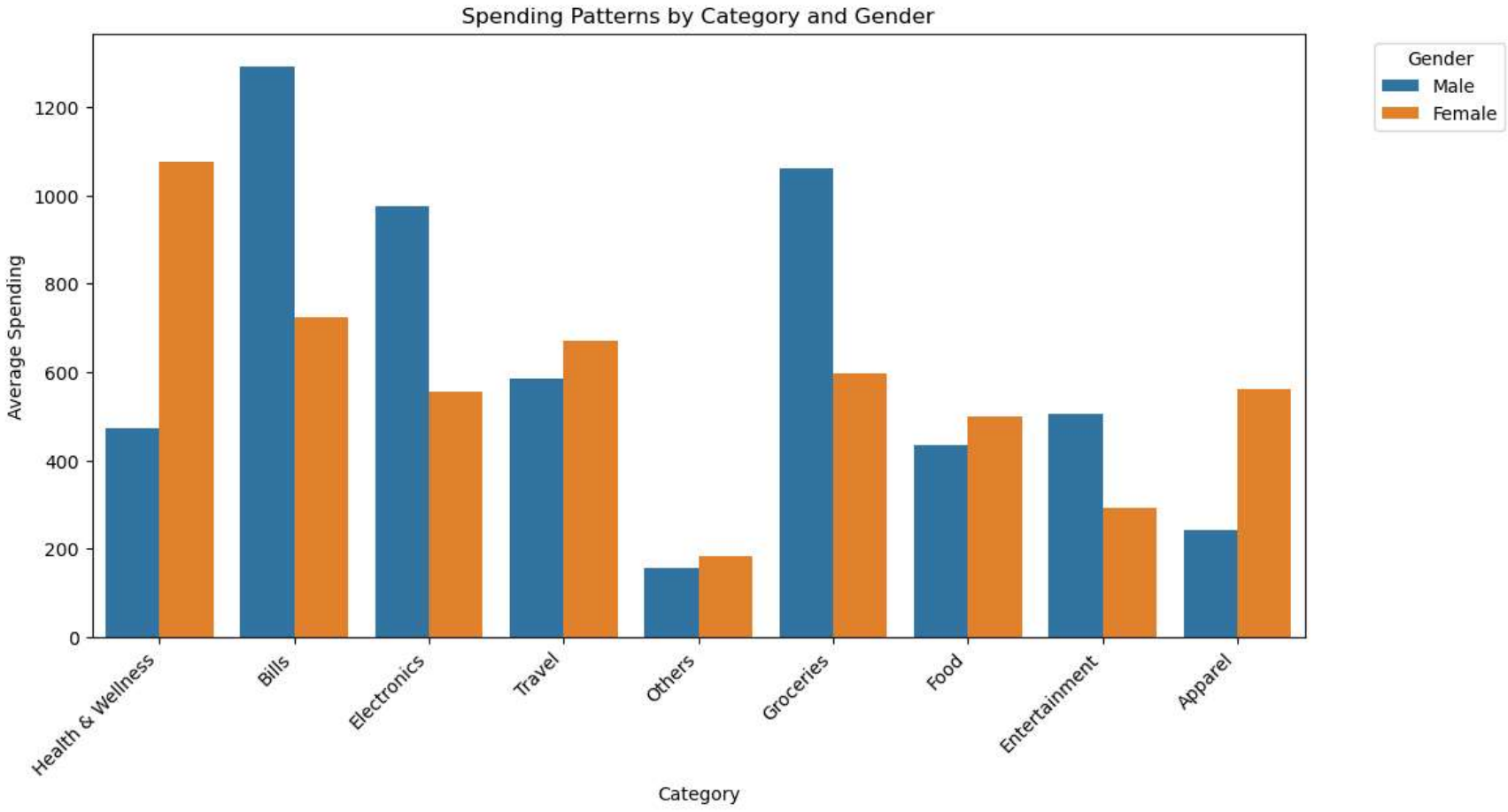
Provide recommendations on key features for credit cards based on insights from the data and secondary research.

In [35]:
```
# Example: Credit card feature recommendations based on spending patterns
credit_card_features = merged_data.groupby('category')['spend'].mean()
print(credit_card_features)

# Visualize spending patterns by category for feature recommendations
# Add appropriate visualizations based on your preference
```

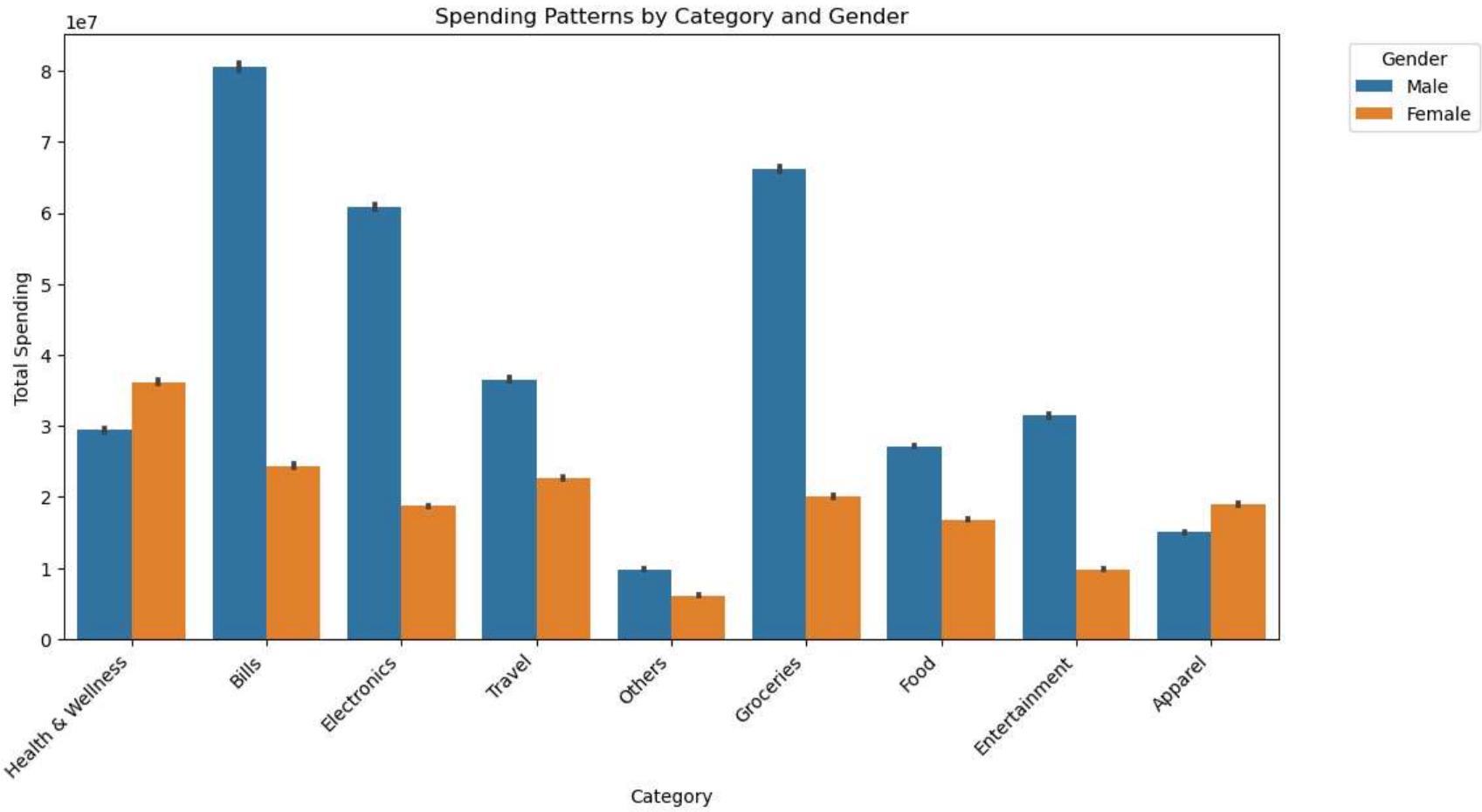
```
category
Apparel               354.541677
Bills                1092.841333
Electronics           828.773125
Entertainment         430.095438
Food                  458.473646
Groceries             898.997510
Health & Wellness     683.331948
Others                166.220646
Travel                616.909625
Name: spend, dtype: float64
```

In [36]:

```
# import matplotlib.pyplot as plt
# import seaborn as sns

# Assuming 'category' and 'gender' are columns in your dataset
plt.figure(figsize=(12, 6))
sns.barplot(x='category', y='spend', hue='gender', data=merged_data, estimator='mean', errwidth=0)
plt.title('Spending Patterns by Category and Gender')
plt.xlabel('Category')
plt.ylabel('Average Spending')
plt.xticks(rotation=45, ha='right')
plt.legend(title='Gender', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



In [37]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming 'category' and 'gender' are columns in your dataset
plt.figure(figsize=(12, 6))
sns.barplot(x='category', y='spend', hue='gender', data=merged_data,estimator='sum')
plt.title('Spending Patterns by Category and Gender')
plt.xlabel('Category')
plt.ylabel('Total Spending')
plt.xticks(rotation=45, ha='right')
plt.legend(title='Gender', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```


Spending Patterns by Category and Gender
1e7
Total Spending
0
1
2
3
4
5
6
7
8
Gender
Male
Female
Health & Wellness
Bills
Electronics
Travel
Others
Groceries
Food
Entertainment
Apparel
Category
