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

%matplotlib inline
```

```
marketSales = pd.read_csv('supermarket_sales - Sheet1.csv')
```

```
marketSales.head()
```

```
    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  
```

```
marketSales.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 17 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Invoice ID               1000 non-null   object 
 1   Branch                   1000 non-null   object 
 2   City                     1000 non-null   object 
 3   Customer type            1000 non-null   object 
 4   Gender                   1000 non-null   object 
 5   Product line             1000 non-null   object 
 6   Unit price               1000 non-null   float64
 7   Quantity                 1000 non-null   int64  
 8   Tax 5%                   1000 non-null   float64
 9   Total                    1000 non-null   float64
 10  Date                     1000 non-null   object
 11  Time                     1000 non-null   object
 12  Payment                  1000 non-null   object
 13  cogs                     1000 non-null   float64
 14  gross margin percentage  1000 non-null   float64
 15  gross income             1000 non-null   float64
 16  Rating                   1000 non-null   float64
dtypes: float64(7), int64(1), object(9)
memory usage: 132.9+ KB
```

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

```
marketSales.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 17 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Invoice ID               1000 non-null   object
 1   Branch                   1000 non-null   object
 2   City                     1000 non-null   object
 3   Customer type            1000 non-null   object
 4   Gender                   1000 non-null   object
 5   Product line             1000 non-null   object
 6   Unit price               1000 non-null   float64
 7   Quantity                 1000 non-null   int64
 8   Tax 5%                   1000 non-null   float64
 9   Total                    1000 non-null   float64
 10  Date                     1000 non-null   datetime64[ns]
 11  Time                     1000 non-null   object
 12  Payment                  1000 non-null   object
 13  cogs                     1000 non-null   float64
 14  gross margin percentage  1000 non-null   float64
 15  gross income             1000 non-null   float64
 16  Rating                   1000 non-null   float64
dtypes: datetime64[ns](1), float64(7), int64(1), object(8)
memory usage: 132.9+ KB
```

```
marketSales.describe()
```

```
        Unit price     Quantity       Tax 5%        Total         cogs  \
count  1000.000000  1000.000000  1000.000000  1000.000000  1000.00000
mean     55.672130     5.510000    15.379369   322.966749   307.58738
std      26.494628     2.923431    11.708825   245.885335   234.17651
min      10.080000     1.000000     0.508500    10.678500    10.17000
25%      32.875000     3.000000     5.924875   124.422375   118.49750
```

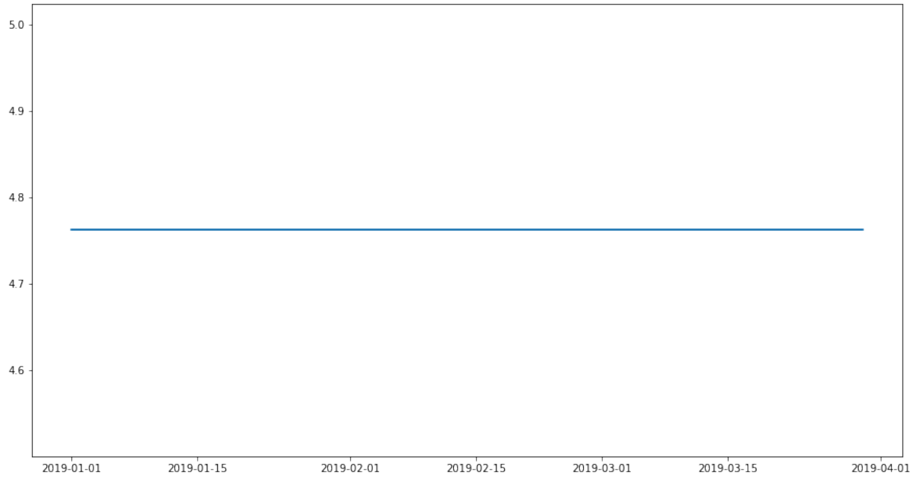
```
50%      55.230000     5.000000    12.088000   253.848000   241.76000
75%      77.935000     8.000000    22.445250   471.350250   448.90500
max      99.960000    10.000000    49.650000  1042.650000   993.00000

       gross margin percentage  gross income      Rating
count              1000.000000   1000.000000  1000.00000
mean                  4.761905     15.379369     6.97270
std                   0.000000     11.708825     1.71858
min                   4.761905      0.508500     4.00000
25%                   4.761905      5.924875     5.50000
50%                   4.761905     12.088000     7.00000
75%                   4.761905     22.445250     8.50000
max                   4.761905     49.650000    10.00000
```

```
marketSales.isnull().sum()
```

```
Invoice ID                 0
Branch                     0
City                       0
Customer type              0
Gender                     0
Product line               0
Unit price                 0
Quantity                   0
Tax 5%                     0
Total                      0
Date                       0
Time                       0
Payment                    0
cogs                       0
gross margin percentage    0
gross income               0
Rating                     0
dtype: int64
```

1- How often does the gross margin percentage flactuates?

```
plt.figure(figsize = (15,8))
plt.plot(marketSales['Date'], marketSales['gross margin percentage'])
plt.show()
```

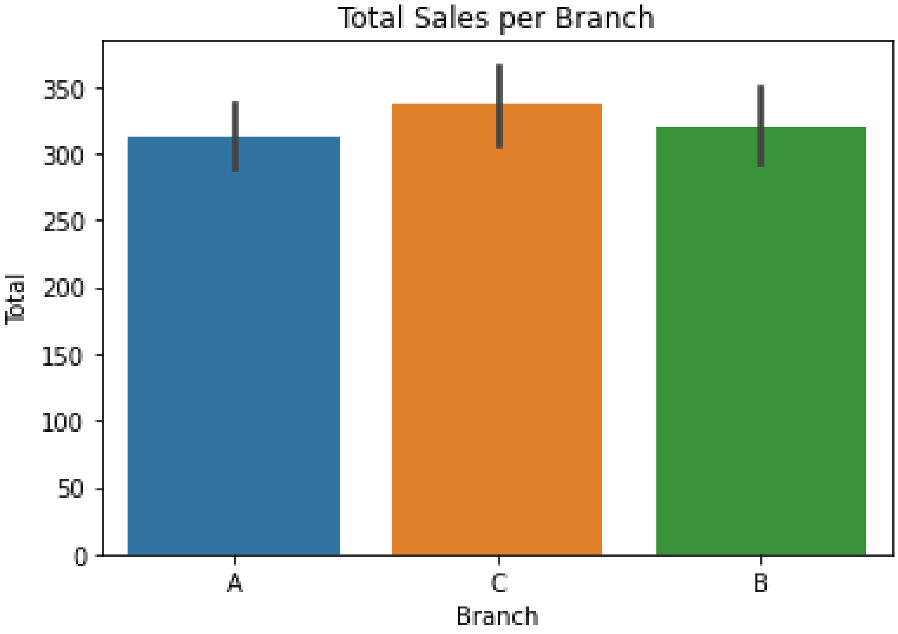


The gross profit percentage is steady indicating good financial health

2- Which branch has the highest sales?

```
sns.barplot(x='Branch', y='Total', data=marketSales).set(title='Total Sales per Branch')
```

```
[Text(0.5, 1.0, 'Total Sales per Branch')]
```



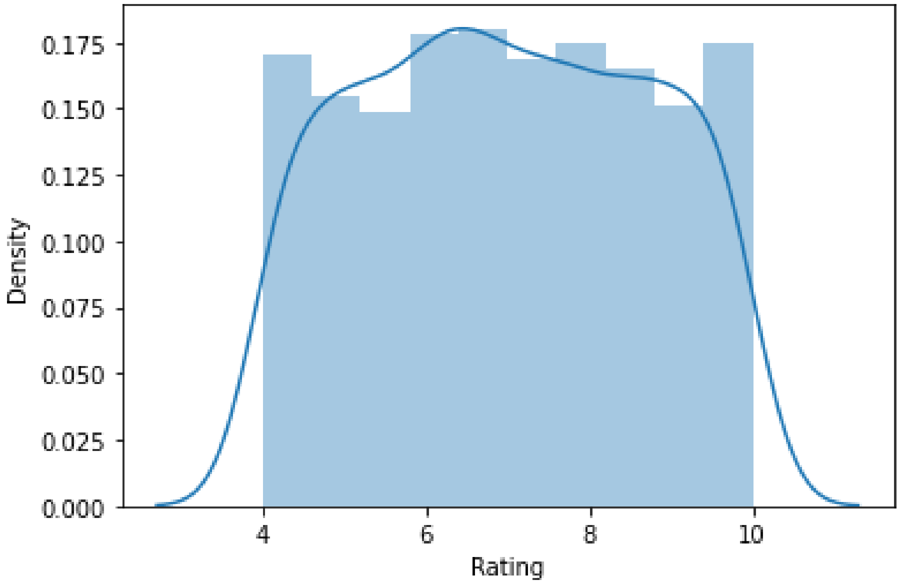
C Branch has the highest sales

3- Are customers satisfied by the services provided? ‘

```
sns.distplot(marketSales['Rating'])
```

```
C:\Users\Nadeen\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\distributi
  warnings.warn(msg, FutureWarning)
```

```
<AxesSubplot:xlabel='Rating', ylabel='Density'>
```

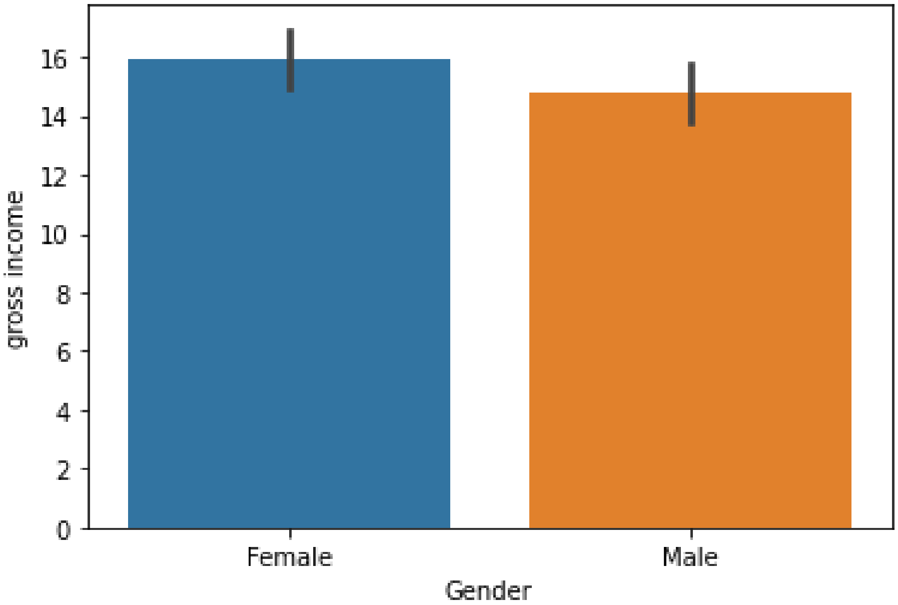


Most customers give ratings 6 and above, therefore; customers are satisfied

Which gender spends more?

```
sns.barplot(x = 'Gender', y = 'gross income', data= marketSales)
```

```
<AxesSubplot:xlabel='Gender', ylabel='gross income'>
```

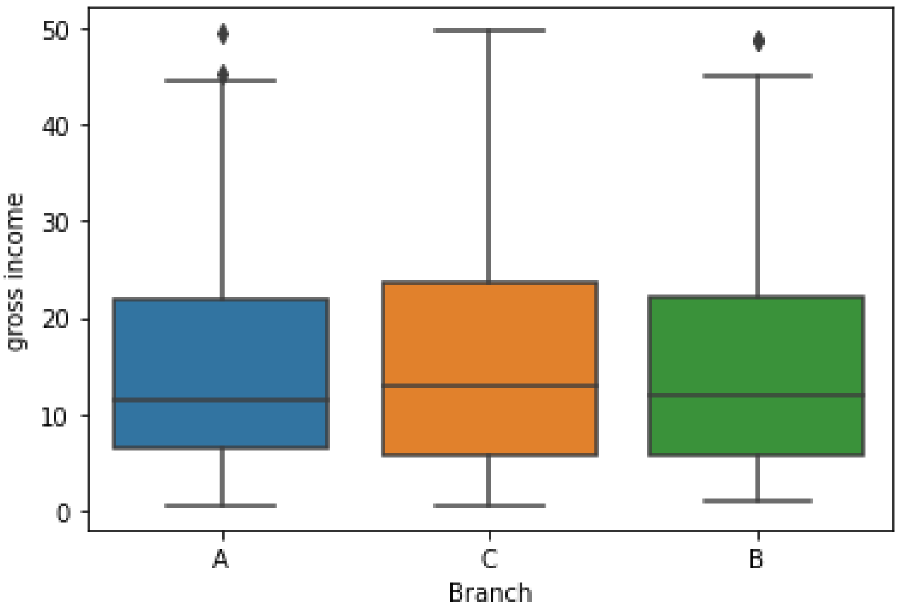


females shoppers spends more than male shoppers

5- Which branch yields the highest profit?

```
sns.boxplot(x = 'Branch', y = 'gross income', data= marketSales)
```

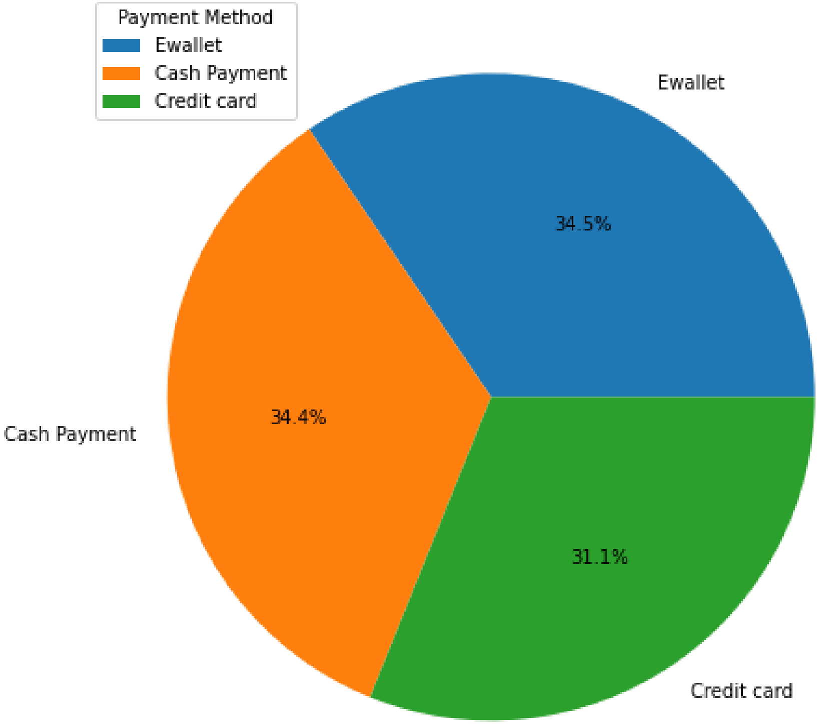
```
<AxesSubplot:xlabel='Branch', ylabel='gross income'>
```



Branch C yield the highest income

6- What is the most common payment method?

```
payment=marketSales.Payment.value_counts()
payment_labels=['Ewallet','Cash Payment', 'Credit card']
plt.figure(figsize = (10,8))
plt.pie(payment,labels=payment_labels,autopct='%.01f%%')
plt.legend(title='Payment Method')
plt.show()
```

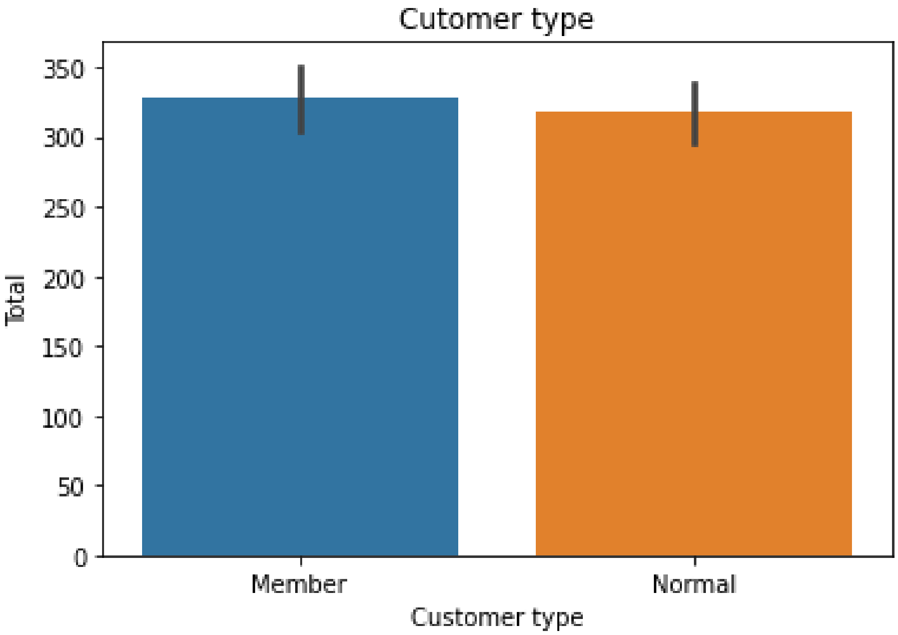


Customers prefer to use Ewallet payment method

7- which cutomer type visits more?

```
sns.barplot(x='Customer type', y='Total', data=marketSales).set(title='Cutomer type')
```

```
[Text(0.5, 1.0, 'Cutomer type')]
```

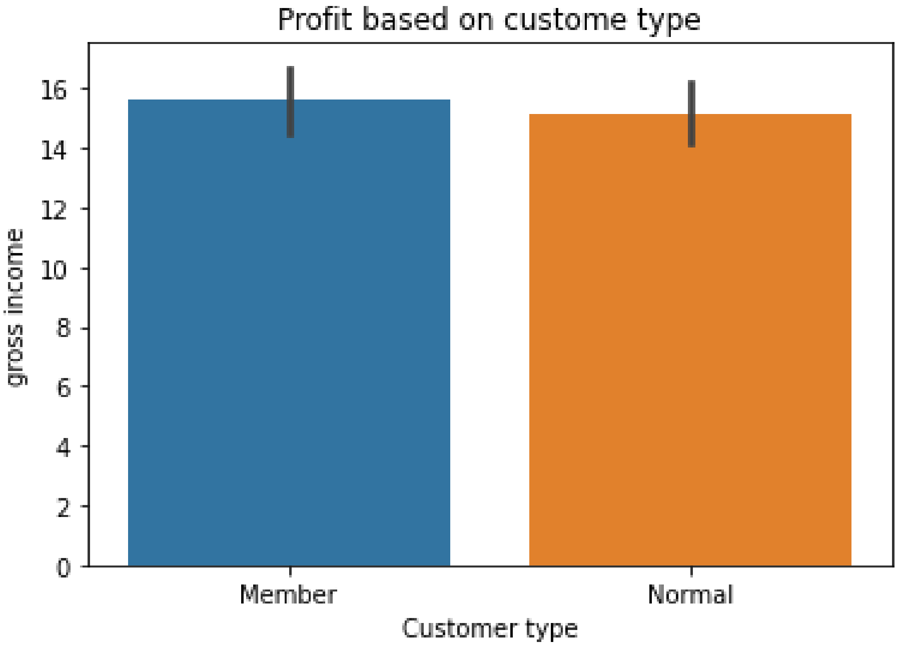


Member customers visit more ofen than normal customers

8- Which customer type is most profitable?

```
sns.barplot(x='Customer type', y='gross income', data=marketSales).set(title='Profit based
```

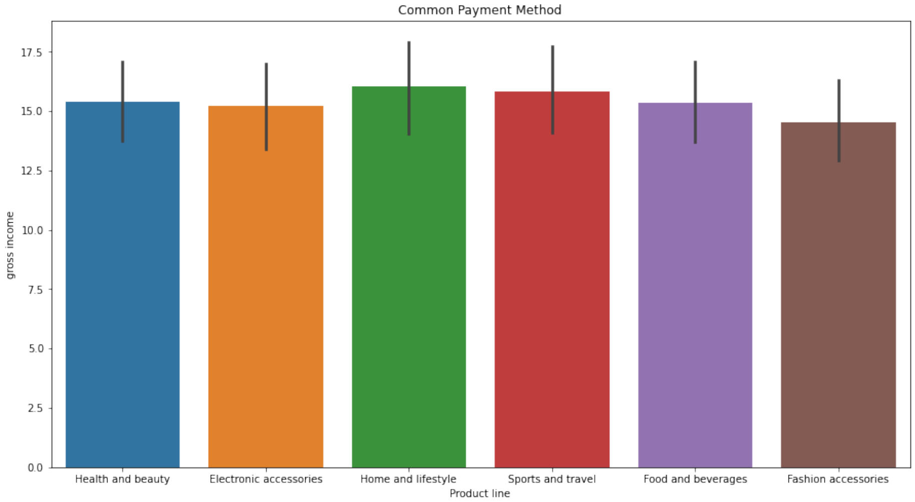
```
[Text(0.5, 1.0, 'Profit based on custome type')]
```



Member customers are mosr profitable

9- Which product line produce the most income

```
plt.figure(figsize = (15,8))
sns.barplot(x='Product line', y='gross income', data=marketSales).set(title='Product Line Pr
plt.show()
```

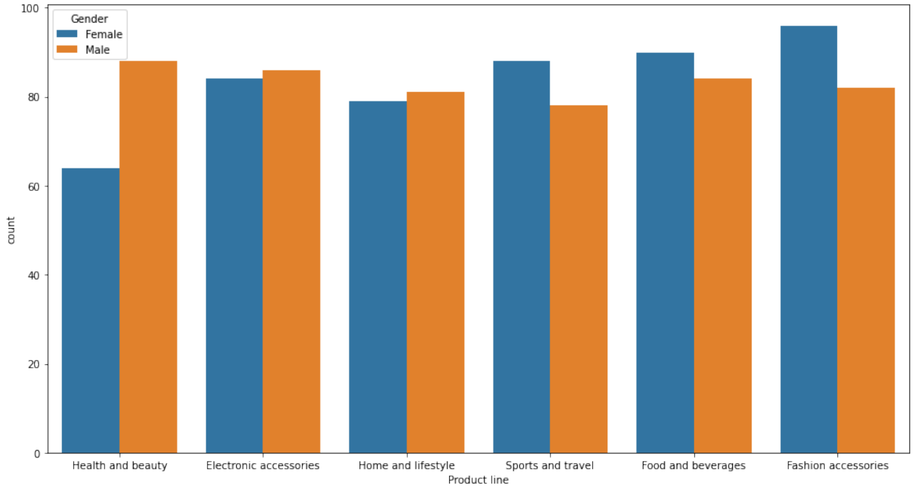


10- In which product line does each gender spends the most?

```
plt.figure(figsize=(15,8))
sns.countplot(marketSales['Product line'],hue = marketSales['Gender'])
```

```
C:\Users\Nadeen\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\_decorato
  warnings.warn(
```

```
<AxesSubplot:xlabel='Product line', ylabel='count'>
```

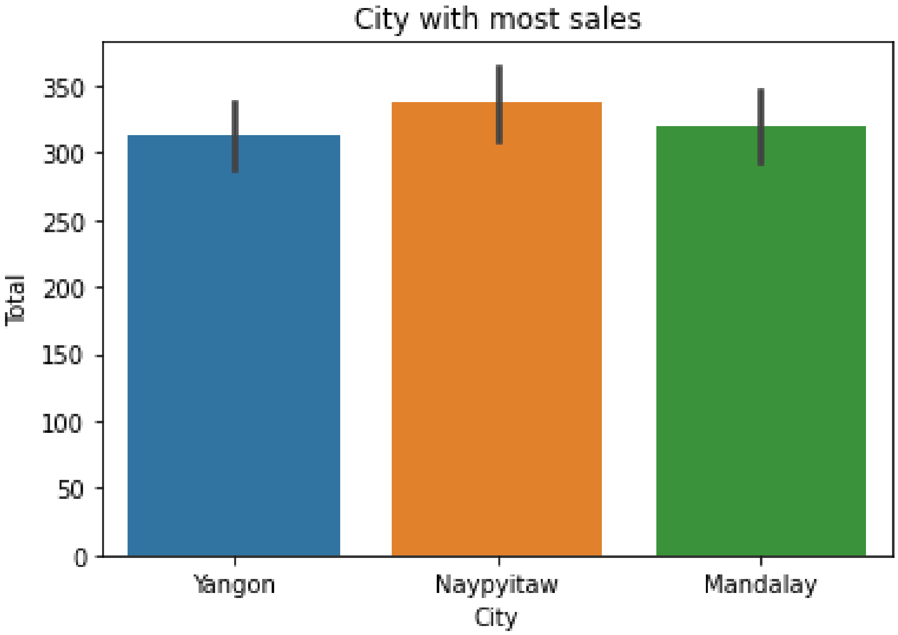


Male customers spend in health and beaty the most ans female customers in fashion accessories

11- Which city has the highest sales?

```
sns.barplot(x='City', y='Total', data=marketSales).set(title='City with most sales')
```

```
[Text(0.5, 1.0, 'City with most sales')]
```

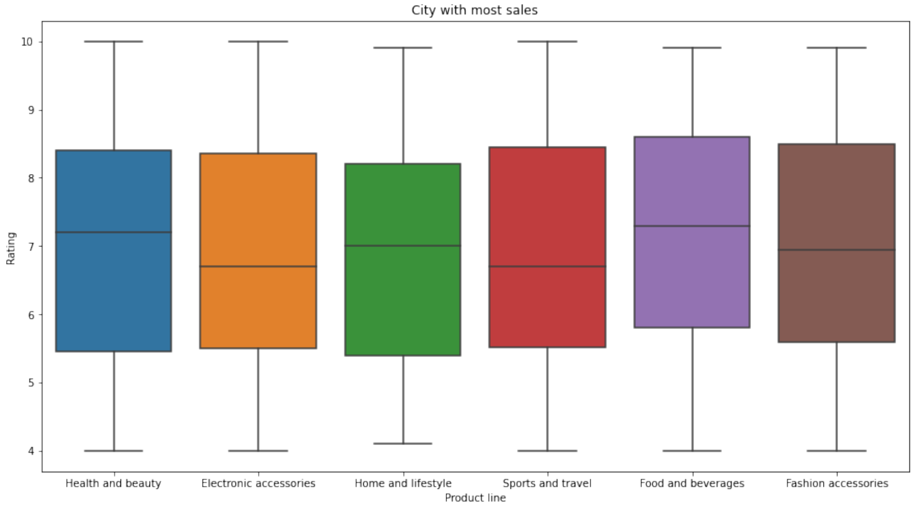


Naypitaw City has the highest sales

12- Which product line has the highest rating

```
plt.figure(figsize=(15,8))
sns.boxplot(x='Product line', y='Rating', data=marketSales).set(title='City with most sales'
```

```
[Text(0.5, 1.0, 'City with most sales')]
```

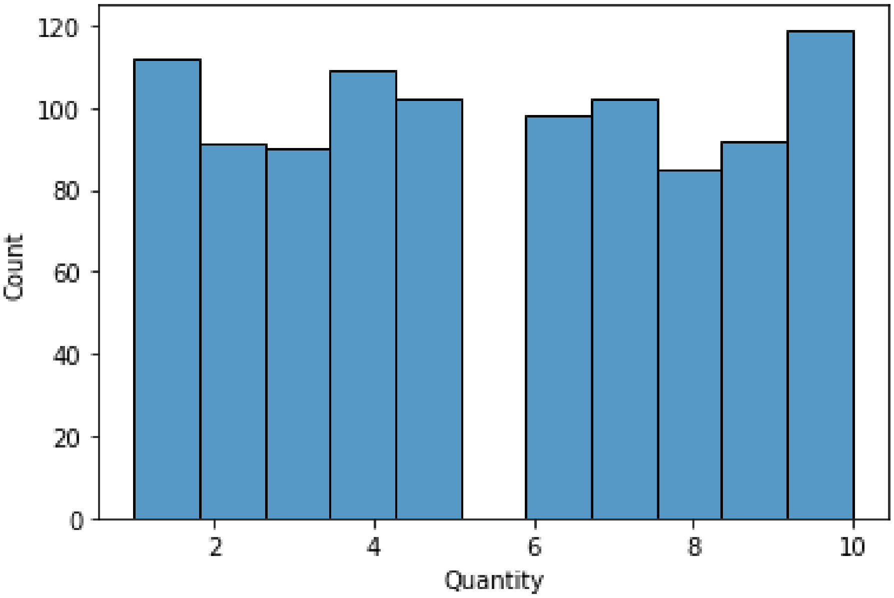


Food and beverages have the highet ratings

13- What is the highest number of items bought by a customer?

```
sns.histplot(marketSales['Quantity'])
```

```
<AxesSubplot:xlabel='Quantity', ylabel='Count'>
```



The highest number of items is 10