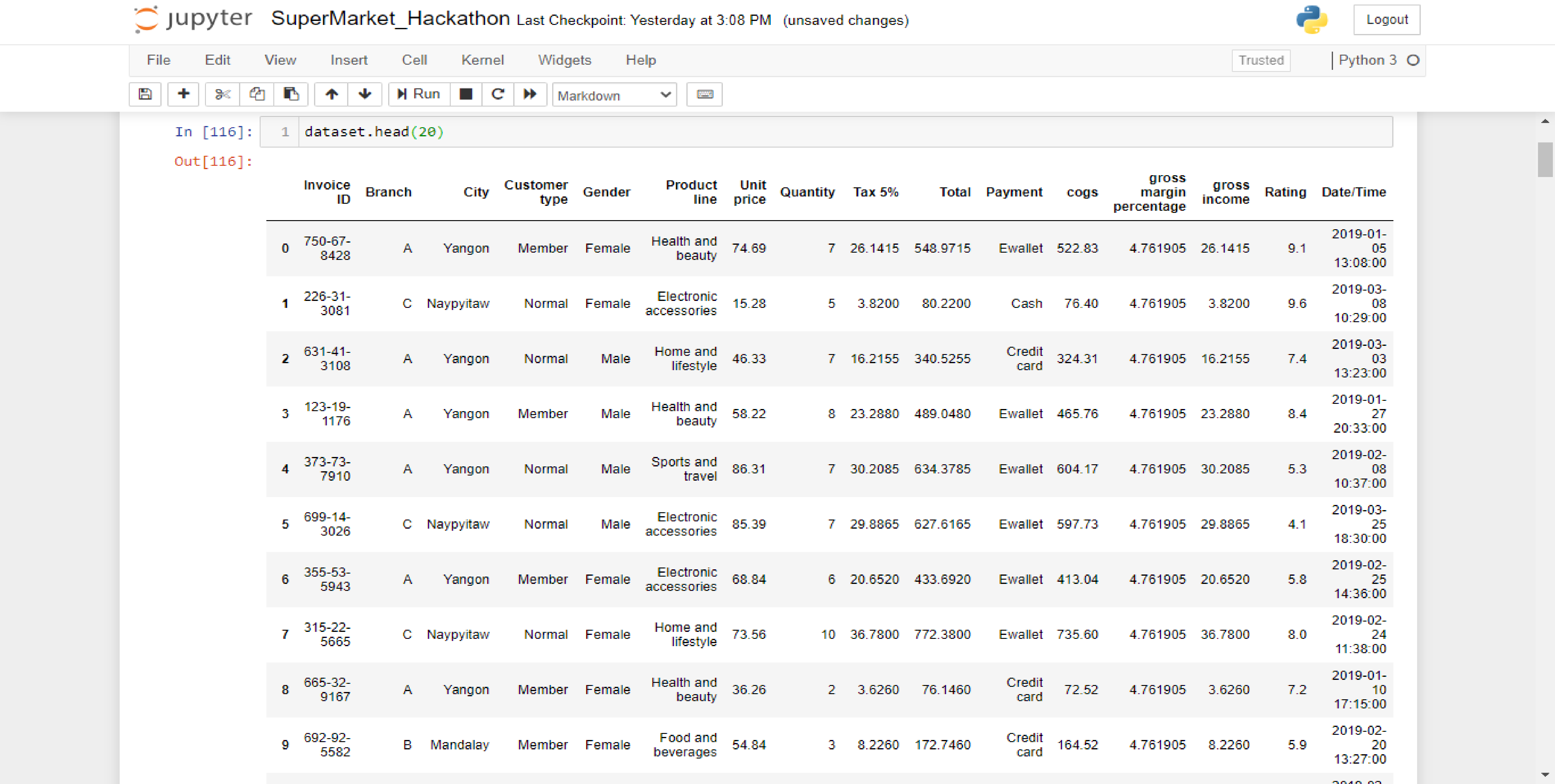
# SUPER-MARKET DATA ANALYSIS BY PYTHON

- GANESH DEVARE

# DATASET

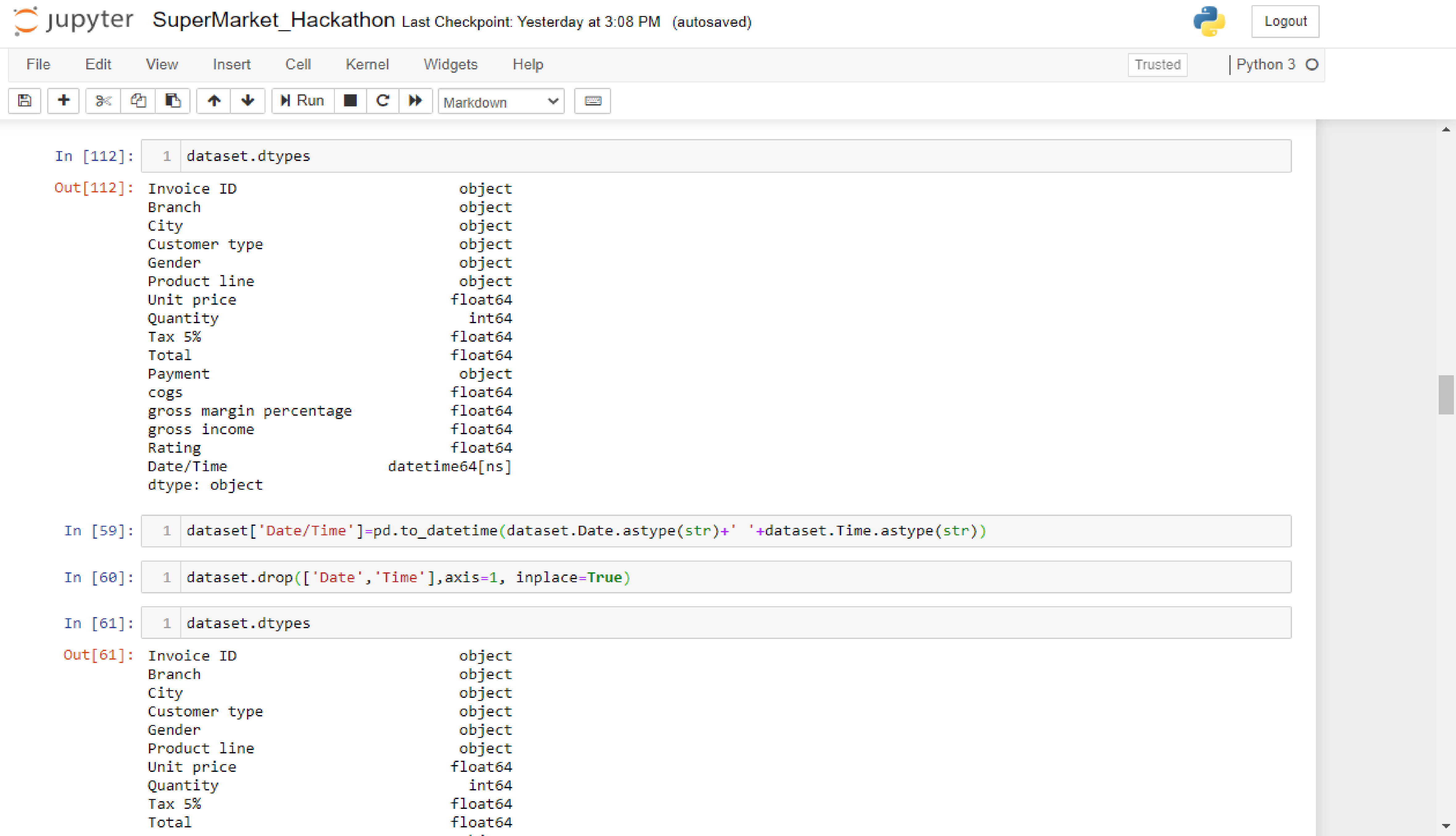


```
In [116]: dataset.head(20)
```

Out[116]:

| | Invoice ID | Branch | City | Customer type | Gender | Product line | Unit price | Quantity | Tax 5% | Total | Payment | cogs | gross margin percentage | gross income | Rating | Date/Time |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 750-67-8428 | A | Yangon | Member | Female | Health and beauty | 74.69 | 7 | 26.1415 | 548.9715 | Ewallet | 522.83 | 4.761905 | 26.1415 | 9.1 | 2019-01-05 13:08:00 |
| 1 | 226-31-3081 | C | Naypyitaw | Normal | Female | Electronic accessories | 15.28 | 5 | 3.8200 | 80.2200 | Cash | 76.40 | 4.761905 | 3.8200 | 9.6 | 2019-03-08 10:29:00 |
| 2 | 631-41-3108 | A | Yangon | Normal | Male | Home and lifestyle | 46.33 | 7 | 16.2155 | 340.5255 | Credit card | 324.31 | 4.761905 | 16.2155 | 7.4 | 2019-03-03 13:23:00 |
| 3 | 123-19-1176 | A | Yangon | Member | Male | Health and beauty | 58.22 | 8 | 23.2880 | 489.0480 | Ewallet | 465.76 | 4.761905 | 23.2880 | 8.4 | 2019-01-27 20:33:00 |
| 4 | 373-73-7910 | A | Yangon | Normal | Male | Sports and travel | 86.31 | 7 | 30.2085 | 634.3785 | Ewallet | 604.17 | 4.761905 | 30.2085 | 5.3 | 2019-02-08 10:37:00 |
| 5 | 699-14-3026 | C | Naypyitaw | Normal | Male | Electronic accessories | 85.39 | 7 | 29.8865 | 627.6165 | Ewallet | 597.73 | 4.761905 | 29.8865 | 4.1 | 2019-03-25 18:30:00 |
| 6 | 355-53-5943 | A | Yangon | Member | Female | Electronic accessories | 68.84 | 6 | 20.6520 | 433.6920 | Ewallet | 413.04 | 4.761905 | 20.6520 | 5.8 | 2019-02-25 14:36:00 |
| 7 | 315-22-5665 | C | Naypyitaw | Normal | Female | Home and lifestyle | 73.56 | 10 | 36.7800 | 772.3800 | Ewallet | 735.60 | 4.761905 | 36.7800 | 8.0 | 2019-02-24 11:38:00 |
| 8 | 665-32-9167 | A | Yangon | Member | Female | Health and beauty | 36.26 | 2 | 3.6260 | 76.1460 | Credit card | 72.52 | 4.761905 | 3.6260 | 7.2 | 2019-01-10 17:15:00 |
| 9 | 692-92-5582 | B | Mandalay | Member | Female | Food and beverages | 54.84 | 3 | 8.2260 | 172.7460 | Credit card | 164.52 | 4.761905 | 8.2260 | 5.9 | 2019-02-20 13:27:00 |

# Check Datatypes of each column and Bring Date time together in one column.



```
In [112]: dataset.dtypes
```

```
Out[112]:
Invoice ID                                object
Branch                                    object
City                                      object
Customer type                             object
Gender                                    object
Product line                              object
Unit price                               float64
Quantity                                   int64
Tax 5%                                   float64
Total                                    float64
Payment                                   object
cogs                                     float64
gross margin percentage                  float64
gross income                             float64
Rating                                   float64
Date/Time                         datetime64[ns]
dtype: object
```

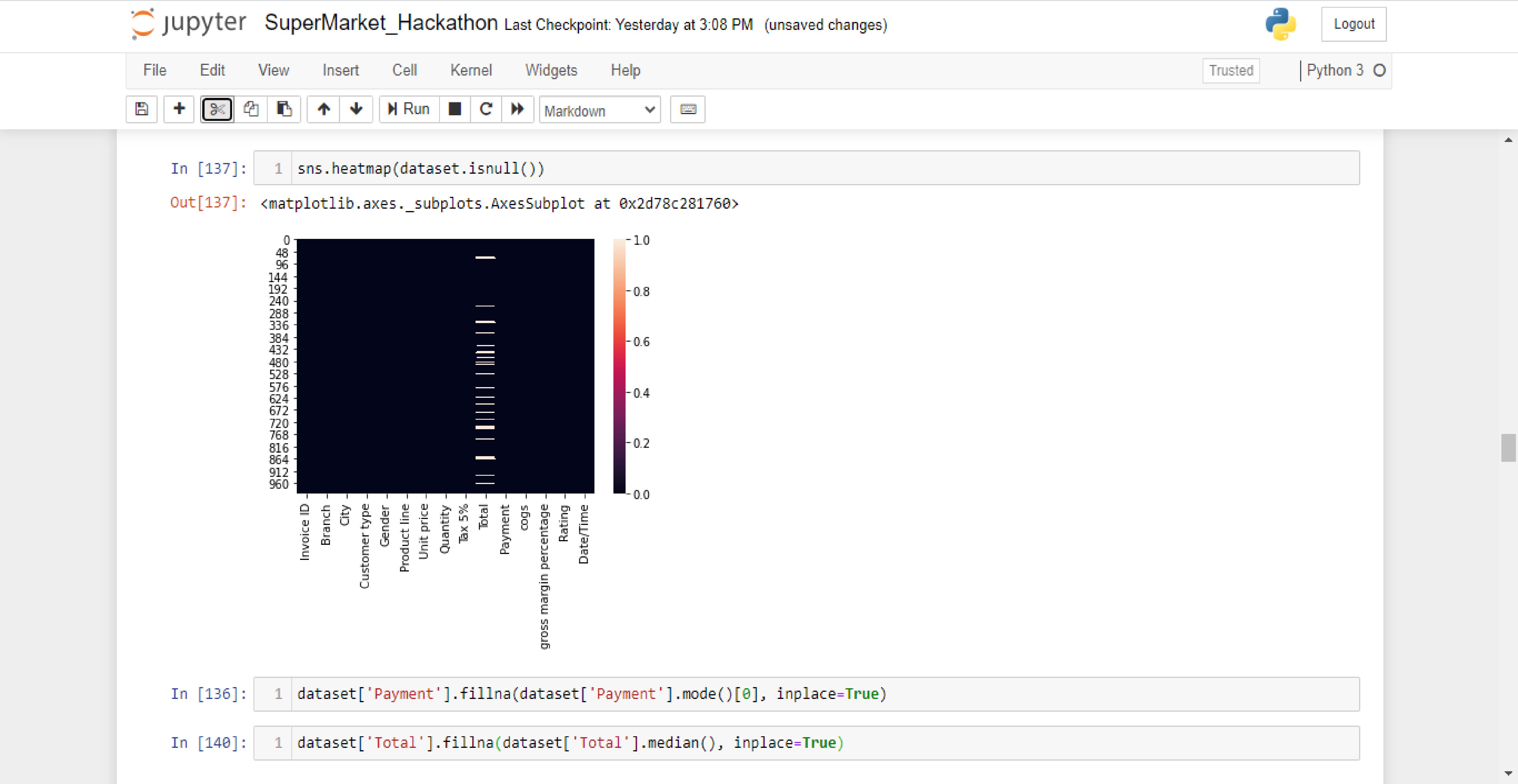
```
In [59]: dataset['Date/Time']=pd.to_datetime(dataset.Date.astype(str)+' '+dataset.Time.astype(str))
```

```
In [60]: dataset.drop(['Date','Time'],axis=1, inplace=True)
```

```
In [61]: dataset.dtypes
```

```
Out[61]:
Invoice ID                                object
Branch                                    object
City                                      object
Customer type                             object
Gender                                    object
Product line                              object
Unit price                               float64
Quantity                                   int64
Tax 5%                                   float64
Total                                    float64
```

# FIND THE NULL VALUES AND TREAT THEM APPROPRIATELY



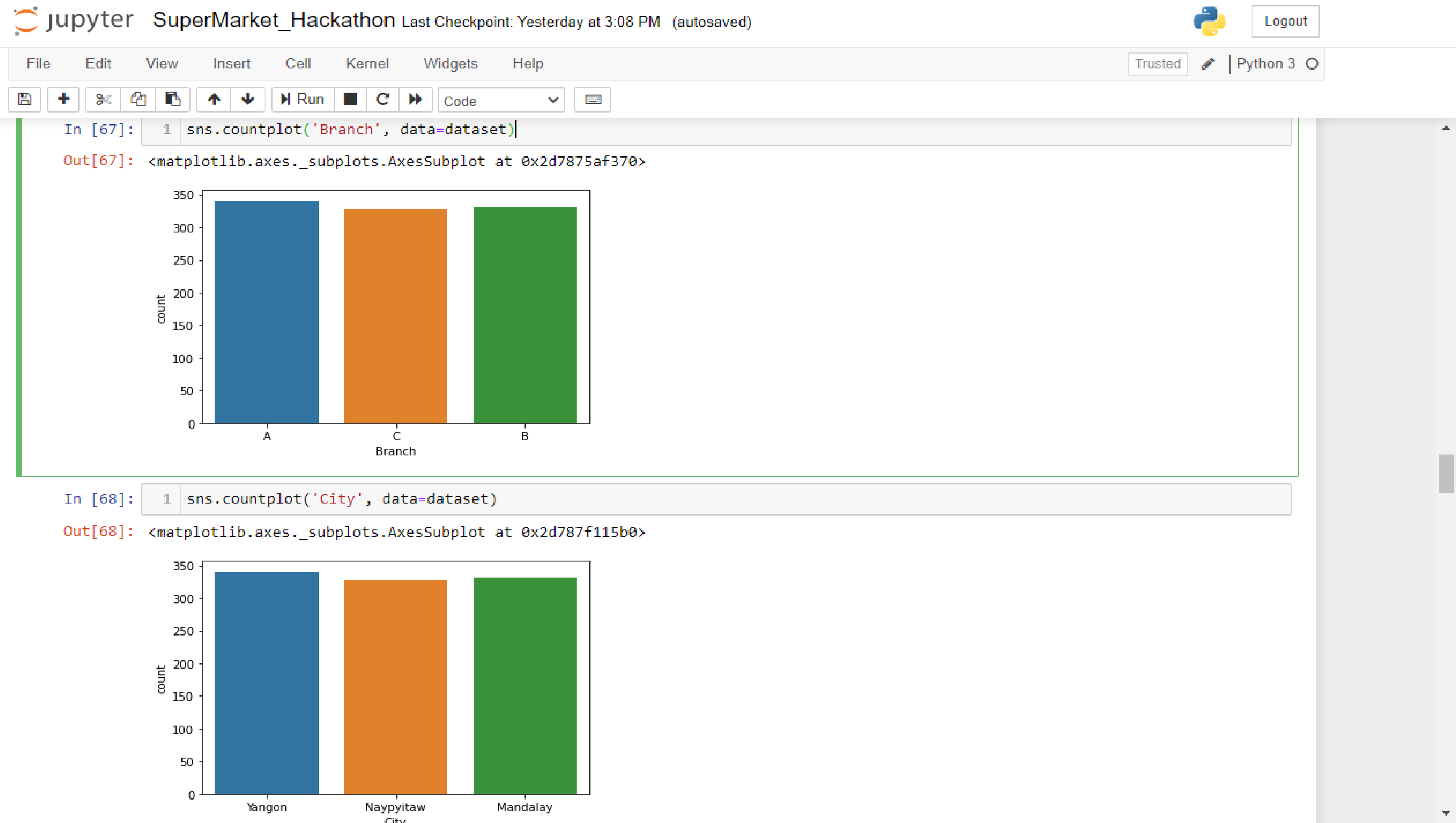
```
In [137]: sns.heatmap(dataset.isnull())
```

```
Out[137]: <matplotlib.axes._subplots.AxesSubplot at 0x2d78c281760>
```

```
In [136]: dataset['Payment'].fillna(dataset['Payment'].mode()[0], inplace=True)
```

```
In [140]: dataset['Total'].fillna(dataset['Total'].median(), inplace=True)
```

# COUNT THE NO. OF BRANCHES AND CITIES



In [67]:

```
sns.countplot('Branch', data=dataset)
```

Out[67]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2d7875af370>
```

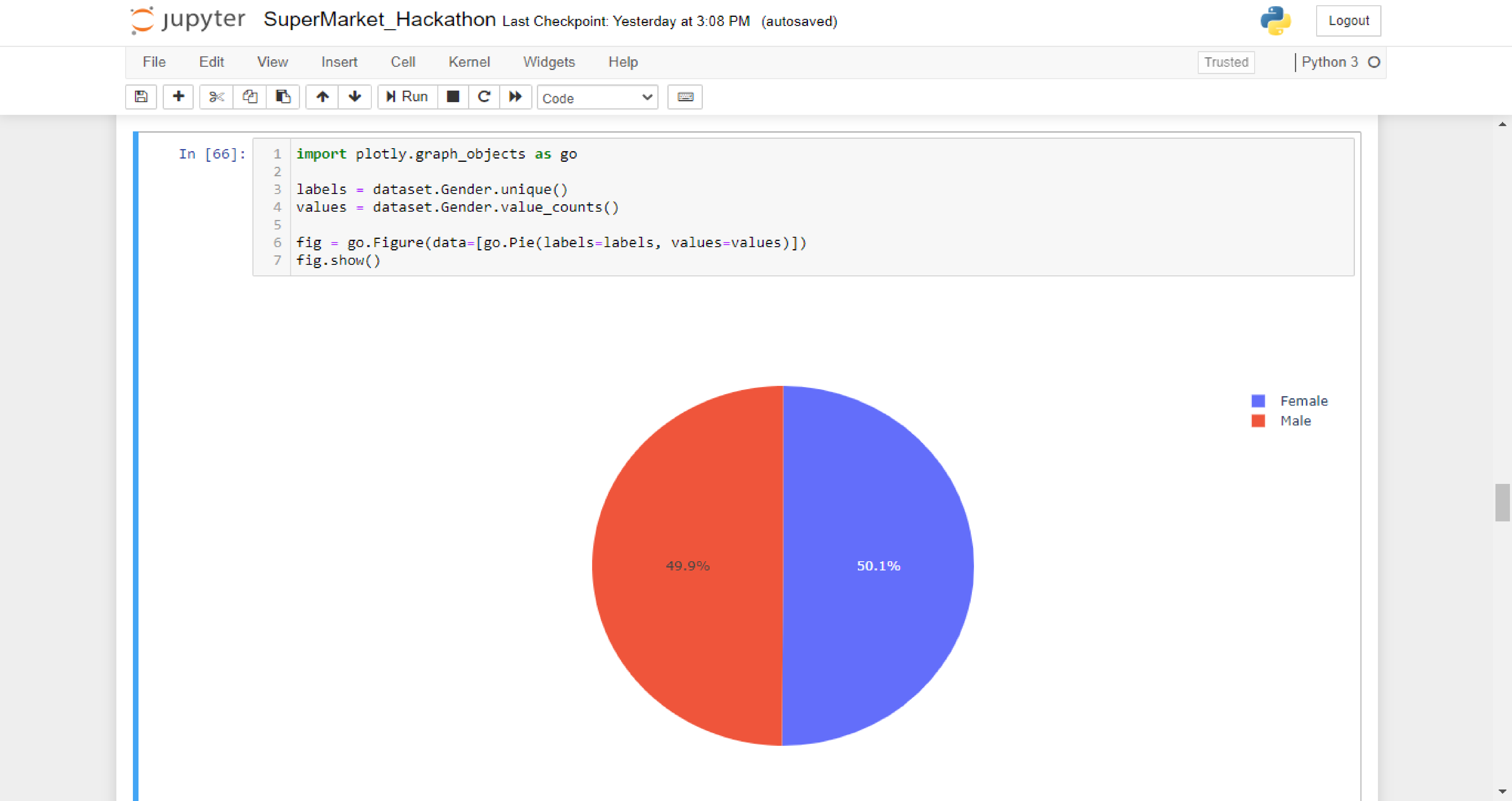
In [68]:

```
sns.countplot('City', data=dataset)
```

Out[68]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2d787f115b0>
```

# PIE CHART OF GENDER

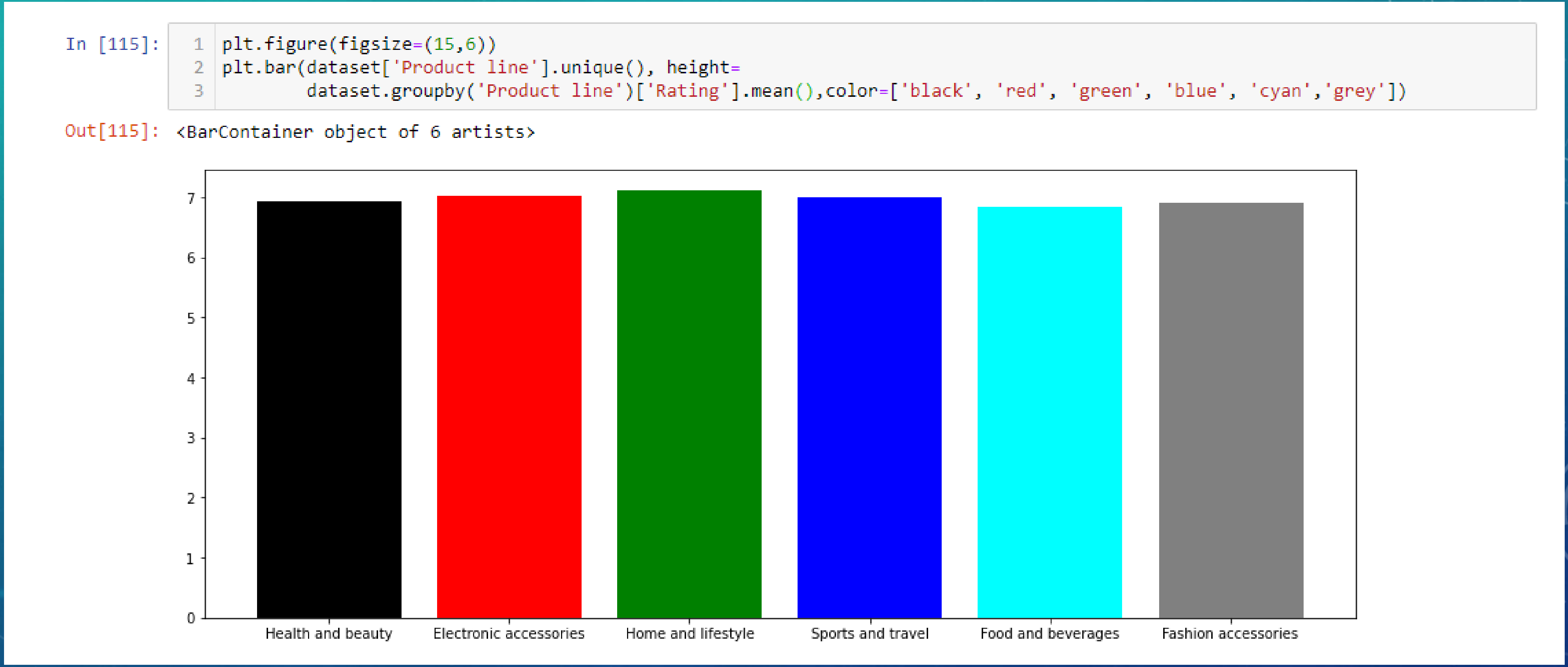


```python
import plotly.graph_objects as go

labels = dataset.Gender.unique()
values = dataset.Gender.value_counts()

fig = go.Figure(data=[go.Pie(labels=labels, values=values)])
fig.show()
```

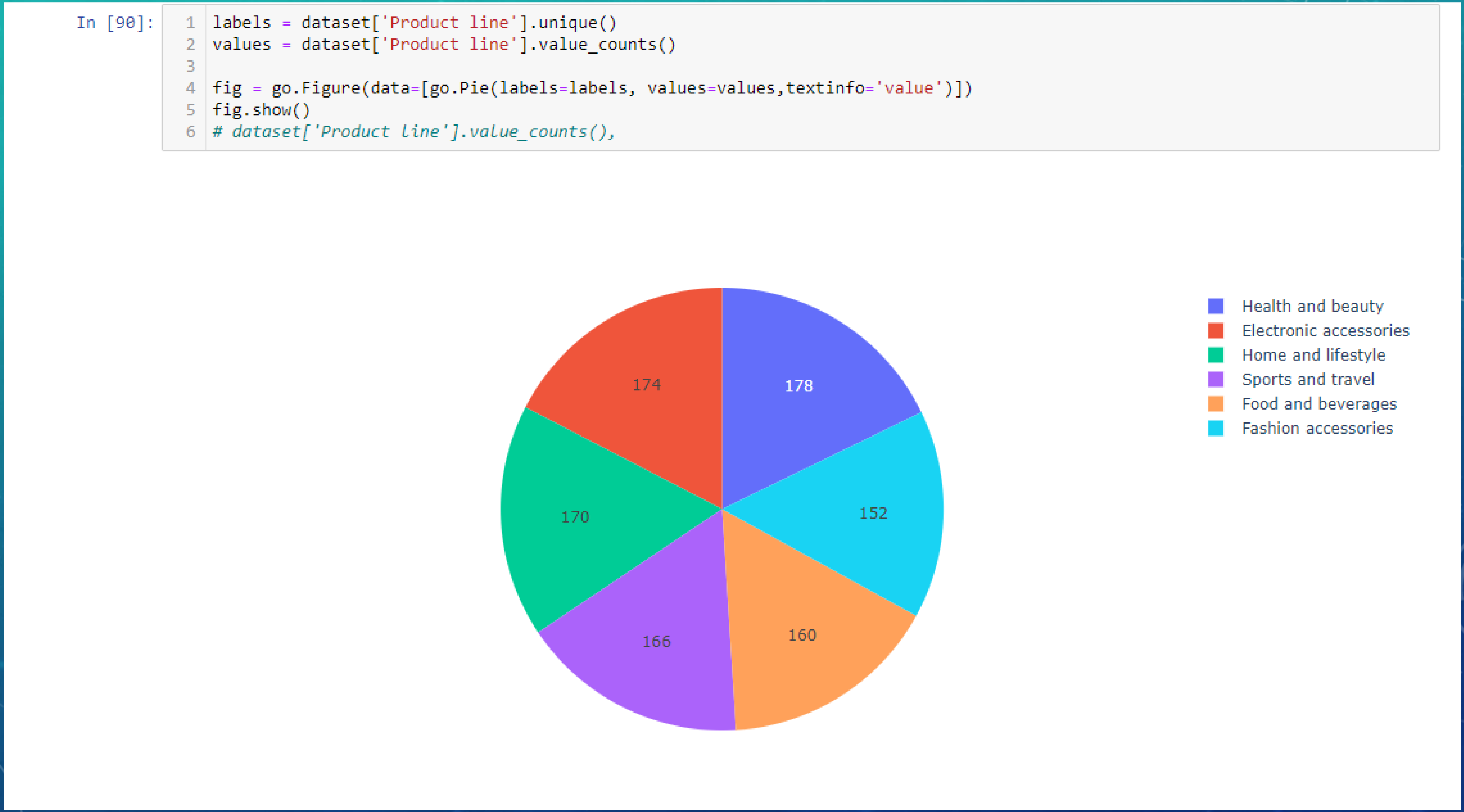
# CALCULATE AVERAGE RATING OF EACH PRODUCT LINE



```
In [115]:  plt.figure(figsize=(15,6))
           plt.bar(dataset['Product line'].unique(), height=
                   dataset.groupby('Product line')['Rating'].mean(),color=['black', 'red', 'green', 'blue', 'cyan','grey'])

Out[115]:  <BarContainer object of 6 artists>
```

# NUMBER PRODUCTS IN EACH CATEGORY

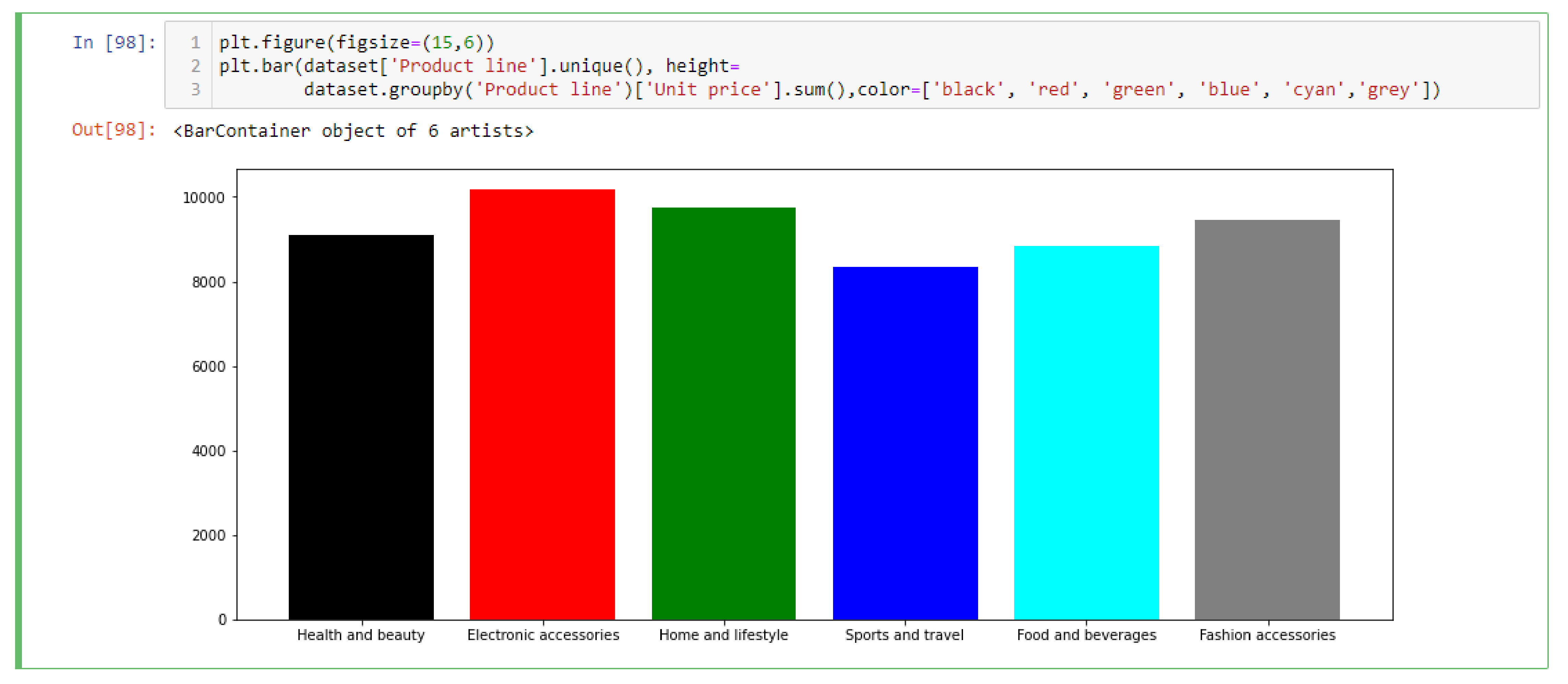


In [90]:

```
labels = dataset['Product line'].unique()
values = dataset['Product line'].value_counts()

fig = go.Figure(data=[go.Pie(labels=labels, values=values,textinfo='value')])
fig.show()
# dataset['Product line'].value_counts(),
```

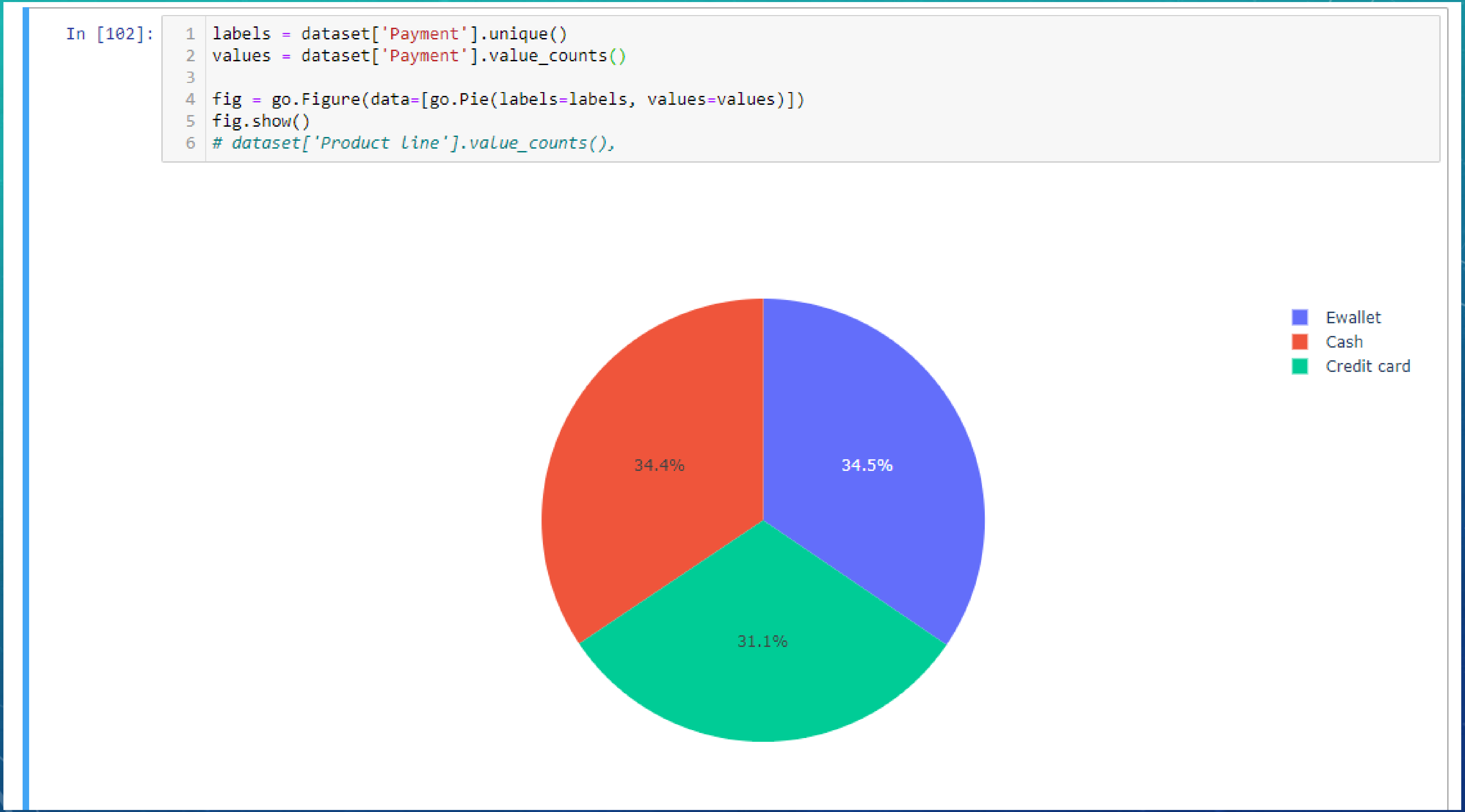
# TOTAL AMOUNT COLLECTED IN EACH PRODUCT LINE



```
In [98]:  plt.figure(figsize=(15,6))
          plt.bar(dataset['Product line'].unique(), height=
                  dataset.groupby('Product line')['Unit price'].sum(),color=['black', 'red', 'green', 'blue', 'cyan','grey'])
```

```
Out[98]: <BarContainer object of 6 artists>
```

# FIND OUT HIGHEST PERCENTAGE OF PAYMENT METHOD

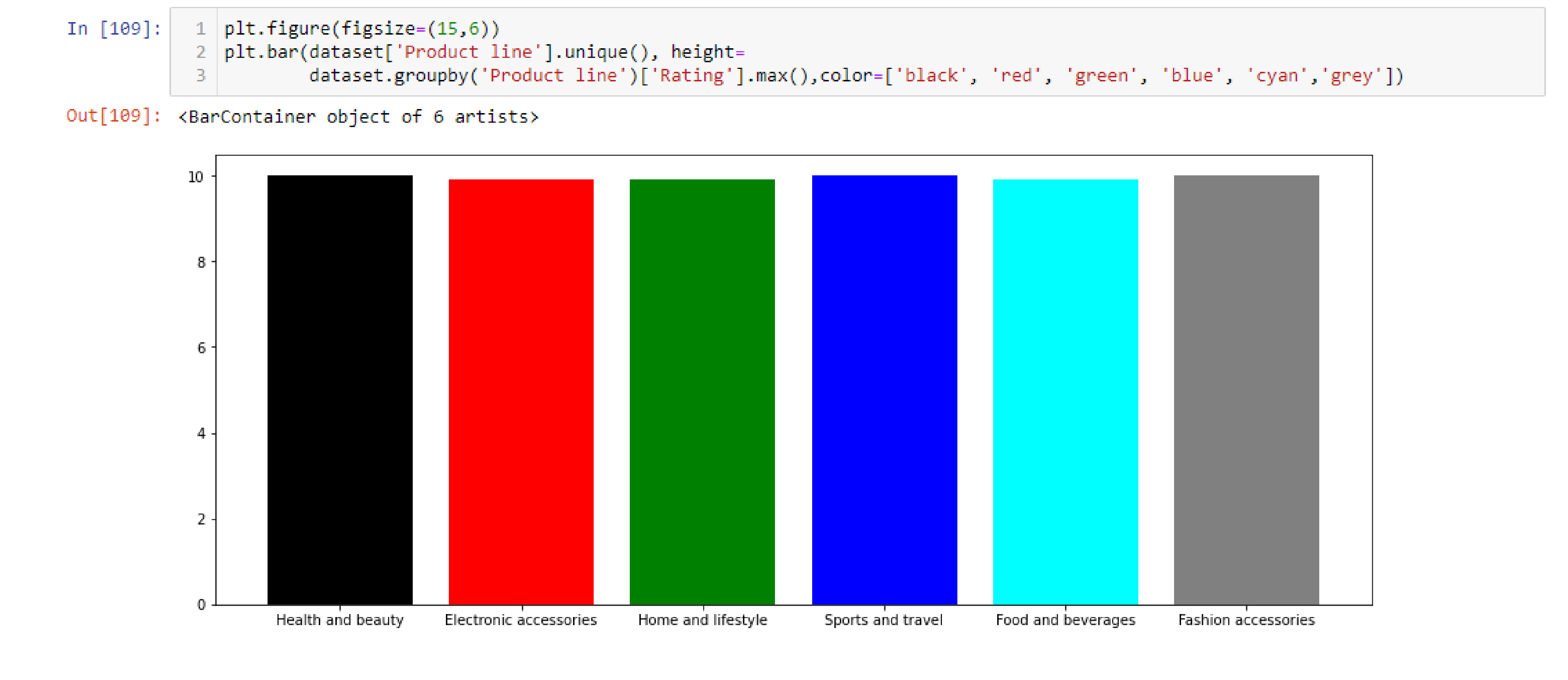


In [102]:

```
labels = dataset['Payment'].unique()
values = dataset['Payment'].value_counts()

fig = go.Figure(data=[go.Pie(labels=labels, values=values)])
fig.show()
# dataset['Product line'].value_counts(),
```

# FIND OUT THE CATEGORY WITH HIGHEST RATING



In [109]:

```
plt.figure(figsize=(15,6))
plt.bar(dataset['Product line'].unique(), height=
        dataset.groupby('Product line')['Rating'].max(),color=['black', 'red', 'green', 'blue', 'cyan','grey'])
```

Out[109]:

```
<BarContainer object of 6 artists>
```