Diwali Sales Analysis

Objective: We have diwali sales data and need to identify Sales and Orders data as per Gender, State, Age group, Product Category and Occupation.

In [1]:
```
pip install pandas
```

```
Requirement already satisfied: pandas in c:\users\hites\appdata\local\programs\python\python311\lib\site-package
s (2.0.1)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\hites\appdata\local\programs\python\python311\
lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\hites\appdata\local\programs\python\python311\lib\site-p
ackages (from pandas) (2023.3)
Requirement already satisfied: tzdata>=2022.1 in c:\users\hites\appdata\local\programs\python\python311\lib\site
-packages (from pandas) (2023.3)
Requirement already satisfied: numpy>=1.21.0 in c:\users\hites\appdata\local\programs\python\python311\lib\site-
packages (from pandas) (1.24.3)
Requirement already satisfied: six>=1.5 in c:\users\hites\appdata\local\programs\python\python311\lib\site-packa
ges (from python-dateutil>=2.8.2->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
pip install numpy
```

```
Requirement already satisfied: numpy in c:\users\hites\appdata\local\programs\python\python311\lib\site-packages
(1.24.3)
Note: you may need to restart the kernel to use updated packages.
```

In [3]:
```
pip install matplotlib
```

```
Requirement already satisfied: matplotlib in c:\users\hites\appdata\local\programs\python\python311\lib\site-pac
kages (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\hites\appdata\local\programs\python\python311\lib\si
te-packages (from matplotlib) (1.0.7)
Requirement already satisfied: cycler>=0.10 in c:\users\hites\appdata\local\programs\python\python311\lib\site-p
ackages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\hites\appdata\local\programs\python\python311\lib\s
ite-packages (from matplotlib) (4.39.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\hites\appdata\local\programs\python\python311\lib\s
ite-packages (from matplotlib) (1.4.4)
Requirement already satisfied: numpy>=1.20 in c:\users\hites\appdata\local\programs\python\python311\lib\site-pa
ckages (from matplotlib) (1.24.3)
Requirement already satisfied: packaging>=20.0 in c:\users\hites\appdata\local\programs\python\python311\lib\sit
e-packages (from matplotlib) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\hites\appdata\local\programs\python\python311\lib\site-
packages (from matplotlib) (9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\hites\appdata\local\programs\python\python311\lib\si
te-packages (from matplotlib) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\hites\appdata\local\programs\python\python311\li
b\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\hites\appdata\local\programs\python\python311\lib\site-packa
ges (from python-dateutil>=2.7->matplotlib) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [4]:
```
pip install seaborn
```

```
Requirement already satisfied: seaborn in c:\users\hites\appdata\local\programs\python\python311\lib\site-packag
es (0.12.2)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in c:\users\hites\appdata\local\programs\python\python311\li
b\site-packages (from seaborn) (1.24.3)
Requirement already satisfied: pandas>=0.25 in c:\users\hites\appdata\local\programs\python\python311\lib\site-p
ackages (from seaborn) (2.0.1)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in c:\users\hites\appdata\local\programs\python\python311
\lib\site-packages (from seaborn) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\hites\appdata\local\programs\python\python311\lib\si
te-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.0.7)
Requirement already satisfied: cycler>=0.10 in c:\users\hites\appdata\local\programs\python\python311\lib\site-p
ackages (from matplotlib!=3.6.1,>=3.1->seaborn) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\hites\appdata\local\programs\python\python311\lib\s
ite-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (4.39.4)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\hites\appdata\local\programs\python\python311\lib\s
ite-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\hites\appdata\local\programs\python\python311\lib\sit
e-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (23.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\hites\appdata\local\programs\python\python311\lib\site-
packages (from matplotlib!=3.6.1,>=3.1->seaborn) (9.5.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\hites\appdata\local\programs\python\python311\lib\si
te-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\hites\appdata\local\programs\python\python311\li
b\site-packages (from matplotlib!=3.6.1,>=3.1->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\hites\appdata\local\programs\python\python311\lib\site-p
ackages (from pandas>=0.25->seaborn) (2023.3)
Requirement already satisfied: tzdata>=2022.1 in c:\users\hites\appdata\local\programs\python\python311\lib\site
-packages (from pandas>=0.25->seaborn) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\hites\appdata\local\programs\python\python311\lib\site-packa
ges (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.1->seaborn) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [5]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [6]:
```
df = pd.read_csv(r'C:\Users\hites\Downloads\Diwali Sales Data (1).csv',encoding = 'unicode_escape')
```

In [7]:
```
df.head()
```

Out[7]:

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

In [8]:
```
df.tail()
```

Out[8]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Catego
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Offi
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterina
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Offi
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Offi
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Offi

In [9]:
```
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 [10]:
```
df.shape
```

Out[10]:
```
(11251, 15)
```

In [11]:
```
pd.isnull(df).sum()
```

Out[11]:
```
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
Status             11251
unnamed1           11251
dtype: int64
```

In [12]:
```
null_amount = df['Amount'].isnull()
print(null_amount)
```

```
0        False
1        False
2        False
3        False
4        False
         ...
11246    False
11247    False
11248    False
11249    False
11250    False
Name: Amount, Length: 11251, dtype: bool
```

In [13]:
```
x = df[null_amount]
print(x)
```

```
     User_ID Cust_name Product_ID Gender Age Group  Age  Marital_Status
7    1002092  Shivangi  P00273442      F       55+   61               0  \
14   1003858      Cano  P00293742      M     46-50   46               1
16   1005447       Amy  P00275642      F     46-50   48               1
109  1005265    Sakshi  P00296242      F     46-50   48               1
111  1005261   Apoorva  P00057942      F     36-45   41               1
184  1005538    Kartik  P00269542      F     46-50   49               1
293  1000326  Jonathan  P00120542      M     51-55   53               0
344  1002507   Lakshmi  P00045842      F     26-35   35               1
345  1004498   Srishti  P00030842      F     51-55   55               0
452  1004601    Gaurav  P00014442      F     36-45   40               1
464  1004528    Anurag  P00338442      F     26-35   33               1
493  1002994    Hemant   P0009942      F     36-45   38               0

              State      Zone   Occupation  Product_Category  Orders  Amount
7       Maharashtra   Western    IT Sector              Auto       1     NaN  \
14   Madhya Pradesh   Central  Hospitality              Auto       3     NaN
16   Andhra Pradesh  Southern    IT Sector              Auto       3     NaN
109           Delhi   Central      Banking  Footwear & Shoes       1     NaN
111           Delhi   Central    IT Sector  Footwear & Shoes       2     NaN
184       Karnataka  Southern      Banking  Footwear & Shoes       1     NaN
293         Gujarat   Western    IT Sector  Footwear & Shoes       3     NaN
344         Gujarat   Western     Chemical         Furniture       1     NaN
345           Delhi   Central      Textile  Footwear & Shoes       1     NaN
452  Madhya Pradesh   Central  Hospitality              Food       4     NaN
464   Uttar Pradesh   Central   Automobile              Food       2     NaN
493   Uttar Pradesh   Central    IT Sector              Food       4     NaN

     Status  unnamed1
7       NaN       NaN
14      NaN       NaN
16      NaN       NaN
109     NaN       NaN
111     NaN       NaN
184     NaN       NaN
293     NaN       NaN
344     NaN       NaN
345     NaN       NaN
452     NaN       NaN
464     NaN       NaN
493     NaN       NaN
```

Updating null values of Amount with the mean value of amount.

In [14]:
```
df['Amount']= df['Amount'].fillna(df['Amount'].mean())
```

In [15]:
```
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            11251 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 [16]:
```
df.loc[0:16,'Amount']
```

Out[16]:
```
0     23952.000000
1     23934.000000
2     23924.000000
3     23912.000000
4     23877.000000
5     23877.000000
6     23841.000000
7      9453.610858
8     23809.000000
9     23799.990000
10    23770.000000
11    23752.000000
12    23730.000000
13    23718.000000
14     9453.610858
15    23664.000000
16     9453.610858
Name: Amount, dtype: float64
```

In [18]:
```
df['Amount']=df['Amount'].astype('int')
```

In [19]:
```
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            11251 non-null  int32  
 13  Status            0 non-null      float64
 14  unnamed1          0 non-null      float64
dtypes: float64(2), int32(1), int64(4), object(8)
memory usage: 1.2+ MB
```

Dropping Unwanted Columns from the table containing Null Values

In [20]:
```
df.drop(['Status','unnamed1'],axis=1,inplace = True)
```

In [21]:
```
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            11251 non-null  int32 
dtypes: int32(1), int64(4), object(8)
memory usage: 1.1+ MB
```

In [22]:
```
df.columns
```

Out[22]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [23]:
```
df.rename(columns = {'Amount':'Sale'}, inplace = True)
```

In [24]:
```
df.columns
```

Out[24]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Sale'],
      dtype='object')
```

In [25]:
```
df.describe()
```

Out[25]:

	User_ID	Age	Marital_Status	Orders	Sale
count	1.125100e+04	11251.000000	11251.000000	11251.000000	11251.000000
mean	1.003004e+06	35.421207	0.420318	2.489290	9453.609901
std	1.716125e+03	12.754122	0.493632	1.115047	5219.569169
min	1.000001e+06	12.000000	0.000000	1.000000	188.000000
25%	1.001492e+06	27.000000	0.000000	1.500000	5443.500000
50%	1.003065e+06	33.000000	0.000000	2.000000	8110.000000
75%	1.004430e+06	43.000000	1.000000	3.000000	12671.000000
max	1.006040e+06	92.000000	1.000000	4.000000	23952.000000

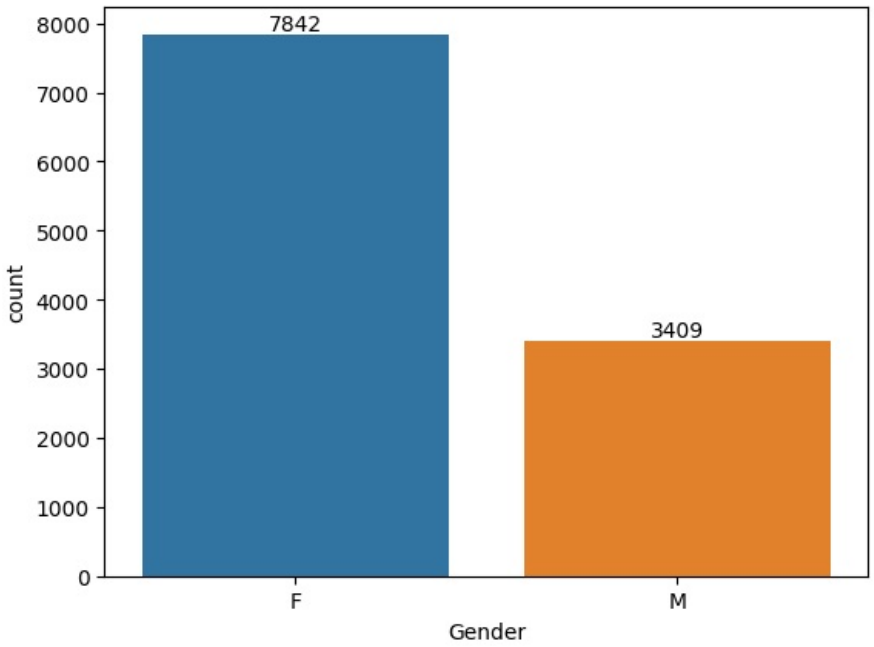
In [26]:
```
df['Sale'].describe()
```

Out[26]:
```
count    11251.000000
mean      9453.609901
std       5219.569169
min        188.000000
25%       5443.500000
50%       8110.000000
75%      12671.000000
max      23952.000000
Name: Sale, dtype: float64
```

Graph

Bar Chart showing Orders count between Males and Females

In [27]:
```
ax = sns.countplot(x='Gender',data = df)

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



As per above mentioned chart it is clear that Females have more purchasing power as compared to Males.

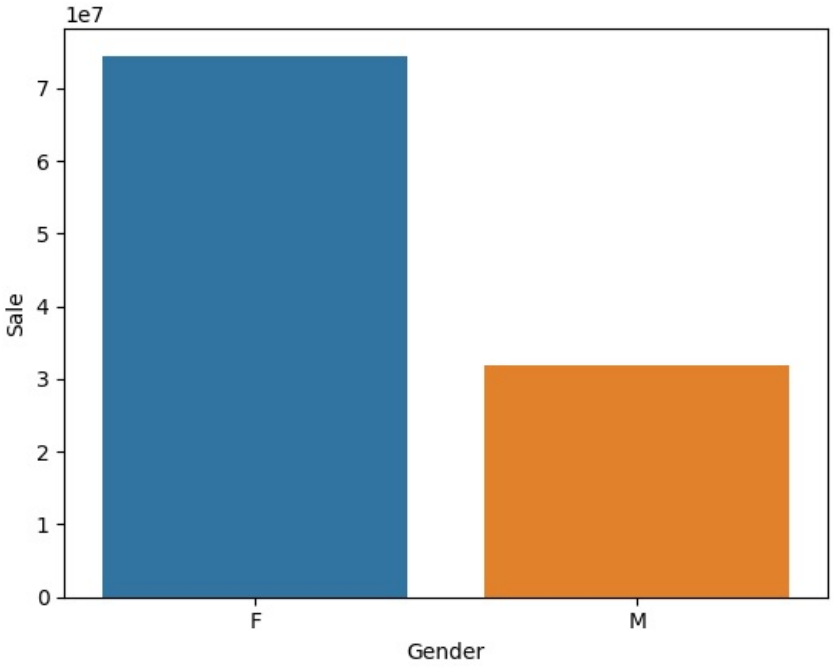
In [28]:
```
df.groupby(['Gender'], as_index=False)['Sale'].sum().sort_values(by='Sale',ascending = False)
```

Out[28]:

	Gender	Sale
0	F	74430383
1	M	31932182

Bar Chart Showing Sales amount between Males & Females

In [29]:
```
Sale_aud = df.groupby(['Gender'], as_index=False)['Sale'].sum().sort_values(by='Sale',ascending = False)
cx = sns.barplot(x = 'Gender', y='Sale', data = Sale_aud)
```



Result: Women have more spending power than men.

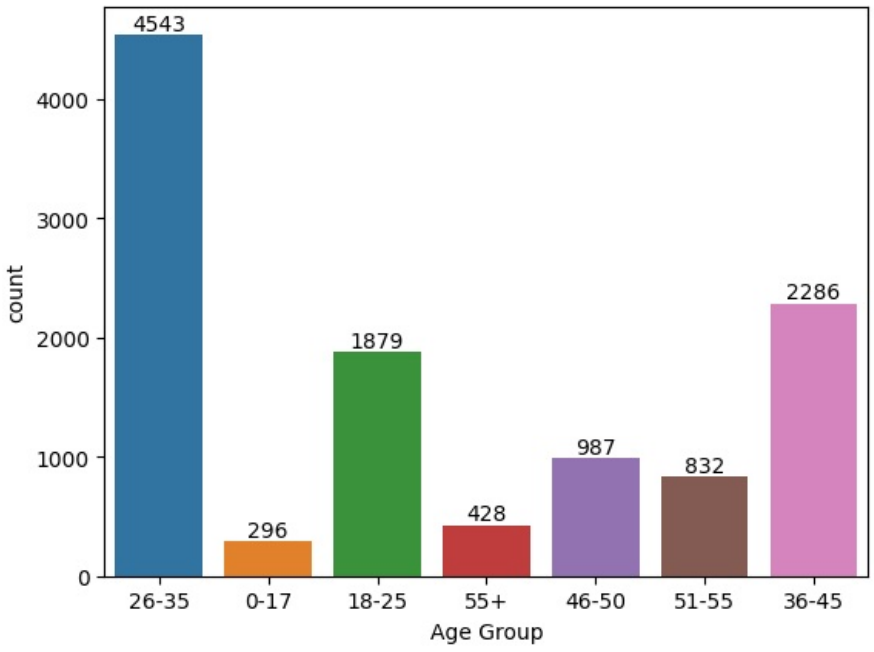
Age

In [30]:
```
df.columns
```

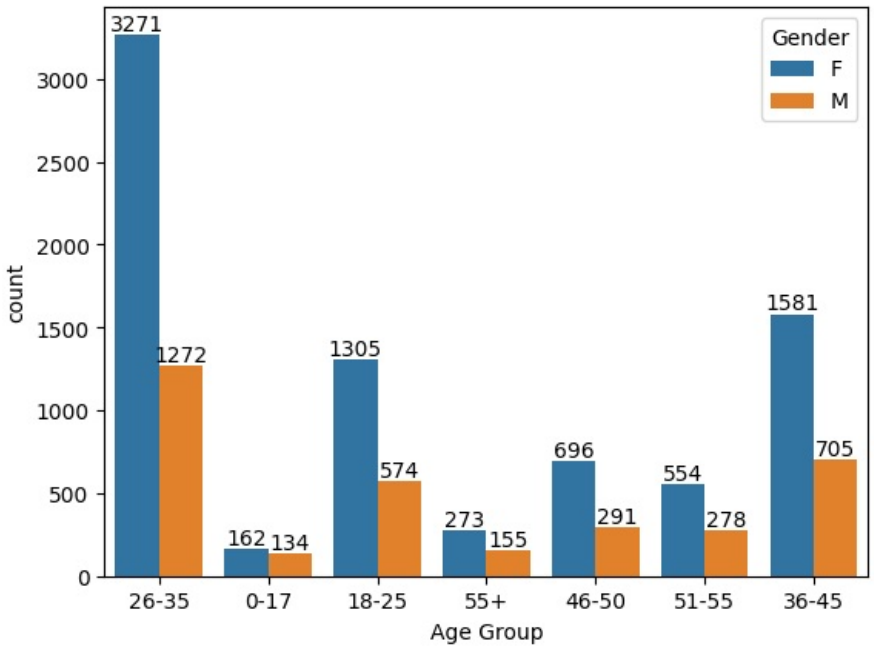
Out[30]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Sale'],
      dtype='object')
```

Bar Chart showing Orders count in between different age groups.

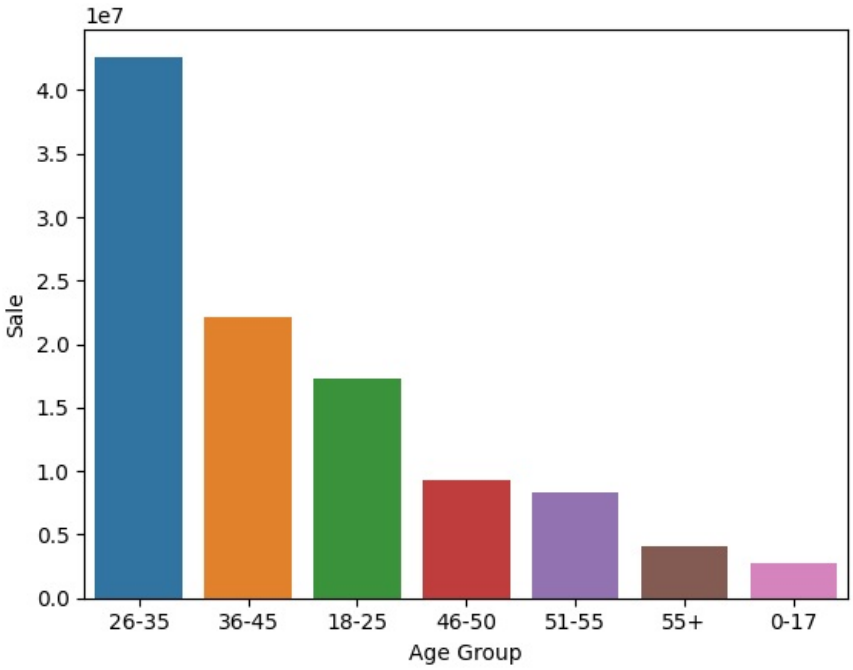
In [31]:
```
ax=sns.countplot(x='Age Group',data = df)
for i in ax.containers:
    ax.bar_label(i)
```



In [32]:
```
ax = sns.countplot(data = df,x='Age Group',hue = 'Gender')
for i in ax.containers:
    ax.bar_label(i)
```



In [33]:
```
Sale_age = df.groupby(['Age Group'], as_index=False)['Sale'].sum().sort_values(by='Sale',ascending = False)
cx = sns.barplot(x = 'Age Group', y='Sale', data = Sale_age)
```



Result: Again Women won the race of purchasing things and mostly women in between 26-35 years age group.

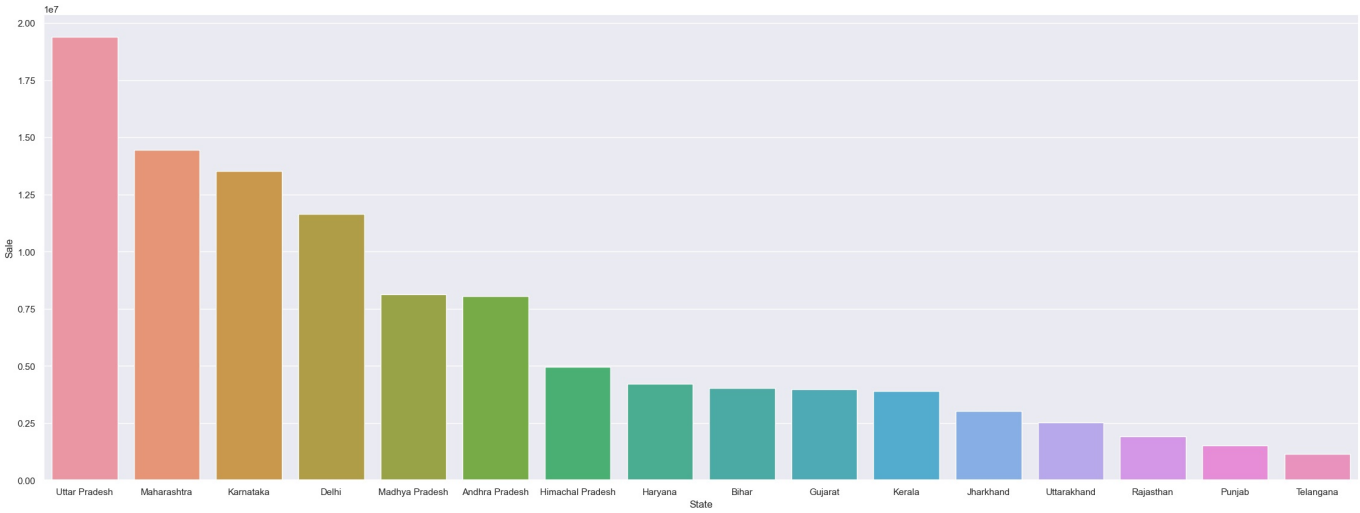
State

In [34]:
```
df.columns
```

Out[34]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Sale'],
      dtype='object')
```

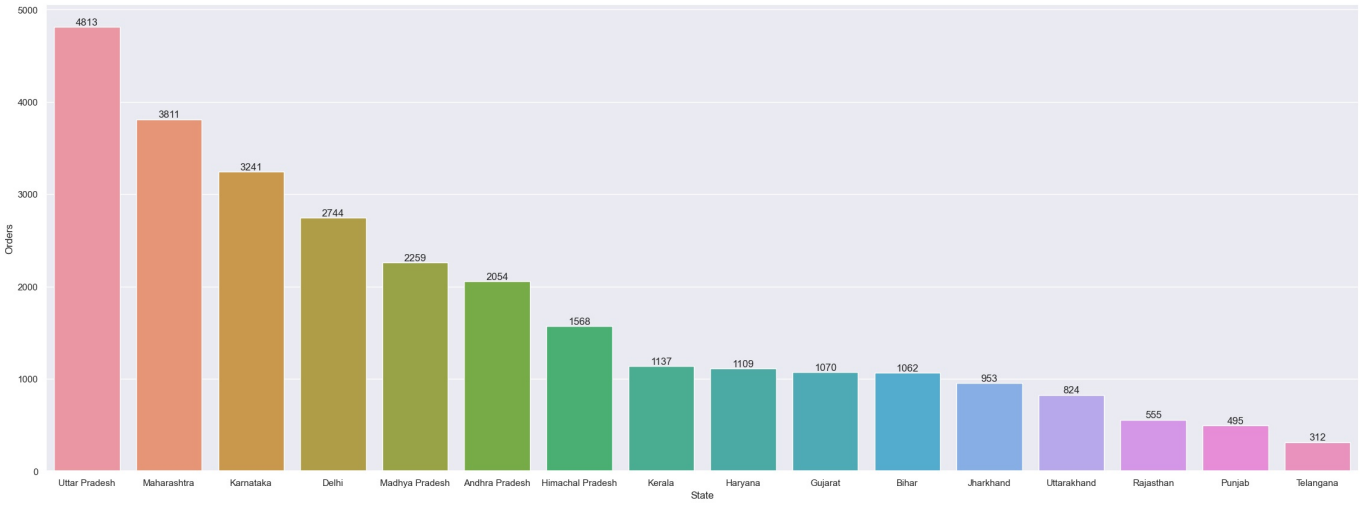
Sales per State

In [35]:
```
Sale_in = df.groupby(['State'], as_index = False)['Sale'].sum().sort_values(by='Sale',ascending=False).head(16)

sns.set(rc={'figure.figsize':(28,10)})
bx = sns.barplot(x='State',y= 'Sale',data = Sale_in)
```



Orders per State

In [36]:
```
Sale_in = df.groupby(['State'], as_index = False)['Orders'].sum().sort_values(by='Orders',ascending=False).head

sns.set(rc={'figure.figsize':(28,10)})
bx = sns.barplot(x='State',y= 'Orders',data = Sale_in)
for i in bx.containers:
    bx.bar_label(i)
```



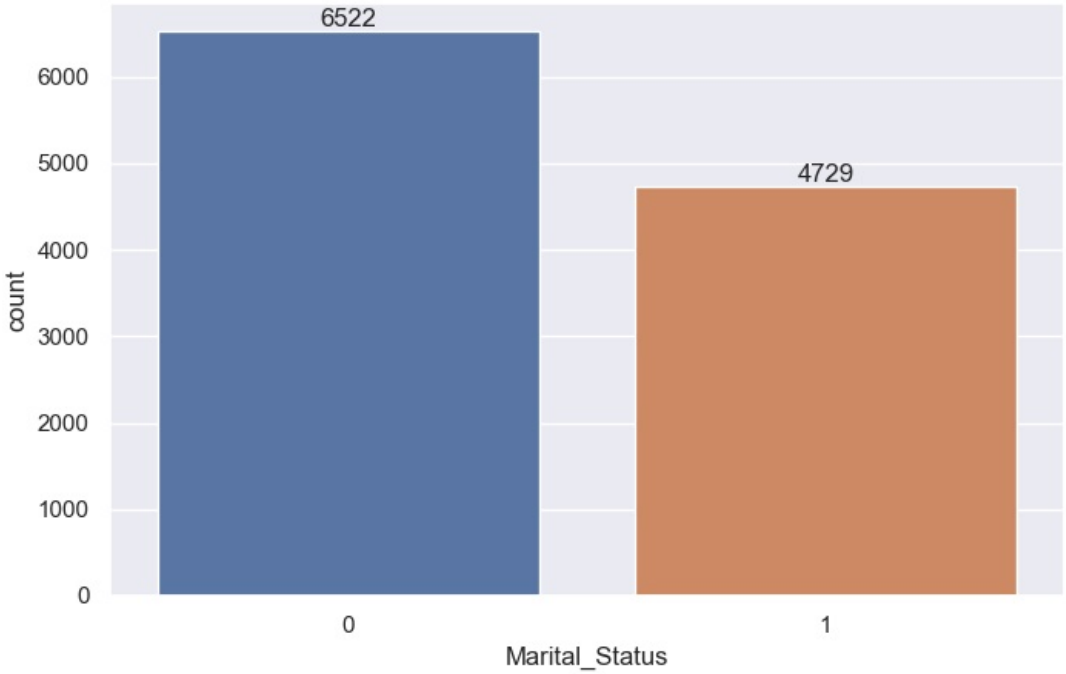
Results: As per above analysis, it is clear that most of the orders take place from Uttar Pradesh state and also UP's sales amount is also more than any other state.

Marital Status

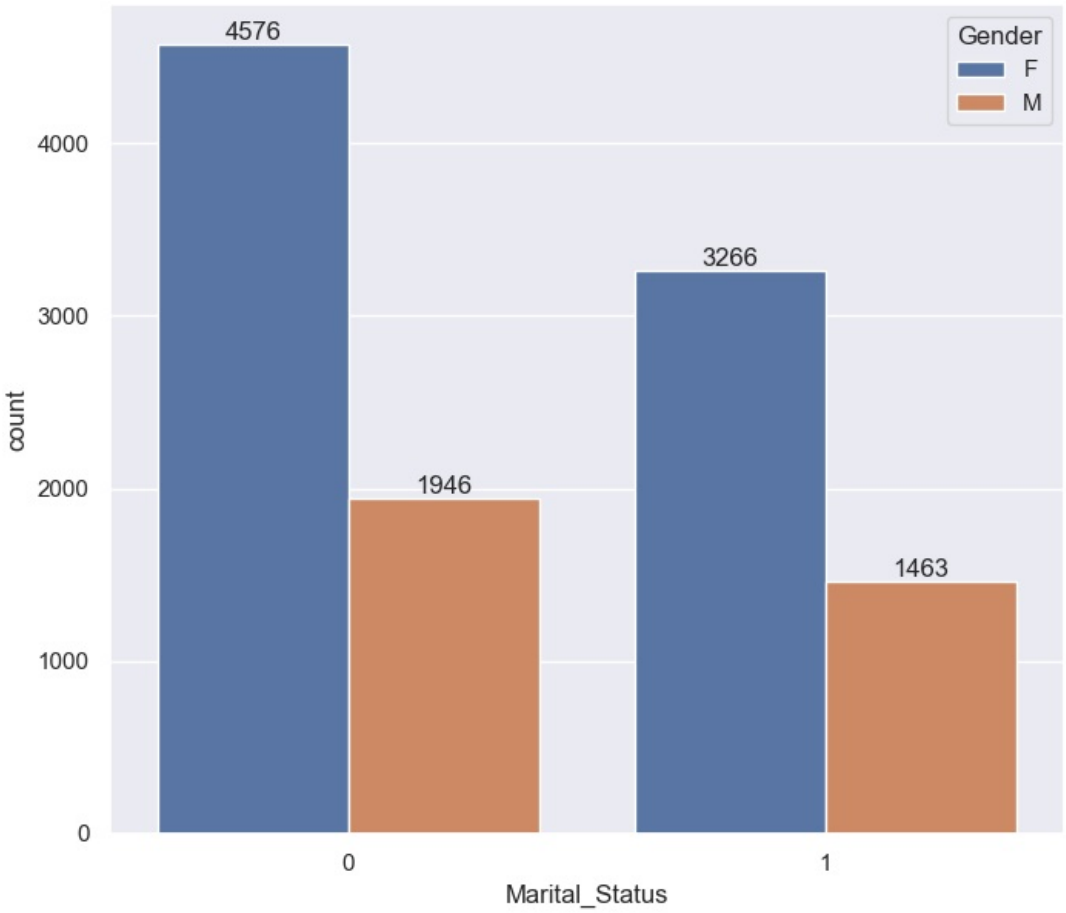
In [37]:
```
df.columns
```

Out[37]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Sale'],
      dtype='object')
```

In [56]:
```
ax=sns.countplot(x='Marital_Status',data = df)

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



In [55]:
```
ax=sns.countplot(x='Marital_Status',data = df, hue = 'Gender')

sns.set(rc={'figure.figsize':(8,5)})
for i in ax.containers:
    ax.bar_label(i)
```



Results: Married Women have more purchasing more than others.

Occupation

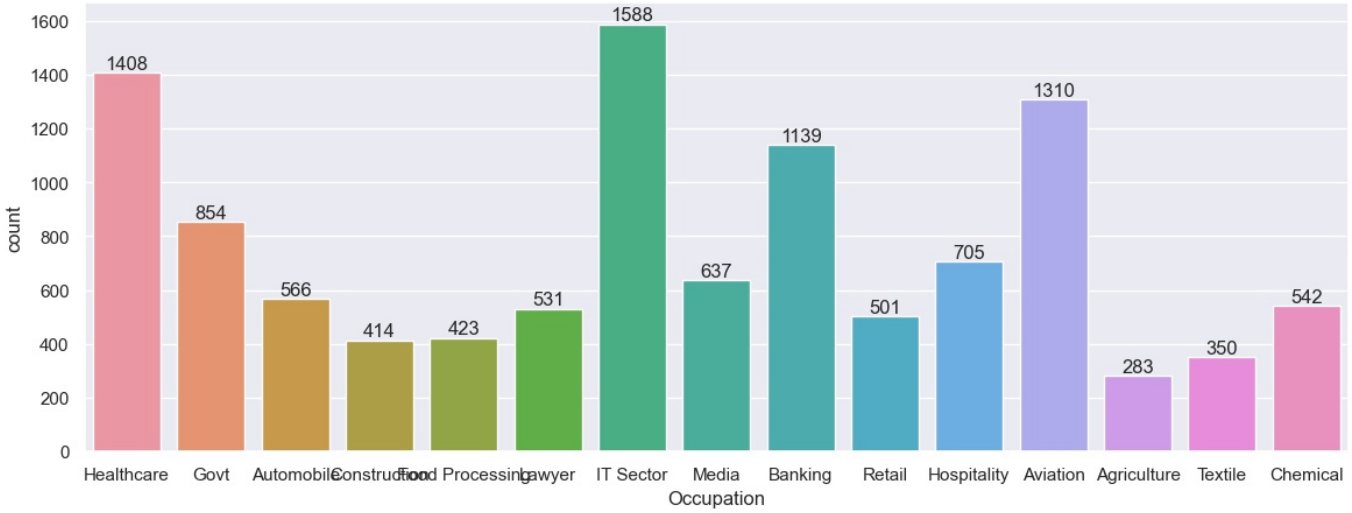
In [40]:
```
df.columns
```

Out[40]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Sale'],
      dtype='object')
```

Sector vs Orders

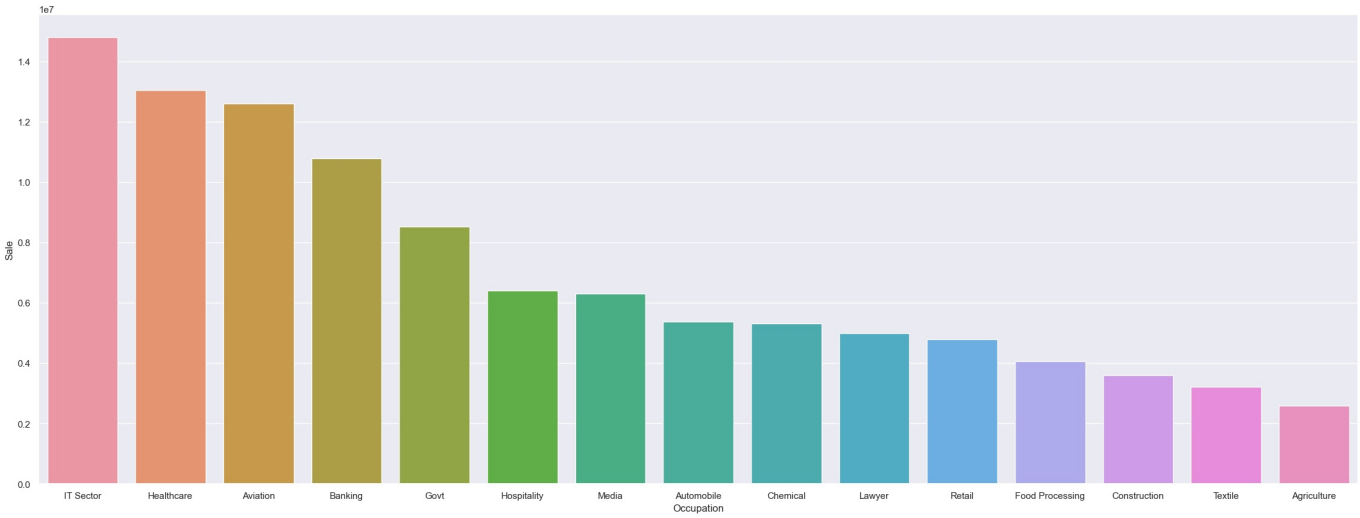
Sector vs Orders

In [41]:
```
ax=sns.countplot(x='Occupation',data = df)

sns.set(rc={'figure.figsize':(25,10)})
for i in ax.containers:
    ax.bar_label(i)
```



Sector vs Sale

In [42]:
```
Sale_in = df.groupby(['Occupation'], as_index = False)['Sale'].sum().sort_values(by='Sale',ascending=False).hea

sns.set(rc={'figure.figsize':(28,10)})
bx = sns.barplot(x='Occupation',y= 'Sale',data = Sale_in)
```



Results: As per above analysis we found that people from IT sectors are purchasing and Spending more as compare to other sectors.

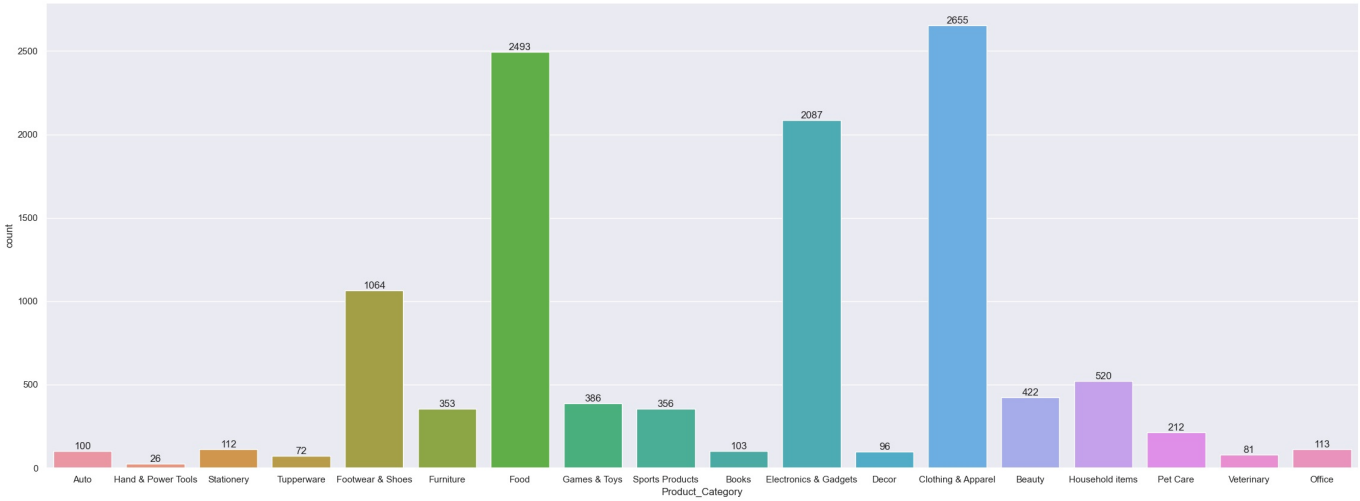
Product Category wise Orders and Sales

In [43]:
```
df.columns
```

Out[43]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Sale'],
      dtype='object')
```

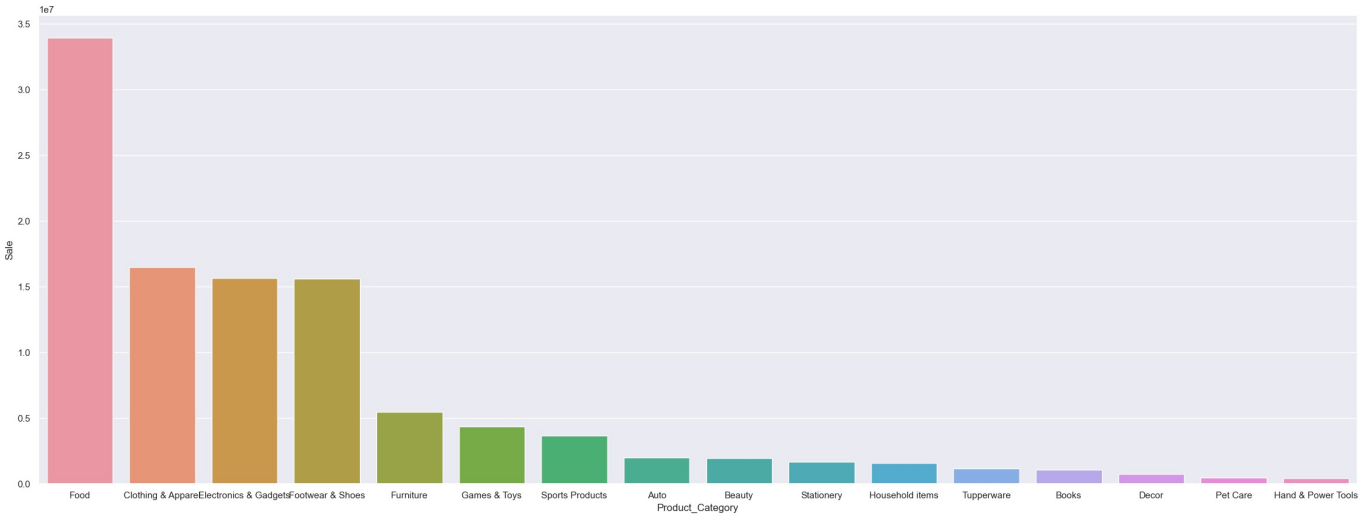
Product category vs Orders

In [44]:
```
ax=sns.countplot(x='Product_Category',data = df)

sns.set(rc={'figure.figsize':(25,10)})
for i in ax.containers:
    ax.bar_label(i)
```



Product Category vs Sales

In [45]:
```
Sale_in = df.groupby(['Product_Category'], as_index = False)['Sale'].sum().sort_values(by='Sale',ascending=Fals

sns.set(rc={'figure.figsize':(28,10)})
bx = sns.barplot(x='Product_Category',y= 'Sale',data = Sale_in)
```



Results: Now it is clear that people spending more on food as compare to other products.

Conclusion

1. As per above mentioned chart it is clear that Females have more purchasing power as compared to Males.
2. Females have more spending power than men.
3. Females of age group 26-35 spending more than other age groups.
4. Most of the orders take place from Uttar Pradesh state and also UP's sales amount is also more than any other state.
5. Married Women have more purchasing more than others.
6. People from IT sectors are purchasing and Spending more as compare to other sectors.
7. People spending more on food as compare to other products.

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: