python-sales-analysis

March 13, 2025

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

```
[4]: df = pd.read_csv(r"C:\Users\hilom\OneDrive -␣
       ↪Hilo\My_Own_Stuff\Python_Diwali_Sales_Analysis\Python_Diwali_Sales_Analysis\Diwali␣
       ↪Sales Data.csv", encoding = 'latin1')
```

```
[5]: df
```

```
[5]:        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]
```

[6]:

```
df.head(10)
```

[6]:

```
   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
```

[7]:

```
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
```

[8]:

```
# Drop Function to Delete column
df.drop(['Status', 'unnamed1'], axis = 1, inplace = True)
```

[12]:

```
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
```

[17]:

```
#Checking Null Values
pd.isnull(df)
```

[17]:

```
       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]:
```
pd.isnull(df).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
```

[21]:
```
# Drop Null Values
df.dropna(inplace = True)
```

[23]:
```
df.shape
```

[23]:
```
(11239, 13)
```

[25]:
```
pd.isnull(df).sum()
```

[25]:
```
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
```

[27]:

```
#Change Data Type
df['Amount'] = df['Amount'].astype('int')
```

[29]:

```
df['Amount'].dtypes # Checking the Data Type
```

[29]:

```
dtype('int32')
```

[31]:

```
# Renaming the Column
df.rename(columns={'Gender': 'Gender_Category'}, inplace = True)
```

[33]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 11239 entries, 0 to 11250
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   User_ID           11239 non-null  int64
 1   Cust_name         11239 non-null  object
 2   Product_ID        11239 non-null  object
 3   Gender_Category   11239 non-null  object
 4   Age Group         11239 non-null  object
 5   Age               11239 non-null  int64
 6   Marital_Status    11239 non-null  int64
 7   State             11239 non-null  object
 8   Zone              11239 non-null  object
 9   Occupation        11239 non-null  object
 10  Product_Category  11239 non-null  object
 11  Orders            11239 non-null  int64
 12  Amount            11239 non-null  int32
dtypes: int32(1), int64(4), object(8)
memory usage: 1.2+ MB
```

[35]:

```
# Exploring the numerical Columns to infer about the data
df[['Age', 'Amount', 'Orders']].describe().round()
```

[35]:

```
           Age   Amount   Orders
count  11239.0  11239.0  11239.0
mean      35.0   9454.0      2.0
std       13.0   5222.0      1.0
min       12.0    188.0      1.0
25%       27.0   5443.0      2.0
50%       33.0   8109.0      2.0
75%       43.0  12675.0      3.0
max       92.0  23952.0      4.0
```

[37]:
```
df_stat_graph= df[['Age', 'Amount', 'Orders']].describe().round() # Extracting the data in a sperate excel
```

[39]:
```
df_stat_graph
```

[39]:
```
            Age   Amount   Orders
count  11239.0  11239.0  11239.0
mean      35.0   9454.0      2.0
std       13.0   5222.0      1.0
min       12.0    188.0      1.0
25%       27.0   5443.0      2.0
50%       33.0   8109.0      2.0
75%       43.0  12675.0      3.0
max       92.0  23952.0      4.0
```

[41]:
```
df_stat_graph.to_csv("stat.csv", index = True) # Saving it in an excel
```

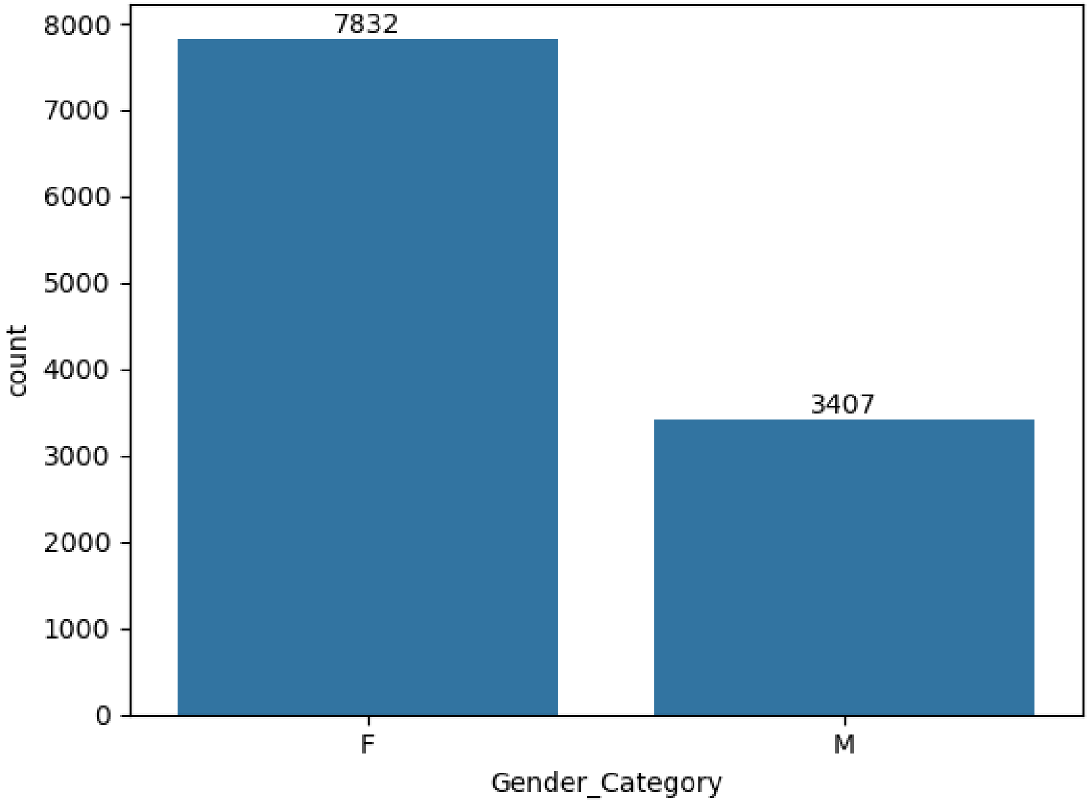
0.1 Exploratory Data Analysis

Gender

[45]:
```
df.columns
```

[45]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender_Category', 'Age Group',
       'Age', 'Marital_Status', 'State', 'Zone', 'Occupation',
       'Product_Category', 'Orders', 'Amount'],
      dtype='object')
```

[47]:
```
ax= sns.countplot(x = 'Gender_Category', data = df)
for bars in ax.containers:
    ax.bar_label(bars)
plt.show()
```



[55]:

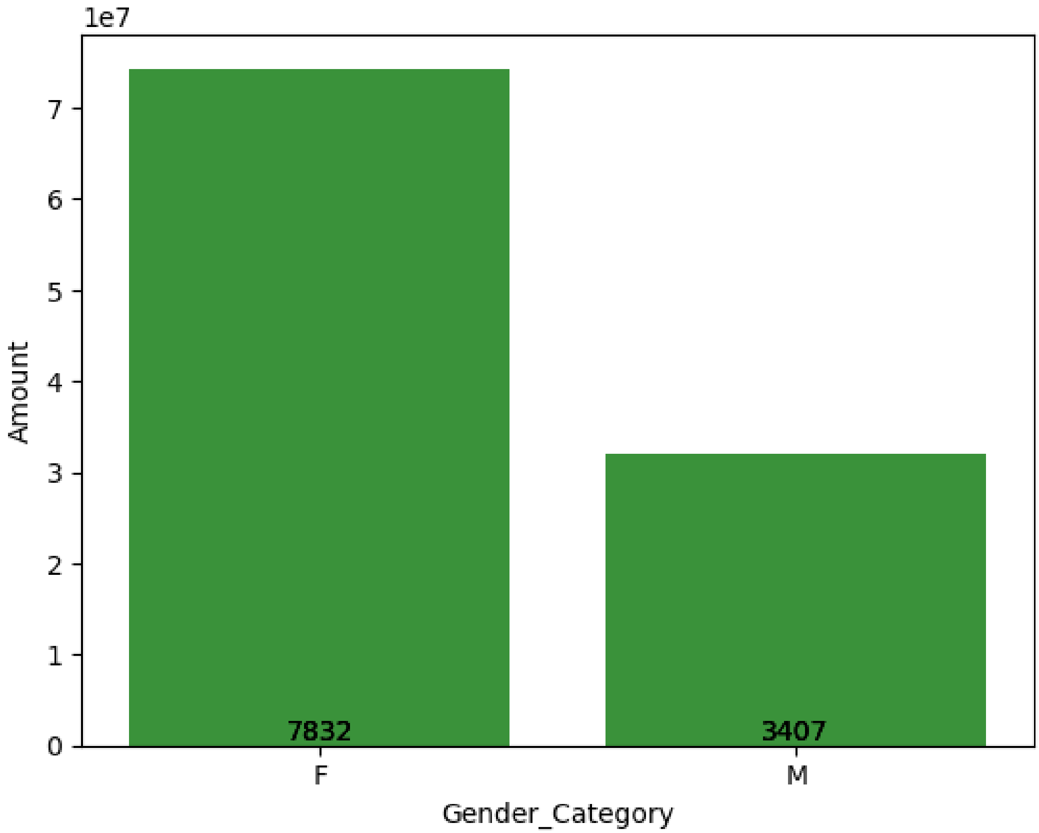
```
df.groupby('Gender_Category', as_index=False)['Amount'].sum().
 ↪sort_values(by='Amount', ascending=False)
```

[55]:

```
  Gender_Category    Amount
0               F  74335853
1               M  31913276
```

[69]:

```
sales_gen = df.groupby('Gender_Category', as_index=False)['Amount'].sum().
 ↪sort_values(by='Amount', ascending=False)
ad= sns.barplot(x='Gender_Category', y = 'Amount', data = sales_gen)
for bar in ad.containers:
    ad.bar_label(bars)
plt.show()
```



From the above graph we can conclude that Females have higher purchasing power and also by count the number of female buyers are more.

0.2 Age Group

[71]:
```
df
```

[71]:
```
       User_ID    Cust_name Product_ID Gender_Category Age Group  Age  \
0      1002903    Sanskriti  P00125942               F     26-35   28
1      1000732       Kartik  P00110942               F     26-35   35
2      1001990        Bindu  P00118542               F     26-35   35
3      1001425       Sudevi  P00237842               M      0-17   16
4      1000588         Joni  P00057942               M     26-35   28
…          …            …          …              …         …   …
11246  1000695      Manning  P00296942               M     18-25   19
11247  1004089  Reichenbach  P00171342               M     26-35   33
11248  1001209        Oshin  P00201342               F     36-45   40
11249  1004023       Noonan  P00059442               M     36-45   37
11250  1002744      Brumley  P00281742               F     18-25   19
```

```
       Marital_Status           State     Zone       Occupation  \
0                   0     Maharashtra  Western       Healthcare   
1                   1  Andhra Pradesh Southern             Govt   
2                   1   Uttar Pradesh  Central       Automobile   
3                   0       Karnataka Southern     Construction   
4                   1         Gujarat  Western  Food Processing   
…               …             …       …              …   
11246               1     Maharashtra  Western         Chemical   
11247               0         Haryana Northern       Healthcare   
11248               0  Madhya Pradesh  Central          Textile   
11249               0       Karnataka Southern      Agriculture   
11250               0     Maharashtra  Western       Healthcare   

      Product_Category  Orders  Amount  
0                 Auto       1   23952  
1                 Auto       3   23934  
2                 Auto       3   23924  
3                 Auto       2   23912  
4                 Auto       2   23877  
…                  …     …     …  
11246           Office       4     370  
11247       Veterinary       3     367  
11248           Office       4     213  
11249           Office       3     206  
11250           Office       3     188  

[11239 rows x 13 columns]
```

0.3 Age Group

[123]:

```
ss= df.groupby(["Age Group"], as_index = False)["Amount"].sum().round(4)
```

[125]:

```
print(ss)
```

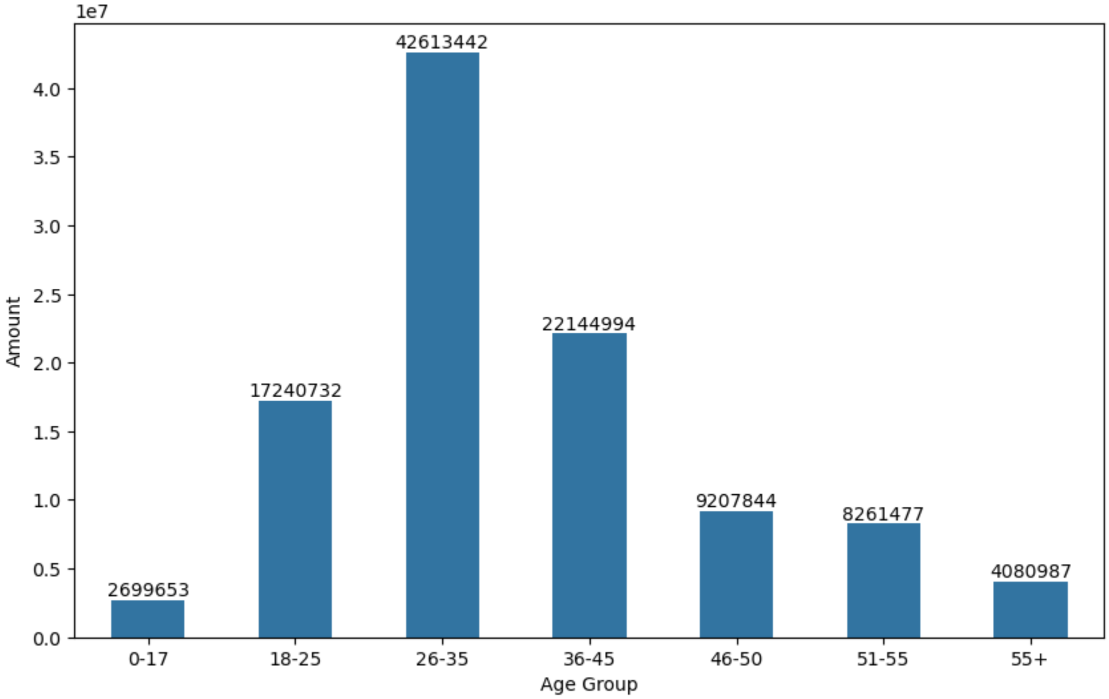
```
  Age Group    Amount
0      0-17   2699653
1     18-25  17240732
2     26-35  42613442
3     36-45  22144994
4     46-50   9207844
5     51-55   8261477
6       55+   4080987
```

[139]:

```
plt.figure(figsize=(10, 6))
ae=sns.barplot(x= "Age Group", y = "Amount", data = ss,  width=0.5)
```

[141]:

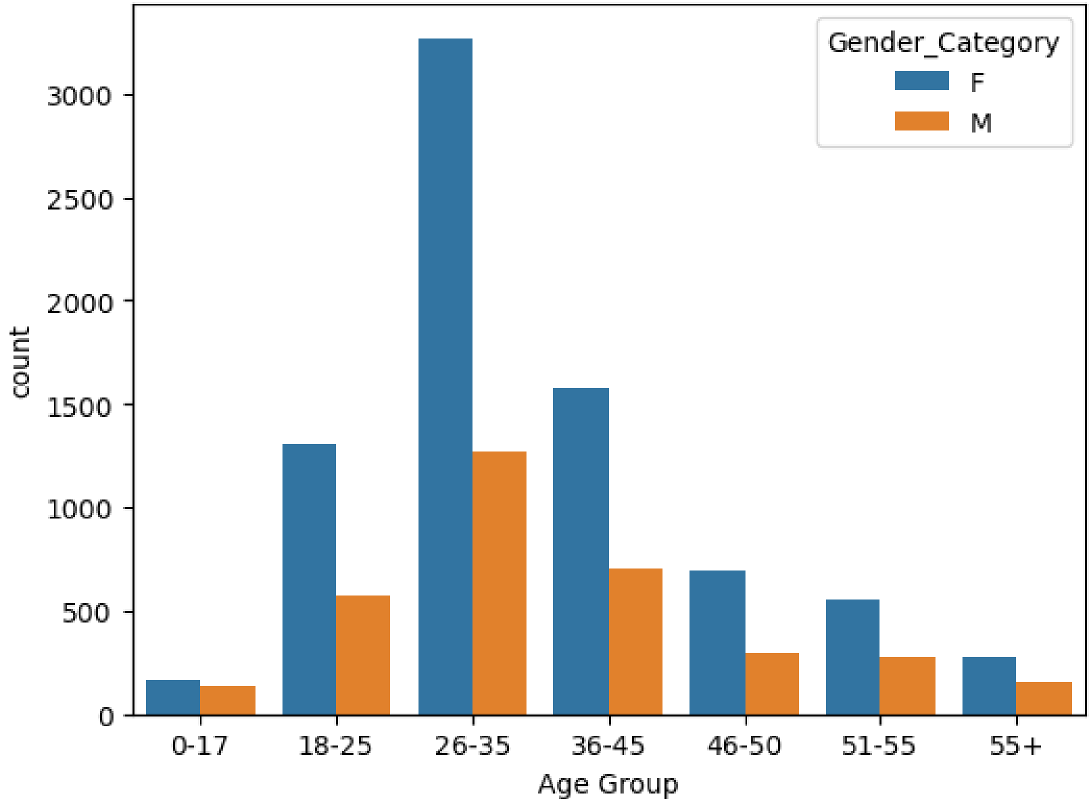
```
for bars in ae.containers:
    ae.bar_label(bars, fmt='%.0f')  # Format as integers (no decimal places)
plt.show()
```



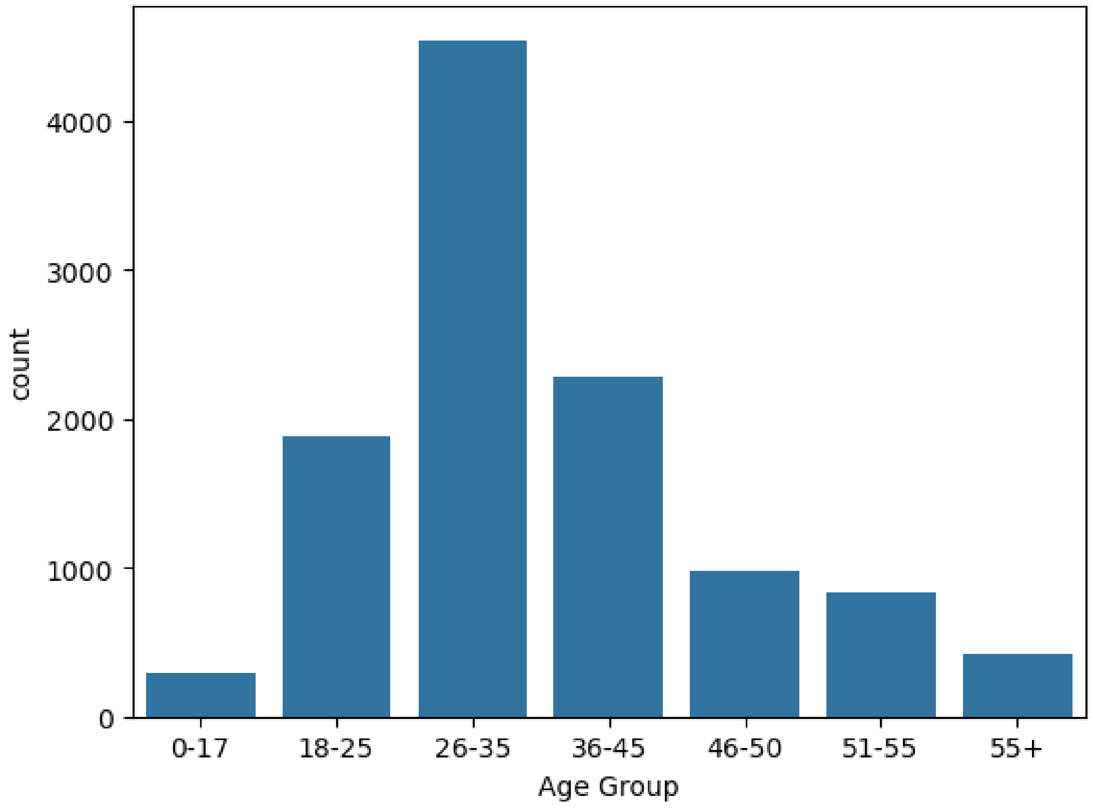
From Above Graph, we can conclude that the age group of 26-35 has the most contribution in the sale

[157]:

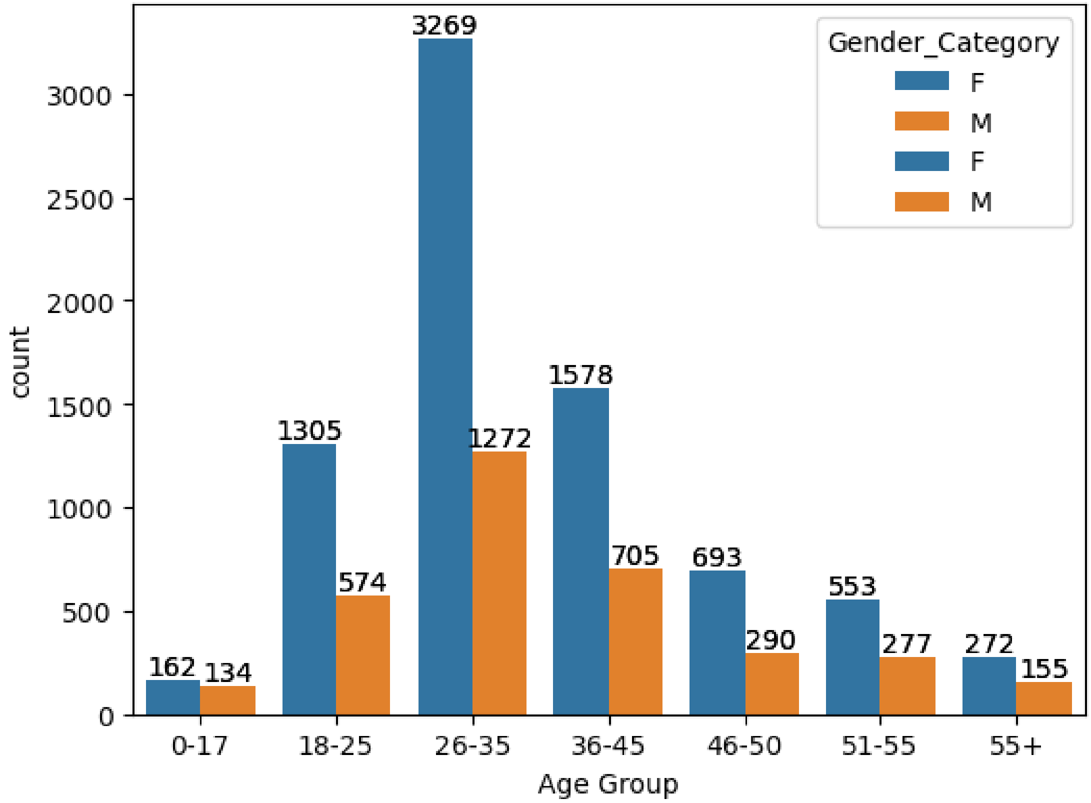
```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

# Define the desired order of age groups
age_group_order = ['0-17', '18-25', '26-35', '36-45', '46-50', '51-55', '55+']
# Create the countplot with the specified order
sns.countplot(data=df, x="Age Group", hue='Gender_Category',␣
 ↪order=age_group_order)

# Display the plot
plt.show()
```



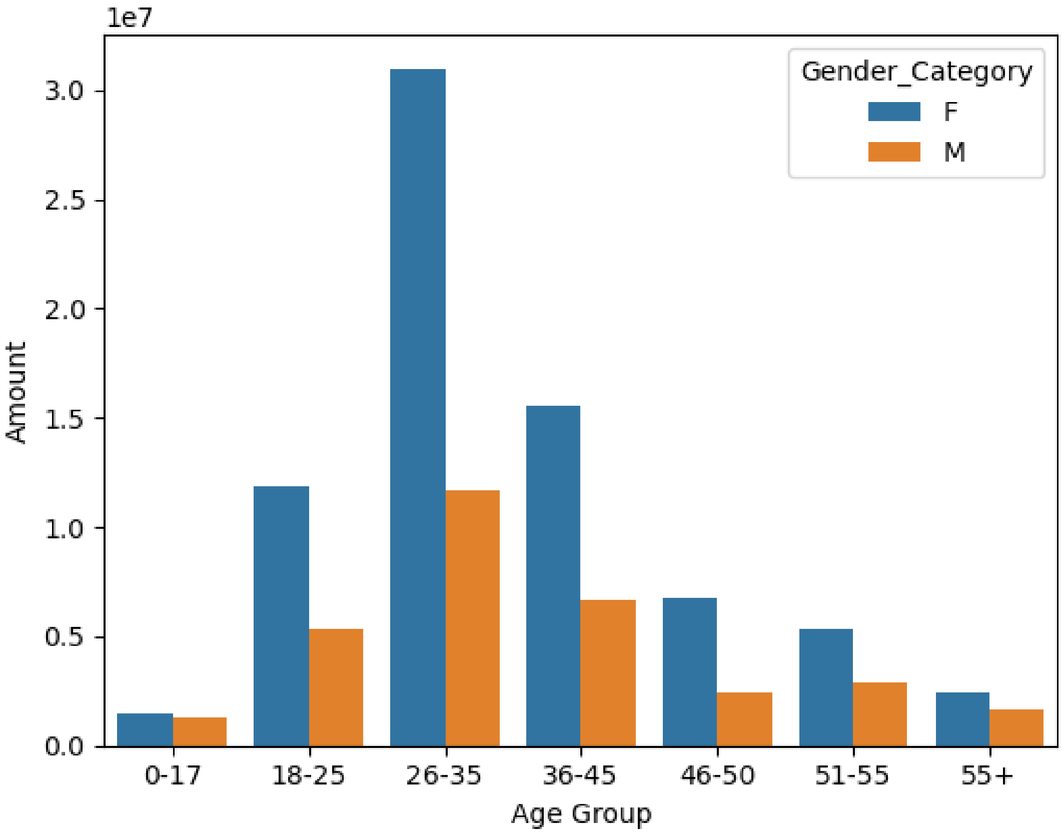
[159]:
```
# Define the desired order of age groups
age_group_order = ['0-17', '18-25', '26-35', '36-45', '46-50', '51-55', '55+']
# Create the countplot with the specified order
sns.countplot(data=df, x="Age Group", order=age_group_order)

# Display the plot
plt.show()
```



[175]:
```
age_group_order = ['0-17', '18-25', '26-35', '36-45', '46-50', '51-55', '55+']
we = sns.countplot(data=df, x="Age Group", hue='Gender_Category',␣
 ↪order=age_group_order)
for bars in we.containers:
    we.bar_label(bars)
plt.show()
```



[191]:
```
sales_age = df.groupby(['Age Group', "Gender_Category"], as_index =␣
 ↪False)['Amount'].sum().sort_values(by="Amount", ascending = False)
age_group_order = ['0-17', '18-25', '26-35', '36-45', '46-50', '51-55', '55+']
sns.barplot(x = "Age Group", y = "Amount", hue = "Gender_Category", data =␣
 ↪sales_age, order = age_group_order )

plt.show()
```



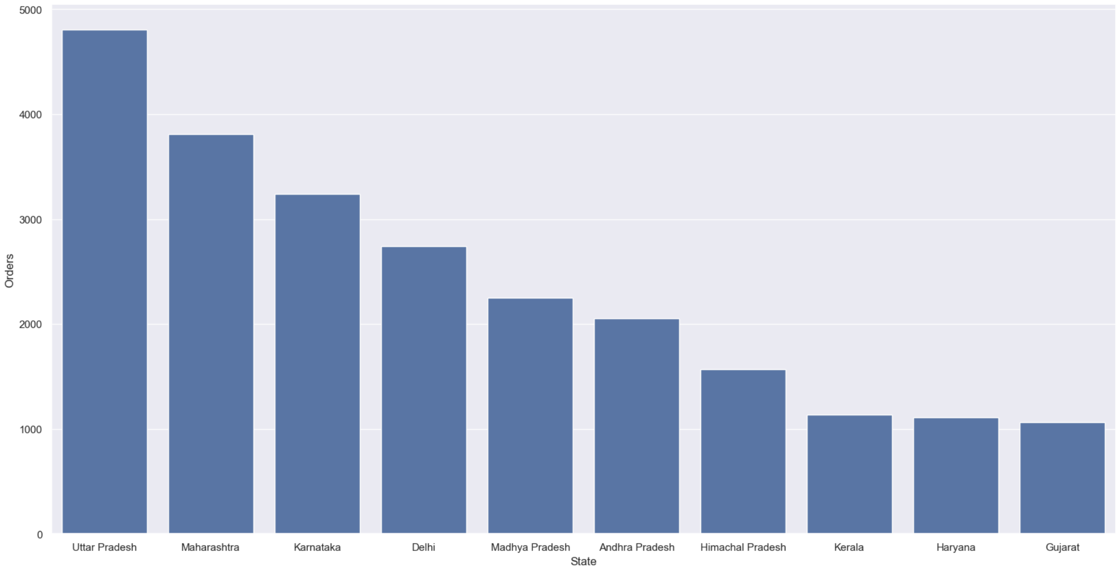
From the above graph we can see that most of the buyers are of age group between 26-35 years female.

0.4 State

```
[195]: df.columns
```

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

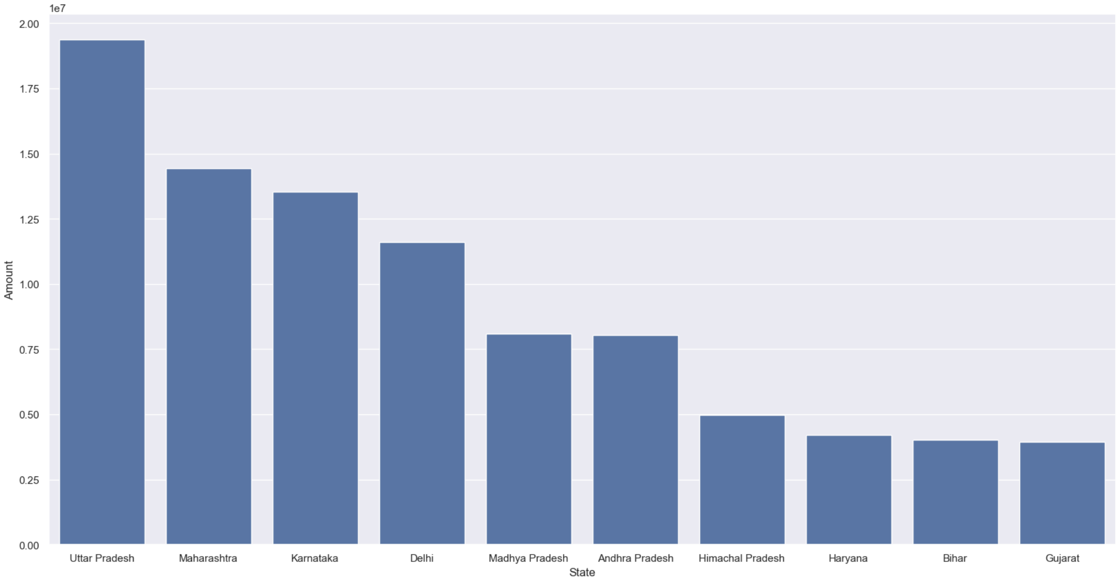
```
[209]: sales_state= df.groupby(["State"], as_index= False)['Orders'].sum().
         ↪sort_values(by = 'Orders', ascending =False).head(10)

       sns.set(rc = {'figure.figsize':(20,10)})
       sns.barplot(data= sales_state, x = 'State', y = 'Orders')
       plt.show()
```



Most Orders State = Uttar Pradesh, Maharashtra, Karanata.

[216]:

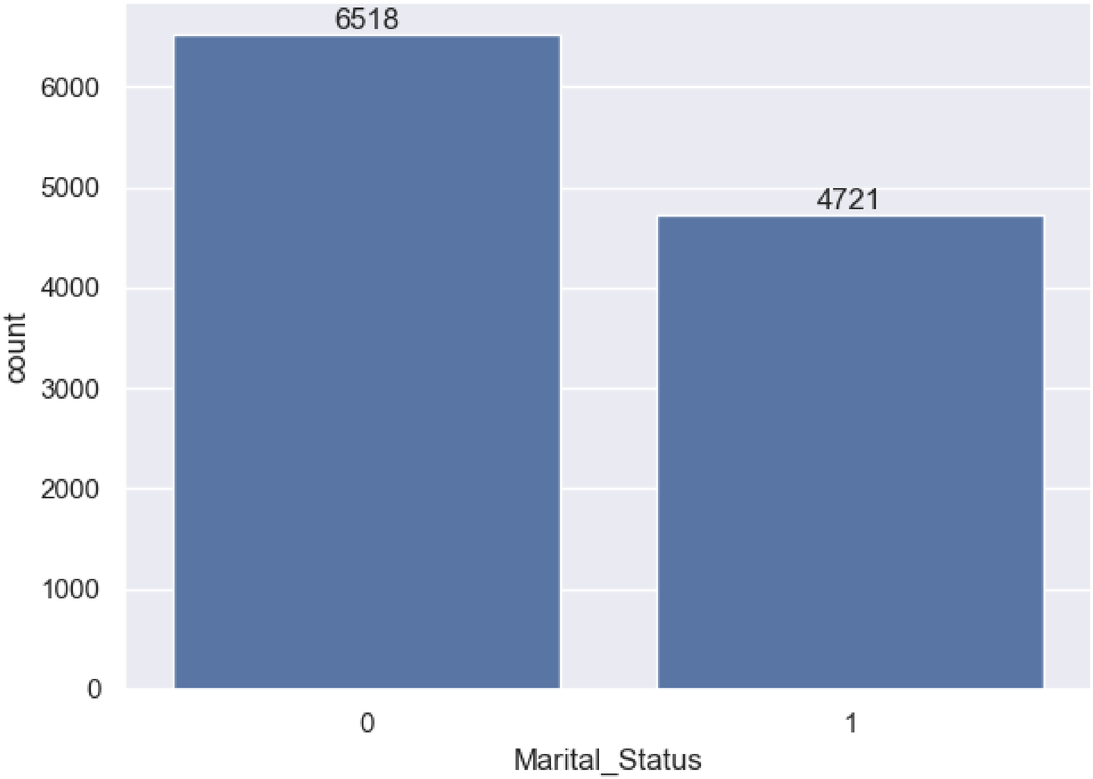
```
# Total amount of sales from top 10 states
state_amount = df.groupby(["State"], as_index = False)["Amount"].sum().
 ↪sort_values(by = "Amount", ascending = False).head(10)

sns.barplot(data = state_amount, x = "State", y = "Amount")
plt.show()
```



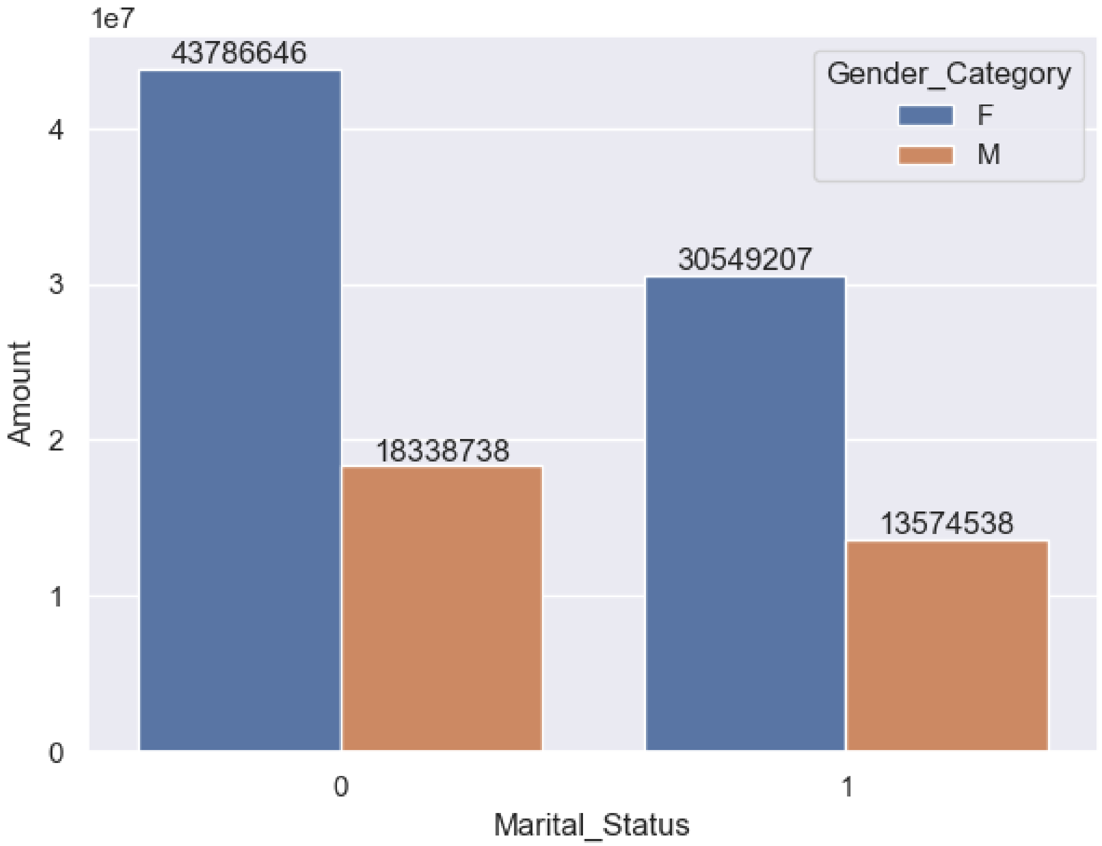
From the above we can see that in case of Orders and Purchasing power the state of UttarPradesh, Maharashtra and Karnataka topped, however unexpected ly in the order chart its evident that "Kerala" bagged the 8th Position but in terms of Purchasing Power (Amount Chart) we can see "Harayana" bagged the 8th position depicting that in term of orders though kerala was at high end but in terms of spending more money Haryana bagged the position.

0.5 Marital Status

[224]:
```
ax = sns.countplot(data = df, x = "Marital_Status")
sns.set(rc={'figure.figsize':(7,5)})
for bars in ax.containers:
    ax.bar_label(bars)
plt.show()
```



[246]:
```
marital_status_amt = df.groupby(['Marital_Status', 'Gender_Category'], as_index␣
↪=  False)["Amount"].sum().sort_values(by="Amount", ascending =False)
sns.set(rc={'figure.figsize':(7,5)})
ax= sns.barplot(data=marital_status_amt, x="Marital_Status", y = 'Amount', hue␣
↪= 'Gender_Category')
for bars in ax.containers:
    ax.bar_label(bars, fmt = '%.0F')
plt.show()
```

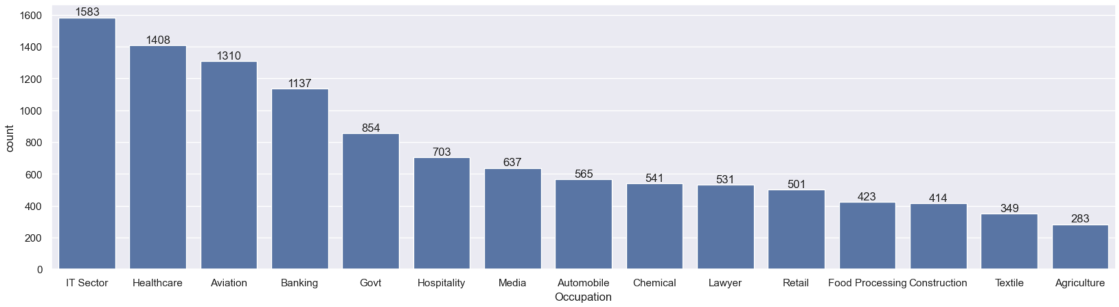


From the above graph we can see that most of the buyers are married (Women) and they have high Purchasing power.

0.6 Occupation

[334]:

```
sns.set(rc={'figure.figsize':(20,5)})
occupation_order = df['Occupation'].value_counts().index
ax = sns.countplot(data = df, x='Occupation', order = occupation_order)

for bars in ax.containers:
    ax.bar_label(bars, fmt = '%.0F')
plt.show()
```



Top performing Occupations area 1. IT 2. Healthcare 3. Aviation

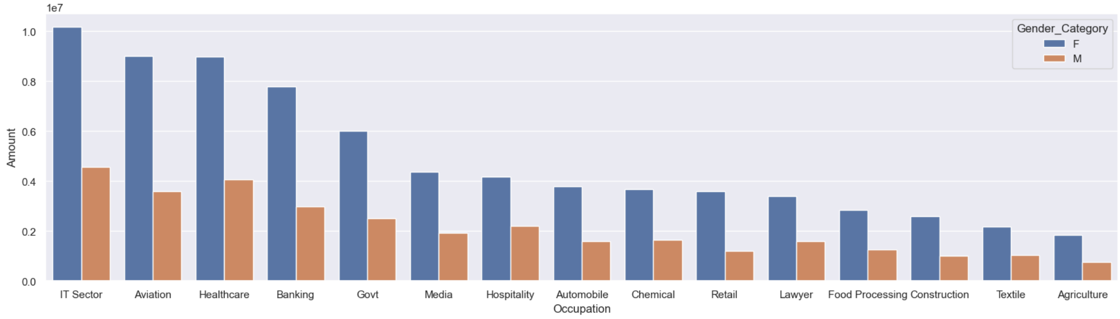
[296]:

```
# Showing Occupation VS Amount

Occu_gen_amt = df.groupby(["Occupation", "Gender_Category"])["Amount"].sum().
 ↪reset_index()
Occu_gen_amt = Occu_gen_amt.sort_values(by="Amount", ascending=False)

print(Occu_gen_amt)
```

```
         Occupation Gender_Category    Amount
20        IT Sector               F  10184835
4          Aviation               F   9007393
16       Healthcare               F   8968231
6           Banking               F   7792295
14             Govt               F   6002907
21        IT Sector               M   4570244
24            Media               F   4375029
18      Hospitality               F   4183199
17       Healthcare               M   4066355
2        Automobile               F   3768843
8          Chemical               F   3665084
5          Aviation               M   3594905
26           Retail               F   3583695
22           Lawyer               F   3383409
7           Banking               M   2978315
12  Food Processing               F   2825277
10     Construction               F   2595422
15             Govt               M   2514305
19      Hospitality               M   2193206
28          Textile               F   2159752
25            Media               M   1920803
0       Agriculture               F   1840482
9          Chemical               M   1632352
3        Automobile               M   1599753
23           Lawyer               M   1598256
```

```
13  Food Processing              M   1245393
27           Retail              M   1199475
29          Textile              M   1045220
11     Construction              M   1002089
1       Agriculture              M    752605
```

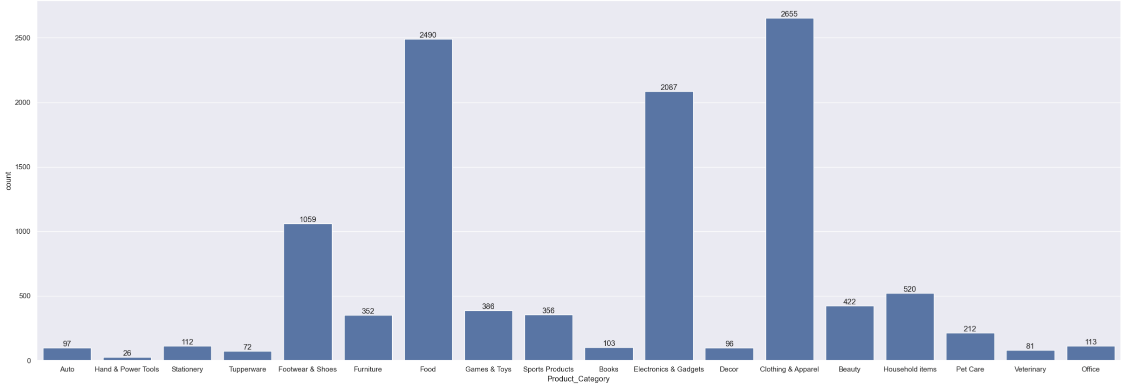
[338]:
```
sns.barplot(data=Occu_gen_amt, x="Occupation", y="Amount",␣
  ↪hue="Gender_Category")
plt.show()
```



From the graph its evident that the it sector has high purchasing powers, follwed by Aviation and Healthcare. With Females performing better.

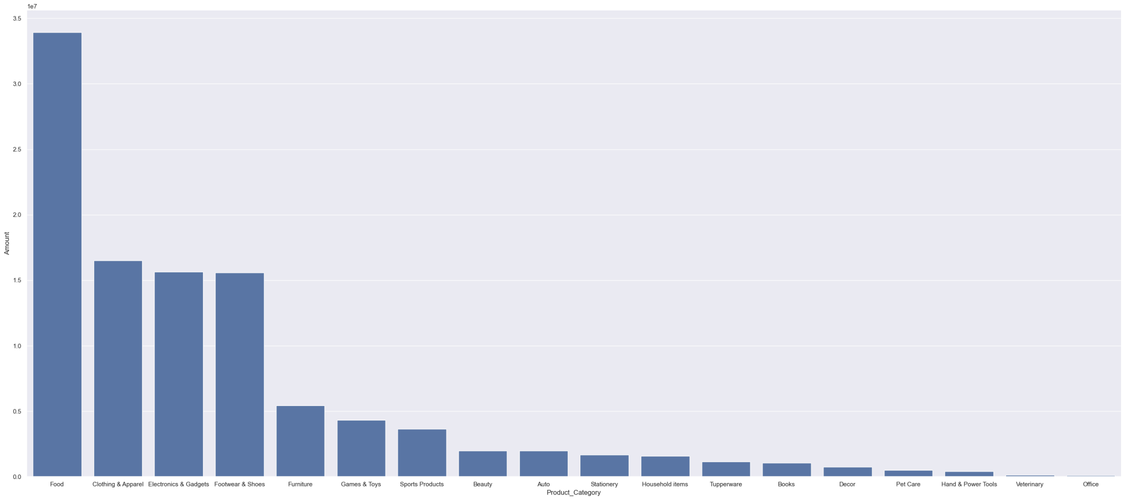
0.7 Category

[348]:
```
sns.set(rc={'figure.figsize':(30,10)})
ax = sns.countplot(data = df, x = 'Product_Category')

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

plt.show()
```



[374]:

```
sales_category = df.groupby(['Product_Category'], as_index = False)['Amount'].
 ↪sum().sort_values(by = 'Amount', ascending= False)
sns.set(rc={'figure.figsize':(35,15)})
sns.barplot(data =sales_category, x = 'Product_Category', y = 'Amount')
plt.show()
```



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

0.8 Conclusion (Overall)

”Married Women age group 26-35 years from UP, Maharashtra and Karnataka working on IT, Healthcare and Aviation are more likely buy products from Food , Clothing and Electronics Category.

[ ]: