**DATA ANLYSIS ON DIWALI SALES**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt # visualizing data

%matplotlib inline

import seaborn as sns

df = pd.read\_csv('Diwali Sales Data.csv', encoding= 'unicode\_escape')

**1. Question:** How many rows and columns are present in the dataset?

df.shape

(11251, 15)

**2. Question:** What are the first 5 rows of the dataset?

df.head()

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| User\_ID | Cust\_name | Product\_ID | Gender | Age Group | Age | Marital\_Status | State | Zone | Occupation | Product\_Category | Orders | Amount | Status | unnamed1 |
| 0 | 1002903 | Sanskriti | P00125942 | F | 26-35 | 28 | 0 | Maharashtra | Western | Healthcare | Auto | 1 | 23952.0 | NaN | NaN |
| 1 | 1000732 | Kartik | P00110942 | F | 26-35 | 35 | 1 | Andhra Pradesh | Southern | Govt | Auto | 3 | 23934.0 | NaN | NaN |
| 2 | 1001990 | Bindu | P00118542 | F | 26-35 | 35 | 1 | Uttar Pradesh | Central | Automobile | Auto | 3 | 23924.0 | NaN | NaN |
| 3 | 1001425 | Sudevi | P00237842 | M | 0-17 | 16 | 0 | Karnataka | Southern | Construction | Auto | 2 | 23912.0 | NaN | NaN |
| 4 | 1000588 | Joni | P00057942 | M | 26-35 | 28 | 1 | Gujarat | Western | Food Processing | Auto | 2 | 23877.0 | NaN | NaN |

**3. Question:** What are the data types of each column?

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 |

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

**4. Question:** How many null values exist in the dataset?

pd.isnull(df).sum()

|  |
| --- |
| 0 |
| 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 |

**Question:** How to drop rows with null values in the 'Amount' column?

df.dropna(inplace=True)

**7. Question:** How to convert the 'Amount' column to integer type?

df['Amount'] = df['Amount'].astype('int')

df['Amount'].dtypes

dtype('int64')

**8. Question:** What are the column names in the dataset?

df.columns

Index(['User\_ID', 'Cust\_name', 'Product\_ID', 'Gender', 'Age Group', 'Age',

'Marital\_Status', 'State', 'Zone', 'Occupation', 'Product\_Category',

'Orders', 'Amount'],

dtype='object')

**9. Question:** How to rename the 'Marital\_Status' column to 'Shaadi'?

df.rename(columns= {'Marital\_Status':'Shaadi'})

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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

**10. Question:** What is the statistical summary of numeric columns?

df.describe()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |

df.describe()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |

df.columns

Index(['User\_ID', 'Cust\_name', 'Product\_ID', 'Gender', 'Age Group', 'Age',

'Marital\_Status', 'State', 'Zone', 'Occupation', 'Product\_Category',

'Orders', 'Amount'],

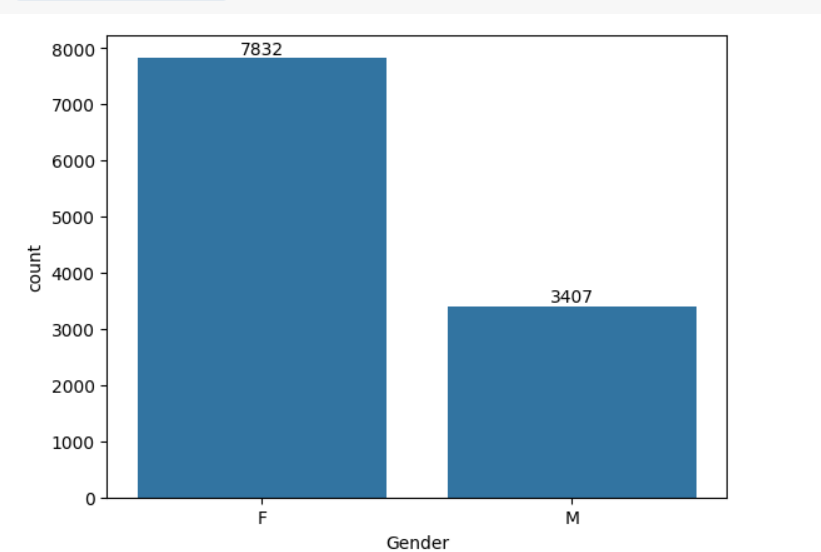
dtype='object')

**11. Question:** What is the gender-wise distribution of customers?

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

for bars in ax.containers:

    ax.bar\_label(bars)



for bars in ax.containers:

    ax.bar\_label(bars)

**12. Question:** What is the total purchase amount made by each gender?

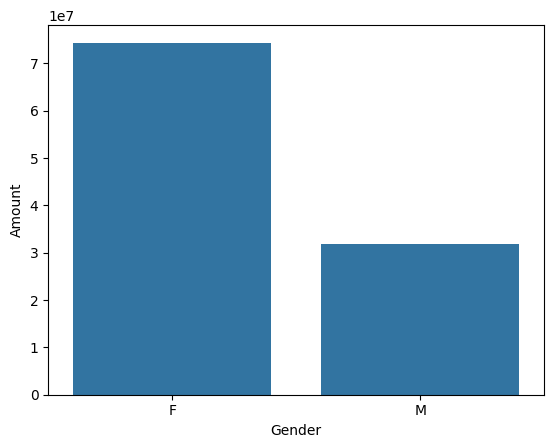
df.groupby(['Gender'],as\_index=False)['Amount'].sum().sort\_values(by='Amount', ascending=False)

|  |  |  |
| --- | --- | --- |
|  | Gender | Amount |
| 0 | F | 74335853 |
| 1 | M | 31913276 |

sales\_age = df.groupby(['Age Group'],as\_index=False)['Amount'].sum().sort\_values(by='Amount', ascending=False)

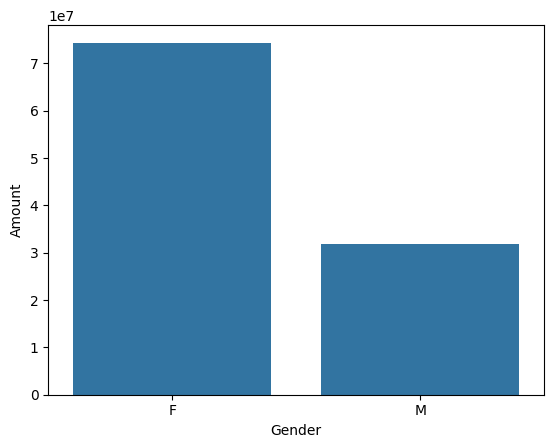
sns.barplot(x = 'Gender',y= 'Amount' ,data = sales\_gen)

<Axes: xlabel='Gender', ylabel='Amount'>



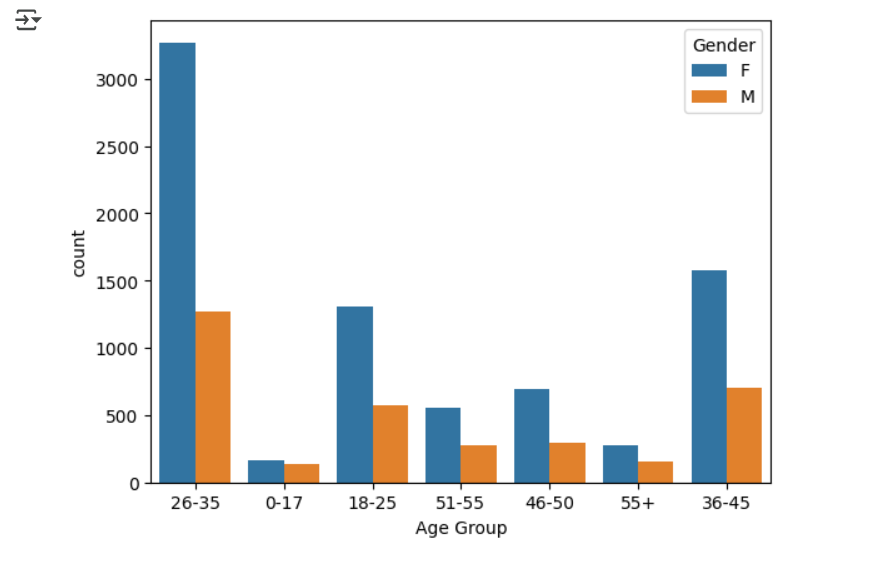
sns.barplot(x = 'Gender',y= 'Amount' ,data = sales\_gen)

<Axes: xlabel='Gender', ylabel='Amount'>



**13. Question:** What is the purchase distribution across age groups and gender?

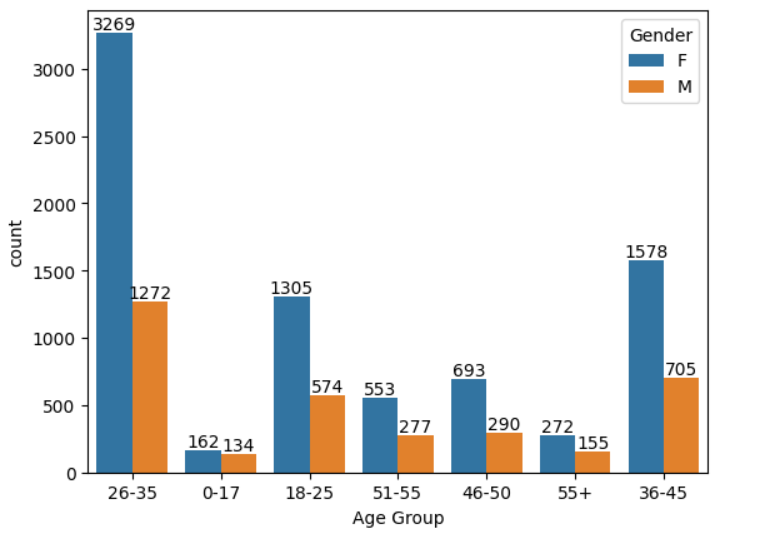
ax = sns.countplot(data = df, x = 'Age Group', hue = 'Gender')



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

for bars in ax.containers:

    ax.bar\_label(bars)

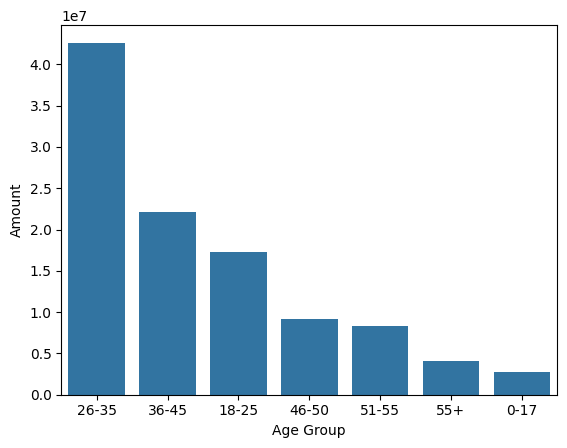


**14. Question:** What is the total purchase amount across different age groups?

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)

<Axes: xlabel='Age Group', ylabel='Amount'>



🗺️ **STATE-WISE & MARITAL STATUS ANALYSIS**

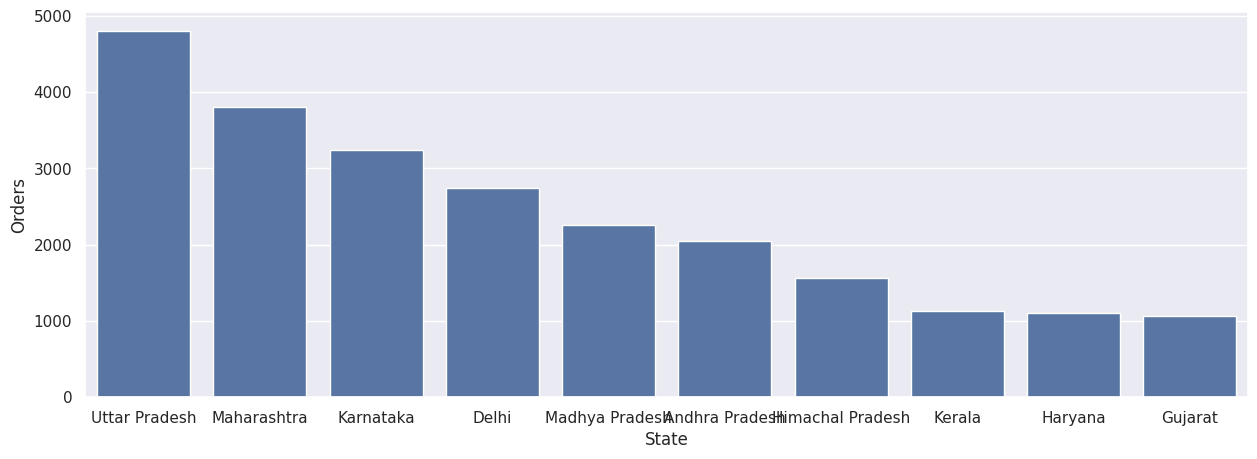
**15. Question:** Which states had the most orders?

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)})

sns.barplot(data = sales\_state, x = 'State',y= 'Orders')

<Axes: xlabel='State', ylabel='Orders'>



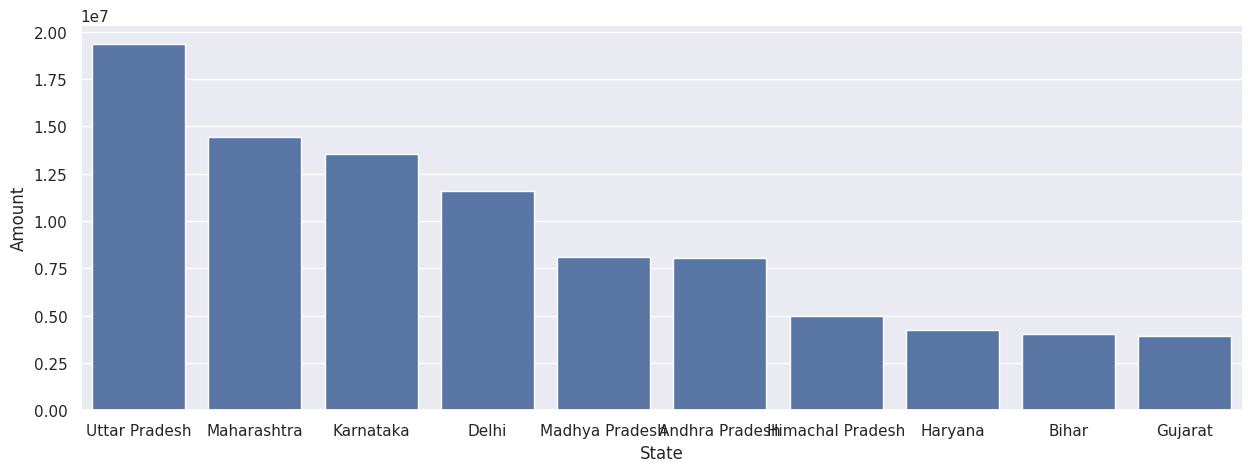
**16. Question:** Which states generated the highest revenue?

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)})

sns.barplot(data = sales\_state, x = 'State',y= 'Amount')

<Axes: xlabel='State', ylabel='Amount'>



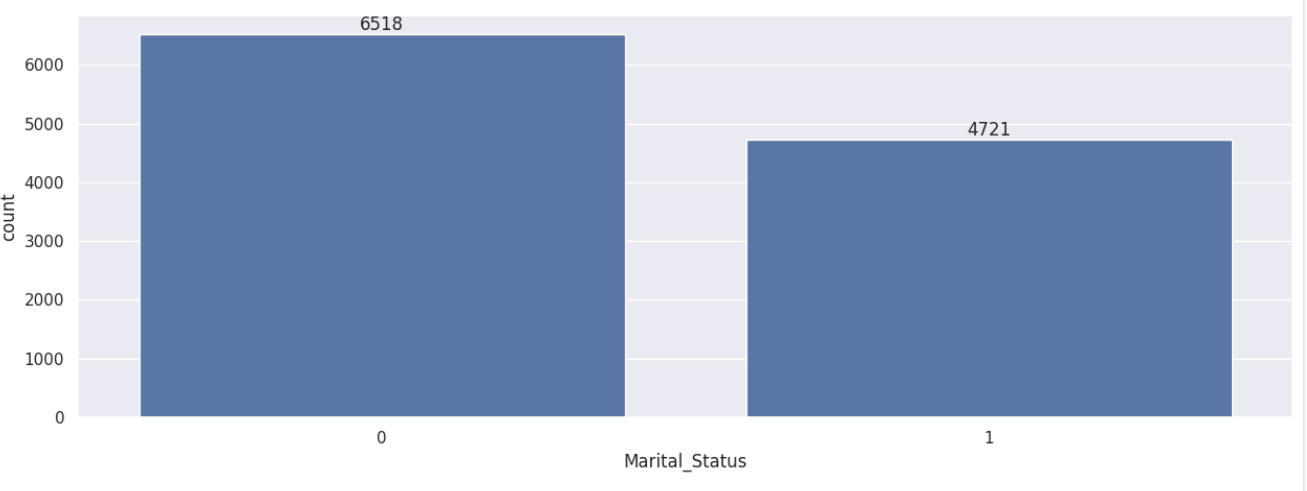
**17. Question:** What is the distribution of customers based on marital status?

ax = sns.countplot(data = df, x = 'Marital\_Status')

sns.set(rc={'figure.figsize':(7,5)})

for bars in ax.containers:

    ax.bar\_label(bars)



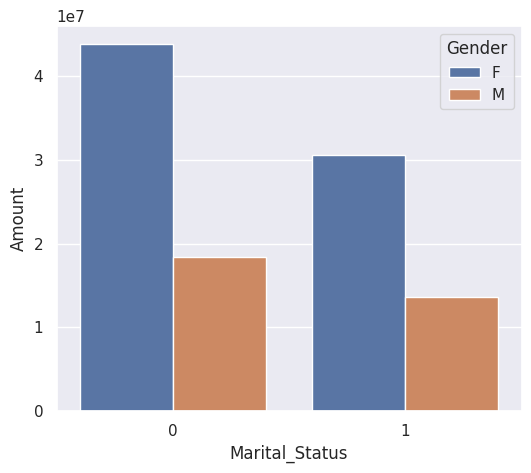
**18. Question:** What is the purchase amount distribution across marital status and gender?

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')

<Axes: xlabel='Marital\_Status', ylabel='Amount'>



👨‍💼 **OCCUPATION & PRODUCT ANALYSIS**

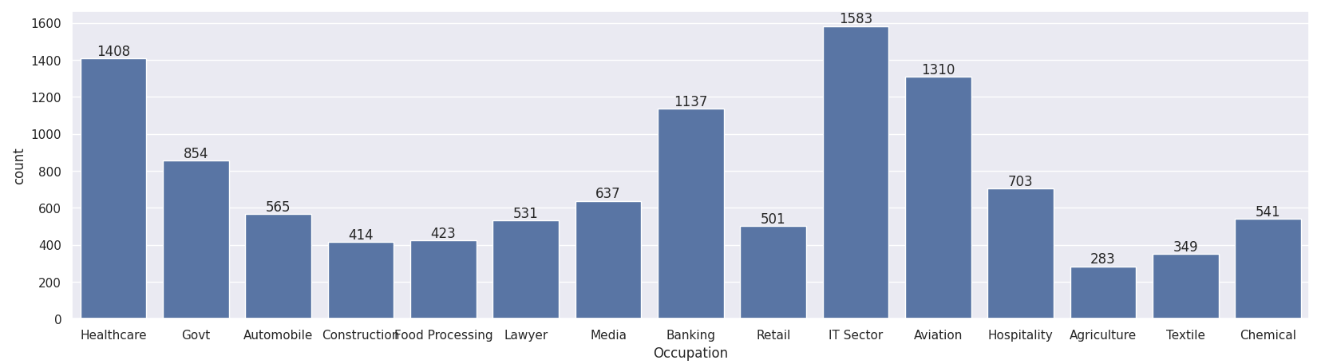
**19. Question:** How are customers distributed across different occupations?

sns.set(rc={'figure.figsize':(20,5)})

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

for bars in ax.containers:

    ax.bar\_label(bars)



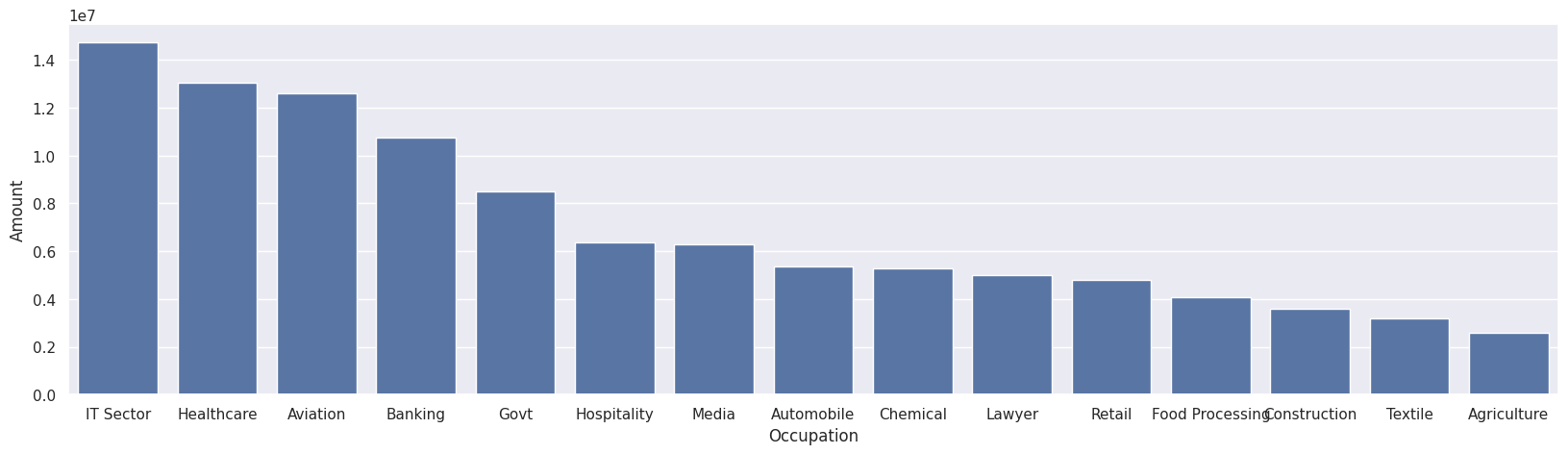
**20. Question:** Which occupations contributed the most to the revenue?

sales\_state = df.groupby(['Occupation'], as\_index=False)['Amount'].sum().sort\_values(by='Amount', ascending=False)

sns.set(rc={'figure.figsize':(20,5)})

sns.barplot(data = sales\_state, x = 'Occupation',y= 'Amount')

<Axes: xlabel='Occupation', ylabel='Amount'>



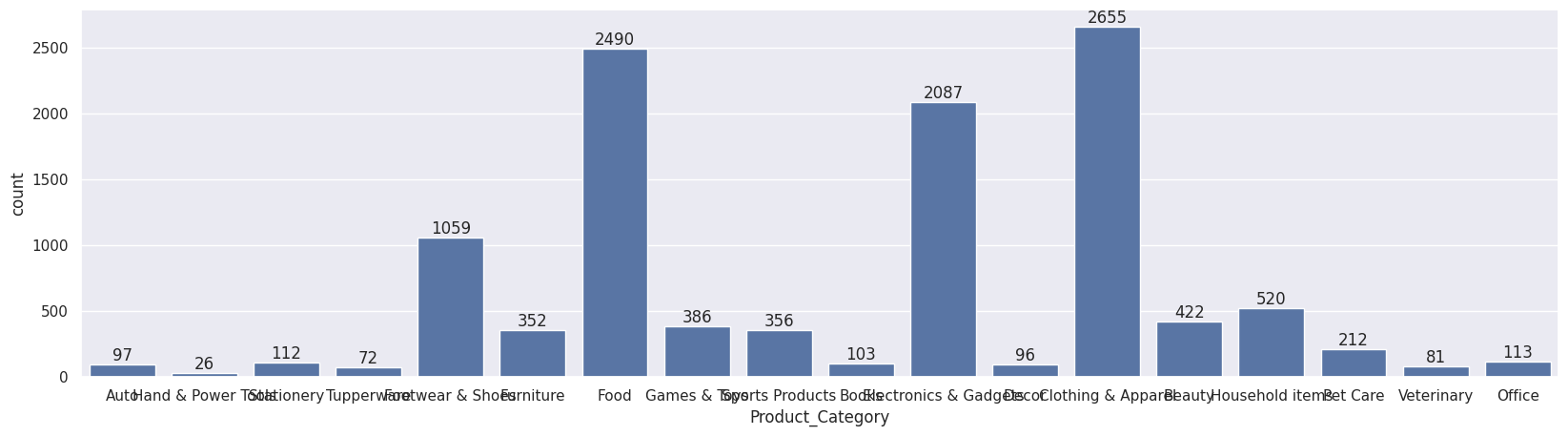
**21. Question:** Which product categories are most popular?

sns.set(rc={'figure.figsize':(20,5)})

ax = sns.countplot(data = df, x = 'Product\_Category')

for bars in ax.containers:

    ax.bar\_label(bars)



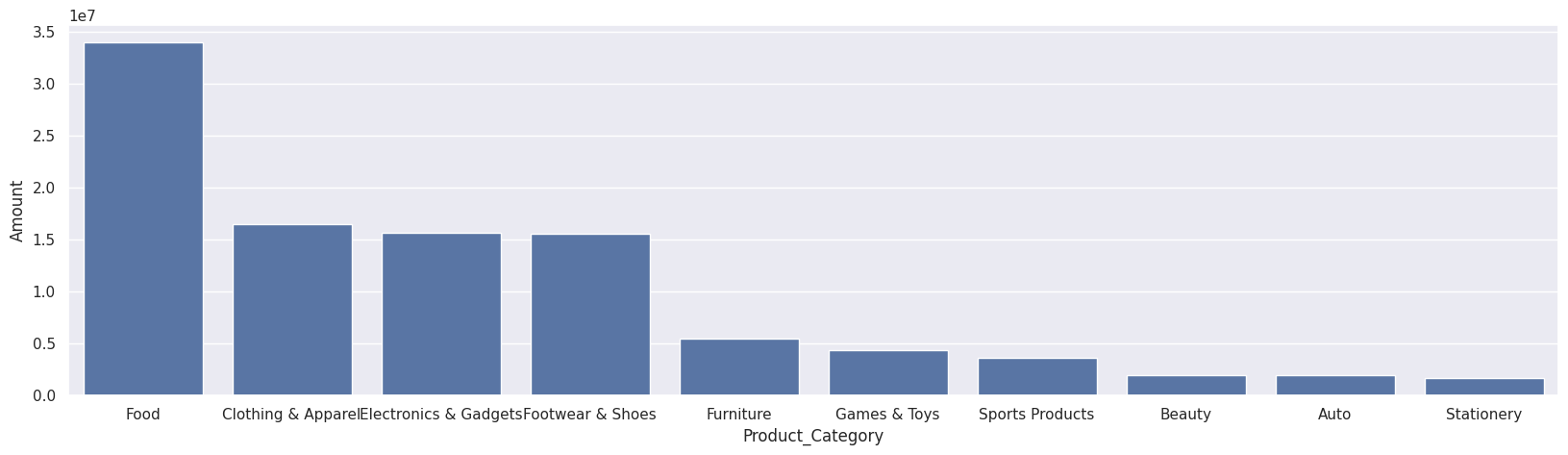
**22. Question:** Which product categories generated the highest revenue?

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')

<Axes: xlabel='Product\_Category', ylabel='Amount'>



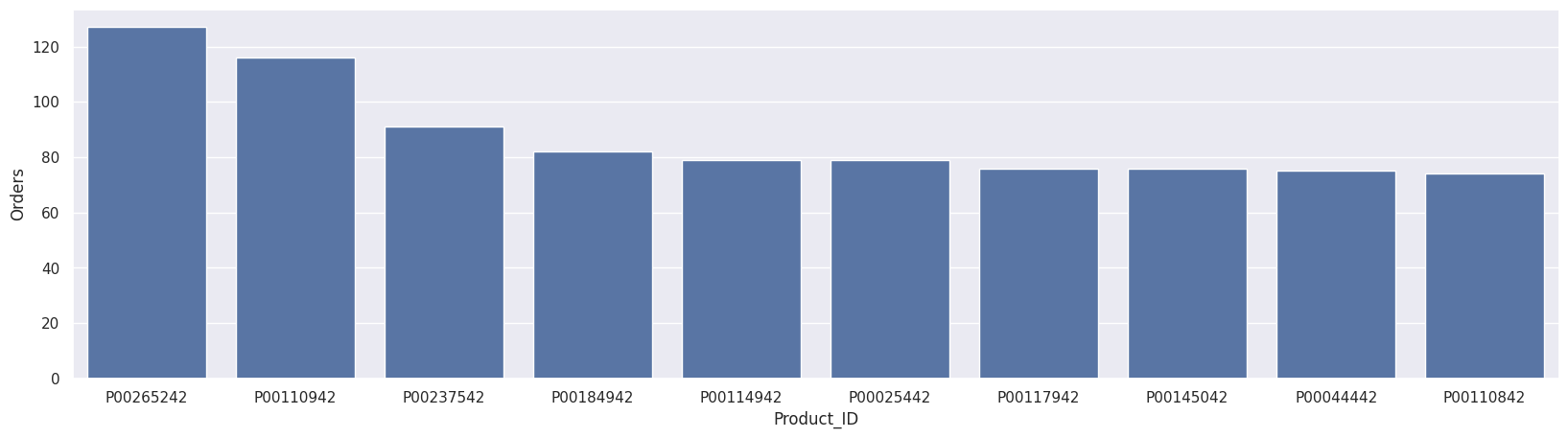
**23. Question:** Which specific products had the highest number of orders?

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)})

sns.barplot(data = sales\_state, x = 'Product\_ID',y= 'Orders')

<Axes: xlabel='Product\_ID', ylabel='Orders'>



**24. Question:** Visualize the top 10 products with highest orders using bar chart.

fig1, ax1 = plt.subplots(figsize=(12,7))

df.groupby('Product\_ID')['Orders'].sum().nlargest(10).sort\_values(ascending=False).plot(kind='bar')

<Axes: xlabel='Product\_ID'>

