diwali sales analysis

In [ ]:
```
#install module /library      !pip install (library name)
```

import basic libraries

In [4]:
```
import numpy as np #helps for working on arrays and mathematical use
import pandas as pd #helps to work on dataframe means tables
import matplotlib.pyplot as plt #visualizing data
%matplotlib inline
import seaborn as sns
```

In [7]:
```
#import data

df =pd.read_csv(r"D:\DIWALI SALES PYTHON\Diwali Sales Data (1).csv",encoding = 'uni
df
```

Out[7]:

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

11251 rows × 15 columns

In [8]:
```
df.shape # for getting row and columns
```

Out[8]:
```
(11251, 15)
```

In [9]:
```
df.head(10) #top 10 rows
```

Out[9]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western
5	1000588	Joni	P00057942	M	26-35	28	1	Himachal Pradesh	Northern
6	1001132	Balk	P00018042	F	18-25	25	1	Uttar Pradesh	Central
7	1002092	Shivangi	P00273442	F	55+	61	0	Maharashtra	Western
8	1003224	Kushal	P00205642	M	26-35	35	0	Uttar Pradesh	Central
9	1003650	Ginny	P00031142	F	26-35	26	1	Andhra Pradesh	Southern

In [10]:
```
df.info() # information about the dataframe and details about it
```

```
<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 [11]:
```
#drop unrelated/blank columns
df.drop(['Status','unnamed1'],axis =1,inplace =True)
```

In [12]:
```
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 [13]:
```
pd.isnull(df)  #to check null values
```

Out[13]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occup
0	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	
11246	False	False	False	False	False	False	False	False	False	
11247	False	False	False	False	False	False	False	False	False	
11248	False	False	False	False	False	False	False	False	False	
11249	False	False	False	False	False	False	False	False	False	
11250	False	False	False	False	False	False	False	False	False	

11251 rows × 13 columns

In [14]:
```
#check for null values
pd.isnull(df).sum()
```

Out[14]:
```
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 [15]:
```
df.shape
```

Out[15]:
```
(11251, 13)
```

In [16]:
```
#drop null values
df.dropna(inplace = True)
```

In [17]:
```
df.shape
```

Out[17]:
```
(11239, 13)
```

In [18]:
```
#intialise list of lists
data_test =[['madhav',11],['Gopi',15],['Keshav',],['Lalita',16]]

#create the pandas dataframe using list
df_test = pd.DataFrame(data_test,columns = ['Name','Age'])
df_test
```

Out[18]:

	Name	Age
0	madhav	11.0
1	Gopi	15.0
2	Keshav	NaN
3	Lalita	16.0

In [20]:
```
df_test.dropna(inplace = True)
```

In [19]:
```
df_test
```

Out[19]:

	Name	Age
0	madhav	11.0
1	Gopi	15.0
2	Keshav	NaN
3	Lalita	16.0

both are same thing

df_test.dropna(inplace = True) df_test = df_test.dropna()

In [21]:
```
#change data type
df['Amount'] = df['Amount'].astype('int')
```

In [22]:
```
df['Amount'].dtypes
```

Out[22]:
```
dtype('int32')
```

In [23]:
```
df.columns
```

Out[23]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [24]:
```
#rename column
df.rename(columns = {'Marital_Status':'Shaadi'})
```

Out[24]:

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

11239 rows × 13 columns

In [25]:
```
#describe () method returns description of the data in the dataframe(i.e count,mean
df.describe()
```

Out[25]:

	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 [26]:
```
#use describe() for specific columns
df[['Age','Orders','Amount']].describe()
```

Out[26]:

	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

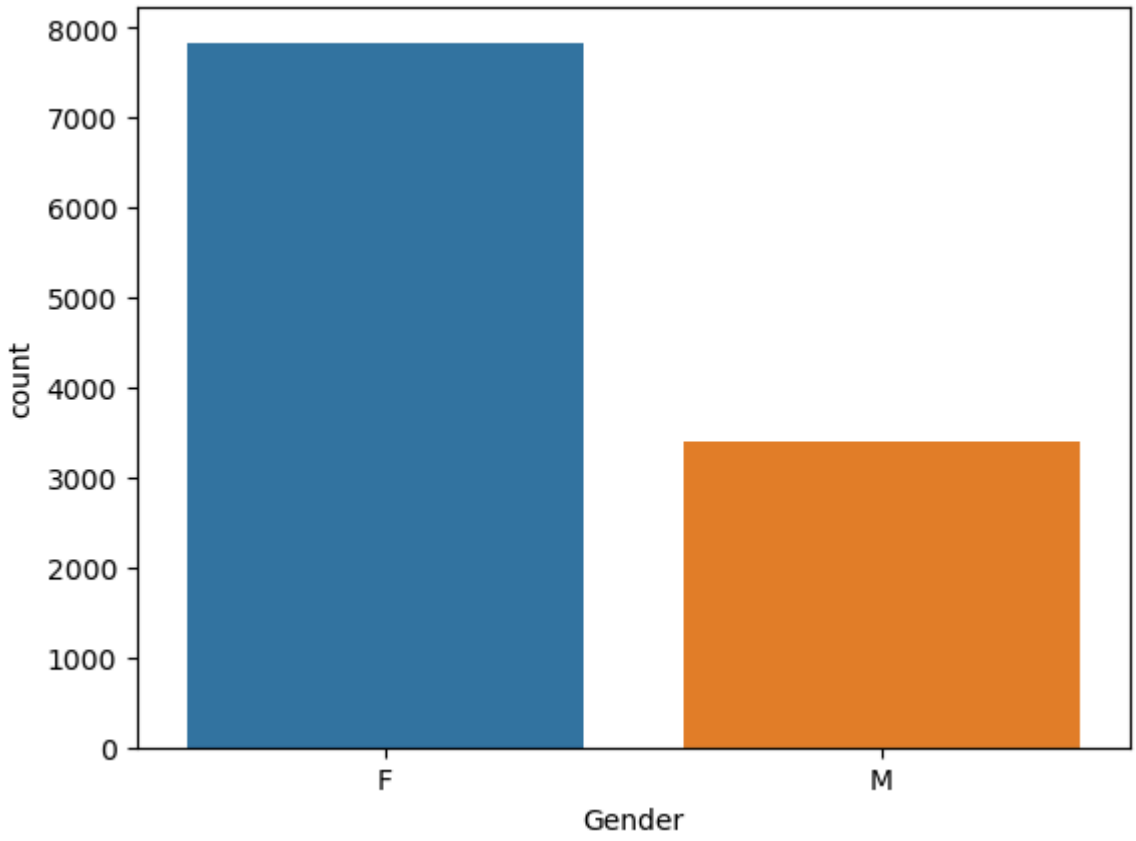
Exploratory data Analysis

Gender

In [27]:
```
df.columns
```

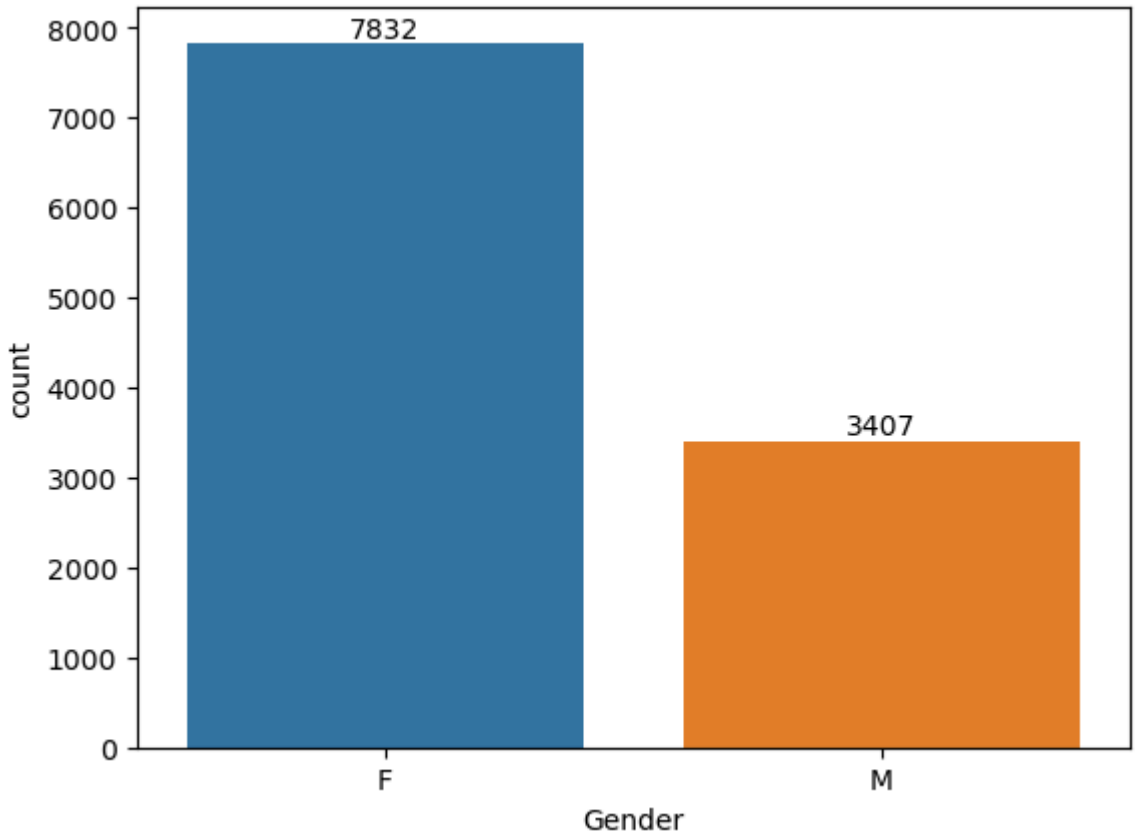
Out[27]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [28]:
```
ax = sns.countplot(x = 'Gender',data = df)
```



In [30]:
```
ax = sns.countplot(x ='Gender',data = df)

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



In [32]:
```
df.groupby(['Gender'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascer
```

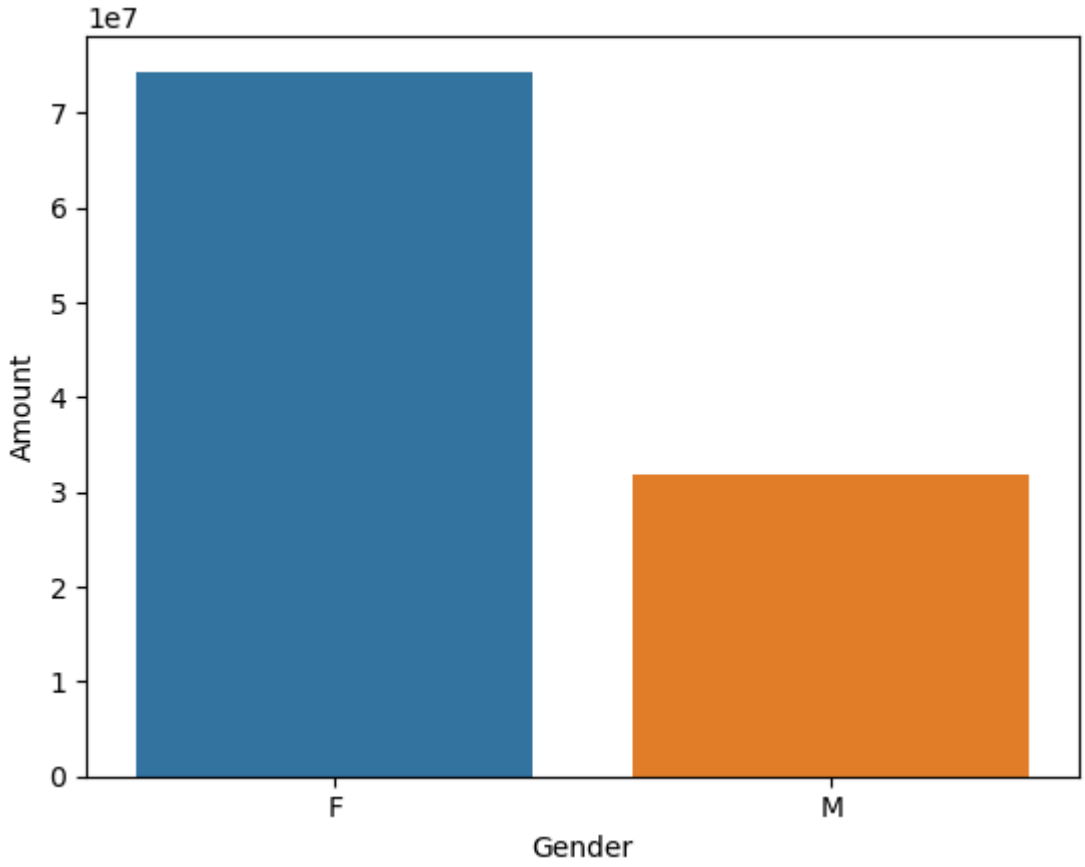
Out[32]:

	Gender	Amount
0	F	74335853
1	M	31913276

In [33]:
```
sales_gen = df.groupby(['Gender'],as_index=False)['Amount'].sum().sort_values(by='A

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

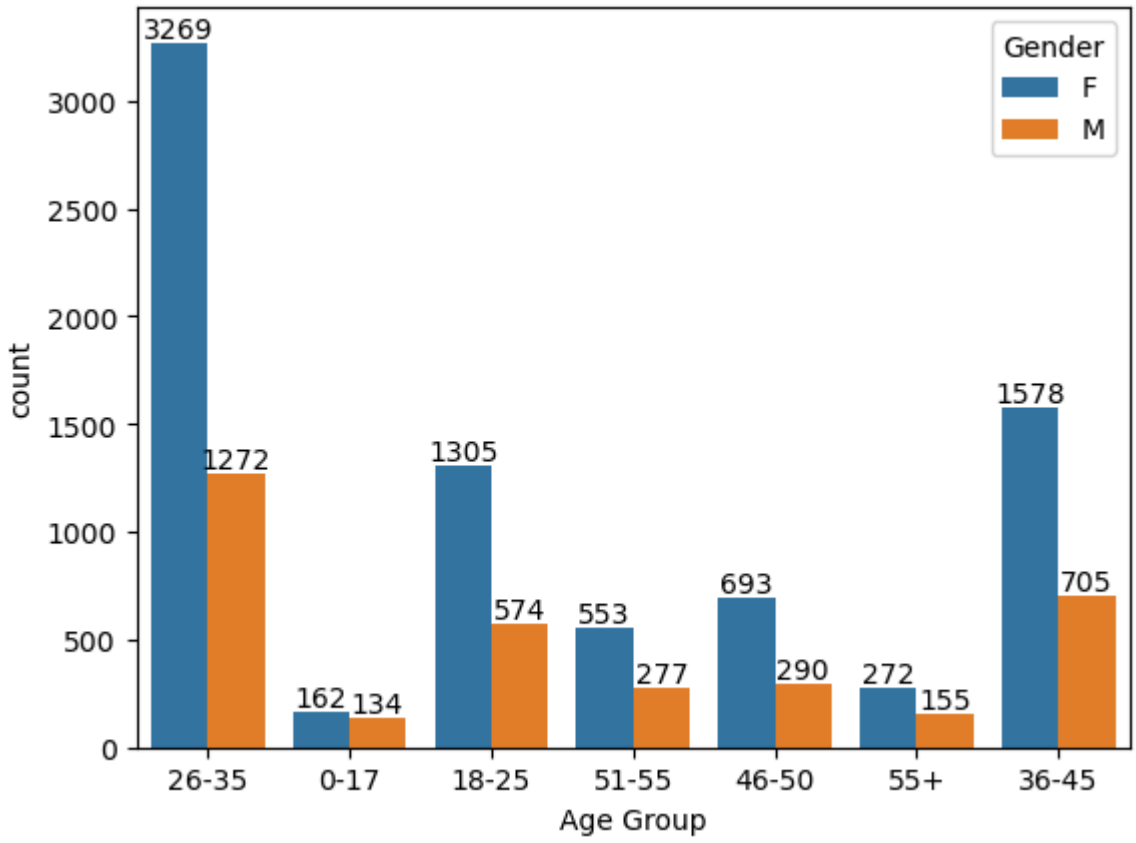
Out[33]:
```
<Axes: xlabel='Gender', ylabel='Amount'>
```



From above graphs,we can see that mose of the buyers are female and even the purchasing power of female are greater than male

Age

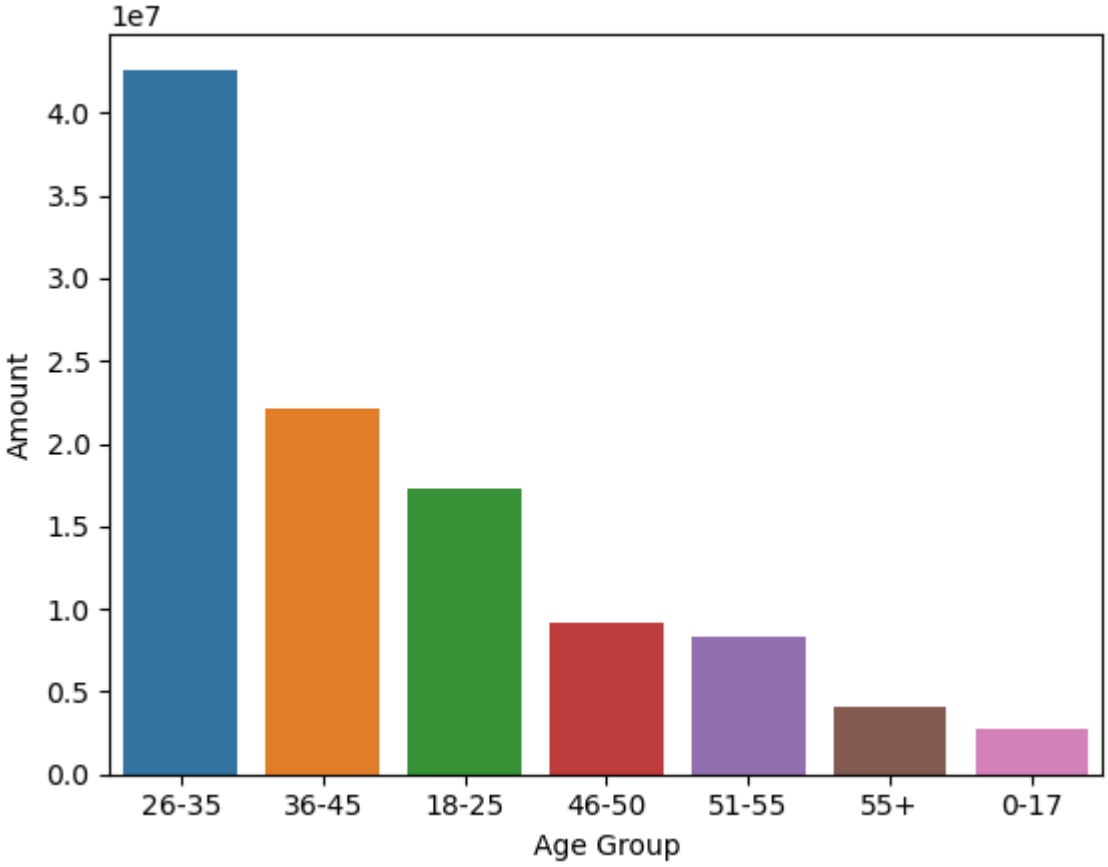
In [34]:
```
ax = sns.countplot(data = df ,x ='Age Group',hue = 'Gender')
for bars in ax.containers :
    ax.bar_label(bars)
```



In [35]:
```
#Totl Amount vs Age Group
sales_age = df.groupby(['Age Group'],as_index=False)['Amount'].sum().sort_values(by

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

Out[35]:
```
<Axes: xlabel='Age Group', ylabel='Amount'>
```



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

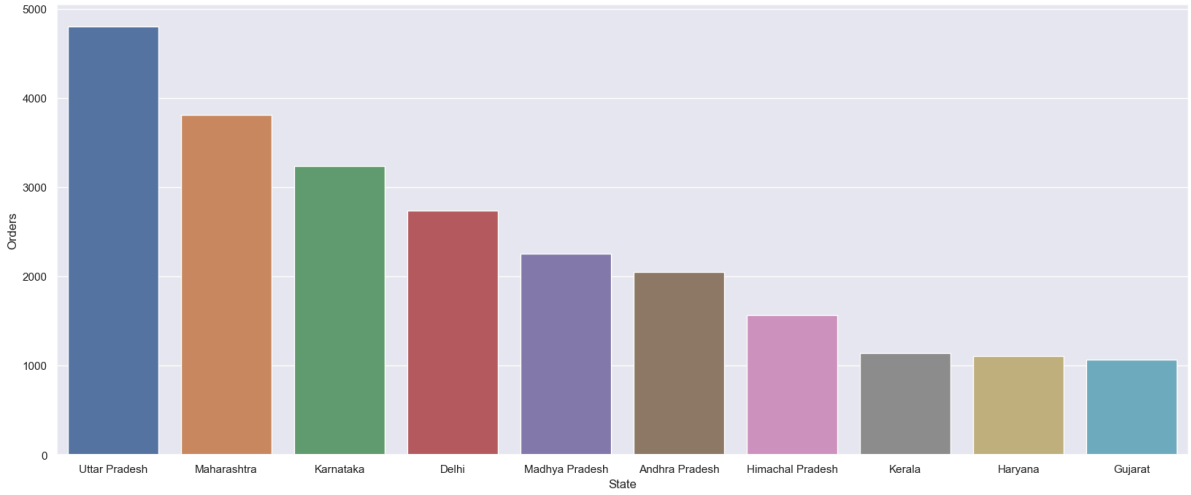
State

In [55]:
```
# Group by State, sum the Orders, sort and take the top 10
sales_state = df.groupby(['State'], as_index=False)['Orders'].sum().sort_values(by=

# Set the figure size
sns.set(rc={'figure.figsize':(20,8)})

# Plot the barplot
sns.barplot(data=sales_state, x='State', y='Orders')
```

Out[55]:
```
<Axes: xlabel='State', ylabel='Orders'>
```

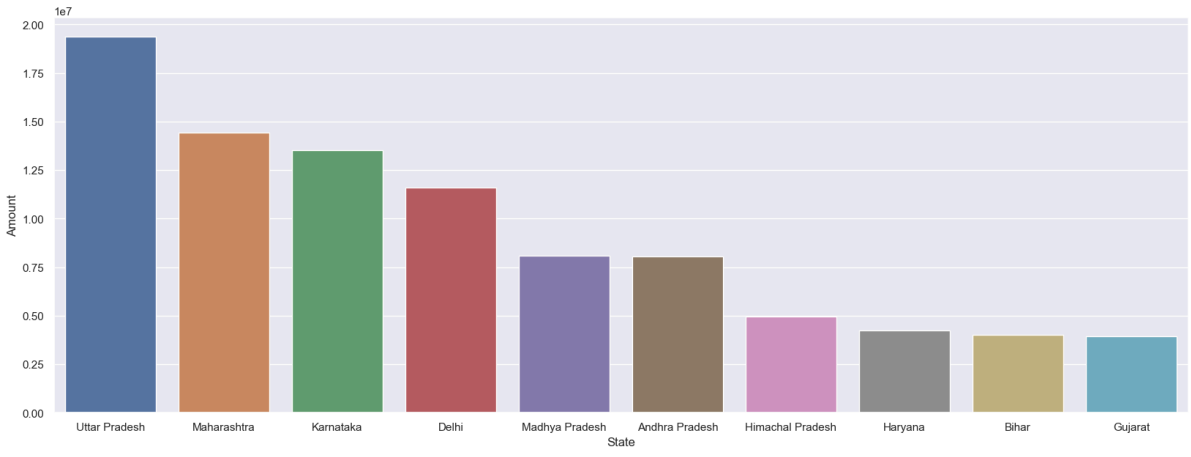


In [54]:
```
#total amount/sales from top 10 states
sales_state = df.groupby(['State'], as_index=False)['Amount'].sum().sort_values(by=

# Set the figure size
sns.set(rc={'figure.figsize':(20,7)})

# Plot the barplot
sns.barplot(data=sales_state, x='State', y='Amount')
```

Out[54]:
```
<Axes: xlabel='State', ylabel='Amount'>
```



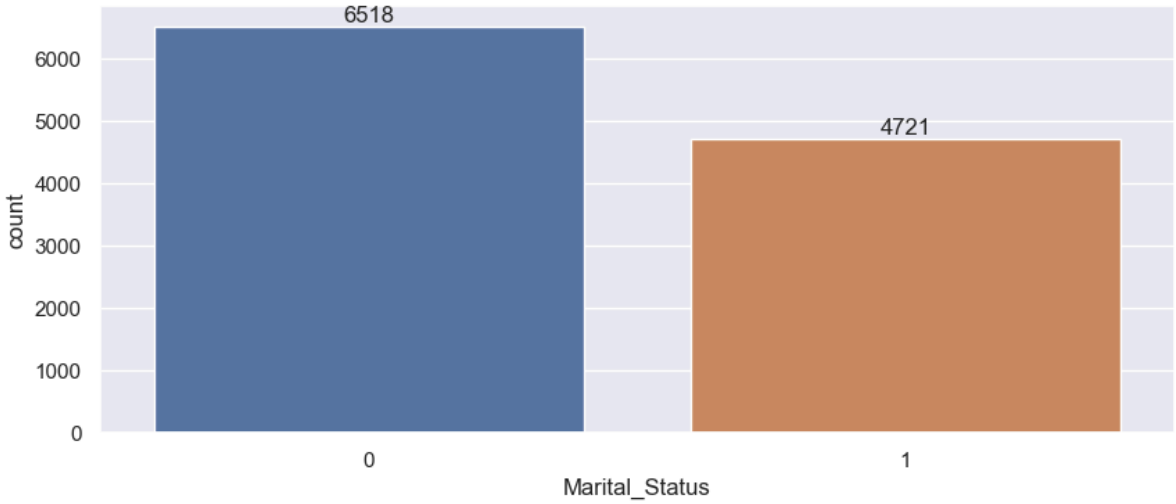
From above graph we can see that unexpectedly,most of the orders are from Uttar Pradesh,maharashtra and Karnataka respecetively but total sales/amount is from

UP,karnataka and then Maharashtra.

Marital Status

In [67]:
```
ax = sns.countplot(data = df , x= 'Marital_Status')

sns.set(rc ={'figure.figsize':(10,5)})
for bars in ax.containers :
    ax.bar_label(bars)
```

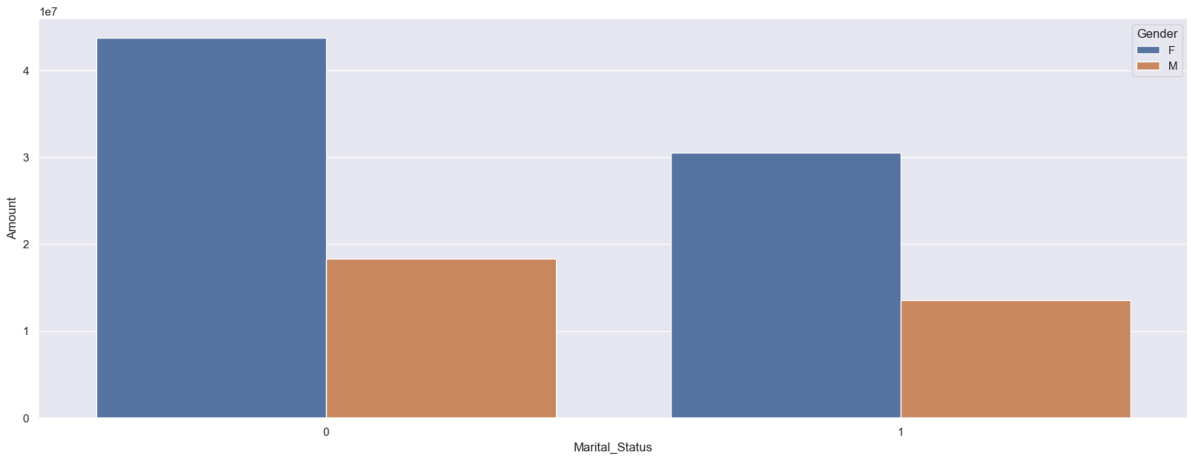


In [59]:
```
sales_state = df.groupby(['Marital_Status','Gender'], as_index=False)['Amount'].sum

# Set the figure size
sns.set(rc={'figure.figsize':(20,7)})

# Plot the barplot
sns.barplot(data=sales_state, x='Marital_Status', y='Amount',hue ='Gender')
```

Out[59]:
```
<Axes: xlabel='Marital_Status', ylabel='Amount'>
```

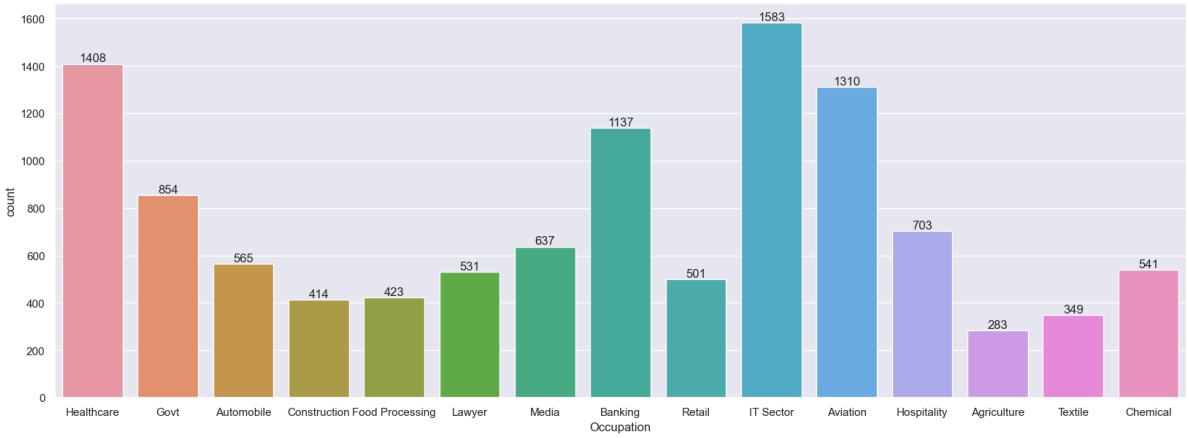


From above graph,we can see that the most of the boys are married(women) and they have high purchasing power.

Occupation

In [159…
```
sns.set(rc={'figure.figsize':(20,7)})
ax = sns.countplot(data = df, x='Occupation')

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

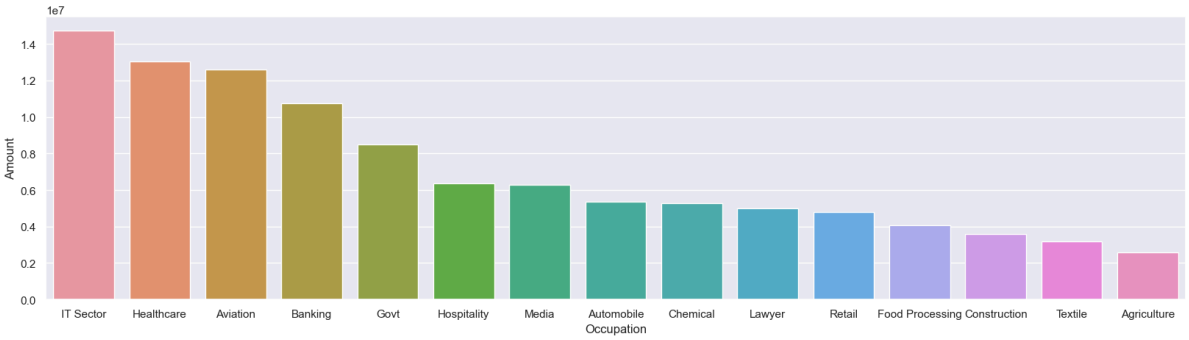


In [161…
```
sales_state = df.groupby(['Occupation'], as_index=False)['Amount'].sum().sort_value

# Set the figure size
sns.set(rc={'figure.figsize':(20,5)})

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

Out[161]:
```
<Axes: xlabel='Occupation', ylabel='Amount'>
```

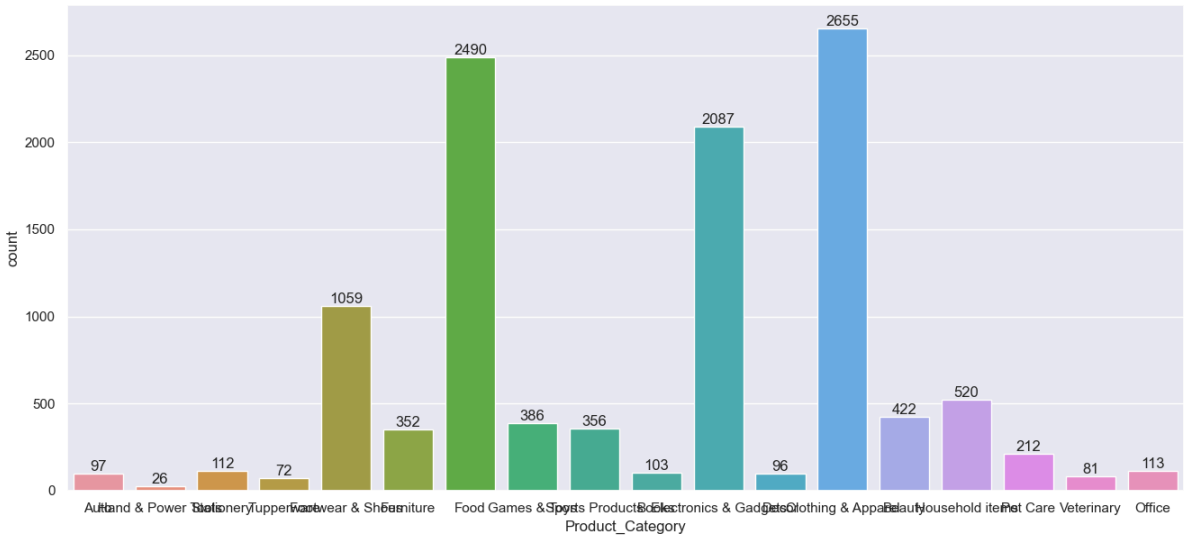


From above graph we can see that the most of the buyers are from IT Sector,Aviation and Healthcare sector.

Product Category

In [173…
```
sns.set(rc={'figure.figsize':(16,7)})
ax = sns.countplot(data = df, x='Product_Category')

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



In [179…

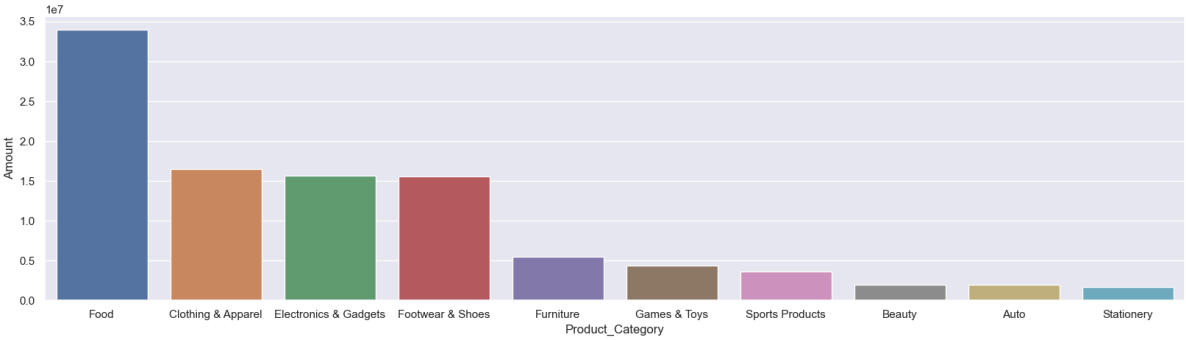
```
sales_state = df.groupby(['Product_Category'], as_index=False)['Amount'].sum().sort

# Set the figure size
sns.set(rc={'figure.figsize':(20,5)})

# Plot the barplot
sns.barplot(data=sales_state, x='Product_Category', y='Amount')
```

Out[179]:

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



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

In [180…

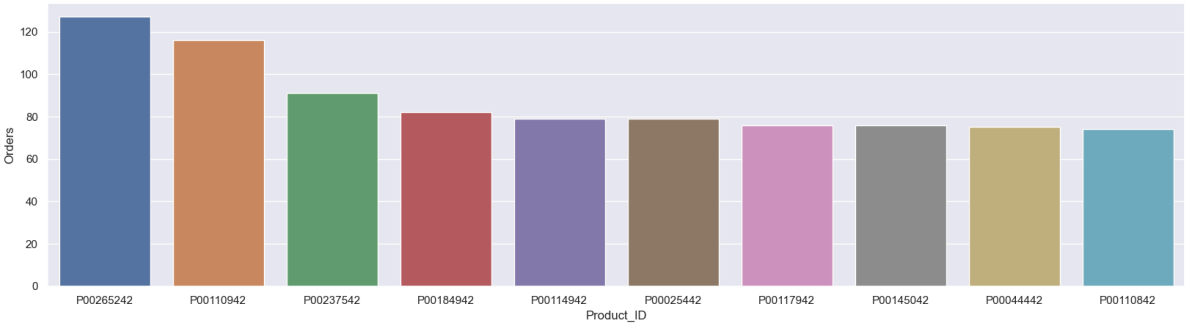
```
sales_state = df.groupby(['Product_ID'], as_index=False)['Orders'].sum().sort_value

# Set the figure size
sns.set(rc={'figure.figsize':(20,5)})

# Plot the barplot
sns.barplot(data=sales_state, x='Product_ID', y='Orders')
```

Out[180]:

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



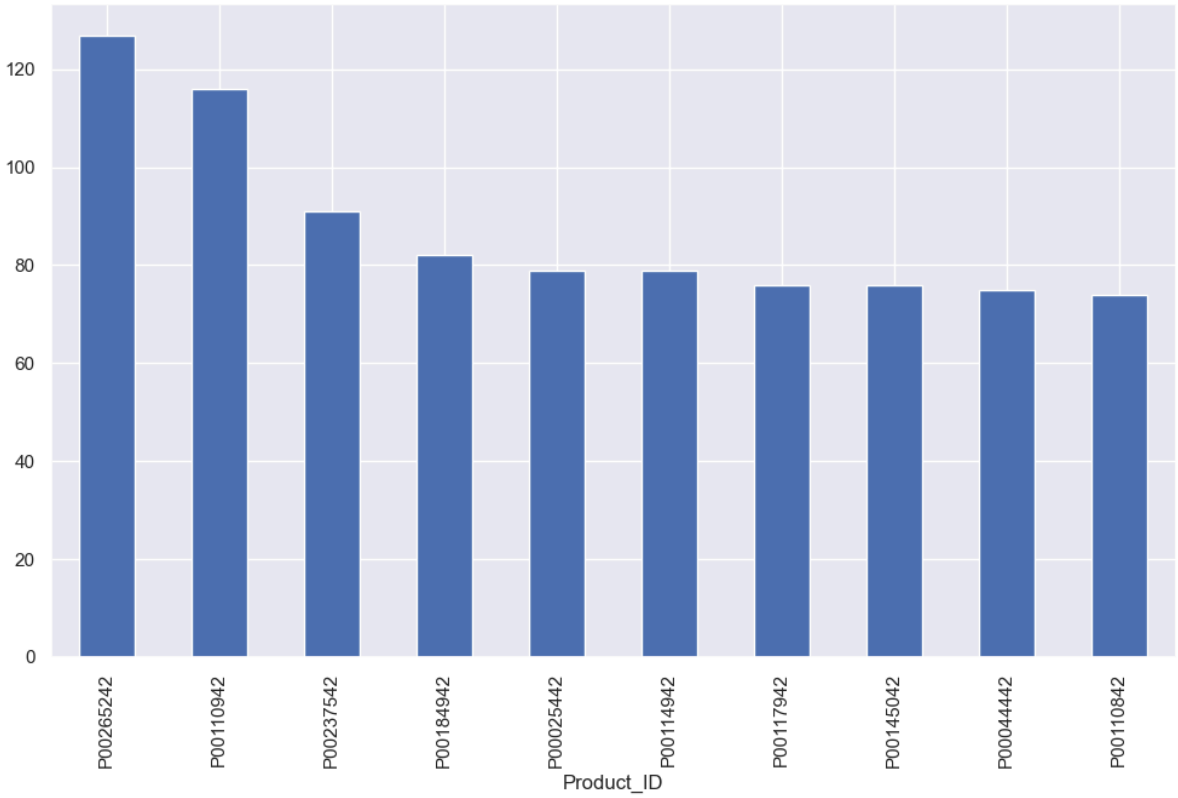
In [182…

```
#top 10 most sold products(same thing as above)

fig1,ax1 =plt.subplots(figsize =(12,7))
df.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascending =False)
```

Out[182]:

```
<Axes: xlabel='Product_ID'>
```



Conclusion

Married women age hgroup 26-35 yrs from UP,Maharashtra and Karnataka working in IT ,Healthcare and Aviation are mkre likely to buy products from food clothing and electronic category.