

PROJECT TASK:

1. Total Transactions count by Gender Wise
2. Gender Wise Distribution
3. Gender Wise Total Sales Amount
4. Age Group Wise Transactions Count
5. Age Group Wise Total Amount
6. Top 5 State by Total Quantity Ordered
7. Top 5 State as per Total Sales Amount
8. Product Category Wise Transactions Count
9. Amount Wise Product Category
10. Product Category and Gender Wise Transactions Count
11. Profession Wise Transactions Count
12. Amount Wise Top Professions
13. Top 10 Product by Total Quantity Ordered

Let's Start Work

```
In [89]: # import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt # visualizing data
%matplotlib inline
import seaborn as sns
```

```
In [90]: # import csv file
df = pd.read_csv('Sales_Data.csv', encoding= 'unicode_escape')
```

```
In [91]: # Check Number of rows and Columns
df.shape
```

```
Out[91]: (11257, 13)
```

```
In [92]: # To see the imported data
df.head() #shows top 5 rows
```

```
Out[92]:
```

	User_ID	Cust_name	Product_ID	Age	Age Group	Gender	State	Zone	Zipcode	Profession	Product_Category	Order_ID
0	1002903.0	Anvi	P00125942	27.0	26-35	Female	Maharashtra	West	NaN	Healthcare	Sports	
1	1000732.0	Shanta	P00110942	34.0	26-35	Female	Andhra Pradesh	South	NaN	Govt	Sports	
2	1001990.0	Sheetal	P00118542	16.0	0-17	Female	Uttar Pradesh	Central	NaN	Automobile	Health	
3	1001425.0	Virendra	P00237842	16.0	0-17	M	Karnataka	South	NaN	Construction	Clothing	
4	1000588.0	Vishal	P00057942	28.0	26-35	M	Gujarat	West	NaN	Food Processing	Electronics	

```
In [93]: # Fill details and Data Type
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11257 entries, 0 to 11256
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   User_ID               11245 non-null  float64
 1   Cust_name             11245 non-null  object
 2   Product_ID           11245 non-null  object
 3   Age                  11245 non-null  float64
 4   Age Group            11245 non-null  object
 5   Gender               11245 non-null  object
 6   State               11245 non-null  object
 7   Zone                11245 non-null  object
 8   Zipcode              0 non-null      float64
 9   Profession           11245 non-null  object
10   Product_Category     11245 non-null  object
11   Orders              11245 non-null  float64
12   Amount              11245 non-null  float64
dtypes: float64(5), object(8)
memory usage: 1.1+ MB
```

Let's Start Data Cleaning

```
In [94]: # Deleting Blank Column
df.drop(['Zipcode'], axis=1, inplace=True)
```

```
In [95]: # list of Column Available
df.columns
```

```
Out[95]: Index(['User_ID', 'Cust_name', 'Product_ID', 'Age', 'Age Group', 'Gender',
              'State', 'Zone', 'Profession', 'Product_Category', 'Orders', 'Amount'],
              dtype='object')
```

```
In [96]: # Check for null values
pd.isnull(df).sum()
```

```
Out[96]: User_ID           12
Cust_name           12
Product_ID          12
Age                 12
Age Group           12
Gender              12
State               12
Zone                12
Profession          12
Product_Category    12
Orders              12
Amount              12
dtype: int64
```

```
In [97]: # Drop Null values
df.dropna(how='all', inplace=True)
```

```
In [98]: df.shape
```

```
Out[98]: (11245, 12)
```

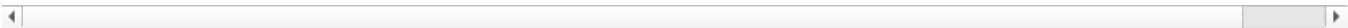
```
In [99]: # Replace value of Gender Column
df['Gender'] = df['Gender'].replace('M', 'Male')
```

```
In [100]: # View only Male Gender Data
df[df['Gender'] == 'Male']
```

Out[100..

	User_ID	Cust_name	Product_ID	Age	Age Group	Gender	State	Zone	Profession	Product_Category	Orders	A
3	1001425.0	Virendra	P00237842	16.0	0-17	Male	Karnataka	South	Construction	Clothing	6.0	2
4	1000588.0	Vishal	P00057942	28.0	26-35	Male	Gujarat	West	Food Processing	Electronics	4.0	2
5	1000588.0	Suuraj	P00057942	28.0	26-35	Male	Himachal Pradesh	Northern	Food Processing	Electronics	3.0	2
8	1003224.0	Kushal	P00205642	35.0	26-35	Male	Uttar Pradesh	Central	Govt	Beauty	4.0	2
11	1003829.0	Harsh	P00200842	34.0	26-35	Male	Delhi	Central	Banking	Health	2.0	2
...
11249	1005446.0	Sheetal	P00297742	53.0	51-55	Male	Gujarat	West	Healthcare	Health	3.0	
11250	1005446.0	Sheetal	P00297742	53.0	51-55	Male	Madhya Pradesh	Central	Healthcare	Health	2.0	
11252	1000695.0	Manning	P00296942	19.0	18-25	Male	Maharashtra	West	Chemical	Health	1.0	
11253	1004089.0	Reichenbach	P00171342	33.0	26-35	Male	Haryana	Northern	Healthcare	Health	5.0	
11255	1004023.0	Noonan	P00059442	37.0	36-45	Male	Karnataka	South	Agriculture	Clothing	4.0	

3408 rows × 12 columns



Exploratory Data Analysis

In [101..

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

Out[101..

	User_ID	Age	Orders	Amount
count	1.124500e+04	11245.000000	11245.000000	11245.000000
mean	1.003004e+06	35.415651	3.500311	9461.934237
std	1.716207e+03	12.756369	1.713706	5234.426634
min	1.000001e+06	12.000000	1.000000	188.000000
25%	1.001492e+06	27.000000	2.000000	5443.000000
50%	1.003065e+06	33.000000	4.000000	8109.000000
75%	1.004429e+06	43.000000	5.000000	12683.000000
max	1.006040e+06	92.000000	6.000000	29350.000000

In [102..

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

Out[102..

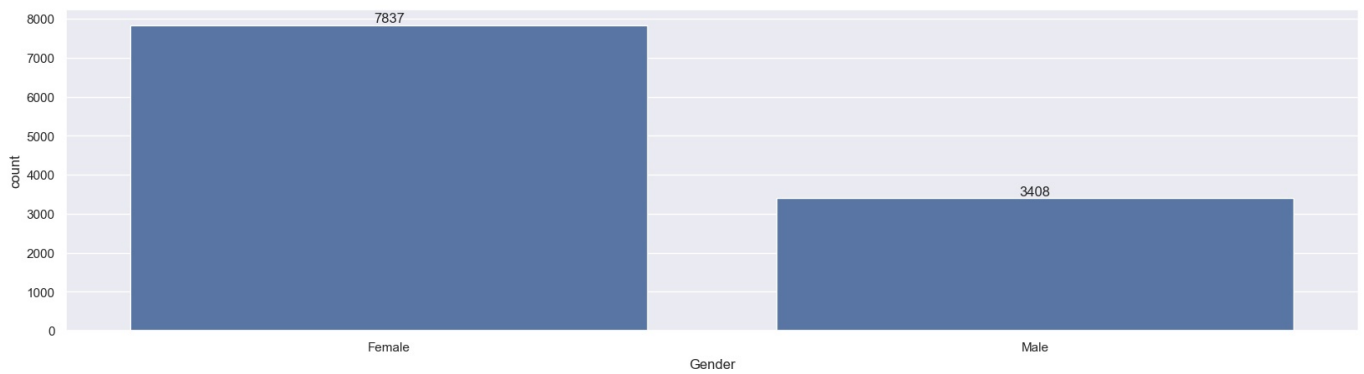
	Age	Orders	Amount
count	11245.000000	11245.000000	11245.000000
mean	35.415651	3.500311	9461.934237
std	12.756369	1.713706	5234.426634
min	12.000000	1.000000	188.000000
25%	27.000000	2.000000	5443.000000
50%	33.000000	4.000000	8109.000000
75%	43.000000	5.000000	12683.000000
max	92.000000	6.000000	29350.000000

1. Total Transactions Count by Gender Wise

In [103..

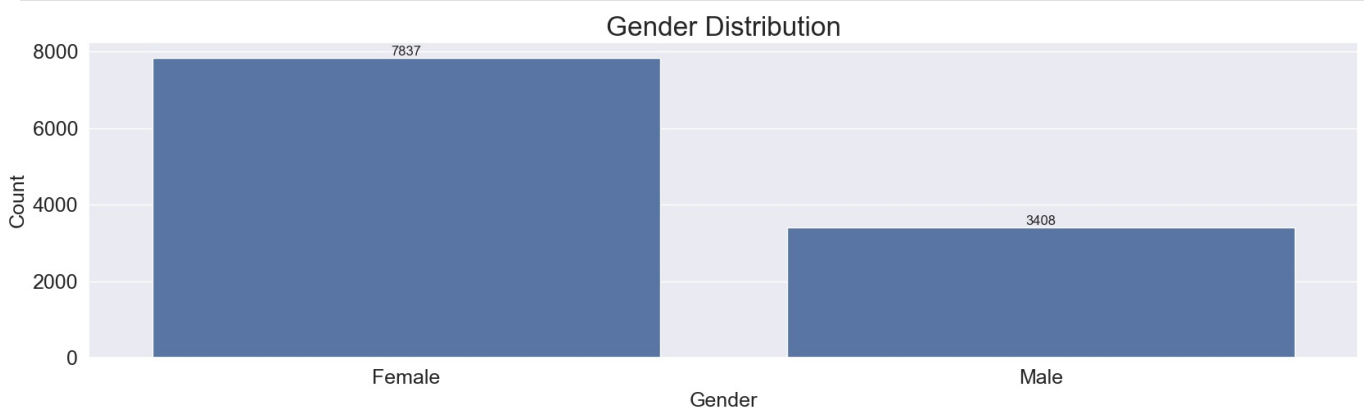
```
ax=sns.countplot(x='Gender', data = df)

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



```
In [104... ax = sns.countplot(x = 'Gender', data = df)

#Set Title and Labels with font size
for bars in ax.containers:
    ax.bar_label(bars)
    ax.set_title("Gender Distribution", fontsize=24) #chart title
    ax.set_xlabel('Gender', fontsize=18) #X-axis label
    ax.set_ylabel('Count', fontsize=18) #Y-axis label
    ax.tick_params(axis='both', labelsize=18) #Axis Parameters
plt.show()
```



2. Gender Wise Distribution

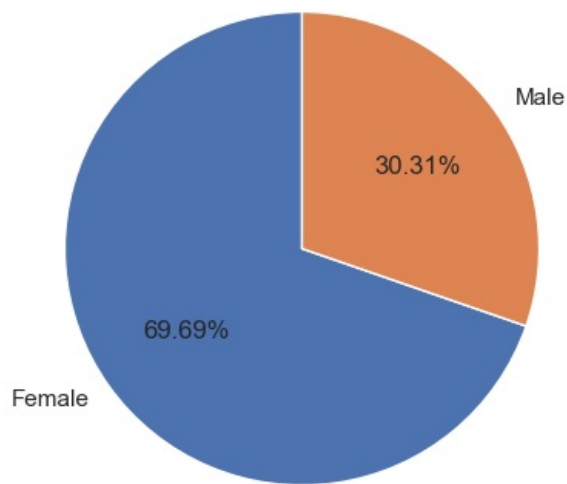
```
In [105... # Total Transactions Count by Gender Wise
gender_counts = df['Gender'].value_counts()
gender_counts
```

```
Out[105... Gender
Female    7837
Male      3408
Name: count, dtype: int64
```

```
In [106... # Total Transactions Count by Gender Wise
gender_counts = df['Gender'].value_counts()

plt.pie(
    gender_counts,                # Data (counts)
    labels = gender_counts.index, # Labels (e.g. Male, Female)
    autopct= '%.2f%%',           # Show Percentage (2 Decimal Place)
    startangle=90,               # Rotate for Better Orientation
)
# Add Title
plt.title("Gender Distribution", fontsize=24)
plt.show()
```

Gender Distribution



3. Gender Wise Total Sales Amount

```
In [107...] df.groupby('Gender', as_index=False)['Amount'].sum() # Return Total Amount by Gender Wise
```

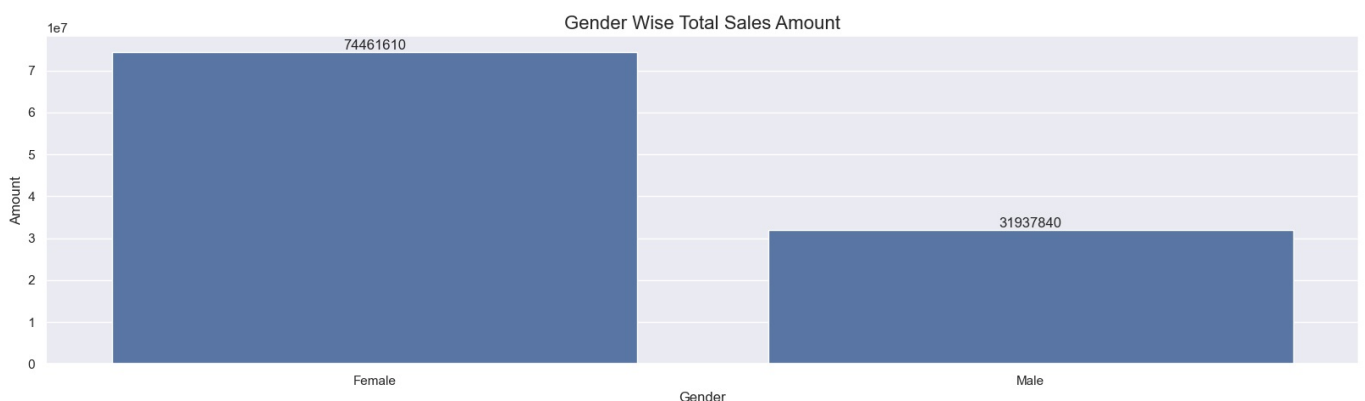
```
Out[107...]
   Gender  Amount
0  Female 74461610.49
1   Male 31937840.00
```

```
In [108...] # Gender Wise Total Sales Amount
# Step 1: Group data by Gender and sum the Amount
Gen_Wise_Sales = df.groupby('Gender', as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)

# Step 2: Create The Bar plot
ax = sns.barplot(x='Gender', y='Amount', data=Gen_Wise_Sales)

# Step 3: Add Data labels on top of bars
for bar in ax.containers:
    ax.bar_label(bar, fmt='%.0f', fontsize=12) # Show counts as whole numbers

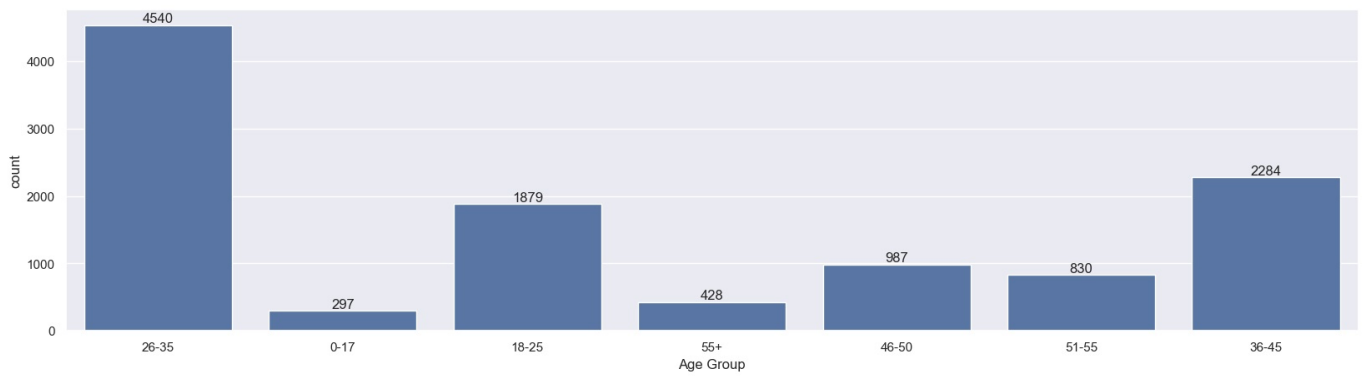
# Step 4: Add titles
plt.title("Gender Wise Total Sales Amount", fontsize=16)
plt.show()
```



4. Age Group Wise Total Transactions

```
In [109...] # Age Group wise Transactions Count
ax = sns.countplot(data=df, x = 'Age Group')

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

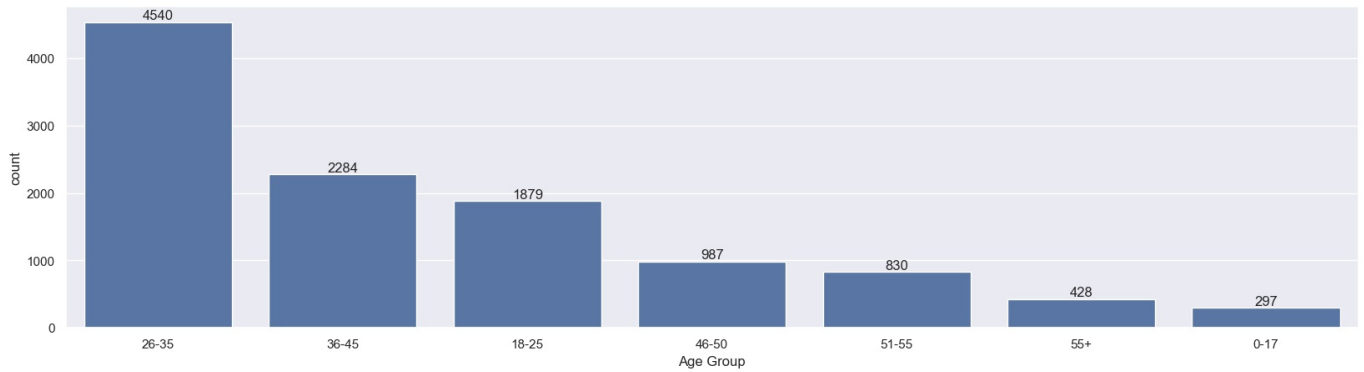


```
In [110]: #step 1: Age Group Wise Transactions Count (Sorting)
age_group_counts=df['Age Group'].value_counts().sort_values(ascending=False)

#Step 2: Use Ordered Categories for sorting
sns_order = age_group_counts.index

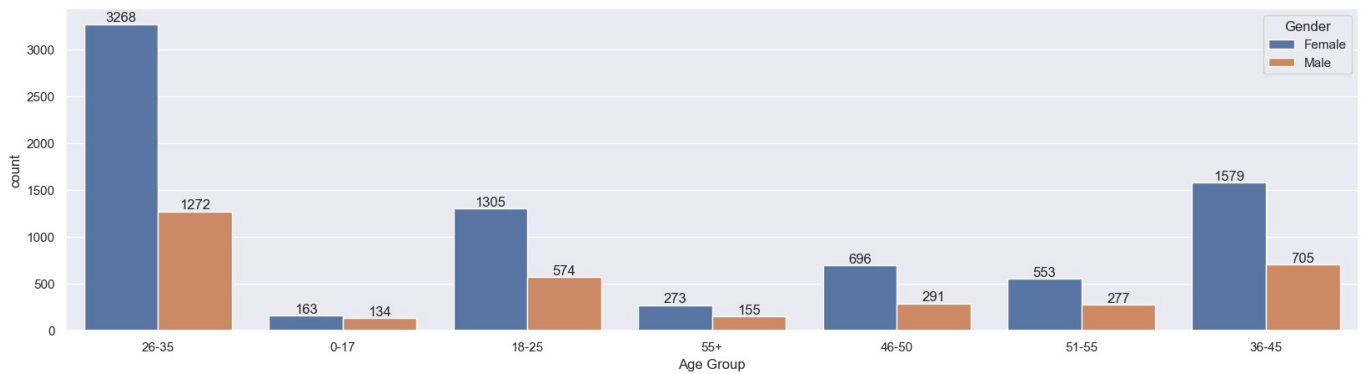
#Step 3:# Age Group wise Transactions Count
ax = sns.countplot(data=df, x = 'Age Group', order = sns_order)

for bars in ax.containers: #Show Data Labels
    ax.bar_label(bars)
plt.show()
```



```
In [111]: # Age Group and Gender Wise Distribution
ax = sns.countplot(data=df, x = 'Age Group', hue = 'Gender')

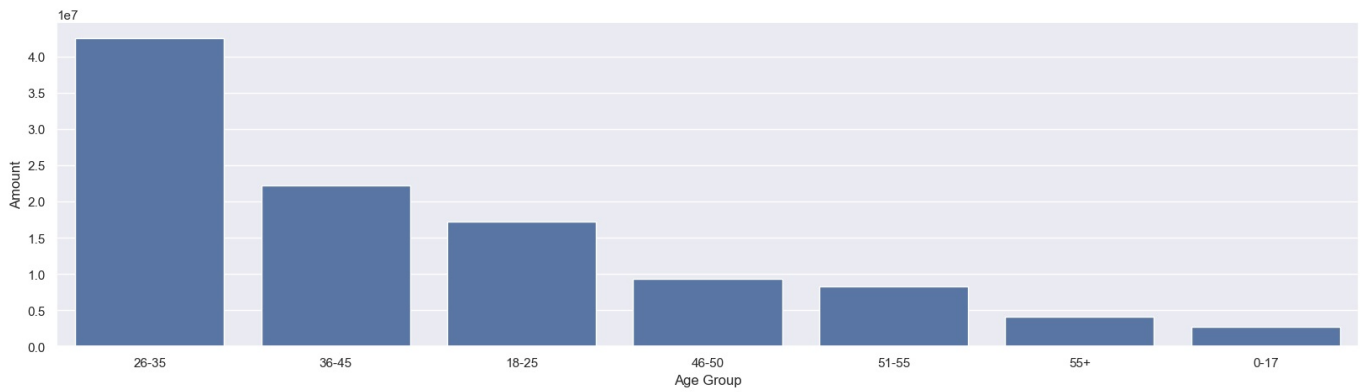
for bars in ax.containers: #Show Data Labels
    ax.bar_label(bars)
plt.show()
```



5. Age Group Wise Total Amount

```
In [112]: # Age Group Wise Total Amount
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()
```



6. Top 5 State by Total Quantity Ordered

```
In [113.. order_state = df.groupby(['State'], as_index=False)['Orders'].sum().sort_values(by='Orders', ascending=False).head(5)
order_state
```

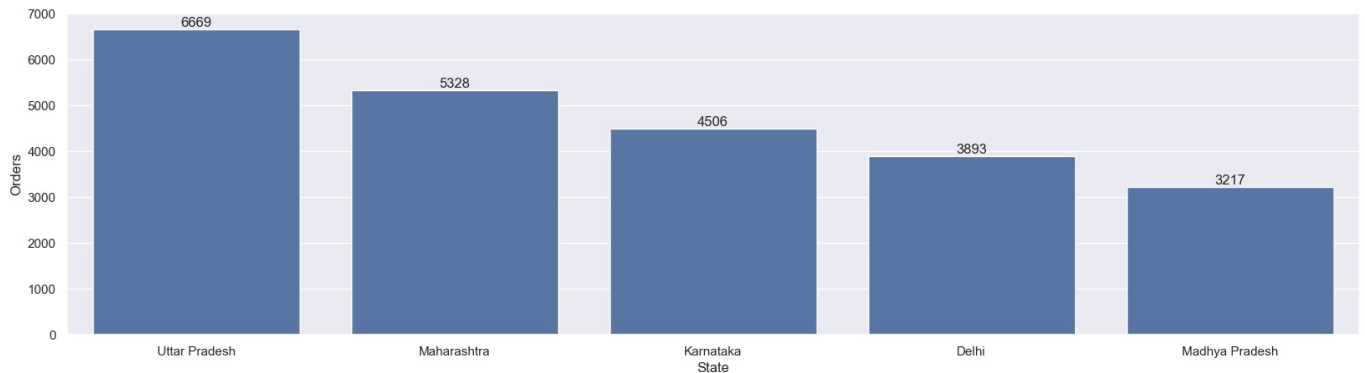
```
Out[113..
```

	State	Orders
14	Uttar Pradesh	6669.0
10	Maharashtra	5328.0
7	Karnataka	4506.0
2	Delhi	3893.0
9	Madhya Pradesh	3217.0

```
In [114.. # Order Wise Top 5 State
order_state = df.groupby(['State'], as_index=False)['Orders'].sum().sort_values(by='Orders', ascending=False).head(5)

#sns.set(rc={'figure.figsize':(15,5)})
ax = sns.barplot(data=order_state, x='State', y='Orders')

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

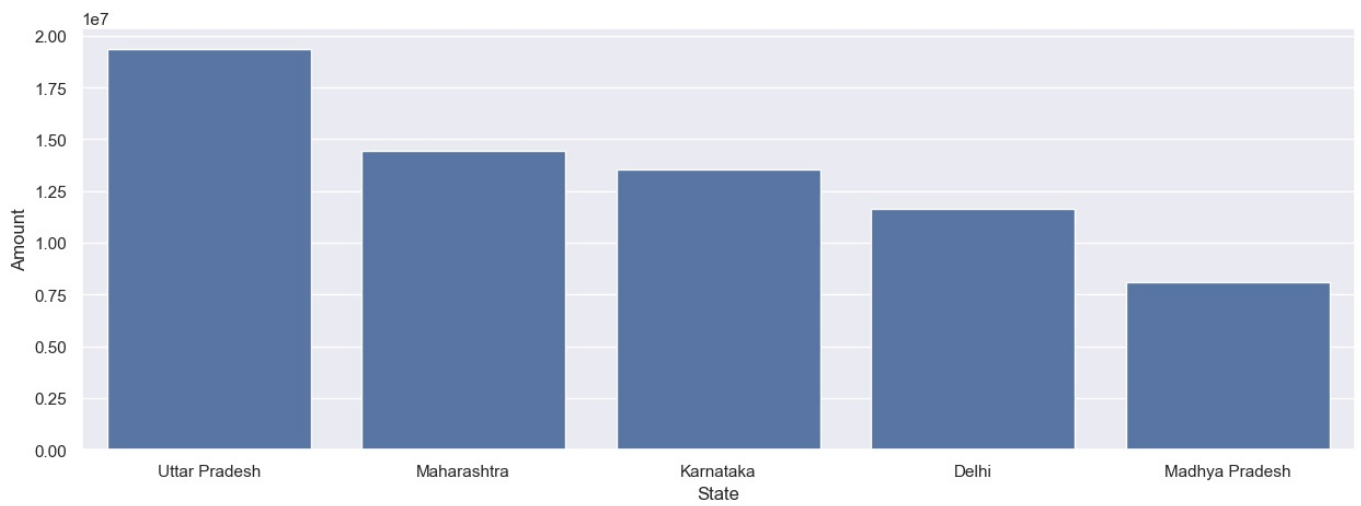


7. Top 5 States as Per Total Sales Amount

```
In [115.. # Amount Wise Top 5 State
sales_state = df.groupby(['State'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False).head(5)

sns.set(rc={'figure.figsize':(15,5)})
ax = sns.barplot(data=sales_state, x='State', y='Amount')

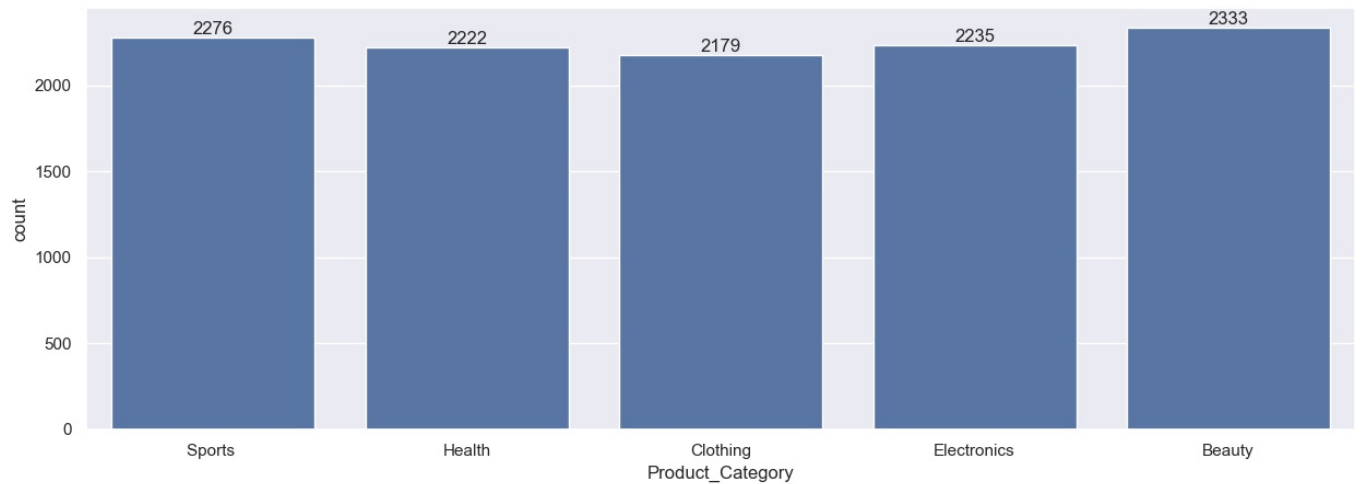
plt.show()
```



8. Product Category Wise Transactions Count

```
In [116]: # Product Category Wise Transaction Count
ax = sns.countplot(data=df, x='Product_Category')

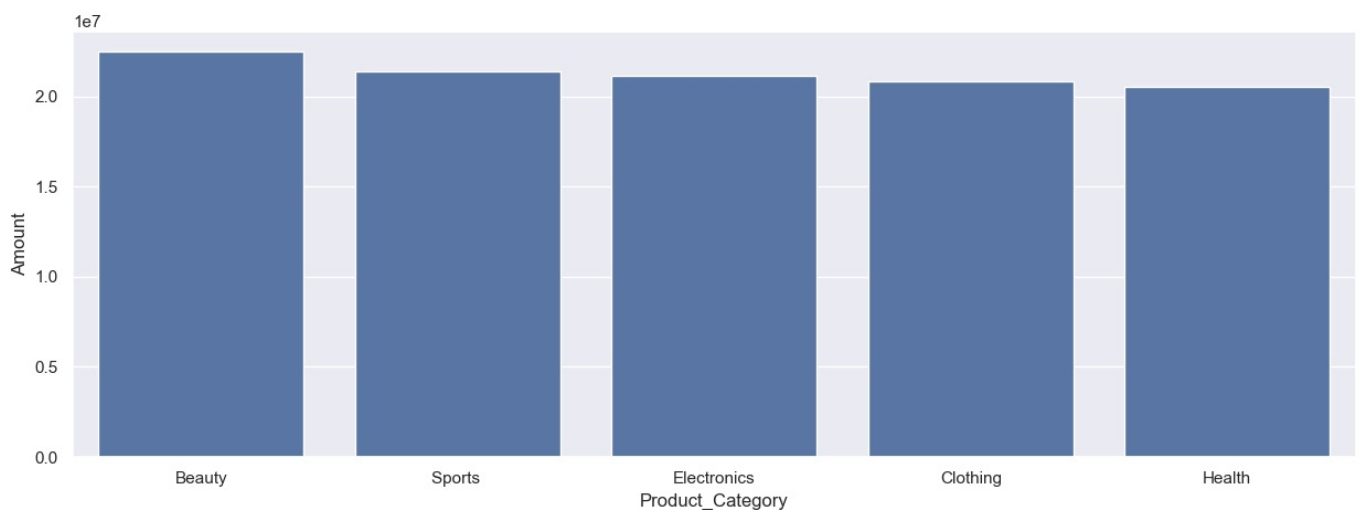
sns.set(rc={'figure.figsize': (15,5)})
for bars in ax.containers:
    ax.bar_label(bars)
plt.show()
```



9. Amount Wise Product Category

```
In [117]: # Amount Wise Product Category
sales_state = df.groupby(['Product_Category'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=True)

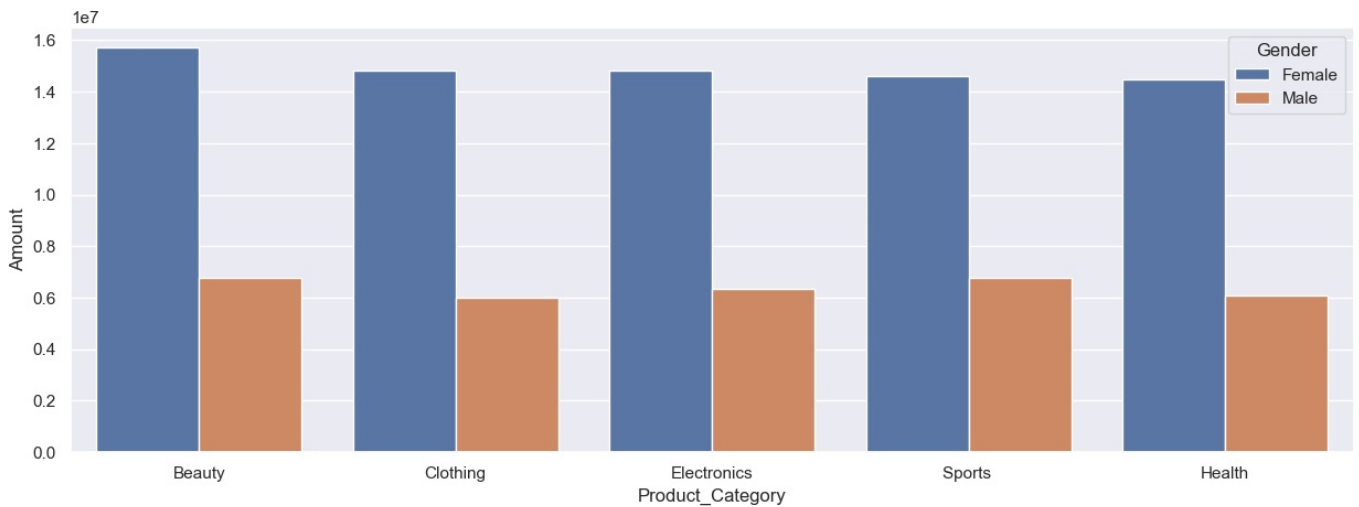
sns.set(rc={'figure.figsize': (15,5)})
ax = sns.barplot(data=sales_state, x='Product_Category', y='Amount')
plt.show()
```



10. Product Category and Gender Wise Transaction Count

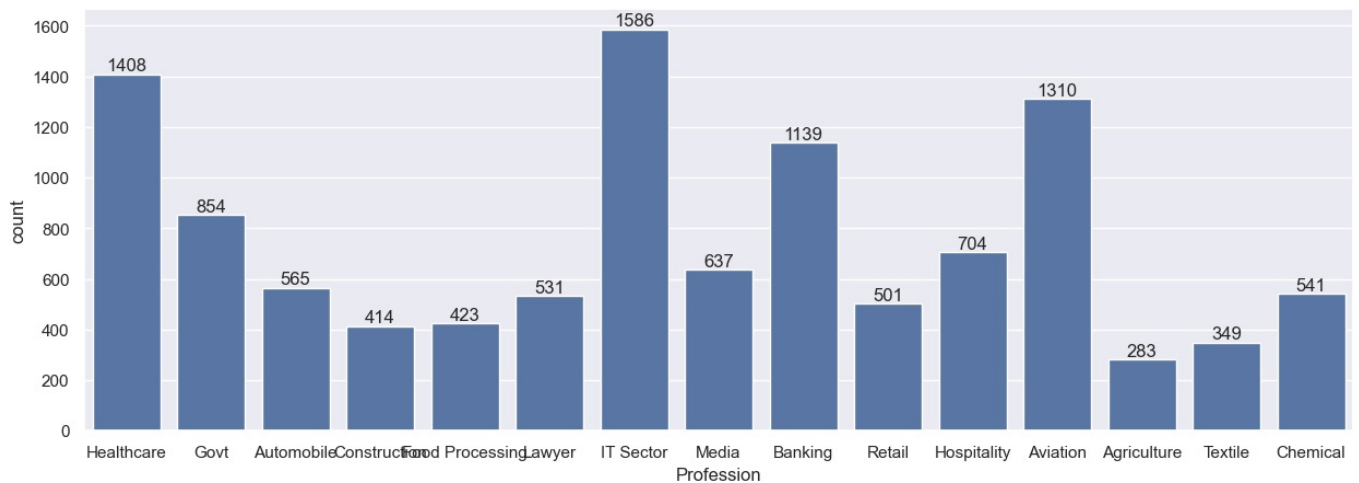
```
In [118.. # Product Category and Gender Wise Transaction Count
sales_product = df.groupby(['Product_Category', 'Gender'], as_index=False)['Amount'].sum().sort_values(by='Amount')

ax = sns.barplot(data=sales_product, x='Product_Category', y='Amount', hue='Gender')
plt.show()
```



11. Profession Wise Total Transactions

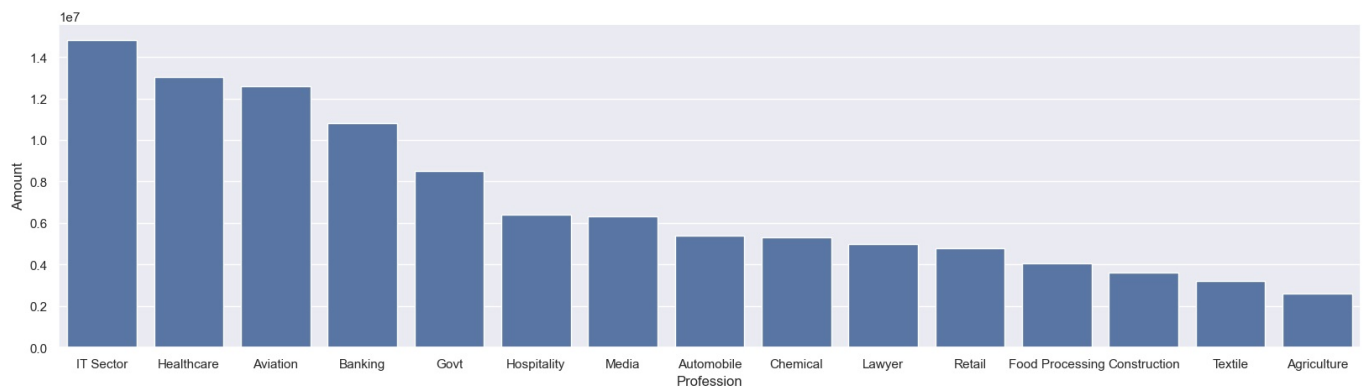
```
In [119.. # Profession Wise Transaction Count
ax = sns.countplot(data=df, x='Profession')
for bars in ax.containers:
    ax.bar_label(bars)
plt.show()
```



12. Amount Wise Top Professions

```
In [120.. # Amount Wise Product Category
sales_state = df.groupby(['Profession'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)

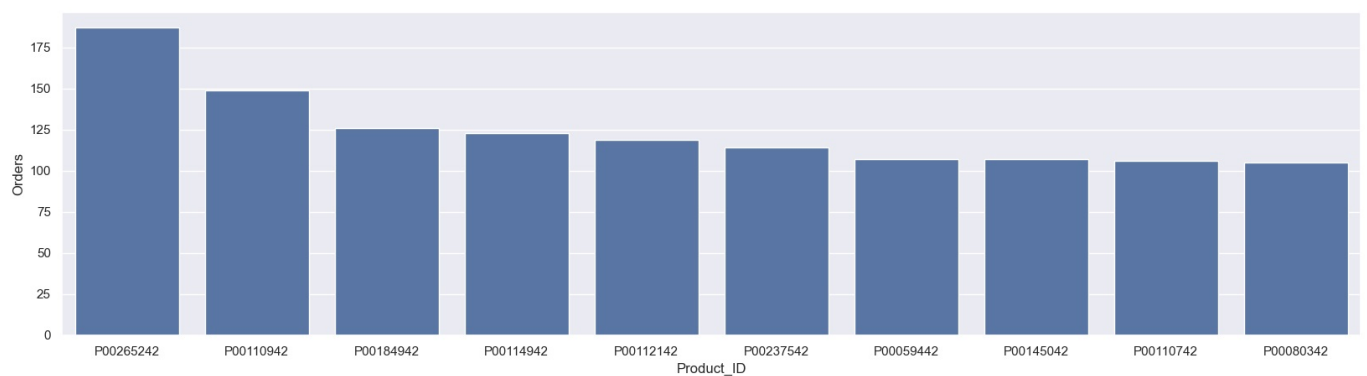
sns.set(rc={'figure.figsize':(20,5)})
ax = sns.barplot(data=sales_state, x='Profession', y='Amount')
plt.show()
```



13. Top 10 Product by Total Quantity Ordered

```
In [121]: # Order Wise Top 10 Product
sales_state = df.groupby(['Product_ID'], as_index=False)['Orders'].sum().sort_values(by='Orders', ascending=False)

sns.set(rc={'figure.figsize':(20,5)})
ax = sns.barplot(data=sales_state, x='Product_ID', y='Orders')
plt.show()
```



Conclusion

Female age group 26-35 yrs from UP, Maharashtra and Karnataka working in IT, Healthcare and Aviation are more likely to buy products from Beauty, Sports and Electronics.