

```
from google.colab import files
uploaded=files.upload()
```

Choose Files

No file chosen

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving Supermarket Sales 2.csv to Supermarket Sales 2.csv

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

```
import io
df= pd.read_csv(io.BytesIO(uploaded['Supermarket Sales 2.csv']))
```

df.head()

	Date	Branch	Customer type	Gender	Product line	Unit price	Quantity	Payment	Rating
0	1/1/2024	Brooklyn	Member	Female	Food & Beverages	84.63	10	Credit card	9.0
1	1/1/2024	Queens	Normal	Female	Electronics	63.22	2	Cash	8.5
2	1/1/2024	Brooklyn	Normal	Female	Electronics	74.71	6	Cash	6.7
3	1/1/2024	Queens	Member	Female	Sports & Travel	36.98	10	Credit card	7.0
4	1/1/2024	Manhattan	Member	Female	Sports & Travel	27.04	4	Ewallet	6.9

```
df['Gross Income'] = df['Unit price'] * df['Quantity']
```

```
sales_by_branch = df.groupby('Branch')['Gross Income'].sum().sort_values(ascending=False)
print("--- Total Sales by Branch (Highest to Lowest) ---")
print(sales_by_branch)
```

---

```
• --- Total Sales by Branch (Highest to Lowest) ---
  Branch
```

```
df['Gross Income'] = df['Unit price'] * df['Quantity']

sales_by_branch = df.groupby('Branch')['Gross Income'].sum().sort_values(ascending=False)
print("--- Total Sales by Branch (Highest to Lowest) ---")
print(sales_by_branch)
```

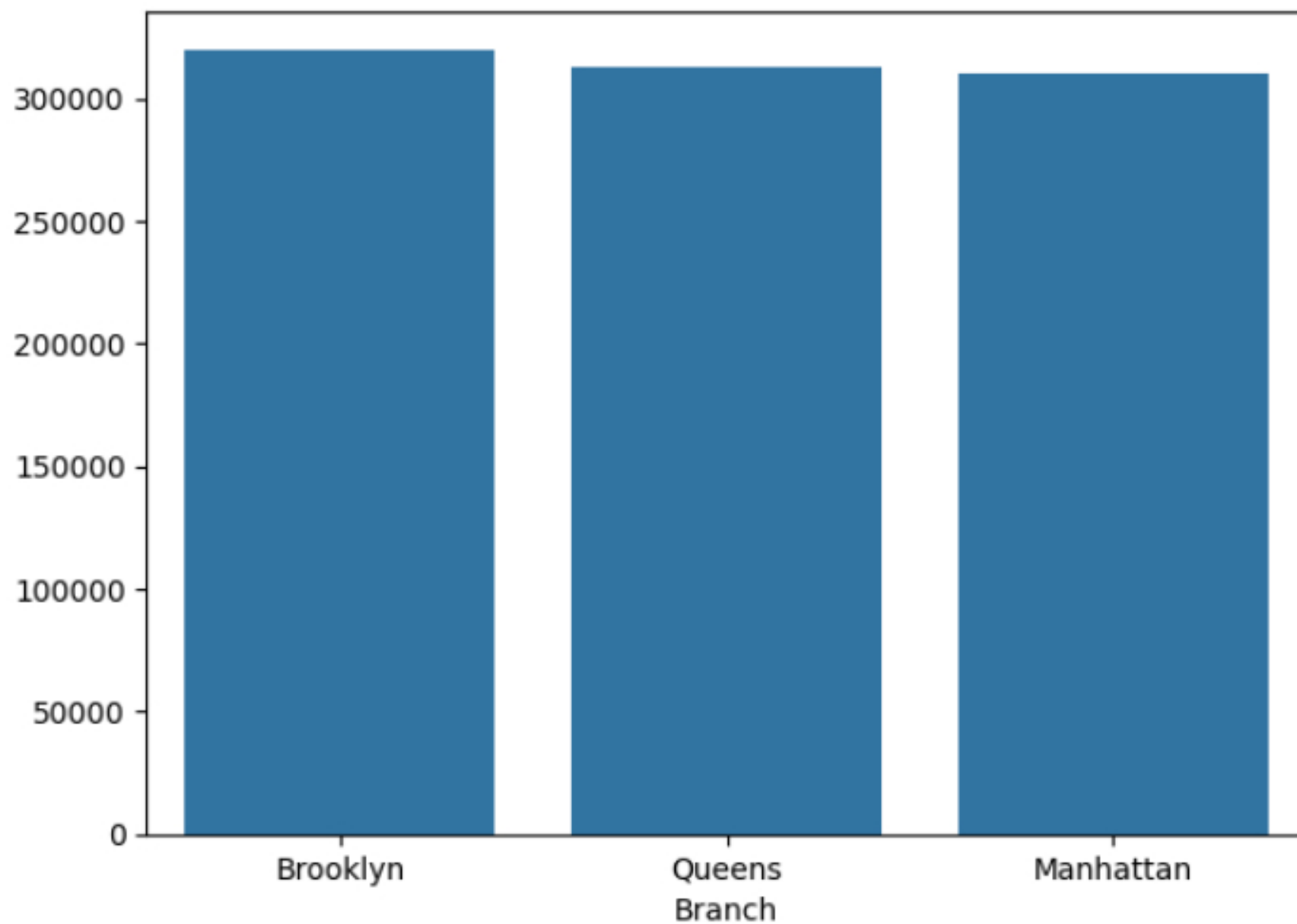
---

```
--- Total Sales by Branch (Highest to Lowest) ---
Branch
Brooklyn      319705.48
Queens        312842.82
Manhattan     309996.55
Name: Gross Income, dtype: float64
```

---

```
plt.figure(figsize=(7, 5))  
sns.barplot(x=sales_by_branch.index, y=sales_by_branch.values)
```

<Axes: xlabel='Branch'>



```
)
sales_by_gender = df.groupby('Gender')['Gross Income'].sum()

gender_sales_array = sales_by_gender.values

female_sales = sales_by_gender.loc['Female']
male_sales = sales_by_gender.loc['Male']

difference = np.abs(female_sales - male_sales)
total_sales = np.sum(gender_sales_array)

percentage_female = (female_sales / total_sales) * 100
percentage_male = (male_sales / total_sales) * 100

print("\n--- Sales Disparity by Gender ")
print(f"Total Female Sales: ${female_sales:.2f}")
print(f"Total Male Sales: ${male_sales:.2f}")
print(f"Revenue Difference: ${difference:.2f} (Females contributed more)")
print(f"Female Contribution to Total Sales: {percentage_female:.2f}%")
print(f"Male Contribution to Total Sales: {percentage_male:.2f}%")
```

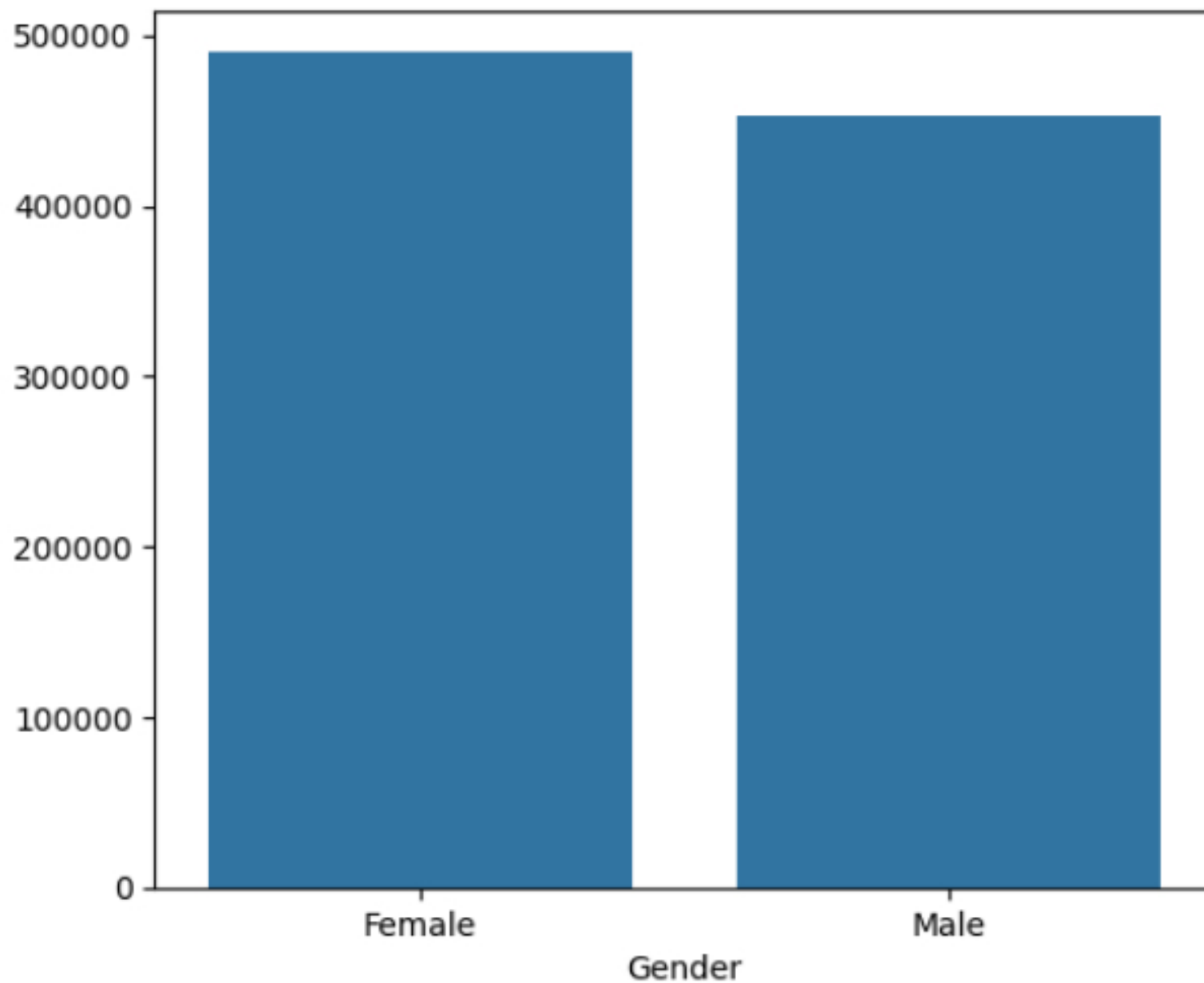
---

```
--- Sales Disparity by Gender
Total Female Sales: $490169.72
Total Male Sales: $452375.13
Revenue Difference: $37794.59 (Females contributed more)
Female Contribution to Total Sales: 52.00%
Male Contribution to Total Sales: 48.00%
```

---

```
plt.figure(figsize=(6, 5))  
sns.barplot(x=sales_by_gender.index, y=sales_by_gender.values)
```

<Axes: xlabel='Gender'>



```
avg_qty_by_customer = df.groupby('Customer type')['Quantity'].mean().sort_values(ascending=False)

member_avg_qty = avg_qty_by_customer.loc['Member']

print("\n--- Average Quantity per Transaction by Customer Type ---")
print(avg_qty_by_customer.to_markdown(numalign="left", stralign="left"))
print(f"\nMember customers have an average basket size of: {member_avg_qty:.2f} units (higher than Normal customers)")
```

---

--- Average Quantity per Transaction by Customer Type ---

Customer type	Quantity
Member	4.84
Normal	4.42

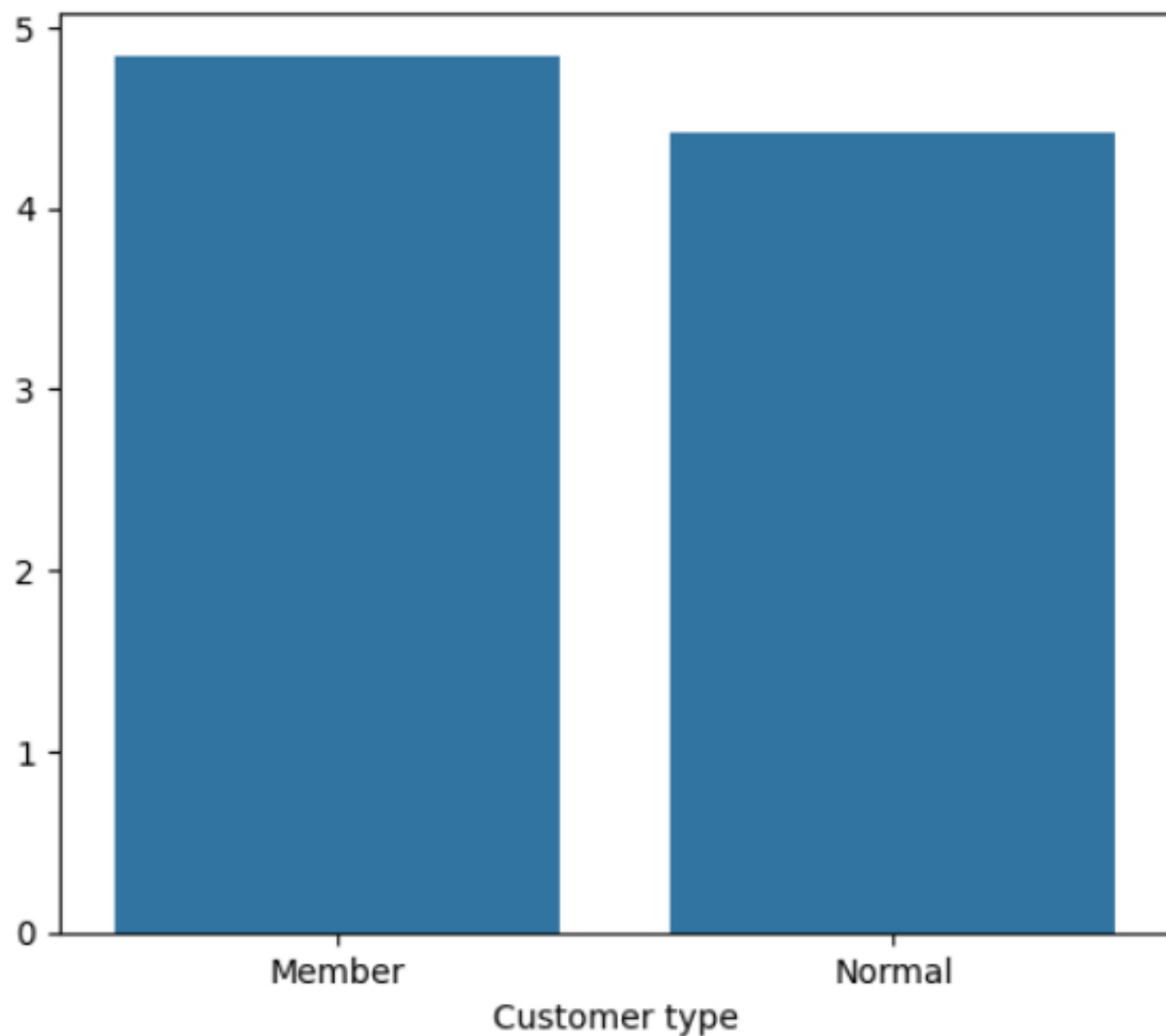
Member customers have an average basket size of: 4.84 units (higher than Normal customers)

---



```
plt.figure(figsize=(6, 5))  
sns.barplot(x=avg_qty_by_customer.index, y=avg_qty_by_customer.values )
```

<Axes: xlabel='Customer type'>



```
payment_method_count = df['Payment'].value_counts()
print("Most Common Payment Method (by transaction count):\n", payment_method_count)
```

---

Most Common Payment Method (by transaction count):

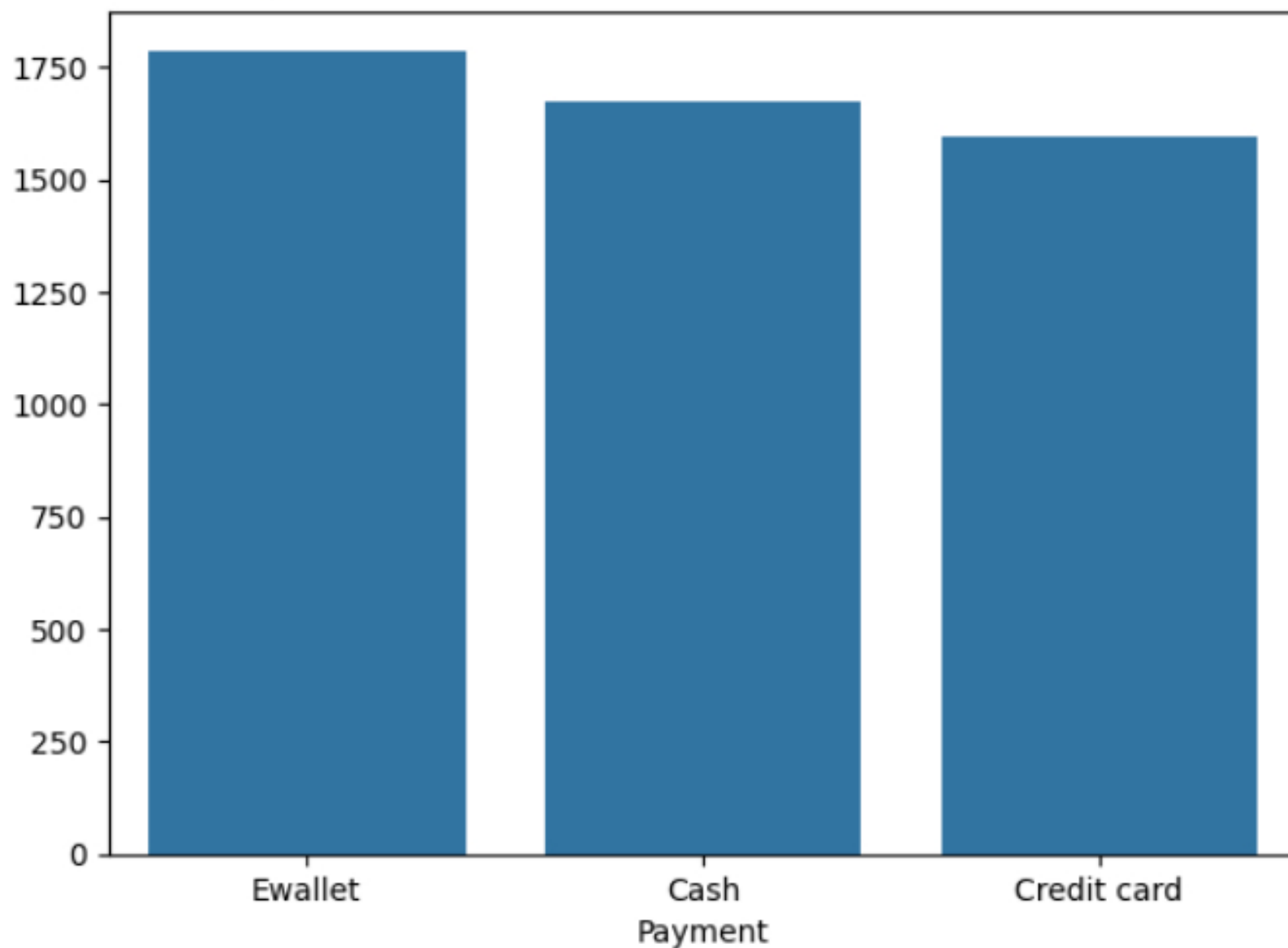
Payment	
Ewallet	1785
Cash	1674
Credit card	1594

Name: count, dtype: int64

---

```
plt.figure(figsize=(7, 5))  
sns.barplot(x=payment_method_count.index, y=payment_method_count.values)
```

<Axes: xlabel='Payment'>



```
avg_rating_by_branch = df.groupby('Branch')['Rating'].mean().sort_values(ascending=False)
print("Average Rating by Branch:\n", avg_rating_by_branch)
least_profitable_branch = df.groupby('Branch')['Gross Income'].sum().idxmin()

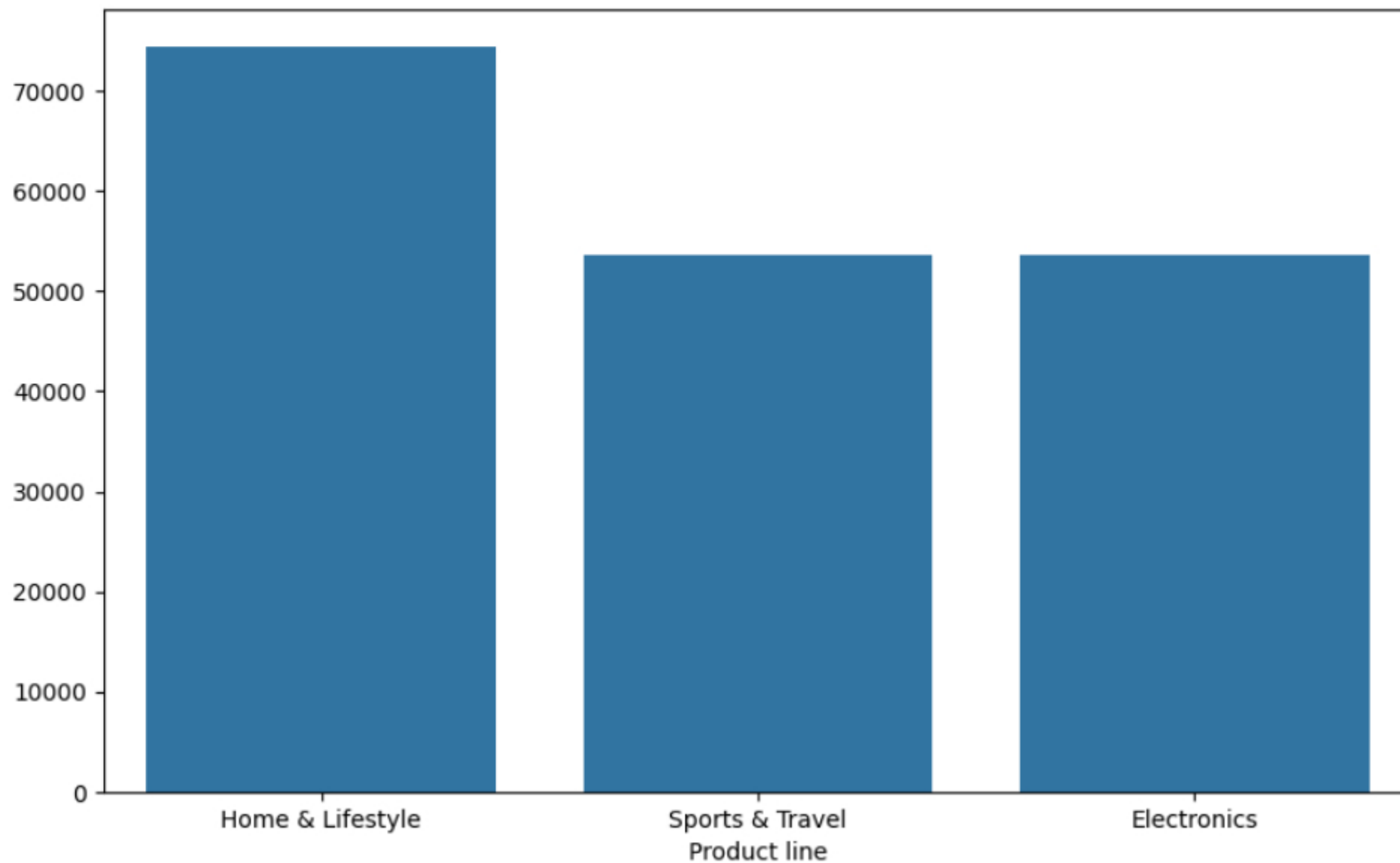
product_sales_in_least_profitable_branch = (
    df[df['Branch'] == least_profitable_branch]
    .groupby('Product line')['Gross Income'].sum()
    .sort_values(ascending=False).head(3).round(2)
)

print(f"Top 3 Product Lines in Least Profitable Branch ({least_profitable_branch}):\n", product_sales_in_least_profitable_branch)
```

---

```
plt.figure(figsize=(10, 6))
sns.barplot(x= product_sales_in_least_profitable_branch.index, y= product_sales_in_least_profitable_branch.values)
```

<Axes: xlabel='Product line'>



```
member_payment_counts = df[df['Customer type'] == 'Member']['Payment'].value_counts().head(3)
print("Top 3 Payment Methods for Member Customers:\n", member_payment_counts)
```

• Top 3 Payment Methods for Member Customers:

Payment	
Ewallet	698
Cash	695
Credit card	692

Name: count, dtype: int64

```
member_payment_counts = df[df['Customer type'] == 'Member']['Payment'].value_counts().head(3)
```

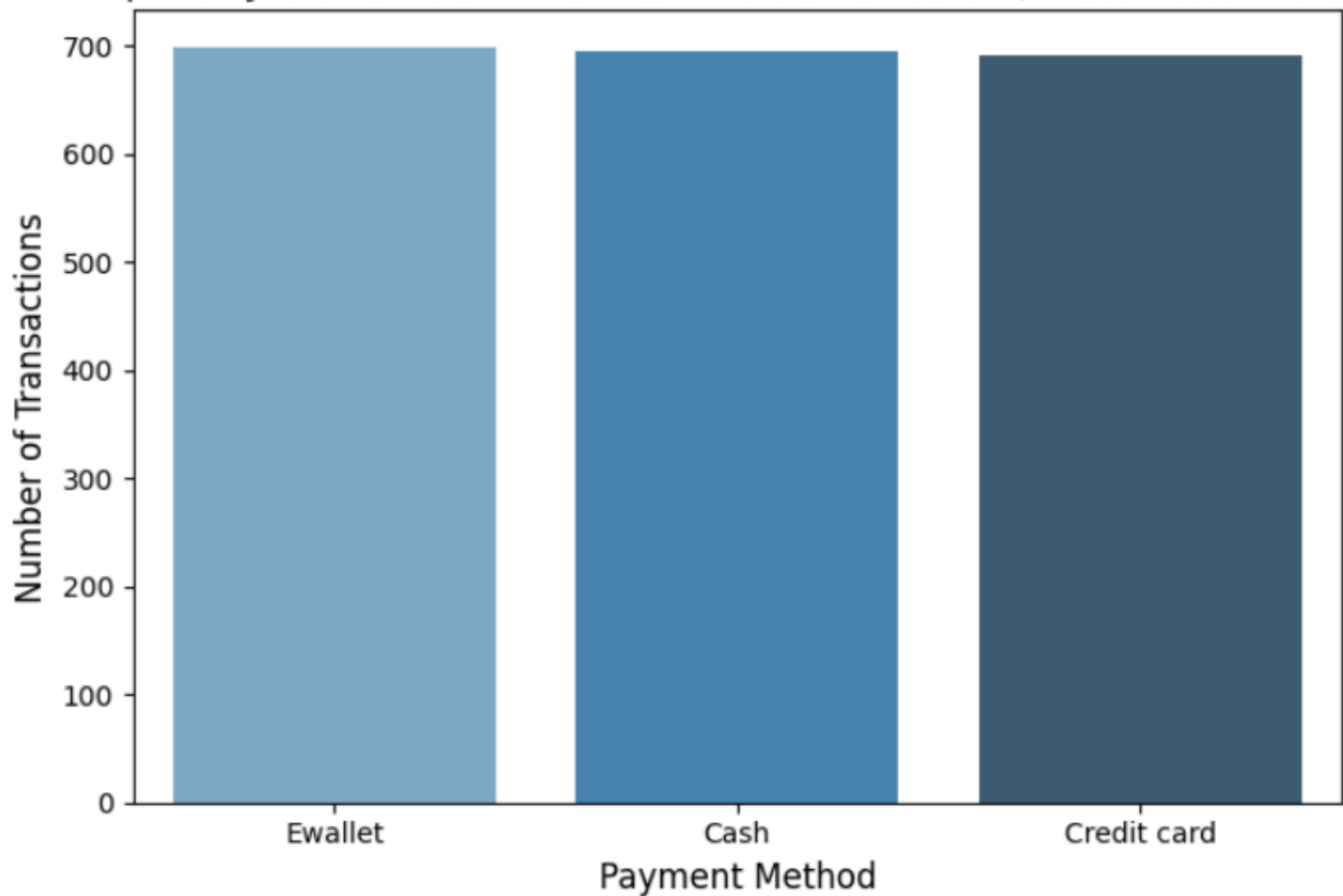
```
plt.figure(figsize=(7, 5))
```

```
sns.barplot(
    x=member_payment_counts.index,
    y=member_payment_counts.values,
    palette="Blues_d"
)
```

```
plt.title('Top 3 Payment Methods for Member Customers (Transaction Count)', fontsize=14)
plt.ylabel('Number of Transactions', fontsize=12)
plt.xlabel('Payment Method', fontsize=12)
plt.tight_layout()
```

```
sns.barplot()
```

Top 3 Payment Methods for Member Customers (Transaction Count)



```
avg_rating_by_branch = df.groupby('Branch')['Rating'].mean().sort_values(ascending=False)
```

---

Average Rating by Branch:

Branch

Manhattan      6.91

Brooklyn       6.87

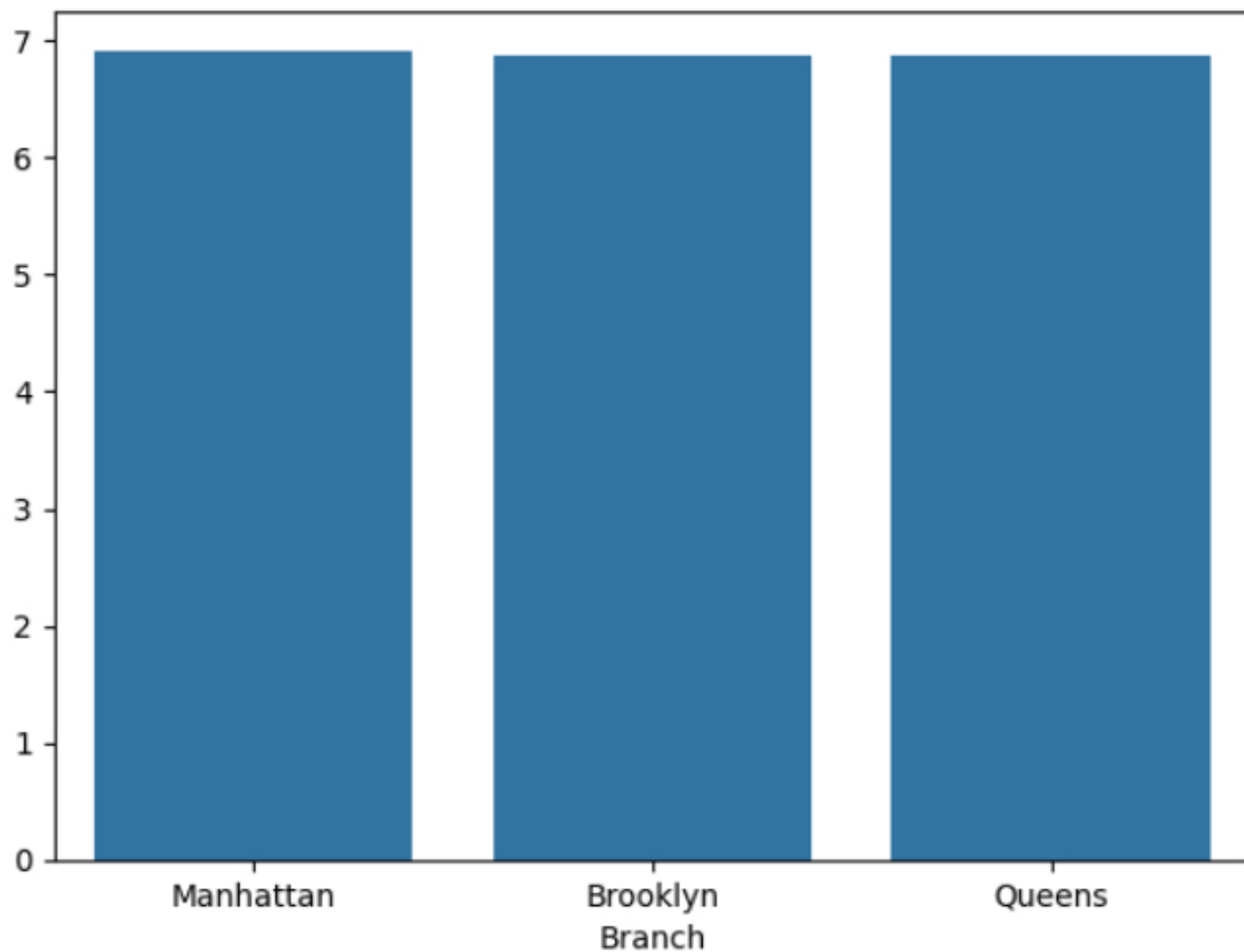
Queens          6.87

Name: Rating, dtype: float64



```
plt.figure(figsize=(7, 5))  
sns.barplot(x=avg_rating_by_branch.index, y=avg_rating_by_branch.values)
```

<Axes: xlabel='Branch'>



```
avg_gross_income_by_product = df.groupby('Product line')['Gross Income'].mean().sort_values(ascending=False)
print("Average Gross Income per Transaction by Product Line:\n", avg_gross_income_by_product)
```

---

Average Gross Income per Transaction by Product Line:

Product line

Electronics            342.49

Sports & Travel        285.16

Home & Lifestyle       278.62

Health & Beauty        175.96

Fashion & Accs         150.66

Food & Beverages       90.63

Name: Gross Income, dtype: float64

---

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

```
sns.barplot(
```

```
    x=avg_gross_income_by_product.index,
```

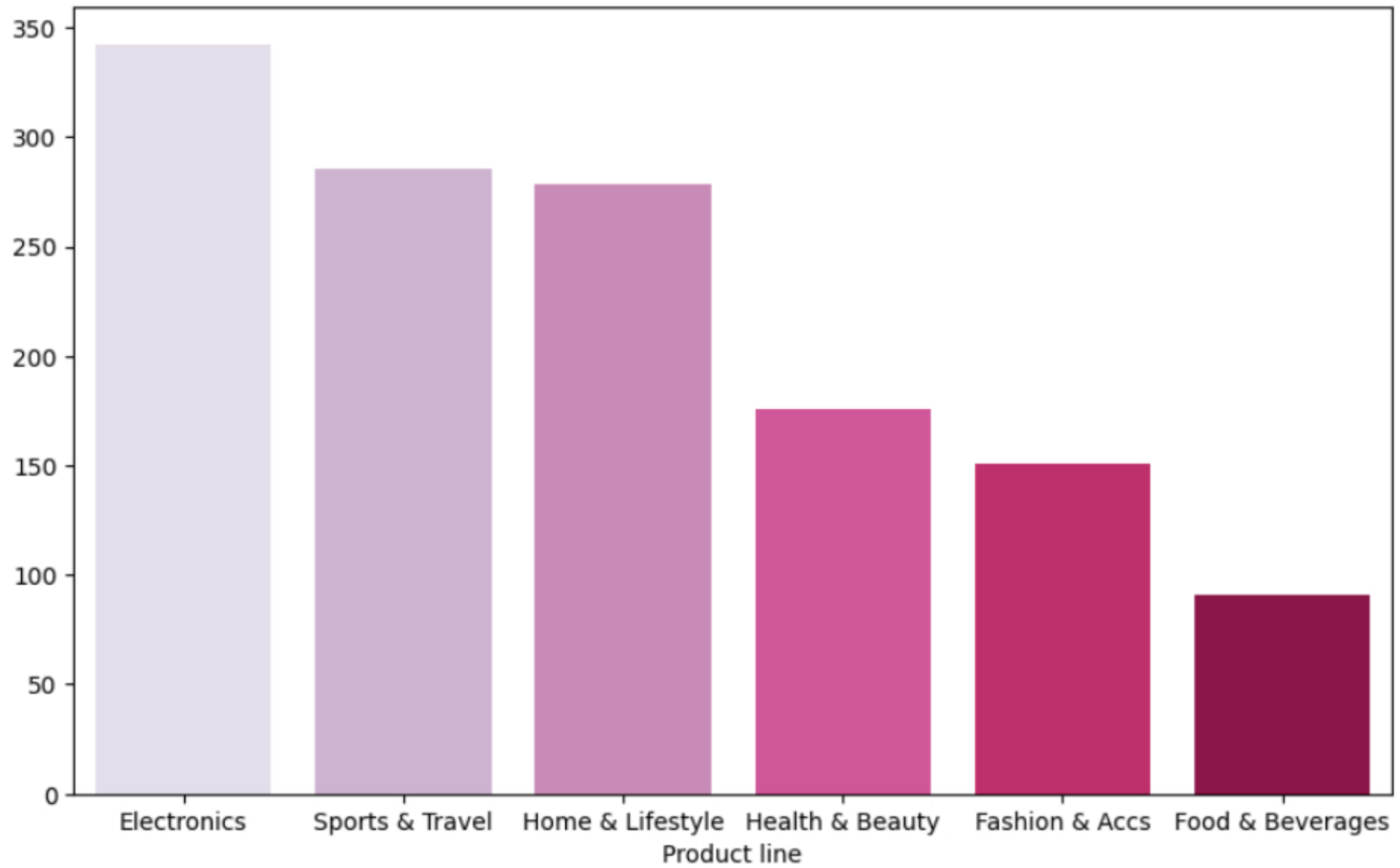
```
    y=avg_gross_income_by_product.values,
```

```
    palette="PuRd"
```

```
)
```

---

```
sns.barplot(  
<Axes: xlabel='Product line'>
```

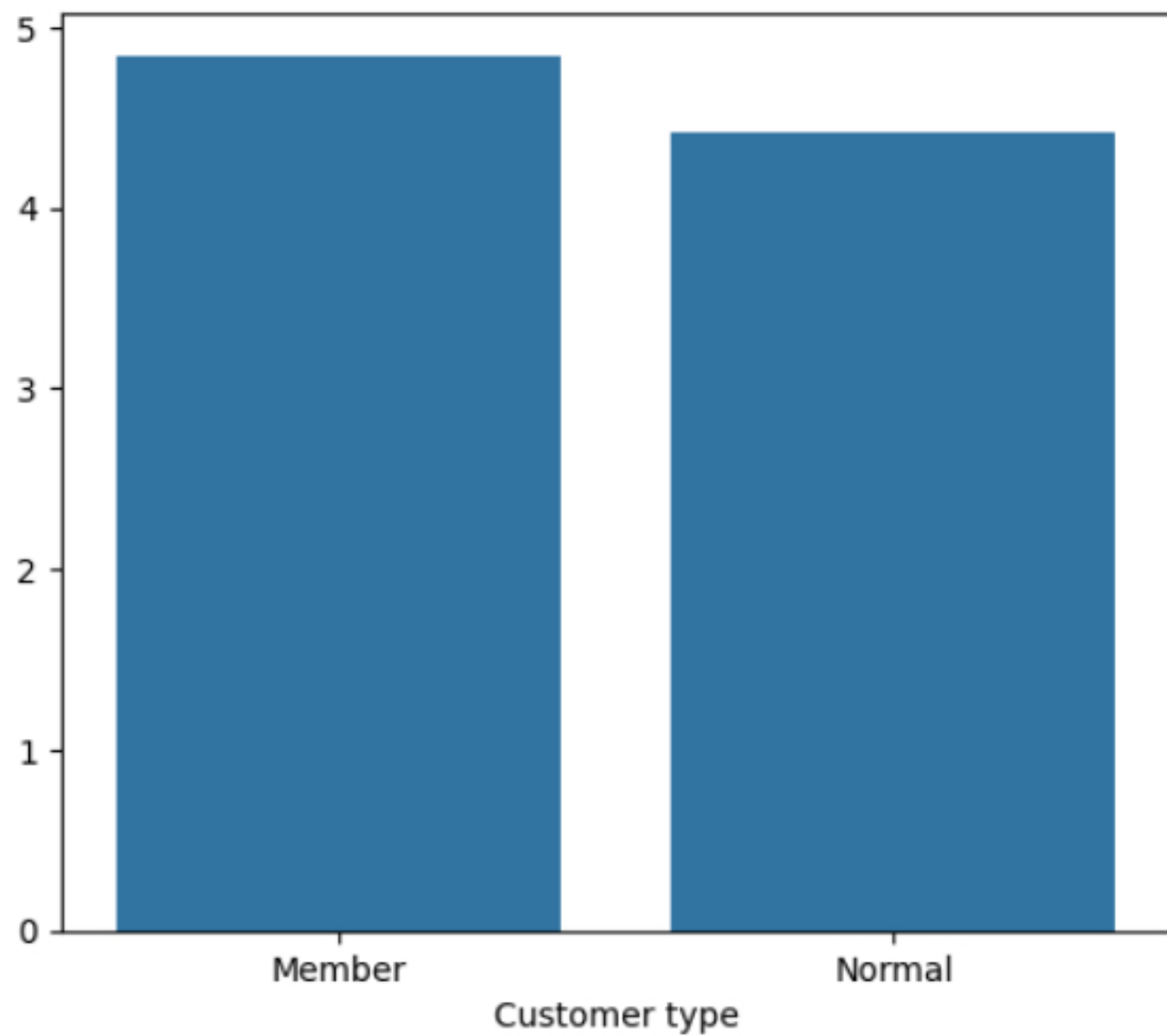


```
avg_qty_by_customer_type = df.groupby('Customer type')['Quantity'].mean().sort_values(ascending=False)
print("Average Quantity per Transaction by Customer Type:\n", avg_qty_by_customer_type)
```

```
*** Average Quantity per Transaction by Customer Type:
      Customer type
Member          4.84
Normal          4.42
Name: Quantity, dtype: float64
```

```
plt.figure(figsize=(6, 5))
sns.barplot(x=avg_qty_by_customer_type.index, y=avg_qty_by_customer_type.values,)
```

<Axes: xlabel='Customer type'>



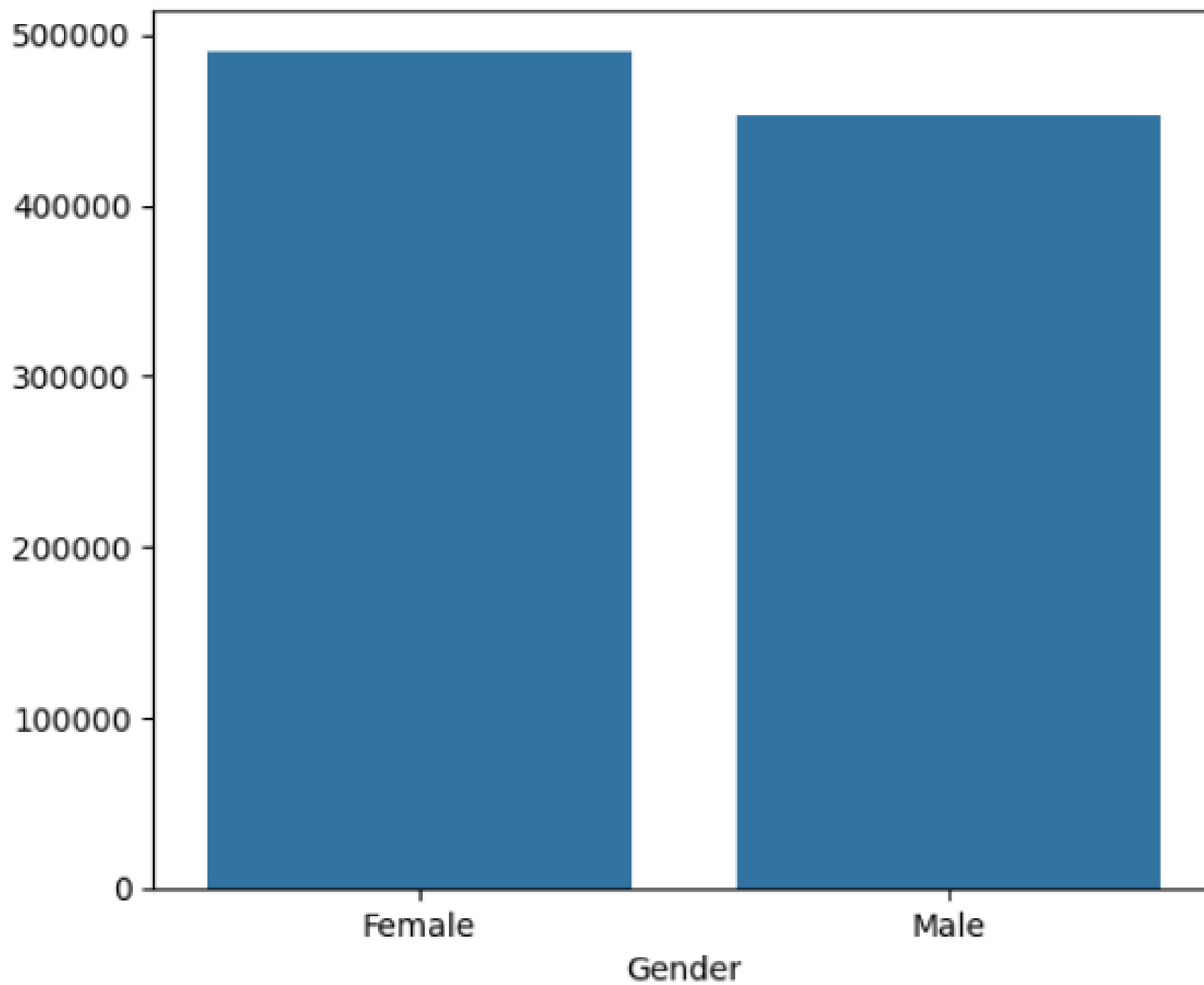
```
)  
sales_by_gender = df.groupby('Gender')['Gross Income'].sum()  
  
gender_diff = np.abs(sales_by_gender.loc['Female'] - sales_by_gender.loc['Male'])  
  
print("Sales by Gender:\n", sales_by_gender)  
print(f"Revenue Difference (Female vs Male): ${gender_diff}")
```

```
• Sales by Gender:  
  Gender  
Female    490169.72  
Male      452375.13  
Name: Gross Income, dtype: float64  
Revenue Difference (Female vs Male): $37794.589999999997
```

```
plt.figure(figsize=(6, 5))  
sns.barplot(  
    x=sales_by_gender.index,  
    y=sales_by_gender.values,  
  
)
```

```
<Axes: xlabel='Gender'>
```

<Axes: xlabel='Gender'>



```
avg_qty_by_customer_type = dt.groupby('Customer type')['Quantity'].mean().sort_values

plt.figure(figsize=(7, 5))

sns.barplot(
    x=avg_qty_by_customer_type.index,
    y=avg_qty_by_customer_type.values,
    palette="copper"
)

plt.title('Average Quantity per Transaction (Basket Size) by Customer Type', fontsize=12)
plt.ylabel('Average Quantity of Items', fontsize=12)
plt.xlabel('Customer Type', fontsize=12)
plt.tight_layout()

plt.savefig('avg_qty_by_customer_type.png')
```

/tmp/ipython-input-2193312749.py:7: FutureWarning:



```
avg_qty_by_customer_type = df.groupby('Customer type')['Quantity'].mean().sort_values(ascending=False).round(2)

plt.figure(figsize=(7, 5))

sns.barplot(
    x=avg_qty_by_customer_type.index,
    y=avg_qty_by_customer_type.values,
    palette="copper"
)

plt.title('Average Quantity per Transaction (Basket Size) by Customer Type', fontsize=14)
plt.ylabel('Average Quantity of Items', fontsize=12)
plt.xlabel('Customer Type', fontsize=12)
plt.tight_layout()

plt.savefig('avg_qty_by_customer_type.png')
```

```
sns.barplot()
```

