In [ ]:

```
# import python libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt # visualizing data
import seaborn as sns
```

In [ ]:

```
df=pd.read_csv(r"C:\Users\DELL\Downloads\Diwali Sales Data.csv", encoding='unicode_
df.head()
```

Out[ ]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	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

In [ ]:

```
df.shape
```

Out[ ]:

```
(11251, 15)
```

In [ ]:

```
df.columns
```

Out[ ]:

```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount', 'Status', 'unnamed1'],
      dtype='object')
```

In [ ]:

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

In [ ]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11251 entries, 0 to 11250
Data columns (total 13 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
dtypes: float64(1), int64(4), object(8)
memory usage: 1.1+ MB
```

In [ ]:
```
df.isnull().sum()
```

Out[ ]:

```
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 [ ]:

```
df.dropna(inplace=True)
```

In [ ]:

```
df["Amount"]=df["Amount"].astype("int")
```

In [ ]:

```
df["Amount"].dtypes
```

Out[ ]:

```
dtype('int32')
```

In [ ]:

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

Out[ ]:

	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 [ ]:

```
df.describe()
```

Out[ ]:

	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 [ ]:

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

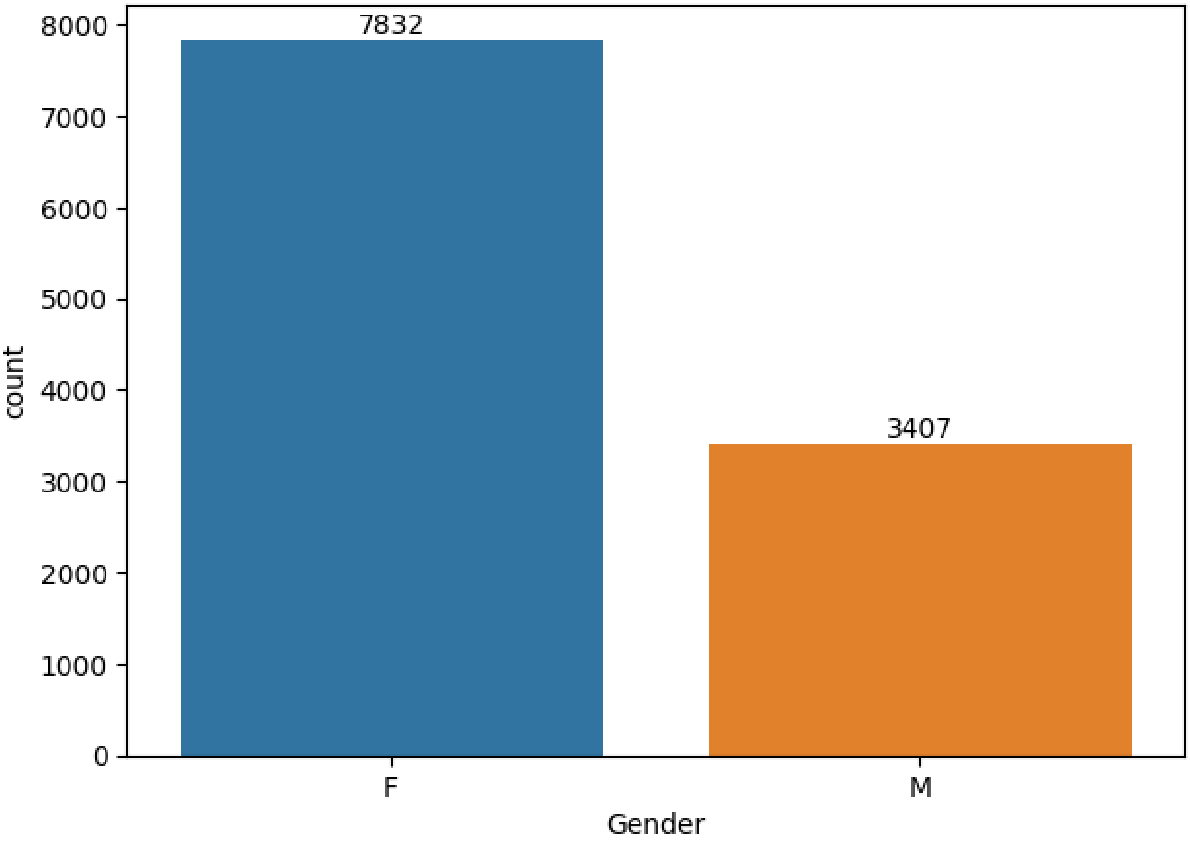
Out[ ]:

	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 [ ]:

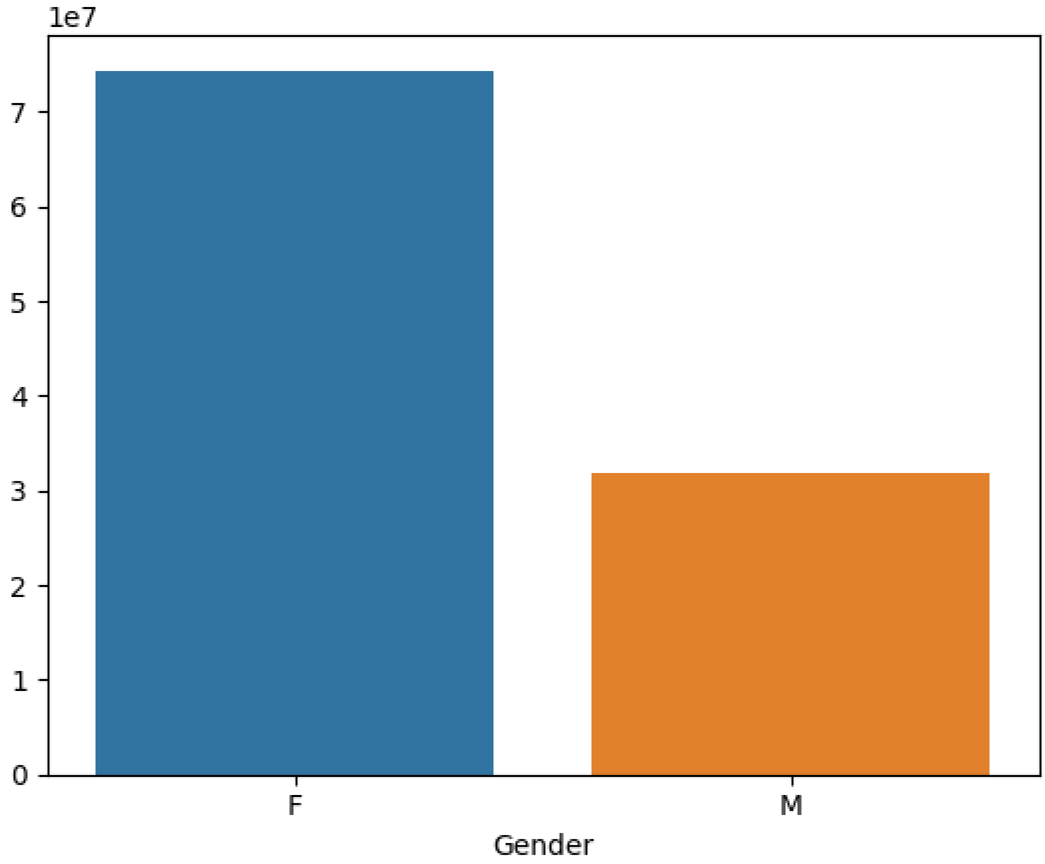
```
plt.figure(figsize=(7,5))
ax = sns.countplot(x = 'Gender',data = df)

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



```
In [ ]: new=df.groupby("Gender")["Amount"].sum()
        sns.barplot(x=new.index, y=new.values)
```

```
Out[ ]: <Axes: xlabel='Gender'>
```

Loading [MathJax]/extensions/Safe.js

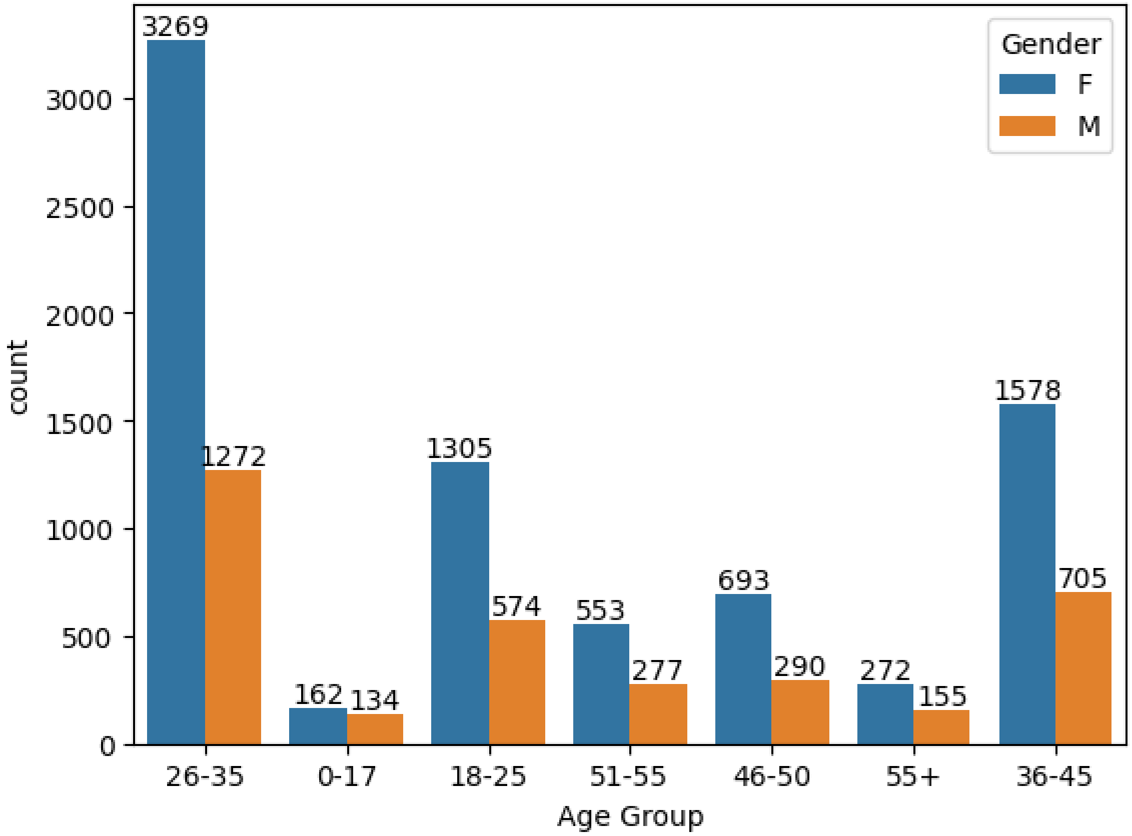


From above graphs we can see that most of the buyers are females and even the purchasing power of females are greater than men

In [ ]:

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

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

Loading [MathJax]/extensions/Safe.js



In [ ]:

```
age_group=df.groupby("Age Group")["Amount"].sum().reset_index()
age_group.sort_values(by="Amount",ascending=False, inplace=True)
age_group
```

Out[ ]:

	Age Group	Amount
2	26-35	42613442
3	36-45	22144994
1	18-25	17240732
4	46-50	9207844
5	51-55	8261477
6	55+	4080987
0	0-17	2699653

```
<Figure size 1300x500 with 0 Axes>
```

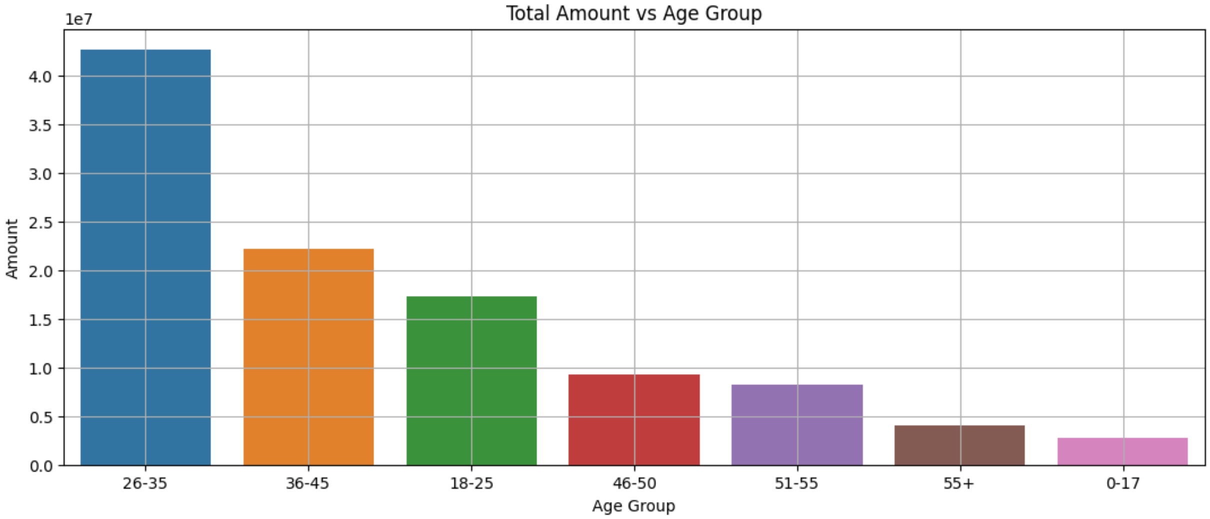
In [ ]:

```
plt.figure(figsize=(13,5))
sns.barplot(x ="Age Group", y="Amount" ,data = age_group)
plt.grid()
plt.title("Total Amount vs Age Group")
```

Out[ ]:

```
Text(0.5, 1.0, 'Total Amount vs Age Group')
```

Loading [MathJax]/extensions/Safe.js



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

In [ ]:

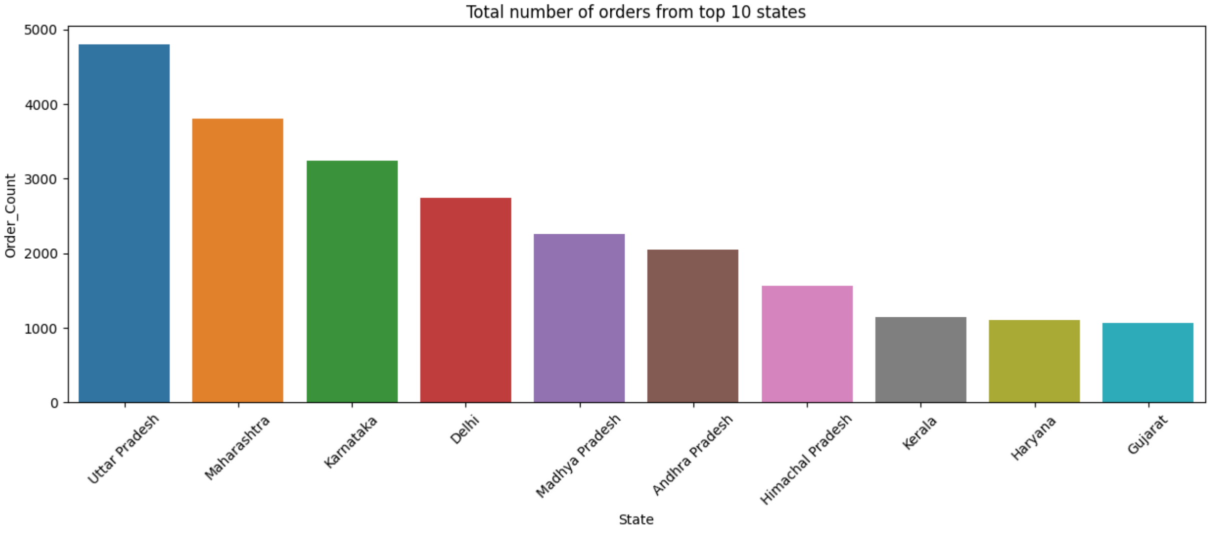
```
#State
state=df.groupby("State")["Orders"].sum().reset_index(name="Order_Count")
state.sort_values(by="Order_Count",ascending=False, inplace=True)
state=state.head(10)
```

In [ ]:

```
plt.figure(figsize=(15,5))
sns.barplot(data = state, x = 'State',y= 'Order_Count')
plt.xticks(rotation=45)
plt.title("Total number of orders from top 10 states")
```

Out[ ]:

```
Text(0.5, 1.0, 'Total number of orders from top 10 states')
```



In [ ]:

```
# total amount/sales from top 10 states

sales_state=df.groupby("State")["Amount"].sum().reset_index(name="total_amount")
sales_state.sort_values(by="total_amount",ascending=False, inplace=True)
_    es_state.head(10)
sales_state
```

Loading [MathJax]/extensions/Safe.js

Out[ ]:

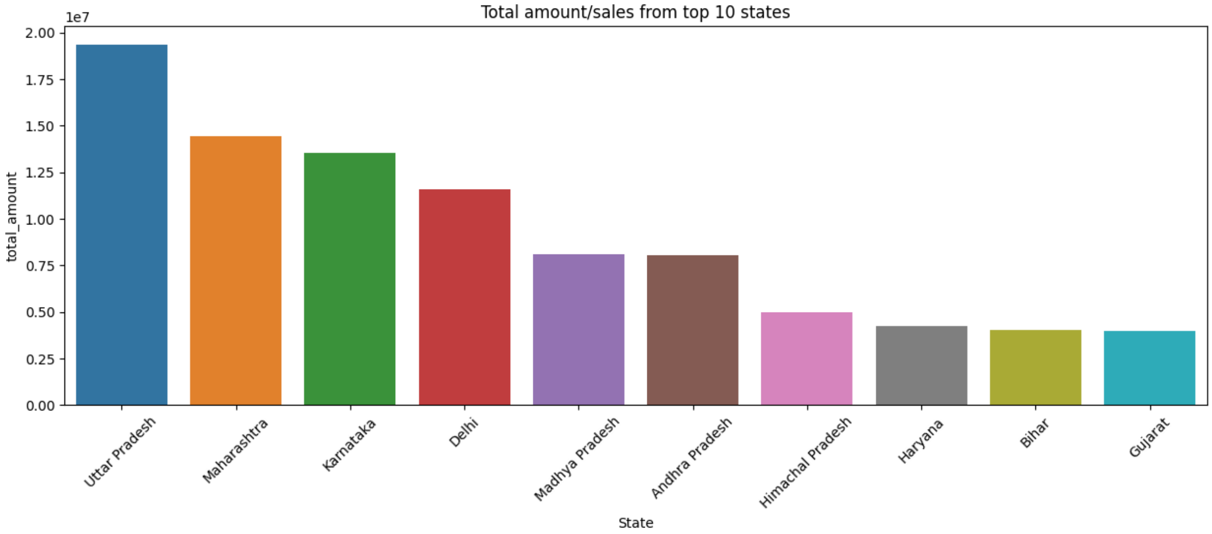
	State	total_amount
14	Uttar Pradesh	19374968
10	Maharashtra	14427543
7	Karnataka	13523540
2	Delhi	11603818
9	Madhya Pradesh	8101142
0	Andhra Pradesh	8037146
5	Himachal Pradesh	4963368
4	Haryana	4220175
1	Bihar	4022757
3	Gujarat	3946082

In [ ]:

```
plt.figure(figsize=(15,5))
sns.barplot(data = sales_state, x = 'State',y= 'total_amount')
plt.xticks(rotation=45)
plt.title("Total amount/sales from top 10 states")
```

Out[ ]:

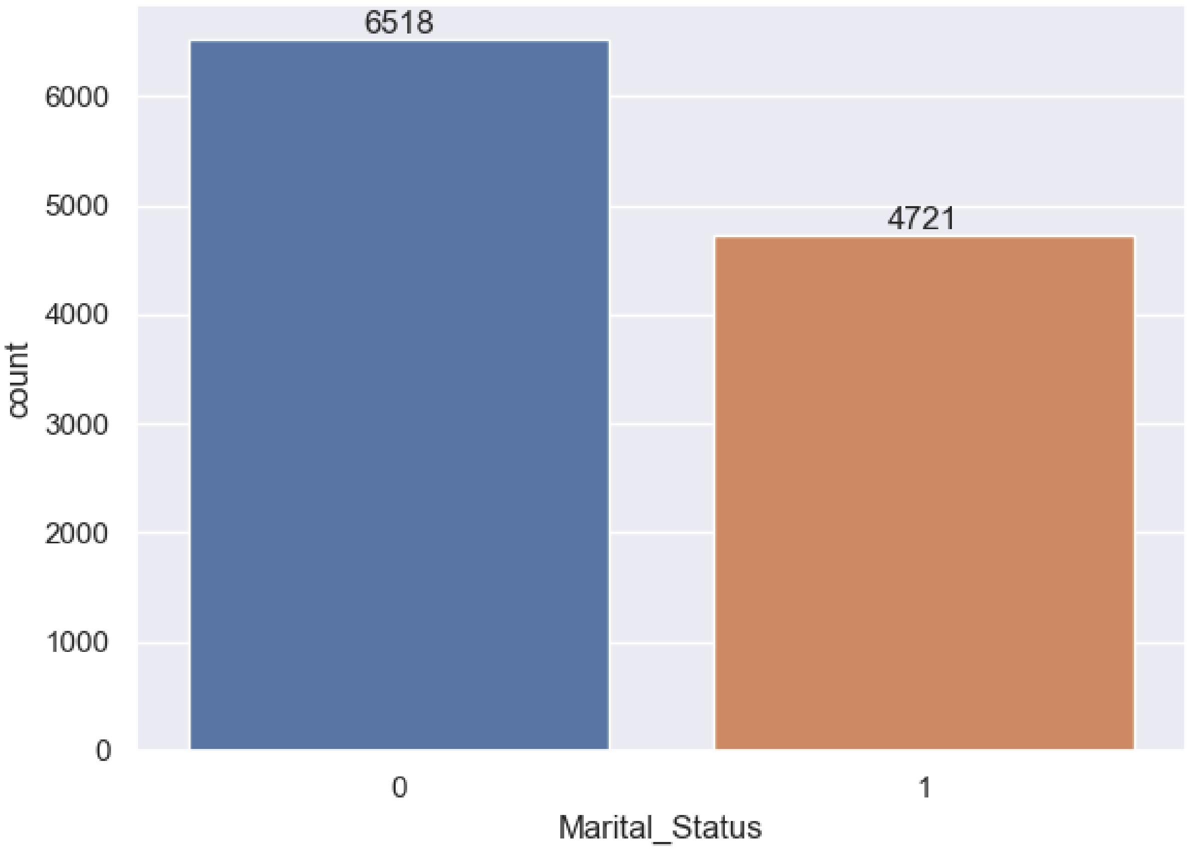
```
Text(0.5, 1.0, 'Total amount/sales from top 10 states')
```



In [ ]:

```
#Marital Status

ax = sns.countplot(data = df, x = 'Marital_Status')
for bars in ax.containers:
    ax.bar_label(bars)
```

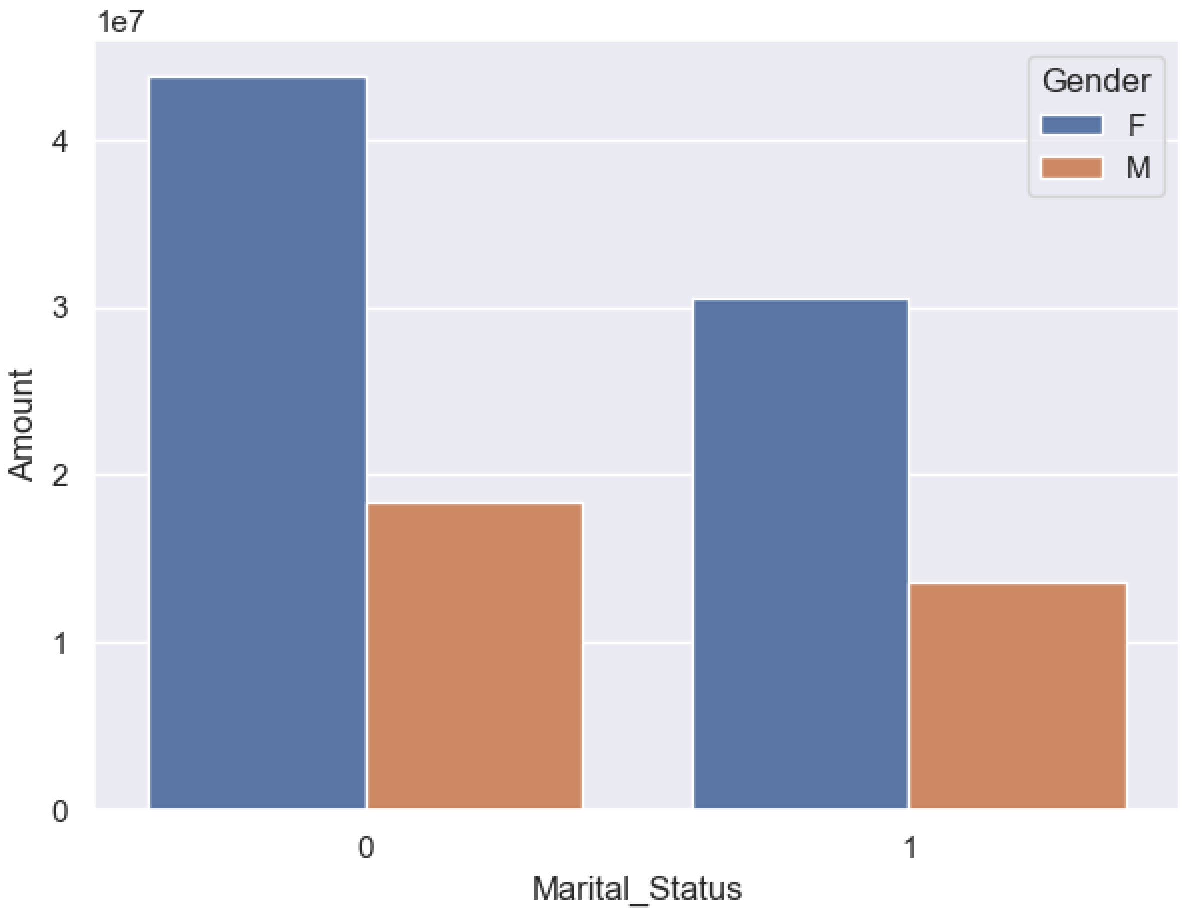
Loading [MathJax]/extensions/Safe.js



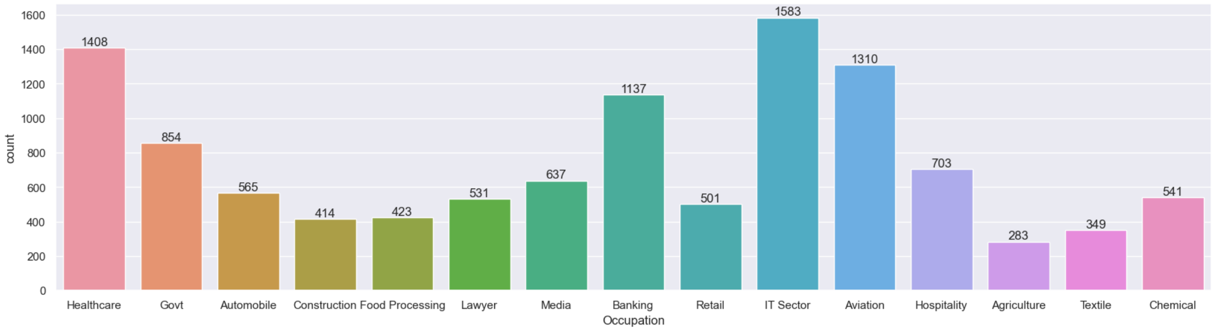
```
In [ ]: sales_state = df.groupby(['Marital_Status', 'Gender'])['Amount'].sum().reset_index(
        sales_state.sort_values(by='Amount', ascending=False)
        sns.barplot(data = sales_state, x = 'Marital_Status',y= 'Amount', hue='Gender')
```

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

Loading [MathJax]/extensions/Safe.js



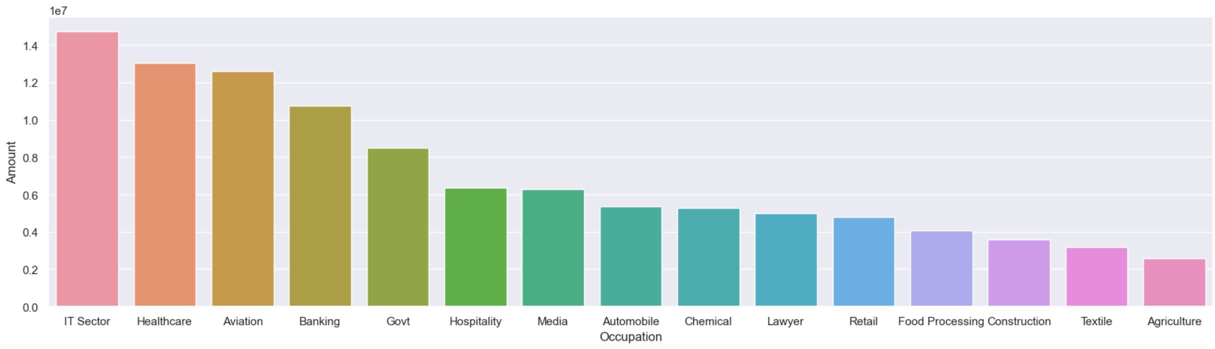
```
In [ ]: ax = sns.countplot(data = df, x = 'Occupation')

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



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

        sns.barplot(data = sales_state, x = 'Occupation',y= 'Amount')
```

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

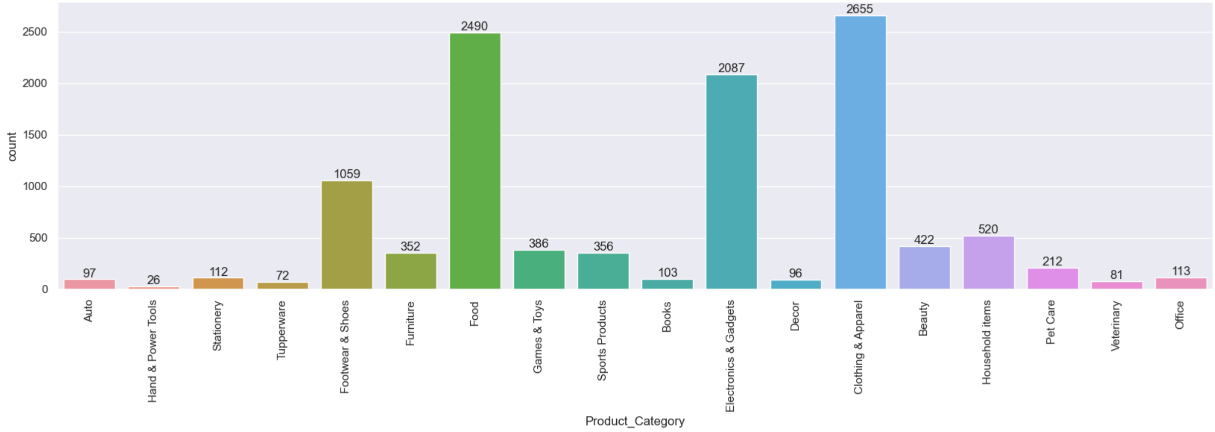
Loading [MathJax]/extensions/Safe.js



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

In [ ]:

```
#Product Category

ax = sns.countplot(data = df, x = 'Product_Category')
plt.xticks(rotation=90)
for bars in ax.containers:
    ax.bar_label(bars)
```

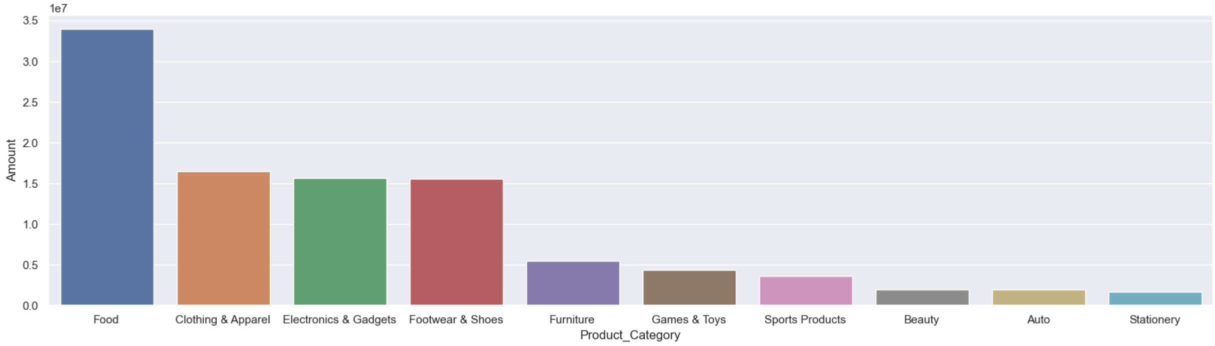


In [ ]:

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

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

Out[ ]:

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

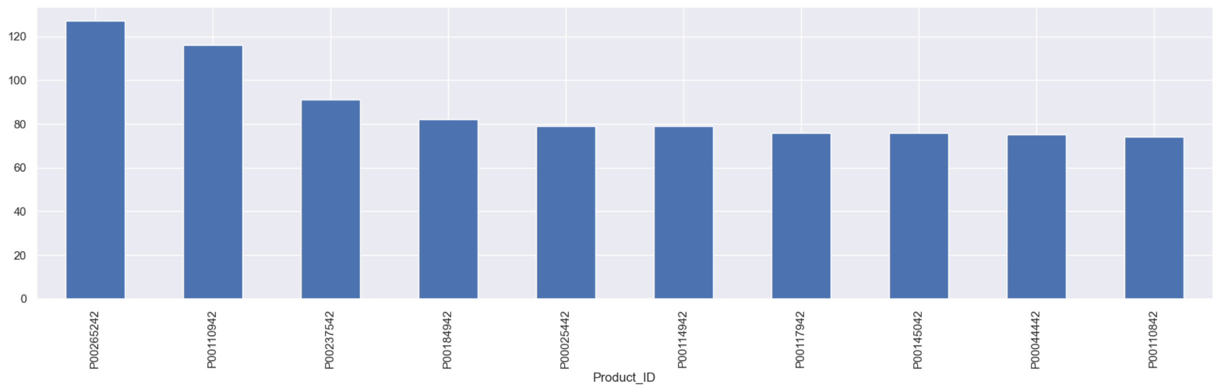


Loading [MathJax]/extensions/Safe.js

In [ ]:

```
df.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascending=False).
```

```
Out[ ]: <Axes: xlabel='Product_ID'>
```



Loading [MathJax]/extensions/Safe.js