In [39]:
```
#Import python liabraries

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

In [40]:
```
df = pd.read_csv('C:\\Users\\Manas\\Downloads\\Python_Diwali_Sales_Analysis
```

In [41]:
```
df
```

Out[41]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	Stat
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtr
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
...	...	...	...	...	...	...	...	.
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtr
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryan
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhy Prades
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnatak
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtr

11251 rows × 15 columns

In [42]:
```
df.shape
```

Out[42]:
```
(11251, 15)
```

In [43]:
```
df.head()
```

Out[43]:

r_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	C
2903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	
0732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	
1990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	
1425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	C
0588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	

In [44]:
```
df.tail()
```

Out[44]:

ser_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	C
000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	
004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	
001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	
004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	
002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	

In [45]:
```
df.info()
```

```
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 [46]:
```
#drop unrelated blank values
df.drop(["Status", "unnamed1"],axis = 1,inplace = True)
```

In [47]:
```
#check for null values
pd.isnull(df).sum()
```

Out[47]:
```
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 [48]:
```
# drop null values
df.dropna(inplace = True)
```

In [49]:
```
pd.isnull(df).sum()
```

Out[49]:
```
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              0
dtype: int64
```

In [29]:
```
# change data type
df['Amount'] = df['Amount'].astype('int')
```

In [30]:
```
df['Amount'].dtypes
```

Out[30]:
```
dtype('int32')
```

In [31]:
```
df.columns
```

Out[31]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Categor
y',
       'Orders', 'Amount'],
      dtype='object')
```

In [14]:
```
#rename column
df.rename(columns= {'Marital_Status':'Shaadi'})
```

Out[14]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Shaadi	State	Z
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	We
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Sou
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Ce
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Sou
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	We
...	...	...	...	...	...	...	...	...	
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	We
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Nor
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Ce
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Sou
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	We

11239 rows × 13 columns

In [15]:
```
# describe() method returns description of the data in the DataFrame (i.e.
df.describe()
```

Out[15]:

	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 [16]:
```
# use describe() for specific columns
df[['Age', 'Orders', 'Amount']].describe()
```

Out[16]:

	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

In [34]:
```
# what are the transaction entries where amount exceed 1000
high_amount_data = df[df['Amount'] >1000]
print("Rows where 'Amount'is greater than 1000:")
print(high_amount_data.head())
```

```
Rows where 'Amount'is greater than 1000:
   User_ID  Cust_name Product_ID Gender Age Group  Age  Marital_Status  \
0  1002903  Sanskriti  P00125942      F     26-35   28               0
1  1000732     Kartik  P00110942      F     26-35   35               1
2  1001990      Bindu  P00118542      F     26-35   35               1
3  1001425     Sudevi  P00237842      M      0-17   16               0
4  1000588       Joni  P00057942      M     26-35   28               1

            State      Zone       Occupation Product_Category  Orders  Amo
unt
0     Maharashtra   Western       Healthcare             Auto       1   23
952
1  Andhra Pradesh  Southern             Govt             Auto       3   23
934
2   Uttar Pradesh   Central       Automobile             Auto       3   23
924
3       Karnataka  Southern     Construction             Auto       2   23
912
4         Gujarat   Western  Food Processing             Auto       2   23
877
```

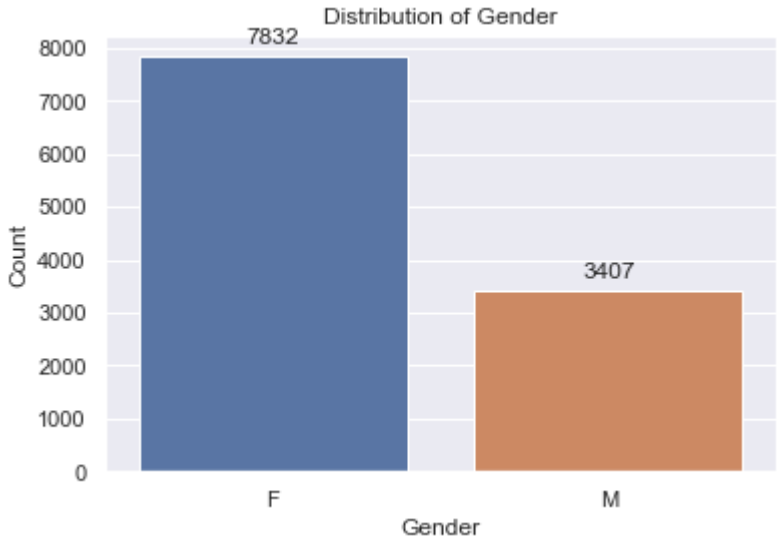
In [35]:
```
# identifing top 10 customer based on their total purchase amount:
customer_purchase_totals = df.groupby('User_ID')['Amount'].sum()
high_value_customers = customer_purchase_totals.nlargest(10)
print("Top 10 High-value customers:")
print(high_value_customers)
```

```
Top 10 High-value customers:
User_ID
1001680    281034
1001941    239147
1003476    220435
1002665    201104
1003808    197660
1004425    194343
1003618    189921
1000424    187679
1004682    185122
1001298    184045
Name: Amount, dtype: int32
```

Exploratory Data Analysis

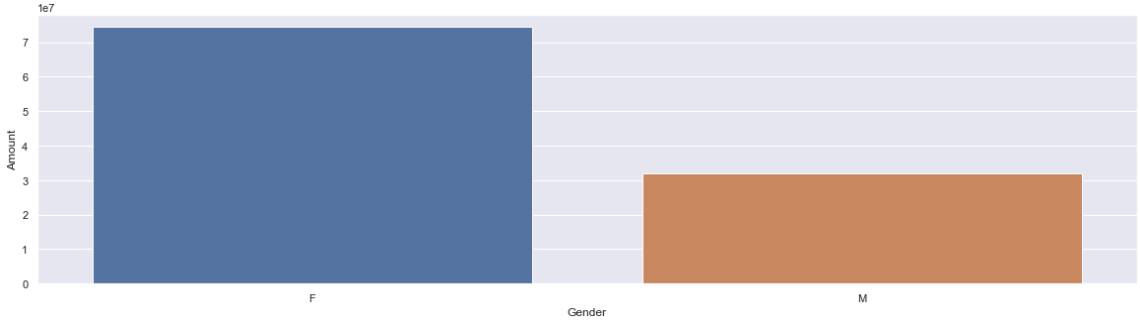
Gender

In [45]:
```
# plotting a bar chart for Gender and it's count
#analyze the distribution of gender in dataset

import seaborn as sns
import matplotlib.pyplot as plt

sns.set(rc={'figure.figsize':(6,4)})
ax = sns.countplot(x='Gender', data=df)
ax.set_title('Distribution of Gender')
ax.set_xlabel('Gender')
ax.set_ylabel('Count')

# Add labels to the bars
for p in ax.patches:
    height = p.get_height()
    ax.annotate(f'{height}', (p.get_x() + p.get_width() / 2., height), ha='

plt.show()
```



In [38]:
```
# plotting a bar chart for gender vs total amount

sales_gen = df.groupby(['Gender'], as_index=False)['Amount'].sum().sort_val

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

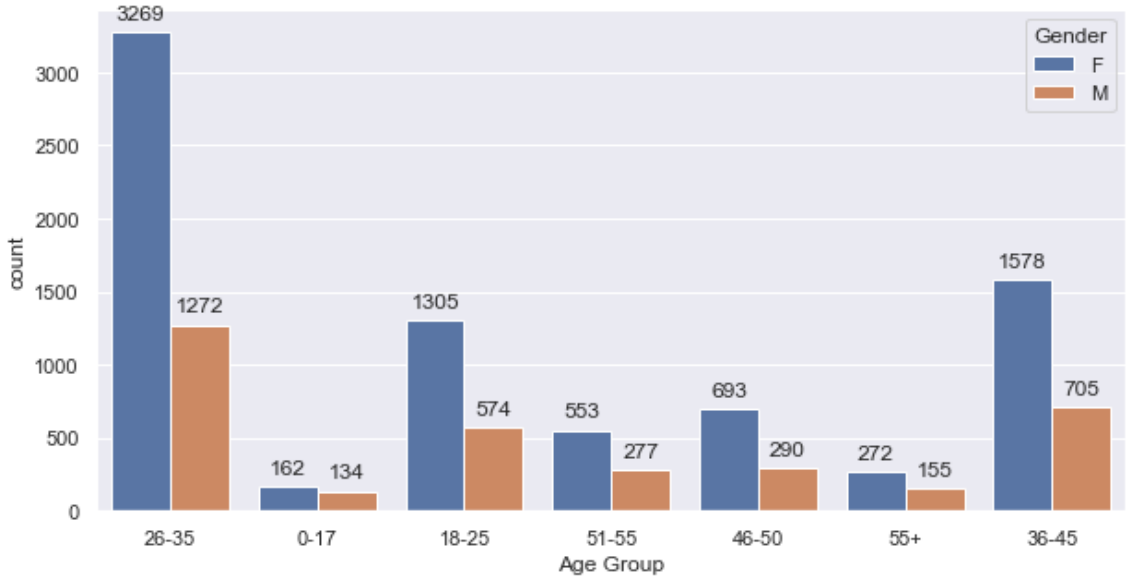
Out[38]:
```
<AxesSubplot:xlabel='Gender', ylabel='Amount'>
```



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

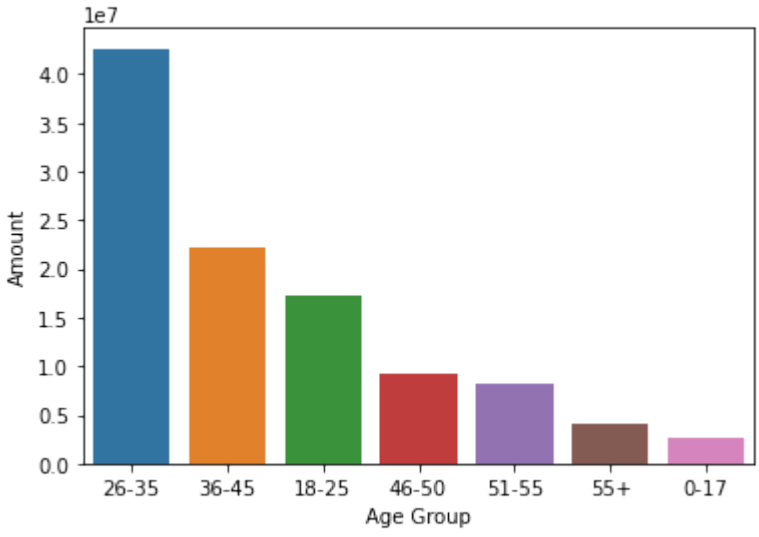
Age

In [42]:
```
sns.set(rc={'figure.figsize':(10,5)})
ax = sns.countplot(data=df, x='Age Group', hue='Gender')

# Add labels to the bars
for p in ax.patches:
    height = p.get_height()
    ax.annotate(f'{height}', (p.get_x() + p.get_width() / 2., height), ha='

plt.show()
```



In [18]:
```
# Total Amount vs Age Group
sales_age = df.groupby(['Age Group'], as_index=False)['Amount'].sum().sort_

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

Out[18]:
```
<AxesSubplot:xlabel='Age Group', ylabel='Amount'>
```

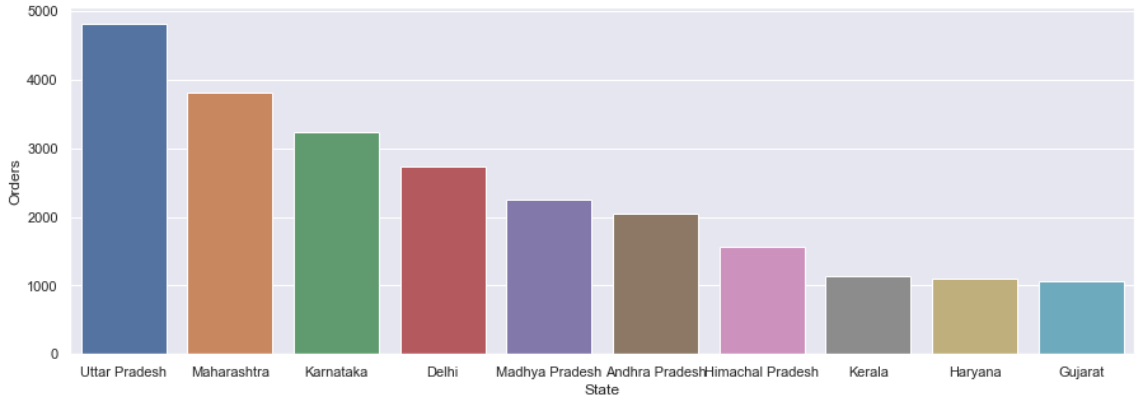


From above graphs we can see that most of the buyers are of age group between 26-35 yrs female

State

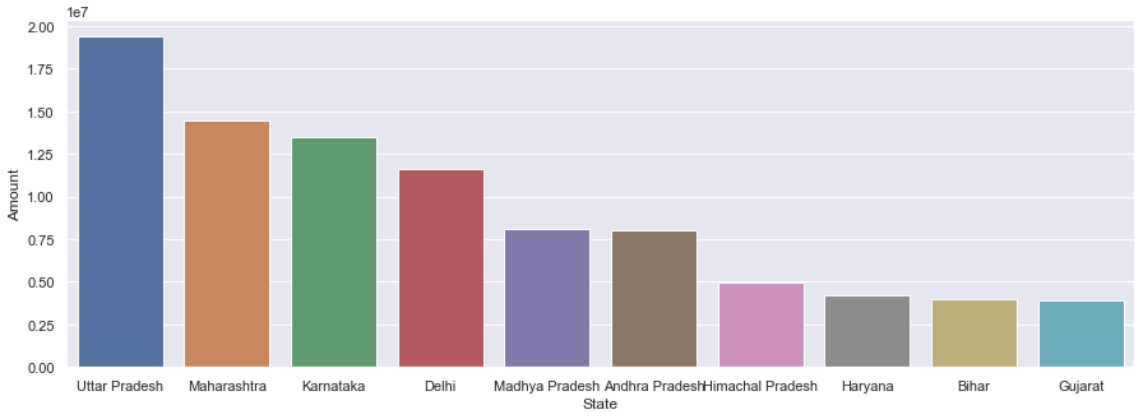
In [19]:
```
# total number of orders from top 10 states

sales_state = df.groupby(['State'], as_index=False)['Orders'].sum().sort_va

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

Out[19]: <AxesSubplot:xlabel='State', ylabel='Orders'>



In [20]:
```
# total amount/sales from top 10 states

sales_state = df.groupby(['State'], as_index=False)['Amount'].sum().sort_va

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

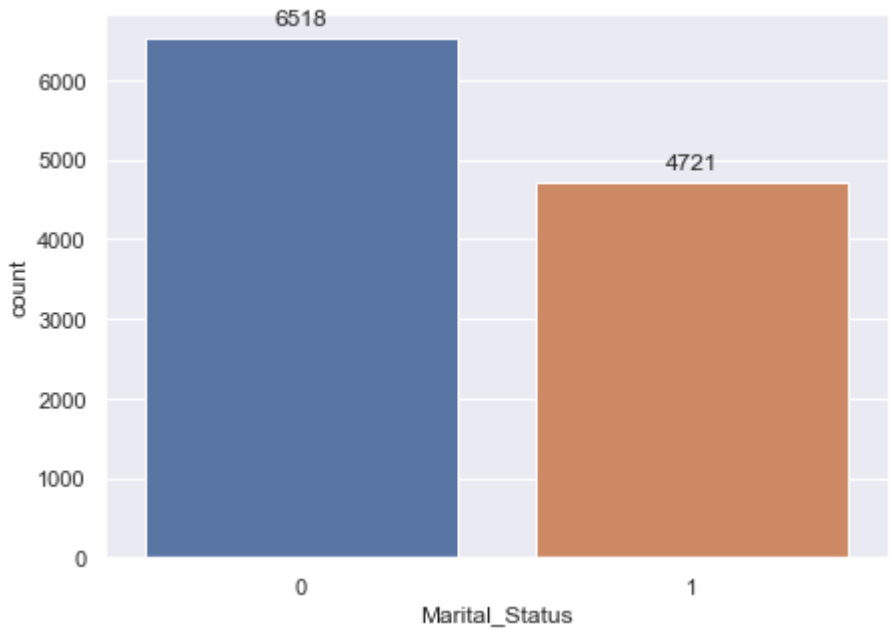
Out[20]: <AxesSubplot:xlabel='State', ylabel='Amount'>



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

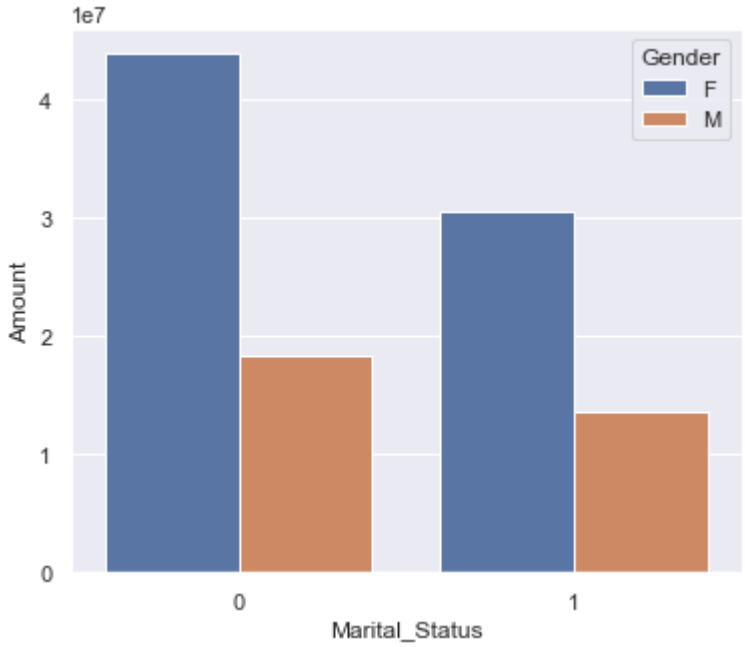
Marital Status

In [46]:
```
import seaborn as sns
import matplotlib.pyplot as plt

sns.set(rc={'figure.figsize':(7,5)})
ax = sns.countplot(data=df, x='Marital_Status')

# Add labels to the bars
for p in ax.patches:
    height = p.get_height()
    ax.annotate(f'{height}', (p.get_x() + p.get_width() / 2., height), ha='

plt.show()
```



In [22]:
```
sales_state = df.groupby(['Marital_Status', 'Gender'], as_index=False)['Amo

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

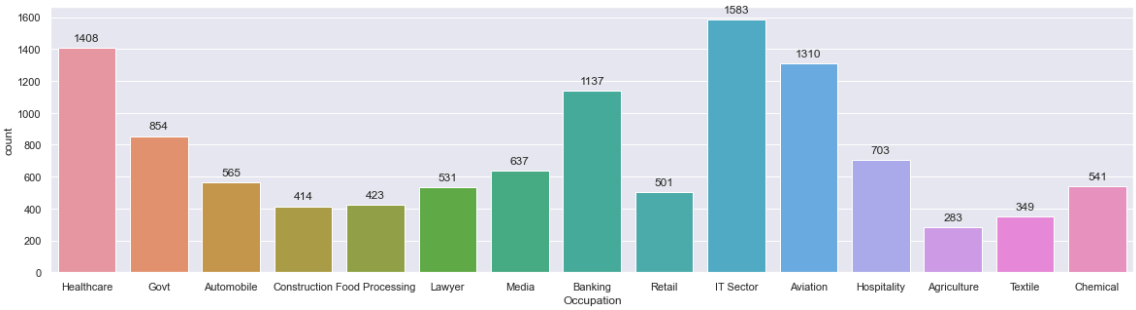
Out[22]: <AxesSubplot:xlabel='Marital_Status', ylabel='Amount'>



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

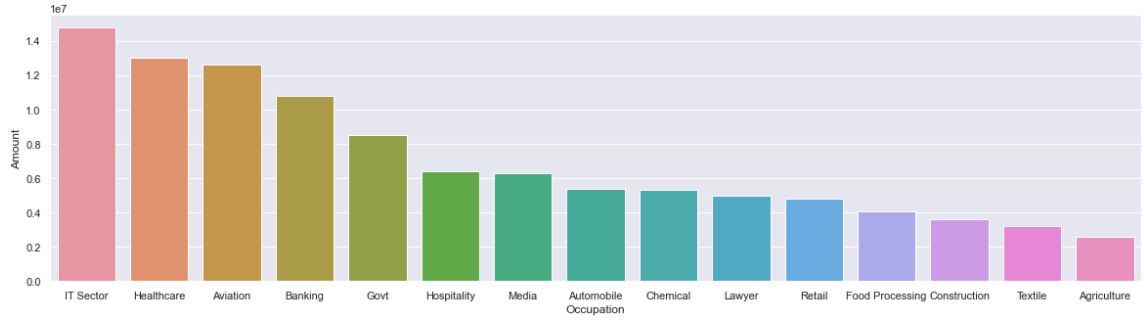
Occupation

In [47]:
```
import seaborn as sns
import matplotlib.pyplot as plt

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

# Add labels to the bars
for p in ax.patches:
    height = p.get_height()
    ax.annotate(f'{height}', (p.get_x() + p.get_width() / 2., height), ha='

plt.show()
```



In [24]:
```
sales_state = df.groupby(['Occupation'], as_index=False)['Amount'].sum().so

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

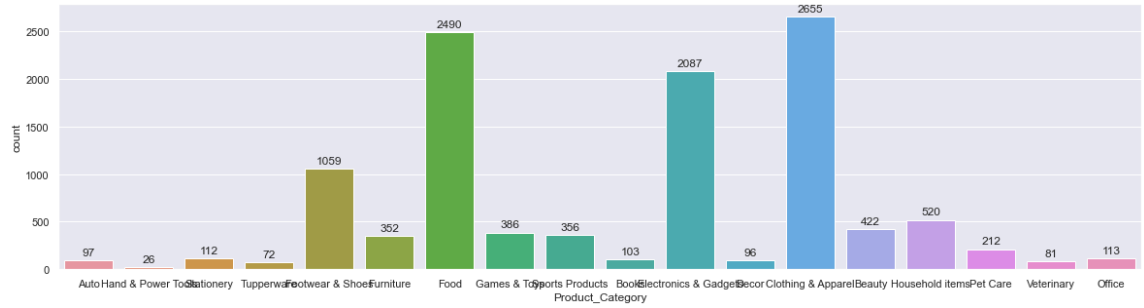
Out[24]: <AxesSubplot:xlabel='Occupation', ylabel='Amount'>



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

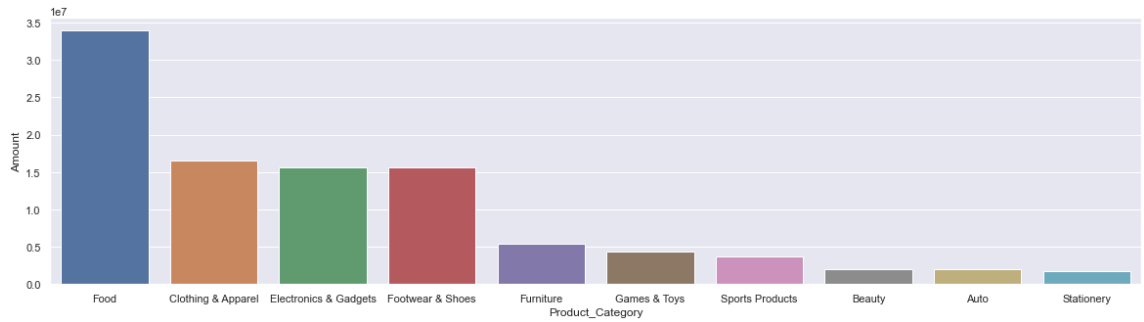
Product Category

In [54]:
```
ort seaborn as sns
ort matplotlib.pyplot as plt

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

dd labels to the bars
p in ax.patches:
 height = p.get_height()
 ax.annotate(f'{height}', (p.get_x() + p.get_width() / 2., height), ha='cen

.show()
```



In [26]:
```
sales_state = df.groupby(['Product_Category'], as_index=False)['Amount'].su

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

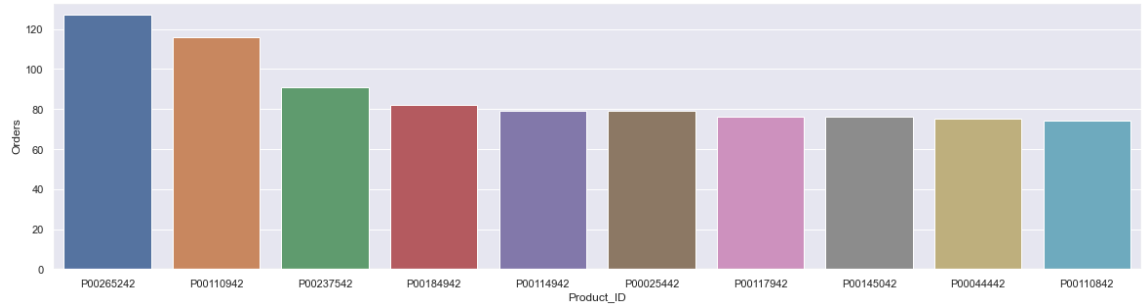
Out[26]: <AxesSubplot:xlabel='Product_Category', ylabel='Amount'>



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

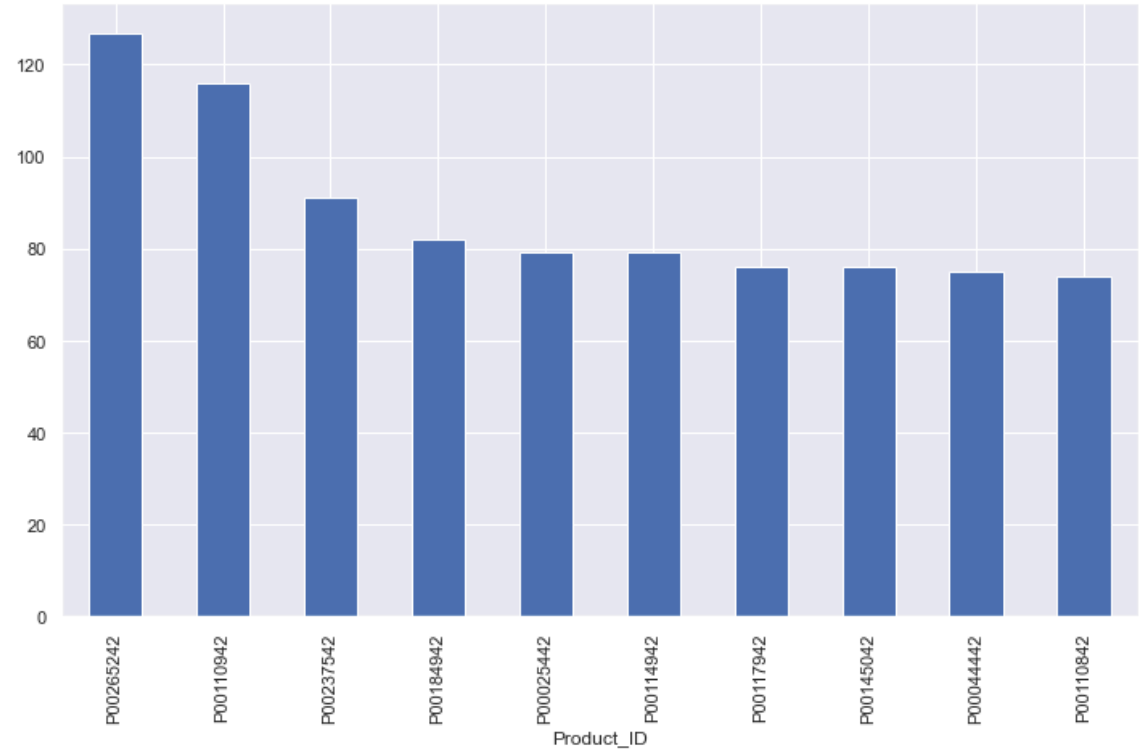
In [27]:
```
sales_state = df.groupby(['Product_ID'], as_index=False)['Orders'].sum().so

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

Out[27]: <AxesSubplot:xlabel='Product_ID', ylabel='Orders'>



In [28]:
```
# 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
```

Out[28]:
```
<AxesSubplot:xlabel='Product_ID'>
```



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