fbnj94530

December 15, 2024

[3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

[24]:
```
df = pd.read_csv(r'C:\Users\Asus\OneDrive\Desktop\EDA Project\Diwali Sales␣
 ↪Analysis\Diwali Sales Data.csv', encoding ='unicode_escape')
df
```

[24]:
```
       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
…          …           …          …     …         …  …               …
11246  1000695      Manning  P00296942      M     18-25   19               1
11247  1004089  Reichenbach  P00171342      M     26-35   33               0
11248  1001209        Oshin  P00201342      F     36-45   40               0
11249  1004023       Noonan  P00059442      M     36-45   37               0
11250  1002744      Brumley  P00281742      F     18-25   19               0

                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
…                 …         …               …               …      …
11246     Maharashtra   Western         Chemical           Office       4
11247         Haryana  Northern       Healthcare       Veterinary       3
11248  Madhya Pradesh   Central          Textile           Office       4
11249       Karnataka  Southern      Agriculture           Office       3
11250     Maharashtra   Western       Healthcare           Office       3

        Amount  Status  unnamed1
0      23952.0     NaN       NaN
```

```
1       23934.0     NaN       NaN
2       23924.0     NaN       NaN
3       23912.0     NaN       NaN
4       23877.0     NaN       NaN
…         …      …        …
11246     370.0     NaN       NaN
11247     367.0     NaN       NaN
11248     213.0     NaN       NaN
11249     206.0     NaN       NaN
11250     188.0     NaN       NaN

[11251 rows x 15 columns]
```

1 To inspect the data

[25]:

```
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
```

2 Droping unrelated/blank columns

[26]:

```
df.drop(['Status', 'unnamed1'], axis = 1, inplace = True)
df
```

[26]:

```
       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
…          …          …         …     …         …   …               …
11246  1000695      Manning  P00296942      M     18-25   19               1
11247  1004089  Reichenbach  P00171342      M     26-35   33               0
11248  1001209        Oshin  P00201342      F     36-45   40               0
11249  1004023       Noonan  P00059442      M     36-45   37               0
11250  1002744      Brumley  P00281742      F     18-25   19               0

                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
…                 …       …               …                …      …
11246     Maharashtra   Western         Chemical           Office       4
11247         Haryana  Northern       Healthcare       Veterinary       3
11248  Madhya Pradesh   Central          Textile           Office       4
11249       Karnataka  Southern      Agriculture           Office       3
11250     Maharashtra   Western       Healthcare           Office       3

        Amount
0      23952.0
1      23934.0
2      23924.0
3      23912.0
4      23877.0
…          …
11246    370.0
11247    367.0
11248    213.0
11249    206.0
11250    188.0

[11251 rows x 13 columns]
```

#We have dropped the Status and unnamed1 columns as they were containing null values

3 Checking for null values

```
[28]: df.isnull().sum()
```

```
[28]: 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
```

#We can see that Amount has 12 null values

4 Dropping null values

```
[30]: df.shape
```

```
[30]: (11251, 13)
```

#We can see 11251 rows and 13 columns

```
[32]: df.dropna(inplace = True)
      df.shape
```

```
[32]: (11239, 13)
```

#Now we have removed the null values

5 To Check all the Column Names

```
[34]: df.columns
```

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

6 To get Descriptive Analysis

[36]:

```
df.describe()
```

[36]:

```
            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   9453.610858
std    1.716039e+03     12.753866        0.493589      1.114967   5222.355869
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
```

#Here we get descriptive analysis of all the columns

[38]:

```
df[["Age","Orders","Amount"]].describe()
```

[38]:

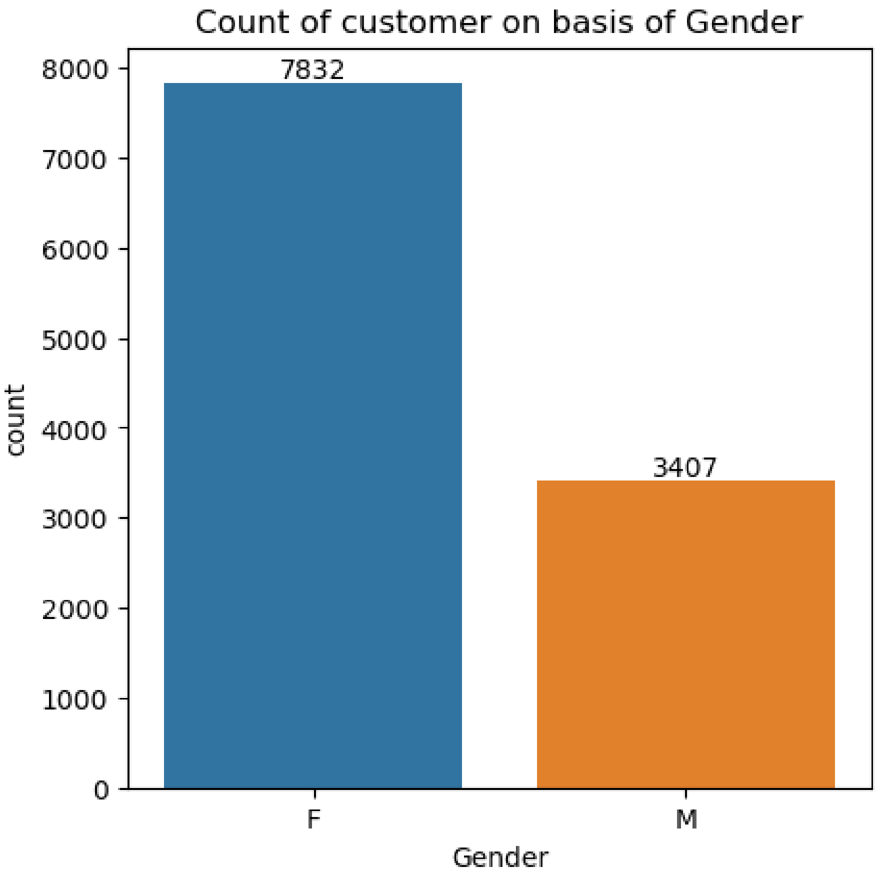
```
                Age        Orders        Amount
count  11239.000000  11239.000000  11239.000000
mean      35.410357      2.489634   9453.610858
std       12.753866      1.114967   5222.355869
min       12.000000      1.000000    188.000000
25%       27.000000      2.000000   5443.000000
50%       33.000000      2.000000   8109.000000
75%       43.000000      3.000000  12675.000000
max       92.000000      4.000000  23952.000000
```

#To get Descriptive analysis of selected columns

7 Exploratory Data Analysis

[42]:

```
plt.figure(figsize = (5,5))
plt.title("Count of customer on basis of Gender")
ax = sns.countplot(x = "Gender", data = df)
for bars in ax.containers:
    ax.bar_label(bars)
```



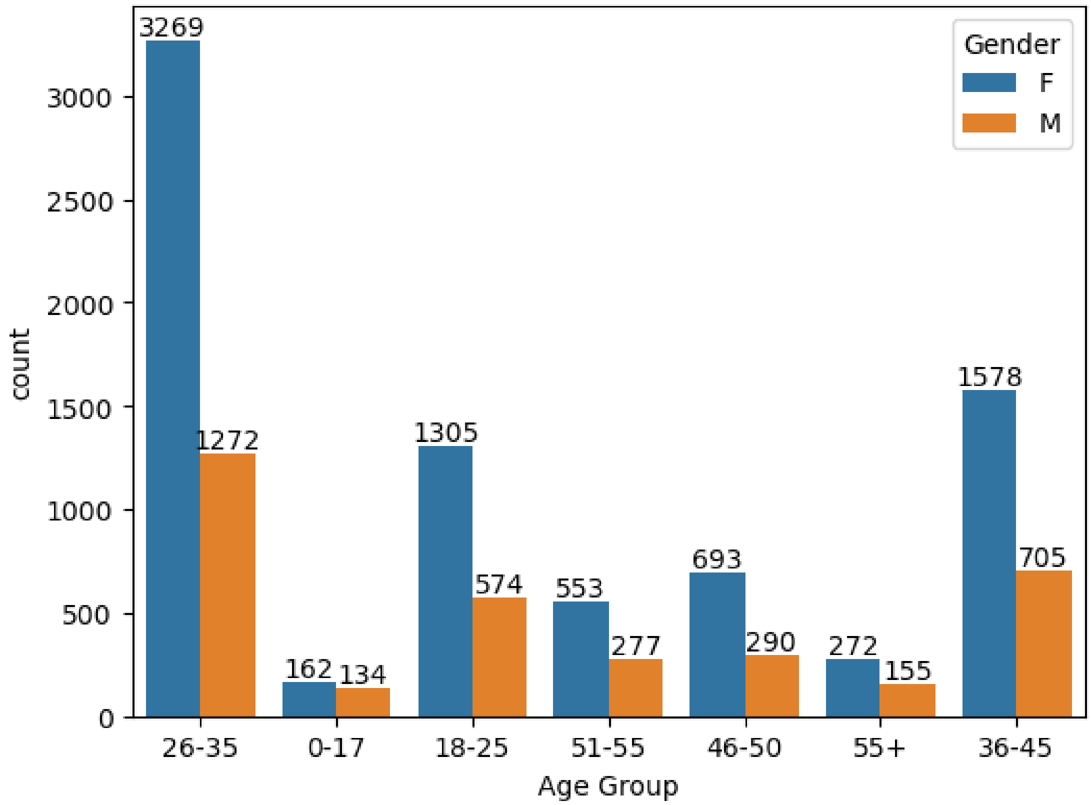
#We can see that Female Customer are more than Male Customer

[48]:
```
sales_gen = df.groupby(["Gender"], as_index=False)["Amount"].sum().
 ↪sort_values(by="Amount", ascending=False)
sales_gen
```

[48]:
```
  Gender       Amount
0      F  74335856.43
1      M  31913276.00
```

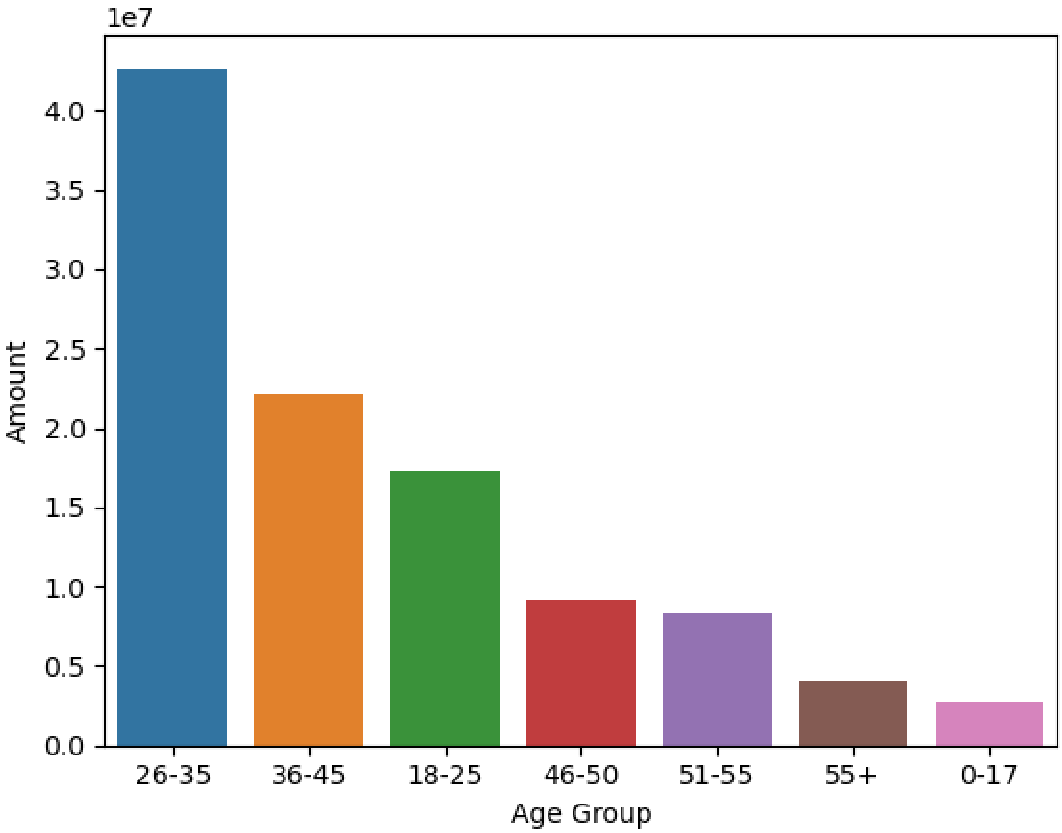
#Female Customer have purchased more than Male customers

[51]:
```
ax = sns.countplot(x = "Age Group", data = df, hue = "Gender")
for bars in ax.containers:
    ax.bar_label(bars)
```



[71]:

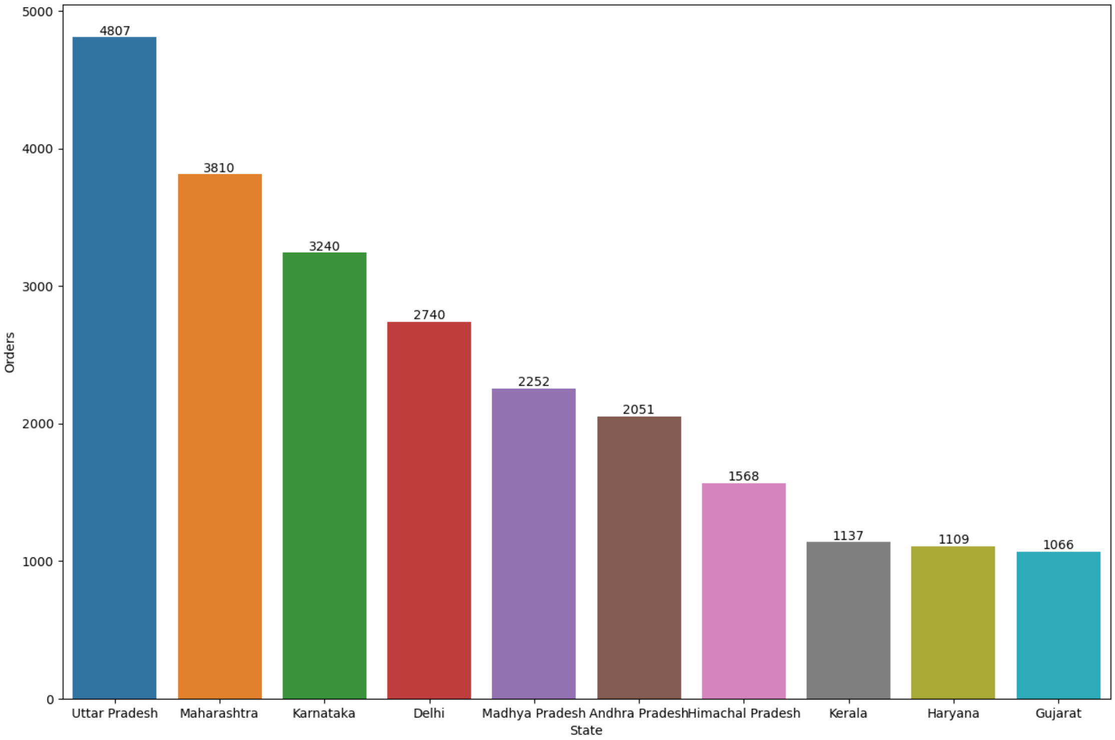
```
sales_age = df.groupby(["Age Group"], as_index=False)["Amount"].sum().
  ↪sort_values(by="Amount", ascending=False)
ax = sns.barplot(x = "Age Group", y = "Amount", data = sales_age)
```



#Here we can see that our most of the Customer belongs to age group 26-35 (Female)

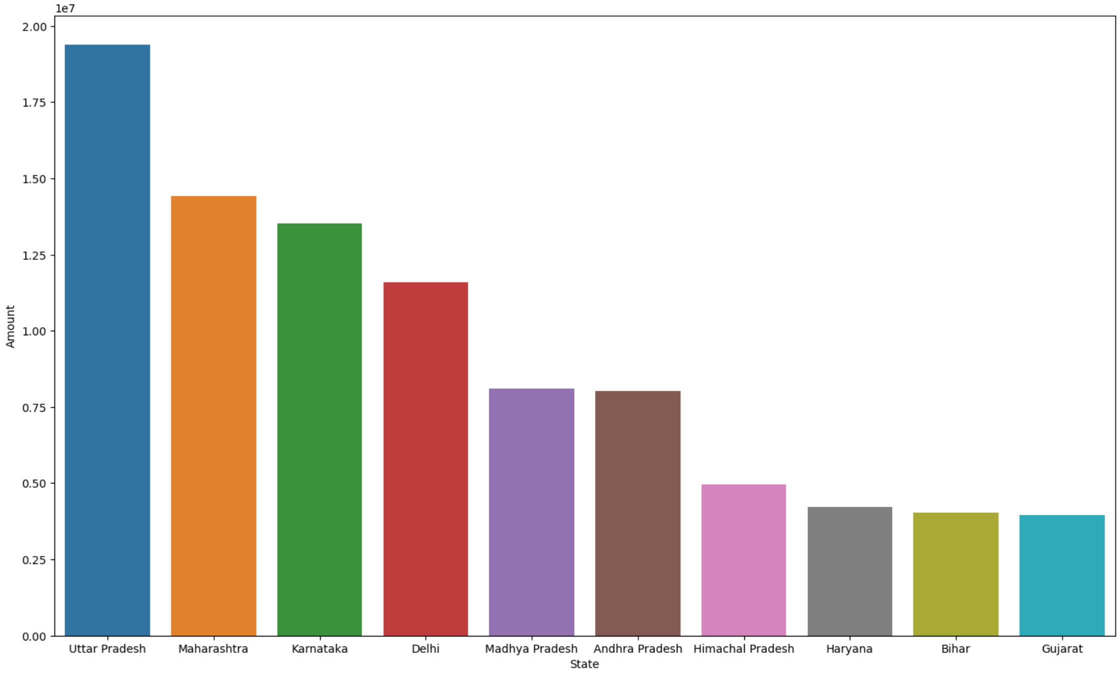
8 Top 10 States

[69]:
```
plt.figure(figsize=(15,10))
sales_state = df.groupby(["State"], as_index=False)["Orders"].sum().
  ↪sort_values(by="Orders", ascending=False).head(10)
ax = sns.barplot(x = "State", y = "Orders", data = sales_state)
for bars in ax.containers:
    ax.bar_label(bars)
```



#Here we can see that top three states with most no. of Orders are UP,Maharashtra and Karnataka

[76]:
```
plt.figure(figsize=(17,10))
sales_amount = df.groupby(["State"], as_index=False)["Amount"].sum().
  ↪sort_values(by="Amount", ascending=False).head(10)
ax = sns.barplot(x = "State", y = "Amount", data = sales_amount)
```

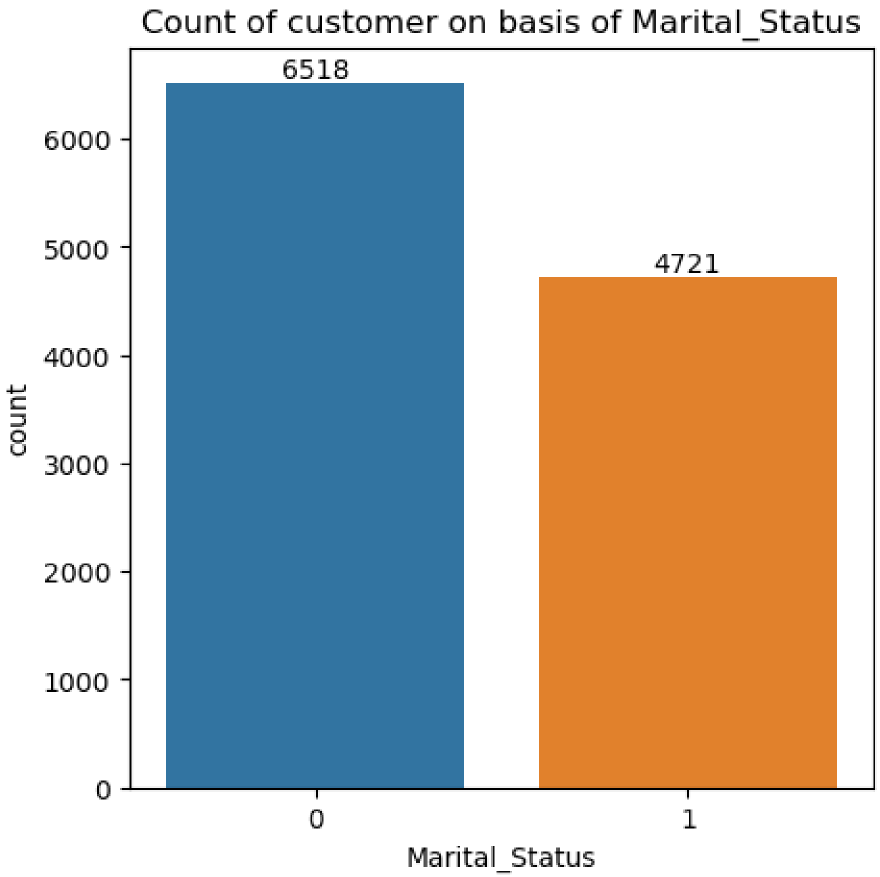


#Top 10 States based on Amount spend and here we can see the order change after Himachal Pradesh compared to above

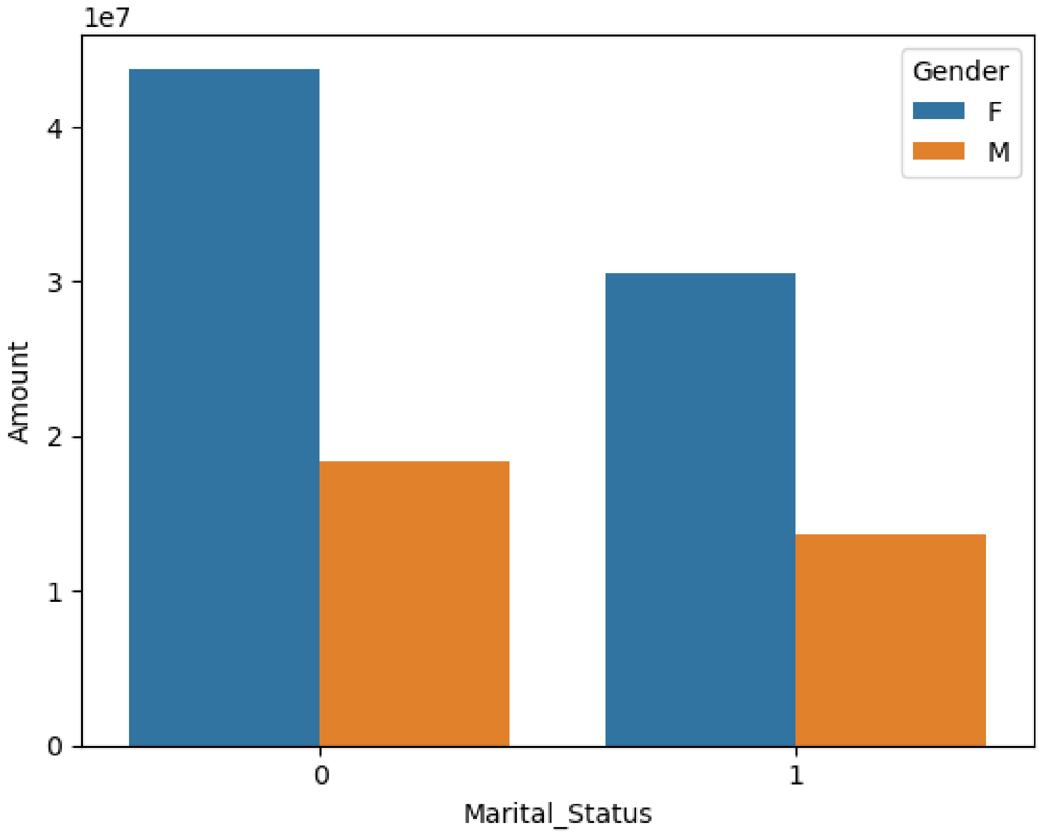
9 Marital_Status

[78]:

```
plt.figure(figsize = (5,5))
plt.title("Count of customer on basis of Marital_Status")
ax = sns.countplot(x = "Marital_Status", data = df)
for bars in ax.containers:
    ax.bar_label(bars)
```



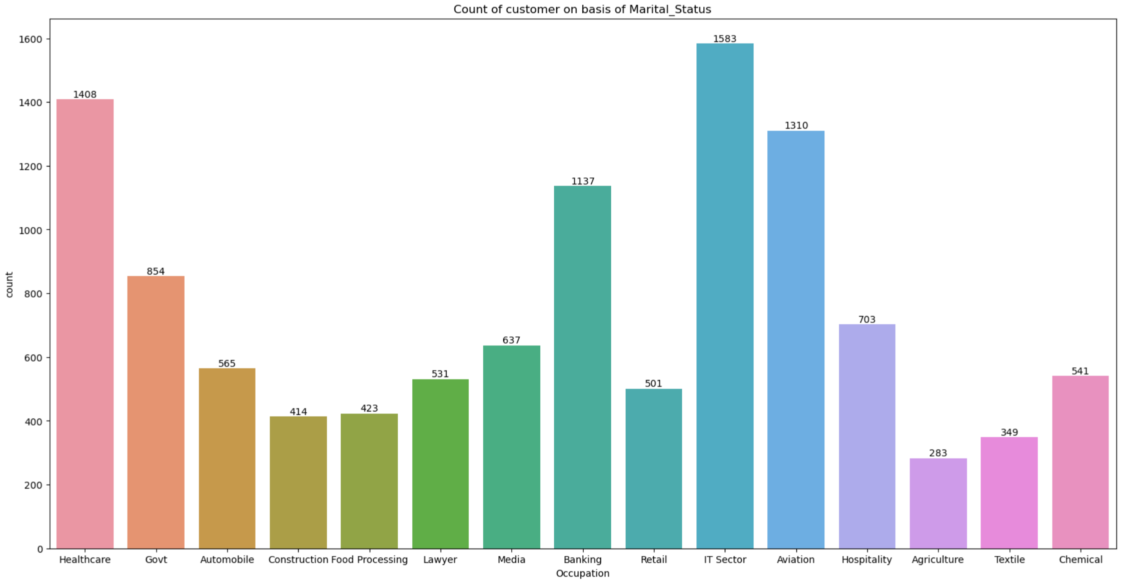
```
[84]: sales_marital = df.groupby(["Marital_Status","Gender"],␣
        ↪as_index=False)["Amount"].sum().sort_values(by="Amount", ascending=False).
        ↪head(10)
      ax = sns.barplot(x = "Marital_Status", y = "Amount", data = sales_marital, hue␣
        ↪= "Gender")
```



#Here we can see that most of our buyers are Unmarried

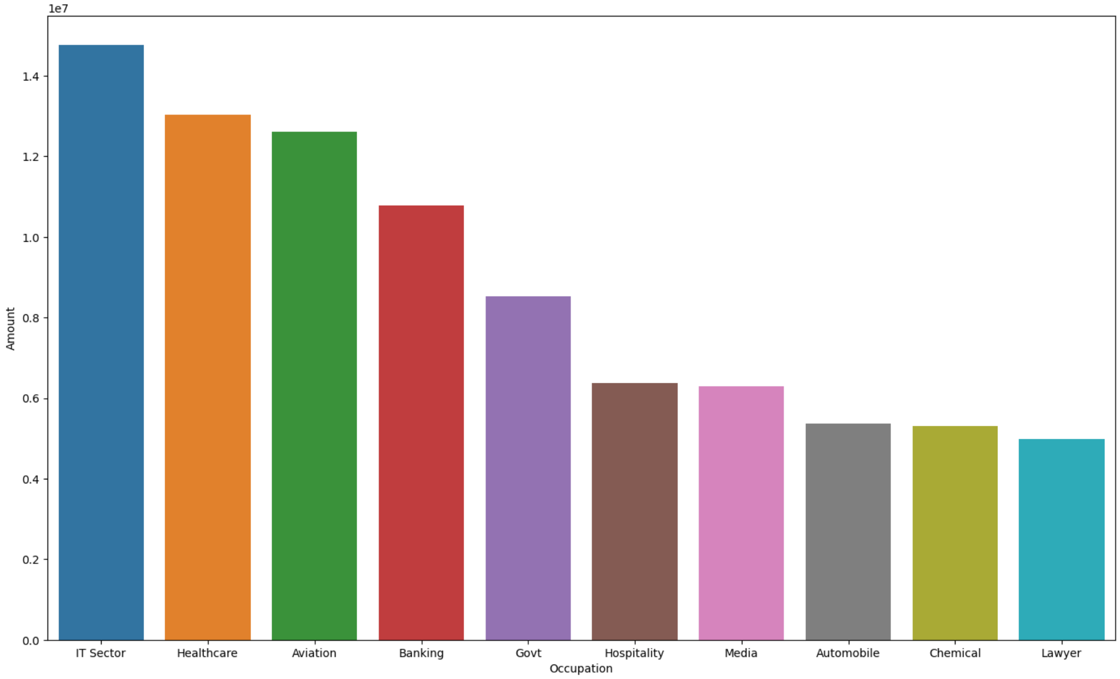
10 Occupation

```
[85]: plt.figure(figsize = (20,10))
      plt.title("Count of customer on basis of Occupation")
      ax = sns.countplot(x = "Occupation", data = df)
      for bars in ax.containers:
          ax.bar_label(bars)
```



#Most of our Customer are from IT Sector

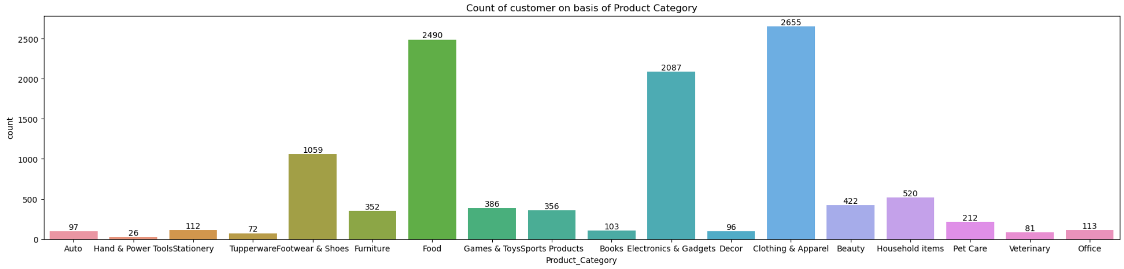
```
[87]: plt.figure(figsize=(17,10))
      sales_occupation = df.groupby(["Occupation"], as_index=False)["Amount"].sum().
       ↪sort_values(by="Amount", ascending=False).head(10)
      ax = sns.barplot(x = "Occupation", y = "Amount", data = sales_occupation)
```



#We can see that most of our buyers are from IT Sector,Healthcare,Aviation and banking Sector

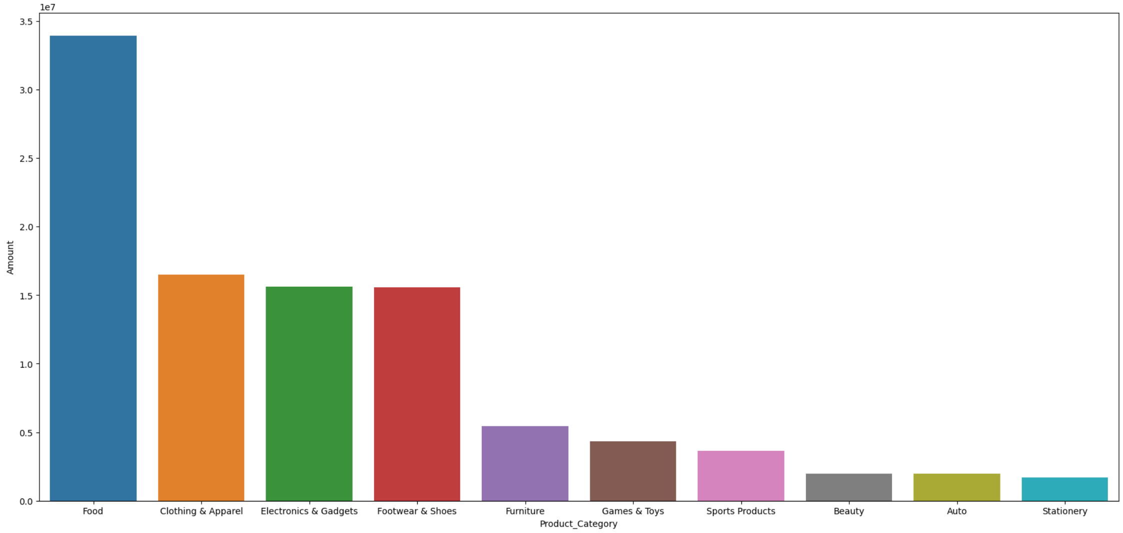
11 Product Category

[98]:
```
plt.figure(figsize = (24,5))
plt.title("Count of customer on basis of Product Category")
ax = sns.countplot(x = "Product_Category", data = df)
for bars in ax.containers:
    ax.bar_label(bars)
```



#Top 3 products are Clothing & Apparel, Food and Electronic and Gadgets

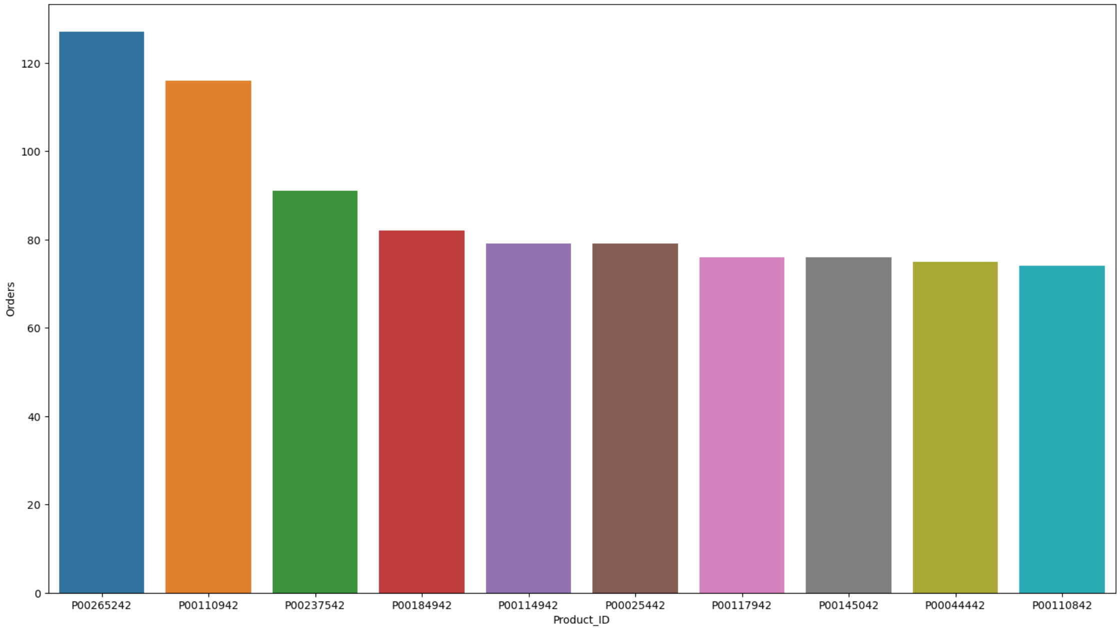
[101]:
```
plt.figure(figsize=(22,10))
sales_product = df.groupby(["Product_Category"], as_index=False)["Amount"].
  ↪sum().sort_values(by="Amount", ascending=False).head(10)
ax = sns.barplot(x = "Product_Category", y = "Amount", data = sales_product)
```



#Here we see that the most amount spend on was Food, Clothing & Apparel and Electronics & Gadget

[111]:

```
plt.figure(figsize=(18,10))
sales_productID = df.groupby(["Product_ID"], as_index=False)["Orders"].sum().
  ↪sort_values(by="Orders", ascending=False).head(10)
ax = sns.barplot(x = "Product_ID", y = "Orders", data = sales_productID)
```



#Most ordered product ID id P0026524

[ ]:

```
```