Sales Analysis

In [2]:
```
#importing important libraries for data analysis and graph plots
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:
```
#importing csv data file for analysis
df = pd.read_csv(r'C:\Users\Lenovo\OneDrive\Documents\sales_data.csv')
```

In [4]:
```
#checking total columns and rows
df.shape
```

Out[4]:
```
(11251, 15)
```

In [6]:
```
#checking first 10 heads of the data sets
df.head(10)
```

Out[6]:

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

In [9]:
```
#getting basic info on the data sets
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 [10]:
```
#droping the columns containing zero values
df.drop(['Status', 'unnamed1'], axis=1, inplace=True)
```

In [10]:
```
#verifying the data set after drop operation
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 [11]:
```
#checking the data sets at glance for null values
pd.isnull(df).sum()
```

Out[11]:
```
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 [12]:
```
#droping null values
df.dropna(inplace=True)
```

In [13]:
```
#revisiting total columns and rows
df.shape
```

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

In [14]:
```
#change data type
df['Amount'] = df['Amount'].astype('int')
```

In [15]:
```
#verifying the data type change
```

```
df['Amount'].dtypes
```

Out[15]:
```
dtype('int32')
```

In [16]:
```
#checking all the indexes
df.columns
```

Out[16]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [17]:
```
#quick look at the statistics
df.describe()
```

Out[17]:

	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

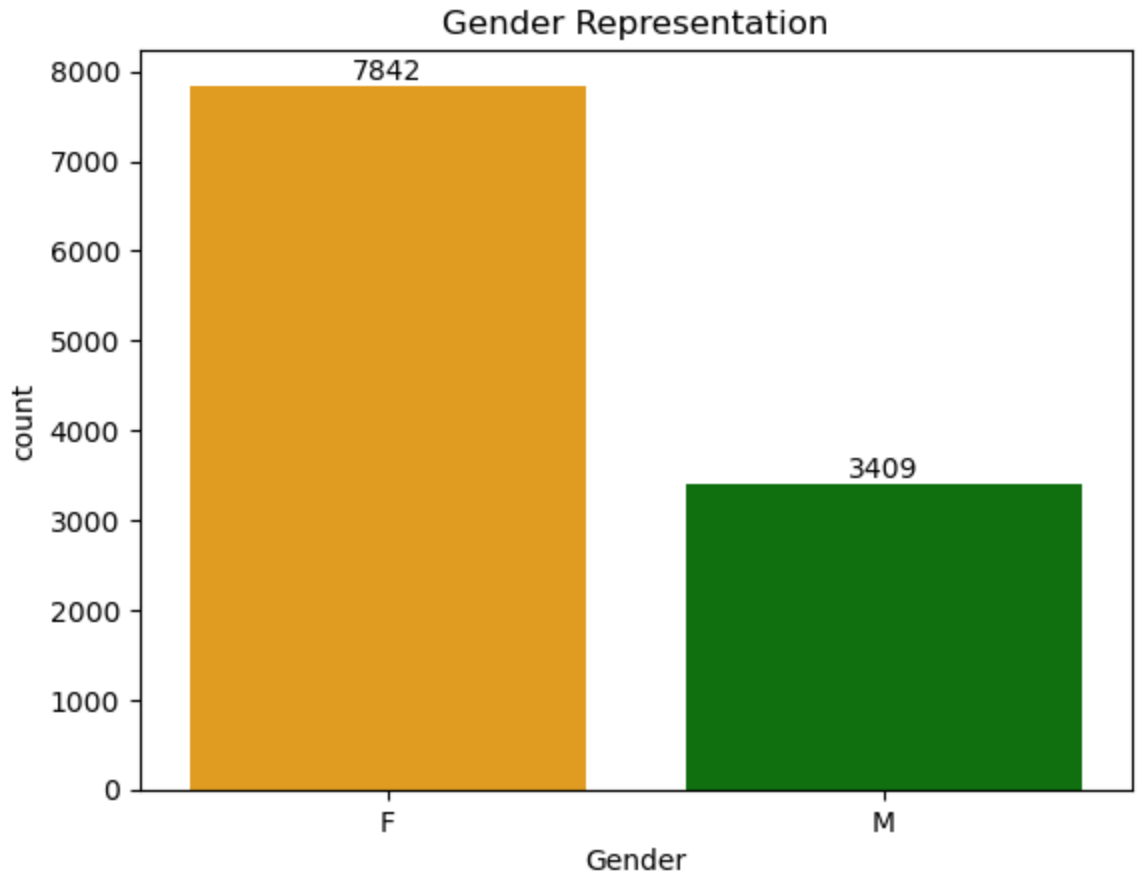
In [18]:
```
#on specific describe
df[['Age', 'Orders', 'Amount']].describe()
```

Out[18]:

	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 [15]:
```
#gender participation in shoping
sns.countplot(x = 'Gender', data = df, palette=['Orange', 'Green'])
ax = plt.gca()

plt.title('Gender Representation')
for bars in ax.containers:
    ax.bar_label(bars)
```



In [20]:
```
#grouping gender and calculating amount spending
grouped = df.groupby(['Gender'], as_index = False) ['Amount'].sum()
```

In [21]:
```
#sorting them on the basis of amount
sorted_result = grouped.sort_values(by = 'Amount', ascending= False)
```

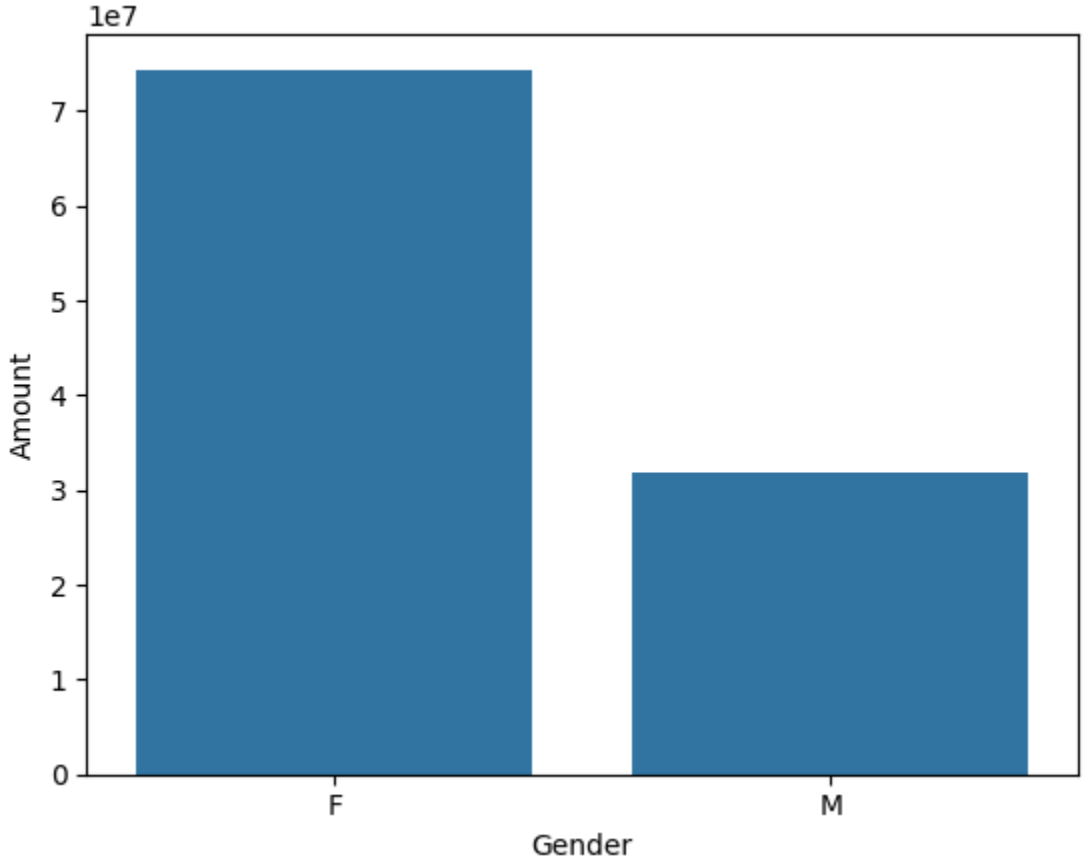
In [22]:
```
#declaring results
sorted_result
```

Out[22]:

	Gender	Amount
0	F	74335853
1	M	31913276

In [30]:
```
#percentage of amount spent by both gender
gender_sales = df.groupby(['Gender'], as_index=False) ["Amount"].sum().sort_values(by='A

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

Out[30]:
```
<Axes: xlabel='Gender', ylabel='Amount'>
```



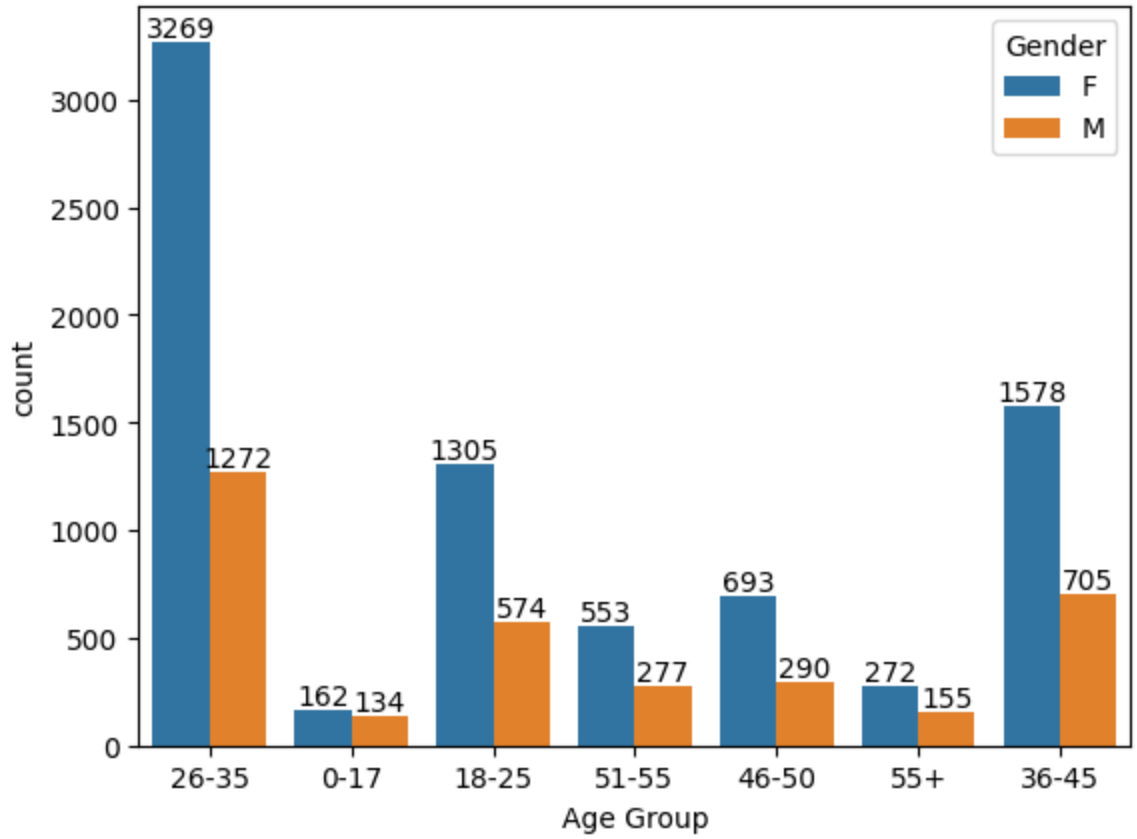
In [ ]:
```
##most purchases was made by the women than mens
```

Age

In [32]:
```
#checking all the indexes
df.columns
```

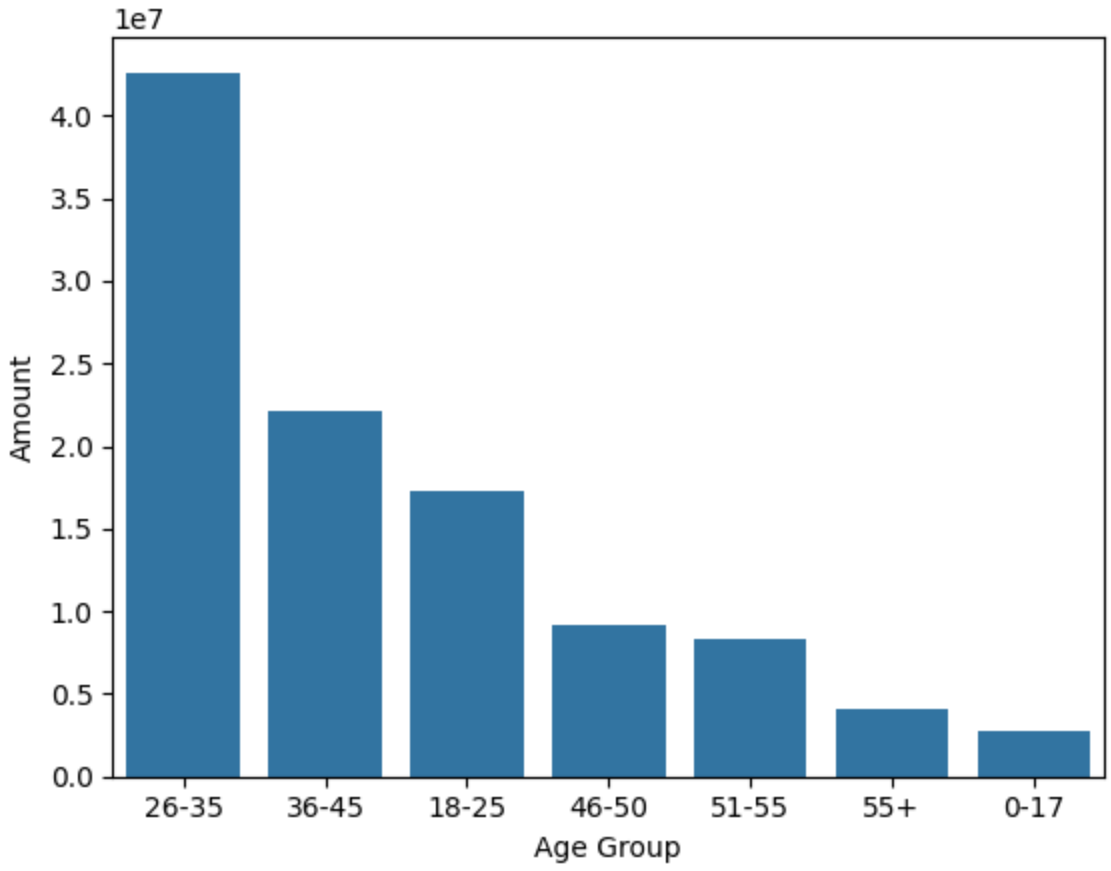
Out[32]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [33]:
```
#representation of age group participation in shoping
ax = sns.countplot (data = df, x= 'Age Group', hue= 'Gender')

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



In [34]:
```
#Total amount vs Age group
sales_age = df.groupby(['Age Group'], as_index=False)['Amount'].sum().sort_values(by="Am

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

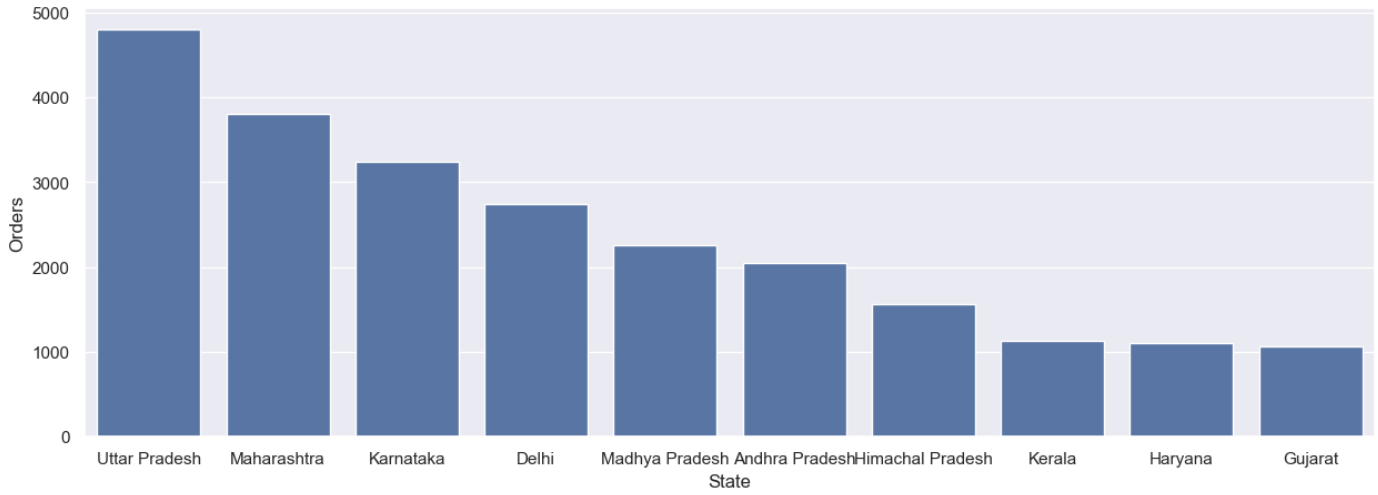
Out[34]:
```
<Axes: xlabel='Age Group', ylabel='Amount'>
```



State-wise orders and order amount

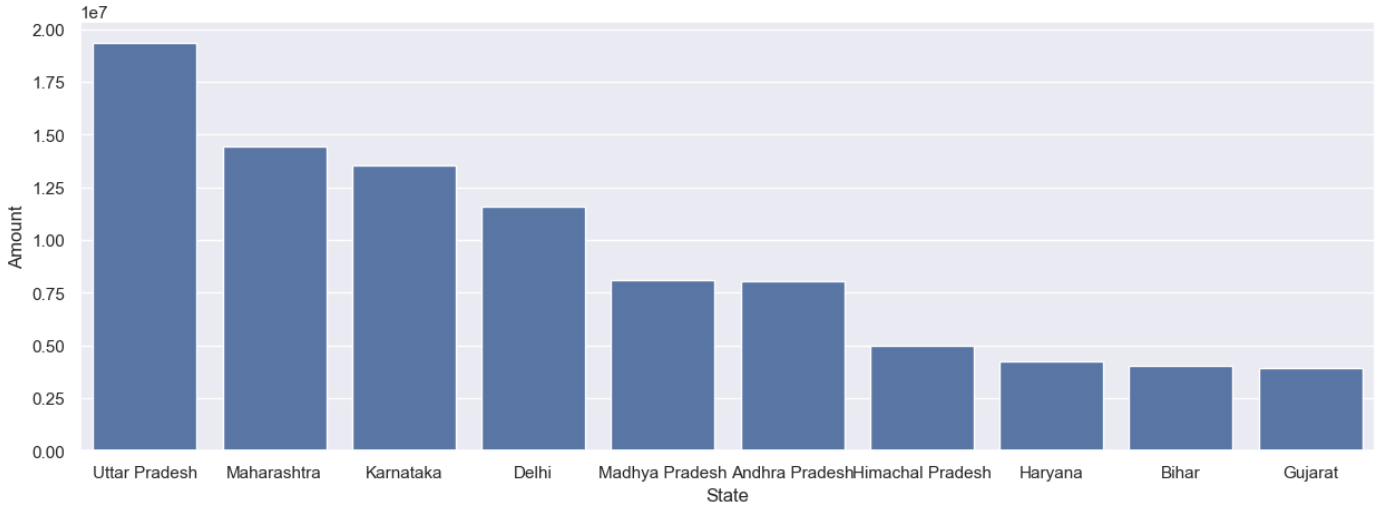
In [42]:
```
#total no.of orders from top 10 states
sales_state = df.groupby(['State'], as_index=False)['Orders'].sum().sort_values(by='Orde

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

Out[42]:
```
<Axes: xlabel='State', ylabel='Orders'>
```



In [37]:
```
#total amount/sales from top 10 states
sales_state = df.groupby(['State'], as_index=False)['Amount'].sum().sort_values(by = 'Am

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

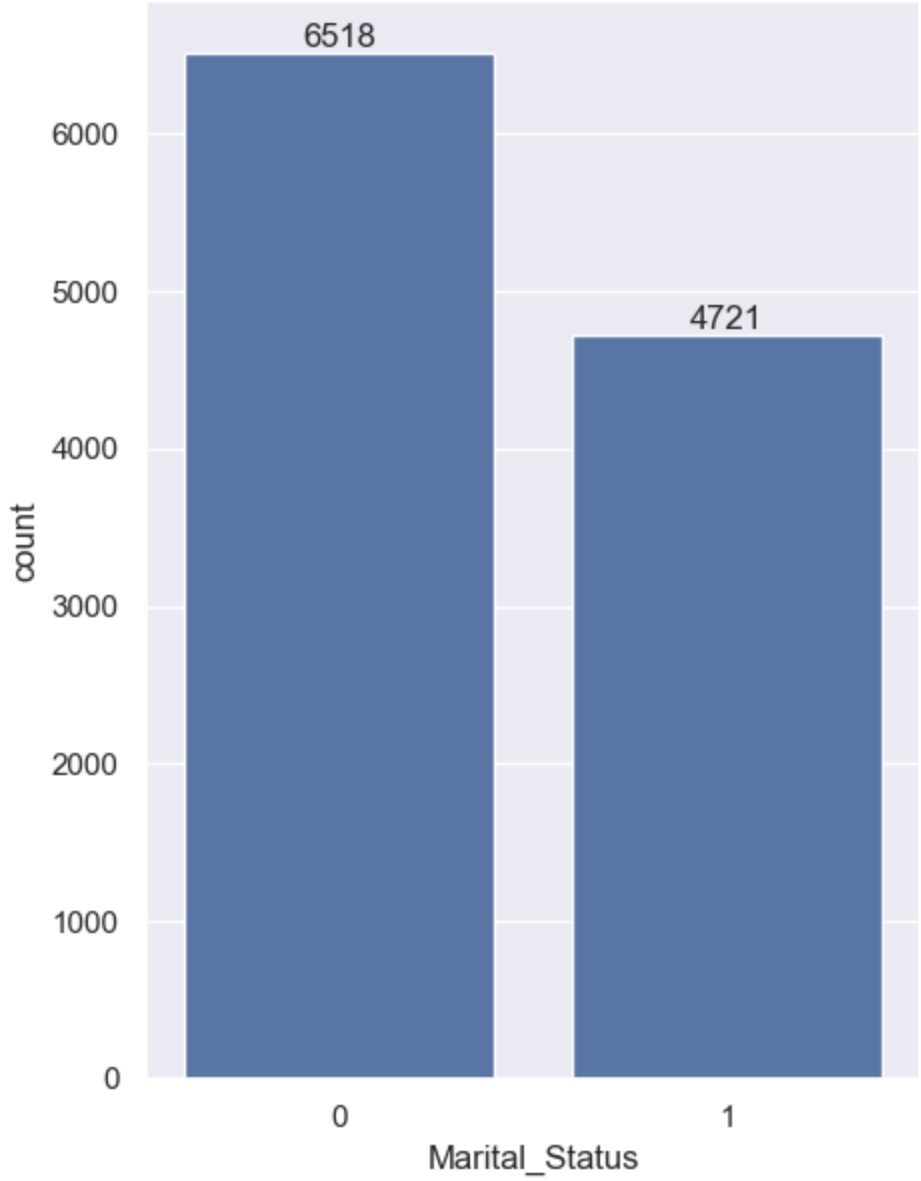
Out[37]:
```
<Axes: xlabel='State', ylabel='Amount'>
```



the above graph shows that most of the orders are from UP, Maharashtra and karnataka respectively but total sales/amount is from UP, karnataka and then maharashtra

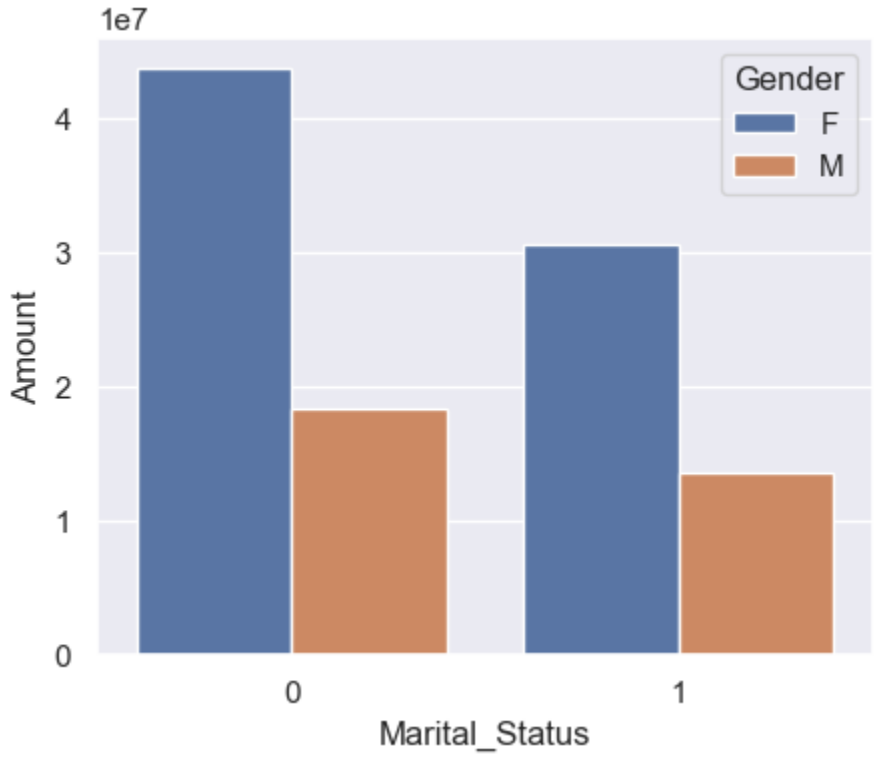
Marital Status

In [45]:
```
#here 0 is married and 1 is unmarried. shows which group shops most
ax = sns.countplot(data = df, x = 'Marital_Status')

sns.set(rc={'figure.figsize' : (5,4)})
for bars in ax.containers: ax.bar_label(bars)
```



In [49]:
```
#amount spend by married and unmarried people on shoping
sales_state = df.groupby(['Marital_Status', 'Gender'], as_index=False) ['Amount'].sum().

sns.set(rc={'figure.figsize' : (5,4)})
sns.barplot(data=sales_state, x= 'Marital_Status', y='Amount', hue='Gender')
```

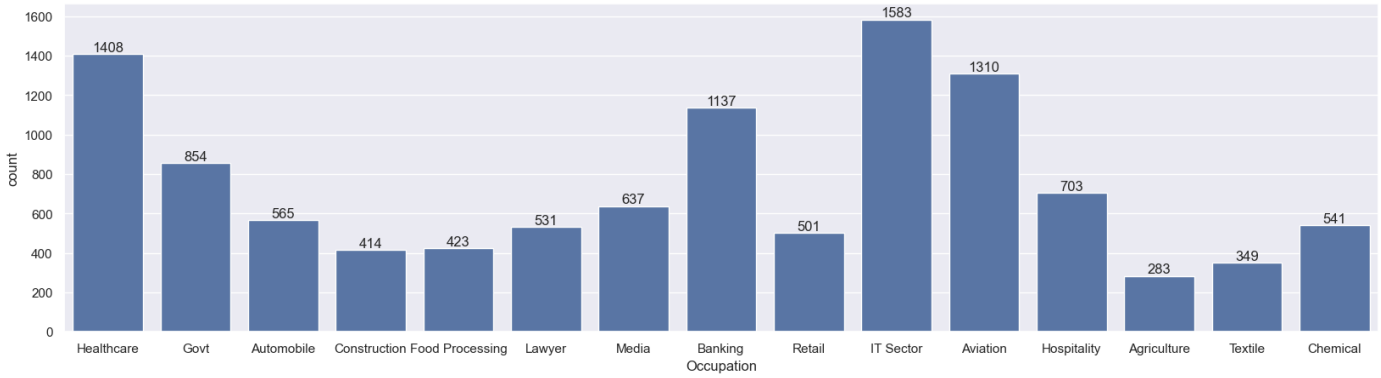
Out[49]:
```
<Axes: xlabel='Marital_Status', ylabel='Amount'>
```



This graph show that married people have bought most goods than those unmarried ones. Female has bought most goods in both categories than man.

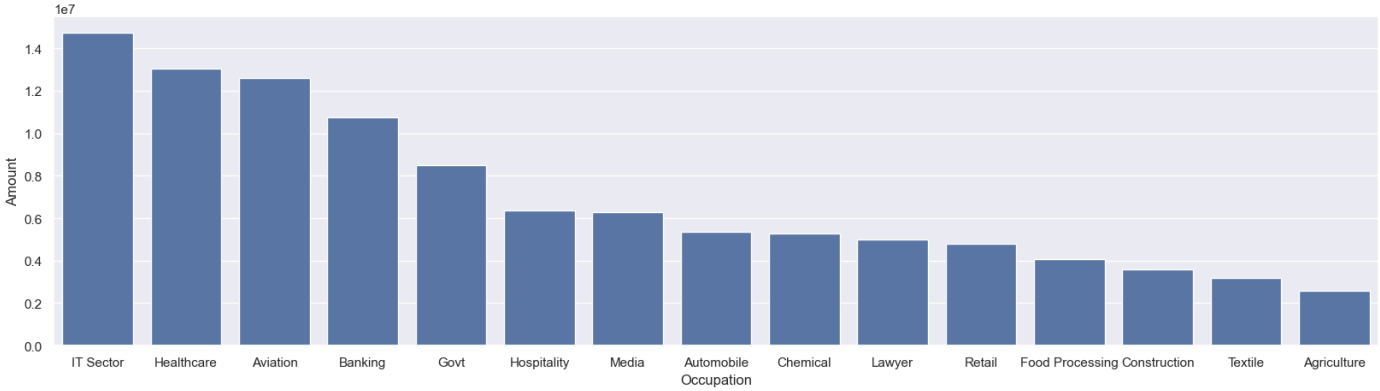
Occupation

In [58]:
```
#buyers or customers on the basis of their occupation
sns.set(rc={'figure.figsize' : (20,5)})
ax = sns.countplot(data = df, x= 'Occupation')

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



In [60]:
```
#purchasing power of customers on the basis of employment sector
sales_state = df.groupby(['Occupation'], as_index=False)['Amount'].sum().sort_values(by=

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

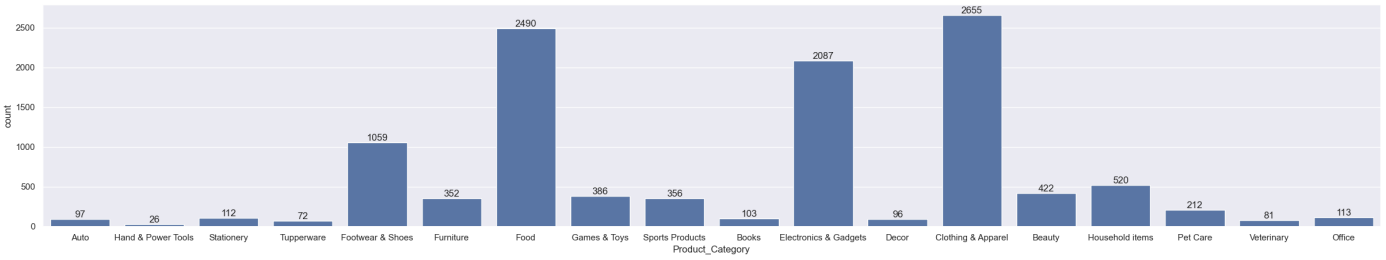
Out[60]:
```
<Axes: xlabel='Occupation', ylabel='Amount'>
```



Above graph shows the purchasing power sector-wise

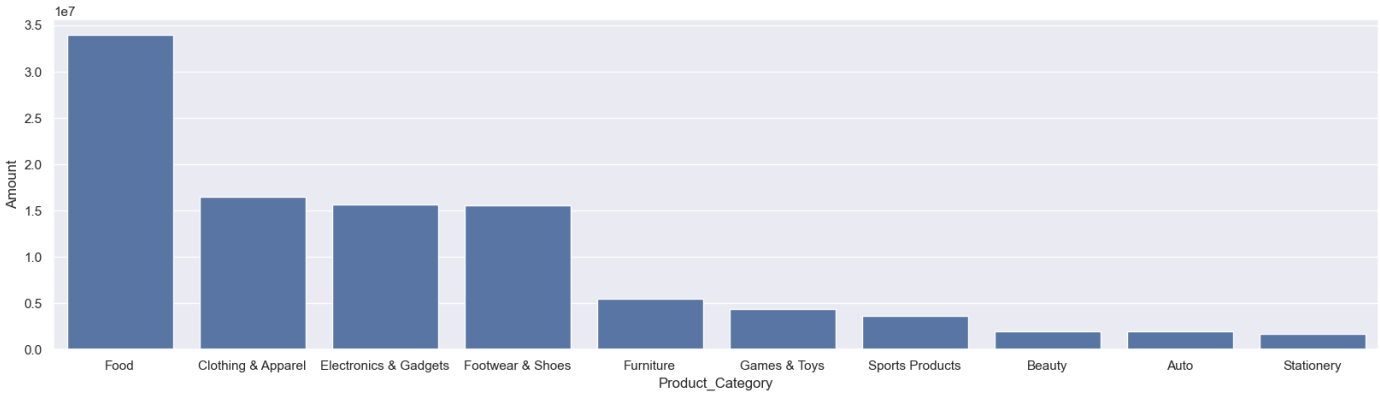
Product Category

In [66]:
```
#product prefered by the customers
sns.set(rc={'figure.figsize' : (30,5)})
ax = sns.countplot(data = df, x= 'Product_Category')

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



In [82]:
```
#amount spent on most prefered product cotegories
sales_state = df.groupby(['Product_Category'], as_index=False)['Amount'].sum().sort_valu

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

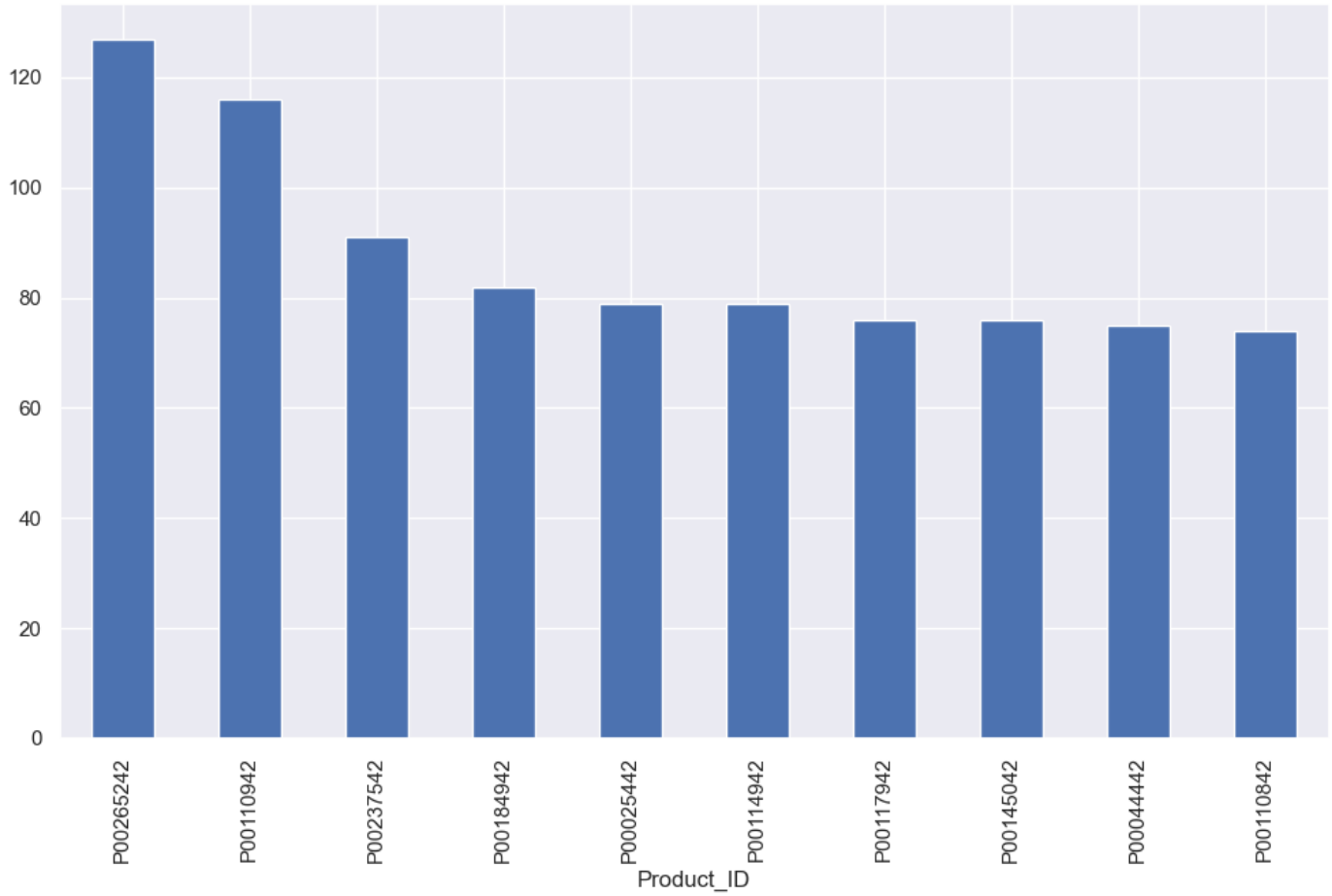
Out[82]:
```
<Axes: xlabel='Product_Category', ylabel='Amount'>
```



The graph shows that most famous product category is food followed by clothing and elecronics

In [85]:
```
#top 10 most sold product by product-id

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

Out[85]:
```
<Axes: xlabel='Product_ID'>
```



Conclusion

Married Womens age between 26-36 years have bought the most products. These purchases mostly comes from UP, Maharashtra and Karnataka State. Occupation-wise IT sector tops the list followed by Health Care and Aviation Sector. Most Famous Categories are Food followerd by Clothing and Electronics.