```python
#import python libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

#import csv file
df = pd.read_csv("Sales Data.csv",encoding="unicode_escape")

df.shape
```

```
(11251, 15)
```

```python
df.head()
```

{"summary":"{\n \"name\": \"df\",\n \"rows\": 11251,\n \"fields\": [\n   {\n   \"column\": \"User_ID\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 1716,\n   \"min\": 1000001,\n   \"max\": 1006040,\n   \"num_unique_values\": 3755,\n   \"samples\": [\n   1005905,\n   1003730,\n   1005326\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Cust_name\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 1250,\n   \"samples\": [\n   \"Nida\",\n   \"Lacy\",\n   \"Caudle\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Product_ID\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 2351,\n   \"samples\": [\n   \"P00224442\",\n   \"P00205242\",\n   \"P00347442\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Gender\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 2,\n   \"samples\": [\n   \"M\",\n   \"F\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Age Group\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 7,\n   \"samples\": [\n   \"26-35\",\n   \"0-17\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Age\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 12,\n   \"min\": 12,\n   \"max\": 92,\n   \"num_unique_values\": 81,\n   \"samples\": [\n   18,\n   28\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Marital_Status\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0,\n   \"min\": 0,\n   \"max\": 1,\n   \"num_unique_values\": 2,\n   \"samples\":

```
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"State\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 16,\n
\"samples\": [\n          \"Maharashtra\",\n          \"Andhra\\
u00a0Pradesh\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Zone\",\n      \"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n
\"Southern\",\n          \"Eastern\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Occupation\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 15,\n        \"samples\": [\n
\"Retail\",\n          \"Aviation\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Product_Category\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 18,\n        \"samples\": [\n
\"Auto\",\n          \"Hand & Power Tools\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Orders\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 1,\n
\"min\": 1,\n        \"max\": 4,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          3,\n          4\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Amount\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
5222.355869186444,\n        \"min\": 188.0,\n        \"max\":
23952.0,\n        \"num_unique_values\": 6584,\n        \"samples\":
[\n          19249.0,\n          13184.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Status\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": null,\n
\"min\": null,\n        \"max\": null,\n        \"num_unique_values\":
0,\n        \"samples\": [],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"unnamed1\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": null,\n        \"min\": null,\n
\"max\": null,\n        \"num_unique_values\": 0,\n
\"samples\": [],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"df"}
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11251 entries, 0 to 11250
Data columns (total 15 columns):
 #   Column            Non-Null Count  Dtype
```

```
---  ------            --------------  -----
 0   User_ID           11251 non-null  int64
 1   Cust_name         11251 non-null  object
 2   Product_ID        11251 non-null  object
 3   Gender            11251 non-null  object
 4   Age Group         11251 non-null  object
 5   Age               11251 non-null  int64
 6   Marital_Status    11251 non-null  int64
 7   State             11251 non-null  object
 8   Zone              11251 non-null  object
 9   Occupation        11251 non-null  object
 10  Product_Category  11251 non-null  object
 11  Orders            11251 non-null  int64
 12  Amount            11239 non-null  float64
 13  Status            0 non-null      float64
 14  unnamed1          0 non-null      float64
dtypes: float64(3), int64(4), object(8)
memory usage: 1.3+ MB
```

```
#drop unrelated / blank columns
df.drop(['Status','unnamed1'],axis=1,inplace=True)

#check for null values
pd.isnull(df).sum()
```

```
User_ID              0
Cust_name            0
Product_ID           0
Gender               0
Age Group            0
Age                  0
Marital_Status       0
State                0
Zone                 0
Occupation           0
Product_Category     0
Orders               0
Amount              12
dtype: int64
```

```
#drop null values
df.dropna(inplace=True)

#change datatype
df['Amount']=df['Amount'].astype('int')

df['Amount'].dtypes
```

```
dtype('int64')
```

```
df.columns
```

```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group',
'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation',
'Product_Category',
       'Orders', 'Amount'],
      dtype='object')

#rename column
df.rename(columns={'Marital_Status':'Shaadi'})
```

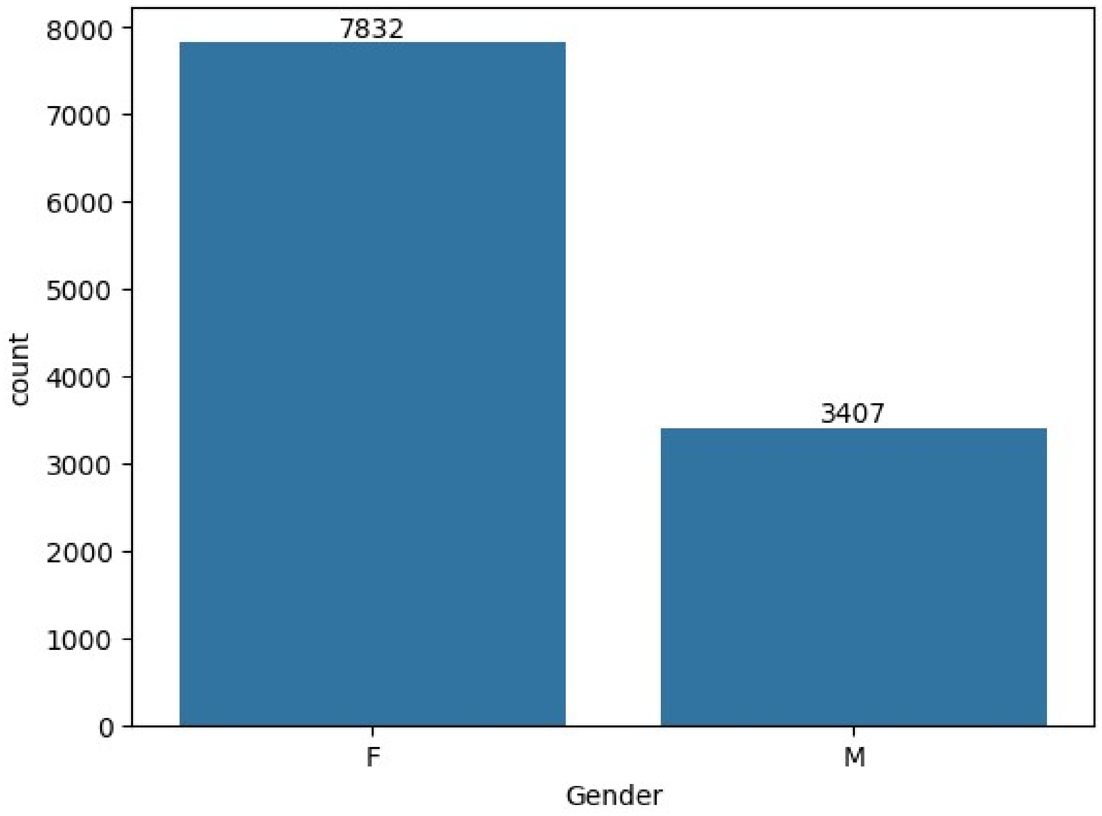
```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 11239,\n  \"fields\":
[\n    {\n      \"column\": \"User_ID\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1716,\n        \"min\":
1000001,\n        \"max\": 1006040,\n        \"num_unique_values\":
3752,\n        \"samples\": [\n          1002014,\n          1003491,\
n          1001842\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Cust_name\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 1250,\n
\"samples\": [\n          \"Hallsten\",\n          \"Shubham\",\n
\"Riya\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Product_ID\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2350,\n
\"samples\": [\n          \"P00133342\",\n          \"P00302142\",\n
\"P00227542\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Gender\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"M\",\n          \"F\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Age Group\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 7,\n        \"samples\": [\n          \"26-
35\",\n          \"0-17\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Age\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 12,\n        \"min\": 12,\n
\"max\": 92,\n        \"num_unique_values\": 81,\n        \"samples\":
[\n          38,\n          28\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Shaadi\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"State\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 16,\n
\"samples\": [\n          \"Maharashtra\",\n          \"Andhra\\
u00a0Pradesh\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Zone\",\n      \"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 5,\n        \"samples\": [\n
\"Southern\",\n          \"Eastern\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Occupation\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 15,\n        \"samples\": [\n          \"IT
Sector\",\n          \"Hospitality\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Product_Category\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 18,\n        \"samples\": [\n
\"Auto\",\n          \"Hand & Power Tools\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Orders\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 1,\n
\"min\": 1,\n        \"max\": 4,\n        \"num_unique_values\": 4,\n
\"samples\": [\n          3,\n          4\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Amount\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 5222,\n
\"min\": 188,\n        \"max\": 23952,\n        \"num_unique_values\":
6583,\n        \"samples\": [\n          19247,\n          5293\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe"}
```

```
#using describe() method to call specific columns
df[['Age','Orders','Amount']].describe()
```

```
{"summary":"{\n  \"name\": \"df[['Age','Orders','Amount']]\",\n
\"rows\": 8,\n  \"fields\": [\n    {\n      \"column\": \"Age\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
3960.7779927819724,\n        \"min\": 12.0,\n        \"max\":
11239.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n
35.41035679330901,\n          33.0,\n          11239.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Orders\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
3972.7985251346995,\n        \"min\": 1.0,\n        \"max\": 11239.0,\
n        \"num_unique_values\": 7,\n        \"samples\": [\n
11239.0,\n          2.4896343091022333,\n          3.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Amount\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
7024.070687950828,\n        \"min\": 188.0,\n        \"max\":
23952.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n
9453.610552540262,\n          8109.0,\n          11239.0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe"}
```

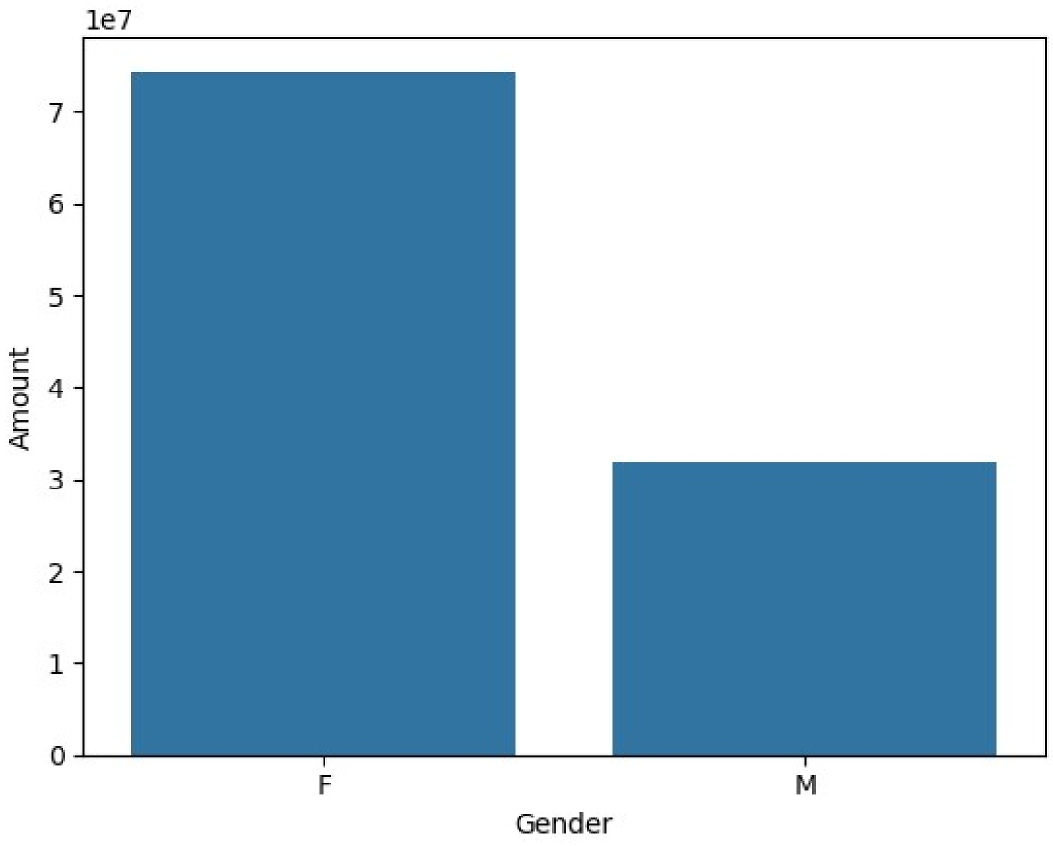
Univariate Analysis corresponds to a single variable, but in Bivariate there is analysis between two variables

```
#Univariate Analysis
#plotting a bar chart for Gender and its count
ax = sns.countplot(x = 'Gender',data = df)
for bars in ax.containers:
    ax.bar_label(bars)
```



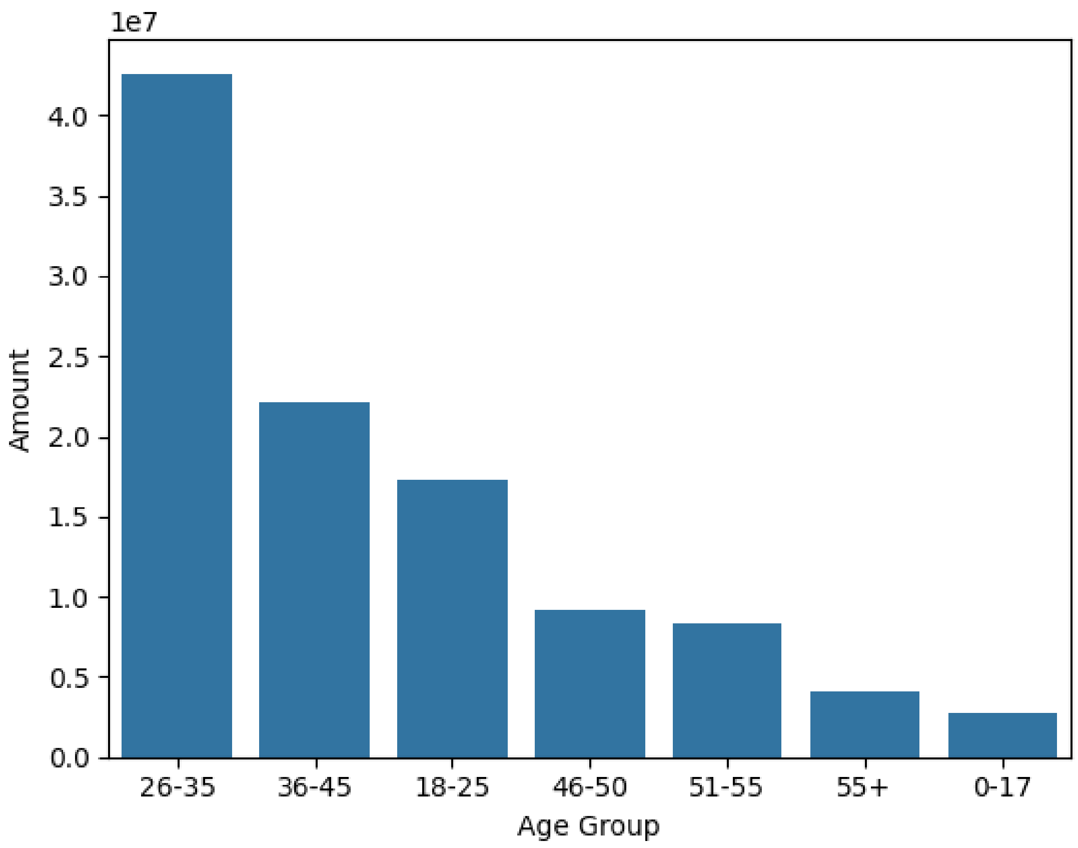
This shows that there are more female customers as compared to male customers

```
#plotting a bar chart for Gender vs Total Amount
sales_gen = df.groupby(['Gender'], as_index=False)
['Amount'].sum().sort_values(by='Amount',ascending=False)
sns.barplot(x = 'Gender',y = 'Amount' ,data = sales_gen)

<Axes: xlabel='Gender', ylabel='Amount'>
```



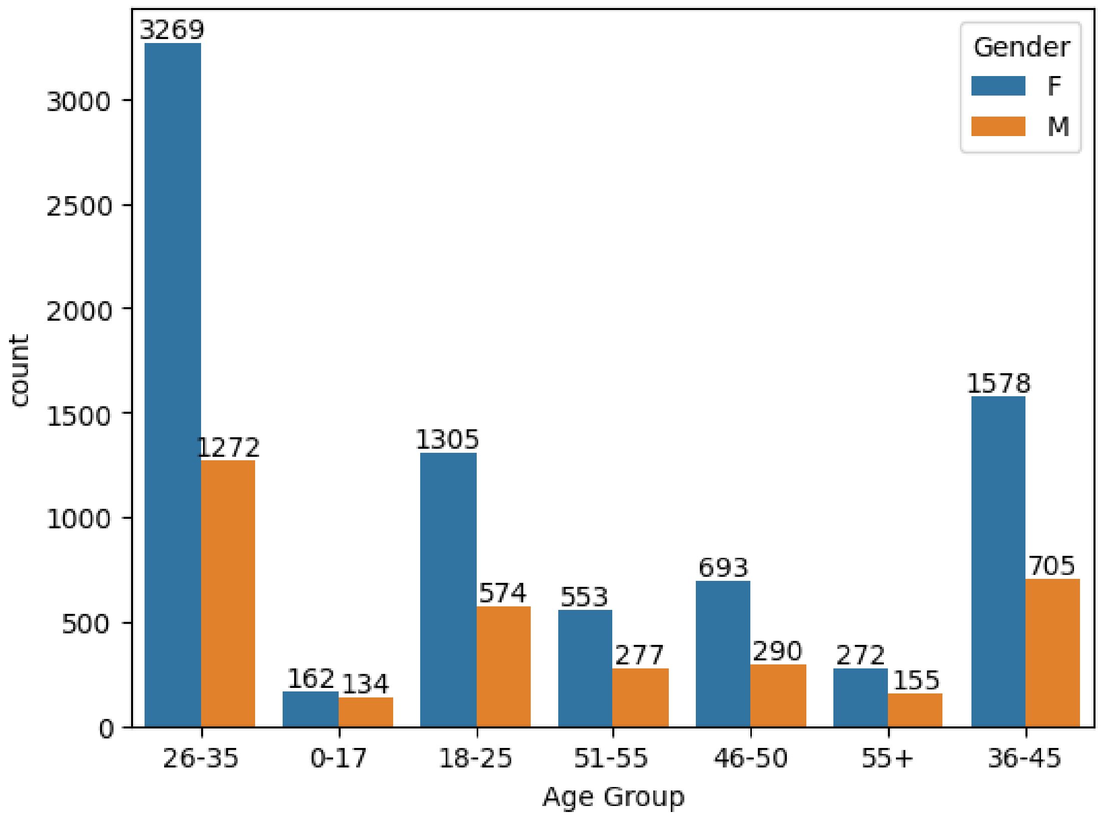
This shows that there are more female customers as compared to male customers

```
sales_gen = df.groupby(['Age Group'], as_index=False)
['Amount'].sum().sort_values(by='Amount',ascending=False)
sns.barplot(x = 'Age Group',y = 'Amount' ,data = sales_gen)

<Axes: xlabel='Age Group', ylabel='Amount'>
```



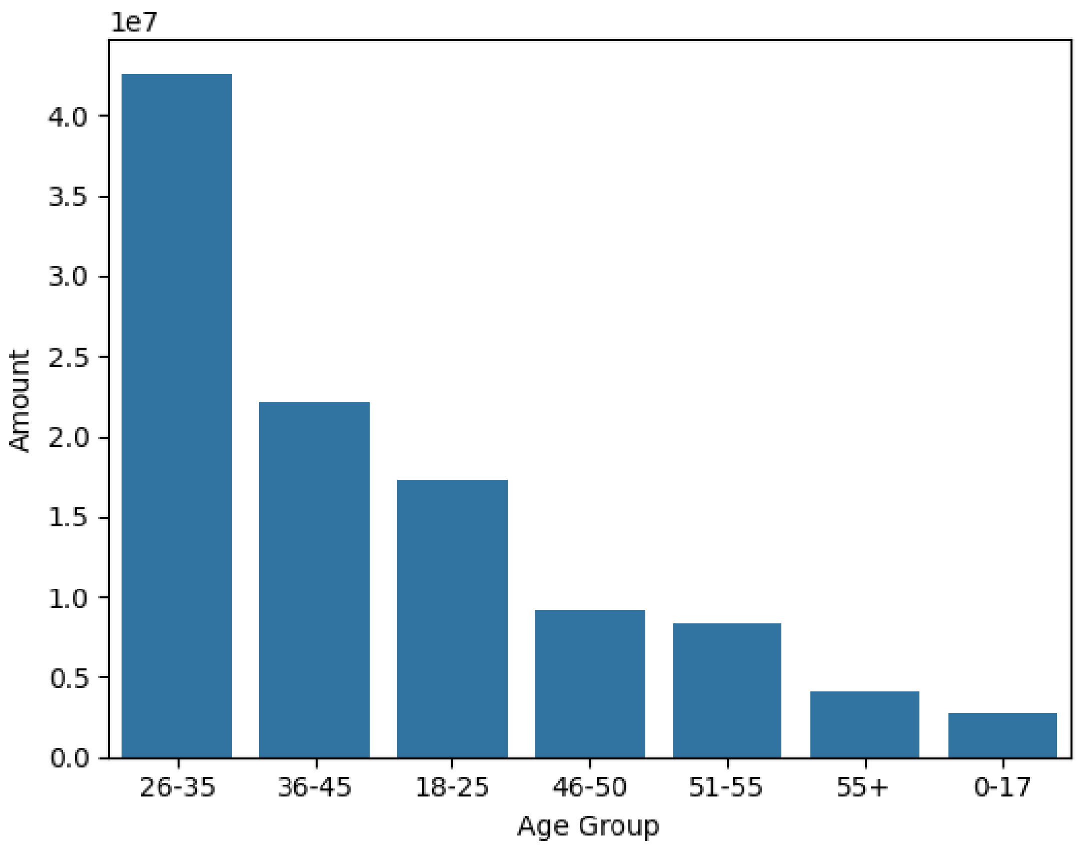
This bar graph shows that the age group in which most of the sales are from is the 26-35 age group

```
#Bivariate Analysis
ax = sns.countplot(data = df, x = 'Age Group', hue = 'Gender')
for bars in ax.containers:
    ax.bar_label(bars)
```



This graph shows the comparision between the different age group and gender

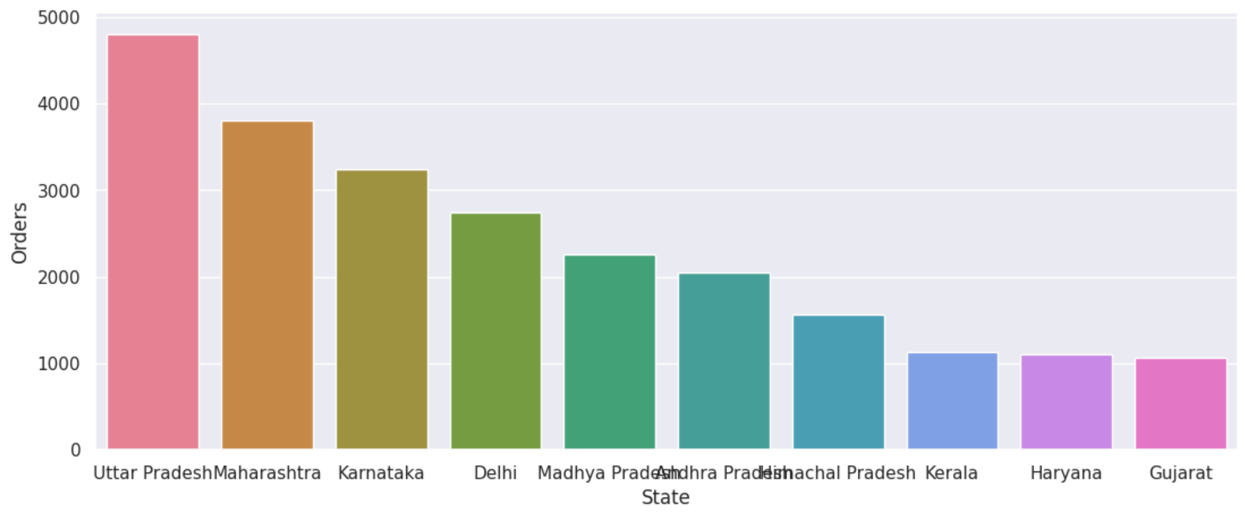
```
#Total Amount vs Age Group
sales_age = df.groupby(['Age Group'], as_index=False)
['Amount'].sum().sort_values(by='Amount', ascending=False)
sns.barplot(x = 'Age Group',y = 'Amount' ,data = sales_age)

<Axes: xlabel='Age Group', ylabel='Amount'>
```



This graph shows the analysis between Amount and Age Group

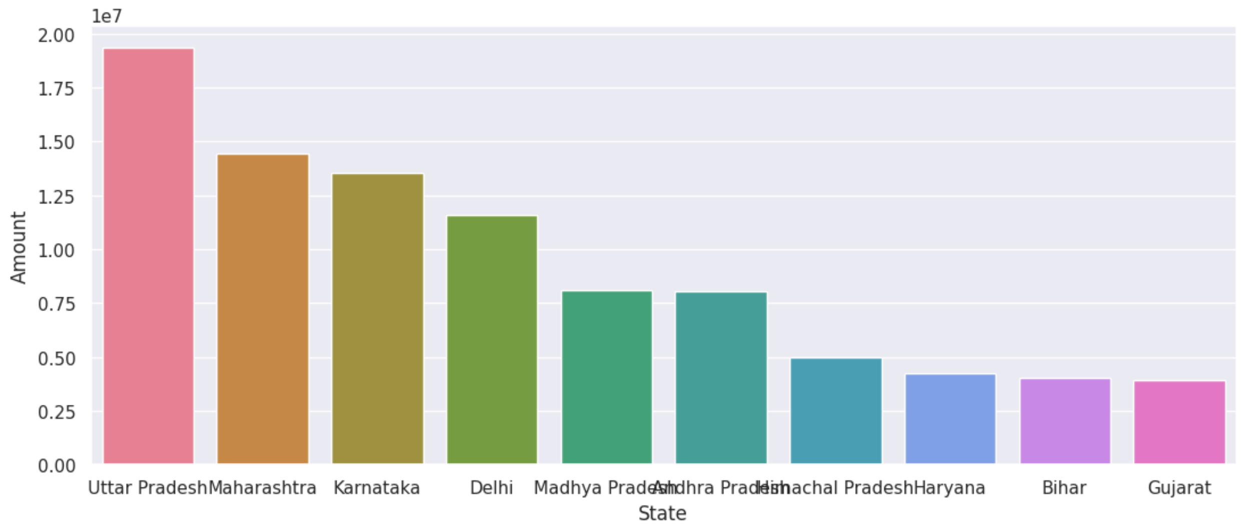
```
#total number of orders from top 10 states
sales_state = df.groupby(['State'], as_index=False)
['Orders'].sum().sort_values(by='Orders', ascending=False).head(10)
custom_color=sns.color_palette("husl",10)
sns.set(rc={'figure.figsize': (13, 5)})
sns.barplot(data = sales_state,x = 'State',y = 'Orders',palette =
custom_color)
```

```
<ipython-input-52-6924b22f82df>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(data = sales_state,x = 'State',y = 'Orders',palette =
custom_color)

<Axes: xlabel='State', ylabel='Orders'>
```



This graph shows top 10 states having highest orders

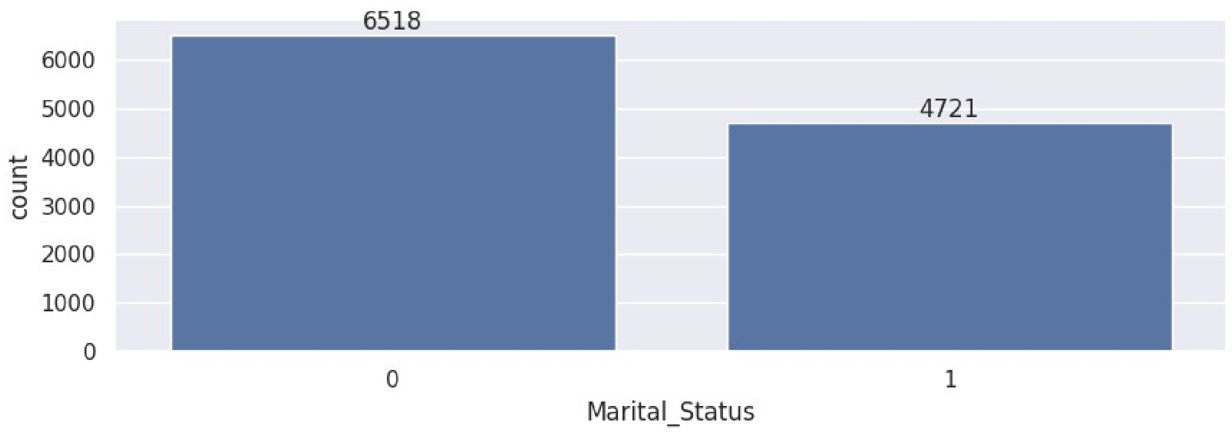
```
#total amount / sales from top 10 states
total_sales_state = df.groupby(['State'], as_index=False)
['Amount'].sum().sort_values(by='Amount', ascending=False).head(10)
custom_color=sns.color_palette("husl",10)
sns.set(rc={'figure.figsize': (13, 5)})
sns.barplot(data = total_sales_state,x = 'State',y = 'Amount',palette
= custom_color)

<ipython-input-58-836b20efe60b>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(data = total_sales_state,x = 'State',y =
'Amount',palette = custom_color)

<Axes: xlabel='State', ylabel='Amount'>
```



This graph shows the total amount from top 10 states

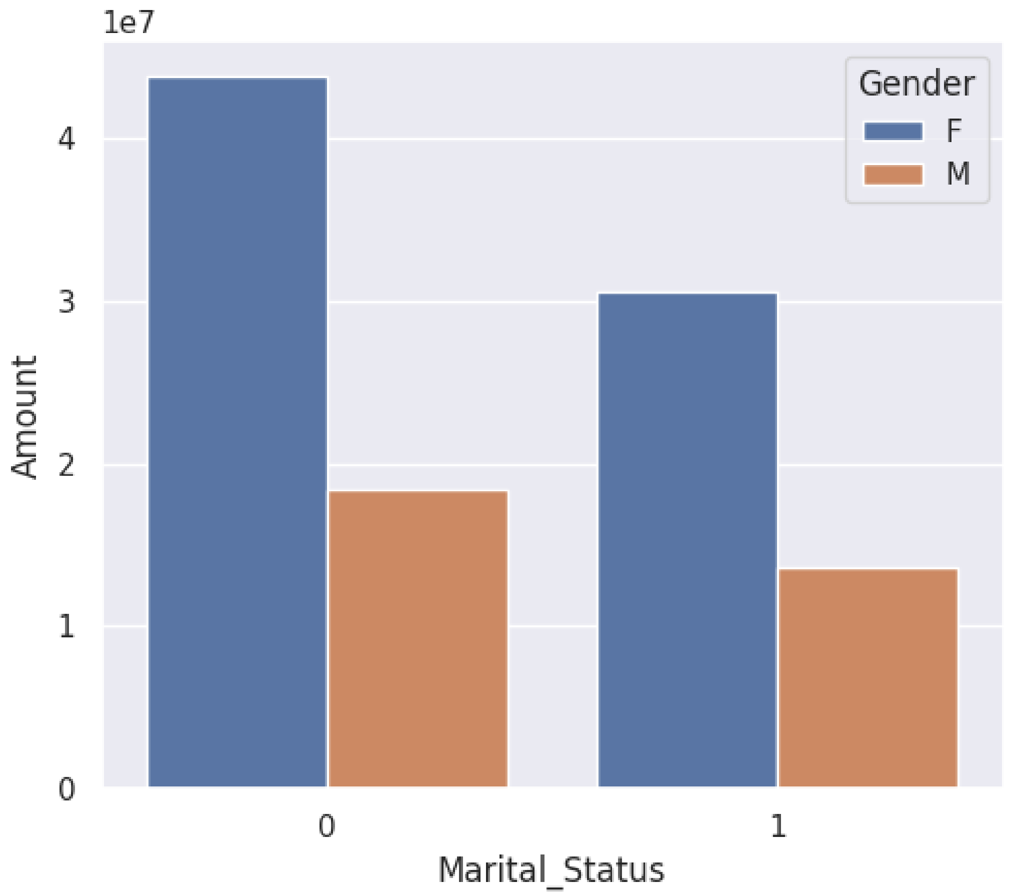
Marital Status

```
ax = sns.countplot(data = df, x = 'Marital_Status')
sns.set(rc={'figure.figsize':(7,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



This graph shows the marital data analysis, 0 stands for unmarried, and 1 stands for married

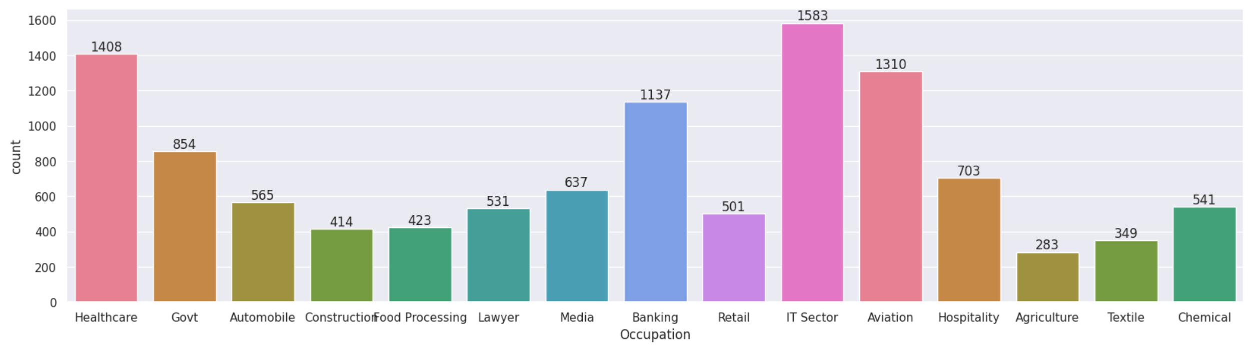
```
sales_state = df.groupby(['Marital_Status', 'Gender'], as_index=False)
['Amount'].sum().sort_values(by='Amount', ascending=False)
sns.set(rc={'figure.figsize':(6,5)})
sns.barplot(data = sales_state, x = 'Marital_Status',y= 'Amount',
hue='Gender')

<Axes: xlabel='Marital_Status', ylabel='Amount'>
```



This graph shows the analysis between different gender, their marital status and the amount

```
sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = df, x = 'Occupation',palette=custom_color)
for bars in ax.containers:
    ax.bar_label(bars)
```

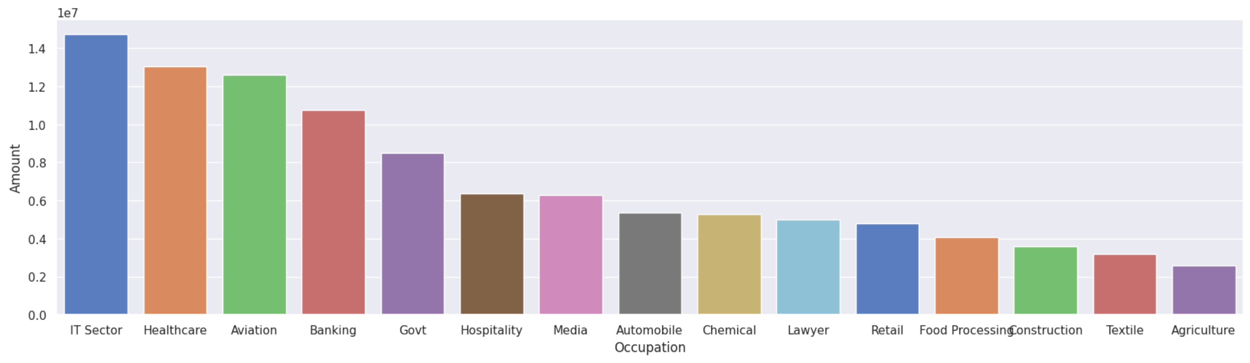
```
<ipython-input-72-6b6b7dd6eb61>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  ax = sns.countplot(data = df, x = 'Occupation',palette=custom_color)
<ipython-input-72-6b6b7dd6eb61>:2: UserWarning:
The palette list has fewer values (10) than needed (15) and will
cycle, which may produce an uninterpretable plot.
  ax = sns.countplot(data = df, x = 'Occupation',palette=custom_color)
```



This graph shows the count of various occupations in the DataFrame

```
sales_state = df.groupby(['Occupation'],as_index=False)
['Amount'].sum().sort_values(by='Amount',ascending=False)
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state,x = 'Occupation',y= 'Amount',palette =
"muted")
```

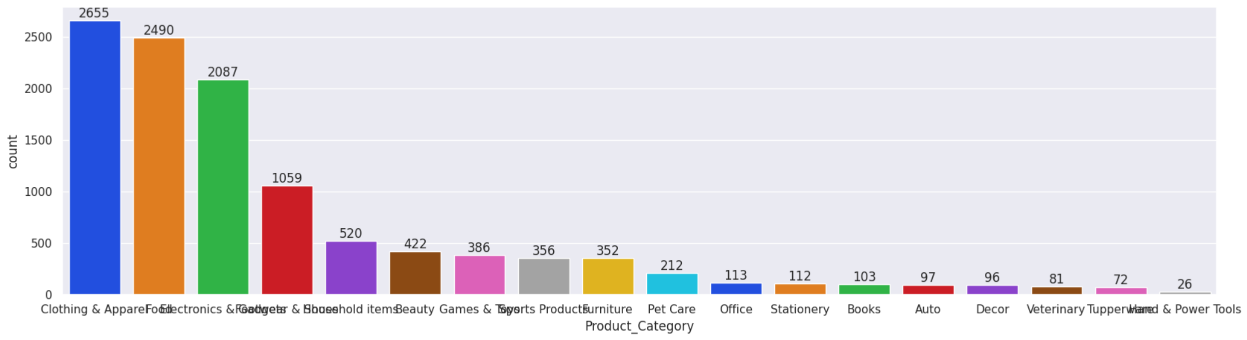
```
<ipython-input-99-4c9654f8f9ae>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(data = sales_state,x = 'Occupation',y= 'Amount',palette
= "muted")

<Axes: xlabel='Occupation', ylabel='Amount'>
```



This graph shows the analysis of the total amount spent by each sector in the DataFrame

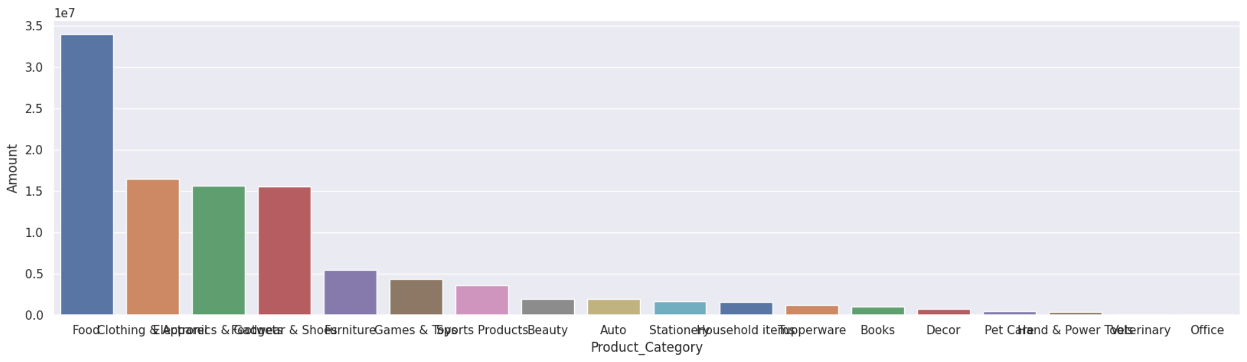
```
sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = df, x = 'Product_Category',palette =
"bright",order = df['Product_Category'].value_counts().index)
for bars in ax.containers:
    ax.bar_label(bars)
#df['Product_Category'].value_counts().index shows the data in
ascending order in countplot
```

```
<ipython-input-98-b57225c6b556>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  ax = sns.countplot(data = df, x = 'Product_Category',palette =
"bright",order = df['Product_Category'].value_counts().index)
```



This graph shows the count of various products

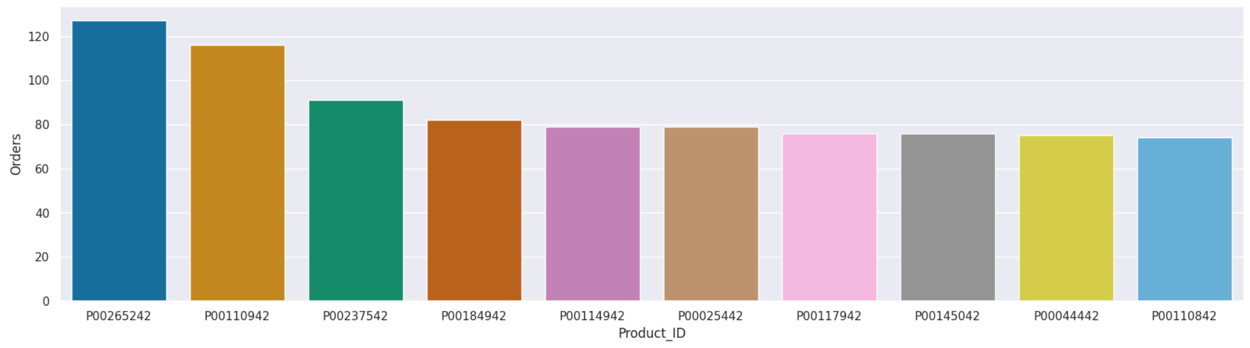
```
sales_state = df.groupby(['Product_Category'], as_index=False)
['Amount'].sum().sort_values(by='Amount',ascending=False)
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state,x = 'Product_Category',y =
'Amount',palette = "deep")

<ipython-input-97-b667f43e45b3>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(data = sales_state,x = 'Product_Category',y =
'Amount',palette = "deep")

<Axes: xlabel='Product_Category', ylabel='Amount'>
```



This graph shows the analysis between different Product Category and the amount spent on them

```
sales_state = df.groupby(['Product_ID'], as_index=False)
['Orders'].sum().sort_values(by='Orders', ascending=False).head(10)
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state,x = 'Product_ID',y= 'Orders',palette =
"colorblind")
```

```
<ipython-input-100-a381ff1dcb5e>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.barplot(data = sales_state,x = 'Product_ID',y= 'Orders',palette
= "colorblind")

<Axes: xlabel='Product_ID', ylabel='Orders'>
```



This graph shows that which top 10 Products has the maximum orders placed

The above graphs show the relationships between various columns of a csv file. countplot() is used to show the univariate analysis and barplot() is used to show bivariate analysis

```
# prompt: i want to conclude my analysis and generate recommendations
and key findings from the analysis

# Recommendations and Key Findings

# Key Findings:

# 1. **Female Customer Dominance:** Female customers constitute a
larger customer base and contribute more significantly to overall
sales compared to male customers.  Marketing efforts could be tailored
to further engage this demographic.

# 2. **Age Group 26-35 as Key Market Segment:** The 26-35 age group
demonstrates the highest purchasing power, indicating a prime target
```

```
audience for focused marketing campaigns and product development.

# 3. **Geographical Sales Concentration:**  Certain states (as
identified in the visualizations) show significantly higher order
volumes and sales revenue.  Further market research within these
regions could uncover opportunities for expansion and localized
marketing initiatives. Investigate reasons behind high sales in these
regions.

# 4. **Product Category Performance:**  Some product categories have
higher sales and order volumes than others.  Promotions or new product
development could be prioritized based on this analysis to potentially
boost overall sales. Identify top performing products within each
category.

# 5. **Impact of Marital Status:**  Analyze whether marital status
influences spending habits. This could lead to targeted promotions and
product recommendations.


# Recommendations:

# 1. **Targeted Marketing:** Design specific marketing campaigns
tailored towards female customers within the 26-35 age group, focusing
on high-performing product categories.

# 2. **Regional Focus:** Allocate more resources to the top-performing
states, while also investigating the potential for expansion into
underperforming regions. Localized advertising and promotions could be
particularly effective.

# 3. **Product Optimization:**  Invest in product development within
high-demand categories. Simultaneously, analyze underperforming
products to determine if improvements or discontinuation are
necessary.

# 4. **Customer Segmentation:**  Conduct further analysis to create
detailed customer segments based on demographics, purchasing behavior,
and product preferences. This will allow for more personalized
marketing and targeted product recommendations.  Consider using
clustering algorithms to segment customers.

# 5. **Promotional Strategies:** Develop promotional strategies that
incentivize purchases in lower-performing product categories or during
slower sales periods.

# 6. **Inventory Management:** Optimize inventory levels based on
sales patterns for the top-performing products and regions to avoid
stockouts or overstocking.
```