

```
In [2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [3]: data=pd.read_csv('market.csv')
```

```
In [4]: print(data)

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
.. ..
995 233-67-5758 C Naypyitaw Normal Male
996 305-96-2227 B Mandalay Normal Female
997 727-02-1313 A Yangon Member Male
998 347-56-2442 A Yangon Normal Male
999 849-09-3807 A Yangon Member Female

Product line Unit price Quantity Tax 5% Total \
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
.. ..
995 Health and beauty 40.35 1 2.0175 42.3675
996 Home and lifestyle 97.38 10 48.6900 1022.4900
997 Food and beverages 31.84 1 1.5920 33.4320
998 Home and lifestyle 65.82 1 3.2910 69.1110
999 Fashion accessories 88.34 7 30.9190 649.2990

Date Time Payment cogs gross margin percentage \
0 05-01-2019 13:08 Ewallet 522.83 4.761905
1 08-03-2019 10:29 Cash 76.40 4.761905
2 03-03-2019 13:23 Credit card 324.31 4.761905
3 27-01-2019 20:33 Ewallet 465.76 4.761905
4 08-02-2019 10:37 Ewallet 604.17 4.761905
.. ..
995 29-01-2019 13:46 Ewallet 40.35 4.761905
996 02-03-2019 17:16 Ewallet 973.80 4.761905
997 09-02-2019 13:22 Cash 31.84 4.761905
998 22-02-2019 15:33 Cash 65.82 4.761905
999 18-02-2019 13:28 Cash 618.38 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
.. ..
995 2.0175 6.2
996 48.6900 4.4
997 1.5920 7.7
998 3.2910 4.1
999 30.9190 6.6

[1000 rows x 17 columns]
```

```
In [5]: data.columns
```

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

```
In [6]: data.isna().sum()
```

```
Out[6]: 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
```

```
In [7]: def split_date(data):
data['Date']=pd.to_datetime(data['Date'])
data['Year']=data.Date.dt.year
data['Month']=data.Date.dt.month
data['Day']=data.Date.dt.day
```

```
In [8]: split_date(data)
#check how now our data looks like
data.head()
```

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	cogs	gross margin percentage	gross income	Rating	Year	Month	Day
0	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715	2019-05-01	13:08	Ewallet	522.83	4.761905	26.1415	9.1	2019	5	1
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80.2200	2019-08-03	10:29	Cash	76.40	4.761905	3.8200	9.6	2019	8	3
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	2019-03-03	13:23	Credit card	324.31	4.761905	16.2155	7.4	2019	3	3
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	489.0480	2019-01-27	20:33	Ewallet	465.76	4.761905	23.2880	8.4	2019	1	27
4	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	2019-08-02	10:37	Ewallet	604.17	4.761905	30.2085	5.3	2019	8	2

```
In [9]: data.Year.value_counts()
```

```
Out[9]: 2019    1000
Name: Year, dtype: int64
```

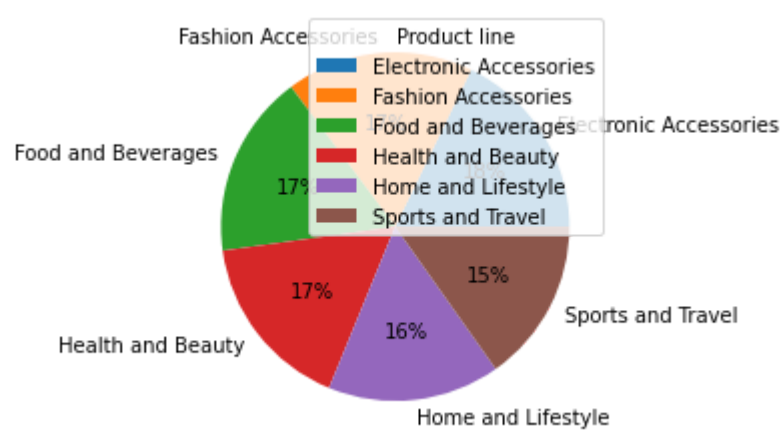
```
In [10]: data.rename(columns = {'Product line':'product'}, inplace = True)
data.Month.value_counts()
```

```
Out[10]: 1    262
3    228
2    201
5     41
8     41
7     38
9     37
6     33
10    32
12     31
4     29
11     27
Name: Month, dtype: int64
```

```
In [11]: print(data['product'].count())
#Get total numbe of customers per their Gender Male of female
data['product'].value_counts()
```

```
Out[11]: 1000
Fashion accessories      178
Food and beverages      174
Electronic accessories   170
Sports and travel        166
Home and lifestyle       169
Health and beauty        152
Name: product, dtype: int64
```

```
In [12]: pl=data['product'].value_counts()
pl.labels=['Electronic Accessories','Fashion Accessories','Food and Beverages','Health and Beauty','Home and Lifestyle','Sports and Travel']
plt.pie(pl,labels=pl_labels,autopct='%0.1f%%')
plt.legend(title='Product line')
plt.show()
```



```
In [13]: data[['Gender','Payment']].value_counts(ascending=True).to_dict()
```

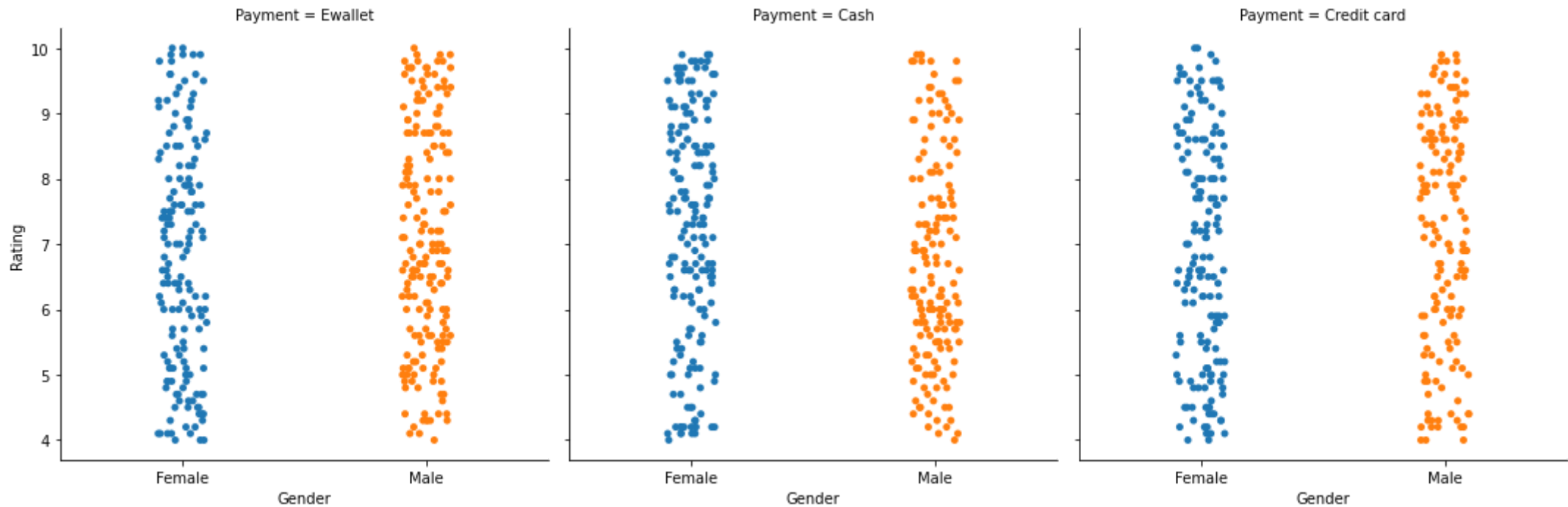
```
Out[13]: {'Male', 'Credit card': 148,
('Female', 'Ewallet'): 160,
('Female', 'Credit card'): 163,
('Male', 'Cash'): 166,
('Female', 'Cash'): 178,
('Male', 'Ewallet'): 185}
```

```
In [14]: paymentcounts=data['Payment'].value_counts().to_dict()
df=pd.DataFrame(list(paymentcounts.items()),columns=['Payment','Counts'])
df
```

	Payment	Counts
0	Ewallet	345
1	Cash	344
2	Credit card	311

```
In [16]: sns.catplot(x='Gender',y='Rating',col='Payment',data=data)
```

```
Out[16]: <seaborn.axisgrid.FacetGrid at 0x24c55779ee0>
```



```
In [ ]: p=data.head(max(data['gross margin percentage'],10))
print(p)
```

```
In [ ]:
```