

analysis-of-superstore

May 4, 2023

```
[1]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
import numpy as np
df=pd.read_csv('supermarket_sales - Sheet1.csv')
```

```
[4]: df.head(10)
```

```
[4]: 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
5  699-14-3026      C  Naypyitaw      Normal  Male
6  355-53-5943      A      Yangon      Member  Female
7  315-22-5665      C  Naypyitaw      Normal  Female
8  665-32-9167      A      Yangon      Member  Female
9  692-92-5582      B  Mandalay      Member  Female
```

```
Product line Unit price Quantity Tax 5% Net sales \
0 Health and beauty 74.69 7 26.1415 548.9715
1 Electronic accessories 15.28 5 3.8200 80.2200
2 Home and lifestyle 46.33 7 16.2155 340.5255
3 Health and beauty 58.22 8 23.2880 489.0480
4 Sports and travel 86.31 7 30.2085 634.3785
5 Electronic accessories 85.39 7 29.8865 627.6165
6 Electronic accessories 68.84 6 20.6520 433.6920
7 Home and lifestyle 73.56 10 36.7800 772.3800
8 Health and beauty 36.26 2 3.6260 76.1460
9 Food and beverages 54.84 3 8.2260 172.7460
```

```
Date Time Payment cogs gross margin percentage \
0 1/5/2019 13:08 Ewallet 522.83 4.761905
1 3/8/2019 10:29 Cash 76.40 4.761905
2 3/3/2019 13:23 Credit card 324.31 4.761905
3 1/27/2019 20:33 Ewallet 465.76 4.761905
```

| | | | | | |
|---|-----------|-------|-------------|--------|----------|
| 4 | 2/8/2019 | 10:37 | Ewallet | 604.17 | 4.761905 |
| 5 | 3/25/2019 | 18:30 | Ewallet | 597.73 | 4.761905 |
| 6 | 2/25/2019 | 14:36 | Ewallet | 413.04 | 4.761905 |
| 7 | 2/24/2019 | 11:38 | Ewallet | 735.60 | 4.761905 |
| 8 | 1/10/2019 | 17:15 | Credit card | 72.52 | 4.761905 |
| 9 | 2/20/2019 | 13:27 | Credit card | 164.52 | 4.761905 |

| | gross income | Rating |
|---|--------------|--------|
| 0 | 26.1415 | 9.1 |
| 1 | 3.8200 | 9.6 |
| 2 | 16.2155 | 7.4 |
| 3 | 23.2880 | 8.4 |
| 4 | 30.2085 | 5.3 |
| 5 | 29.8865 | 4.1 |
| 6 | 20.6520 | 5.8 |
| 7 | 36.7800 | 8.0 |
| 8 | 3.6260 | 7.2 |
| 9 | 8.2260 | 5.9 |

```
[5]: df.rename(columns = {'Total':'Net sales'}, inplace = True)
```

```
[6]: df.columns
```

```
[6]: Index(['Invoice ID', 'Branch', 'City', 'Customer type', 'Gender',
          'Product line', 'Unit price', 'Quantity', 'Tax 5%', 'Net sales', 'Date',
          'Time', 'Payment', 'cogs', 'gross margin percentage', 'gross income',
          'Rating'],
          dtype='object')
```

```
[9]: df.isnull().sum()
```

```
[9]: Invoice ID      0
     Branch        0
     City          0
     Customer type  0
     Gender        0
     Product line   0
     Unit price     0
     Quantity      0
     Tax 5%        0
     Net sales     0
     Date          0
     Time          0
     Payment       0
     cogs          0
     gross margin percentage  0
     gross income  0
```

```
Rating
dtype: int64
```

```
[18]: 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
```

```
[19]: df.max()
```

```
[19]: Invoice ID             898-04-2717
      Branch              C
      City              Yangon
      Customer type      Normal
      Gender             Male
      Product line       Sports and travel
      Unit price         99.96
      Quantity           10
      Tax 5%             49.65
      Total              1042.65
      Date               3/9/2019
      Time               20:59
      Payment            Ewallet
      cogs               993.0
      gross margin percentage 4.761905
```

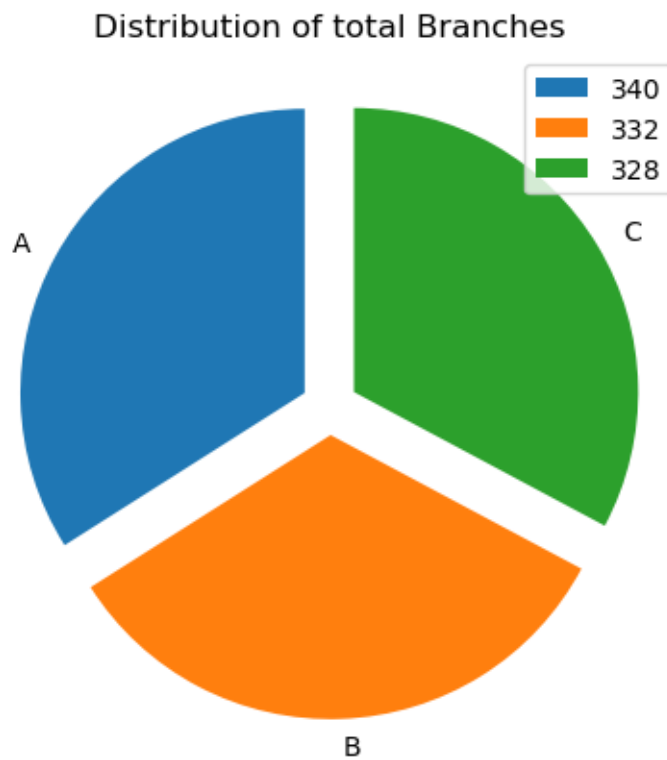
```
gross income          49.65
Rating                10.0
dtype: object
```

```
[25]: df['Branch'].value_counts()
```

```
[25]: A    340
      B    332
      C    328
      Name: Branch, dtype: int64
```

```
[36]: x=df['Branch'].value_counts()
      plt.pie(x,labels=['A','B','C'],startangle=90,explode=[0.1,0.1,0.1])
      plt.title('Distribution of total Branches')
      plt.legend(x)
```

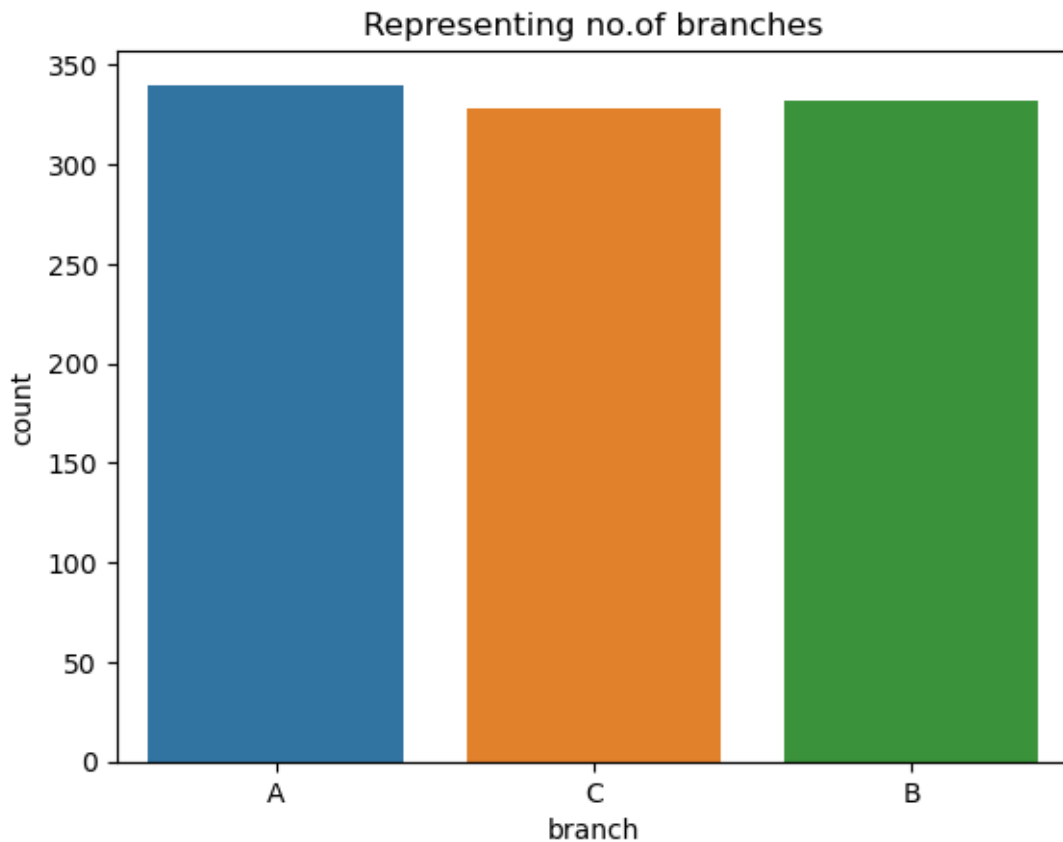
```
[36]: <matplotlib.legend.Legend at 0x25d7d62a970>
```



```
[40]: x=df['Branch']
      sb.countplot(x)
      plt.xlabel('branch')
      plt.title('Representing no.of branches')
```

```
C:\Users\Mokshogna Teja\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variable as a keyword arg: x. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

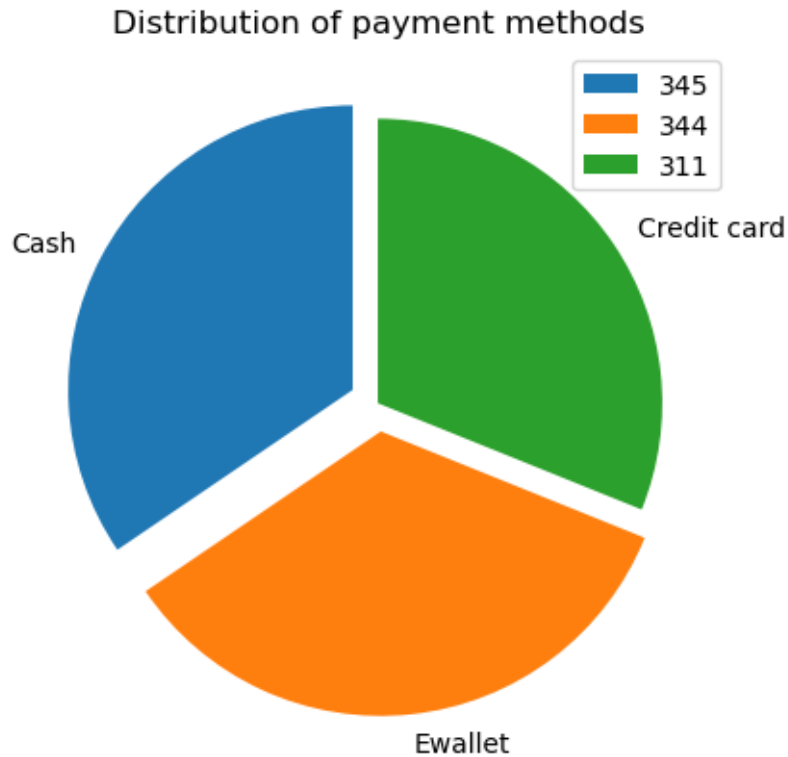
```
[40]: Text(0.5, 1.0, 'Representing no.of branches')
```



```
[70]: x=df['Payment'].value_counts()
```

```
[47]: x=df['Payment'].value_counts()
plt.pie(x,labels=['Cash','Ewallet','Credit card'],startangle=90,explode=[0.1,0.
↪1,0])
plt.title('Distribution of payment methods')
plt.legend(x)
```

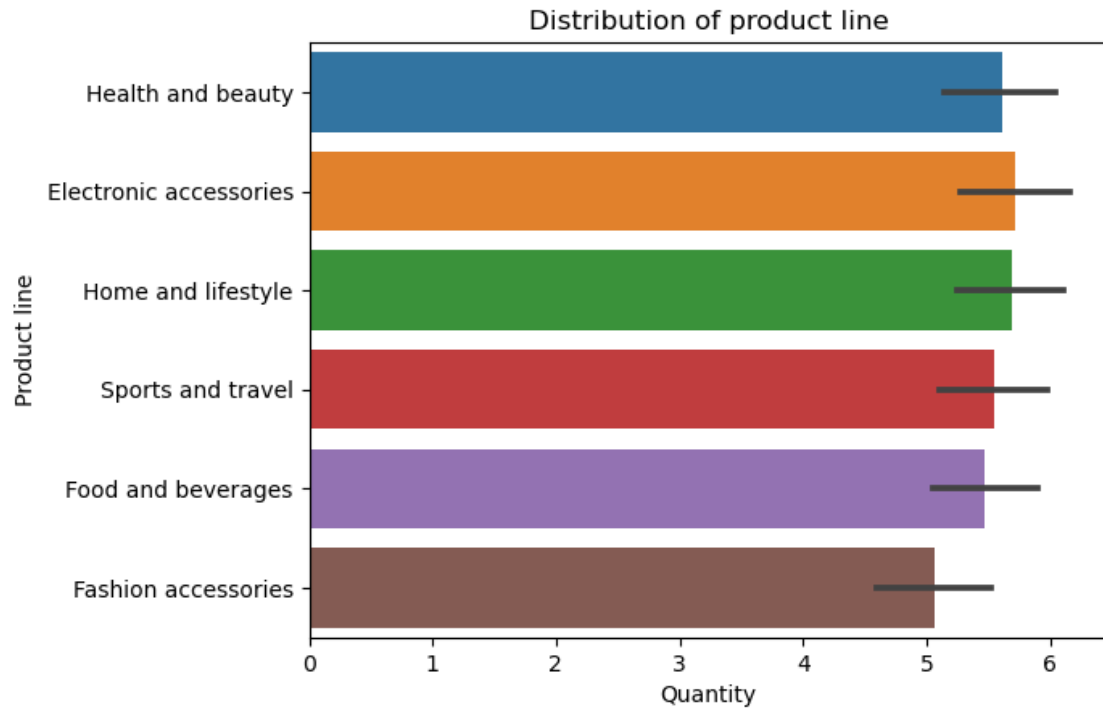
```
[47]: <matplotlib.legend.Legend at 0x25d7f17d340>
```



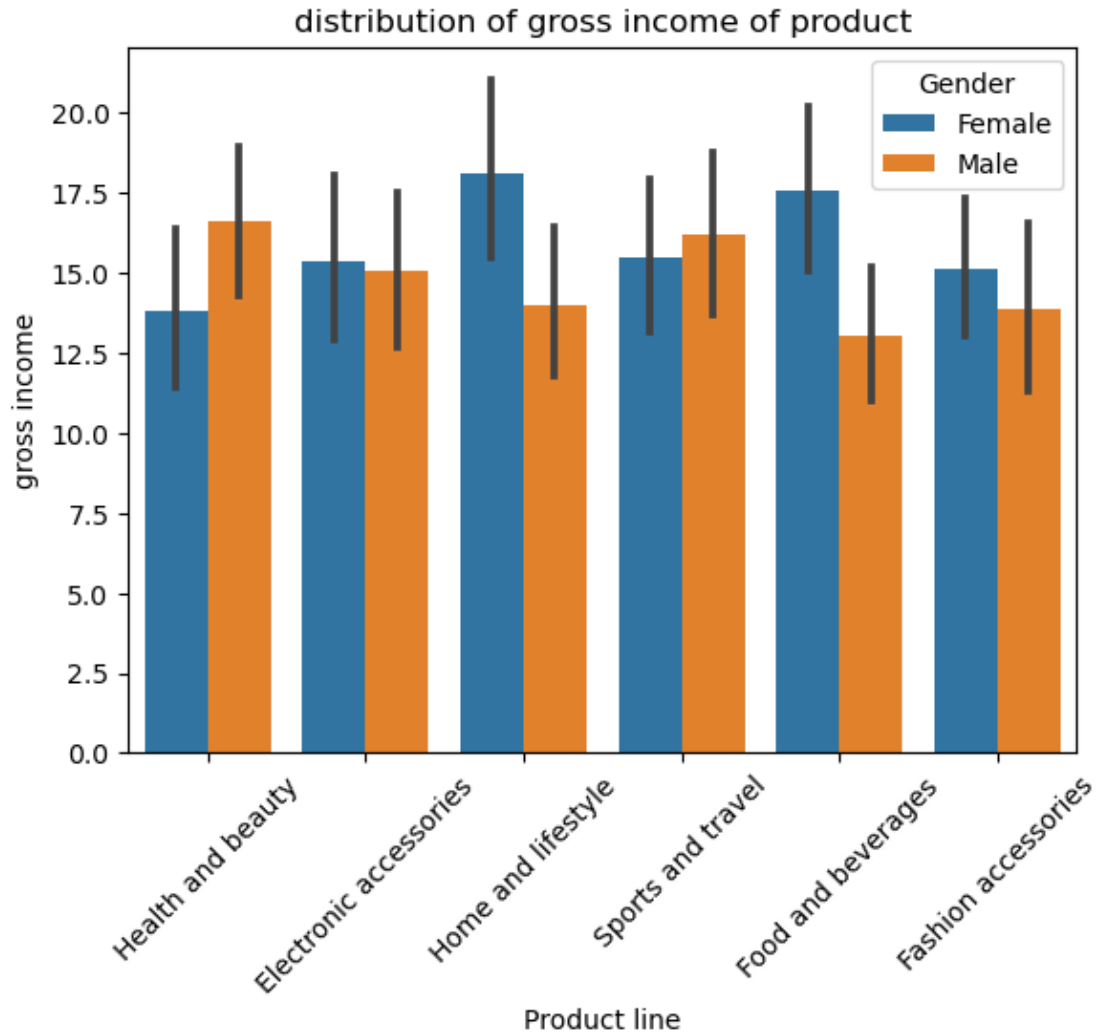
```
[40]: y=df['Product line']
      x=df['Quantity']
      sb.barplot(x,y)
      plt.xlabel('Quantity')
      plt.ylabel('Product line')
      plt.title('Distribution of product line')
```

C:\Users\Mokshogna Teja\anaconda3\lib\site-packages\seaborn_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
warnings.warn(

```
[40]: Text(0.5, 1.0, 'Distribution of product line')
```

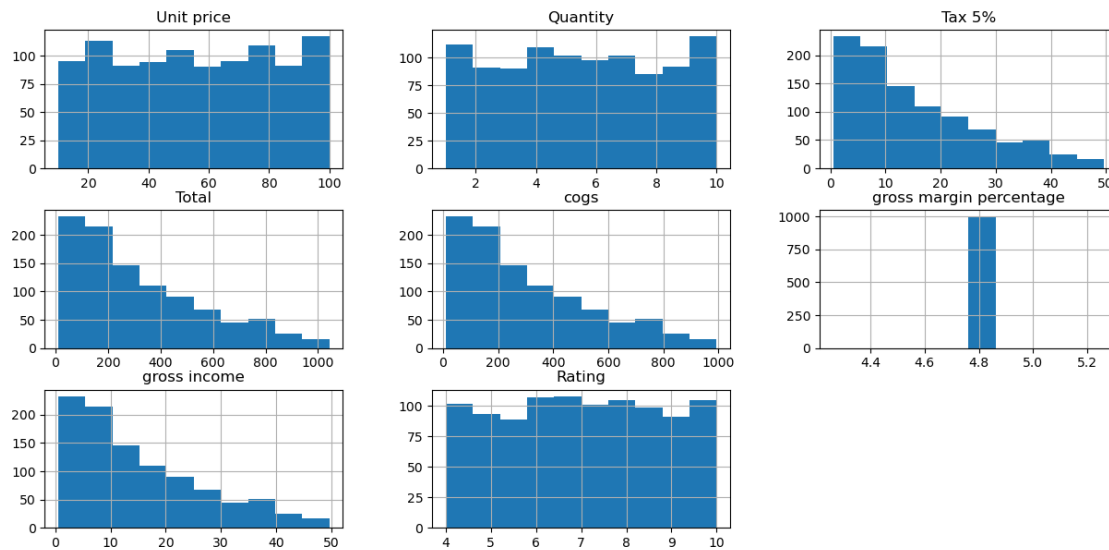


```
[93]: sb.barplot(data = df, x = "Product line", y = "gross income", hue = "Gender")
plt.xticks(rotation = 45)
plt.title('distribution of gross income of product')
plt.show()
```



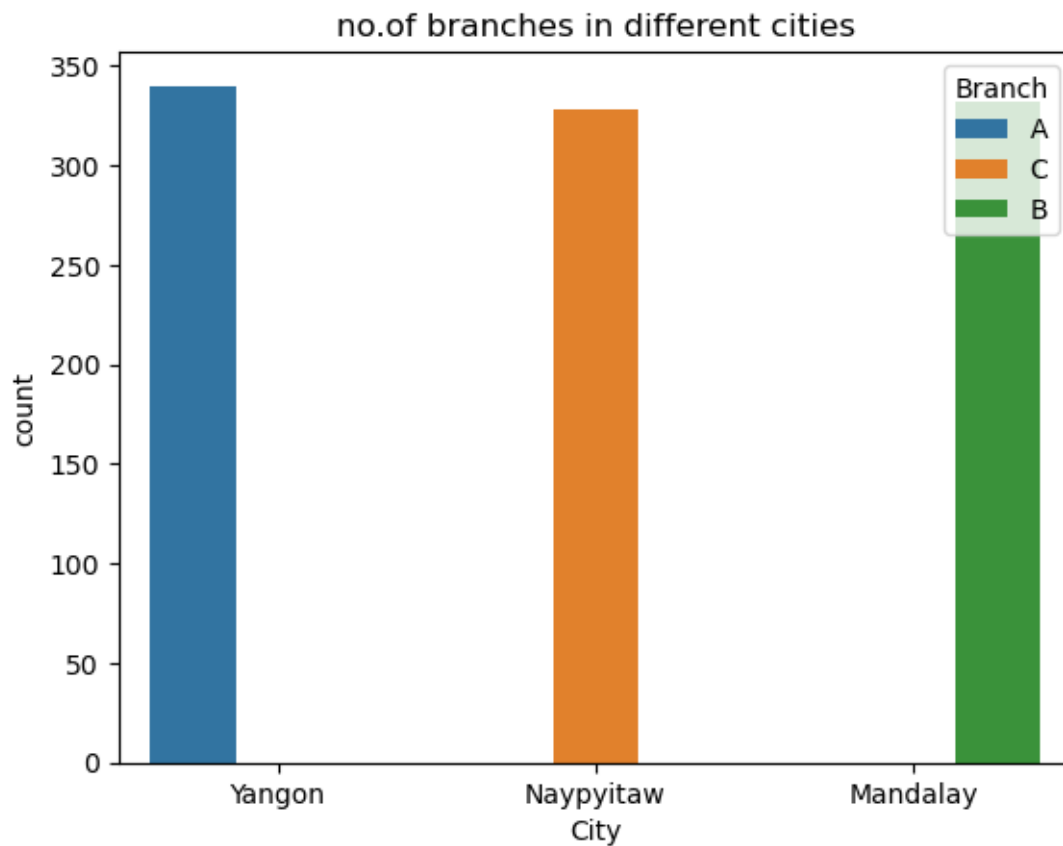
```
[94]: df.hist(figsize = (15, 7))
```

```
[94]: array([[<AxesSubplot:title={'center':'Unit price'}>,
<AxesSubplot:title={'center':'Quantity'}>,
<AxesSubplot:title={'center':'Tax 5%'}>],
[<AxesSubplot:title={'center':'Total'}>,
<AxesSubplot:title={'center':'cogs'}>,
<AxesSubplot:title={'center':'gross margin percentage'}>],
[<AxesSubplot:title={'center':'gross income'}>,
<AxesSubplot:title={'center':'Rating'}>, <AxesSubplot:>]],
dtype=object)
```

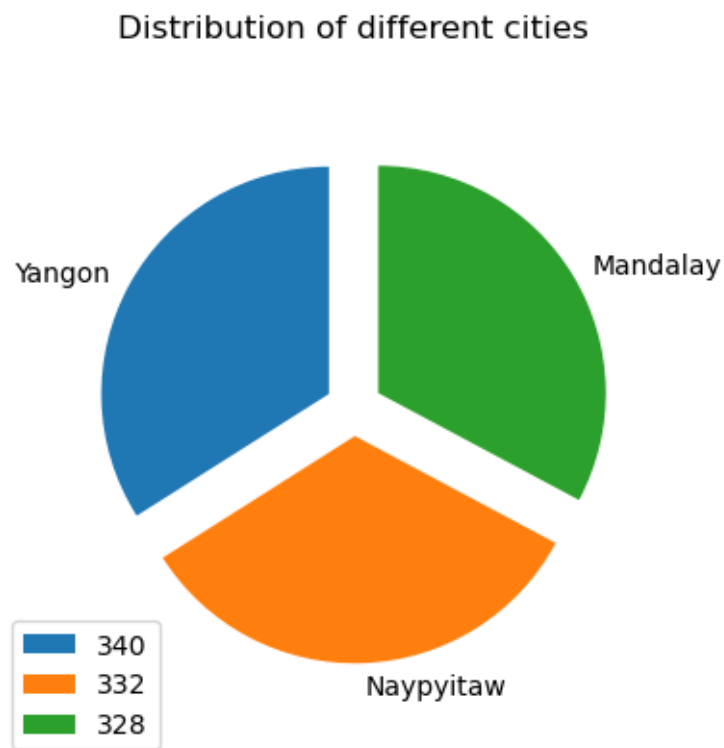
```
[102]: sb.countplot(data= df, x='City', hue= 'Branch')
plt.title('no.of branches in different cities')
```

```
[102]: Text(0.5, 1.0, 'no.of branches in different cities')
```



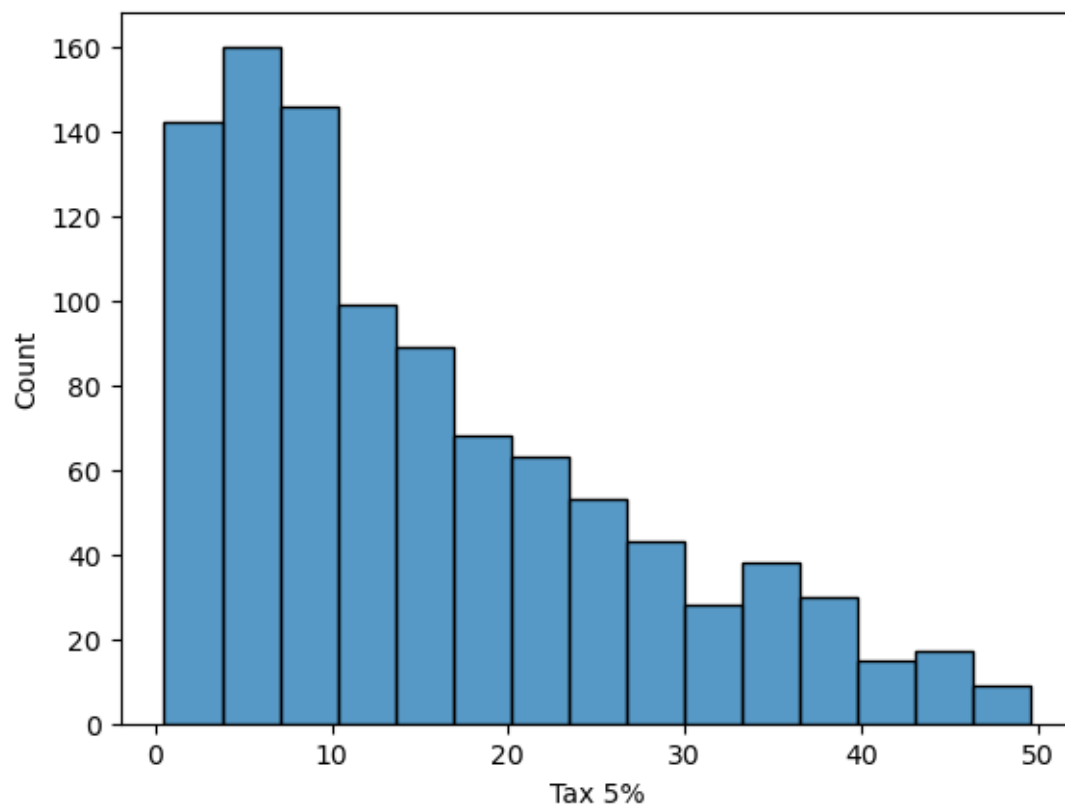
```
[6]: x=df['City'].value_counts()
plt.pie(x,labels=['Yangon','Naypyitaw','Mandalay'],startangle=90,explode=[0.1,0.1,0.1],radius=0.8)
plt.title('Distribution of different cities')
plt.legend(x)
```

```
[6]: <matplotlib.legend.Legend at 0x1ffe2891310>
```

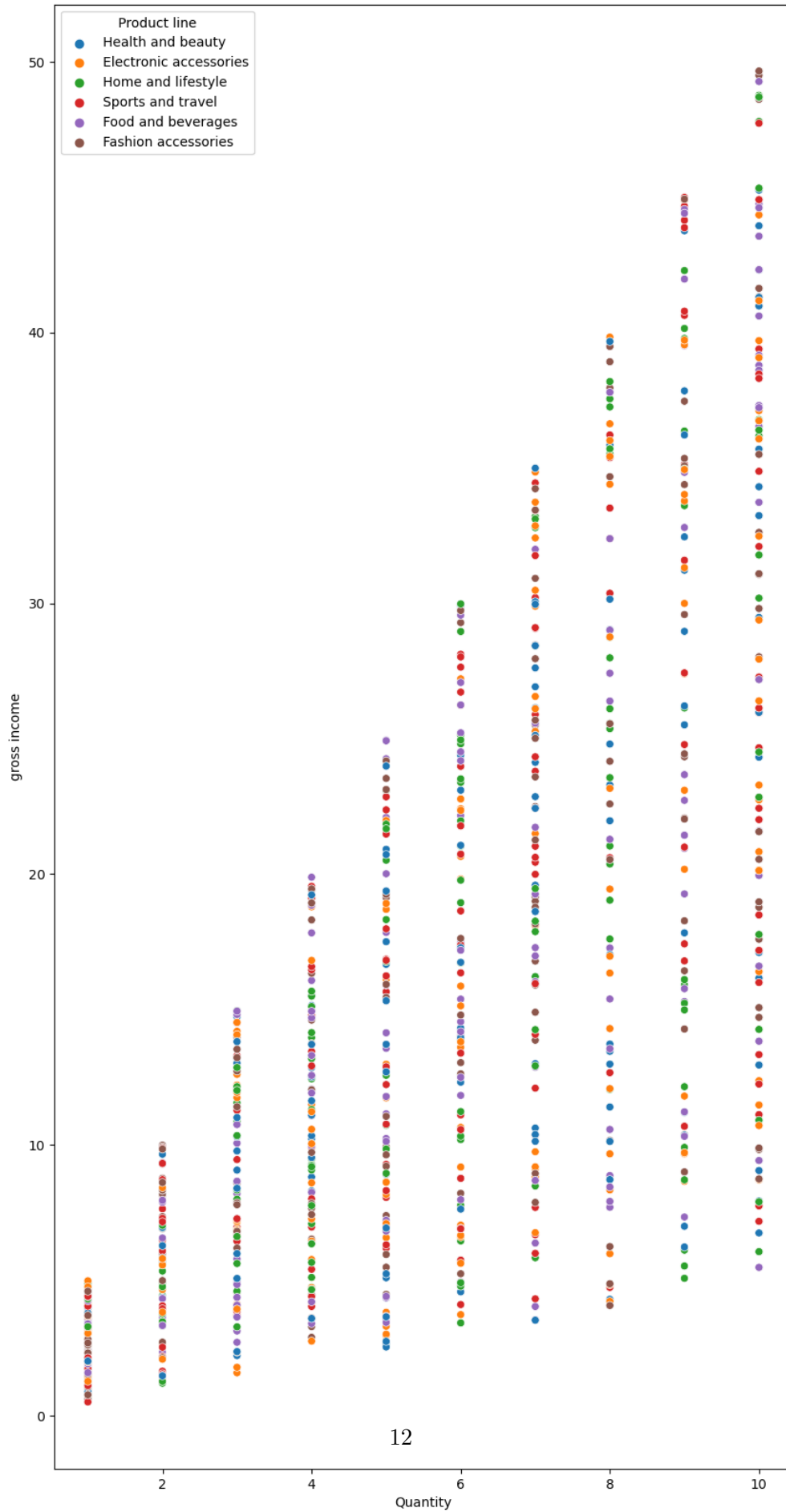


```
[115]: y=df['Tax 5%']
sb.histplot(y)
```

```
[115]: <AxesSubplot:xlabel='Tax 5%', ylabel='Count'>
```

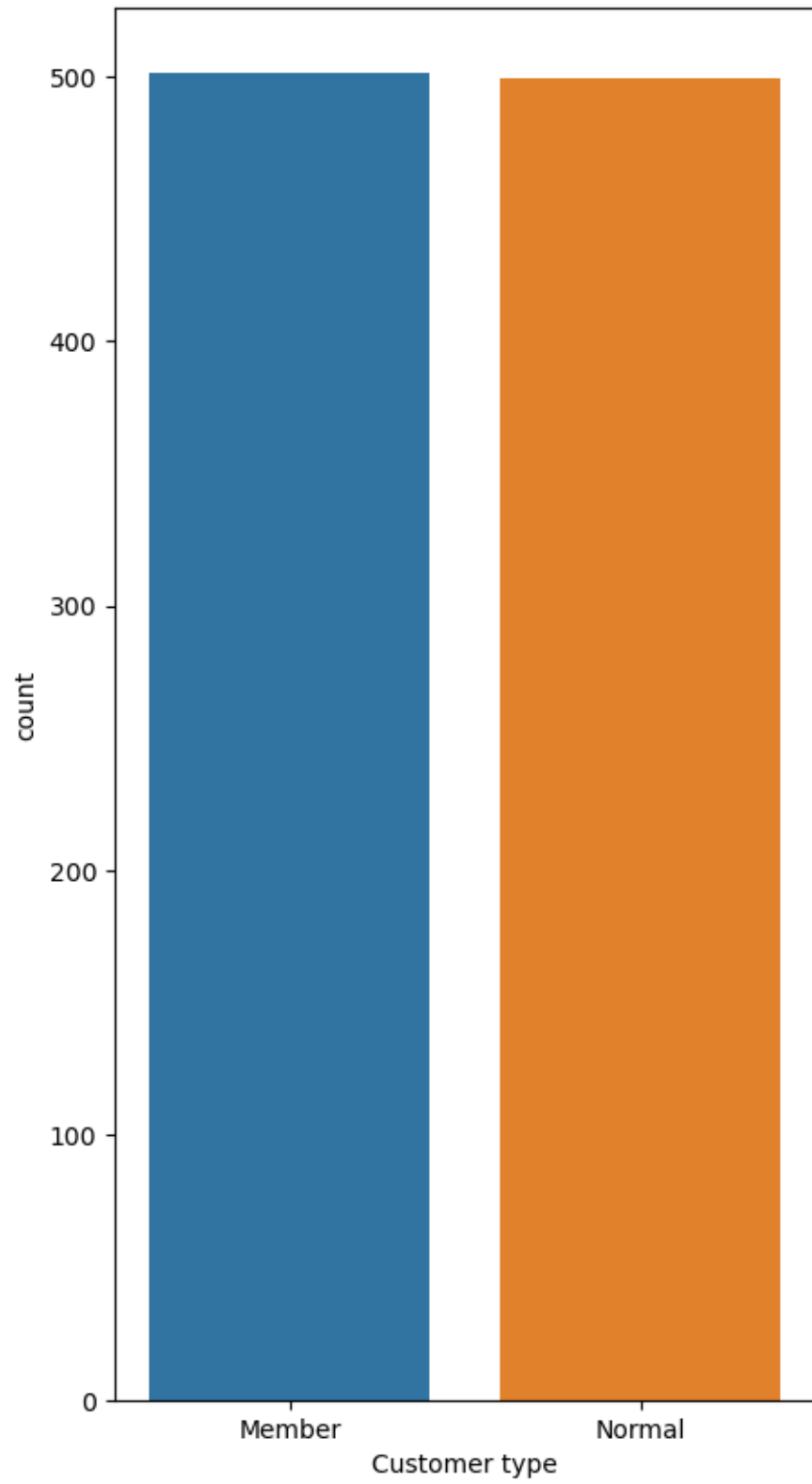


```
[127]: sb.scatterplot(data=df,x='Quantity',y='gross income',hue='Product line')
plt.rcParams["figure.figsize"]=(5,10)
plt.show()
```

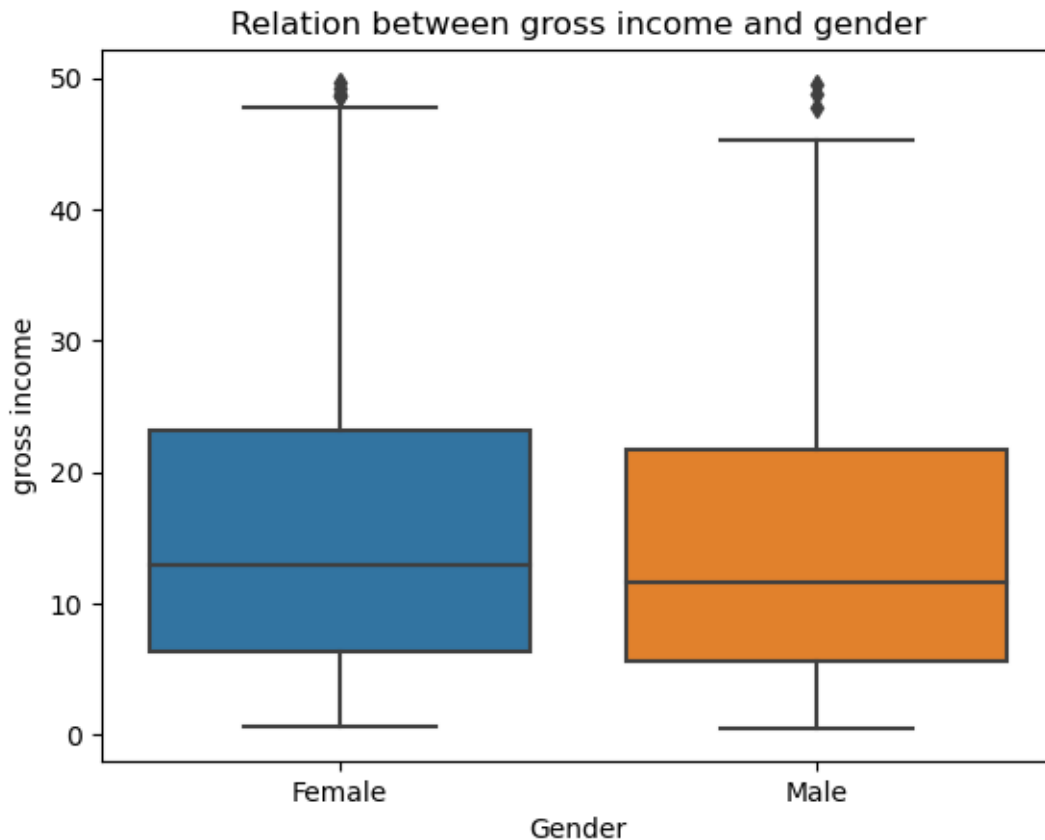


```
[138]: sb.countplot(data= df, x='Customer type')
```

```
[138]: <AxesSubplot:xlabel='Customer type', ylabel='count'>
```



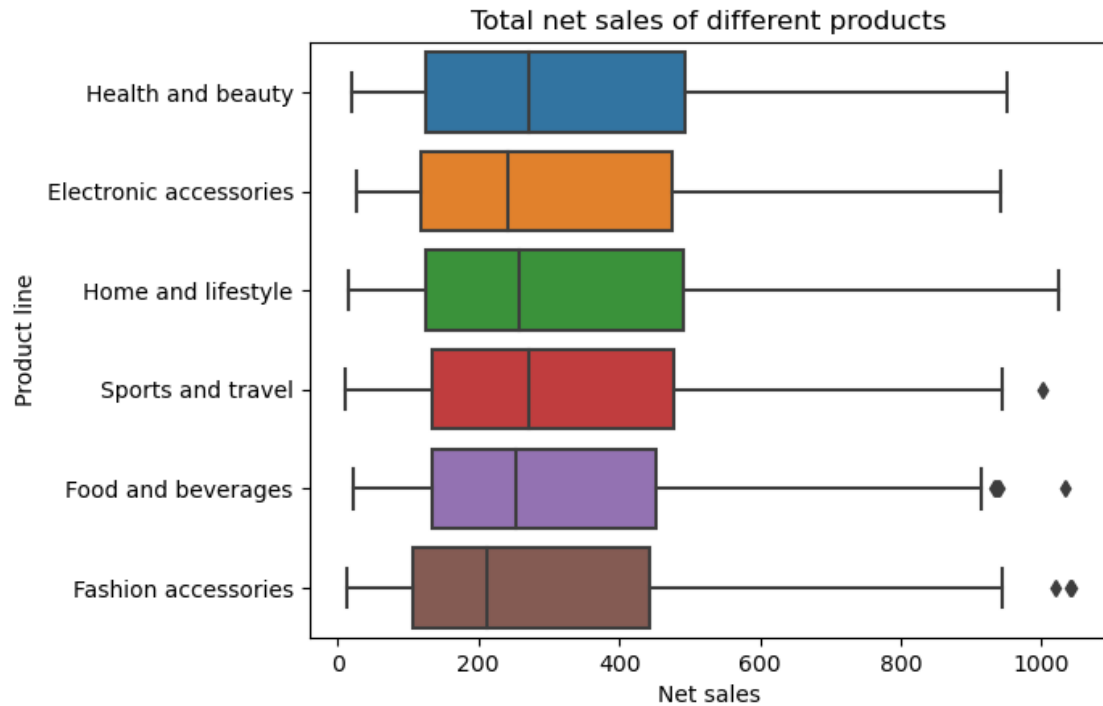
```
[43]: sb.boxplot(x=df['Gender'],y=df['gross income'])
plt.title('Relation between gross income and gender')
plt.show()
```



```
[17]: x=df['Net sales']
y=df['Product line']
sb.boxplot(x,y)
plt.title('Total net sales of different products')
```

C:\Users\Mokshogna Teja\anaconda3\lib\site-packages\seaborn_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
warnings.warn(

```
[17]: Text(0.5, 1.0, 'Total net sales of different products')
```



```
[32]: table1 = pd.pivot_table(df, values = ['Net sales'], index = ['Product line'],
    ↪columns = 'Gender', aggfunc = np.sum)
table1
```

```
[32]:
```

| Product line | Net sales | |
|------------------------|------------|------------|
| | Female | Male |
| Electronic accessories | 27102.0225 | 27235.5090 |
| Fashion accessories | 30437.4000 | 23868.4950 |
| Food and beverages | 33170.9175 | 22973.9265 |
| Health and beauty | 18560.9865 | 30632.7525 |
| Home and lifestyle | 30036.8775 | 23825.0355 |
| Sports and travel | 28574.7210 | 26548.1055 |

```
[31]: table2 = pd.pivot_table(df, values = ['Net sales'], index = ['Product line'],
    ↪columns = 'Gender', aggfunc = np.min)
table2
```

```
[31]:
```

| Product line | Net sales | |
|------------------------|-----------|---------|
| | Female | Male |
| Electronic accessories | 26.7225 | 37.6110 |
| Fashion accessories | 12.6945 | 13.4190 |
| Food and beverages | 23.7510 | 22.6590 |

| | | |
|--------------------|---------|---------|
| Health and beauty | 18.6375 | 26.2500 |
| Home and lifestyle | 25.2630 | 14.6790 |
| Sports and travel | 16.1070 | 10.6785 |

```
[30]: table3= pd.pivot_table(df, values = ['Net sales'], index = ['Product line'],
    ↪columns = 'Gender', aggfunc = np.max)
table3
```

```
[30]:
```

| | Net sales | |
|------------------------|-----------|----------|
| Gender | Female | Male |
| Product line | | |
| Electronic accessories | 942.4485 | 931.035 |
| Fashion accessories | 1042.6500 | 1039.290 |
| Food and beverages | 1034.4600 | 939.540 |
| Health and beauty | 794.6505 | 950.250 |
| Home and lifestyle | 1022.4900 | 1023.750 |
| Sports and travel | 1002.1200 | 944.622 |

```
[28]: table4 = pd.pivot_table(df, values = ['Net sales'], index = ['Product line'],
    ↪columns = 'Gender', aggfunc = np.mean)
table4
```

```
[28]:
```

| | Net sales | |
|------------------------|------------|------------|
| Gender | Female | Male |
| Product line | | |
| Electronic accessories | 322.643125 | 316.691965 |
| Fashion accessories | 317.056250 | 291.079207 |
| Food and beverages | 368.565750 | 273.499125 |
| Health and beauty | 290.015414 | 348.099460 |
| Home and lifestyle | 380.213639 | 294.136241 |
| Sports and travel | 324.712739 | 340.360327 |

```
[27]: table5 = pd.pivot_table(df, values = ['Net sales'], index = ['Product line'],
    ↪columns = 'City', aggfunc = np.sum)
table5
```

```
[27]:
```

| | Net sales | |
|------------------------|------------|------------|
| Gender | Female | Male |
| Product line | | |
| Electronic accessories | 322.643125 | 316.691965 |
| Fashion accessories | 317.056250 | 291.079207 |
| Food and beverages | 368.565750 | 273.499125 |
| Health and beauty | 290.015414 | 348.099460 |
| Home and lifestyle | 380.213639 | 294.136241 |
| Sports and travel | 324.712739 | 340.360327 |

```
[24]: table6 = pd.pivot_table(df, values = ['Net sales'], index = ['Product line'],
    ↪columns = 'City', aggfunc = np.min)
table6
```

```
[24]:
```

| | Net sales | | |
|------------------------|-----------|-----------|---------|
| City | Mandalay | Naypyitaw | Yangon |
| Product line | | | |
| Electronic accessories | 26.7225 | 31.7520 | 30.4080 |
| Fashion accessories | 35.1960 | 13.1670 | 12.6945 |
| Food and beverages | 56.4060 | 22.6590 | 33.4320 |
| Health and beauty | 18.6375 | 32.2770 | 19.2465 |
| Home and lifestyle | 33.9360 | 14.6790 | 19.1940 |
| Sports and travel | 34.6290 | 10.6785 | 16.1070 |

```
[25]: table7 = pd.pivot_table(df, values = ['Net sales'], index = ['Product line'],
    ↪columns = 'City', aggfunc = np.max)
table7
```

```
[25]:
```

| | Net sales | | |
|------------------------|-----------|-----------|-----------|
| City | Mandalay | Naypyitaw | Yangon |
| Product line | | | |
| Electronic accessories | 942.4485 | 864.57 | 931.0350 |
| Fashion accessories | 874.1250 | 1042.65 | 1039.2900 |
| Food and beverages | 888.6150 | 1034.46 | 932.3370 |
| Health and beauty | 922.6350 | 950.25 | 752.6400 |
| Home and lifestyle | 1022.4900 | 1023.75 | 951.8250 |
| Sports and travel | 944.6220 | 1002.12 | 926.9505 |

```
[26]: table8 = pd.pivot_table(df, values = ['Net sales'], index = ['Product line'],
    ↪columns = 'City', aggfunc = np.mean)
table8
```

```
[26]:
```

| | Net sales | | |
|------------------------|------------|------------|------------|
| City | Mandalay | Naypyitaw | Yangon |
| Product line | | | |
| Electronic accessories | 310.026245 | 344.890445 | 305.285225 |
| Fashion accessories | 264.730911 | 331.693385 | 320.245265 |
| Food and beverages | 304.297770 | 360.103864 | 295.915526 |
| Health and beauty | 376.993585 | 319.525500 | 268.037298 |
| Home and lifestyle | 350.983290 | 308.790067 | 344.879931 |
| Sports and travel | 322.390306 | 350.265067 | 328.350839 |

```
[35]: table9 = pd.pivot_table(df, values = ['Net sales'], index = ['Product line'],
    ↪columns = 'Customer type', aggfunc = np.sum)
table9
```

```
[35]:
```

| | Net sales | |
|------------------------|------------|------------|
| Customer type | Member | Normal |
| Product line | | |
| Electronic accessories | 24498.4950 | 29839.0365 |
| Fashion accessories | 26323.9620 | 27981.9330 |
| Food and beverages | 31357.6200 | 24787.2240 |
| Health and beauty | 25831.0395 | 23362.6995 |
| Home and lifestyle | 27978.0270 | 25883.8860 |
| Sports and travel | 28234.3005 | 26888.5260 |

```
[34]: table10 = pd.pivot_table(df, values = ['Net sales'], index = ['Product line'],
    ↪columns = 'Customer type', aggfunc = np.mean)
table10
```

```
[34]:
```

| | Net sales | |
|------------------------|------------|------------|
| Customer type | Member | Normal |
| Product line | | |
| Electronic accessories | 314.083269 | 324.337353 |
| Fashion accessories | 306.092581 | 304.151446 |
| Food and beverages | 333.591702 | 309.840300 |
| Health and beauty | 353.849856 | 295.730373 |
| Home and lifestyle | 337.084663 | 336.154364 |
| Sports and travel | 324.532190 | 340.361089 |