In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [2]:
```
df=pd.read_csv('C:/Users/NIDHI VIRUPAXI/Desktop/Diwali Sales Data.csv',encoding='uni
```

In [3]:
```
df.shape
```

Out[3]:
```
(11251, 15)
```

In [4]:
```
df.head(10)
```

Out[4]:

	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 [5]:
```
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
```

Dropping unrealated/blank columns

In [10]:
```
df.drop(['Status','unnamed1'],axis=1,inplace=True)#axis 1 refers vertical column and
```

In [11]:
```
#check for null values
pd.isnull(df)
```

Out[11]:

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

11251 rows × 13 columns

In [12]:
```
pd.isnull(df).sum()
```

Out[12]:
```
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 [13]:
```
df.shape
```

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

In [14]:
```
#drop null value
df.dropna(inplace=True)
```

In [15]:
```
df['Amount'].dtype
```

Out[15]:
```
dtype('float64')
```

In [16]:
```
#change datatype of amount from float to int
df['Amount']=df['Amount'].astype('int')
```

In [17]:
```
df['Amount'].dtype
```

Out[17]:
```
dtype('int32')
```

In [18]:
```
df.columns
```

Out[18]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [19]:
```
df.describe()
```

Out[19]:

	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 [21]:
```
#describe for specific columns
df[['Age','Orders']].describe()
```

Out[21]:

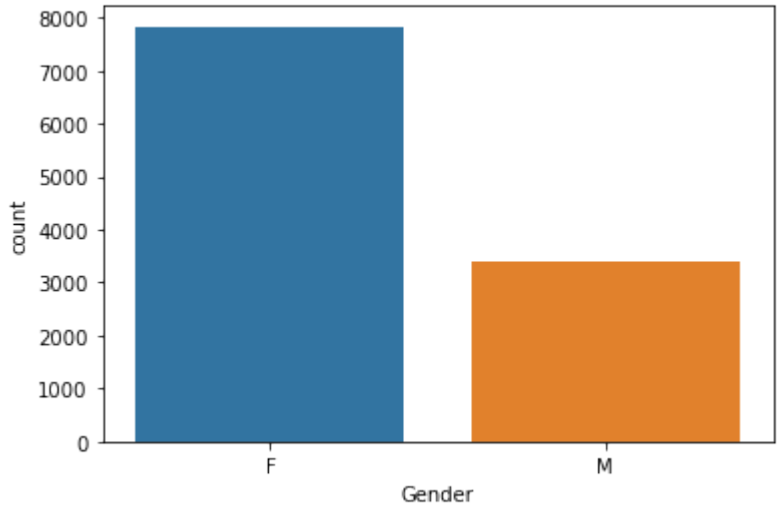
	Age	Orders
count	11239.000000	11239.000000
mean	35.410357	2.489634
std	12.753866	1.114967
min	12.000000	1.000000
25%	27.000000	2.000000

	Age	Orders
50%	33.000000	2.000000
75%	43.000000	3.000000
max	92.000000	4.000000

Exploratory Data Analysis

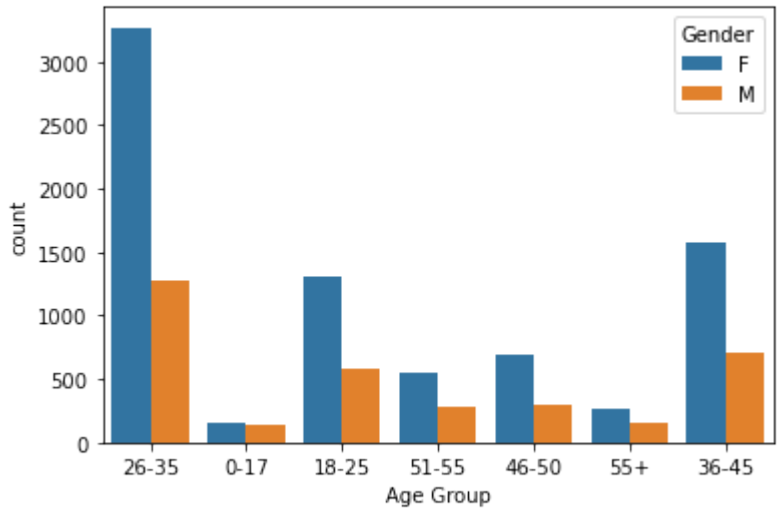
In [22]:
```
sns.countplot('Gender',data=df)
plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only valid po
sitional argument will be `data`, and passing other arguments without an explicit key
word will result in an error or misinterpretation.
  warnings.warn(
```



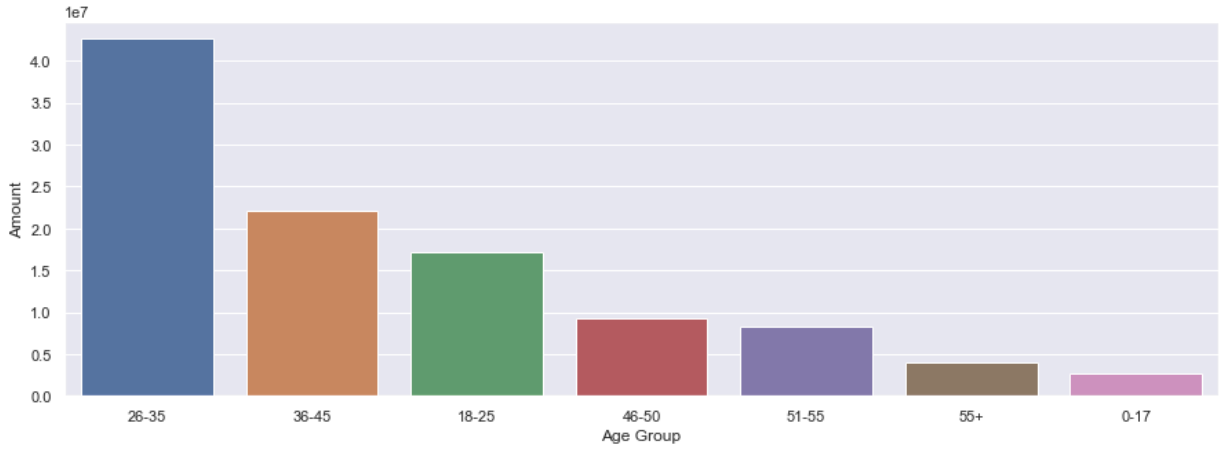
From above graph we can see that most of the buyers are female and more purchase than men.

In [23]:
```
sns.countplot(x='Age Group',data=df,hue='Gender')
plt.show()
```



In [29]:
```
#Total amount vs age group
sales_age=df.groupby(['Age Group'],as_index=False)['Amount'].sum().sort_values(by='A
sns.barplot(x='Age Group',y='Amount',data=sales_age)
```

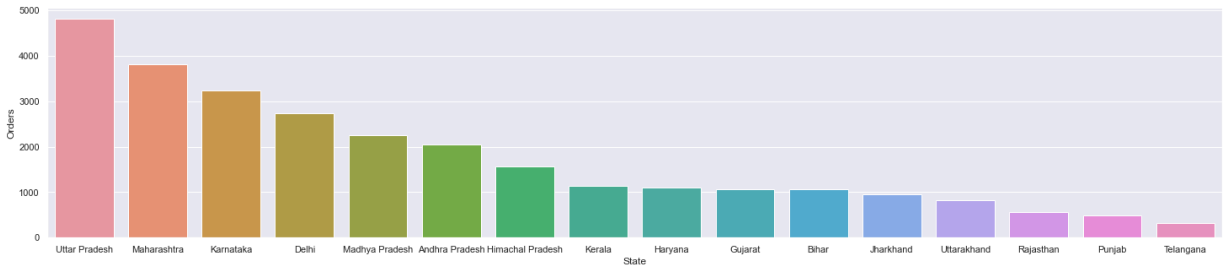
Out[29]:
```
<AxesSubplot:xlabel='Age Group', ylabel='Amount'>
```



STATE

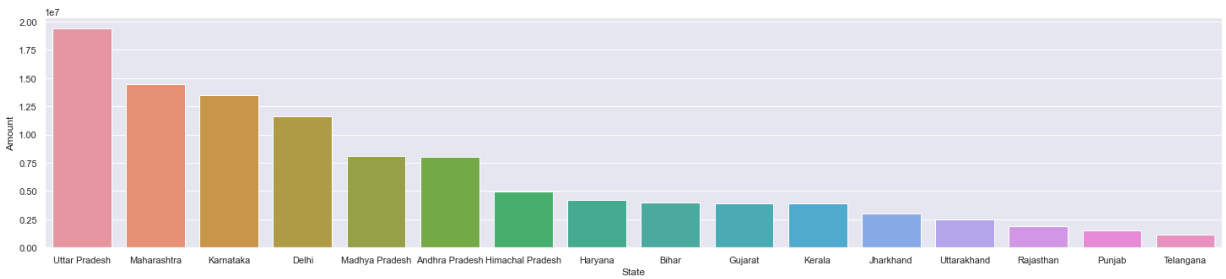
In [44]:
```
#State VS Order

sales_state=df.groupby(['State'],as_index=False)['Orders'].sum().sort_values(by='Ord
sns.set(rc={'figure.figsize':(25,5)})

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

Out[44]:
```
<AxesSubplot:xlabel='State', ylabel='Orders'>
```



In [45]:
```
sales_state=df.groupby(['State'],as_index=False)['Amount'].sum().sort_values(by='Amo
sns.set(rc={'figure.figsize':(25,5)})

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

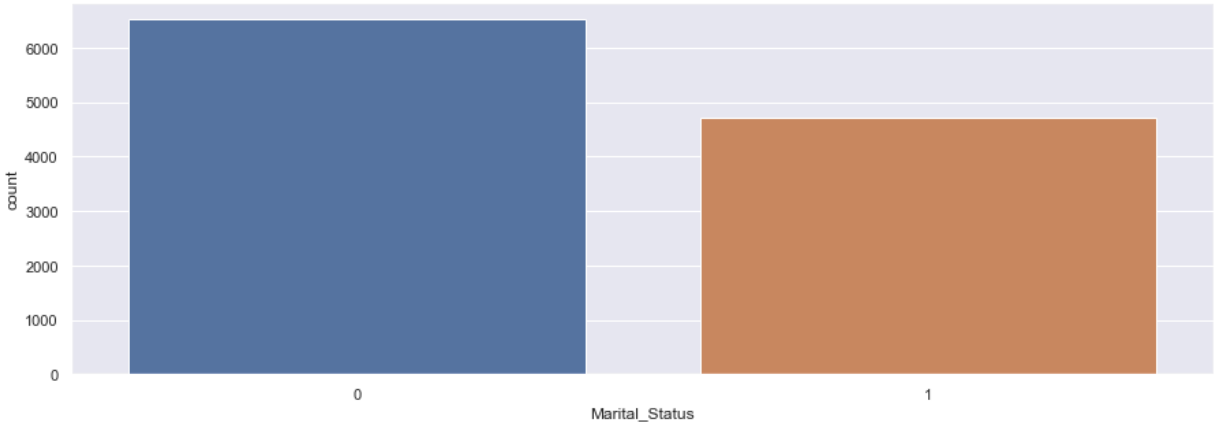
Out[45]:
```
<AxesSubplot:xlabel='State', ylabel='Amount'>
```



Marital Status

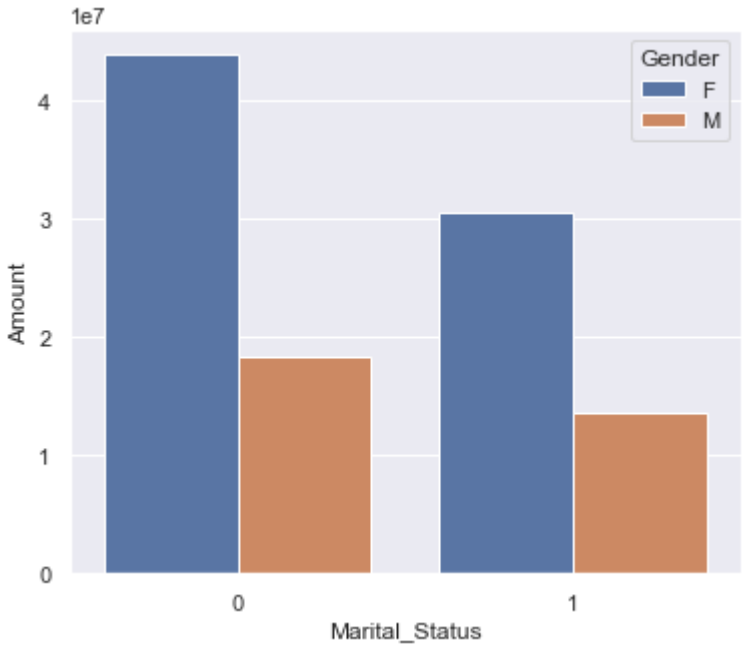
In [36]:
```
sns.countplot('Marital_Status',data=df)
plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only valid po
sitional argument will be `data`, and passing other arguments without an explicit key
word will result in an error or misinterpretation.
  warnings.warn(
```



In [37]:
```
sales=df.groupby(['Marital_Status','Gender'],as_index=False)['Amount'].sum().sort_va
sns.set(rc={'figure.figsize':(6,5)})

sns.barplot(x='Marital_Status',y='Amount',data=sales,hue='Gender')
```

Out[37]: <AxesSubplot:xlabel='Marital_Status', ylabel='Amount'>



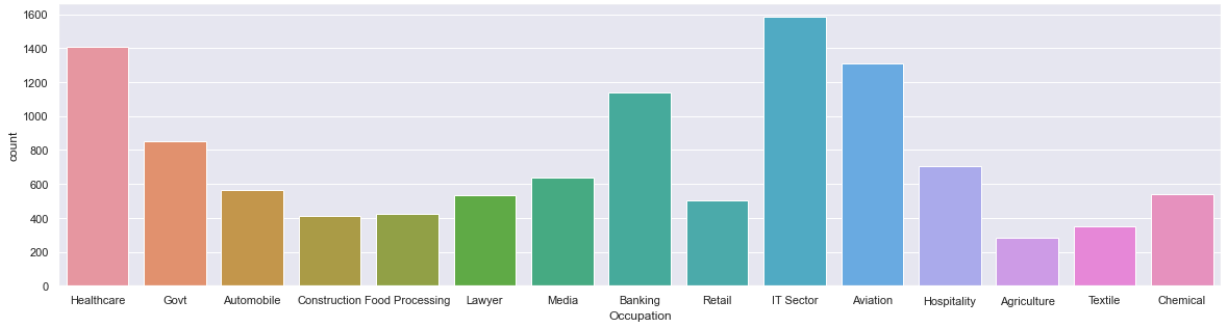
From above graph we can see that most of the married womens are buyers with higher purchase.

OCCUPATION

In [41]:
```
sns.countplot('Occupation',data=df)
sns.set(rc={'figure.figsize':(50,5)})
plt.show()
```

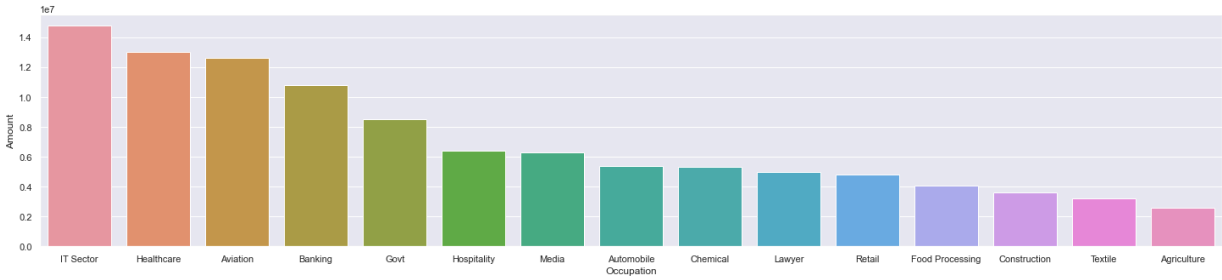
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only valid po
sitional argument will be `data`, and passing other arguments without an explicit key
```

```
word will result in an error or misinterpretation.
  warnings.warn(
```



In [50]:
```
sales_occ=df.groupby(['Occupation'],as_index=False)['Amount'].sum().sort_values(by='
sns.set(rc={'figure.figsize':(25,5)})

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

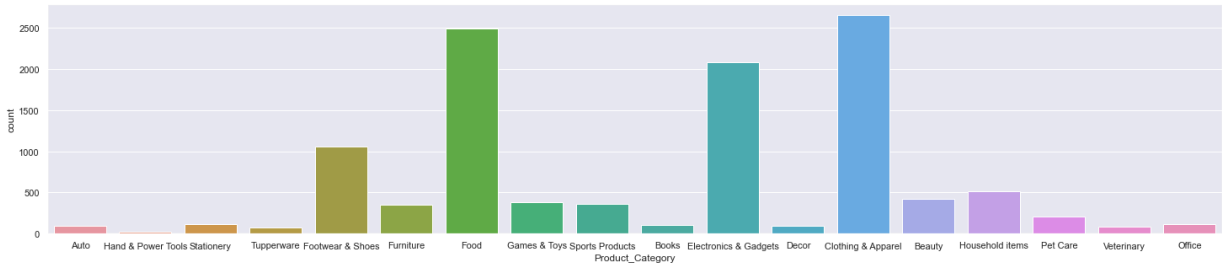
Out[50]:
```
<AxesSubplot:xlabel='Occupation', ylabel='Amount'>
```



PRODUCT CATEGORY

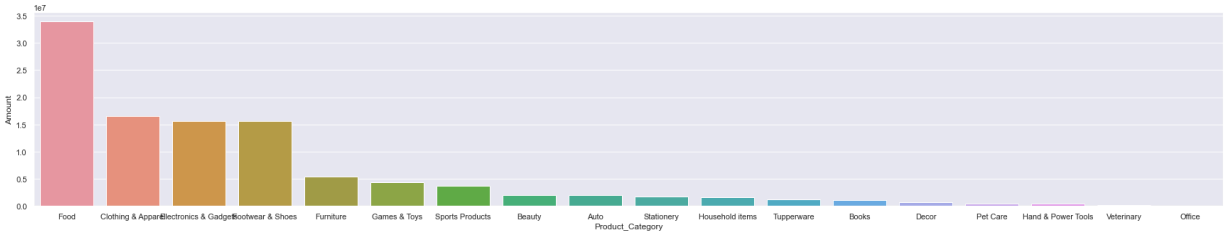
In [58]:
```
sns.countplot('Product_Category',data=df)
sns.set(rc={'figure.figsize':(50,5)})
plt.show()
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only valid po
sitional argument will be `data`, and passing other arguments without an explicit key
word will result in an error or misinterpretation.
  warnings.warn(
```



In [62]:
```
sales_pc=df.groupby(['Product_Category'],as_index=False)['Amount'].sum().sort_values
sns.set(rc={'figure.figsize':(30,5)})

sns.barplot(x='Product_Category',y='Amount',data=sales_pc)
```

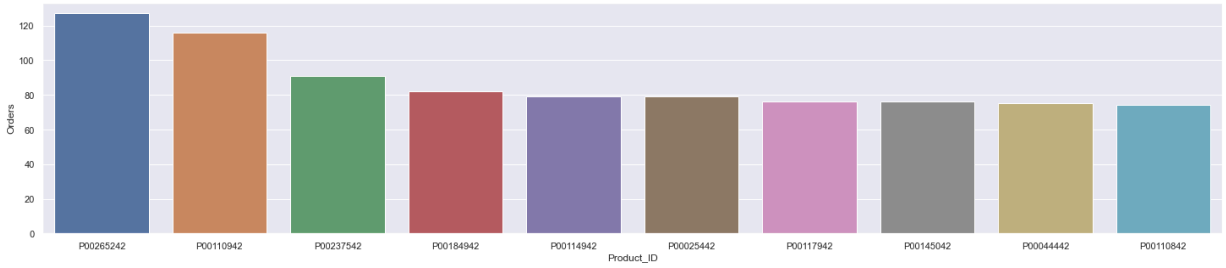
Out[62]:
```
<AxesSubplot:xlabel='Product_Category', ylabel='Amount'>
```



PRODUCT ID

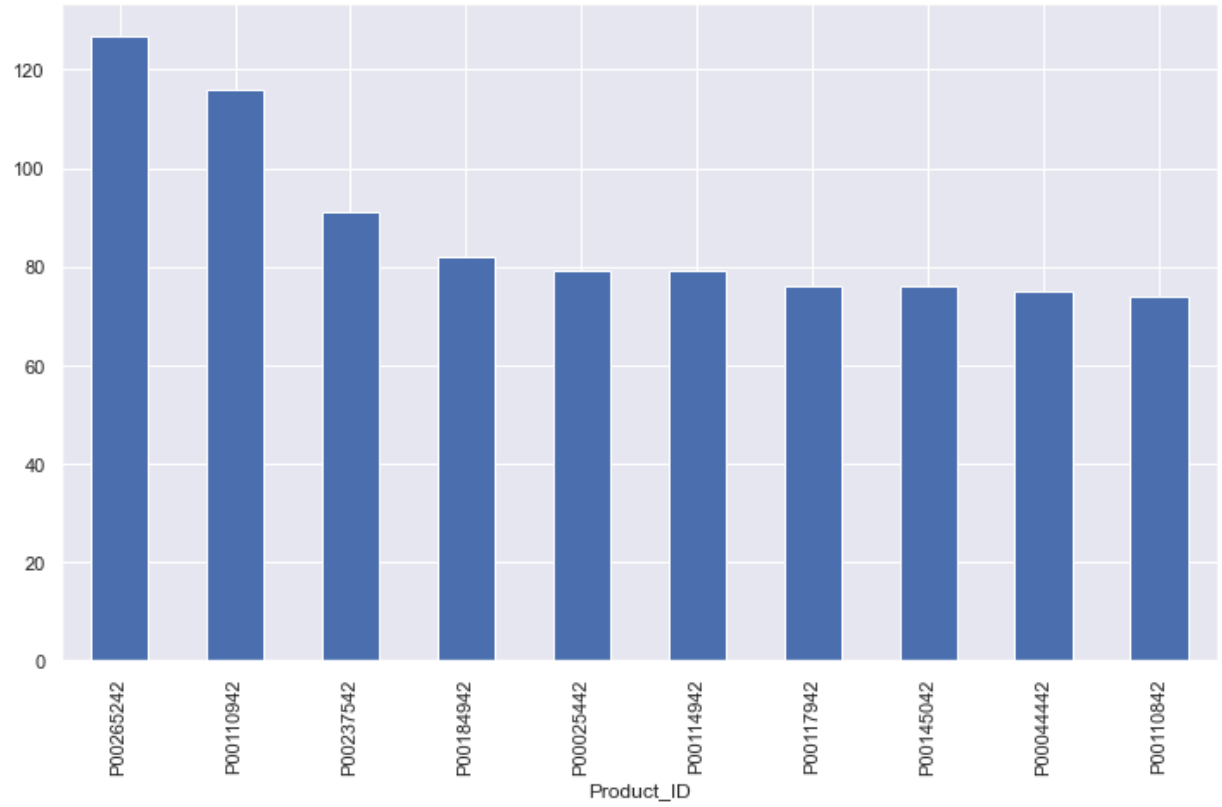
In [70]:
```
sales_po=df.groupby(['Product_ID'],as_index=False)['Orders'].sum().sort_values(by='O
sns.set(rc={'figure.figsize':(25,5)})

sns.barplot(x='Product_ID',y='Orders',data=sales_po)
```

Out[70]: <AxesSubplot:xlabel='Product_ID', ylabel='Orders'>



In [76]:
```
fig1,ax1=plt.subplots(figsize=(12,7))
df.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascending=False).p
```

Out[76]: <AxesSubplot:xlabel='Product_ID'>



CONCLUSION:

Married women of age group 26-35 years from Uttarpradesh,Maharashtra and Karnataka working in IT,Healthcare and Aviation are more likely to buy products from Food,Clothing and Electronics Category.