In [6]:
```
import numpy as np # used to help on arrays
import pandas as pd # used to help on dataframe
import matplotlib.pyplot as plt # used as charts and visualization
%matplotlib inline
```

In [7]:
```
df=pd.read_csv(r'C:\Users\win10\Videos\Python_Diwali_Sales_Analysis-main\Py
# to avoid encoding error use 'unicode_escape'
```

In [8]:

Out[8]: (11251, 15)

In [9]:

Out[9]:

	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	So
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	(
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	So
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	W

In [10]:

Out[10]:

	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	So
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	(
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	So
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	W
5	1000588	Joni	P00057942	M	26-35	28	1	Himachal Pradesh	Nc
6	1001132	Balk	P00018042	F	18-25	25	1	Uttar Pradesh	(
7	1002092	Shivangi	P00273442	F	55+	61	0	Maharashtra	W
8	1003224	Kushal	P00205642	M	26-35	35	0	Uttar Pradesh	(
9	1003650	Ginny	P00031142	F	26-35	26	1	Andhra Pradesh	So

In [11]:

```
<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 [13]:

In [14]:

```
<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 [15]:

Out[15]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone
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 [16]:

Out[16]:
```
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 [17]:

In [18]:

Out[18]: (11239, 13)

In [19]:

Out[19]:
```
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 [20]:
```
data_test=[['madhav',11],['Gopi',15],['Keshav',], ['Lalita',16]] #initializ
df_test=pd.DataFrame(data_test, columns=['Name','Age']) # create the pandas
```

Out[20]:

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

In [23]:

In [24]:

Out[24]:

	Name	Age
0	madhav	11.0
1	Gopi	15.0
3	Lalita	16.0

In [27]:

In [28]:

Out[28]:

	Name	Age
0	madhav	11.0
1	Gopi	15.0
3	Lalita	16.0

In [31]:

In [32]:

Out[32]:
```
dtype('int32')
```

In [33]:

Out[33]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Categor
y',
       'Orders', 'Amount'],
      dtype='object')
```

In [37]:
```
df.rename(columns={'Marital_Status':'Shaadi'}) # rename the column.
```

Out[37]:

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

11239 rows × 13 columns

In [38]:

Out[38]:

	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 [40]:

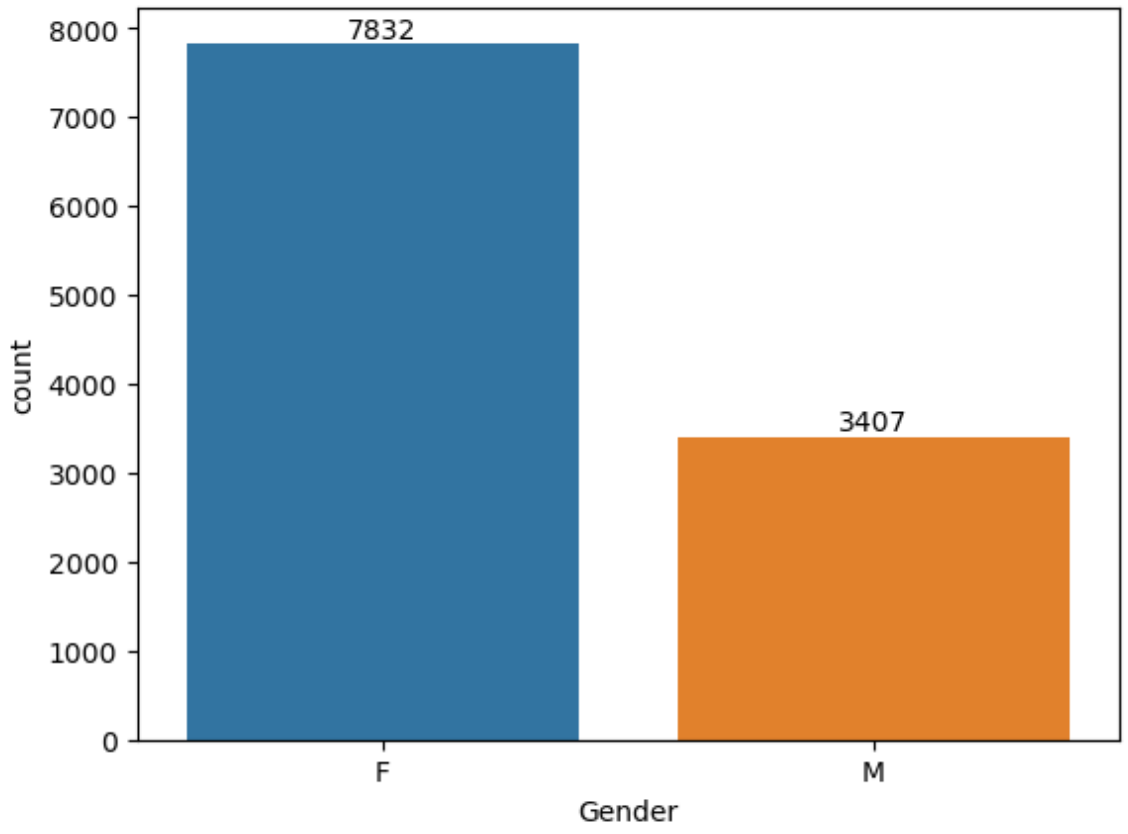
Out[40]:

	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 [41]:
```
ax=sns.countplot(x='Gender',data=df) # ut is used to see the only label not
for bars in ax.containers: # for bars in ax.containers: # ax.bar_label(bars,
```



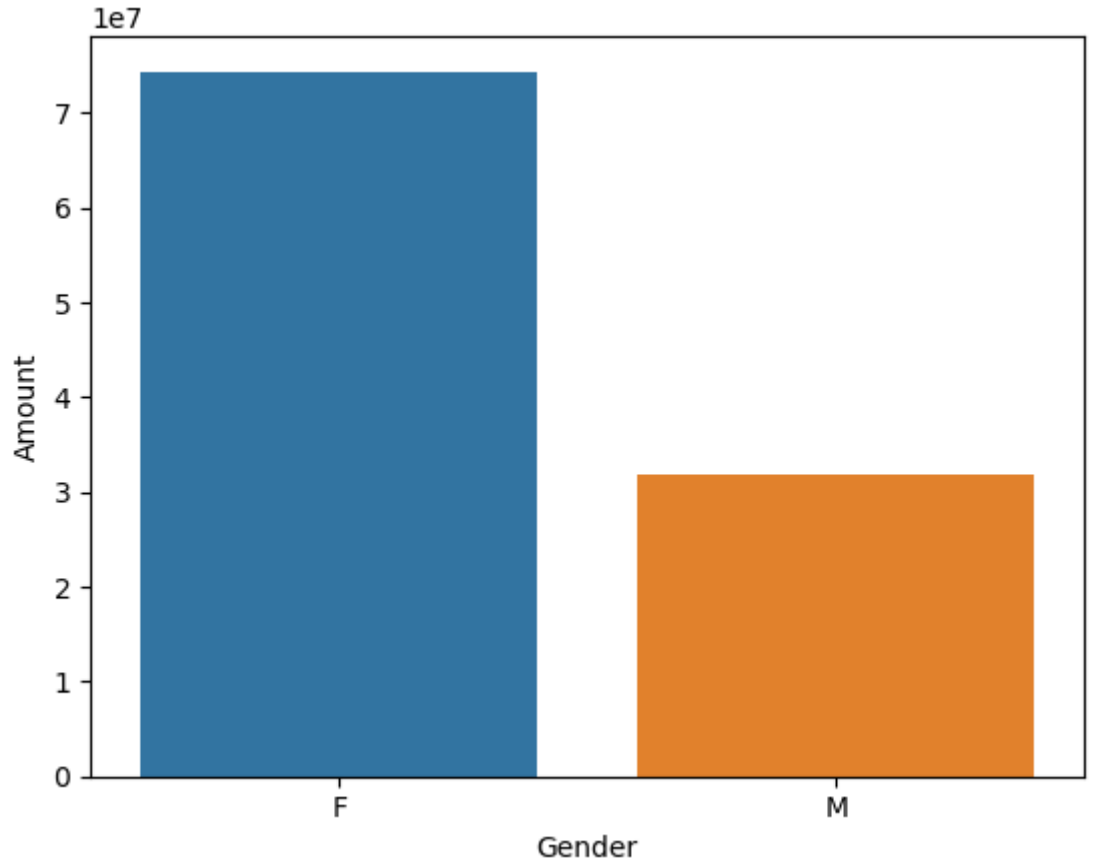
In [44]:

Out[44]:

	Gender	Amount
0	F	74335853
1	M	31913276

In [45]:
```
sales_gen=df.groupby(['Gender'], as_index=False)['Amount'].sum().sort_value
```

Out[45]: <Axes: xlabel='Gender', ylabel='Amount'>

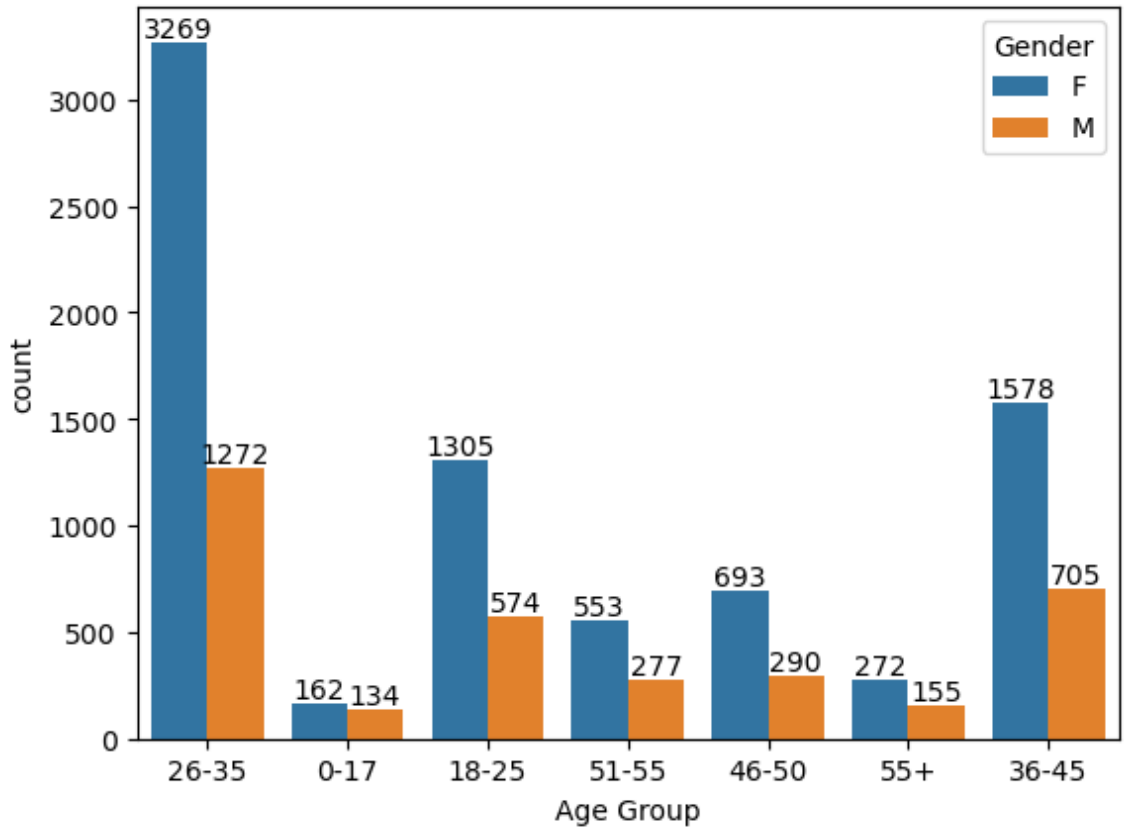


Age

In [46]:

Out[46]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Categor
y',
       'Orders', 'Amount'],
      dtype='object')
```

In [47]:
```
ax=sns.countplot(x='Age Group',hue='Gender',data=df) # hue is used to indic
for bars in ax.containers: # for bars in ax.containers: # ax.bar_label(bars
```



In [ ]:
```
# Total amount vs age group
sales_age=df.groupby(['Age Group'], as_index=False)['Amount'].sum().sort_va
```

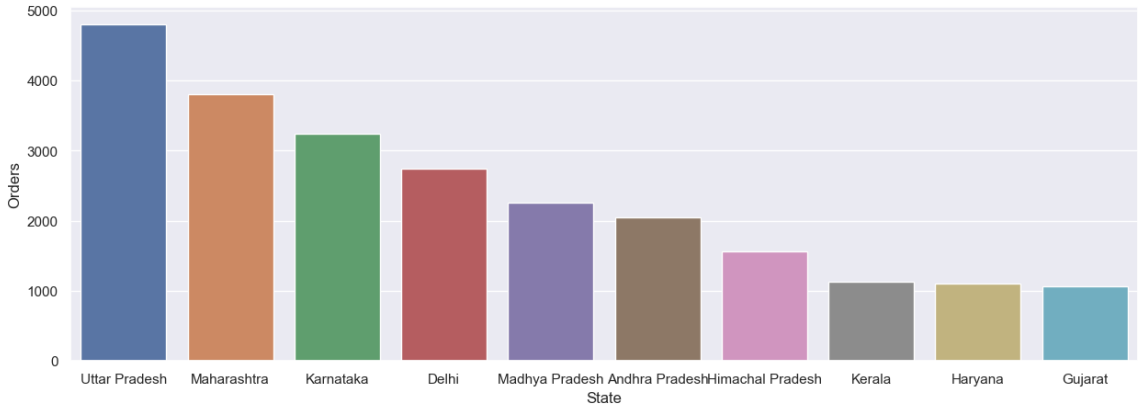
state

In [49]:

Out[49]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Categor
y',
       'Orders', 'Amount'],
      dtype='object')
```

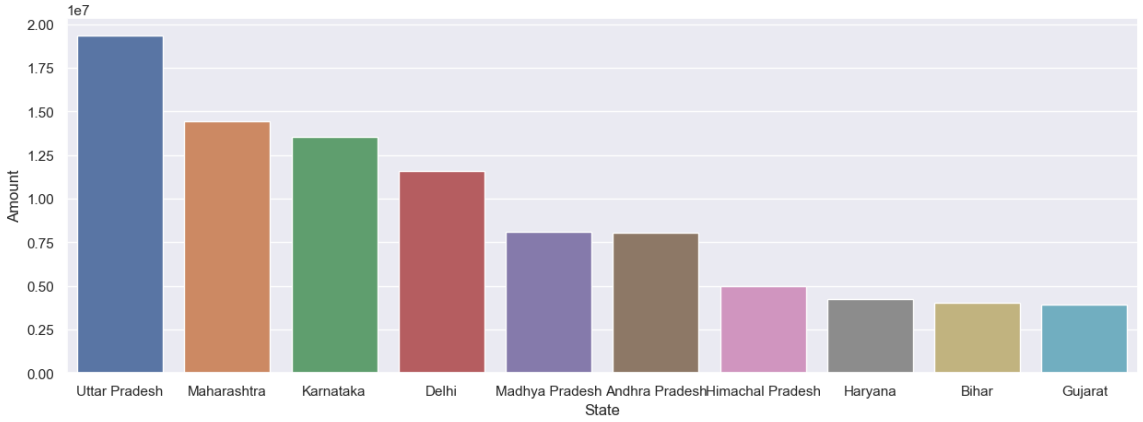
In [53]:
```
# total number of orders of top 10 states
sales_state=df.groupby(['State'],as_index=False)['Orders'].sum().sort_value
sns.set(rc={'figure.figsize':(15,5)})
```

Out[53]: <Axes: xlabel='State', ylabel='Orders'>



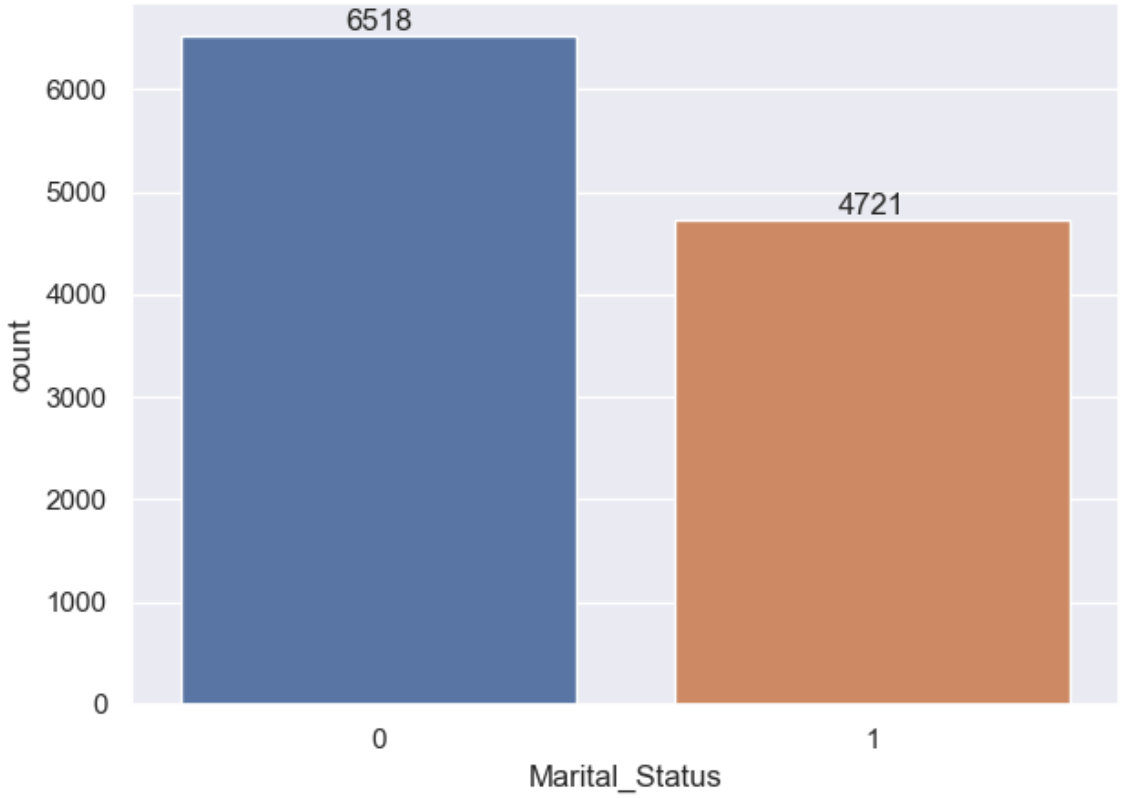
In [54]:
```
# total amount/sales from top 10 states
sales_state=df.groupby(['State'],as_index=False)['Amount'].sum().sort_value
sns.set(rc={'figure.figsize':(15,5)})
```

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



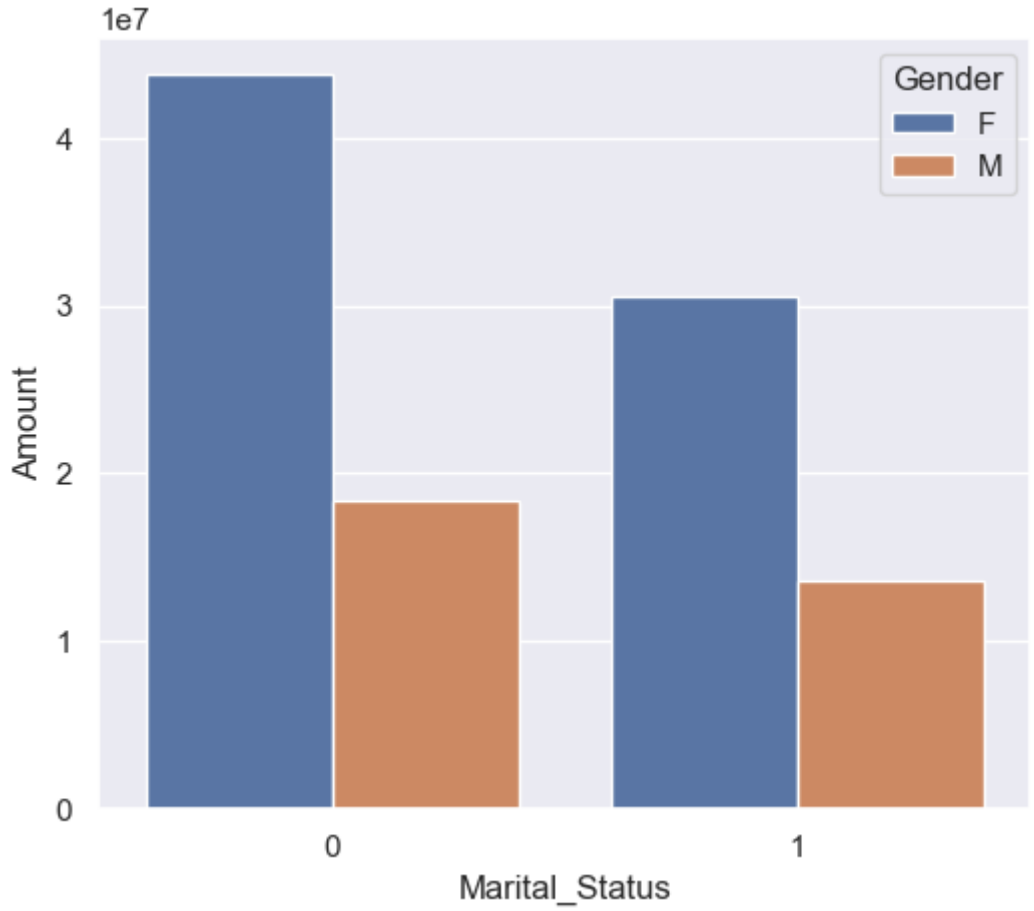
Marital status

In [59]:
```
ax=sns.countplot(x='Marital_Status',data=df)
sns.set(rc={'figure.figsize':(7,5)})
for bars in ax.containers: # for bars in ax.containers: # ax.bar_label(bars,
```



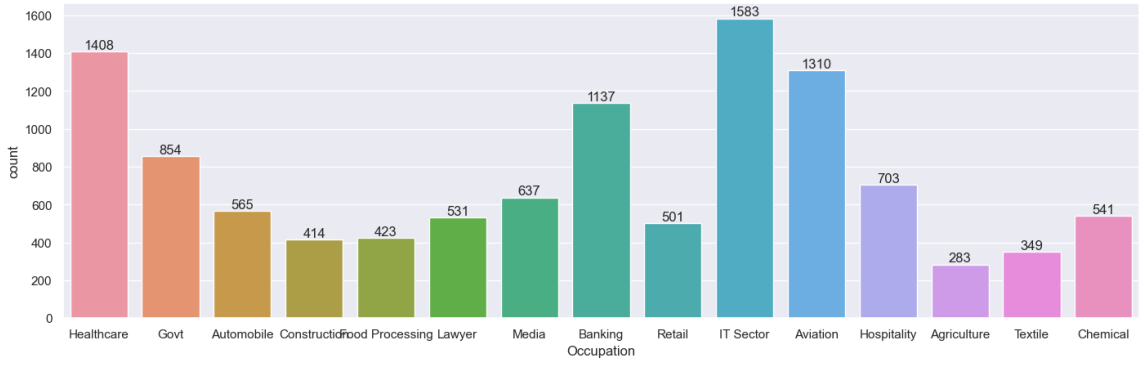
In [60]:
```
sales_state=df.groupby(['Marital_Status','Gender'],as_index=False)['Amount'
sns.set(rc={'figure.figsize':(6,5)})
sns.barplot(data=sales_state,x='Marital_Status',y='Amount',hue='Gender')
```

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



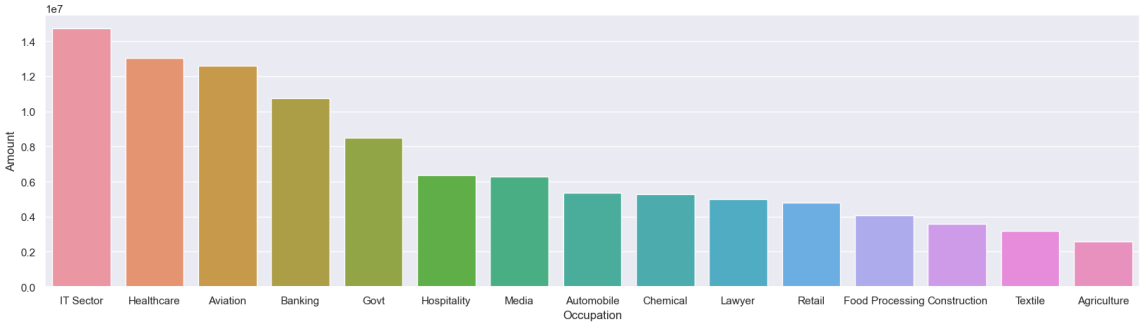
Occupation

In [63]:
```
sns.set(rc={'figure.figsize':(17,5)})
ax=sns.countplot(x='Occupation',data=df)
for bars in ax.containers: # for bars in ax.containers: # ax.bar_label(bars,
```



In [66]:
```
sales_state=df.groupby(['Occupation'],as_index=False)['Amount'].sum().sort_
sns.set(rc={'figure.figsize':(20,5)})
```

Out[66]: <Axes: xlabel='Occupation', ylabel='Amount'>

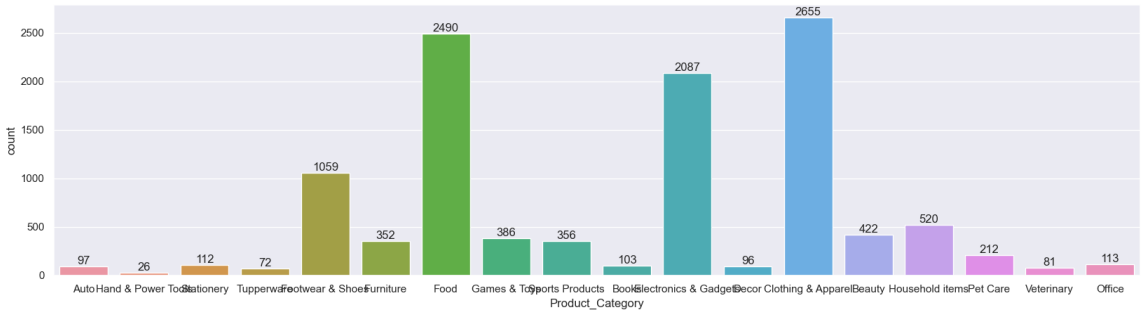


Product category

In [67]:
```
```

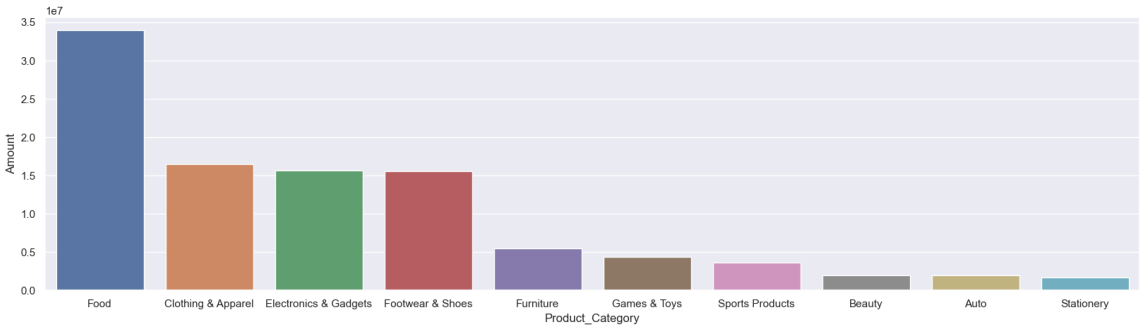
Out[67]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Categor
y',
       'Orders', 'Amount'],
      dtype='object')
```

In [70]:
```
sns.set(rc={'figure.figsize':(20,5)})
ax=sns.countplot(x='Product_Category',data=df)
for bars in ax.containers: # for bars in ax.containers: # ax.bar_label(bars,
```



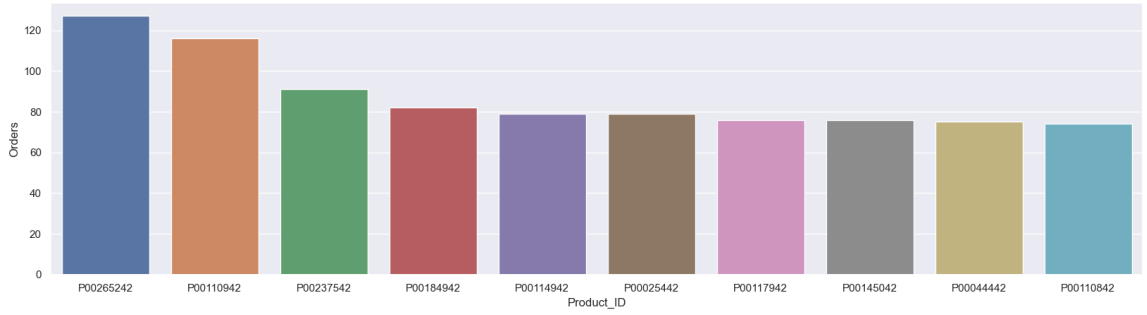
In [72]:
```
sales_state=df.groupby(['Product_Category'],as_index=False)['Amount'].sum()
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data=sales_state,x='Product_Category',y='Amount')
```

Out[72]: <Axes: xlabel='Product_Category', ylabel='Amount'>



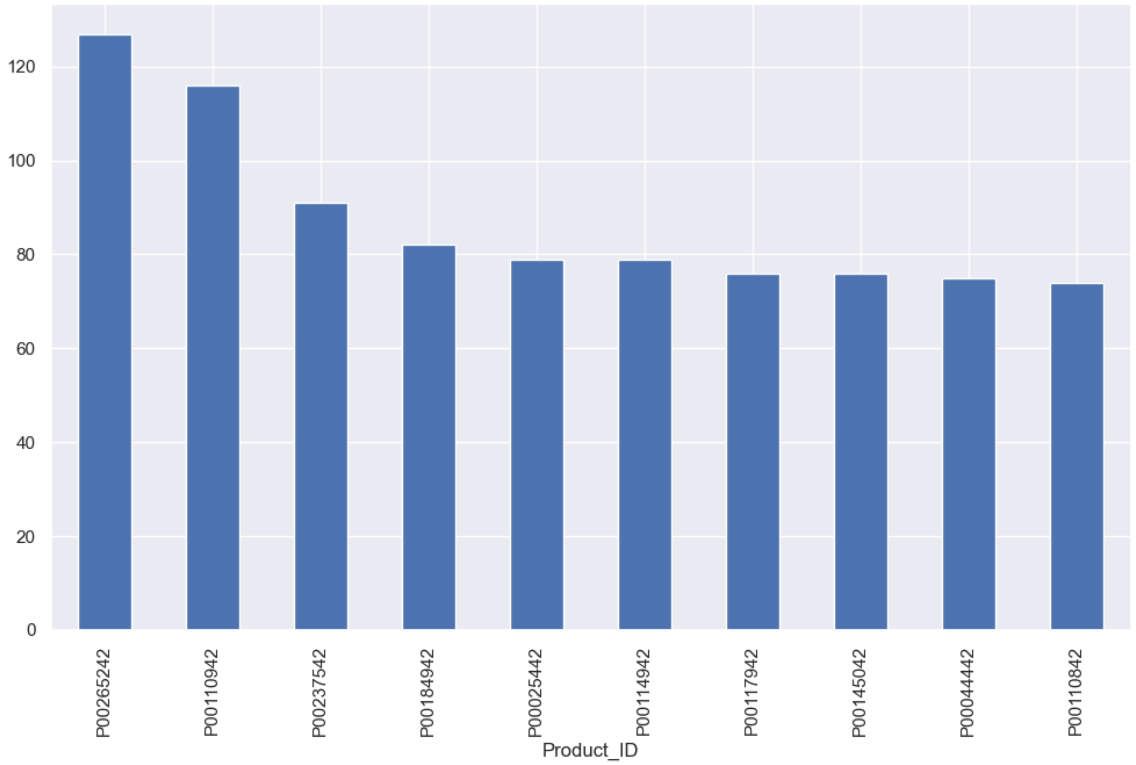
In [73]:
```
sales_state=df.groupby(['Product_ID'],as_index=False)['Orders'].sum().sort_
sns.set(rc={'figure.figsize':(20,5)})
```

Out[73]: <Axes: xlabel='Product_ID', ylabel='Orders'>



In [74]:
```
# top 10 most sold products ( smething as above)
fig1,ax1=plt.subplots(figsize=(12,7))
```

Out[74]: <Axes: 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.