Installing packages, Loading and inspecting dataset to have a sneak peek

In [1]:

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

In [3]:

```
df = pd.read_csv(r'D:\Diwali_Sales_dataset.csv',encoding = "unicode_escape")
df
```

Out[3]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occup
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healt
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Autor
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Constr
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Proce
...	...	...	...	...	...	...	...	...	...	
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Che
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healt
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	-
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agric
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healt

11251 rows × 15 columns

In [4]:

```
df.shape
```

Out[4]:

```
(11251, 15)
```

In [5]:

```
df.head()
```

Out[5]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing

Data Cleaning and Preparing for analysis

In [6]:

```
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
```

In [12]:

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

In [13]:

```
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
```

In [16]:

```
pd.isnull(df).sum()
```

Out[16]:

```
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
```

In [ ]:

```
#changing
```

In [57]:

```
#Checking data type after the conversion
```

In [28]:

```
df["Amount"].dtypes
```

Out[28]:

```
dtype('float64')
```

In [29]:

```
df.columns
```

Out[29]:

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

In [30]:

```
df.rename(columns = {"Cust_name": "Customer_name", "Marital_Status": "Relationship_Statu
```

In [32]:

```
df.columns
```

Out[32]:

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

In [33]:

```
df[["Orders", "Amount"]].describe()
```

Out[33]:

	Orders	Amount
count	11251.000000	11239.000000
mean	2.489290	9453.610858
std	1.115047	5222.355869
min	1.000000	188.000000
25%	1.500000	5443.000000
50%	2.000000	8109.000000
75%	3.000000	12675.000000
max	4.000000	23952.000000

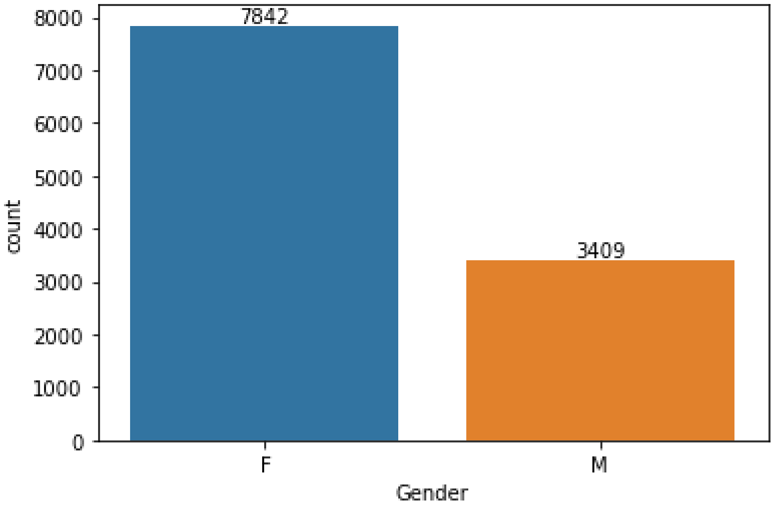
Exploratory Data Analysis

In [34]:

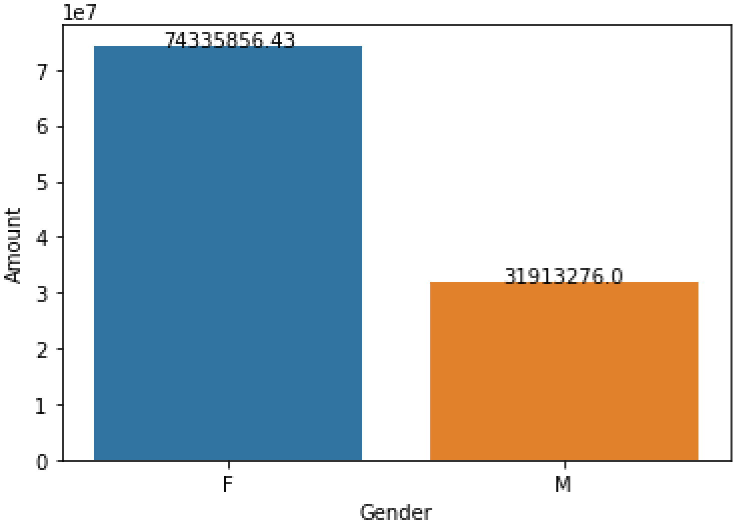
```
data = sns.countplot(x = "Gender", data = df)
```

Loading [MathJax]/extensions/Safe.js

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



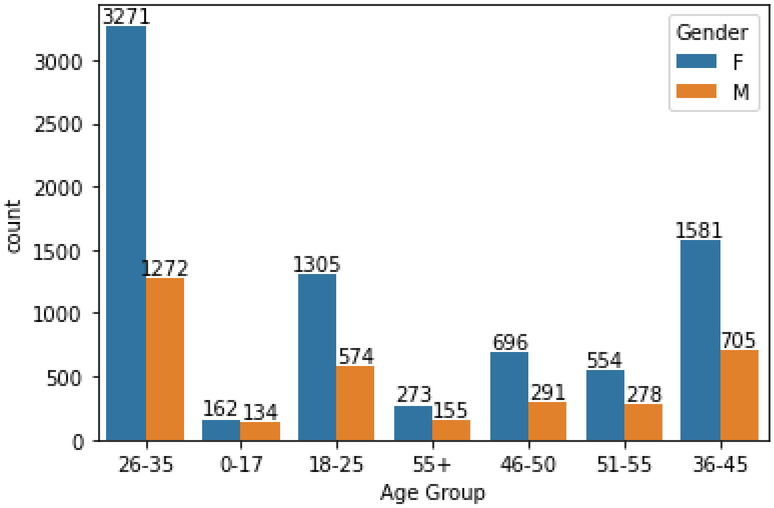
In [35]:
```
sales_gen = df.groupby(["Gender"], as_index = False) ["Amount"].sum().sort_values(by = "

data = sns.barplot(x = "Gender", y = "Amount", data = sales_gen)

for index, row in sales_gen.iterrows():
    data.text(index, row["Amount"], row["Amount"], ha="center")
```



Based on Age

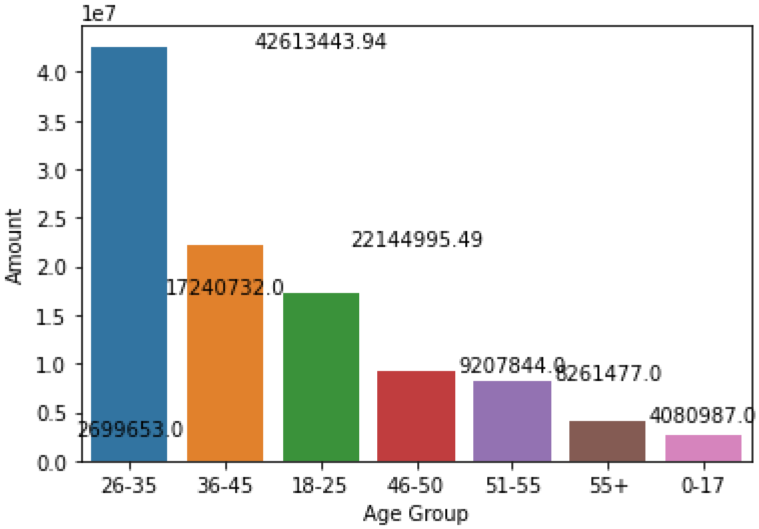
In [36]:
```
data = sns.countplot(x = "Age Group", data = df, hue = "Gender")

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

Loading [MathJax]/extensions/Safe.js



In [37]:

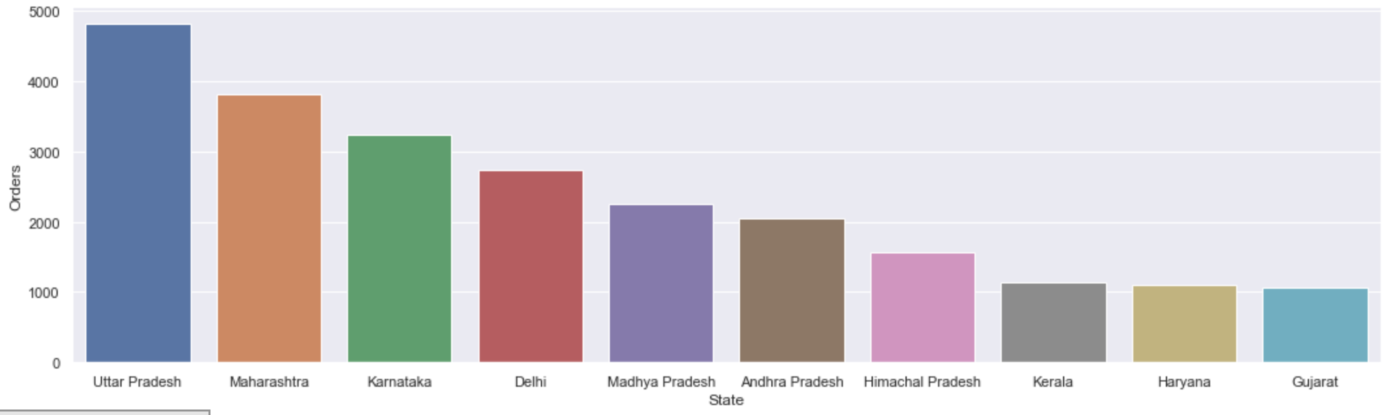
```
sales_age = df.groupby(["Age Group"], as_index = False) ["Amount"].sum().sort_values(by

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

for index, row in sales_age.iterrows():
    data.text(index, row["Amount"], row["Amount"], ha="center")
```



Based on States

In [38]:

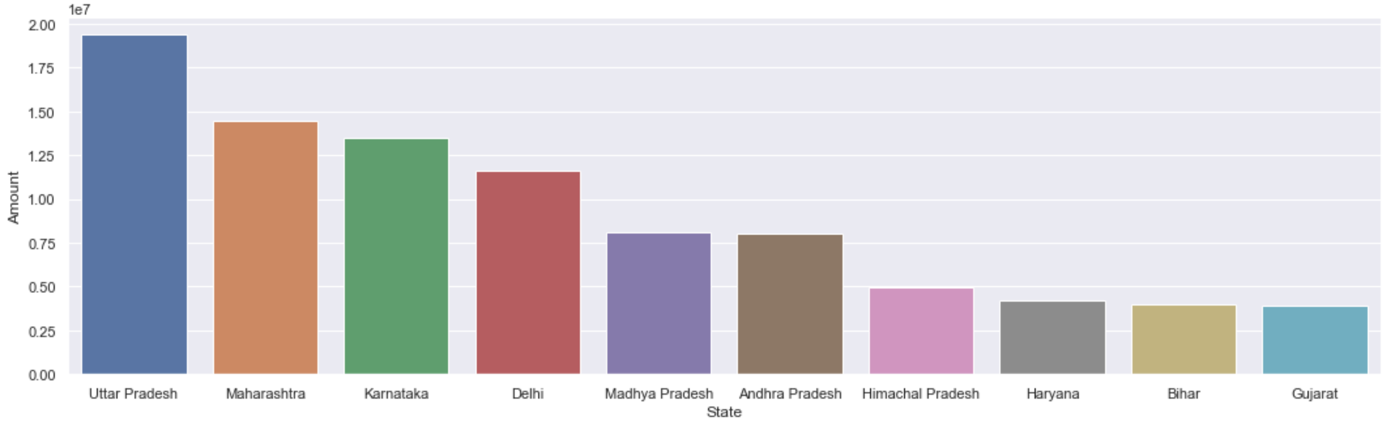
```
sales_state = df.groupby(["State"], as_index = False) ["Orders"].sum().sort_values(by =

sns.set(rc={"figure.figsize":(18,5)})
data = sns.barplot(x = "State", y = "Orders", data = sales_state)
```



Loading [MathJax]/extensions/Safe.js

In [39]:

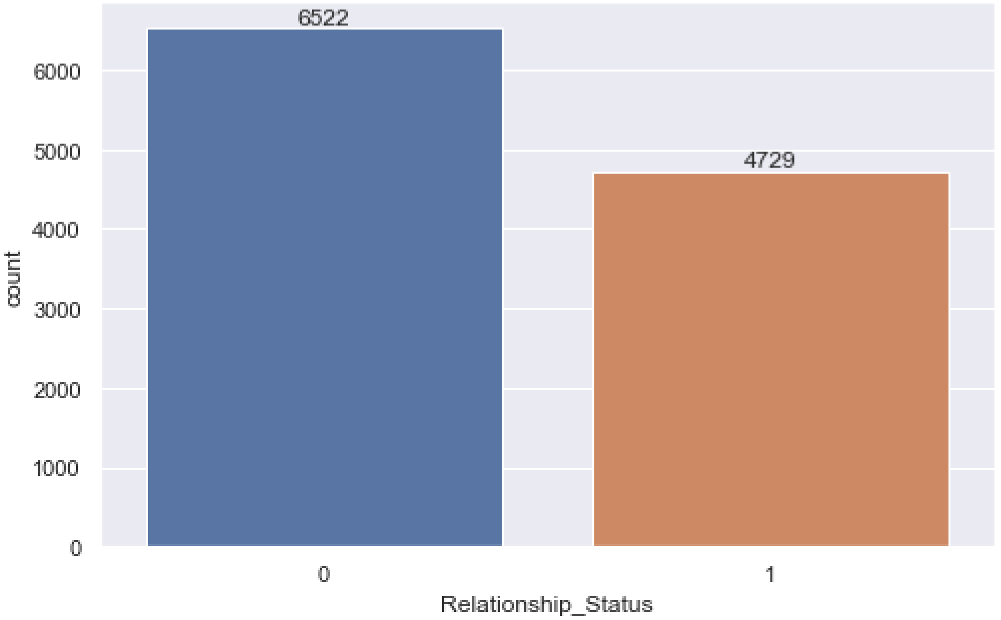
```
sales_state = df.groupby(["State"], as_index = False) ["Amount"].sum().sort_values(by =

sns.set(rc={"figure.figsize":(18,5)})
data = sns.barplot(x = "State", y = "Amount", data = sales_state)
```



Based on Relationship Statsu

In [42]:

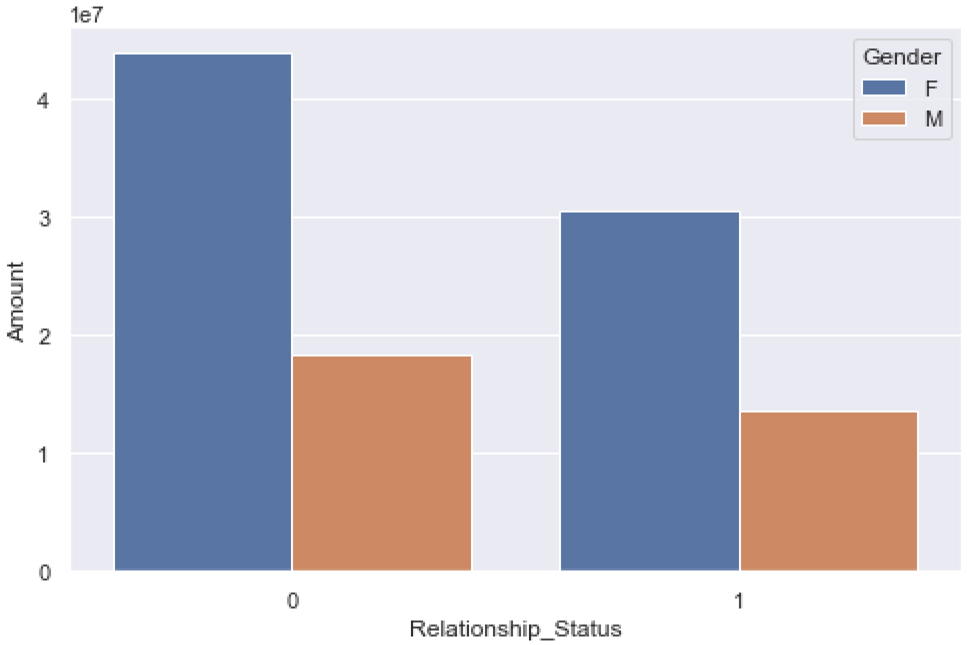
```
data = sns.countplot(x = "Relationship_Status", data = df)

sns.set(rc={"figure.figsize":(8,5)})
for bars in data.containers:
    data.bar_label(bars)
```



In [43]:

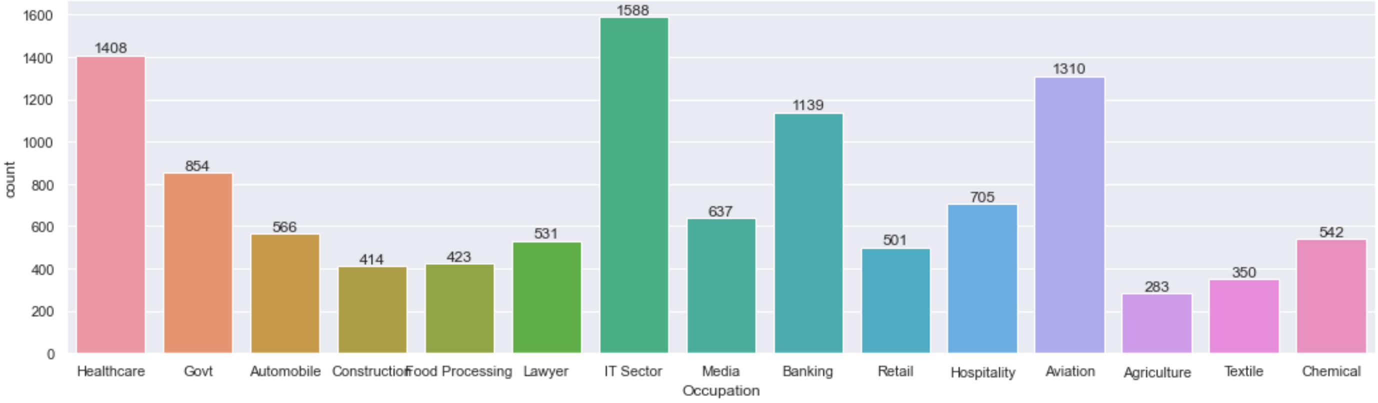
```
sales_state = df.groupby(["Relationship_Status", "Gender"], as_index = False) ["Amount"]

sns.set(rc={"figure.figsize":(8,5)})
data = sns.barplot(x = "Relationship_Status", y = "Amount", data = sales_state, hue = "G
```

Loading [MathJax]/extensions/Safe.js

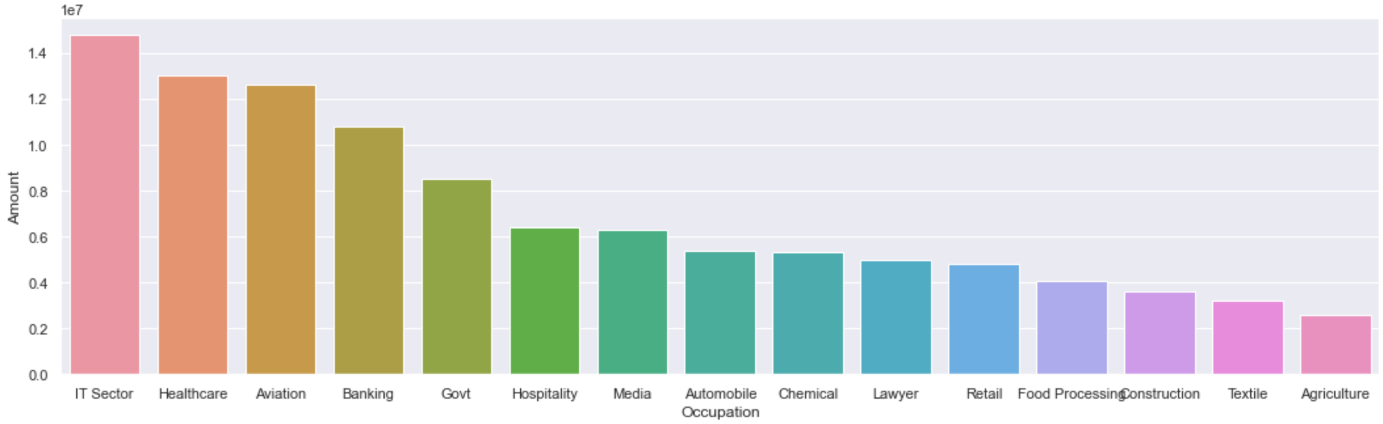


Based on Occupation

In [46]:
```
data = sns.countplot(x = "Occupation", data = df)

sns.set(rc={"figure.figsize":(18,5)})
for bars in data.containers:
    data.bar_label(bars)
```



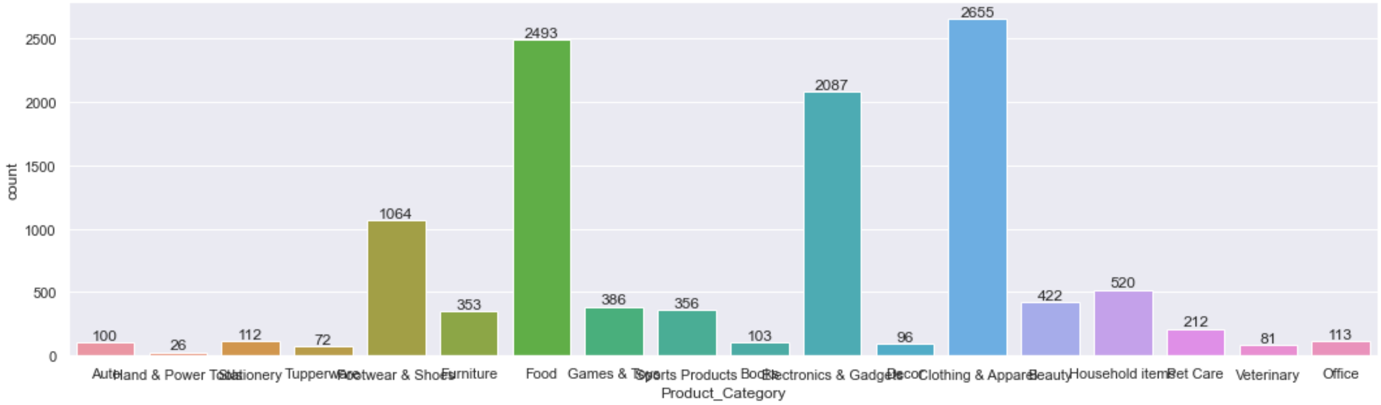
In [47]:
```
sales_state = df.groupby(["Occupation"], as_index = False) ["Amount"].sum().sort_values(

sns.set(rc={"figure.figsize":(18,5)})
data = sns.barplot(x = "Occupation", y = "Amount", data = sales_state)
```



Based on Product Category

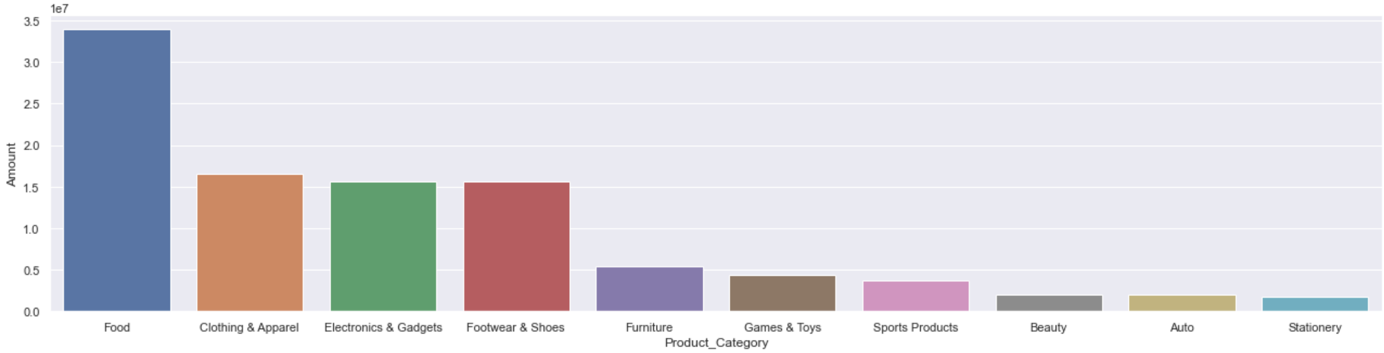
In [48]:

```
data = sns.countplot(x = "Product_Category", data = df)

sns.set(rc={"figure.figsize":(20,5)})
for bars in data.containers:
    data.bar_label(bars)
```



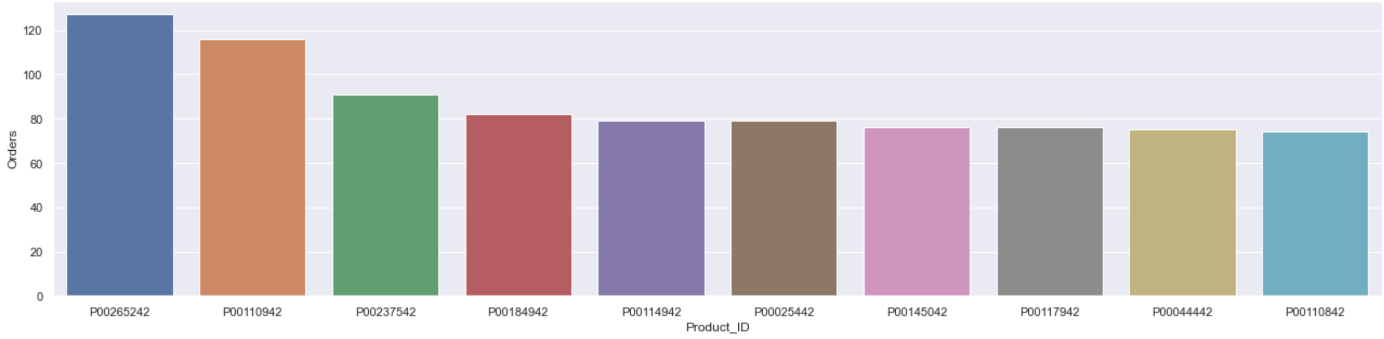
In [50]:

```
sales_state = df.groupby(["Product_Category"], as_index = False) ["Amount"].sum().sort_v

sns.set(rc={"figure.figsize":(22,5)})
data = sns.barplot(x = "Product_Category", y = "Amount", data = sales_state)
```



In [53]:

```
sales_state = df.groupby(["Product_ID"], as_index = False) ["Orders"].sum().sort_values(

sns.set(rc={"figure.figsize":(22,5)})
data = sns.barplot(x = "Product_ID", y = "Orders", data = sales_state)
```



In [ ]:

In [ ]:

Loading [MathJax]/extensions/Safe.js