```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

df = pd.read_csv('supermarket_sales.csv')
df.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
```

```
df.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

df.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             1.000000e+03   1000.000000  1000.00000
mean              4.761905e+00     15.379369     6.97270
std               6.131498e-14     11.708825     1.71858
min               4.761905e+00      0.508500     4.00000
25%               4.761905e+00      5.924875     5.50000
50%               4.761905e+00     12.088000     7.00000
75%               4.761905e+00     22.445250     8.50000
max               4.761905e+00     49.650000    10.00000
```

```
df.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
```

```
df['Branch'].unique()
```

```
array(['A', 'C', 'B'], dtype=object)
```

```
df['Customer type'].unique()
```

```
array(['Member', 'Normal'], dtype=object)
```

```
df['Product line'].unique()
```

```
array(['Health and beauty', 'Electronic accessories',
       'Home and lifestyle', 'Sports and travel', 'Food and
beverages',
       'Fashion accessories'], dtype=object)
```

```
df['City'].unique()
```

```
array(['Yangon', 'Naypyitaw', 'Mandalay'], dtype=object)
```

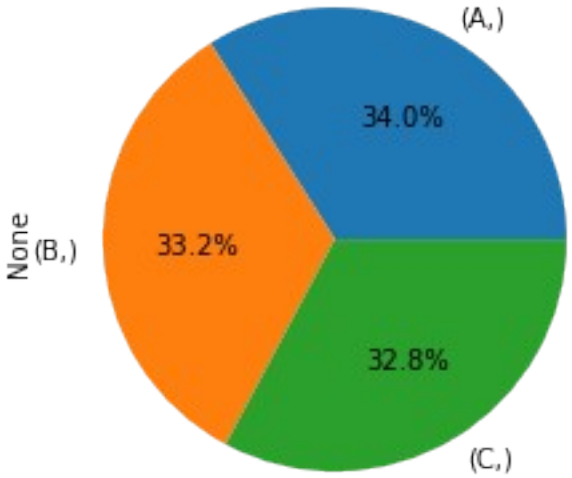
```
df['City'].nunique()
```

```
3
```

```
Branch= df[['Branch']].value_counts()
```

```
Branch.plot.pie(autopct='%1.1f%%')
```

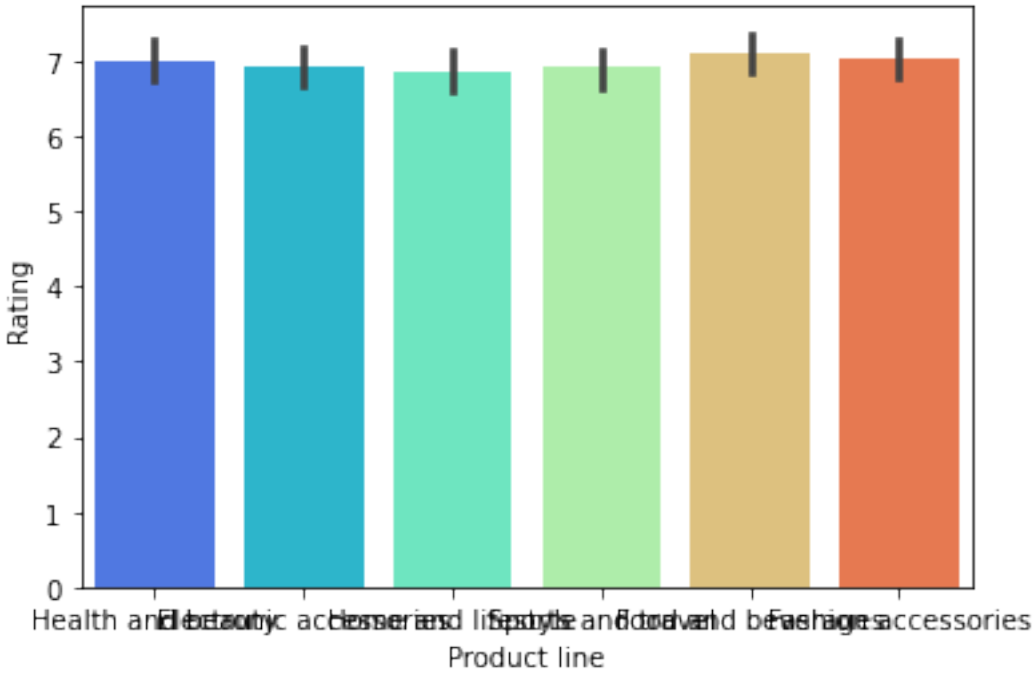
```
<AxesSubplot:ylabel='None'>
```



#Product line vs Rating

```
sns.barplot(x=df['Product line'],y=df['Rating'],palette='rainbow')
```

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



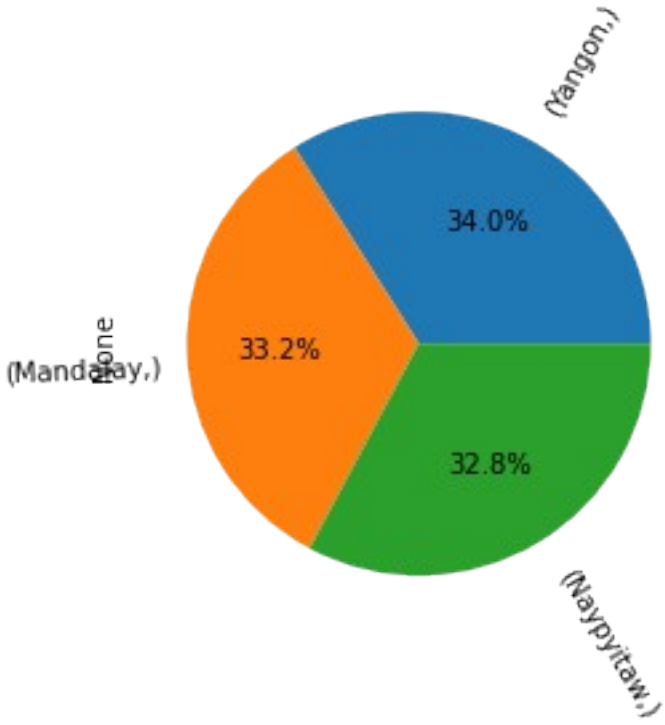

```
df['Product line'].value_counts()
```

```
Fashion accessories       178
Food and beverages        174
Electronic accessories    170
Sports and travel         166
Home and lifestyle        160
Health and beauty         152
Name: Product line, dtype: int64
```

```
City= df[['City']].value_counts()

City.plot.pie(autopct='%1.1f%%',rotatelabels=True)
```

```
<AxesSubplot:ylabel='None'>
```




```
df.groupby('City').sum().sort_values('Rating').tail()
```

```
           Unit price  Quantity     Tax 5%        Total       cogs  \
City
Mandalay     18478.88      1820  5057.0320  106197.6720  101140.64
Naypyitaw    18567.76      1831  5265.1765  110568.7065  105303.53
Yangon       18625.49      1859  5057.1605  106200.3705  101143.21

           gross margin percentage  gross income  Rating
City
Mandalay                1580.952381     5057.0320  2263.6
Naypyitaw               1561.904762     5265.1765  2319.9
Yangon                  1619.047619     5057.1605  2389.2
```

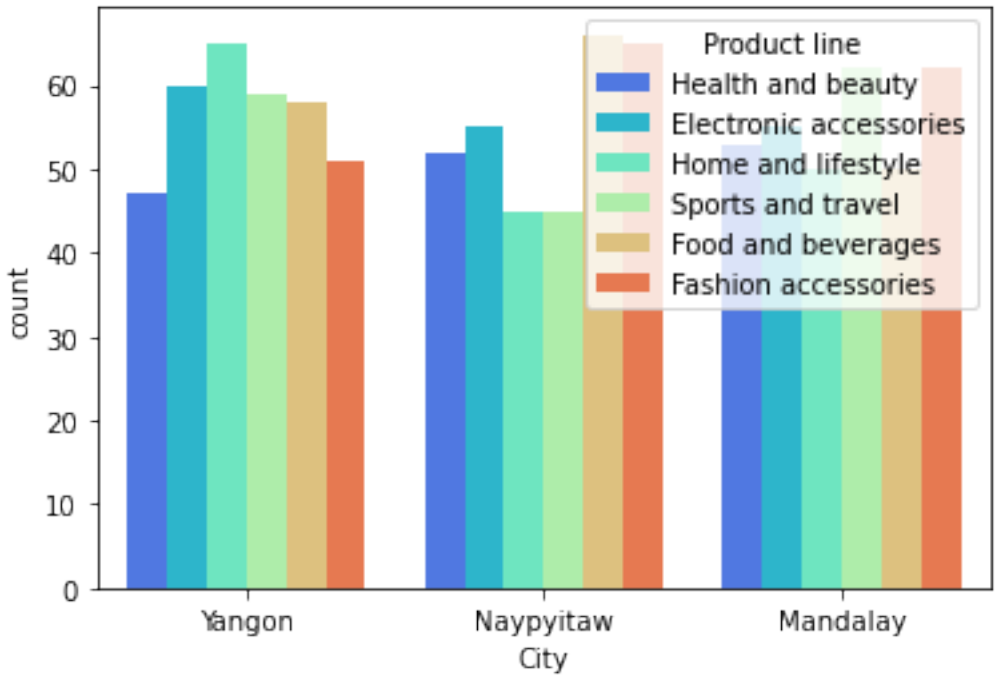
```
newdf = df[df['City'].isin(['Mandalay','Naypyitaw','Yangon'])]
newdf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 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: 140.6+ KB
```

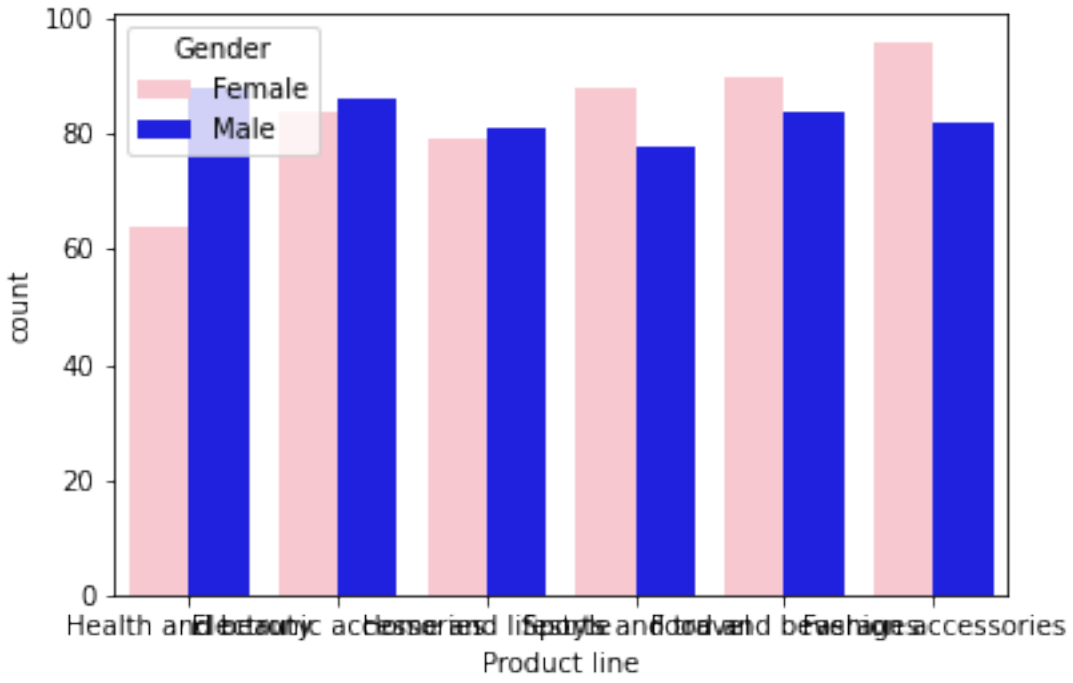
```
sns.countplot(x='City', hue='Product
line',data=newdf,palette='rainbow')
```

```
<AxesSubplot:xlabel='City', ylabel='count'>
```




```
sns.countplot(x='Product line' ,hue = 'Gender', data
=df,palette=['pink','blue','red','teal'])
```

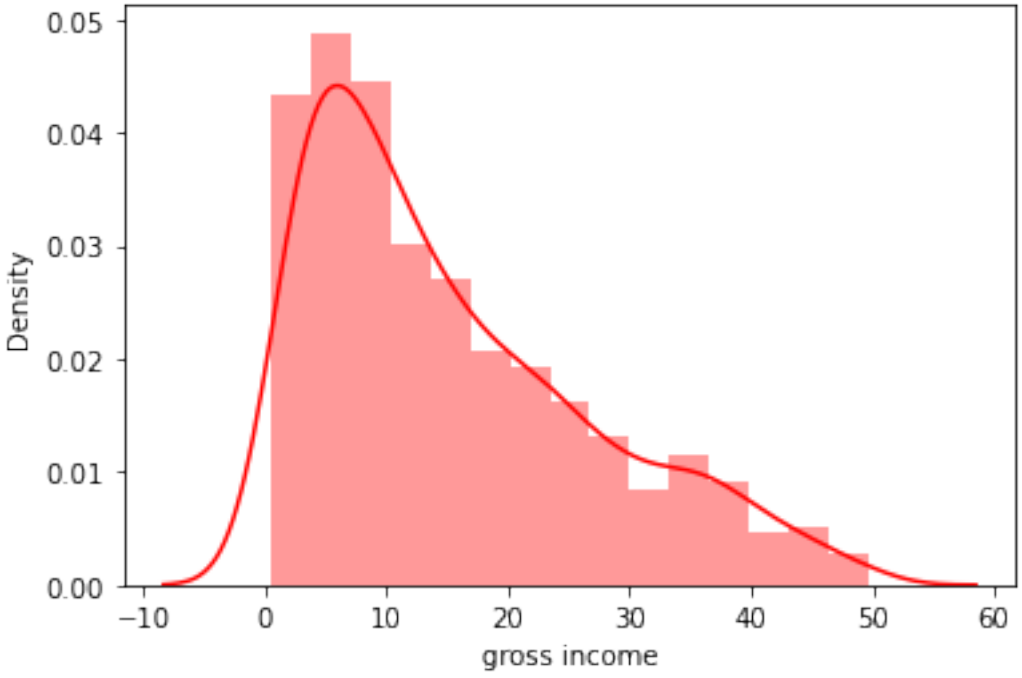
```
<AxesSubplot:xlabel='Product line', ylabel='count'>
```




```
#from Health and beauty,Electronic accessories etc Female customers are more
```

```
sns.distplot(df['gross income'], color='red')
```

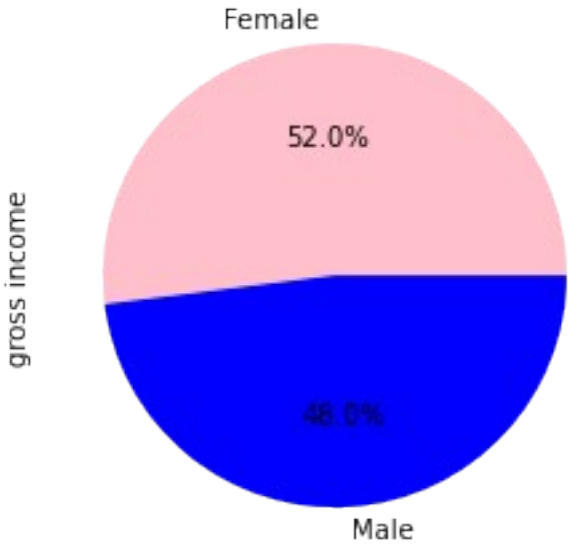
```
<AxesSubplot:xlabel='gross income', ylabel='Density'>
```



```
Gender = df.groupby('Gender').sum()

Gender.plot.pie(autopct='%1.1f%%', y='gross
income',colors=['pink','blue','yellow'],legend=False)
```

```
<AxesSubplot:ylabel='gross income'>
```

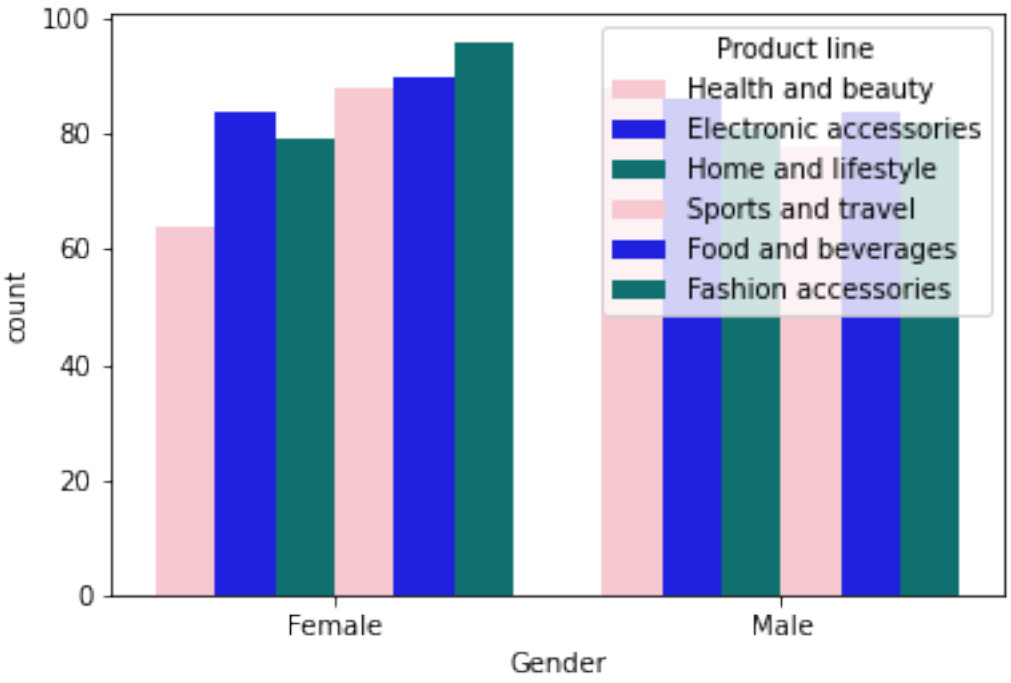



```
df['Product line'].unique()
```

```
array(['Health and beauty', 'Electronic accessories',
       'Home and lifestyle', 'Sports and travel', 'Food and
beverages',
       'Fashion accessories'], dtype=object)
```

```
sns.countplot(x='Gender' ,hue = 'Product line', data
=df,palette=['pink','blue','teal'])
```

```
<AxesSubplot:xlabel='Gender', ylabel='count'>
```


Product line
Health and beauty
Electronic accessories
Home and lifestyle
Sports and travel
Food and beverages
Fashion accessories
100
80
60
40
20
0
count
Female
Male
Gender