```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
df=pd.read_csv("/content/supermarket.csv")
df
```

```
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.head()
df.describe()
```

```
df.isnull()
```

```
df.tail()
df.size
```

```
17000
```

```
df.shape
```

```
(1000, 17)
```

```
df.columns
```

```
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')
```

```
df.duplicated()
```

```
0      False
1      False
2      False
3      False
4      False
       ...
995    False
996    False
997    False
998    False
999    False
Length: 1000, dtype: bool
```

```
g=df['Gender'].value_counts()
g
```

```
Gender
Female    501
Male      499
Name: count, dtype: int64
```

```
sns.countplot(x=df['Gender'])
plt.title("Total number of Male and Female")
```

```
sns.distplot(df['Rating'])   #density check
plt.grid()
```

```
df["Branch"].value_counts()
sns.countplot(x=df['Branch'])
plt.title("count vs Branch")
```

```
sns.countplot(x=df['Payment'])
plt.title("Count vs method of Payment")
plt.xlabel("Method of Payment")
plt.ylabel("Count")
```

```
a=df["Payment"].value_counts()
a
```

```
Payment
Ewallet        345
Cash           344
Credit card    311
Name: count, dtype: int64
```

```
plt.figure(figsize=(14,6))
plt.style.use('classic')
ax=sns.countplot(x="Payment",hue="Branch",data=df,palette="tab20")
ax.set_title(label="Payment distribution in all branches",fontsize=25)
ax.set_xlabel(xlabel="Payment",fontsize=16)
ax.set_ylabel(ylabel="People count",fontsize=16)
```

```
sns.set_style('darkgrid')
sns.scatterplot(x=df['Rating'],y=df['gross income'])
plt.title("Gross income vs Rating")
plt.figure(figsize=(8,4))
ax=sns.boxplot(x="Branch",y="Rating",data=df,palette="RdYlBu")
ax.set_title("Rating distribution in all Branches",fontsize=25)
ax.set_xlabel(xlabel="branches",fontsize=16)
ax.set_ylabel(ylabel="Rating distribution",fontsize=16)
plt.grid()
```

```
sns.boxplot(x=df['Branch'],y=df['gross income'])
plt.title('Gross Income vs Branch',color='red')
```

```
sns.boxplot(x=df['Gender'],y=df['gross income'])
plt.title('gross income vs Gender',color='red')
```

```
df1=df.select_dtypes(include='number')
df1.groupby(df.index).mean()
```

```
sns.lineplot(x=df1.groupby(df.index).mean().index,
             y=df1.groupby(df.index).mean()['gross income'])
```

```
cat=df[["Product line","gross income"]].groupby(['Product line']).sum().reset_index()
plt.figure(figsize=(20,8))
sns.barplot(x='Product line',y='gross income',data=cat)
plt.title("Gross income vs product line")
```

```
plt.figure(figsize=(16,6))
plt.title('Total Monthaly transaction by Gender')
sns.countplot(x=df['Product line'],hue=df.Gender)
```

```
plt.figure(figsize=(12,6))
sns.distplot(x=df['Quantity'])
```

```
df['Date']=pd.to_datetime(df['Date'])
df['Weekday']=df['Date'].dt.day_name()
df.set_index('Date',inplace=True)
df.head()
```

```
plt.figure(figsize=(8,6))
plt.title('Daily Sales by Day of the Week')
sns.countplot(x=df['Weekday'])
```

```
df['Time']=pd.to_datetime(df['Time'])
df['Hour']=(df['Time']).dt.hour
df['Hour'].unique()
```

```
<ipython-input-29-9d601d3b55e2>:1: UserWarning: Could not infer format, so each element will be parsed individually, falling back to `da
  df['Time']=pd.to_datetime(df['Time'])
array([13, 10, 20, 18, 14, 11, 17, 16, 19, 15, 12], dtype=int32)
```

```
sns.lineplot(x="Hour",y='Quantity',data=df).set_title('product sales per Hour')
```

```
plt.figure(figsize=(12,6))
sns.barplot(y=df['Product line'],x=df['Rating'])
```

```
plt.figure(figsize=(20,7))
sns.barplot(x=df['City'],y=df['gross income'])
plt.xlabel('City name',fontsize='16')
plt.xticks(fontsize='16')
plt.ylabel('Gross income',fontsize='16')
plt.yticks(fontsize='16')
```

Start coding or generate with AI.

Start coding or generate with AI.