DIWALI SALES ANALYSIS

Exploratory Data Analysis using Python

Prepared By Punit Pal

punitpalofficial@gmail.com

Project Overview & Objectives

The Diwali Sales Data Analysis project focuses on analyzing customer purchase data during the Diwali festive season using Python-based Exploratory Data Analysis (EDA). The dataset contains demographic and transaction details that help understand festive buying behavior. The project aims to transform raw sales data into meaningful business insights. Data cleaning, analysis, and visualization techniques are used to identify key trends. The primary objective is to study customer demographics, regional performance, and product preferences. This analysis supports data-driven decision-making for marketing and sales strategies.

punitpalofficial@gmail.com

Data Description & Methodology

The dataset used in this project is a structured CSV file containing 11,252 records and 13 attributes related to customers and sales. It includes variables such as gender, age group, state, occupation, orders, and purchase amount. The analysis followed a systematic methodology including data loading, data cleaning, and exploratory analysis. Missing values and unnecessary columns were handled to ensure data quality. Python libraries such as Pandas, NumPy, Matplotlib, and Seaborn were used throughout the analysis process.

punitpalofficial@gmail.com

[1]:

```
# import python libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt # visualizing data
%matplotlib inline
import seaborn as sns
```

[2]:

```
# import csv file 'Diwali Sales Data.csv'
df = pd.read_csv(r"C:\Users\punitpal\OneDrive\DA_Projects\Python_projects\Python
#to avoid encoding error, use 'unicode_escape' and use r for avoid special chara
```

[3]:

```
df.shape
```

[3]:

```
(11251, 15)
```

punitpalofficial@gmail.com

[4]:
```
df.head()
df.head(10)
```

[4]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount	Status	unname
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952.00	NaN	Na
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934.00	NaN	Na
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.00	NaN	Na
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.00	NaN	Na
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877.00	NaN	Na
5	1000588	Joni	P00057942	M	26-35	28	1	Himachal Pradesh	Northern	Food Processing	Auto	1	23877.00	NaN	Na
6	1001132	Balk	P00018042	F	18-25	25	1	Uttar Pradesh	Central	Lawyer	Auto	4	23841.00	NaN	Na
7	1002092	Shivangi	P00273442	F	55+	61	0	Maharashtra	Western	IT Sector	Auto	1	NaN	NaN	Na
8	1003224	Kushal	P00205642	M	26-35	35	0	Uttar Pradesh	Central	Govt	Auto	2	23809.00	NaN	Na
9	1003650	Ginny	P00031142	F	26-35	26	1	Andhra Pradesh	Southern	Media	Auto	4	23799.99	NaN	Na

punitpalofficial@gmail.com

[5]:

```
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
```

punitpalofficial@gmail.com

[6]:

```
#drop unrelated/blank columns
df.drop(['Status', 'unnamed1'], axis=1, inplace=True)
```

[7]:

```
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
```

punitpalofficial@gmail.com

[8]:
```
#check null values
pd.isnull(df)
```

[8]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount
0	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...
11246	False	False	False	False	False	False	False	False	False	False	False	False	False
11247	False	False	False	False	False	False	False	False	False	False	False	False	False
11248	False	False	False	False	False	False	False	False	False	False	False	False	False
11249	False	False	False	False	False	False	False	False	False	False	False	False	False
11250	False	False	False	False	False	False	False	False	False	False	False	False	False

11251 rows × 13 columns

punitpalofficial@gmail.com

[9]:

```
#check for null values
pd.isnull(df).sum()
```

[9]:

```
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
```

[10]:

```
# drop null values
df.dropna(inplace=True)
```

[11]:

```
df.shape
```

[11]:

```
(11239, 13)
```

[12]:

```
# change data type
df['Amount'] = df['Amount'].astype('int')
```

[13]:

```
df['Amount'].dtypes
```

[13]:

```
dtype('int64')
```

punitpalofficial@gmail.com

[14]:

```
df.columns
```

[14]:

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

punitpalofficial@gmail.com

[15]:
```
#rename column
df.rename(columns= {'Marital_Status':'Shaadi'})
```

[15]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Shaadi	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	4	370
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	3	367
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	4	213
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3	206
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	3	188

11239 rows × 13 columns

punitpalofficial@gmail.com

•[16]:

```
# describe() method returns description of the data
#in the DataFrame (i.e. count, mean, std, etc)
df.describe()
```

[16]:

	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

punitpalofficial@gmail.com

•[17]:

```
#describe() method returns description of the data
#in the DataFrame (i.e. count, mean, std, etc)
df.describe()
```

[17]:

	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

punitpalofficial@gmail.com

[18]:

```
# use describe() for specific columns
df[['Age', 'Orders', 'Amount']].describe()
```

[18]:

	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

punitpalofficial@gmail.com

Exploratory Data Analysis

Gender

[19]:

```
df.columns
```

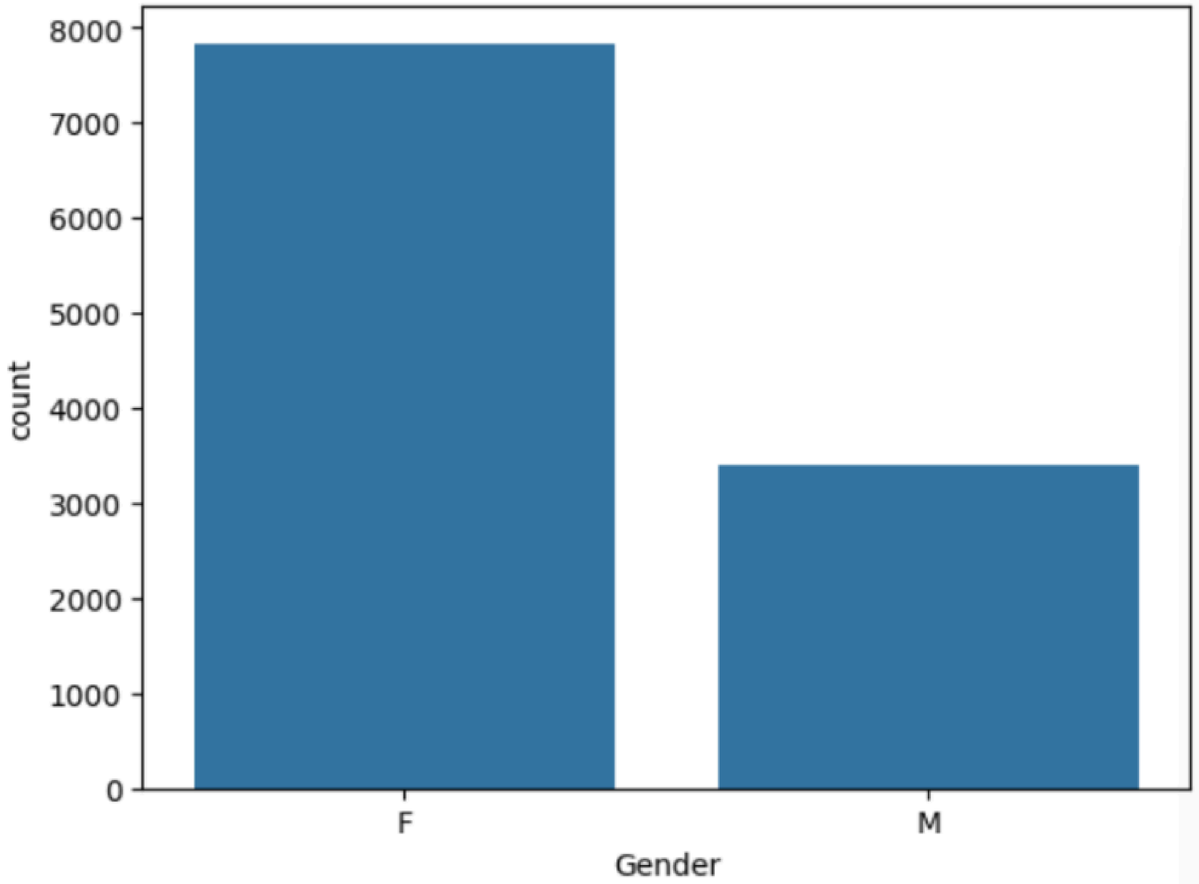
[19]:

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

punitpalofficial@gmail.com

[20]:

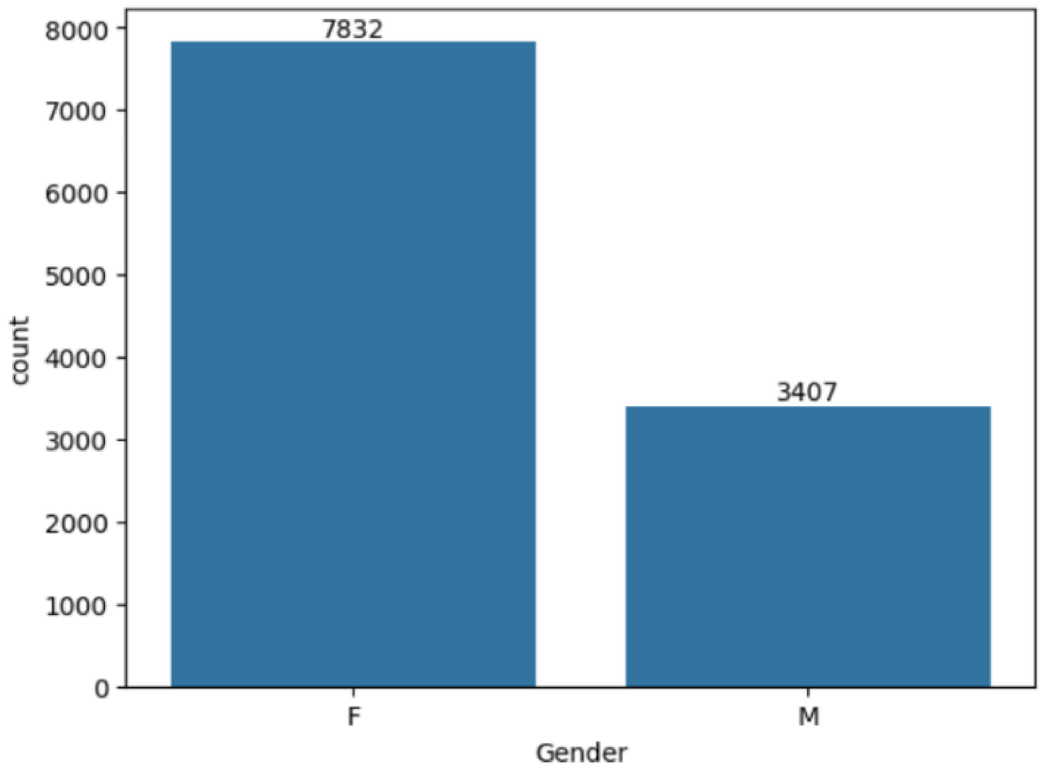
```
ax = sns.countplot(x = 'Gender',data = df)
```



punitpalofficial@gmail.com

[21]:

```
# plotting a bar chart for Gender and it's count

ax = sns.countplot(x = 'Gender',data = df)

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



punitpalofficial@gmail.com

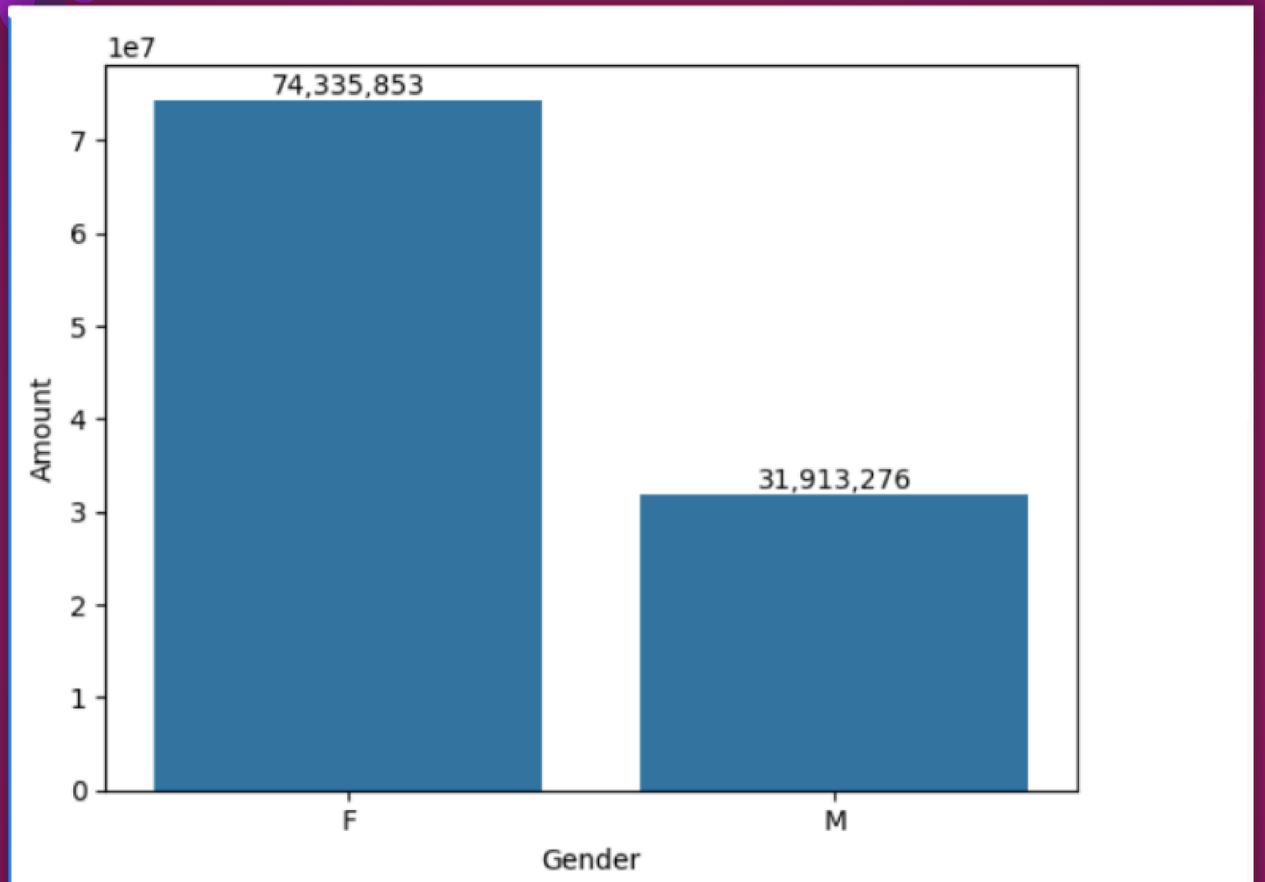
[37]:
```
df.groupby(['Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)
```

[37]:

	Gender	Amount
0	F	74335853
1	M	31913276

[23]:
```
# plotting a bar chart for gender vs total amount

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 p in ax.patches:
    ax.annotate(
        f'{p.get_height():,.0f}',           # value (comma format)
        (p.get_x() + p.get_width() / 2,     # x position
         p.get_height()),                   # y position
        ha='center',
        va='bottom',
        # fontsize=10,
        # fontweight='bold'
    )

plt.show()
```

punitpalofficial@gmail.com



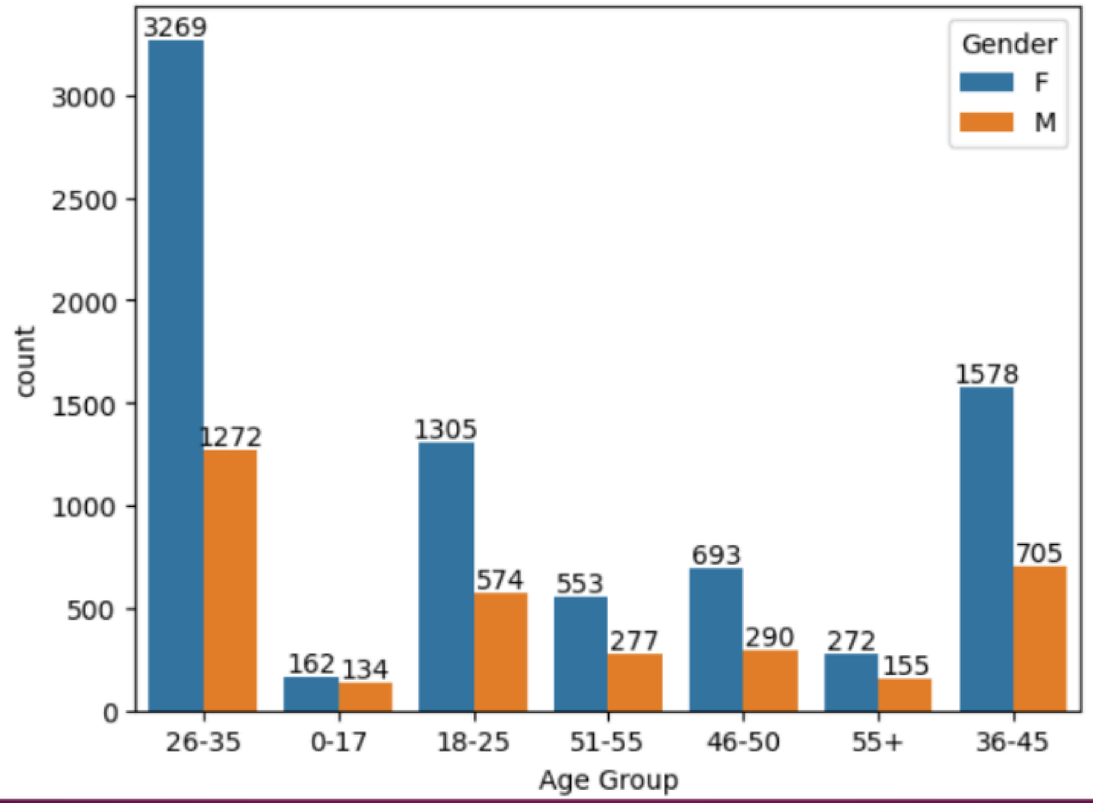
From above graphs we can see that most of the buyers are females and even the purchasing power of females are greater than men

punitpalofficial@gmail.com

Age

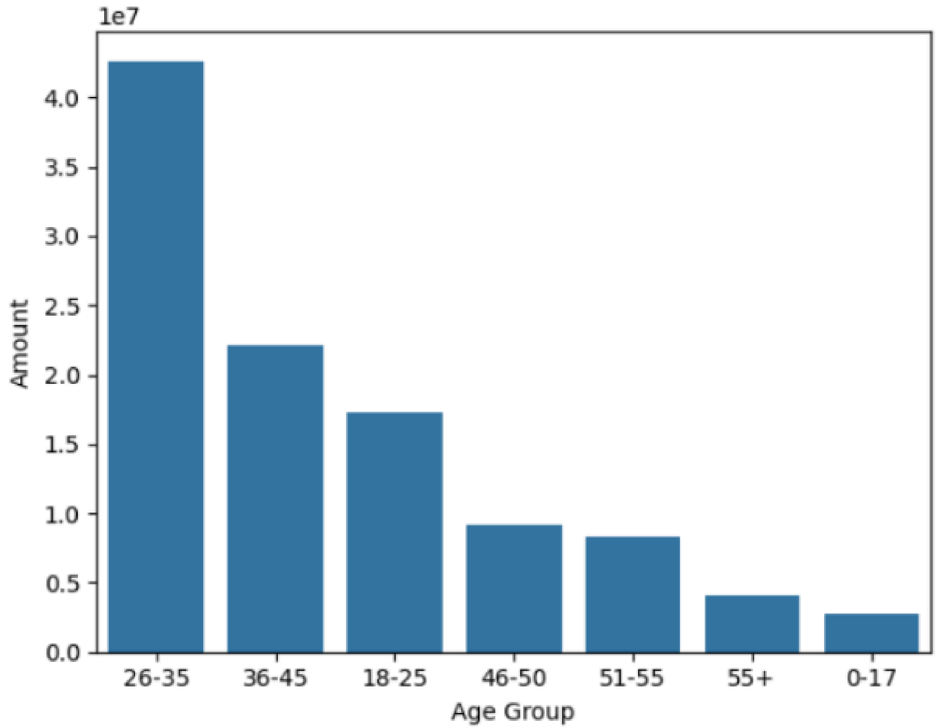
[24]:

```
ax = sns.countplot(data = df, x = 'Age Group', hue = 'Gender')

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



punitpalofficial@gmail.com

[25]:
```
# Total Amount vs Age Group
sales_age = df.groupby(['Age Group'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)

sns.barplot(x = 'Age Group',y= 'Amount' ,data = sales_age)

plt.show()
```

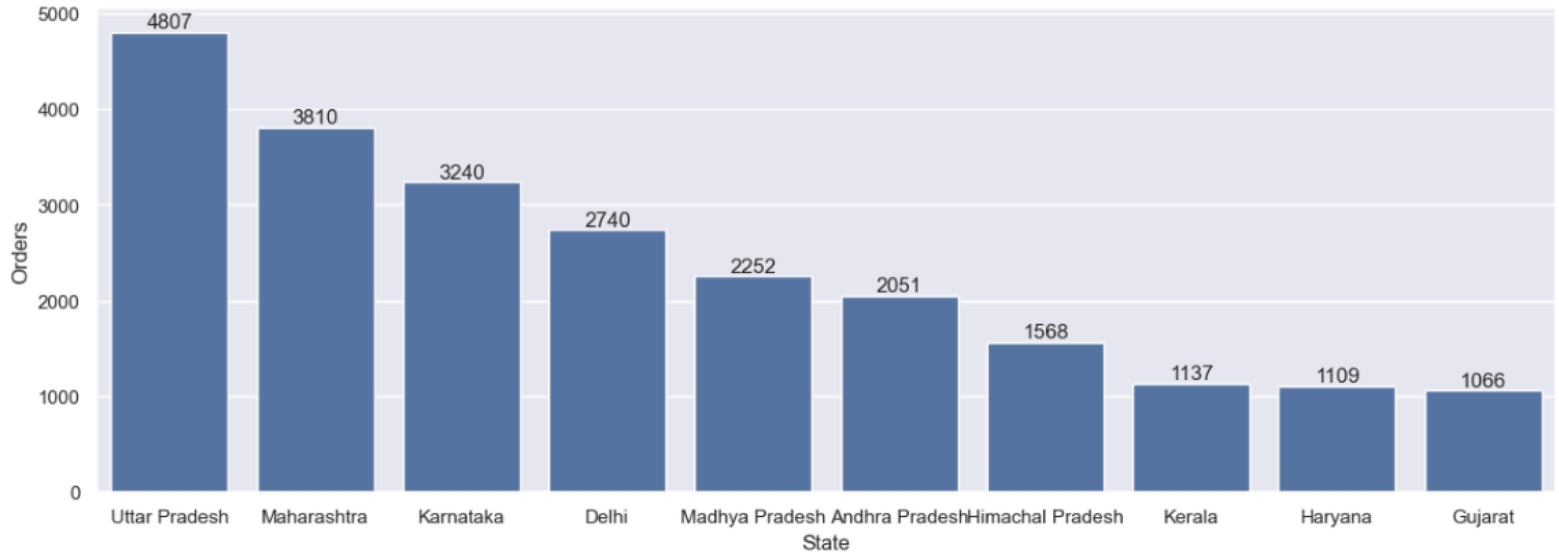


From above graphs we can see that most of the buyers are of age group between 26-35 yrs female

punitpalofficial@gmail.com

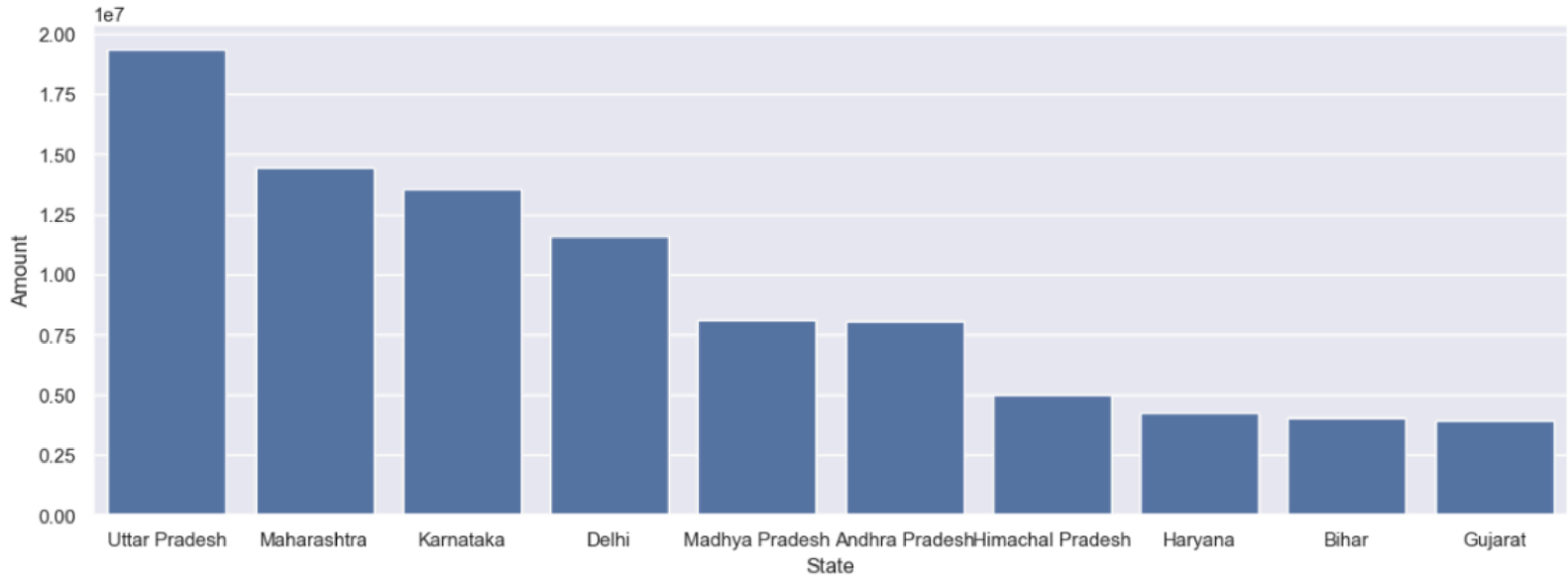
State

[26]:
```
# 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(data = sales_state, x = 'State',y= 'Orders')
for bars in ax.containers:
    ax.bar_label(bars)
```



punitpalofficial@gmail.com

[27]:
```
# total amount/sales from top 10 states

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(data = sales_state, x = 'State',y= 'Amount')
```

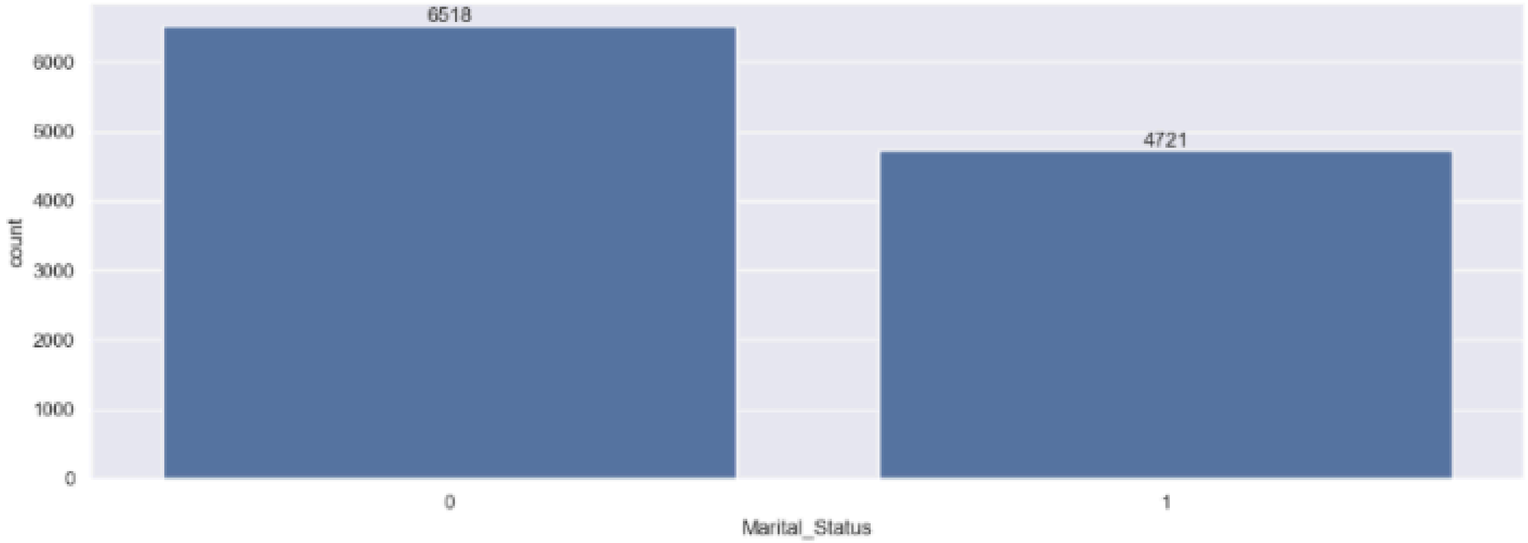


From above graphs we can see that most of the orders & total sales/amount are from Uttar Pradesh, Maharashtra and Karnataka respectively

punitpalofficial@gmail.com

Marital Status

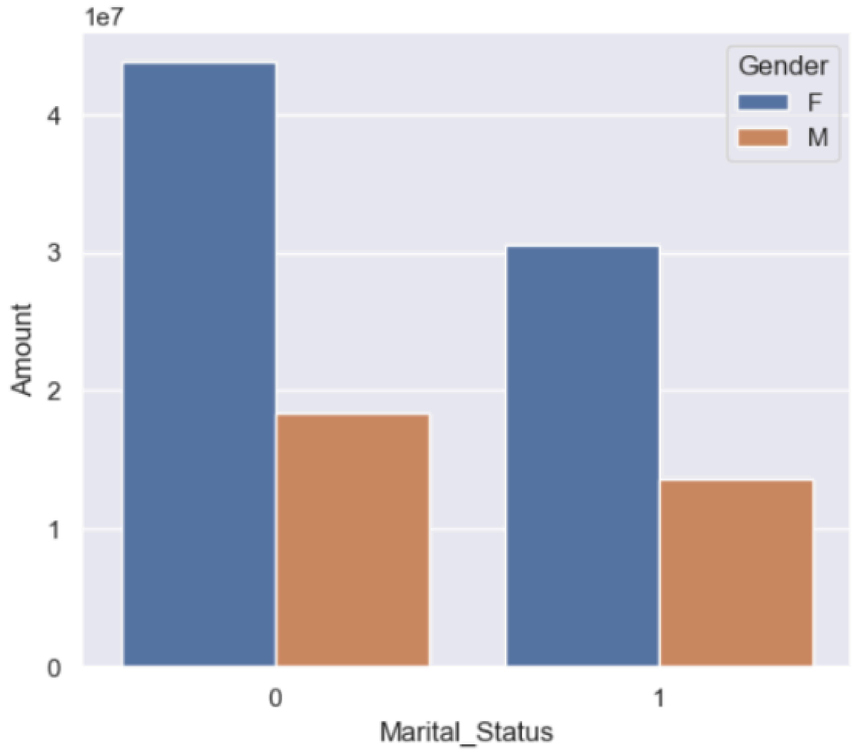
[28]:
```
ax = sns.countplot(data = df, x = 'Marital_Status')

sns.set(rc={'figure.figsize':(7,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



punitpalofficial@gmail.com

[29]:
```
sales_state = df.groupby(['Marital_Status', 'Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)

sns.set(rc={'figure.figsize':(6,5)})
sns.barplot(data = sales_state, x = 'Marital_Status',y= 'Amount', hue='Gender')
```

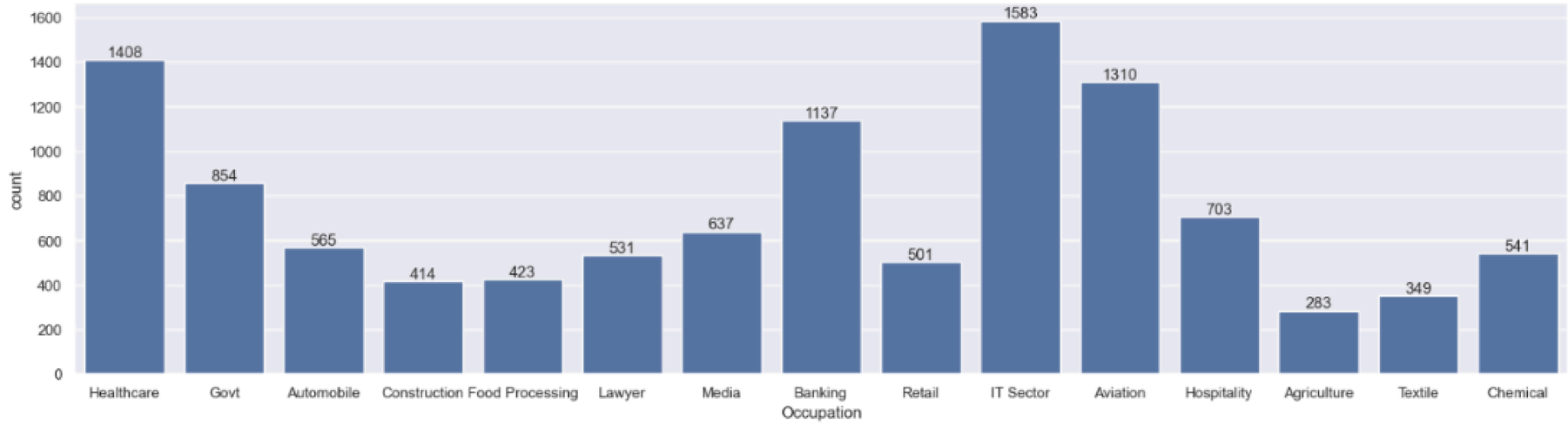
[29]:
```
<Axes: xlabel='Marital_Status', ylabel='Amount'>
```



From above graphs we can see that most of the buyers are married (women) and they have high purchasing power

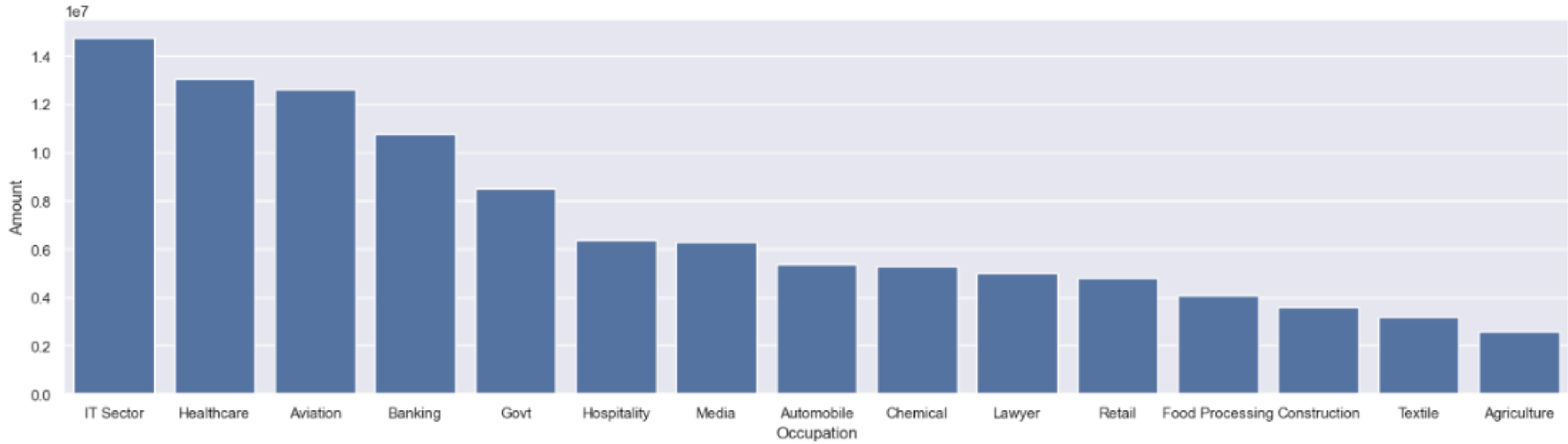
Occupation

[30]:
```
sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data = df, x = 'Occupation')

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



punitpalofficial@gmail.com

[31]:
```
sales_state = df.groupby(['Occupation'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)

sns.set(rc={'figure.figsize':(20,5)})
ax=sns.barplot(data = sales_state, x = 'Occupation',y= 'Amount')
```

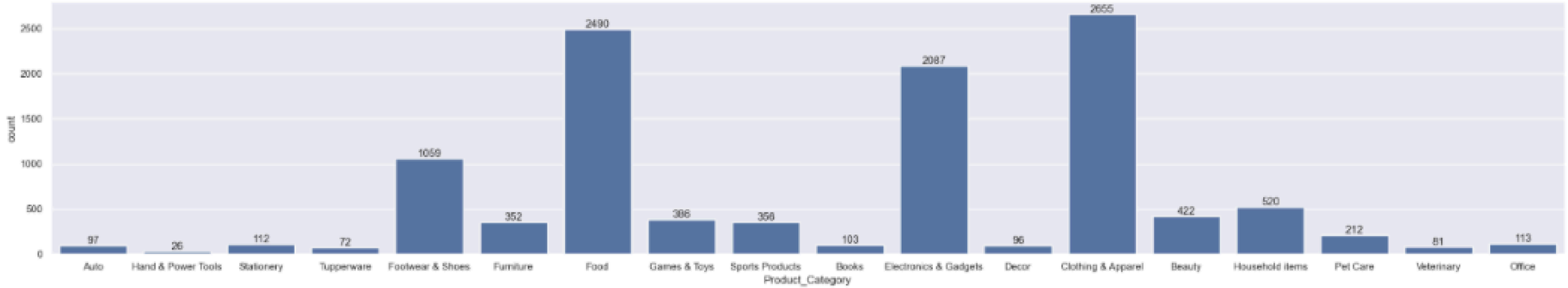


From above graphs we can see that most of the buyers are working in IT, Healthcare and Aviation sector

punitpalofficial@gmail.com

Product Category

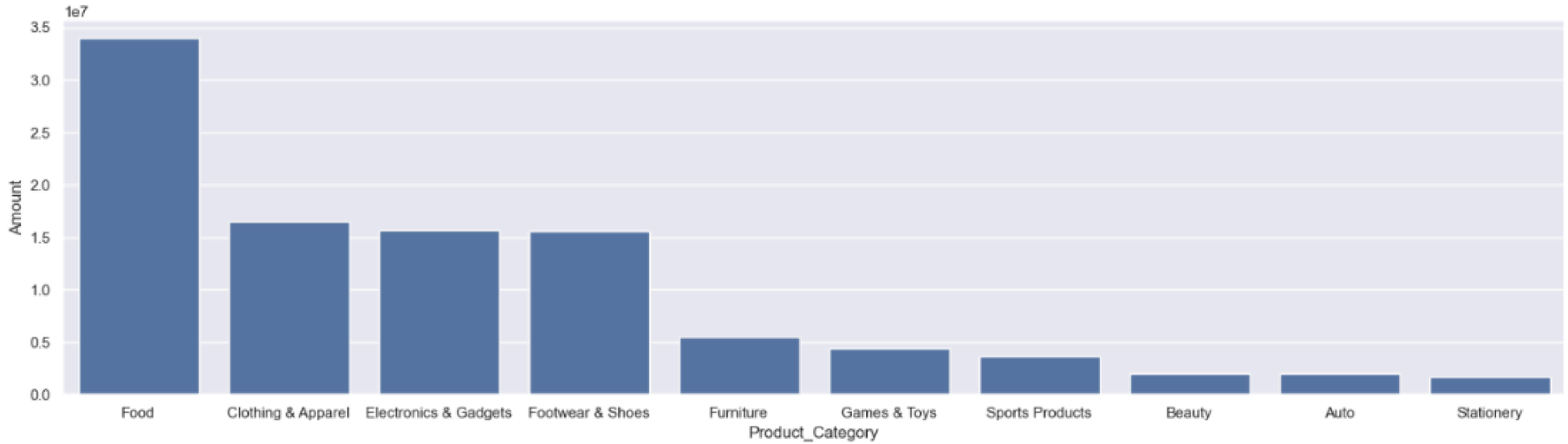
[32]:
```
sns.set(rc={'figure.figsize':(30,5)})
ax = sns.countplot(data = df, x = 'Product_Category')

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



punitpalofficial@gmail.com

```
[33]: sales_state = df.groupby(['Product_Category'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False).head(10)

      sns.set(rc={'figure.figsize':(20,5)})
      sns.barplot(data = sales_state, x = 'Product_Category',y= 'Amount')
```

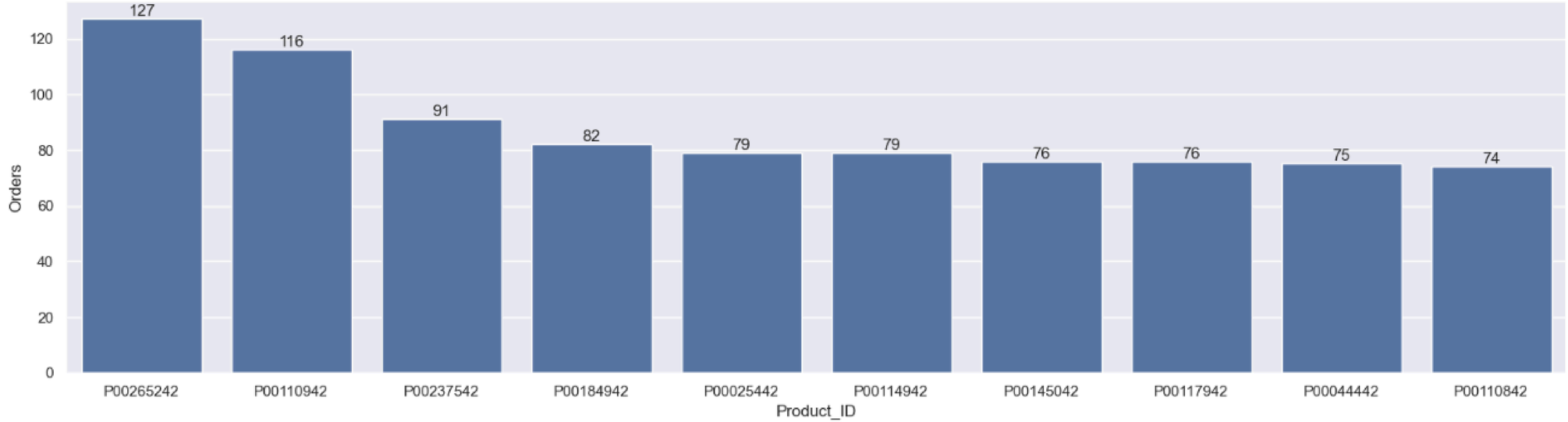
```
[33]: <Axes: xlabel='Product_Category', ylabel='Amount'>
```



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

punitpalofficial@gmail.com

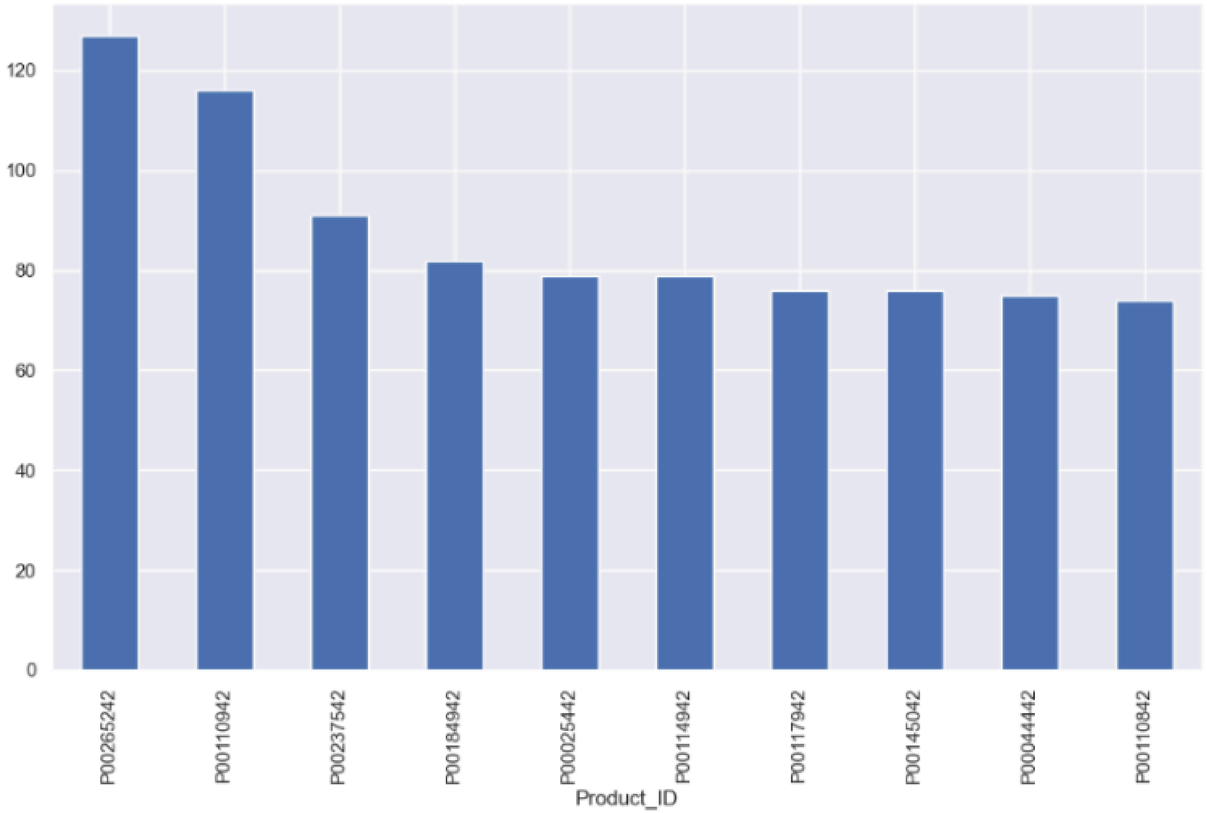
[34]:
```
sales_state = df.groupby(['Product_ID'], as_index=False)['Orders'].sum().sort_values(by='Orders', ascending=False).head(10)

sns.set(rc={'figure.figsize':(20,5)})
ax=sns.barplot(data = sales_state, x = 'Product_ID',y= 'Orders')
for bars in ax.containers:
    ax.bar_label(bars)
```



punitpalofficial@gmail.com

[35]:
```
# top 10 most sold products (same thing as above)

fig1, ax1 = plt.subplots(figsize=(12,7))
df.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascending=False).plot(kind='bar')
```

[35]: <Axes: xlabel='Product_ID'>



punitpalofficial@gmail.com

Analysis & Key Findings

The analysis revealed that female customers contributed more to total sales compared to male customers. The 26–35 age group emerged as the most active and high-spending customer segment. Sales were concentrated in a few states, mainly Uttar Pradesh, Maharashtra, and Karnataka. Customers from professional occupations showed higher purchasing behavior. Certain product categories consistently generated higher revenue during the festive season. These findings highlight strong demographic and regional sales patterns.

punitpalofficial@gmail.com

Analysis & Key Findings

The analysis revealed that female customers contributed more to total sales compared to male customers. The 26–35 age group emerged as the most active and high-spending customer segment. Sales were concentrated in a few states, mainly Uttar Pradesh, Maharashtra, and Karnataka. Customers from professional occupations showed higher purchasing behavior. Certain product categories consistently generated higher revenue during the festive season. These findings highlight strong demographic and regional sales patterns.

punitpalofficial@gmail.com

Conclusion & Business Recommendations

This project demonstrates how Python-based data analysis can extract valuable insights from festive sales data. The findings help identify high-value customers and key sales-driving factors. Businesses should focus marketing efforts on female customers and the 26–35 age group. High-performing states should receive targeted promotions and inventory support. Emphasis on high-revenue product categories can further improve sales performance. Overall, the project showcases an effective end-to-end data analysis workflow.

punitpalofficial@gmail.com

THANK YOU

punitpalofficial@gmail.com