E-Commerce Analysis With Python and Power BI

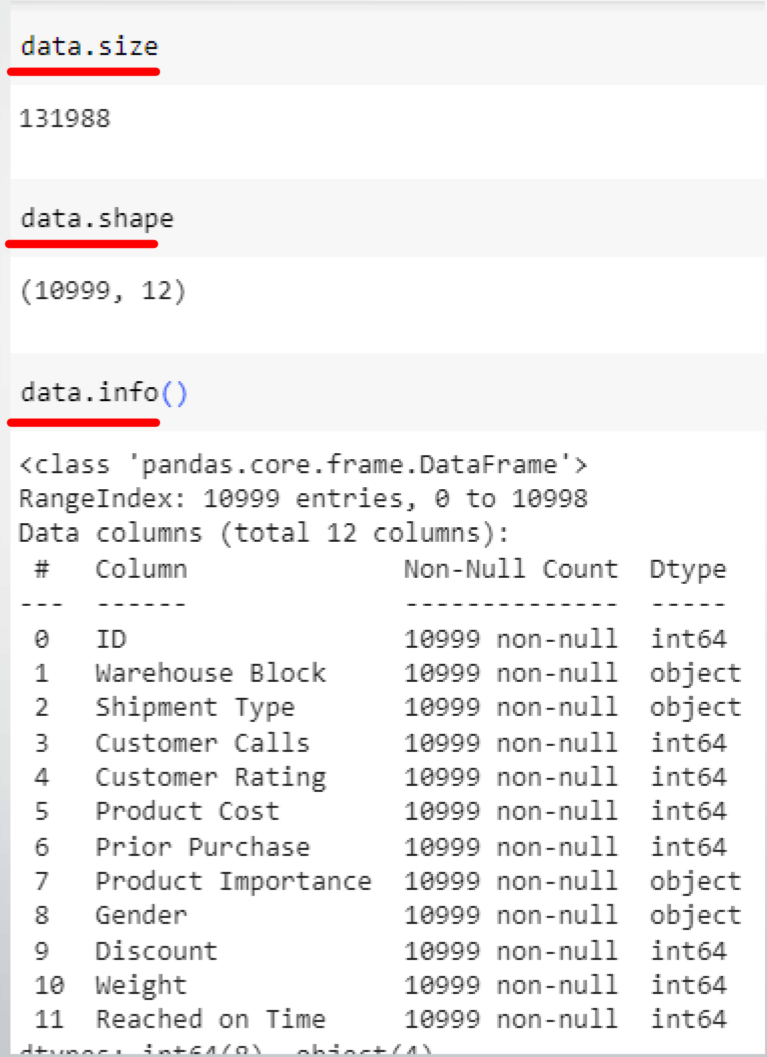
Nigar Aghayeva

General information about Data

Dataset has ID, Warehouse Block, Shipment Type, Customer Calls, Customer Rating, Product Cost, Prior Purchase, Product Importance, Gender, Discount, Weight, Reached on Time columns.

Analyzing data with Python

- Firstly, get more information about data like number of rows, column, size, shape
- Then check null values and duplicates



```
data.size
```

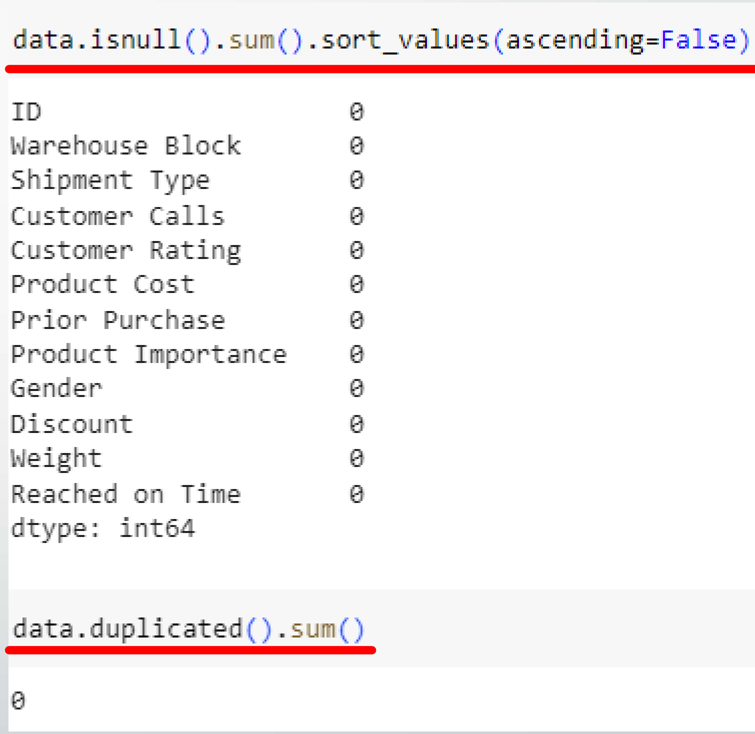
```
131988
```

```
data.shape
```

```
(10999, 12)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10999 entries, 0 to 10998
Data columns (total 12 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   ID                   10999 non-null  int64 
 1   Warehouse Block      10999 non-null  object
 2   Shipment Type        10999 non-null  object
 3   Customer Calls       10999 non-null  int64 
 4   Customer Rating      10999 non-null  int64 
 5   Product Cost         10999 non-null  int64 
 6   Prior Purchase       10999 non-null  int64 
 7   Product Importance   10999 non-null  object
 8   Gender               10999 non-null  object
 9   Discount             10999 non-null  int64 
 10  Weight               10999 non-null  int64 
 11  Reached on Time      10999 non-null  int64 
dtypes: int64(8), object(4)
```



```
data.isnull().sum().sort_values(ascending=False)
```

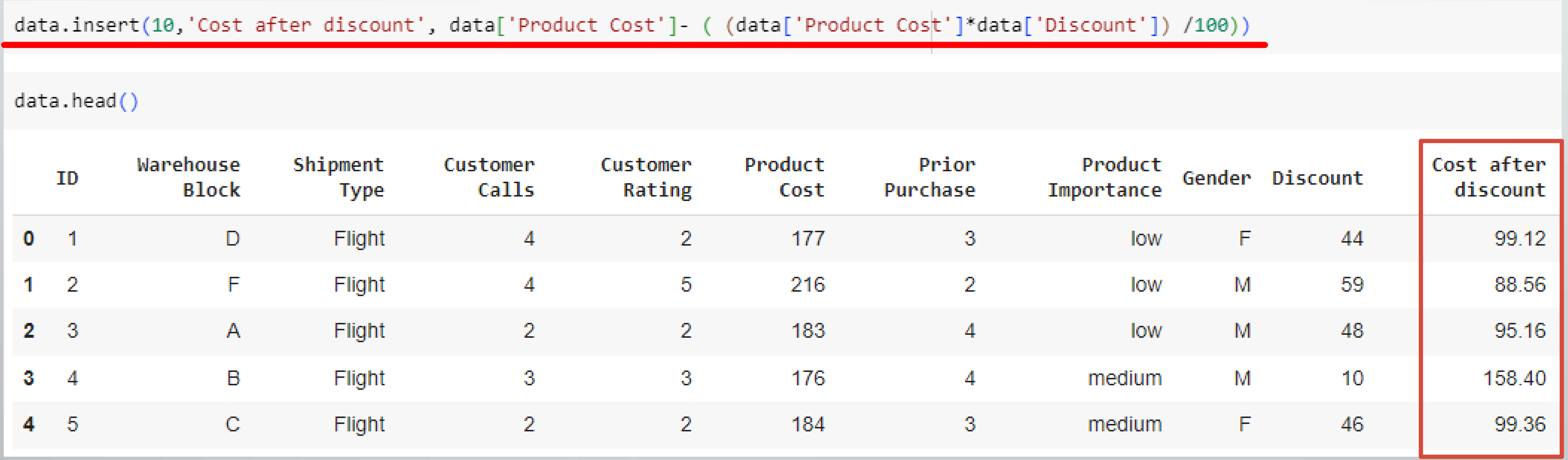
```
ID                    0
Warehouse Block       0
Shipment Type         0
Customer Calls        0
Customer Rating       0
Product Cost          0
Prior Purchase        0
Product Importance    0
Gender                0
Discount              0
Weight                0
Reached on Time       0
dtype: int64
```

```
data.duplicated().sum()
```

```
0
```

Next step is to add columns if it is need, or calculating some calculation in order to get more information about data

- So, we can add column which is represent cost of product after discount

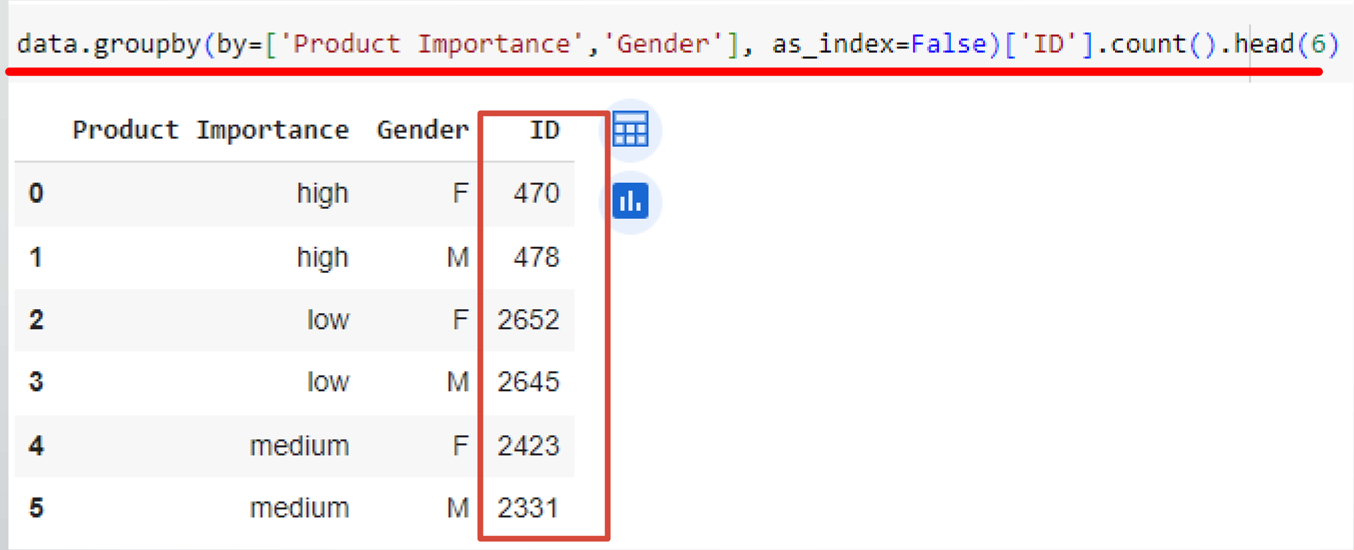


```
data.insert(10,'Cost after discount', data['Product Cost']- ( (data['Product Cost']*data['Discount']) /100))
```

```
data.head()
```

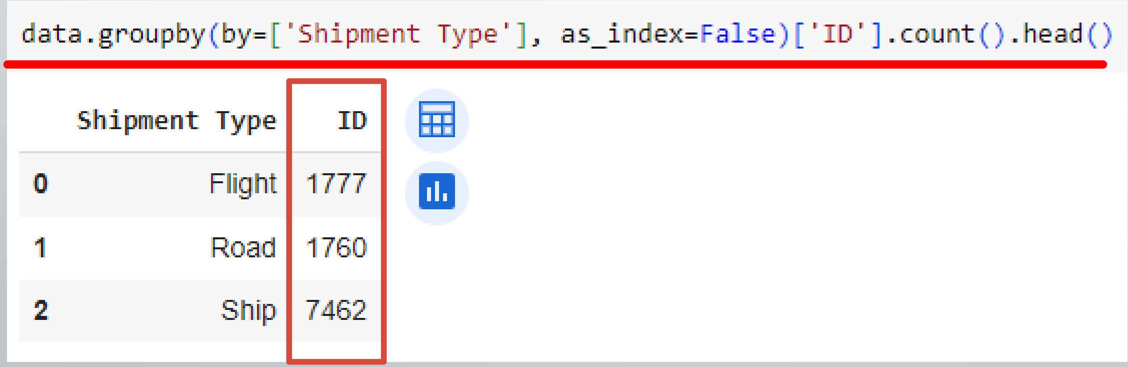
	ID	Warehouse Block	Shipment Type	Customer Calls	Customer Rating	Product Cost	Prior Purchase	Product Importance	Gender	Discount	Cost after discount
0	1	D	Flight	4	2	177	3	low	F	44	99.12
1	2	F	Flight	4	5	216	2	low	M	59	88.56
2	3	A	Flight	2	2	183	4	low	M	48	95.16
3	4	B	Flight	3	3	176	4	medium	M	10	158.40
4	5	C	Flight	2	2	184	3	medium	F	46	99.36

- After that, we use ‘group by’ function so as to group data according to product importance and gender, and shipment type.



```
data.groupby(by=['Product Importance','Gender'], as_index=False)['ID'].count().head(6)
```

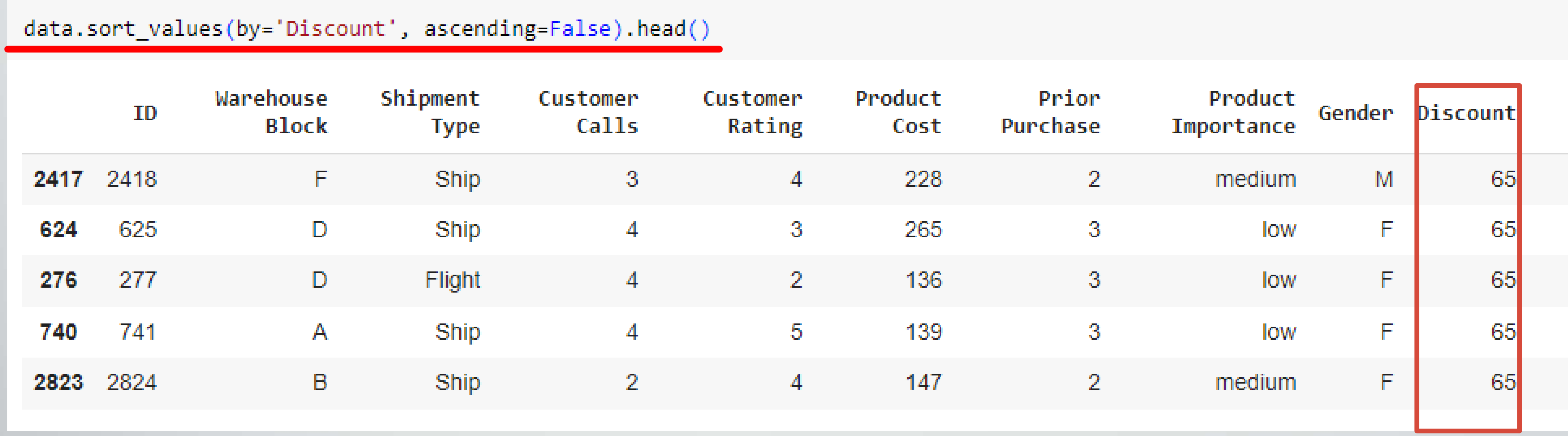
	Product Importance	Gender	ID
0	high	F	470
1	high	M	478
2	low	F	2652
3	low	M	2645
4	medium	F	2423
5	medium	M	2331



```
data.groupby(by=['Shipment Type'], as_index=False)['ID'].count().head()
```

	Shipment Type	ID
0	Flight	1777
1	Road	1760
2	Ship	7462

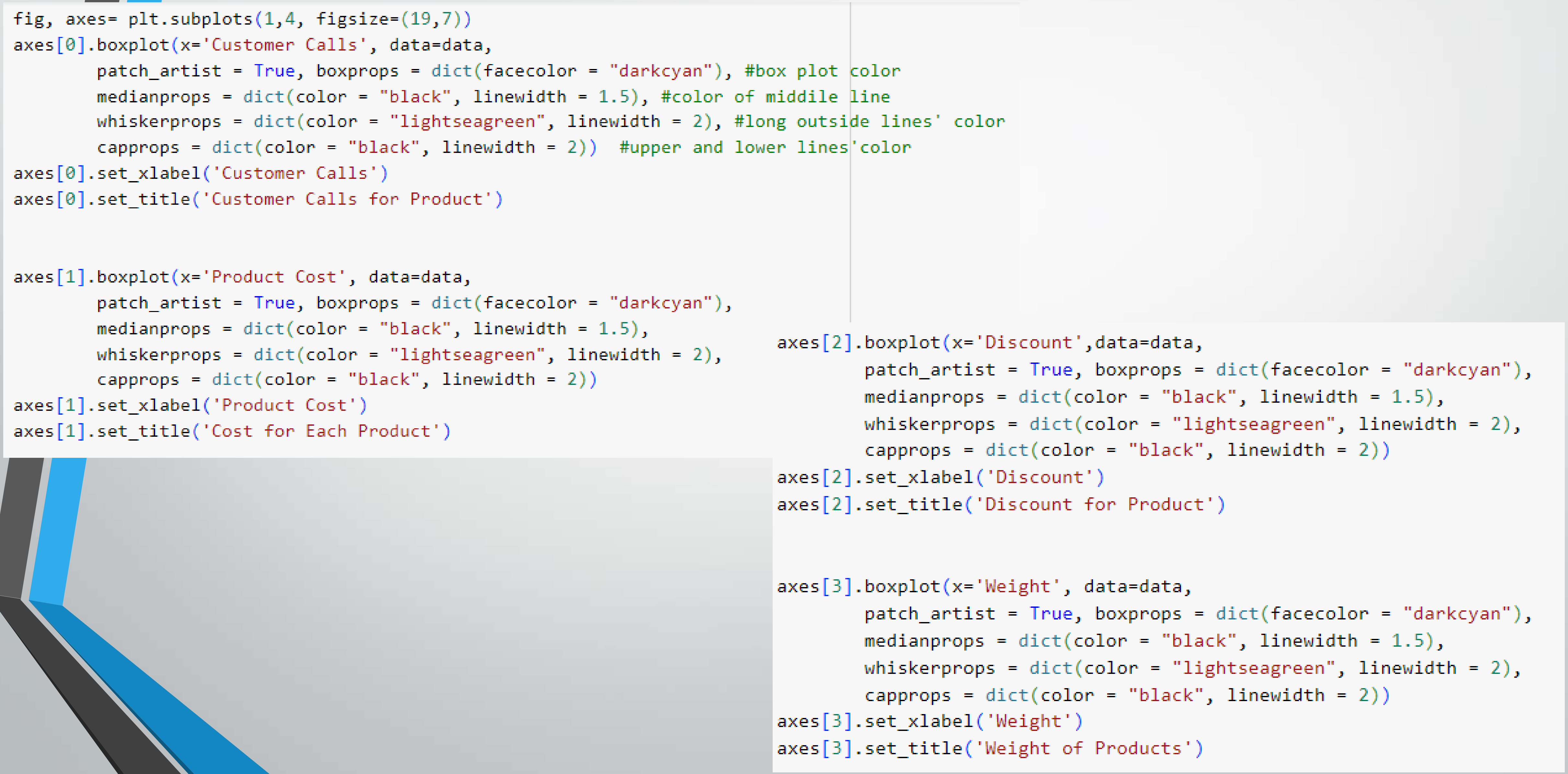
- Sort values to find the biggest discount



```
data.sort_values(by='Discount', ascending=False).head()
```

	ID	Warehouse Block	Shipment Type	Customer Calls	Customer Rating	Product Cost	Prior Purchase	Product Importance	Gender	Discount
2417	2418	F	Ship	3	4	228	2	medium	M	65
624	625	D	Ship	4	3	265	3	low	F	65
276	277	D	Flight	4	2	136	3	low	F	65
740	741	A	Ship	4	5	139	3	low	F	65
2823	2824	B	Ship	2	4	147	2	medium	F	65

- Create boxplots to find outliers



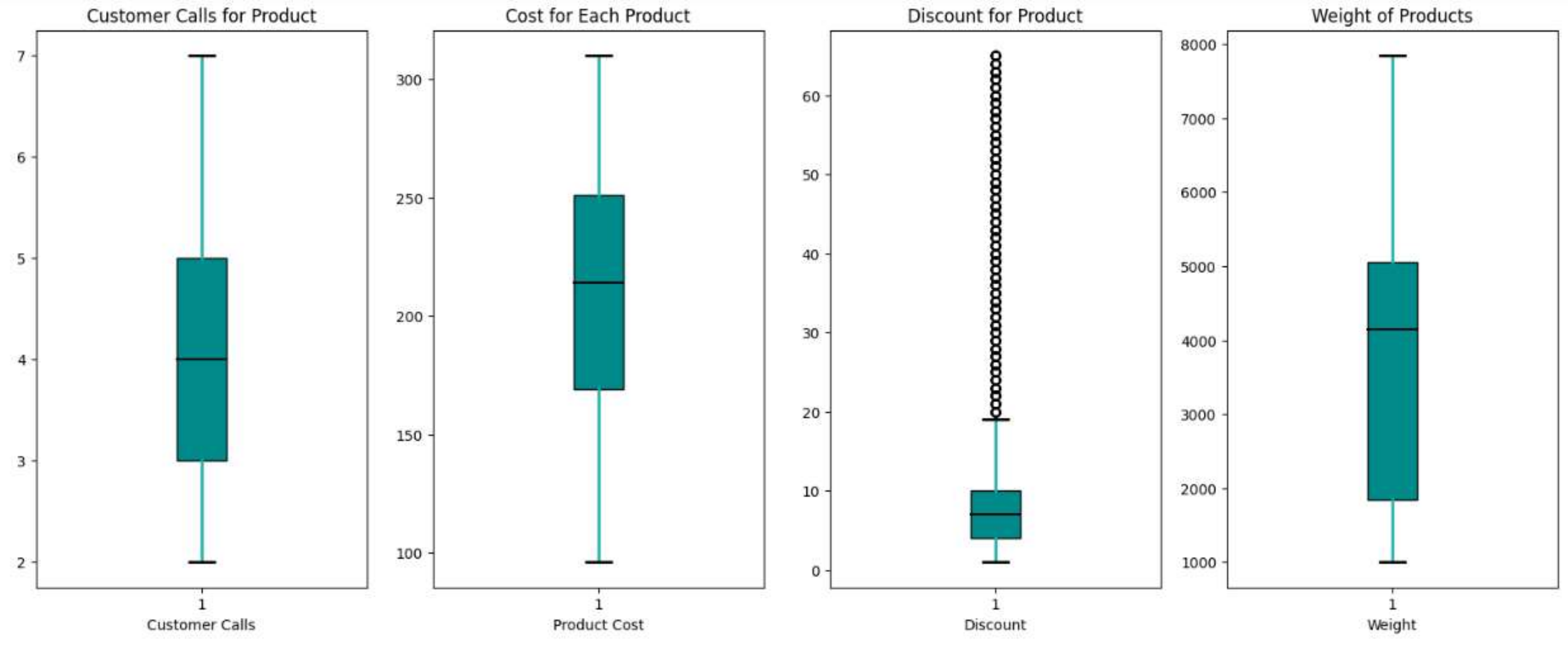
```
fig, axes= plt.subplots(1,4, figsize=(19,7))
axes[0].boxplot(x='Customer Calls', data=data,
        patch_artist = True, boxprops = dict(facecolor = "darkcyan"), #box plot color
        medianprops = dict(color = "black", linewidth = 1.5), #color of middile line
        whiskerprops = dict(color = "lightseagreen", linewidth = 2), #long outside lines' color
        capprops = dict(color = "black", linewidth = 2))  #upper and lower lines'color
axes[0].set_xlabel('Customer Calls')
axes[0].set_title('Customer Calls for Product')


axes[1].boxplot(x='Product Cost', data=data,
        patch_artist = True, boxprops = dict(facecolor = "darkcyan"),
        medianprops = dict(color = "black", linewidth = 1.5),
        whiskerprops = dict(color = "lightseagreen", linewidth = 2),
        capprops = dict(color = "black", linewidth = 2))
axes[1].set_xlabel('Product Cost')
axes[1].set_title('Cost for Each Product')
```

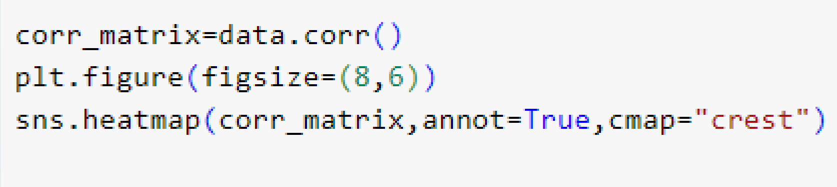
```
axes[2].boxplot(x='Discount',data=data,
        patch_artist = True, boxprops = dict(facecolor = "darkcyan"),
        medianprops = dict(color = "black", linewidth = 1.5),
        whiskerprops = dict(color = "lightseagreen", linewidth = 2),
        capprops = dict(color = "black", linewidth = 2))
axes[2].set_xlabel('Discount')
axes[2].set_title('Discount for Product')


axes[3].boxplot(x='Weight', data=data,
        patch_artist = True, boxprops = dict(facecolor = "darkcyan"),
        medianprops = dict(color = "black", linewidth = 1.5),
        whiskerprops = dict(color = "lightseagreen", linewidth = 2),
        capprops = dict(color = "black", linewidth = 2))
axes[3].set_xlabel('Weight')
axes[3].set_title('Weight of Products')
```

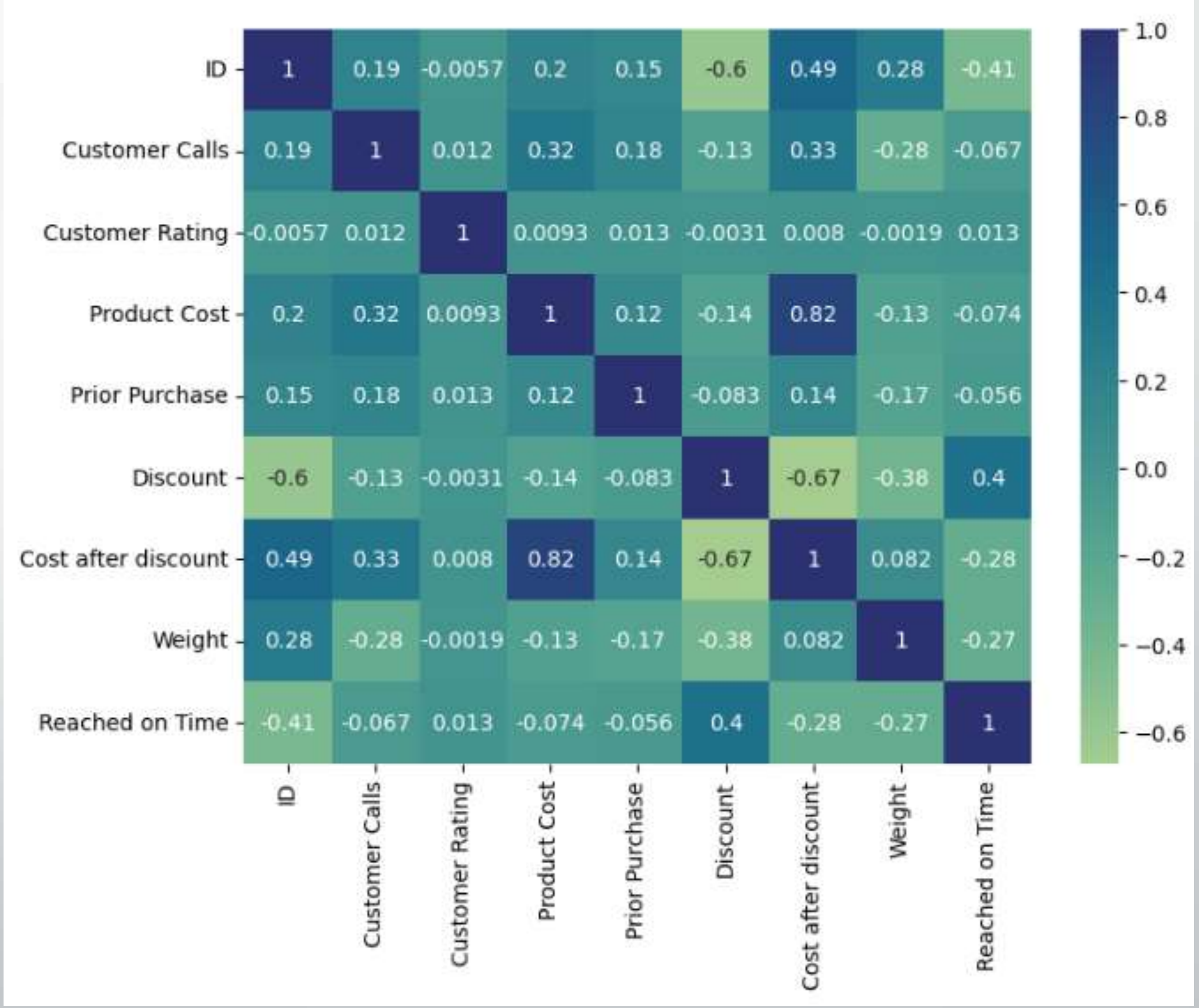
- Result: We observe some outliers in discount chart but it was normal. Because during black Friday and some remarkable days were days which big sales happen.



- Heatmap to find relationship between features



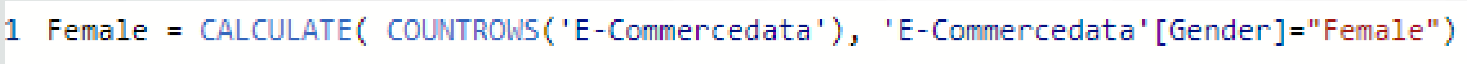
```
corr_matrix=data.corr()
plt.figure(figsize=(8,6))
sns.heatmap(corr_matrix,annot=True,cmap="crest")
```



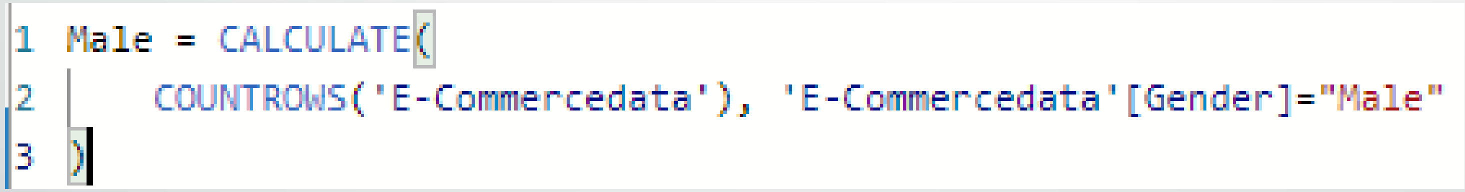
	ID	Customer Calls	Customer Rating	Product Cost	Prior Purchase	Discount	Cost after discount	Weight	Reached on Time
ID	1	0.19	-0.0057	0.2	0.15	-0.6	0.49	0.28	-0.41
Customer Calls	0.19	1	0.012	0.32	0.18	-0.13	0.33	-0.28	-0.067
Customer Rating	-0.0057	0.012	1	0.0093	0.013	-0.0031	0.008	-0.0019	0.013
Product Cost	0.2	0.32	0.0093	1	0.12	-0.14	0.82	-0.13	-0.074
Prior Purchase	0.15	0.18	0.013	0.12	1	-0.083	0.14	-0.17	-0.056
Discount	-0.6	-0.13	-0.0031	-0.14	-0.083	1	-0.67	-0.38	0.4
Cost after discount	0.49	0.33	0.008	0.82	0.14	-0.67	1	0.082	-0.28
Weight	0.28	-0.28	-0.0019	-0.13	-0.17	-0.38	0.082	1	-0.27
Reached on Time	-0.41	-0.067	0.013	-0.074	-0.056	0.4	-0.28	-0.27	1

Power BI to visualize data and create report

- In the first stage, we create measures to find number of women and men. In order to do it we use DAX function.



```
Female = CALCULATE( COUNTROWS('E-Commercedata'), 'E-Commercedata'[Gender]="Female")
```

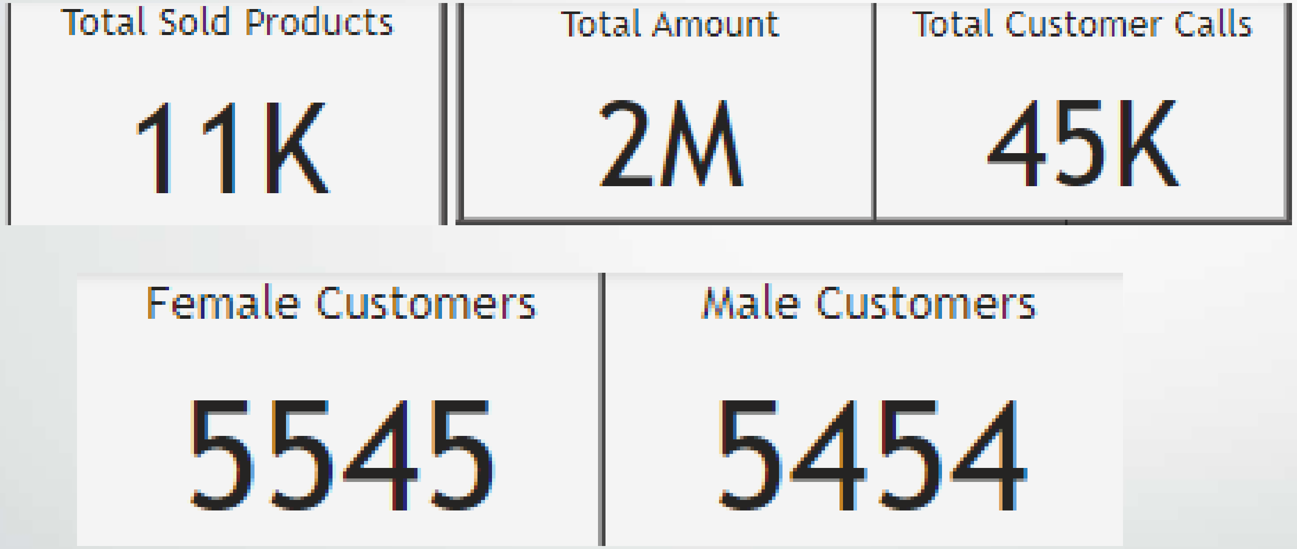


```
Male = CALCULATE(
    COUNTROWS('E-Commercedata'), 'E-Commercedata'[Gender]="Male"
)
```

Now, We can start visualization!

1. Create cards to show total amount, female/male customers , sold products, calls.
2. Slicer to filter data according to Warehouse Block.
3. Pie chart to describe shipment types.
4. Donut chart to represent times of customer calls.
5. Bar chart for Customer rating by Reached on Time.
6. Gauge to show average satisfaction .
7. Tree map to visualize Average of Discount by Product Importance and Gender .
8. Next and last step to make report view clearer make some adjustment add theme.

- Create cards to show total amount, female/male customers , sold products, calls .



Total Sold Products

11K

Total Amount

2M

Total Customer Calls

45K

Female Customers

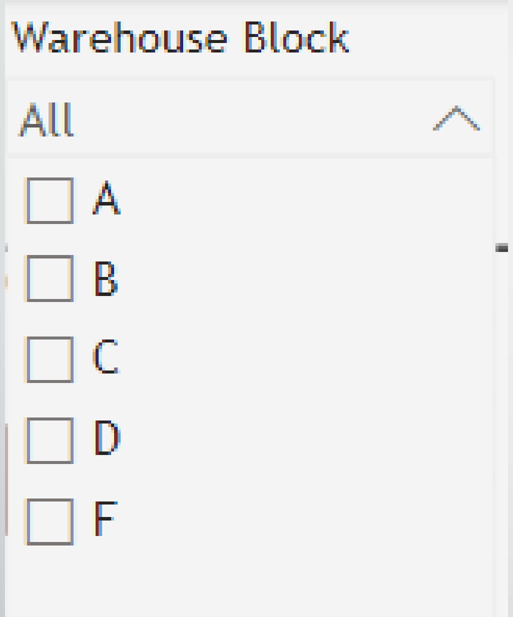
5545

Male Customers

5454

- With the help of these cards, we can easily identify general information about data.

- Slicer to filter data according to Warehouse Block.



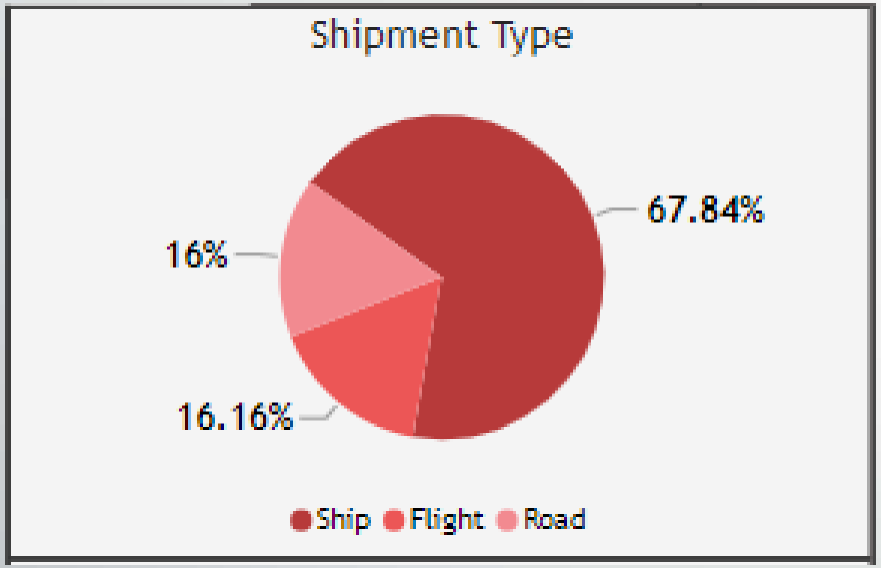
Warehouse Block

All

□ A
□ B
□ C
□ D
□ F

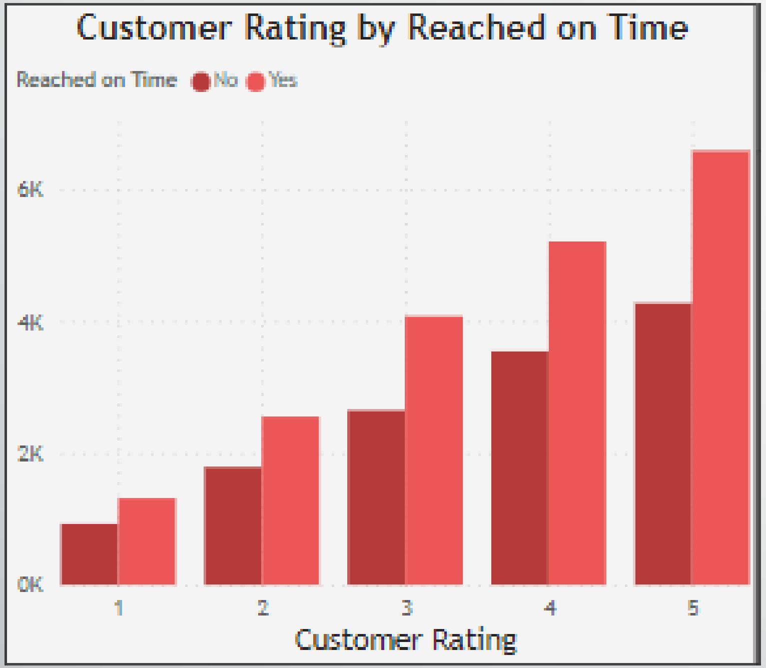
- It gives us chance to filter data as which warehouse block has most/least sold products etc.

- Pie chart to describe shipment types.



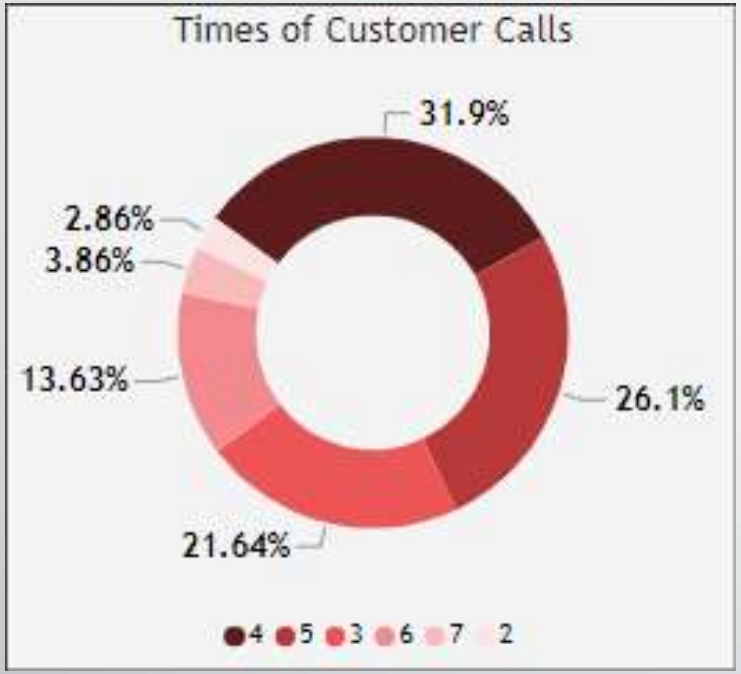
- Now, we can clearly see shipment types and which one is used more than others. So, ship used the most, after that, flight and road coming. It assist us to make some changing about shipment types. Like we can do some sales in other shipment types if we want to boost flight and road choices.

- Bar chart for Customer rating by Reached on Time.



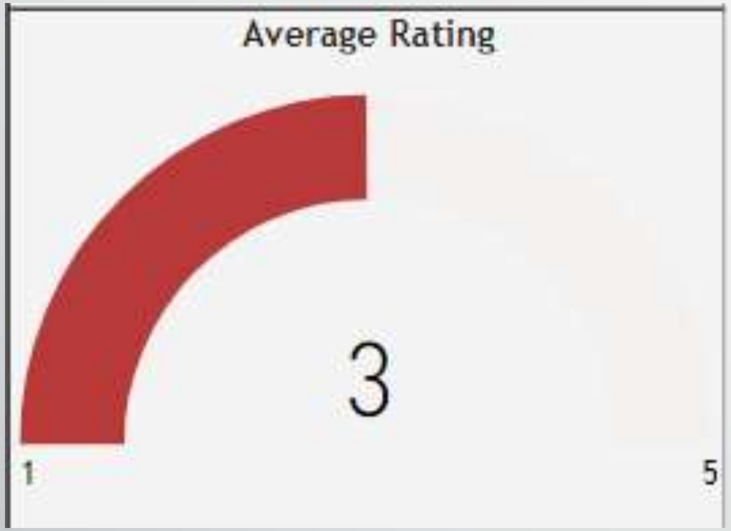
- It can be quickly seen that 5 rating took the biggest part of bar chart. Additionally, reaching on time did not have huge effect on this data.

- Donut chart to represent times of customer calls.



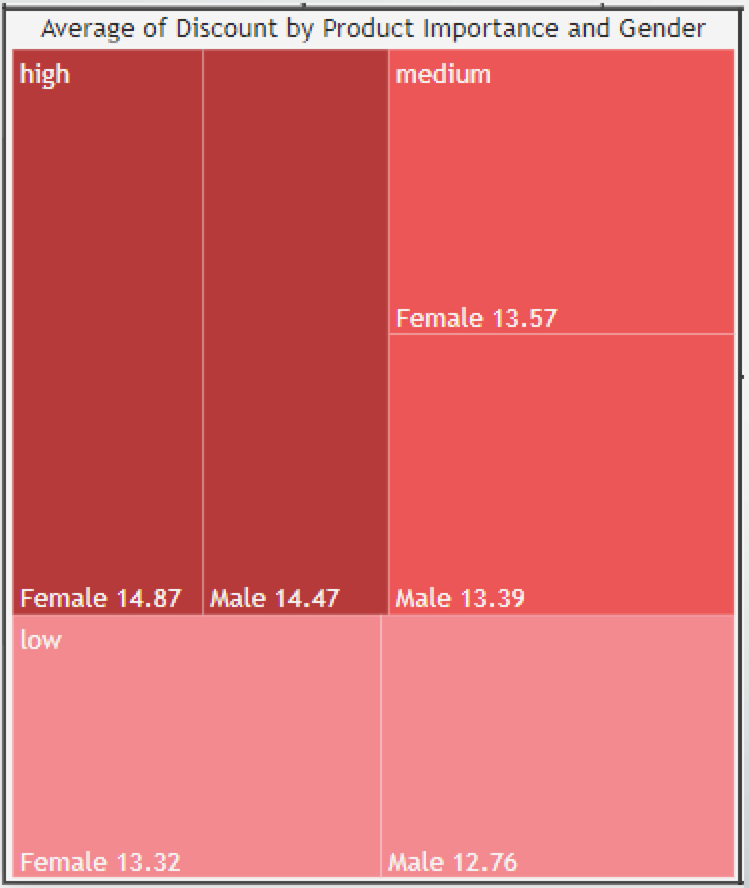
- Some of users called 4 times to get info about their products, minimum times was 2. We can make steps effortless for our users so as to get less phone calls.

- Gauge to show average satisfaction



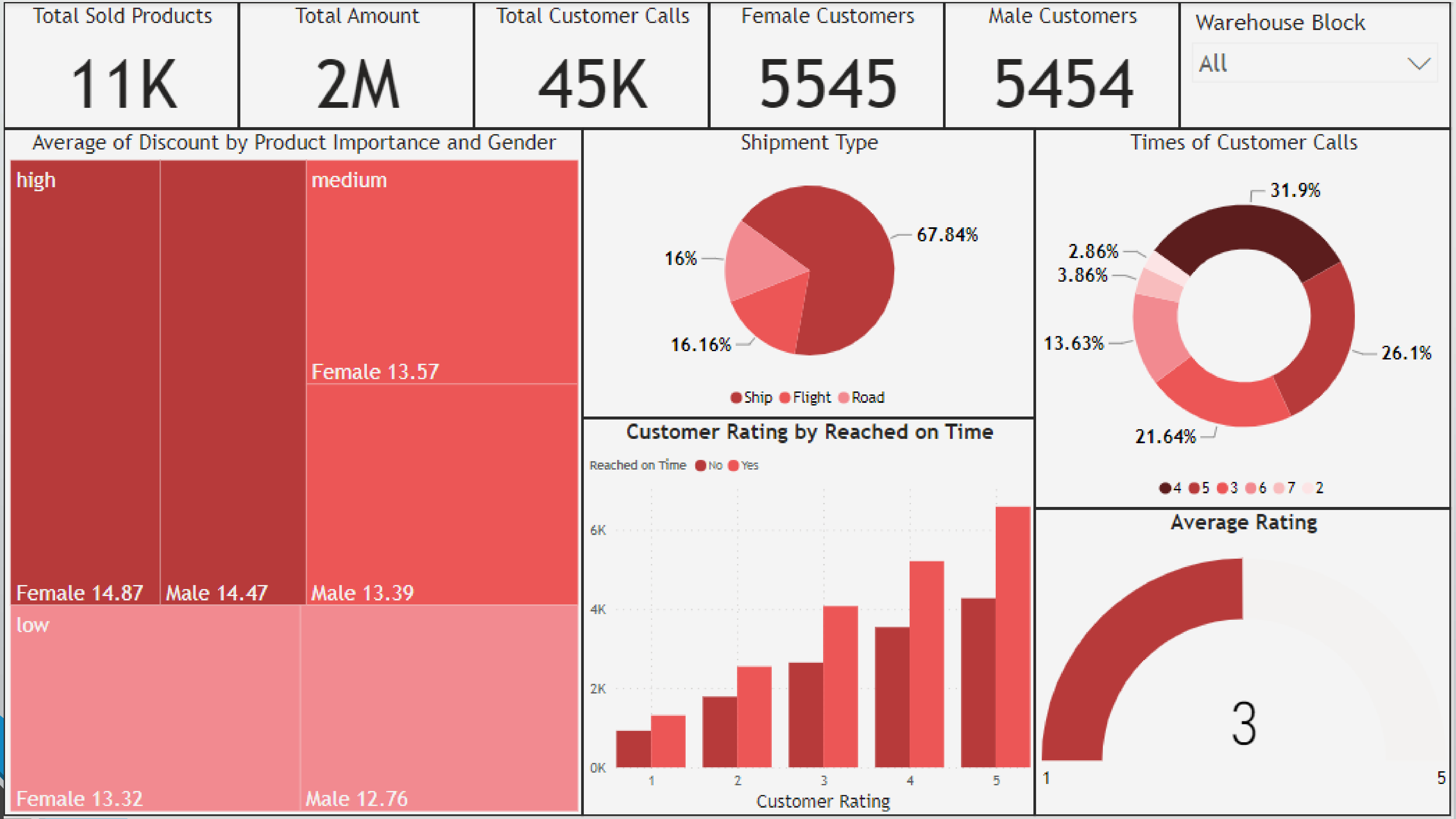
- Average satisfaction was 3, so according to this report, we decide to do something about increasing rate or not.

- Tree map to visualize Average of Discount by Product Importance and Gender .



- It shows that while product importance increased also average product discount rose and gender does not have big effect on it.

Finally, our report is ready!



Total Sold Products
11K

Total Amount
2M

Total Customer Calls
45K

Female Customers
5545

Male Customers
5454

Warehouse Block
All