data-analysis-with-python

January 11, 2024

1 Data Analysis With Python

[5]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

[108]:
```
df = pd.read_csv(r'C:
↪\Users\saksh\Downloads\Python_Diwali_Sales_Analysis\Python_Diwali_Sales_Analysis\Diwali␣
↪Sales Data.csv',encoding= 'unicode_escape')
```

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

[12]:
```
(11251, 15)
```

[14]:
```
df.head(10)
```

[14]:

	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	
5	1000588	Joni	P00057942	M	26-35	28	1	
6	1001132	Balk	P00018042	F	18-25	25	1	
7	1002092	Shivangi	P00273442	F	55+	61	0	
8	1003224	Kushal	P00205642	M	26-35	35	0	
9	1003650	Ginny	P00031142	F	26-35	26	1	

	State	Zone	Occupation	Product_Category	Orders	\
0	Maharashtra	Western	Healthcare	Auto	1	
1	Andhra Pradesh	Southern	Govt	Auto	3	
2	Uttar Pradesh	Central	Automobile	Auto	3	
3	Karnataka	Southern	Construction	Auto	2	
4	Gujarat	Western	Food Processing	Auto	2	
5	Himachal Pradesh	Northern	Food Processing	Auto	1	

```
6     Uttar Pradesh   Central           Lawyer             Auto       4
7       Maharashtra   Western        IT Sector             Auto       1
8     Uttar Pradesh   Central             Govt             Auto       2
9    Andhra Pradesh  Southern             Media             Auto       4

     Amount  Status  unnamed1
0  23952.00     NaN       NaN
1  23934.00     NaN       NaN
2  23924.00     NaN       NaN
3  23912.00     NaN       NaN
4  23877.00     NaN       NaN
5  23877.00     NaN       NaN
6  23841.00     NaN       NaN
7       NaN     NaN       NaN
8  23809.00     NaN       NaN
9  23799.99     NaN       NaN
```

[15]:
```
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
```

[16]:
```
# Drop unrelated and Blank columns
df.drop(['Status','unnamed1'],axis=1,inplace=True)
```

[17]:
```
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
```

[18]:
```
# Check the null value
df.isnull()
```

[18]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	\
0	False	False	False	False	False	False	
1	False	False	False	False	False	False	
2	False	False	False	False	False	False	
3	False	False	False	False	False	False	
4	False	False	False	False	False	False	
…	…	…	…	…	…	…	
11246	False	False	False	False	False	False	
11247	False	False	False	False	False	False	
11248	False	False	False	False	False	False	
11249	False	False	False	False	False	False	
11250	False	False	False	False	False	False	

	Marital_Status	State	Zone	Occupation	Product_Category	Orders	\
0	False	False	False	False	False	False	
1	False	False	False	False	False	False	
2	False	False	False	False	False	False	
3	False	False	False	False	False	False	
4	False	False	False	False	False	False	
…	…	…	…	…	…	…	
11246	False	False	False	False	False	False	
11247	False	False	False	False	False	False	
11248	False	False	False	False	False	False	

```
11249          False  False  False       False             False   False
11250          False  False  False       False             False   False

       Amount
0       False
1       False
2       False
3       False
4       False
…       …
11246   False
11247   False
11248   False
11249   False
11250   False

[11251 rows x 13 columns]
```

[19]:
```
# use the sum() function for shaw  total of  number of null values
df.isnull().sum()
```

[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             12
dtype: int64
```

[20]:
```
#Drop the null values that is Amount
df.dropna(inplace=True)
```

[21]:
```
# Again check the null values
df.isnull().sum()
```

[21]:
```
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
```

[22]:
```
# Change the data types
df['Amount']=df['Amount'].astype('int')
```

[24]:
```
# Check data type of "Amount"
df['Amount'].dtype
```

[24]:
```
dtype('int32')
```

[26]:
```
# Show the columns name
df.columns
```

[26]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

[34]:
```
# Rename columns name
df.rename(columns ={'Cust_name':'Customer_Name'},inplace)
```

[99]:
```
df.head()
```

[99]:

	User_ID	Customer_Name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877

Describe() function return description of the data in the DataFrame(i.e. count,mean,std,percentile,etc)

[98]:
```
df.describe()
```

[98]:

	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

[31]:
```
# Use describe on specific columns
df[['Age','Orders','Amount']].describe()
```

[31]:

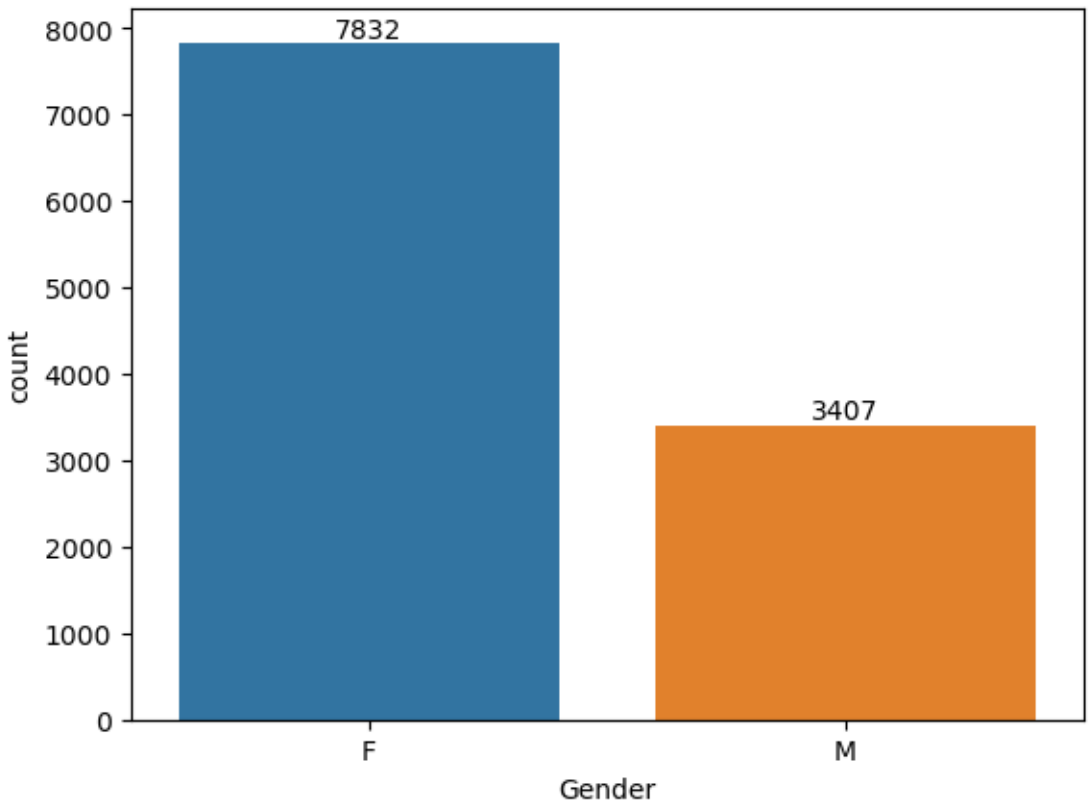
	Age	Orders	Amount
count	11239.000000	11239.000000	11239.000000
mean	35.410357	2.489634	9453.610553
std	12.753866	1.114967	5222.355168
min	12.000000	1.000000	188.000000
25%	27.000000	2.000000	5443.000000
50%	33.000000	2.000000	8109.000000
75%	43.000000	3.000000	12675.000000
max	92.000000	4.000000	23952.000000

2 Exploratory Data Analysis

[36]:
```
df.columns
```

[36]:
```
Index(['User_ID', 'Customer_Name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

[42]:
```
ax = sns.countplot(x ='Gender',data = df)
for bars in ax.containers:
    ax.bar_label(bars)
```



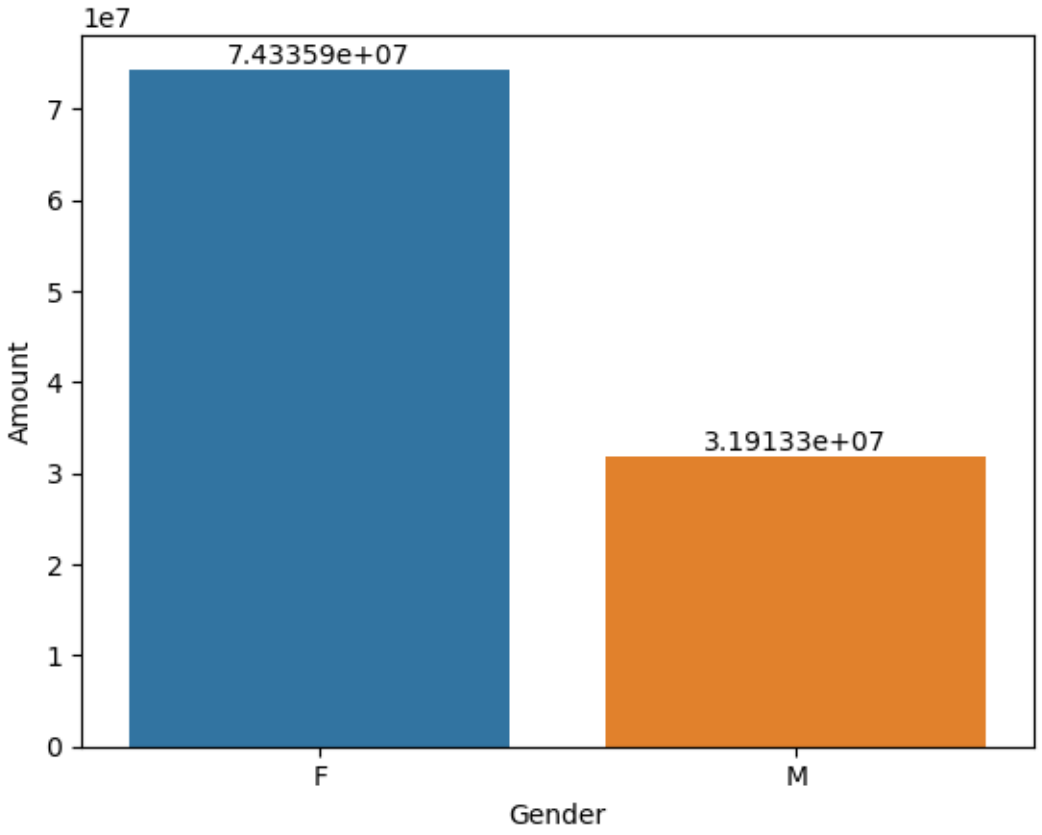
[46]:
```
sales_gen = df.groupby(['Gender'],as_index=False)['Amount'].sum().
 ↪sort_values(by='Amount',ascending=False)
```

[48]:
```
pd.DataFrame(sales_gen)
```

[48]:

	Gender	Amount
0	F	74335853
1	M	31913276

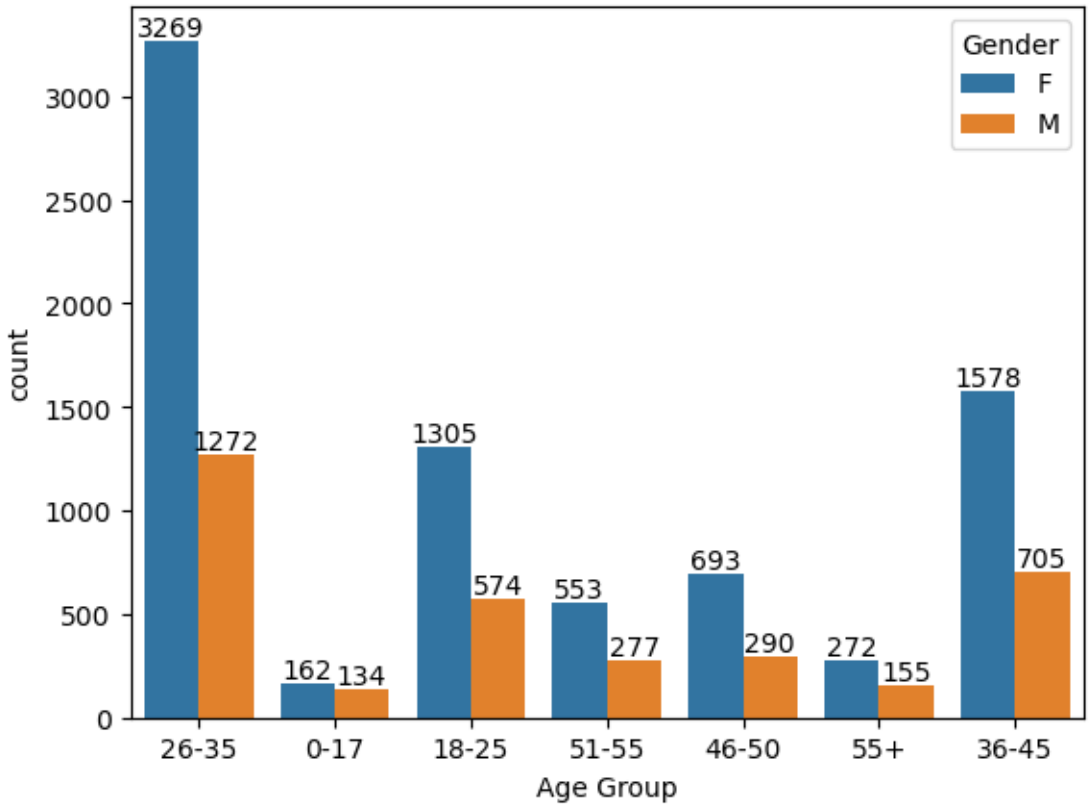
[70]:
```
# sales_gen = df.groupby(['Gender'],as_index=False)['Amount'].sum().
 ↪sort_values(by='Amount',ascending=False)
ax = sns.barplot(x='Gender',y='Amount',data=sales_gen)
for bars in ax.containers:
    ax.bar_label(bars)
```



Above barplot show that female total sale amount is greater than male so mean is that female buys more product in diwali.

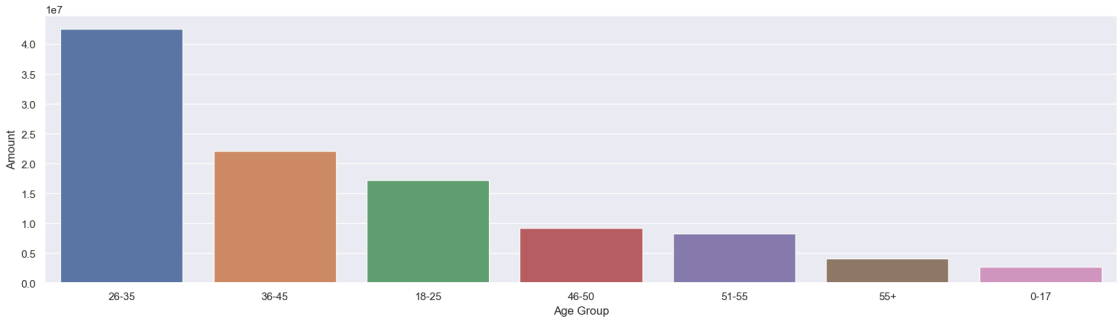
3 Age

[65]:
```
ax = sns.countplot(data = df,x = 'Age Group',hue = 'Gender')
for bars in ax.containers:
    ax.bar_label(bars)
```



[110]:
```
# TotalAmount Vs AgeGroups
sales_ages = df.groupby(['Age Group'],as_index = False)['Amount'].sum().
  ↪sort_values(by = 'Amount',ascending = False)
sns.barplot(x ='Age Group',y ='Amount',data = sales_ages)
```

[110]:
```
<Axes: xlabel='Age Group', ylabel='Amount'>
```

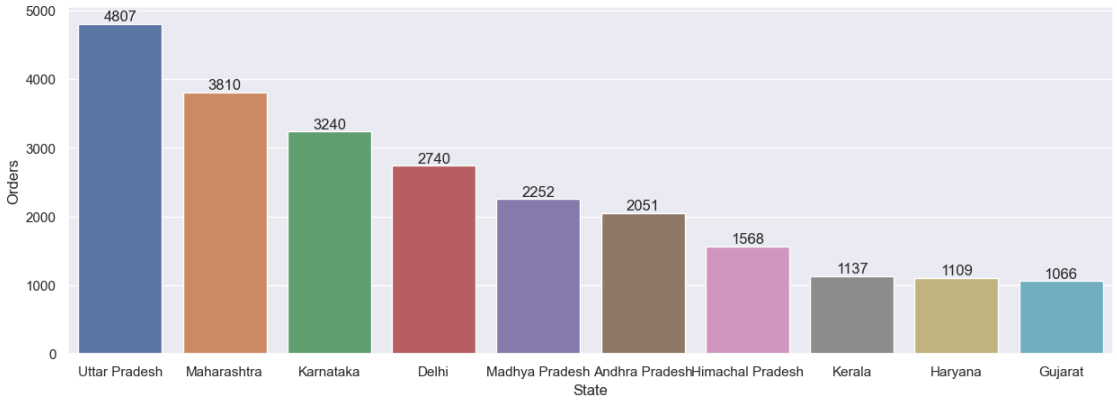


Above graph show age group of 26-35 is spending money remain rest age group.

4 State

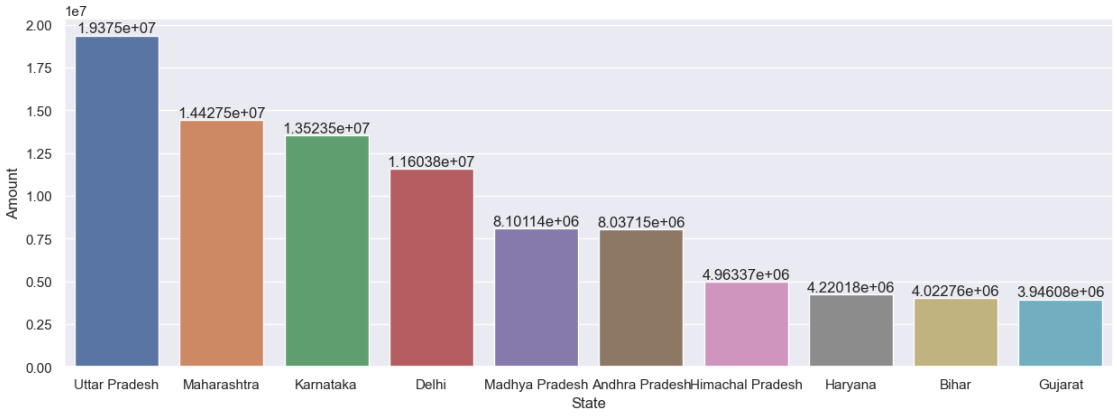
[75]:
```
# Total Number Of Orders from Top 10 States

Sales_state = df.groupby(['State'],as_index = False)['Orders'].sum().
 ↪sort_values(by = 'Orders',ascending = False).head(10)
sns.set(rc={'figure.figsize':(15,5)})
ax = sns.barplot(x ='State',y = 'Orders',data = Sales_state)
for bars in ax.containers:
    ax.bar_label(bars)
```



This barplot tell us that count of number of order on the basis of State,Uttar Pradesh gives highest order among all top 10 states.

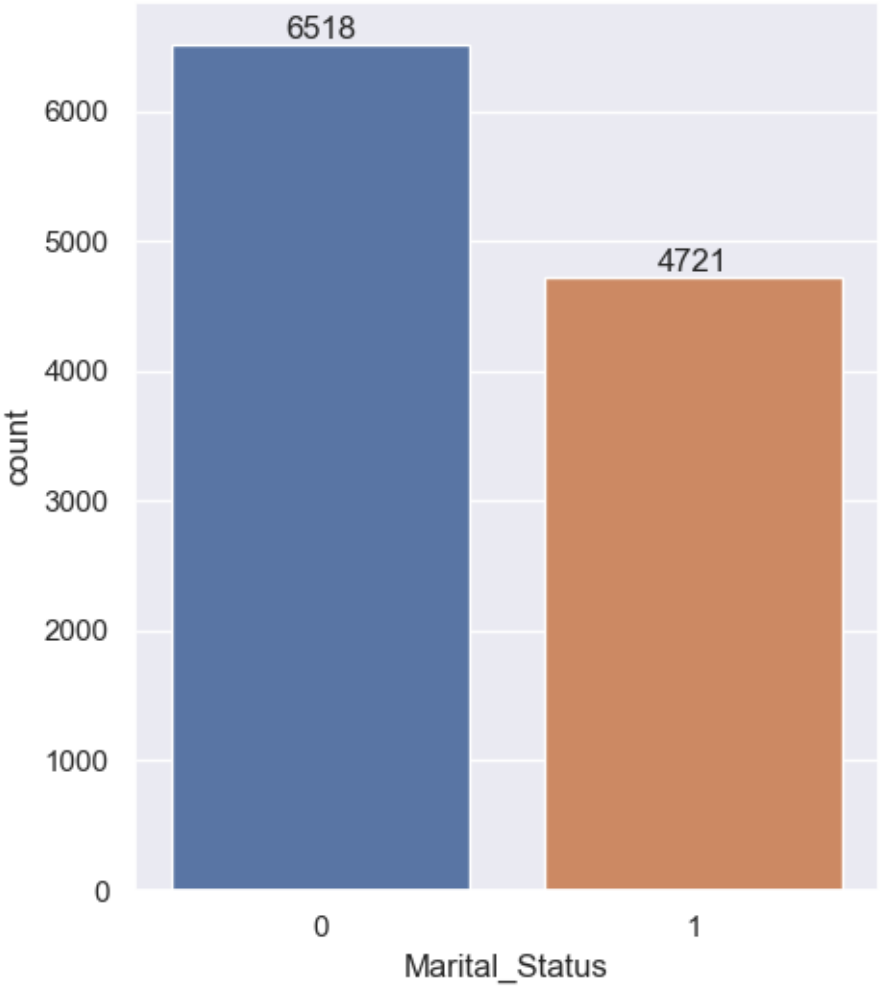
[77]:
```
Sales_state = df.groupby(['State'],as_index = False)['Amount'].sum().
 ↪sort_values(by = 'Amount',ascending = False).head(10)
sns.set(rc={'figure.figsize':(15,5)})
ax = sns.barplot(x ='State',y = 'Amount',data = Sales_state)
for bars in ax.containers:
    ax.bar_label(bars)
```



[ ]:
```
# Uttar Pradesh spend largest number of amount  because of orders.
```

5 Marital Status

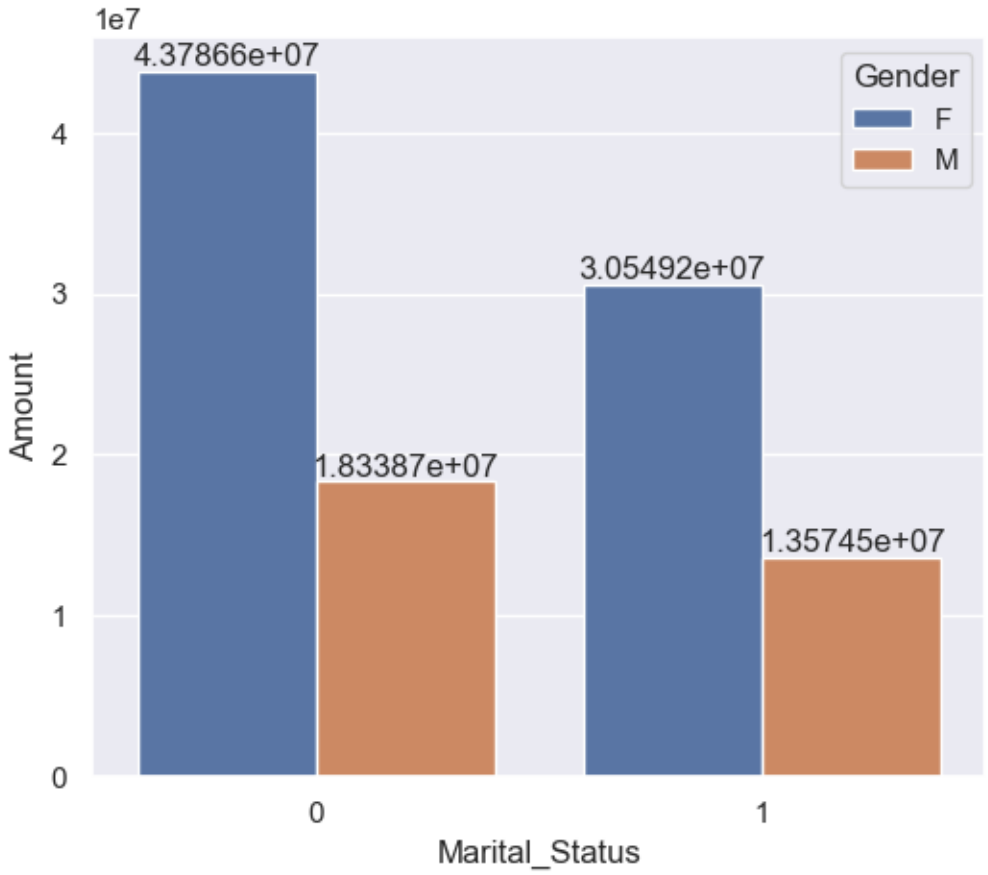
[93]:
```
ax = sns.countplot( x = 'Marital_Status', data = df)
sns.set(rc={'figure.figsize':(5,6)})
for bars in ax.containers:
    ax.bar_label(bars)
```



Here 0 describe the No. of non marital status and 1 describe no. of the marital status

[90]:
```
Sales_MGA = df.groupby(['Marital_Status','Gender'],as_index = False)['Amount'].
 ↪sum().sort_values(by = 'Amount',ascending = False)
sns.set(rc={'figure.figsize':(6,5)})
ax = sns.barplot(x = 'Marital_Status',y = 'Amount',hue = 'Gender',data =␣
 ↪Sales_MGA)
for bars in ax.containers:
```

```
    ax.bar_label(bars)
```



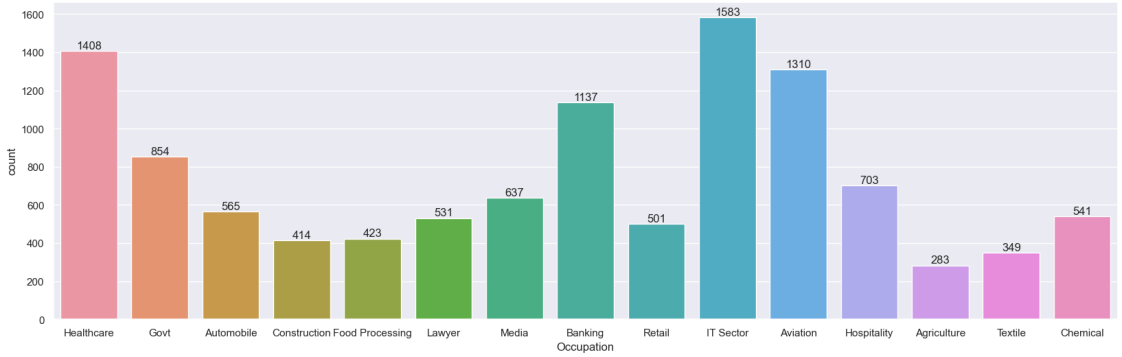
Here show total amount of marital and non marital which is in divide by male an female and we can see that non marital male and female spend more money than marital male female.

6 Occupation

[100]:
```
df.columns
```

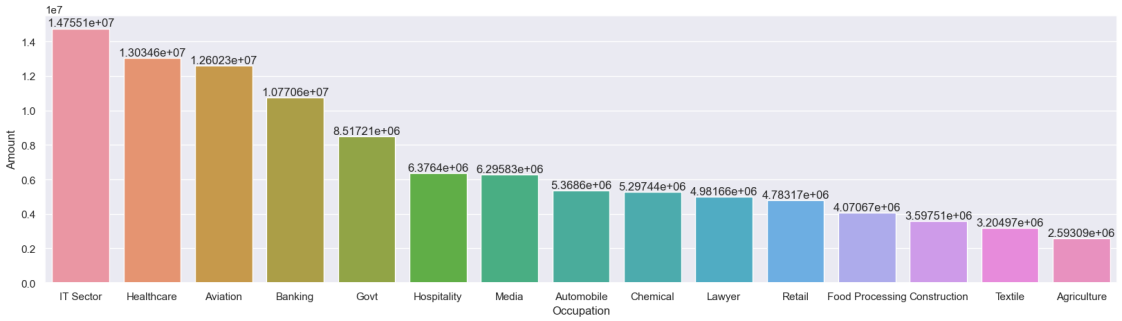
[100]:
```
Index(['User_ID', 'Customer_Name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

[101]:
```
sns.set(rc={'figure.figsize':(20,6)})
ax = sns.countplot(data = df,x = 'Occupation')
for bars in ax.containers:
    ax.bar_label(bars)
```



In this analysis we found that IT Sector User are active in Shopping

[107]:
```
sales_occu = df.groupby(['Occupation'],as_index = False)['Amount'].sum().
 ↪sort_values(by = 'Amount',ascending = False)
sns.set(rc={'figure.figsize':(20,5)})
ax = sns.barplot(x = 'Occupation',y = 'Amount',data = sales_occu)
for bars in ax.containers:
    ax.bar_label(bars)
```



In this graph we found that total amount of IT sector User which is spend in shopping and second largest amount of Healthcare user and more.

7 We found in entire data analysis that Non Marital Female and male who live in Uttar Pradesh and age between 26-35 and work in IT sector are Spends more Money in Shopping Comparison to others.

THANK YOU