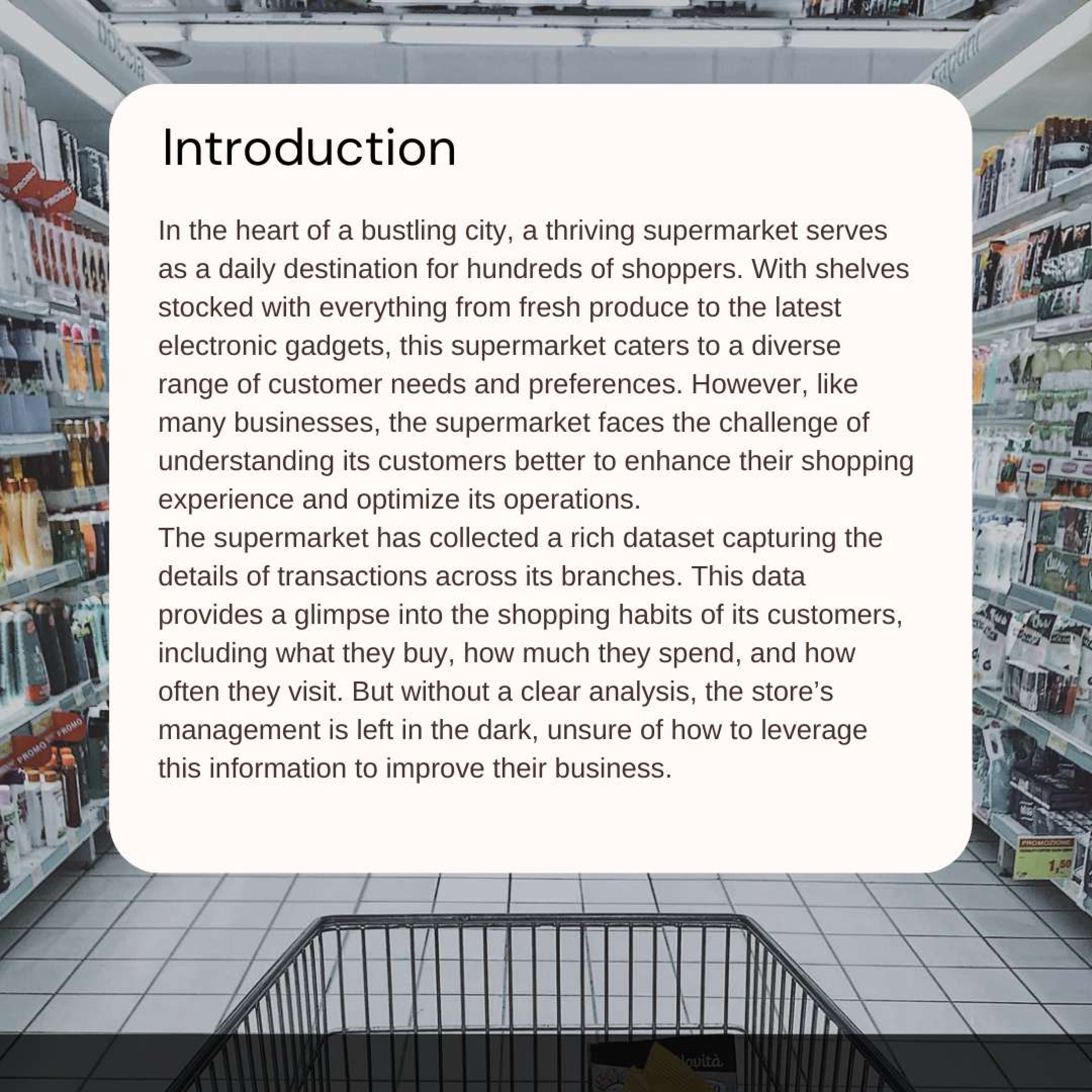
SUPERMARKET SALES REPORT

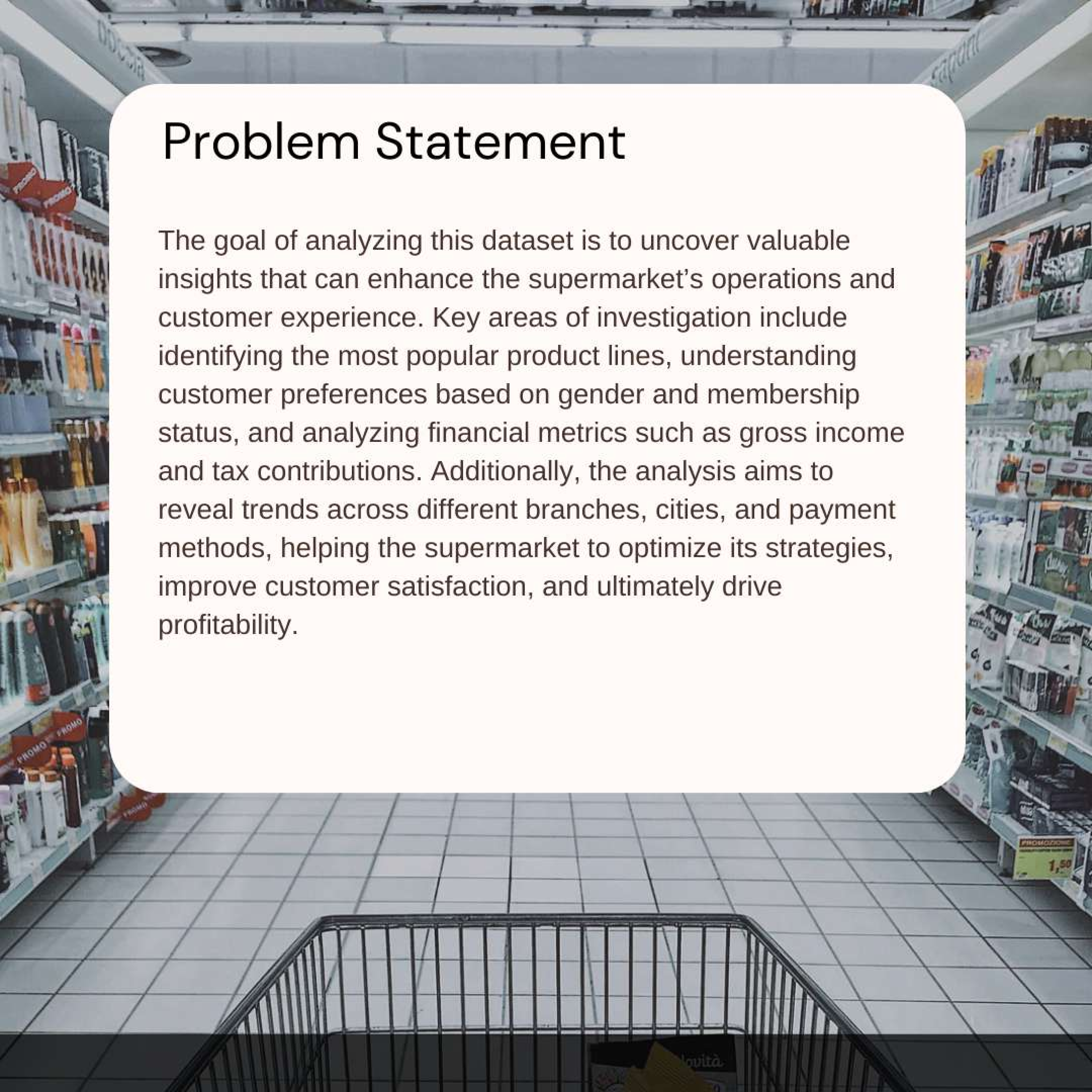
BY RIZWAN SHAH



Introduction

In the heart of a bustling city, a thriving supermarket serves as a daily destination for hundreds of shoppers. With shelves stocked with everything from fresh produce to the latest electronic gadgets, this supermarket caters to a diverse range of customer needs and preferences. However, like many businesses, the supermarket faces the challenge of understanding its customers better to enhance their shopping experience and optimize its operations.

The supermarket has collected a rich dataset capturing the details of transactions across its branches. This data provides a glimpse into the shopping habits of its customers, including what they buy, how much they spend, and how often they visit. But without a clear analysis, the store’s management is left in the dark, unsure of how to leverage this information to improve their business.



Problem Statement

The goal of analyzing this dataset is to uncover valuable insights that can enhance the supermarket’s operations and customer experience. Key areas of investigation include identifying the most popular product lines, understanding customer preferences based on gender and membership status, and analyzing financial metrics such as gross income and tax contributions. Additionally, the analysis aims to reveal trends across different branches, cities, and payment methods, helping the supermarket to optimize its strategies, improve customer satisfaction, and ultimately drive profitability.

Supermarket sales

September 4, 2024

[1]:

```
import opendatasets as od
```

[2]:

```
link=r'https://www.kaggle.com/datasets/aungpyaeap/supermarket-sales?
  ↪select=supermarket_sales+-+Sheet1.csv'
od.download(link)
```

```
Please provide your Kaggle credentials to download this dataset. Learn more:
http://bit.ly/kaggle-creds
Your Kaggle username:

  shahrizwan52

Your Kaggle Key:

  ········

Dataset URL: https://www.kaggle.com/datasets/aungpyaeap/supermarket-sales
Downloading supermarket-sales.zip to .\supermarket-sales

100%|
  | 35.9k/35.9k [00:00<00:00, 211kB/s]
```

[9]:

```
import os
import pandas as pd
file_path=r'C:\Users\Rizwan\Downloads\Shah Rizwan\Python data analyst␣
  ↪project\Supermarket sales\supermarket-sales\supermarket_sales - Sheet1.csv'

clean_path = os.path.normpath(file_path)
df=pd.read_csv(clean_path,encoding='unicode_escape')
```

[10]:

```
df.head()
```

[10]:

```
    Invoice ID Branch       City Customer type  Gender  \
0  750-67-8428      A     Yangon        Member  Female
1  226-31-3081      C  Naypyitaw        Normal  Female
2  631-41-3108      A     Yangon        Normal    Male
3  123-19-1176      A     Yangon        Member    Male
4  373-73-7910      A     Yangon        Normal    Male
```

```
            Product line  Unit price  Quantity   Tax 5%     Total       Date  \
0      Health and beauty       74.69         7  26.1415  548.9715   1/5/2019
1 Electronic accessories       15.28         5   3.8200   80.2200   3/8/2019
2     Home and lifestyle       46.33         7  16.2155  340.5255   3/3/2019
3      Health and beauty       58.22         8  23.2880  489.0480  1/27/2019
4      Sports and travel       86.31         7  30.2085  634.3785   2/8/2019

   Time      Payment    cogs  gross margin percentage  gross income  Rating
0  13:08      Ewallet  522.83                 4.761905       26.1415     9.1
1  10:29         Cash   76.40                 4.761905        3.8200     9.6
2  13:23  Credit card  324.31                 4.761905       16.2155     7.4
3  20:33      Ewallet  465.76                 4.761905       23.2880     8.4
4  10:37      Ewallet  604.17                 4.761905       30.2085     5.3
```

[13]:

```
df.count()
```

[13]:

```
Invoice ID                 1000
Branch                     1000
City                       1000
Customer type              1000
Gender                     1000
Product line               1000
Unit price                 1000
Quantity                   1000
Tax 5%                     1000
Total                      1000
Date                       1000
Time                       1000
Payment                    1000
cogs                       1000
gross margin percentage    1000
gross income               1000
Rating                     1000
dtype: int64
```

no null value in the dataset

[22]:

```
#df['Branch'].unique()
#df['City'].unique()
#df['Customer type'].unique()
#df['Gender'].unique()
#df['Product line'].unique()
#df['Payment'].unique()
```

[22]:

```
array(['Ewallet', 'Cash', 'Credit card'], dtype=object)
```

[23]:

```
df.dtypes
```

[23]:

```
Invoice ID                  object
Branch                      object
City                        object
Customer type               object
Gender                      object
Product line                object
Unit price                 float64
Quantity                     int64
Tax 5%                     float64
Total                      float64
Date                        object
Time                        object
Payment                     object
cogs                       float64
gross margin percentage    float64
gross income               float64
Rating                     float64
dtype: object
```

[24]:

```
df['Date']=pd.to_datetime(df['Date'])
```

[25]:

```
df.dtypes
```

[25]:

```
Invoice ID                         object
Branch                             object
City                               object
Customer type                      object
Gender                             object
Product line                       object
Unit price                        float64
Quantity                            int64
Tax 5%                            float64
Total                             float64
Date                       datetime64[ns]
Time                               object
Payment                            object
cogs                              float64
gross margin percentage           float64
gross income                      float64
Rating                            float64
dtype: object
```

[26]:

```
df.head()
```

[26]:

```
    Invoice ID Branch       City Customer type  Gender  \
0  750-67-8428      A     Yangon        Member  Female
1  226-31-3081      C  Naypyitaw        Normal  Female
```

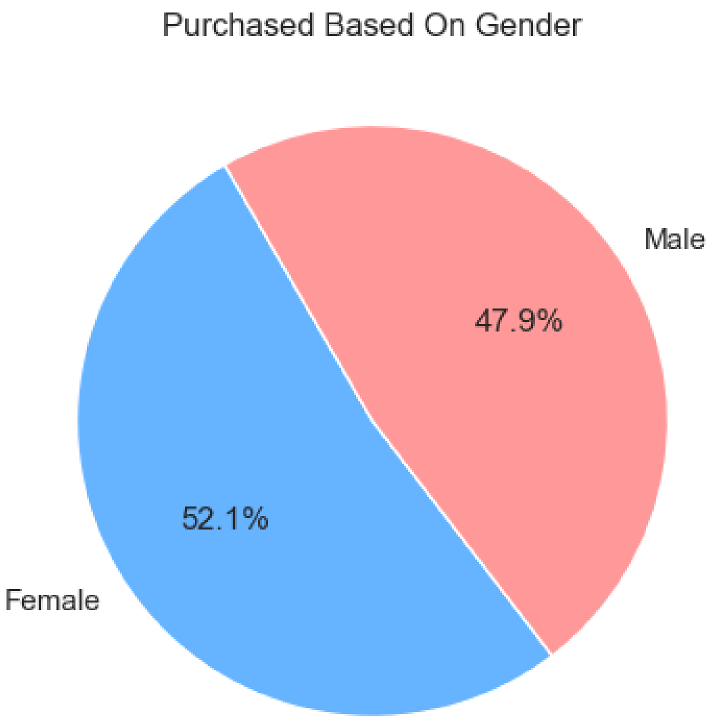
```
2  631-41-3108      A     Yangon        Normal    Male
3  123-19-1176      A     Yangon        Member    Male
4  373-73-7910      A     Yangon        Normal    Male

             Product line  Unit price  Quantity   Tax 5%     Total        Date  \
0       Health and beauty       74.69         7  26.1415  548.9715  2019-01-05
1  Electronic accessories       15.28         5   3.8200   80.2200  2019-03-08
2      Home and lifestyle       46.33         7  16.2155  340.5255  2019-03-03
3       Health and beauty       58.22         8  23.2880  489.0480  2019-01-27
4       Sports and travel       86.31         7  30.2085  634.3785  2019-02-08

    Time      Payment    cogs  gross margin percentage  gross income  Rating
0  13:08      Ewallet  522.83                 4.761905       26.1415     9.1
1  10:29         Cash   76.40                 4.761905        3.8200     9.6
2  13:23  Credit card  324.31                 4.761905       16.2155     7.4
3  20:33      Ewallet  465.76                 4.761905       23.2880     8.4
4  10:37      Ewallet  604.17                 4.761905       30.2085     5.3
```

[27]:

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

[60]:

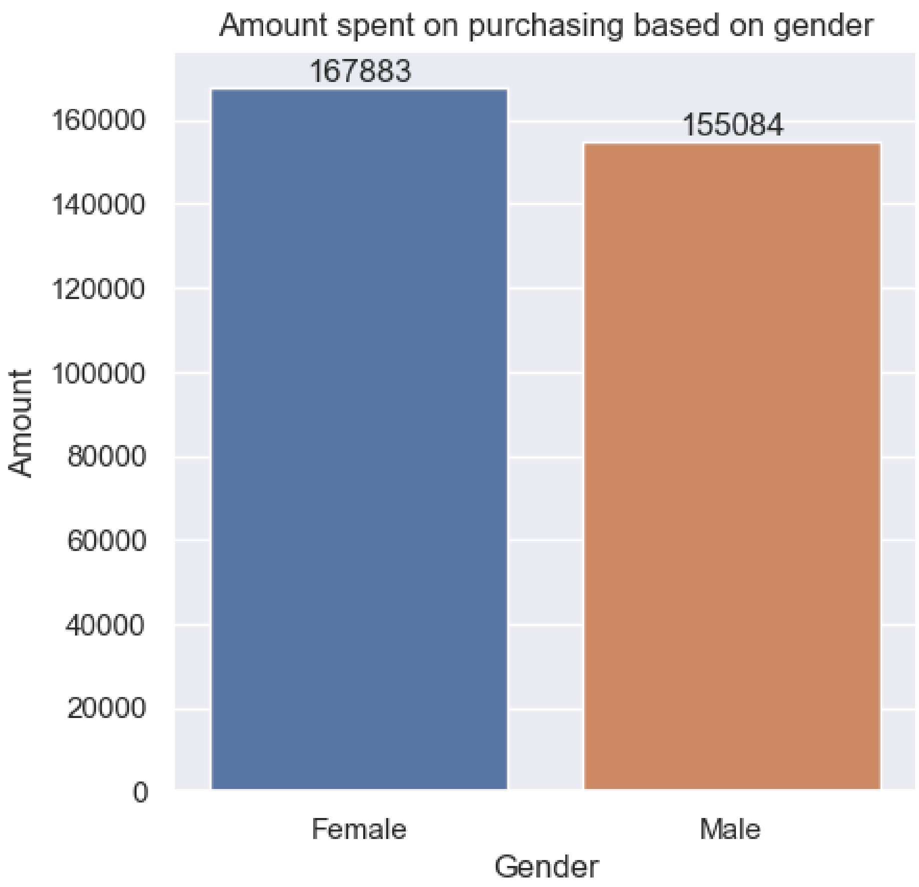
```python
total_pucrhased_gender_counts=df.groupby('Gender')['Quantity'].sum()
#total_pucrhased_gender_counts

sns.set(rc={'figure.figsize':(5,5)})
plt.pie(total_pucrhased_gender_counts,labels=total_pucrhased_gender_counts.
 ↪index,autopct='%1.1f%%',colors=['#66b3ff', '#ff9999'], startangle=120)
plt.title('Purchased Based On Gender')
plt.show()
```



```
Female purchased slightly more than Male
```

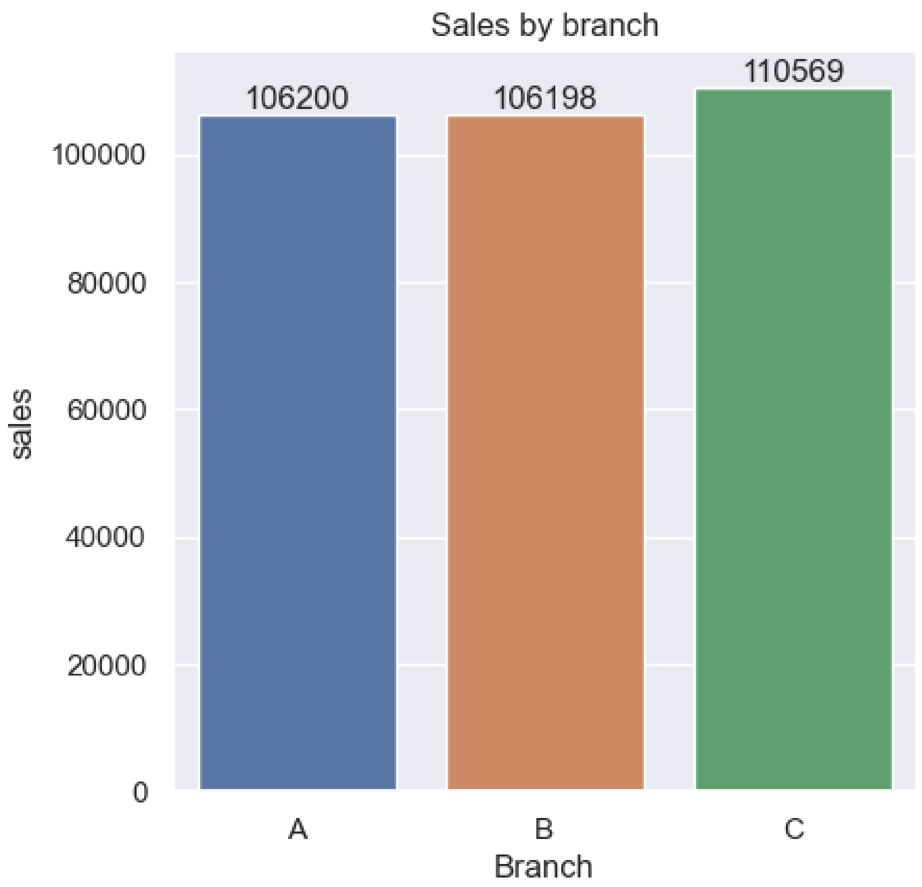
[68]:

```
sales_gen=df.groupby(['Gender'],as_index=False)['Total'].sum()
ax=sns.barplot(x='Gender',y='Total',hue='Gender',data=sales_gen)
plt.ylabel('Amount')
plt.title('Amount spent on purchasing based on gender')

for bar in ax.containers:
    ax.bar_label(bar)

plt.show()
```



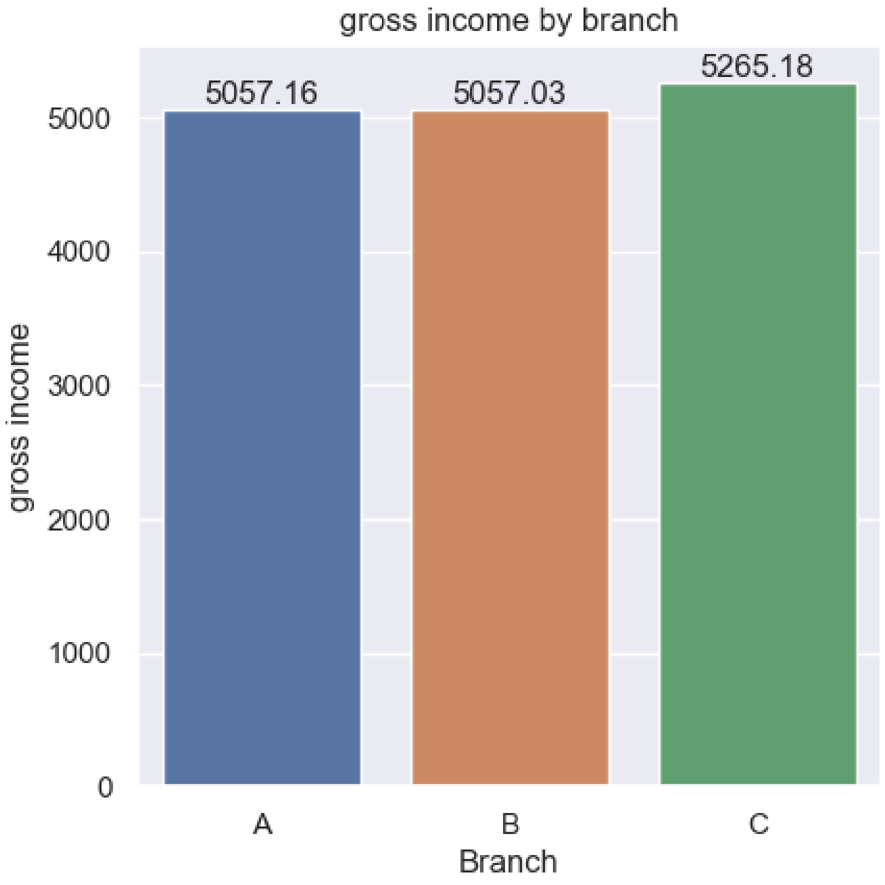
From above graph we can say that Female purchase slightly more than male with more Amount Spent on Purchasing

[71]:
```
sales_branch=df.groupby(['Branch'],as_index=False)['Total'].sum()
ax=sns.barplot(x='Branch',y='Total',hue='Branch',data=sales_branch)
plt.ylabel('sales')
plt.title('Sales by branch')

for bar in ax.containers:
    ax.bar_label(bar)
plt.show()
```



[78]:

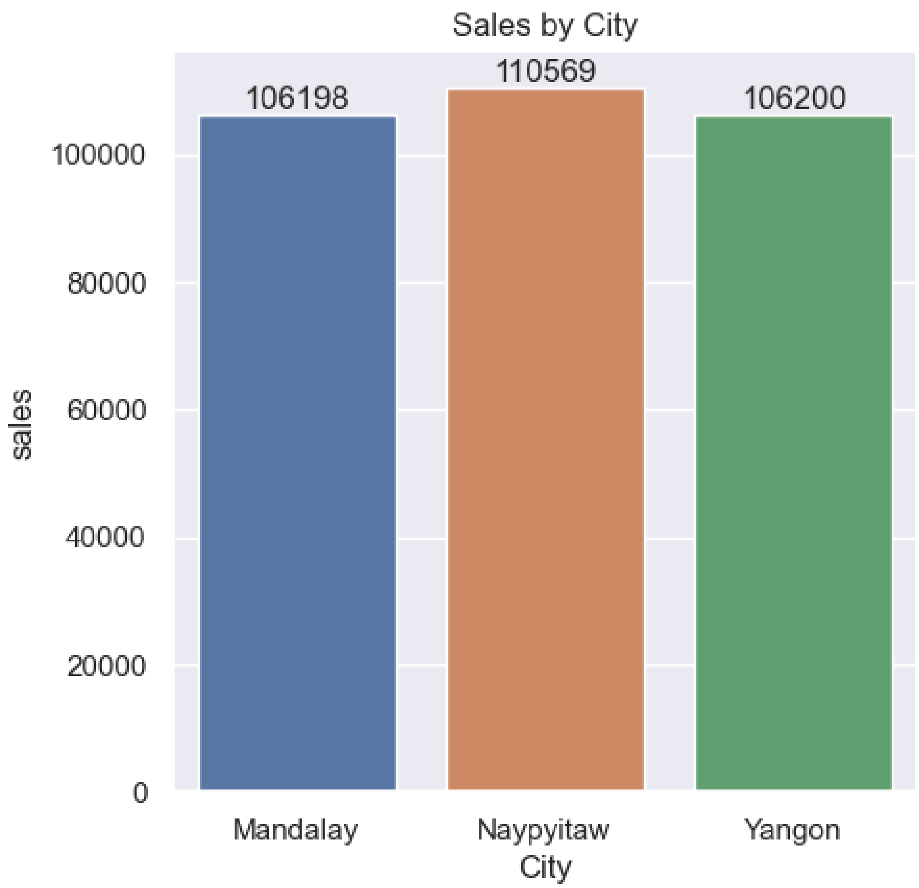
```
gross_income_branch=df.groupby(['Branch'],as_index=False)['gross income'].sum()
ax=sns.barplot(x='Branch',y='gross␣
 ↪income',hue='Branch',data=gross_income_branch)
plt.ylabel('gross income')
plt.title('gross income by branch')

for bar in ax.containers:
    ax.bar_label(bar)
plt.show()
```



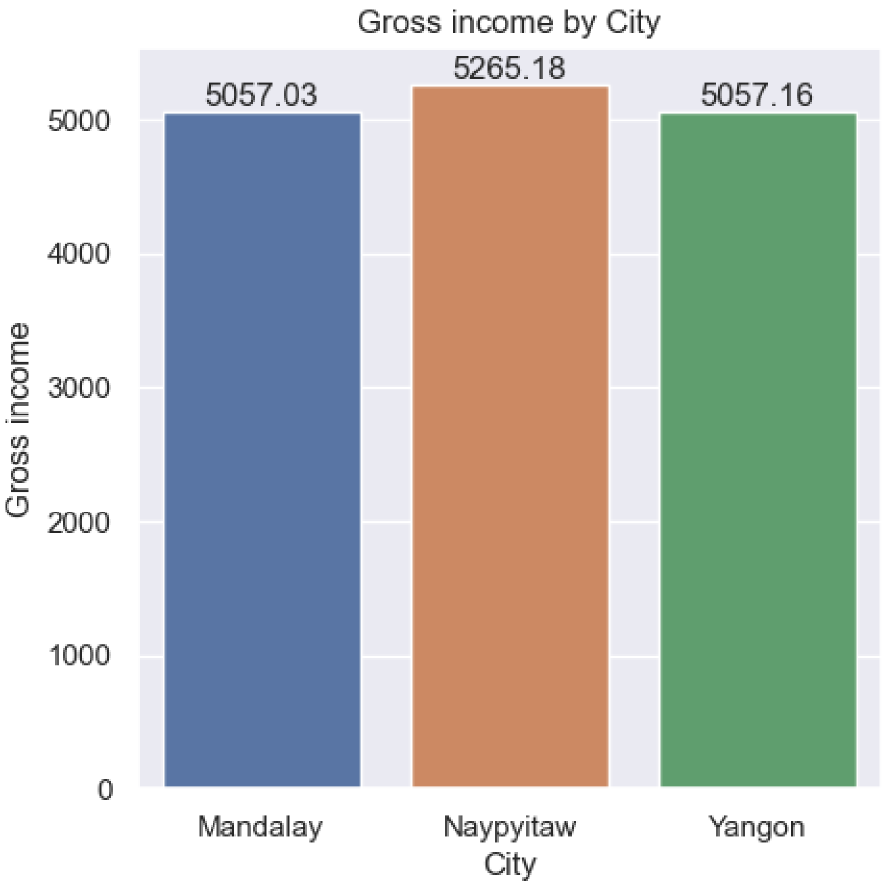
Branch C has slightly more sales and gross margin than other two branches

[75]:

```
sales_city=df.groupby(['City'],as_index=False)['Total'].sum()
ax=sns.barplot(x='City',y='Total',hue='City',data=sales_city)
plt.ylabel('sales')
plt.title('Sales by City')

for bar in ax.containers:
    ax.bar_label(bar)
plt.show()
```



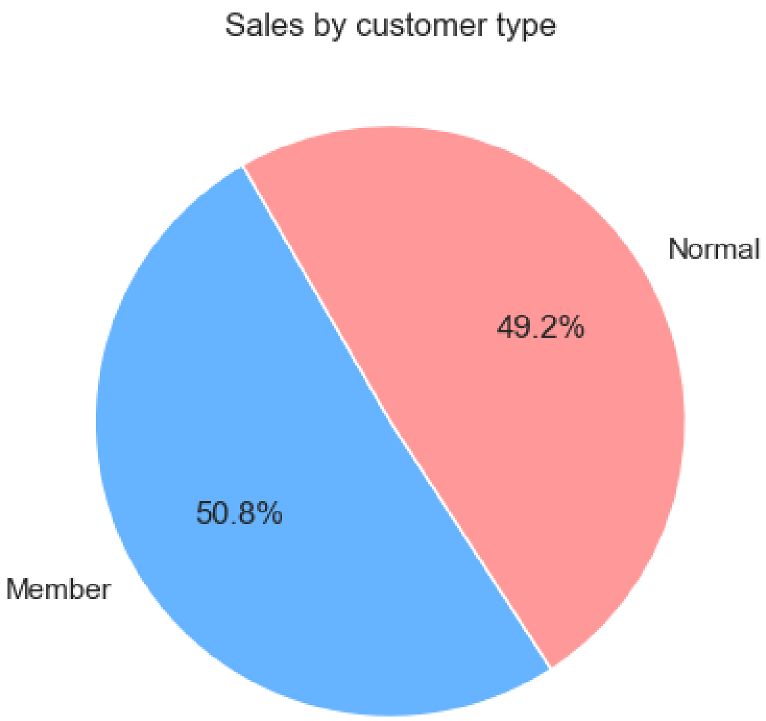
[81]:
```
gross_income_City=df.groupby(['City'],as_index=False)['gross income'].sum()
ax=sns.barplot(x='City',y='gross income',hue='City',data=gross_income_City)
plt.ylabel('Gross income')
plt.title('Gross income by City')

for bar in ax.containers:
    ax.bar_label(bar)
plt.show()
```



Naypyitaw City has slightly more sales and gross margin than other cities

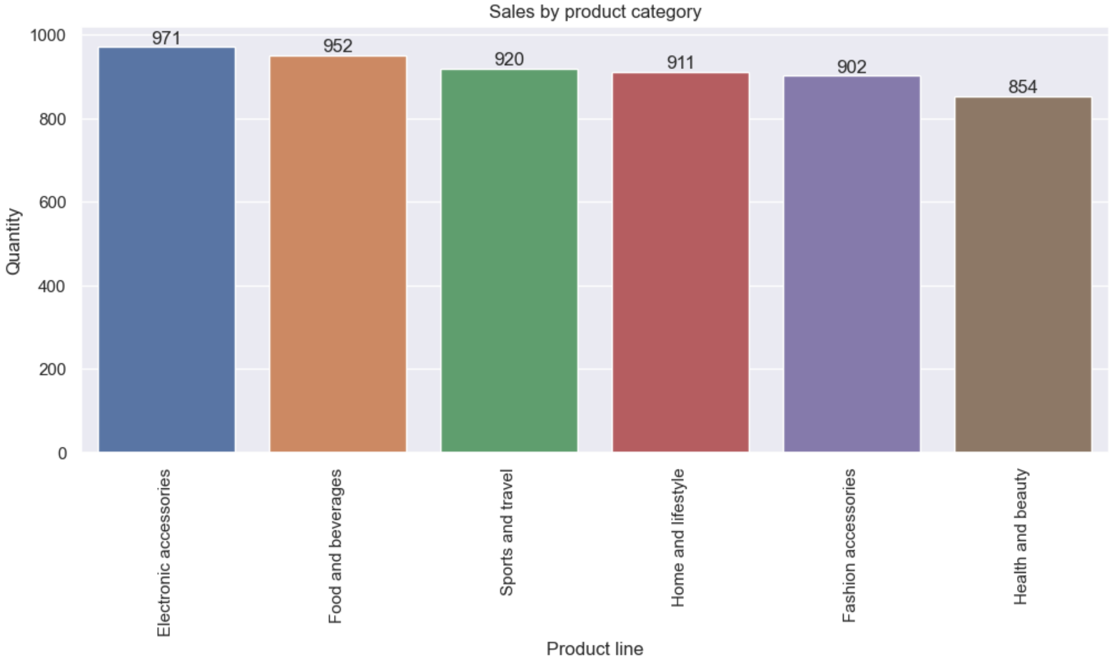
[87]:

```
sales_Customer_type=df.groupby('Customer type')['Total'].sum()
ax=plt.pie(sales_Customer_type,labels=sales_Customer_type.index,autopct='%1.
  ↪1f%%',colors=['#66b3ff', '#ff9999'], startangle=120)
plt.title('Sales by customer type')
plt.show()
```

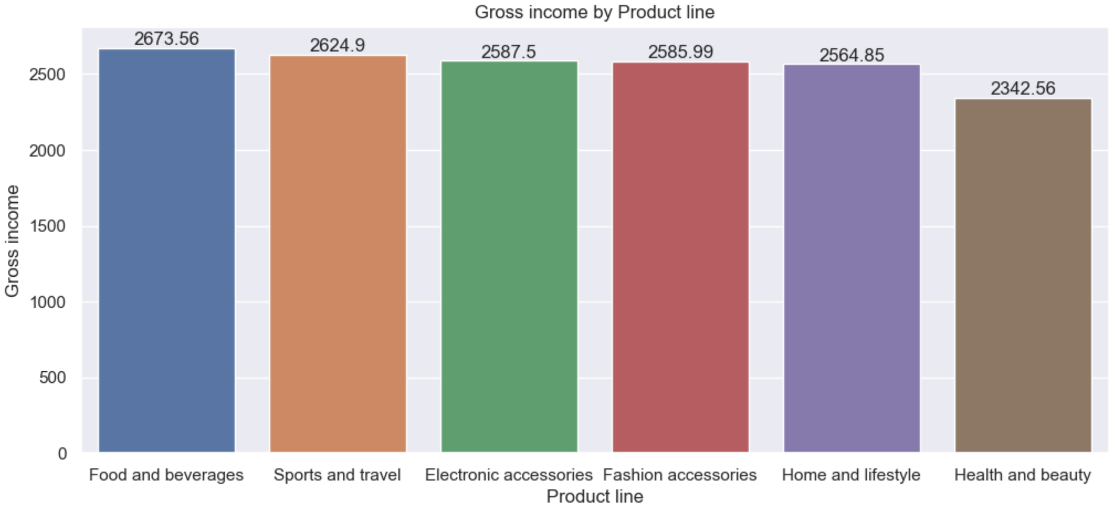


Customer with membership purchased slightly more than normal customer

[92]:
```
sns.set(rc={'figure.figsize':(12,5)})
sales_Product_line=df.groupby(['Product line'],as_index=False)['Quantity'].
 ↪sum().sort_values(by=['Quantity'],ascending=False)
ax=sns.barplot(x='Product line',y='Quantity',hue='Product␣
 ↪line',data=sales_Product_line)
plt.ylabel('Quantity')
plt.title('Sales by product category')
plt.xticks(rotation=90)

for bar in ax.containers:
    ax.bar_label(bar)

plt.show()
```

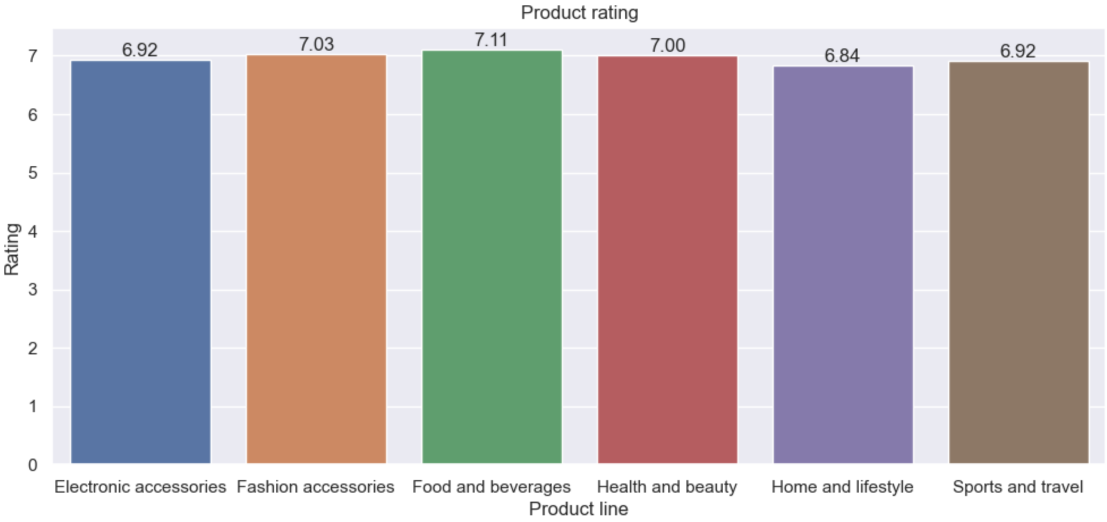


[98]:
```
gross_income_Product_line=df.groupby(['Product line'],as_index=False)['gross␣
 ↪income'].sum().sort_values(by=['gross income'],ascending=False)
ax=sns.barplot(x='Product line',y='gross income',hue='Product␣
 ↪line',data=gross_income_Product_line)
plt.ylabel('Gross income')
plt.title('Gross income by Product line')

for bar in ax.containers:
    ax.bar_label(bar)
plt.show()
```



[122]:

```
product_rating=df.groupby(['Product line'],as_index=False)['Rating'].mean()
ax=sns.barplot(x='Product line',y='Rating',hue='Product␣
 ↪line',data=product_rating)
plt.ylabel('Rating')
plt.title('Product rating')

for bar in ax.containers:
    ax.bar_label(bar,fmt='%.2f')
plt.show()

#ax=sns.barplot(x='Product line',y='Rating',data=df)
#plt.ylabel('Rating')
#plt.title('Product rating')

#for bar in ax.containers:
 #   ax.bar_label(bar,fmt='%.0f')
#plt.show()
```

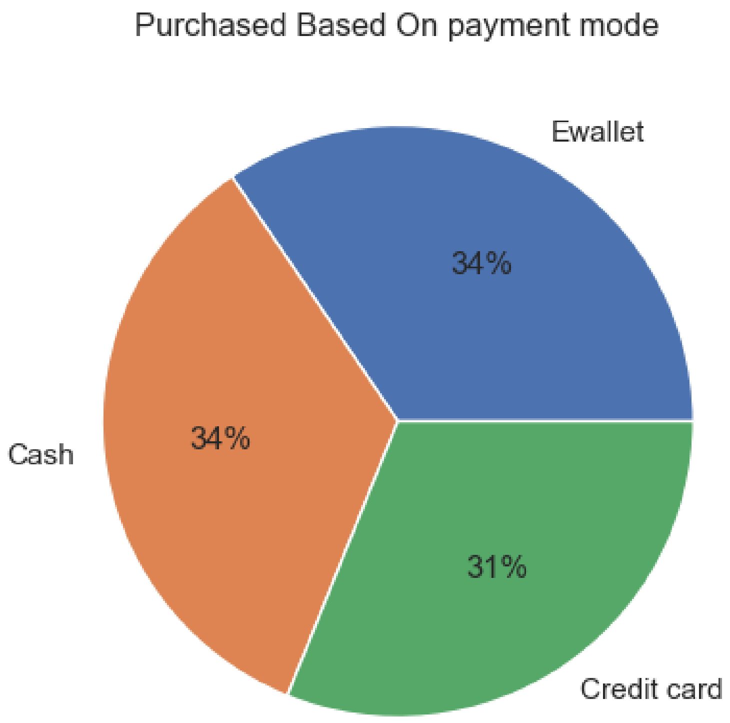


Electronic accessories is slightly more than others but Food & beverages gross margin is more

[107]:

```
purchased_mode_payment=df['Payment'].value_counts()
#purchased_mode_payment
plt.pie(purchased_mode_payment,labels=purchased_mode_payment.index,autopct='%1.
 ↪0f%%')
plt.title('Purchased Based On payment mode')
plt.show()
```

```
#sns.set(rc={'figure.figsize':(5,5)})
#plt.pie(total_pucrhased_gender_counts,labels=total_pucrhased_gender_counts.
 ↪index,autopct='%1.1f%%',colors=['#66b3ff', '#ff9999'], startangle=120)
#plt.title('Purchased Based On Gender')
#plt.show()
```



Payment made by credit card are slightly less as compared to cash and ewallet.

[124]:

```
#df['month']=df['date'].dt.month
df['month'] = df['Date'].dt.month
```

[127]:

```
df['month_name'] = df['Date'].dt.strftime('%B')
```

[128]:

```
df.head()
```

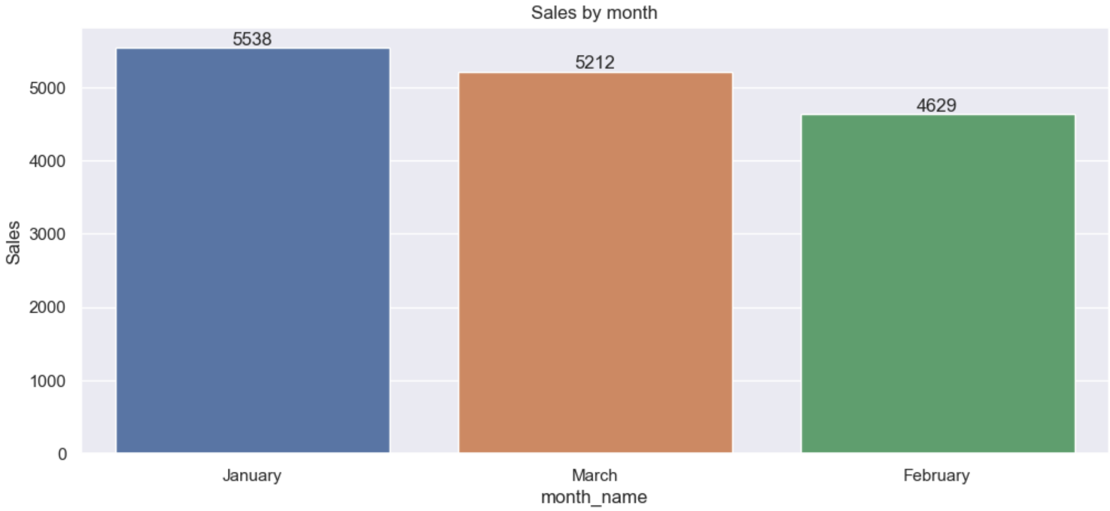
[128]:

```
    Invoice ID Branch       City Customer type  Gender  \
0  750-67-8428      A     Yangon        Member  Female
1  226-31-3081      C  Naypyitaw        Normal  Female
2  631-41-3108      A     Yangon        Normal    Male
3  123-19-1176      A     Yangon        Member    Male
4  373-73-7910      A     Yangon        Normal    Male
```

```
             Product line  Unit price  Quantity   Tax 5%     Total       Date  \
0       Health and beauty       74.69         7  26.1415  548.9715 2019-01-05
1  Electronic accessories       15.28         5   3.8200   80.2200 2019-03-08
2      Home and lifestyle       46.33         7  16.2155  340.5255 2019-03-03
3       Health and beauty       58.22         8  23.2880  489.0480 2019-01-27
4       Sports and travel       86.31         7  30.2085  634.3785 2019-02-08

    Time      Payment    cogs  gross margin percentage  gross income  Rating  \
0  13:08      Ewallet  522.83                 4.761905       26.1415     9.1
1  10:29         Cash   76.40                 4.761905        3.8200     9.6
2  13:23  Credit card  324.31                 4.761905       16.2155     7.4
3  20:33      Ewallet  465.76                 4.761905       23.2880     8.4
4  10:37      Ewallet  604.17                 4.761905       30.2085     5.3

   month month_name
0      1    January
1      3      March
2      3      March
3      1    January
4      2   February
```

[141]:

```
sales_month=df.groupby(['month_name'],as_index=False)['gross income'].sum().
 ↪sort_values(by=['gross income'],ascending=False)
ax=sns.barplot(x='month_name',y='gross␣
 ↪income',hue='month_name',data=sales_month)
plt.ylabel('Sales')
plt.title('Sales by month')

for bar in ax.containers:
    ax.bar_label(bar,fmt='%.0f')
plt.show()
```



In january Sales are more as compared to other month

0.1 Conclusion:

0.2

- Female customers spend slightly more than male customers.
- Branch C has slightly higher sales and gross margin compared to other branches.
- Naypyitaw City leads in sales and gross margin among all cities.
- Customers with membership make slightly more purchases than non-members.
- Electronic accessories category shows higher sales, while Food & Beverages has a higher gross margin.
- Credit card payments are slightly less common compared to cash and e-wallet payments.
- Sales are higher in January compared to other months.

githublink:https://github.com/ShahRizwan007/Supermarket_sales_python_project

linkdin:www.linkedin.com/in/shahrizwan819

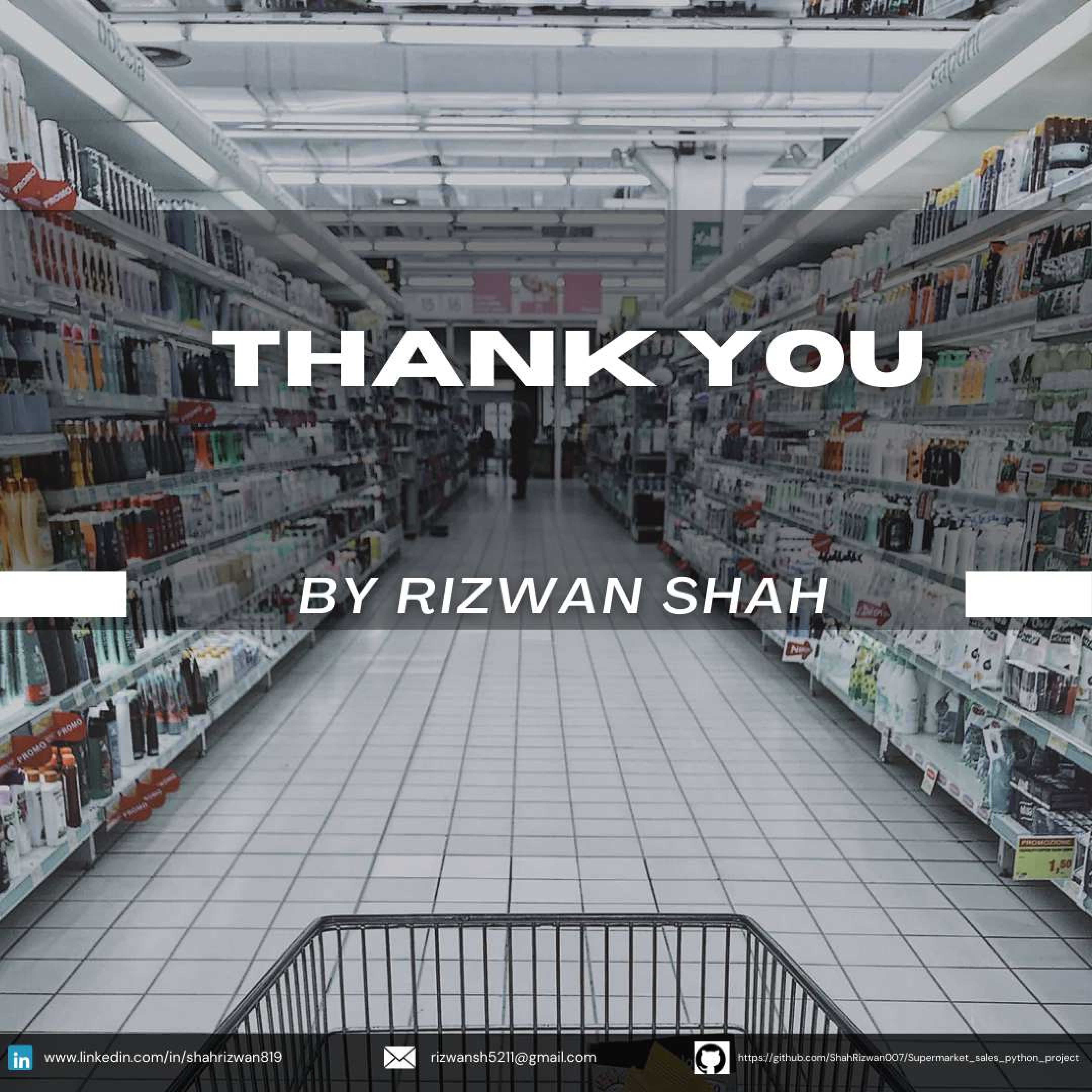
```
THANK YOU
```

[ ]:

```

```

Insights and Observations

- Female customers tend to spend slightly more than male customers.
- Branch C consistently outperforms other branches, showing slightly higher sales and gross margin.
- Naypyitaw City leads in both sales and gross margin among all cities.
- Customers with memberships make slightly more purchases than non-members, indicating a strong engagement with the loyalty program.
- The Electronic Accessories category has the highest sales, while the Food & Beverages category boasts a higher gross margin, suggesting differing profitability across product lines.
- Credit card payments are slightly less common compared to cash and e-wallet transactions, indicating a preference for more immediate payment methods.
- Sales peak in January, surpassing those in other months, which may reflect seasonal shopping trends.



THANK YOU

BY RIZWAN SHAH

www.linkedin.com/in/shahrizwan819

rizwansh5211@gmail.com

https://github.com/ShahRizwan007/Supermarket_sales_python_project