In [126…
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt    # Visualizing data
%matplotlib inline
```

In [128…
```
df=pd.read_csv("Diwali Sales Data.csv",encoding="unicode_escape")
```

In [130…
```
df.shape
```

Out[130… (11251, 15)

In [132…
```
df.head(5)
```

Out[132…

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	Stat
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtı
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Prades
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Prades
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnatak
4	1000588	Joni	P00057942	M	26-35	28	1	Gujara

In [134…
```
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
```

In [136…
```
#drop unrelated/blank columns
df.drop(["Status","unnamed1"],axis=1, inplace=True)
```

In [138…
```
#checking for null values
df.isna().sum()
```

Out[138…
```
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
```

In [140…
```
# drop null values
df.dropna(inplace=True)
```

In [142…
```
df.shape
```

Out[142…
```
(11239, 13)
```

In [144…
```
# change data type
df["Amount"] = df["Amount"].astype(int)
```

In [146…
```
df["Amount"].dtypes
```

Out[146…
```
dtype('int32')
```

In [148…
```
df.columns.tolist()
```

Out[148…
```
['User_ID',
 'Cust_name',
 'Product_ID',
 'Gender',
 'Age Group',
 'Age',
 'Marital_Status',
 'State',
 'Zone',
 'Occupation',
 'Product_Category',
 'Orders',
 'Amount']
```

In [150…
```
#rename column
df.rename(columns= {'Marital_Status':'Shaadi'})
```

Out[150…

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Shaadi	State
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat
...	...	...	...	...	...	...	...	...
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra

11239 rows × 13 columns

In [152…

```
# describe() method returns description of the data in the DataFrame (i.e. count
df.describe()
```

Out[152…

	User_ID	Age	Marital_Status	Orders	Amount
count	1.123900e+04	11239.000000	11239.000000	11239.000000	11239.000000
mean	1.003004e+06	35.410357	0.420055	2.489634	9453.610553
std	1.716039e+03	12.753866	0.493589	1.114967	5222.355168
min	1.000001e+06	12.000000	0.000000	1.000000	188.000000
25%	1.001492e+06	27.000000	0.000000	2.000000	5443.000000
50%	1.003064e+06	33.000000	0.000000	2.000000	8109.000000
75%	1.004426e+06	43.000000	1.000000	3.000000	12675.000000
max	1.006040e+06	92.000000	1.000000	4.000000	23952.000000

In [154…

```
# use describe() for specific columns
df[['Age','Orders','Amount']].describe()
```

Out[154…

	Age	Orders	Amount
count	11239.000000	11239.000000	11239.000000
mean	35.410357	2.489634	9453.610553
std	12.753866	1.114967	5222.355168
min	12.000000	1.000000	188.000000
25%	27.000000	2.000000	5443.000000
50%	33.000000	2.000000	8109.000000
75%	43.000000	3.000000	12675.000000
max	92.000000	4.000000	23952.000000

Exploratory Data Analysis

In [157…

```
df
```

Out[157…

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Mah
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar
3	1001425	Sudevi	P00237842	M	0-17	16	0	Ka
4	1000588	Joni	P00057942	M	26-35	28	1	
...	...	...	...	...	...	...	...	
11246	1000695	Manning	P00296942	M	18-25	19	1	Mah
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	
11248	1001209	Oshin	P00201342	F	36-45	40	0	
11249	1004023	Noonan	P00059442	M	36-45	37	0	Ka
11250	1002744	Brumley	P00281742	F	18-25	19	0	Mah

11239 rows × 13 columns

Gender

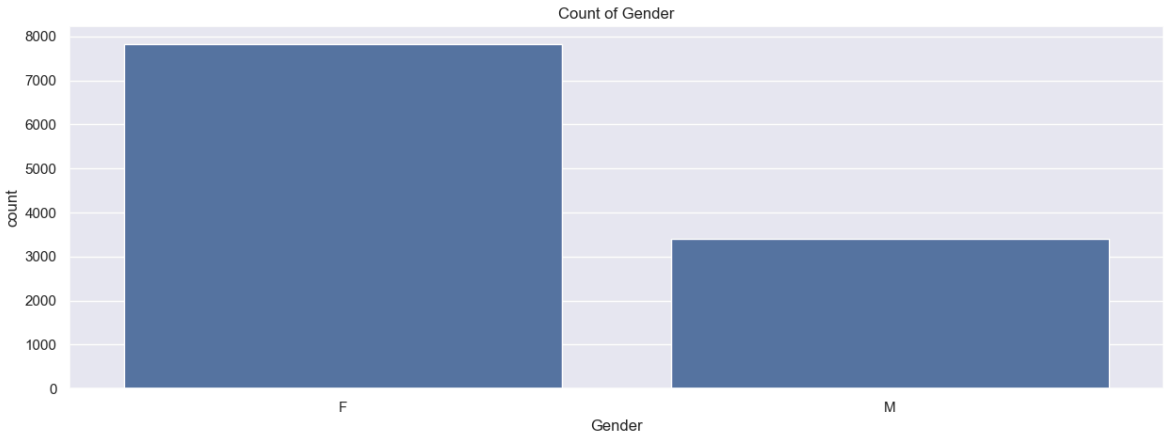
In [160…

```
df.columns
```

Out[160…

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

In [248…
```
sns.countplot(data=df, x='Gender')
plt.title('Count of Gender')
plt.show()
```

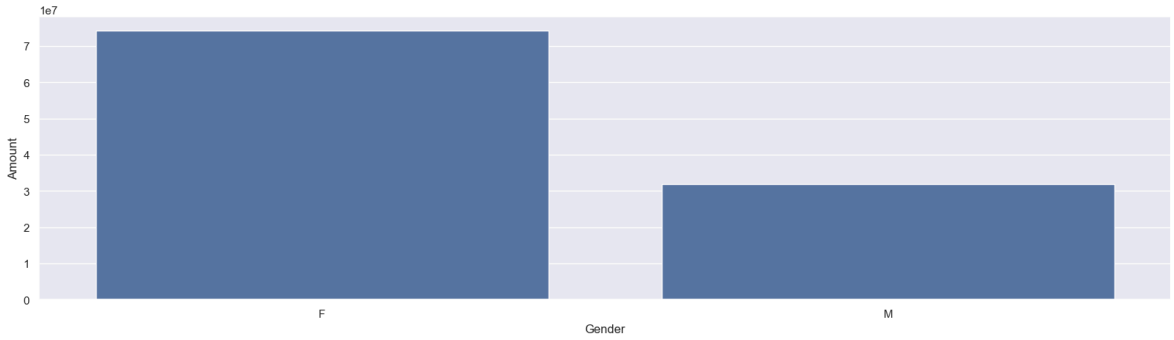


In [163…
```
df.groupby(['Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount',
```

Out[163…

	Gender	Amount
0	F	74335853
1	M	31913276

In [166…
```
sales_gen = df.groupby(["Gender"], as_index=False)['Amount'].sum().sort_values(b

sns.barplot(x = 'Gender', y= 'Amount' ,data = sales_gen)
plt.show()
```



From above graphs we can see that most of the buyers are females and even the purchesing power of females are greater than men

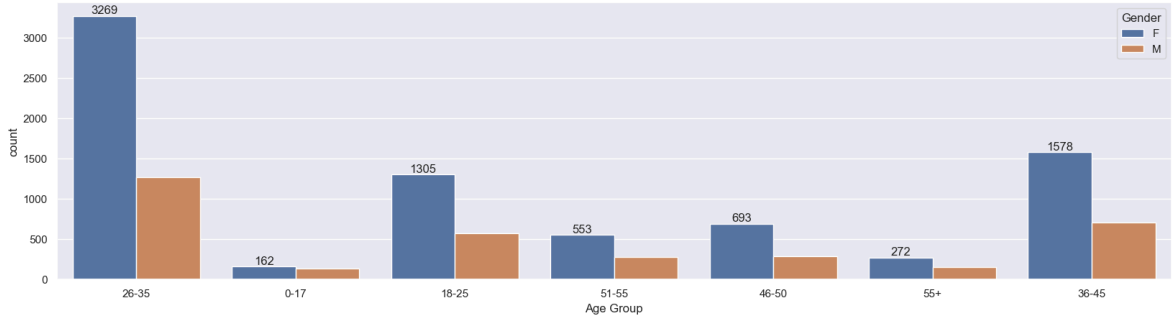
Age

In [170…
```
df.columns
```

Out[170…
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

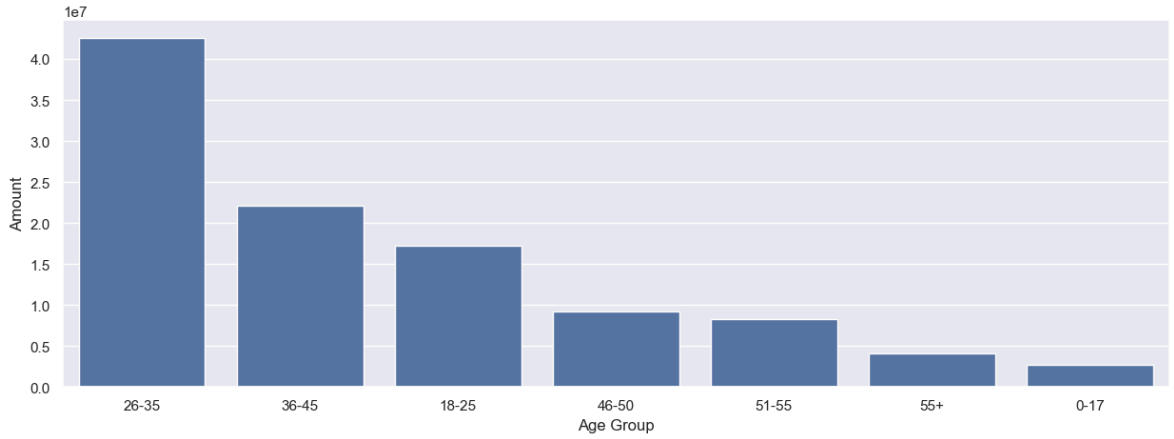
In [172…
```
ax = sns.countplot(data = df, x= 'Age Group', hue = 'Gender')

for bars in ax.containers:
```

```
    ax.bar_label(bars)
    plt.show()
```



In [250…

```
# Total Amount vs Age Group
sales_age = df.groupby(['Age Group'], as_index=False)['Amount'].sum().sort_value

sns.barplot(data = sales_age, x = 'Age Group', y='Amount',)
plt.show()
```

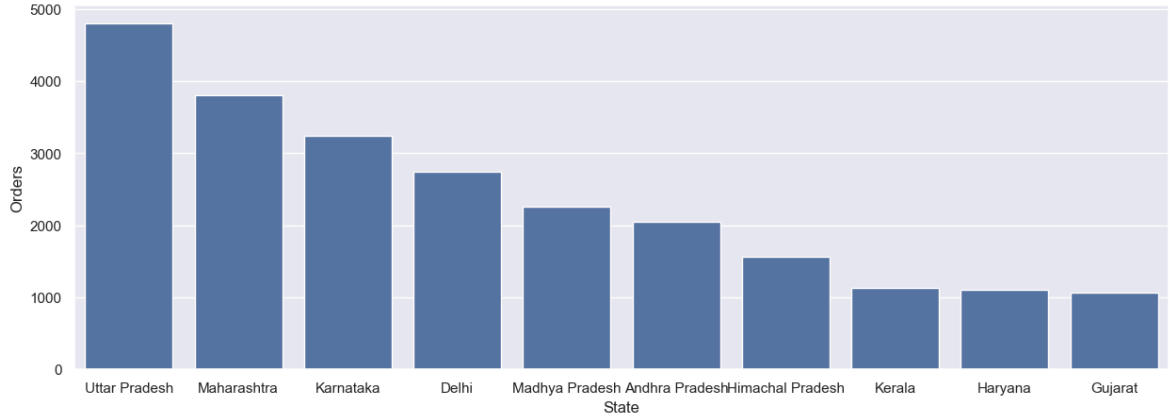


From above graphs we can see that most of the buyers are of age group between 26-35 yrs female

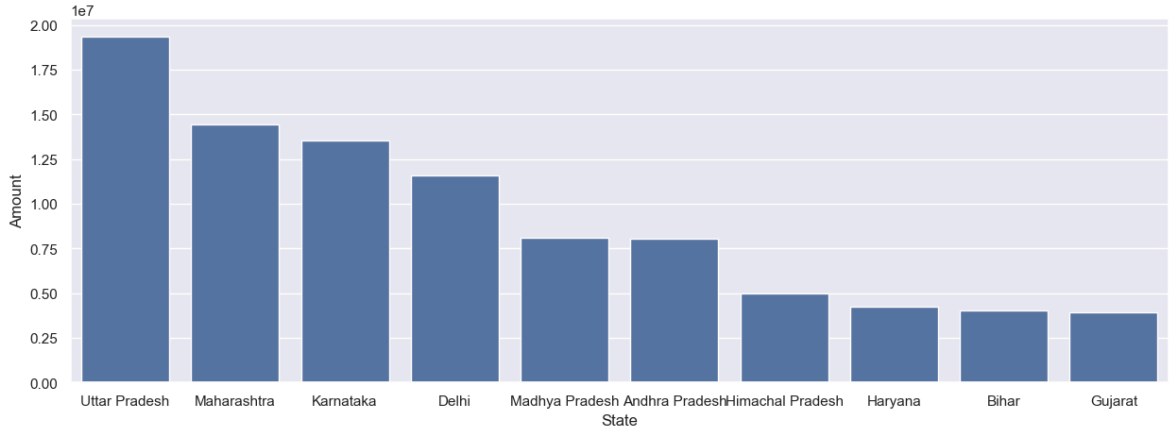
State

In [246…

```
# total number of orders from top 10 states
sales_state = df.groupby(['State'], as_index=False)['Orders'].sum().sort_values(

sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data = sales_state, x = 'State',y= 'Orders')
plt.show()
```



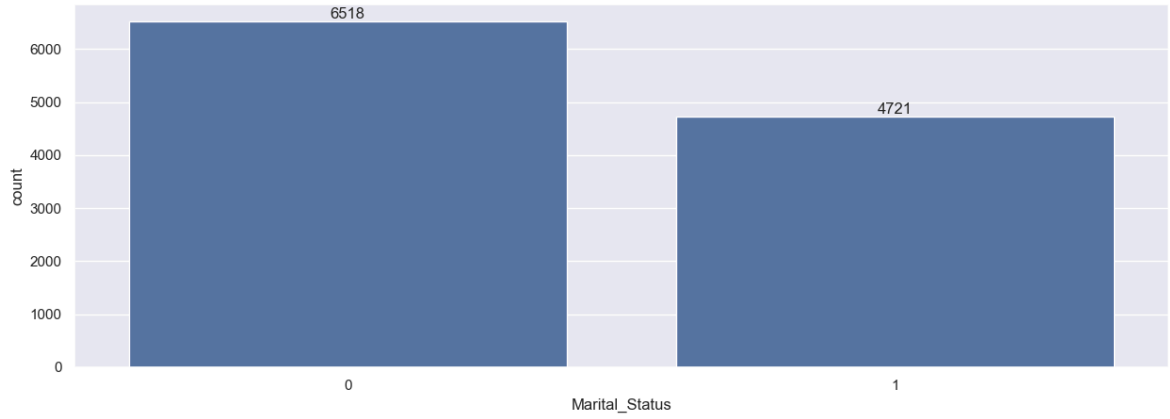
In [179…
```
sales_state = df.groupby(['State'], as_index=False)['Amount'].sum().sort_values(

sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data = sales_state, x = 'State',y= 'Amount')
plt.show()
```



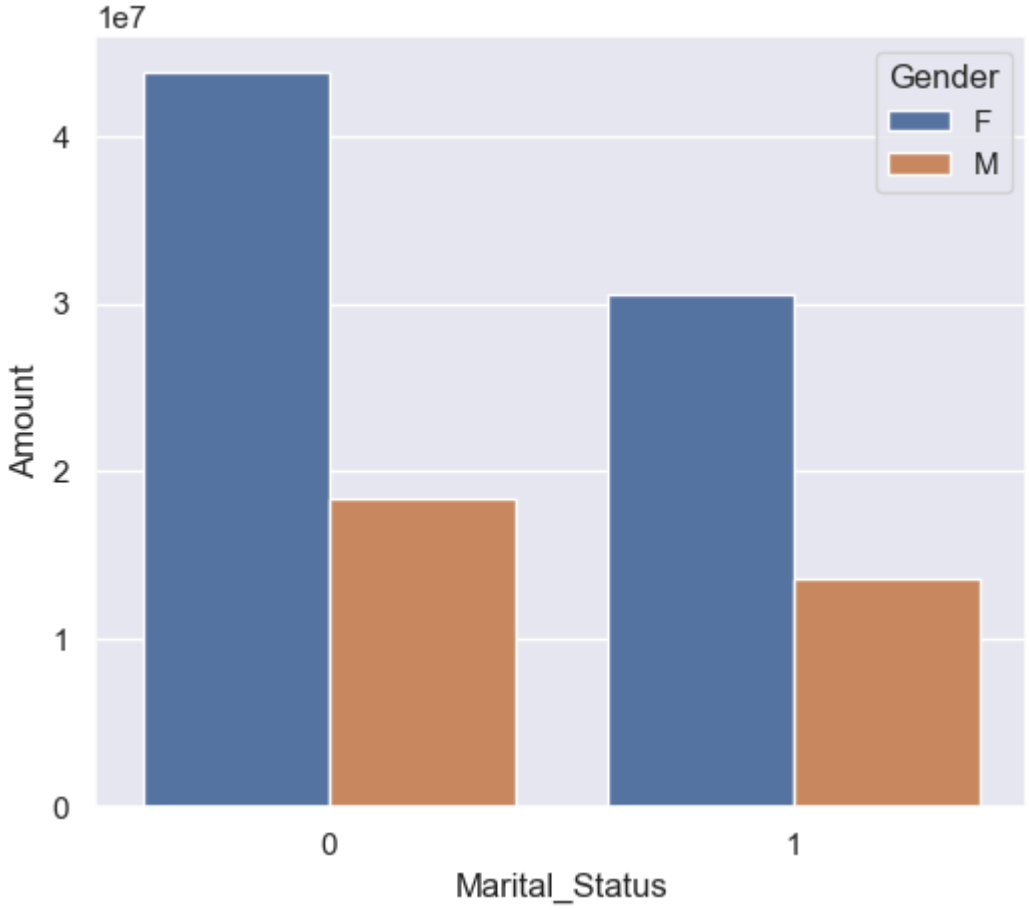
From above graphs we can see that unexpectedly most of the orders are from Uttar Pradesh, Maharashtra and Karnataka respectively but total sales/amount is from UP, Karnataka and then Maharashtra

Marital Status

In [184…
```
ax = sns.countplot(data = df, x = 'Marital_Status')
sns.set(rc={'figure.figsize':(7,5)})
for bars in ax.containers:
    ax.bar_label(bars)
plt.show()
```



In [186…
```
sales_state = df.groupby(['Marital_Status','Gender'], as_index=False)['Amount'].

sns.set(rc={'figure.figsize':(6,5)})
sns.barplot(data = sales_state, x = 'Marital_Status',y= 'Amount', hue='Gender')
plt.show()
```

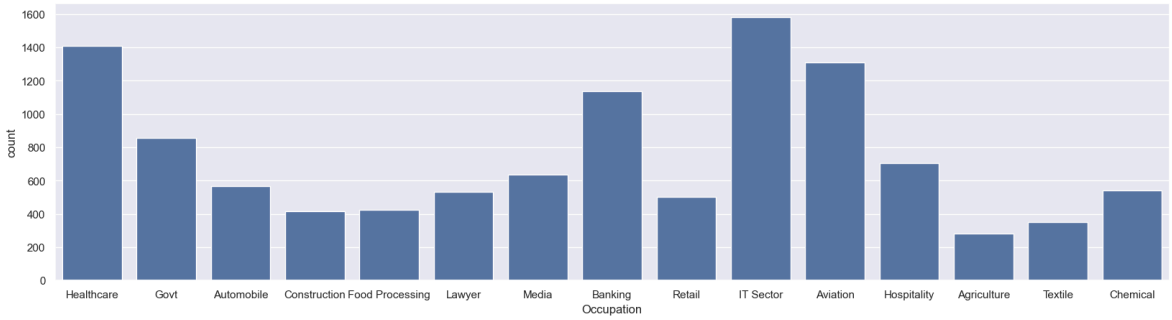


From above graphs we can see that most of the buyers are married (women) and they have high purchasing power

Occupation

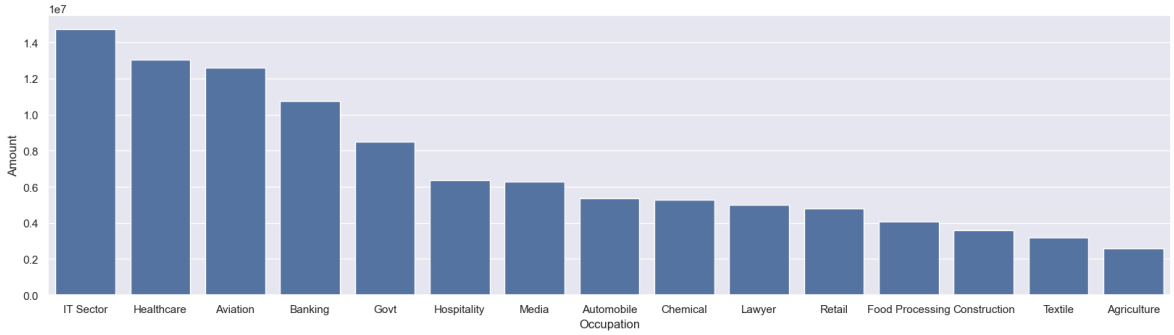
In [190…

```
sns.set(rc={'figure.figsize':(20,5)})
sns.countplot(data = df, x = 'Occupation')

for bars in ax.containers:
    ax.bar_label(bars)
plt.show()
```



In [202…

```
sales_state=df.groupby(['Occupation'], as_index=False)['Amount'].sum().sort_valu

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data=sales_state, x = 'Occupation',y='Amount')
plt.show()
```



In [ ]:

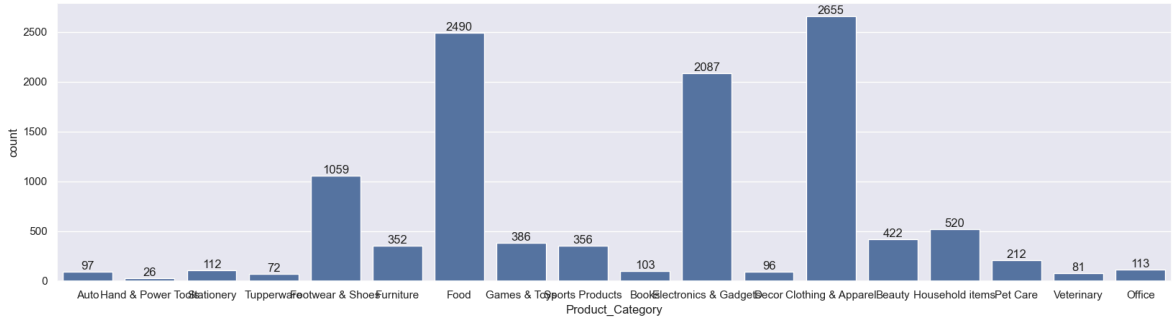
From above graphs we can see that most of the buyers are working in IT. Available and Healthcare sector

In [ ]:

Product Category

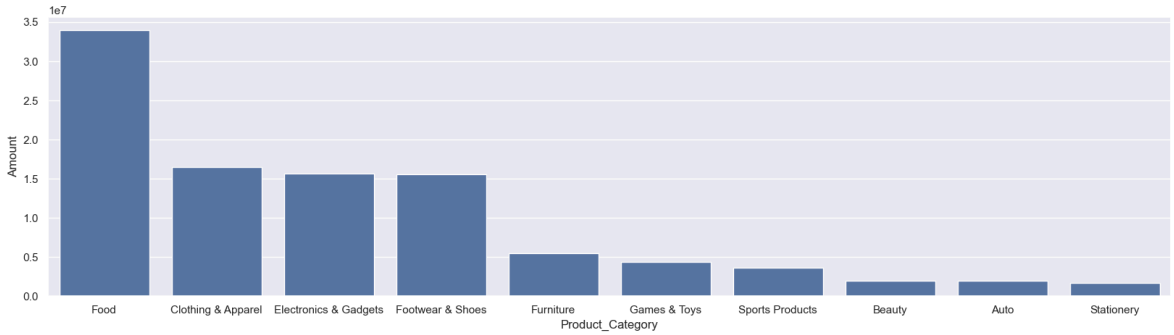
In [209…

```
sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = df, x = 'Product_Category')

for bars in ax.containers:
    ax.bar_label(bars)
plt.show()
```



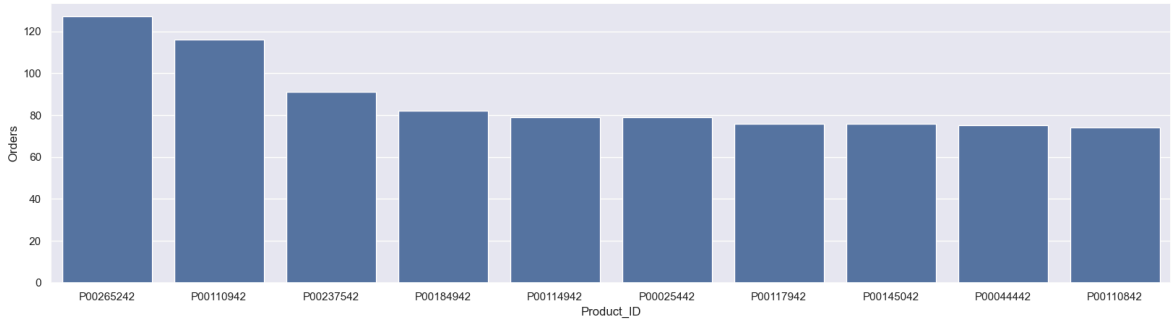
In [218…

```
sales_state = df.groupby(['Product_Category'], as_index=False)['Amount'].sum().s

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data=sales_state, x='Product_Category', y= 'Amount')
plt.show()
```



From above graphs we can see that most of the sold products are from Food, Clothing and Electronics category

In [229…
```
sales_state=df.groupby(['Product_ID'],as_index=False)['Orders'].sum().sort_value

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data=sales_state, x = 'Product_ID', y = 'Orders')
plt.show()
```



Conclusion:

Married women age group 26-35 yrs from UP,Maharastra and Karnataka working in IT, Healthcare and Aviation are more likely to buy products from Food clothing and Electronics category

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: