Diwali sales analysis in python

August 25, 2024

[7]:

```
#pip install numpy
#pip install pandas
#pip install matplotlib
#pip install seaborn
```

```
Collecting seaborn
  Downloading seaborn-0.13.2-py3-none-any.whl.metadata (5.4 kB)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in
c:\users\rizwan\appdata\local\programs\python\python312\lib\site-packages (from
seaborn) (2.1.0)
Requirement already satisfied: pandas>=1.2 in
c:\users\rizwan\appdata\local\programs\python\python312\lib\site-packages (from
seaborn) (2.2.2)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in
c:\users\rizwan\appdata\local\programs\python\python312\lib\site-packages (from
seaborn) (3.9.2)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\rizwan\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (1.2.1)
Requirement already satisfied: cycler>=0.10 in
c:\users\rizwan\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\rizwan\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (4.53.1)
Requirement already satisfied: kiwisolver>=1.3.1 in
c:\users\rizwan\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (1.4.5)
Requirement already satisfied: packaging>=20.0 in
c:\users\rizwan\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (24.1)
Requirement already satisfied: pillow>=8 in
c:\users\rizwan\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (10.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
c:\users\rizwan\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (3.1.2)
Requirement already satisfied: python-dateutil>=2.7 in
```

```
c:\users\rizwan\appdata\local\programs\python\python312\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
c:\users\rizwan\appdata\local\programs\python\python312\lib\site-packages (from
pandas>=1.2->seaborn) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in
c:\users\rizwan\appdata\local\programs\python\python312\lib\site-packages (from
pandas>=1.2->seaborn) (2024.1)
Requirement already satisfied: six>=1.5 in
c:\users\rizwan\appdata\local\programs\python\python312\lib\site-packages (from
python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.16.0)
Downloading seaborn-0.13.2-py3-none-any.whl (294 kB)
   ---------------------------------------- 0.0/294.9 kB ? eta -:--:--
   - -------------------------------------- 10.2/294.9 kB ? eta -:--:--
   ----------- ---------------------------- 81.9/294.9 kB 1.2 MB/s eta 0:00:01
   -------------------------------------- - 286.7/294.9 kB 2.5 MB/s eta 0:00:01
   ---------------------------------------- 294.9/294.9 kB 2.0 MB/s eta 0:00:00
Installing collected packages: seaborn
Successfully installed seaborn-0.13.2
Note: you may need to restart the kernel to use updated packages.


[notice] A new release of pip is available: 24.0 -> 24.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

[2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[3]:
```
data=pd.read_csv(r"C:\Users\Rizwan\Downloads\python data analyst␣
 ↪project\Python_Diwali_Sales_Analysis-main\Diwali Sales Data.
 ↪csv",encoding='unicode_escape')
```

[4]:
```
data.head()
```

[4]:
```
   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

            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

     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
```

[14]:
```
data.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
```

[15]:
```
data.describe()
```

[15]:
```
            User_ID           Age  Marital_Status        Orders        Amount  \
count  1.125100e+04  11251.000000    11251.000000  11251.000000  11239.000000
mean   1.003004e+06     35.421207        0.420318      2.489290   9453.610858
std    1.716125e+03     12.754122        0.493632      1.115047   5222.355869
min    1.000001e+06     12.000000        0.000000      1.000000    188.000000
25%    1.001492e+06     27.000000        0.000000      1.500000   5443.000000
50%    1.003065e+06     33.000000        0.000000      2.000000   8109.000000
75%    1.004430e+06     43.000000        1.000000      3.000000  12675.000000
max    1.006040e+06     92.000000        1.000000      4.000000  23952.000000

       Status  unnamed1
count     0.0       0.0
```

```
mean      NaN       NaN
std       NaN       NaN
min       NaN       NaN
25%       NaN       NaN
50%       NaN       NaN
75%       NaN       NaN
max       NaN       NaN
```

```
[18]: data.nunique()
```

```
[18]: User_ID             3755
      Cust_name           1250
      Product_ID          2351
      Gender                 2
      Age Group              7
      Age                   81
      Marital_Status         2
      State                 16
      Zone                   5
      Occupation            15
      Product_Category      18
      Orders                 4
      Amount              6584
      Status                 0
      unnamed1               0
      dtype: int64
```

```
[21]: data.columns
```

```
[21]: Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
             'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
             'Orders', 'Amount', 'Status', 'unnamed1'],
            dtype='object')
```

```
[23]: data.count()
      #shows total number of non-null value in each column of dataset
```

```
[23]: User_ID             11251
      Cust_name           11251
      Product_ID          11251
      Gender              11251
      Age Group           11251
      Age                 11251
      Marital_Status      11251
      State               11251
      Zone                11251
      Occupation          11251
```

```
Product_Category    11251
Orders              11251
Amount              11239
Status                  0
unnamed1                0
dtype: int64
```

[8]:
```
#we can see that Status,unnamed1 this two columns contain only null value so we␣
 ↪have to drop them
data.drop(['Status','unnamed1'],axis=1,inplace=True)
```

[28]:
```
data
```

[28]:
```
       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]
```

[31]:

```
data.isnull().sum()
```

[31]:

```
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
```

[9]:

```
data.dropna(inplace=True)
```

[35]:

```
data
```

[35]:

```
       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

[11239 rows x 13 columns]
```

```
[36]: data.count()
```

```
[36]: User_ID             11239
      Cust_name           11239
      Product_ID          11239
      Gender              11239
      Age Group           11239
      Age                 11239
      Marital_Status      11239
      State               11239
      Zone                11239
      Occupation          11239
      Product_Category    11239
      Orders              11239
      Amount              11239
      dtype: int64
```

```
[ ]: #no null value in any cell
```

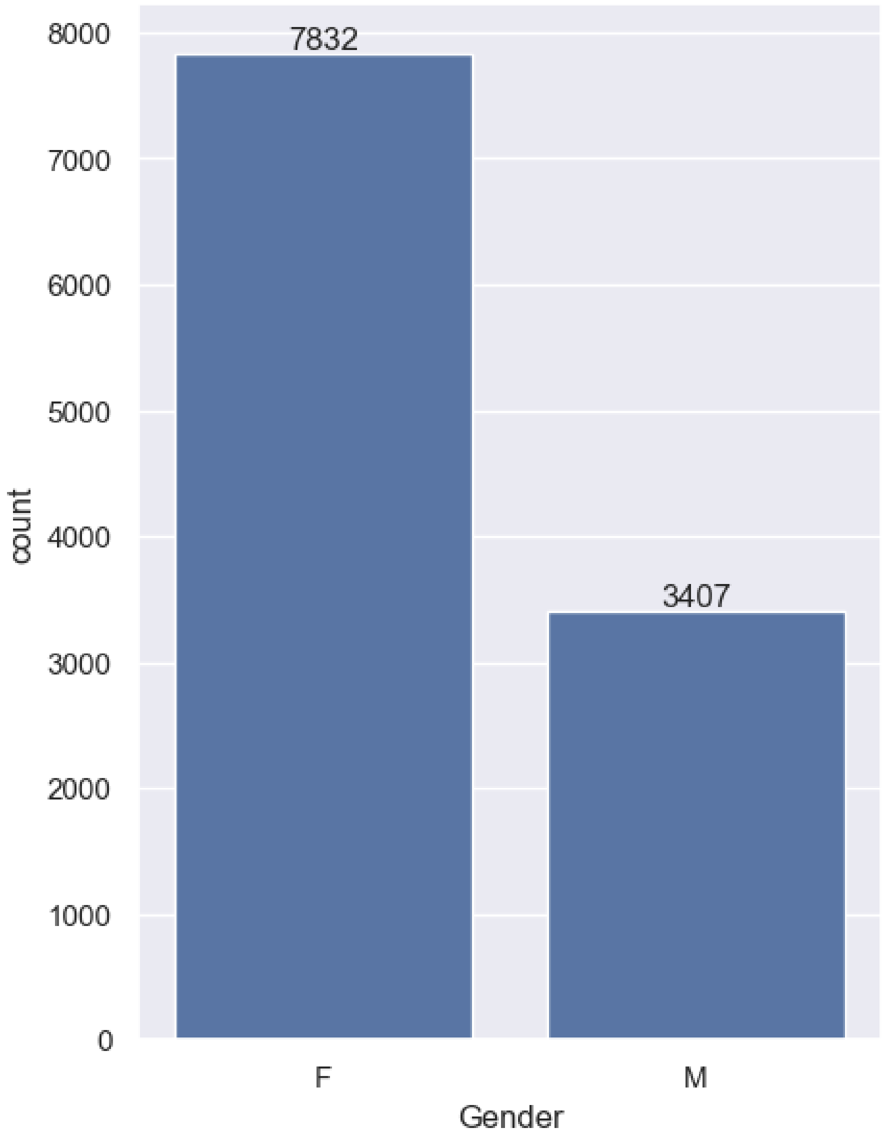
```
[37]: data.head()
```

```
[37]:    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

            State      Zone       Occupation Product_Category  Orders   Amount
0     Maharashtra   Western       Healthcare             Auto       1  23952.0
1  Andhra Pradesh  Southern             Govt             Auto       3  23934.0
2   Uttar Pradesh   Central       Automobile             Auto       3  23924.0
3       Karnataka  Southern     Construction             Auto       2  23912.0
4         Gujarat   Western  Food Processing             Auto       2  23877.0
```

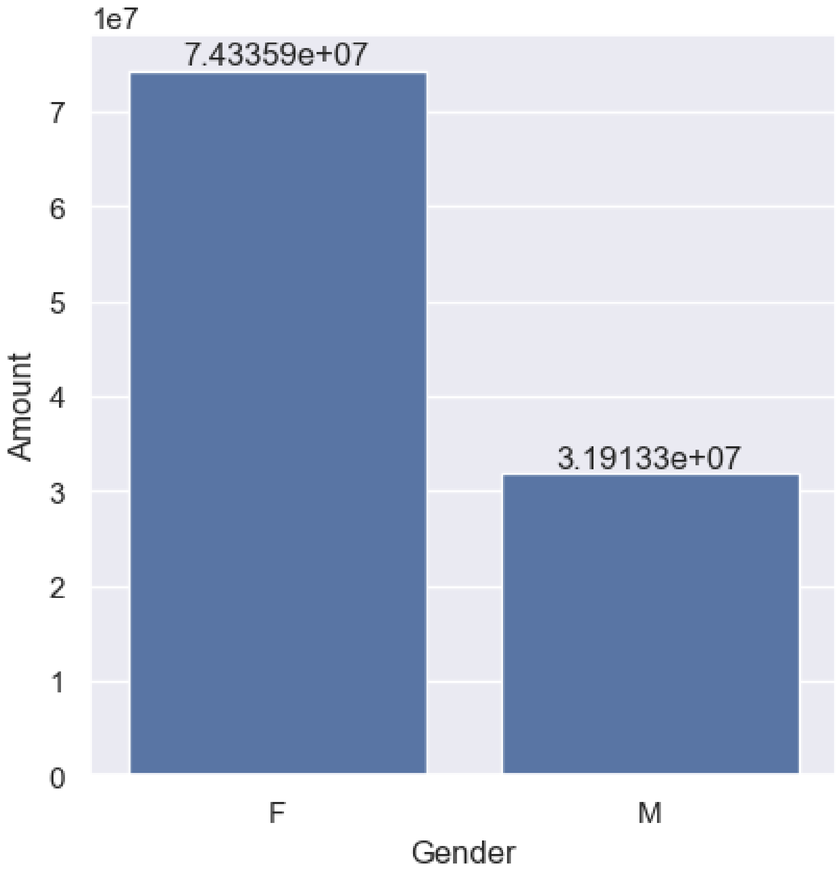
[146]:

```
ax=sns.countplot(data=data,x='Gender')
sns.set(rc={'figure.figsize':(5,5)})

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



[148]:

```
sales_gen=data.groupby(['Gender'],as_index=False)['Amount'].sum().
 ↪sort_values(by='Amount',ascending=False)
ax=sns.barplot(x='Gender',y='Amount',data=sales_gen)

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



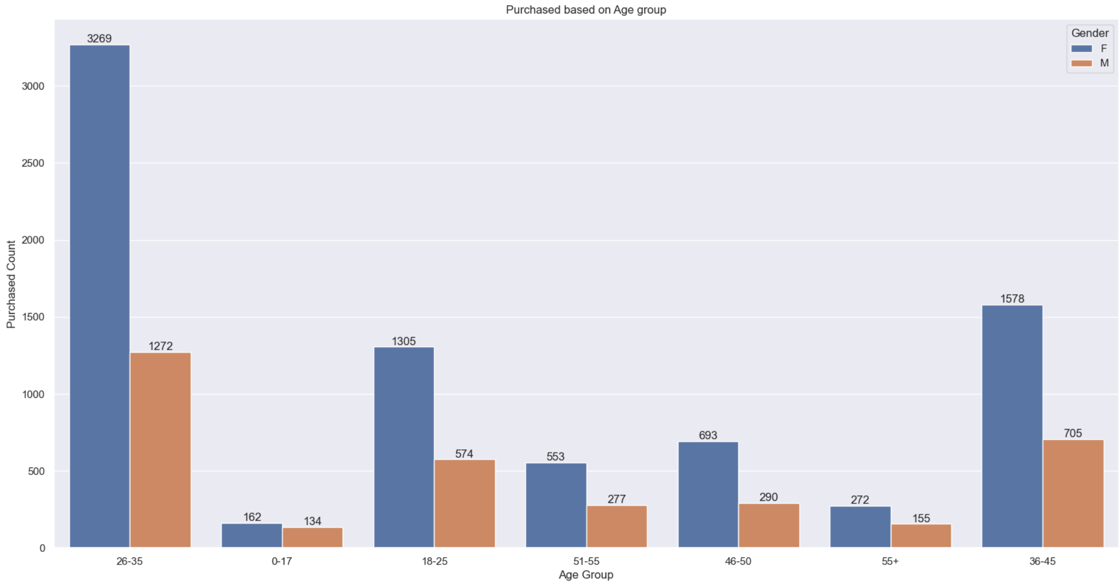
From above graphs we can say that total sales by female and male customer are 7,43,35,900 and 3,19,13,300 respectively

Most buyers are females with purchasing greter than males

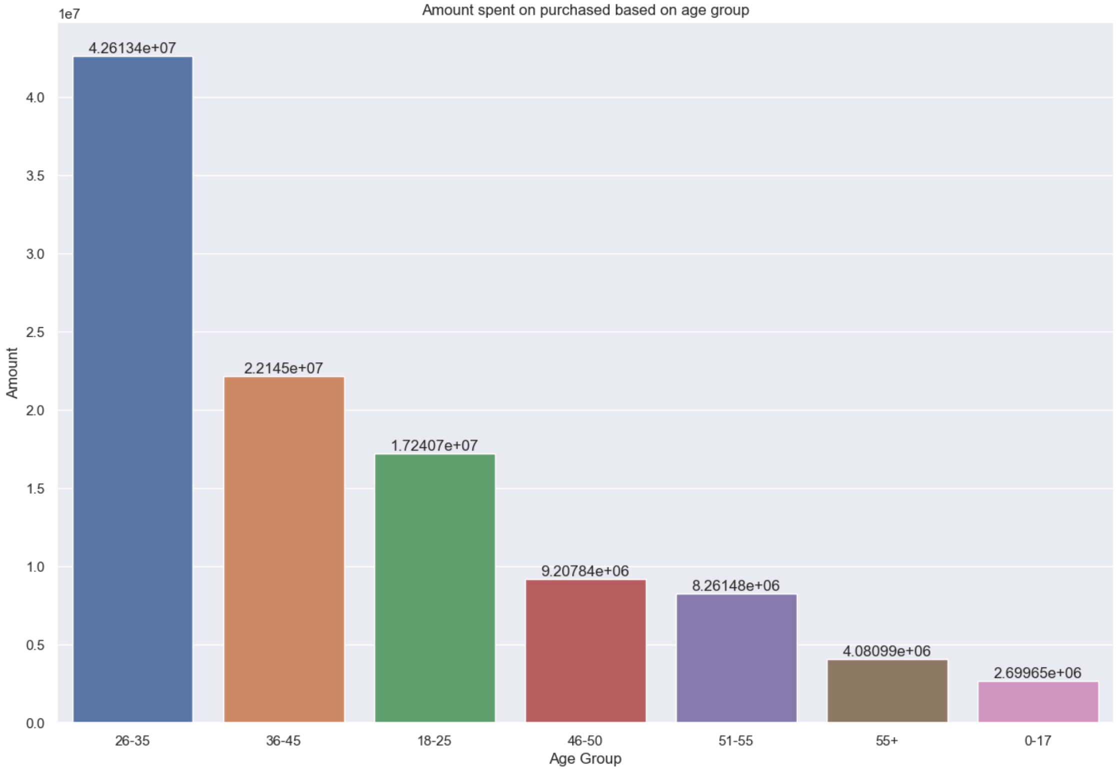
[51]:
```
data.head()
```

[51]:
```
   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

            State      Zone       Occupation Product_Category  Orders   Amount
0     Maharashtra   Western       Healthcare             Auto       1  23952.0
1  Andhra Pradesh  Southern             Govt             Auto       3  23934.0
2   Uttar Pradesh   Central       Automobile             Auto       3  23924.0
3       Karnataka  Southern     Construction             Auto       2  23912.0
4         Gujarat   Western  Food Processing             Auto       2  23877.0
```

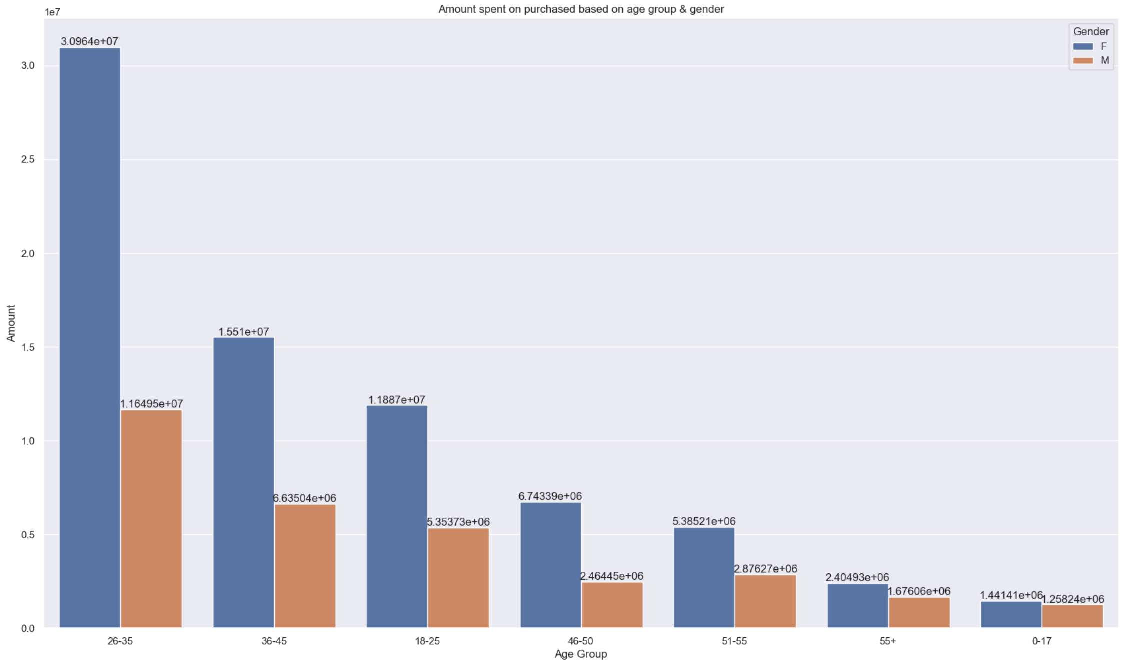
[104]:
```python
ax=sns.countplot(x='Age Group',data=data,hue='Gender')
plt.ylabel('Purchased Count')
plt.title('Purchased based on Age group')
#bar labels
for bar in ax.containers:
    ax.bar_label(bar)
```



[126]:
```python
sale_age_grp=data.groupby(['Age Group'],as_index=False)['Amount'].sum().
 ↪sort_values(by="Amount",ascending=False)
sns.set(rc={'figure.figsize':(15,10)})
ax=sns.barplot(x='Age Group',y='Amount',hue='Age Group',data=sale_age_grp)
plt.title('Amount spent on purchased based on age group')
for bar in ax.containers:
    ax.bar_label(bar)
```

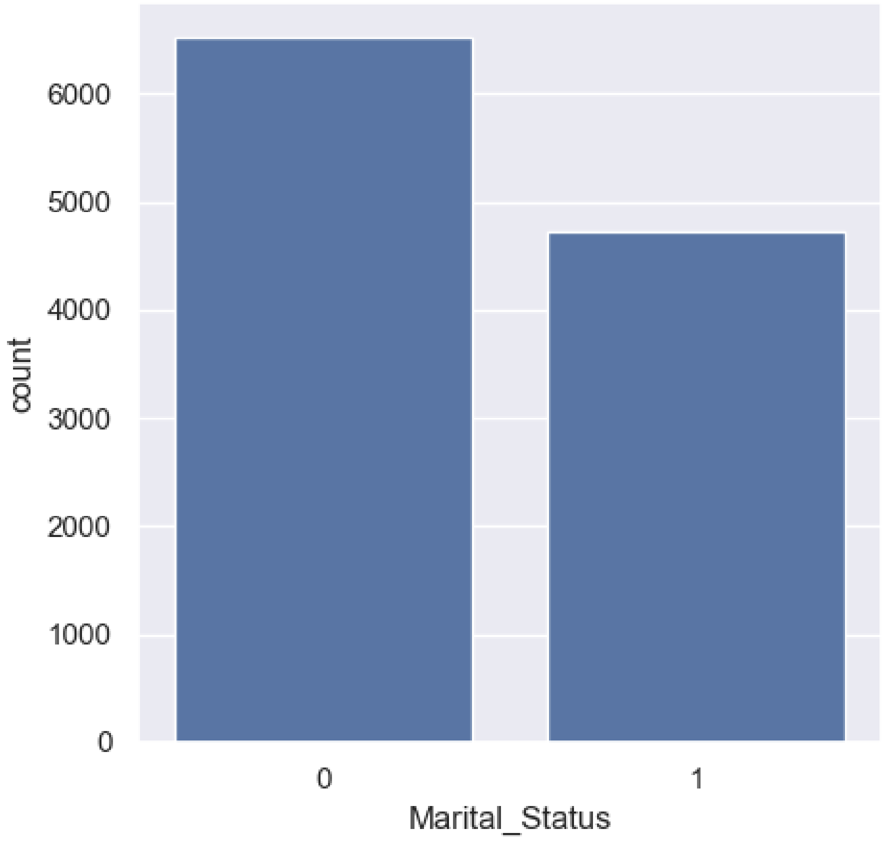


[131]:
```
sales_agegrp_gen=data.groupby(['Age Group','Gender'],as_index=False)['Amount'].
  ↪sum().sort_values(by='Amount',ascending=False)
sns.set(rc={'figure.figsize':(21,12)})
ax=sns.barplot(x='Age Group',y='Amount',hue='Gender',data=sales_agegrp_gen)
plt.title('Amount spent on purchased based on age group & gender')
for bar in ax.containers:
    ax.bar_label(bar)
```



Most buyers are between age group of 26-35 which are females with purchasing greater than Males

[149]:
```
ax=sns.countplot(x='Marital_Status',data=data)
```



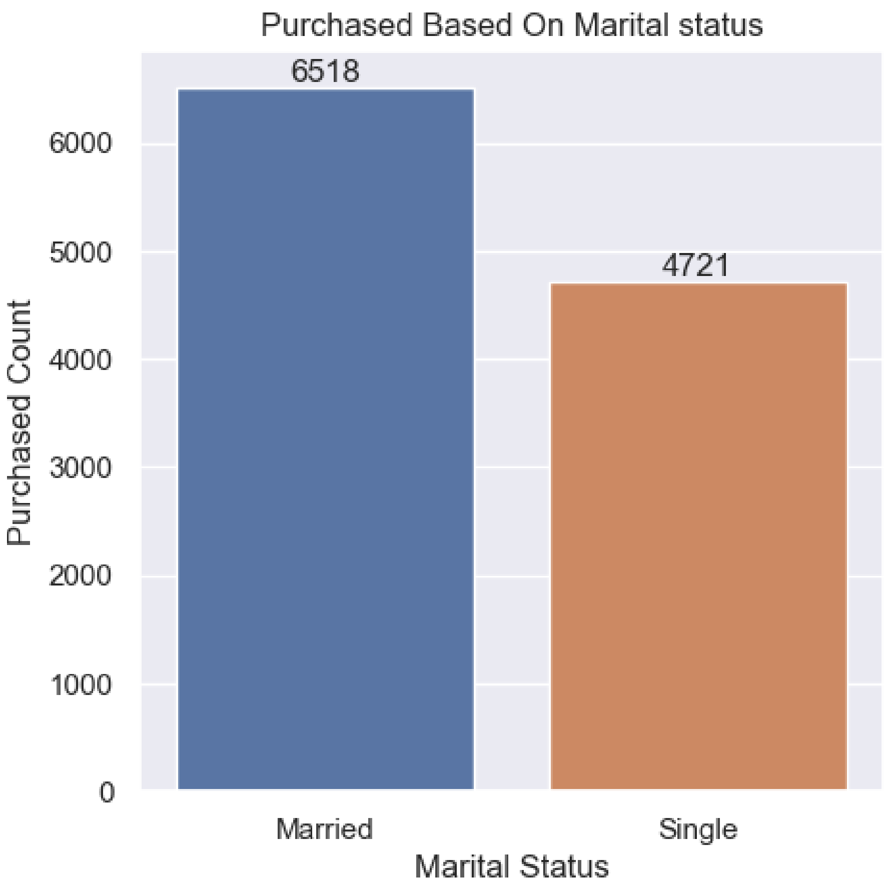
[6]:
```
marital_status_readable={0:'Married',1:'Single'}
data['marital_status_readable']=data['Marital_Status'].
 ↪map(marital_status_readable)
```

[114]:
```
data
```

[114]:
```
       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 marital_status_readable
0      23952.0                 Married
1      23934.0                  Single
2      23924.0                  Single
3      23912.0                 Married
4      23877.0                  Single
…        …                     …
11246    370.0                  Single
11247    367.0                 Married
11248    213.0                 Married
11249    206.0                 Married
11250    188.0                 Married

[11239 rows x 14 columns]
```

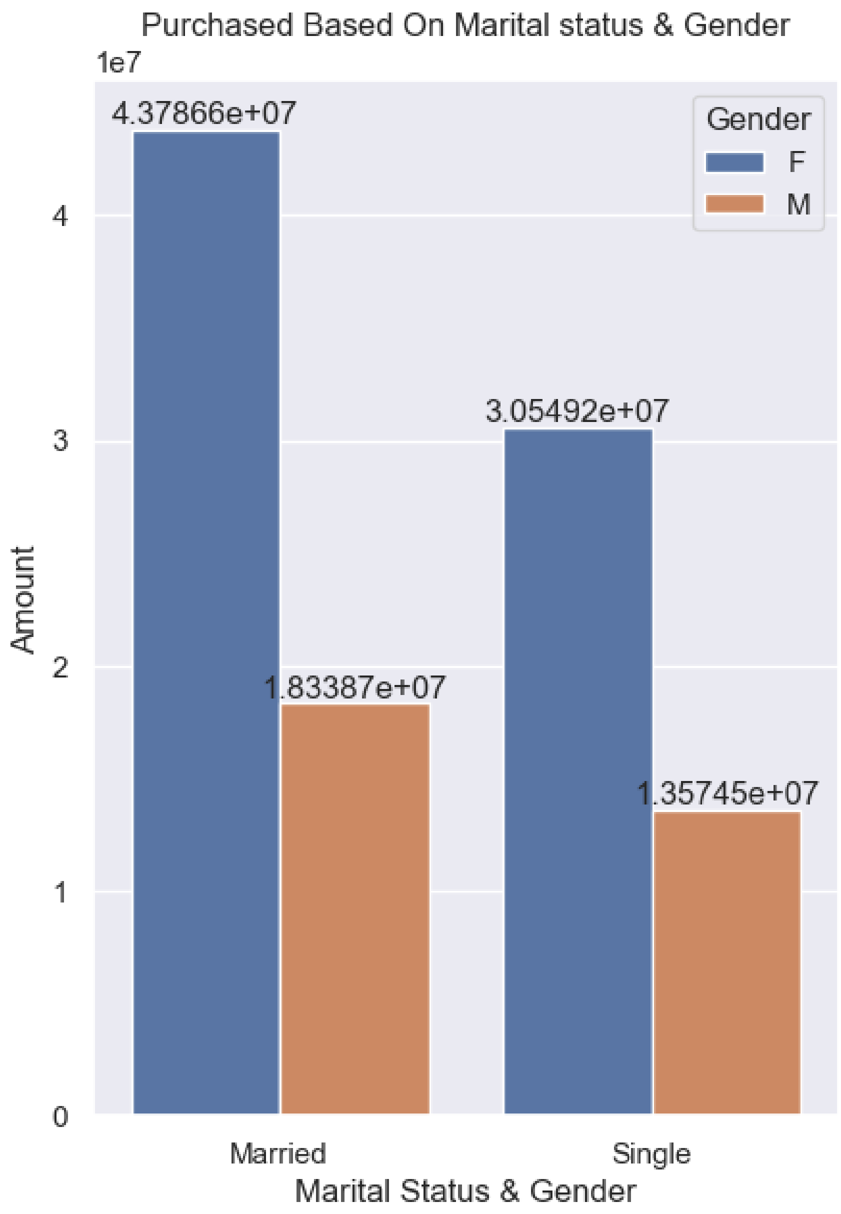
[150]:

```
ax=sns.
 ↪countplot(x='marital_status_readable',hue='marital_status_readable',data=data)
plt.title('Purchased Based On Marital status')
plt.xlabel('Marital Status')
plt.ylabel('Purchased Count')
for bar in ax.containers:
    ax.bar_label(bar)
```



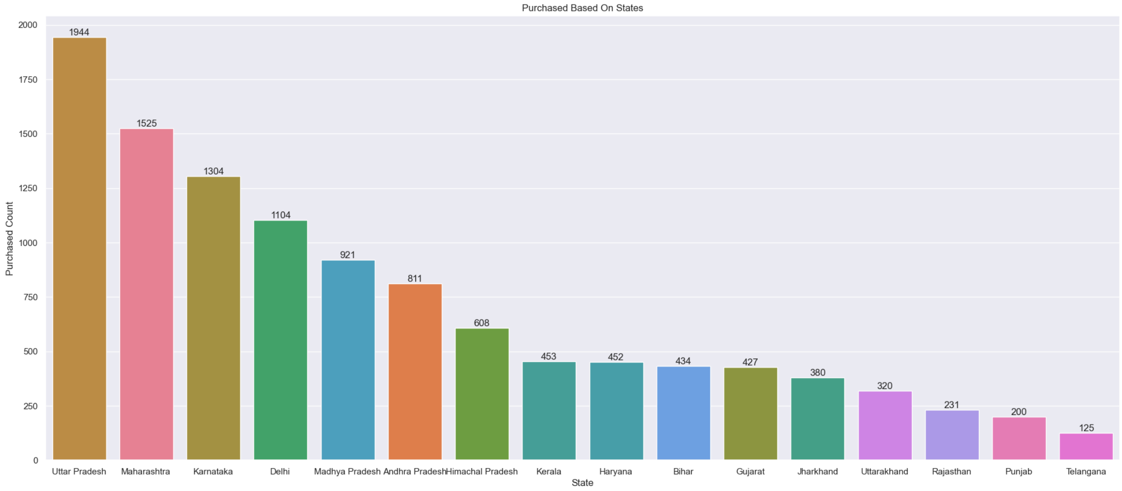
[144]:

```
sales_marital_status_gen=data.
  ↪groupby(['marital_status_readable','Gender'],as_index=False)['Amount'].sum().
  ↪sort_values(by='Amount',ascending=False)
sns.set(rc={'figure.figsize':(5,7)})
ax=sns.
  ↪barplot(x='marital_status_readable',y='Amount',hue='Gender',data=sales_marital_status_gen)
plt.title('Purchased Based On Marital status & Gender')
plt.xlabel('Marital Status & Gender')
plt.ylabel('Amount')
for bar in ax.containers:
    ax.bar_label(bar)
```



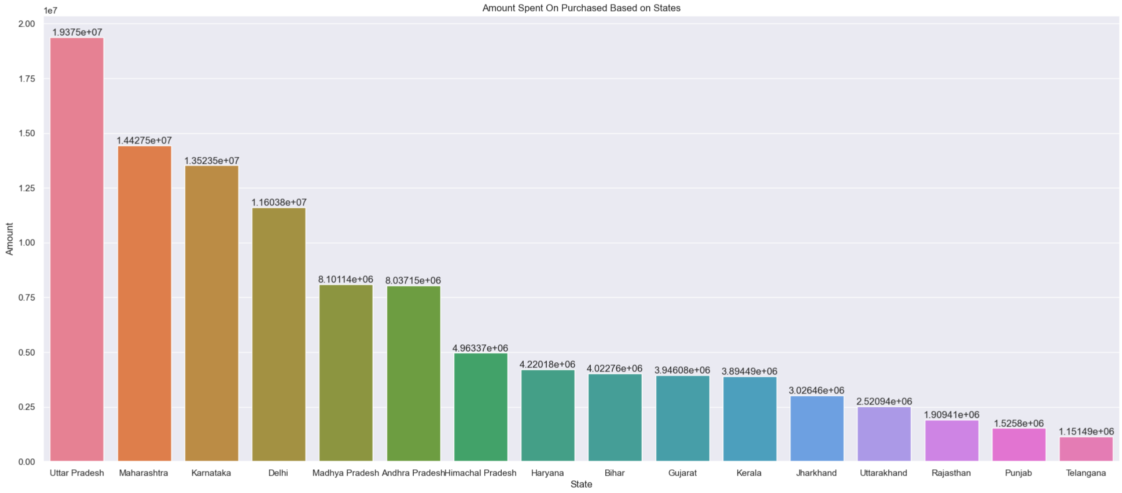
From above graph we can say that most buyers are married women

[16]:
```python
sns.set(rc={'figure.figsize':(24,10)})
state_counts=data['State'].value_counts()
state_sorted=state_counts.index
ax=sns.countplot(x='State',order=state_sorted,hue='State',data=data)
plt.title('Purchased Based On States')
plt.ylabel('Purchased Count')
for bar in ax.containers:
    ax.bar_label(bar)
```



[180]:

```
sales_states=data.groupby(['State'],as_index=False)['Amount'].sum().
  ↪sort_values(by='Amount',ascending=False)
ax=sns.barplot(x='State',y='Amount',hue='State',data=sales_states)
plt.title('Amount Spent On Purchased Based on States')
for i in ax.containers:
    ax.bar_label(i)
```

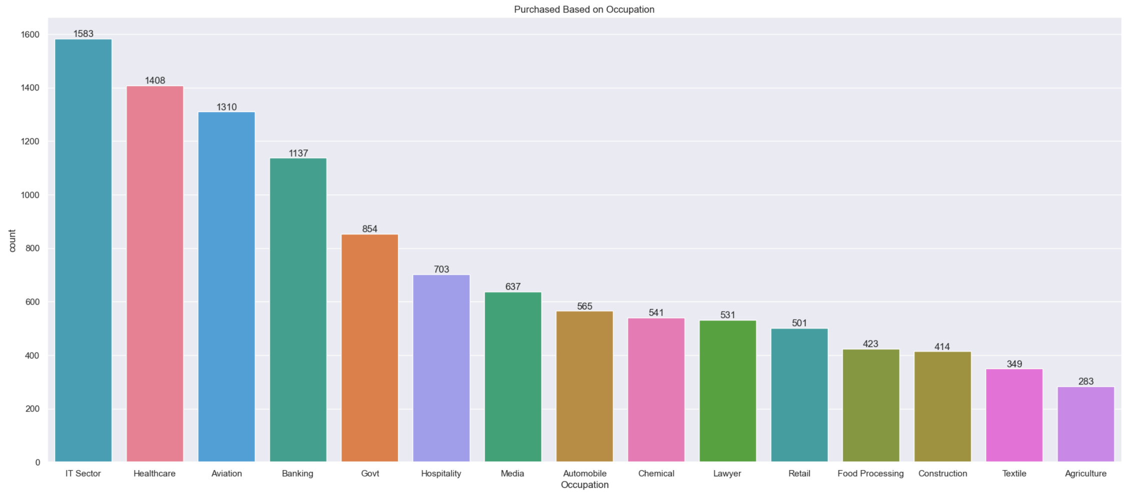


From above two graph we can say that Most purchased and amout spent are from Uttar Pradesh,Maharashtra and Karnataka respectively.

[187]:

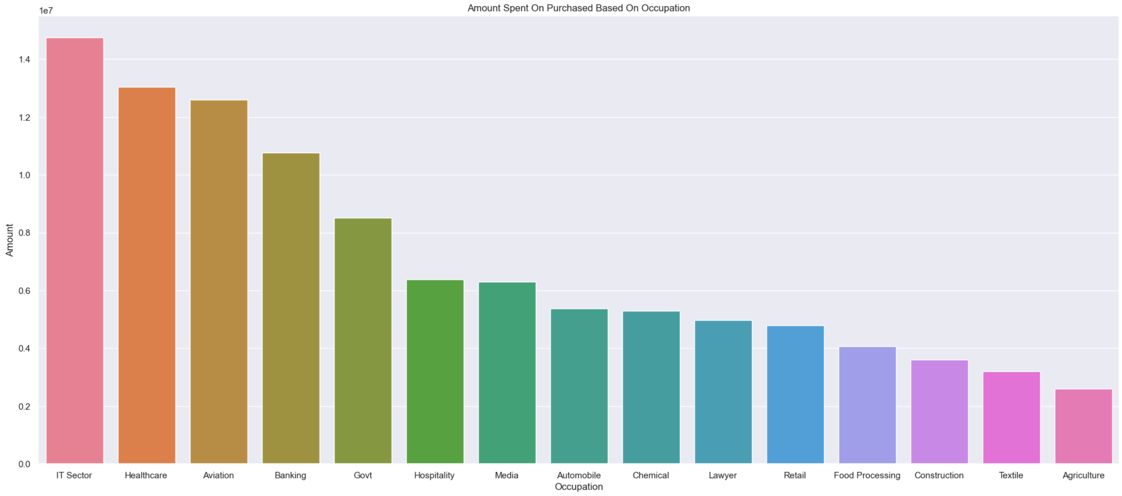
```
occp_counts=data['Occupation'].value_counts().sort_values(ascending=False)
occp_counts_sort=occp_counts.index
```

```
ax=sns.
 ↪countplot(x='Occupation',hue='Occupation',order=occp_counts_sort,data=data)
plt.title('Purchased Based on Occupation')
for i in ax.containers:
    ax.bar_label(i)
```



```
[17]: sal_occp=data.groupby(['Occupation'],as_index=False)['Amount'].sum().
       ↪sort_values(by="Amount",ascending=False)
      ax=sns.barplot(x='Occupation',y='Amount',hue='Occupation',data=sal_occp)
      plt.title('Amount Spent On Purchased Based On Occupation')
```

```
[17]: Text(0.5, 1.0, 'Amount Spent On Purchased Based On Occupation')
```



From above two graph we can say that peoples who made Most purchased and spent most amount on orders are from IT Sector,Healthcare and Aviation industry respectively.

[13]:

```
data.head()
```

[13]:

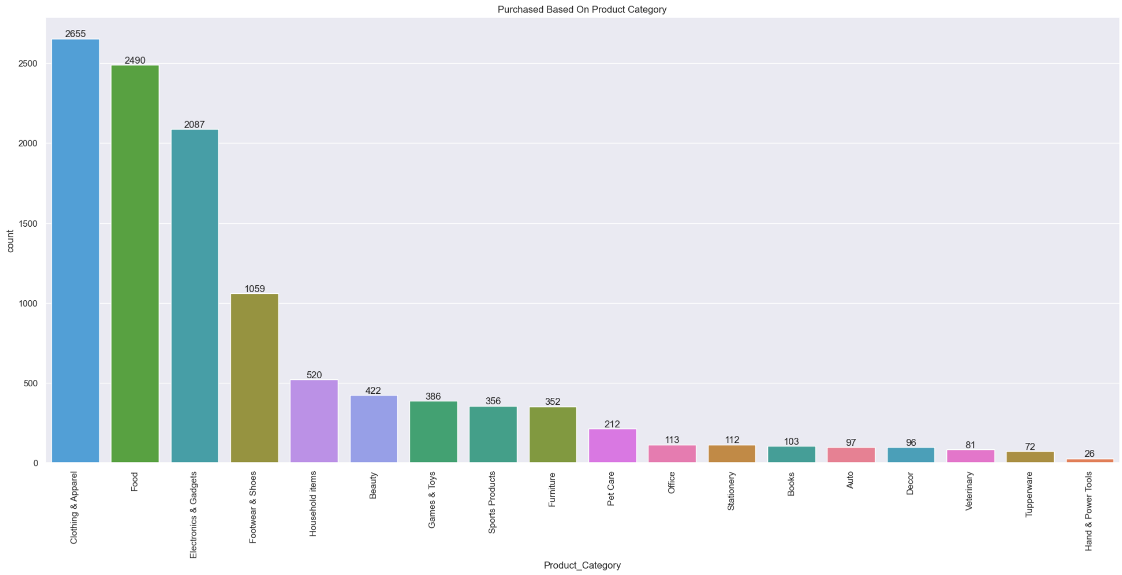
```
   User_ID  Cust_name Product_ID Gender Age Group  Age  Marital_Status  \
0  1002903  Sanskriti  P00125942      F     26-35   28               0
1  1000732     Kartik  P00110942      F     26-35   35               1
2  1001990      Bindu  P00118542      F     26-35   35               1
3  1001425     Sudevi  P00237842      M      0-17   16               0
4  1000588       Joni  P00057942      M     26-35   28               1

            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

    Amount marital_status_readable
0  23952.0                 Married
1  23934.0                  Single
2  23924.0                  Single
3  23912.0                 Married
4  23877.0                  Single
```

[30]:

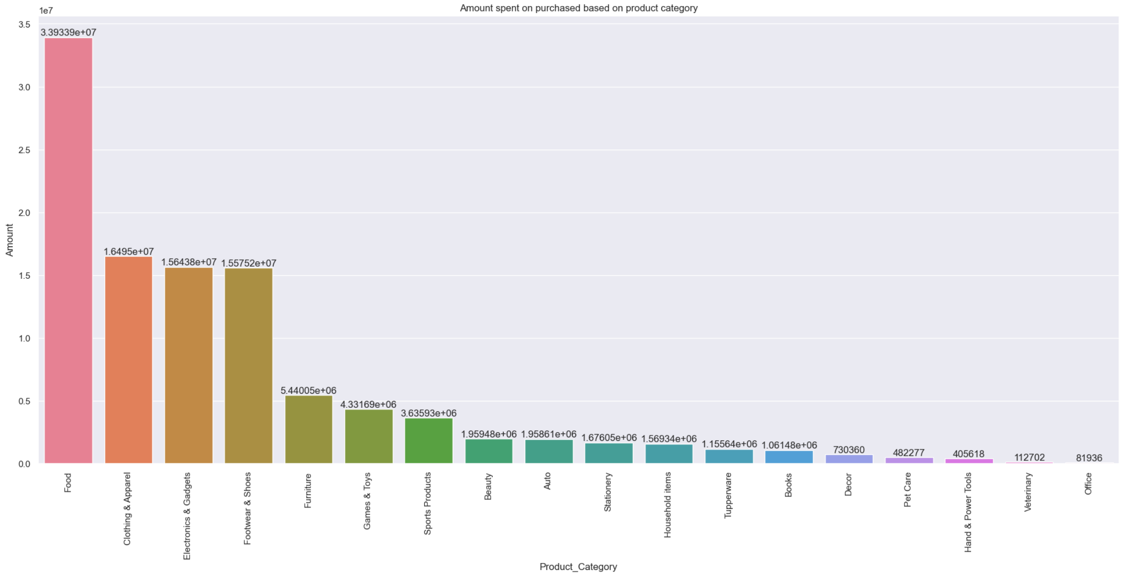
```
prd_ctg=data['Product_Category'].value_counts()
prd_ctg_sort=prd_ctg.index
ax=sns.
 ↪countplot(x='Product_Category',order=prd_ctg_sort,hue='Product_Category',data=data)
plt.xticks(rotation=90)
for bar in ax.containers:
    ax.bar_label(bar)
plt.title('Purchased Based On Product Category')
```

[30]:

```
Text(0.5, 1.0, 'Purchased Based On Product Category')
```



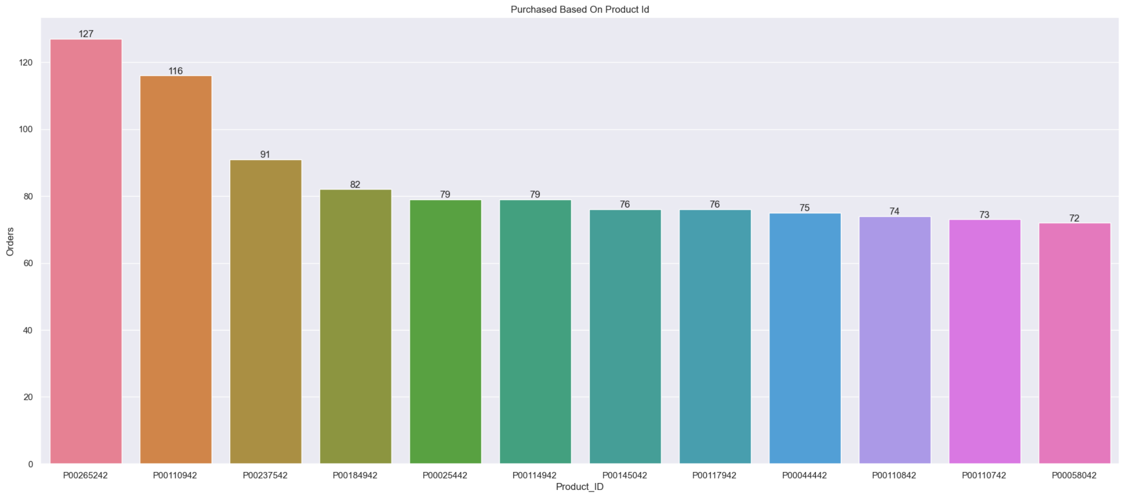
```
[38]: sal_prd_ctg=data.groupby(['Product_Category'],as_index=False)['Amount'].sum().
        ↪sort_values(by='Amount',ascending=False)
      ax=sns.
        ↪barplot(x='Product_Category',y='Amount',hue='Product_Category',data=sal_prd_ctg)
      plt.title('Amount spent on purchased based on product category')
      plt.xticks(rotation=90)
      for bar in ax.containers:
          ax.bar_label(bar)
```



From above graph we can say that top most purchased product category are Food,Clothing & Apparel and Electronics & Gadgets respectively.

[74]:

```
sale_prd_id=data.groupby(['Product_ID'],as_index=False)['Orders'].sum().
 ↪sort_values(by='Orders',ascending=False).head(12)
ax=sns.barplot(x='Product_ID',y='Orders',hue='Product_ID',data=sale_prd_id)
plt.title('Purchased Based On Product Id')
for bar in ax.containers:
    ax.bar_label(bar)
```



Top most purchased product with product id's are P00265242,P00110942 and P00237542 respectively.

[58]:

```
top_user=data['User_ID'].value_counts().head(5)
top_product
```

[58]:

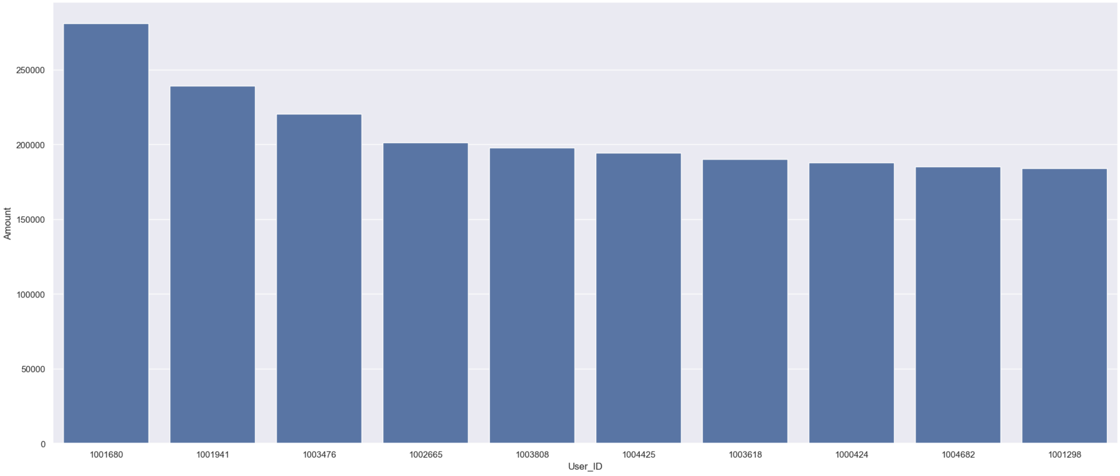
```
User_ID
1001680    24
1003808    23
1001941    22
1004425    20
1002665    19
1004682    19
1006036    19
1000424    19
1003476    19
1004725    18
Name: count, dtype: int64
```

Top customer with most purchased are 1001680,1003808 and 1001941 respectively.

```
[80]: Top_cust_amount=data.groupby(['User_ID'],as_index=False)['Amount'].sum().
        ↪sort_values(by='Amount',ascending=False).head(10)
      Top_cust_amount
```

```
[80]:       User_ID    Amount
      1045  1001680  281034.0
      1197  1001941  239147.0
      2134  1003476  220435.0
      1628  1002665  201104.0
      2355  1003808  197660.0
      2741  1004425  194343.0
      2226  1003618  189921.0
      274   1000424  187679.0
      2897  1004682  185122.0
      807   1001298  184045.0
```

```
[81]: ax=sns.
        ↪barplot(x='User_ID',y='Amount',data=Top_cust_amount,order=Top_cust_amount['User_ID'])
```



from above graph we can say that customers with customer_id's 1001680,1001941 and 1003476 respectively spend The most amount in purchasing product.

0.1 Conclusion:

0.1.1

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

github:

THANK YOU

[ ]:

```
```