Diwali Sales Exploratory Data Analysis

Downloading Important Libraries

In [1]:
```
!pip install pandas
!pip install matplotlib
!pip install seaborn
!pip install numpy
```

```
Requirement already satisfied: pandas in c:\users\agaur\anaconda3\lib\site-package
s (1.4.2)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\agaur\anaconda3
\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: numpy>=1.18.5 in c:\users\agaur\anaconda3\lib\site-
packages (from pandas) (1.21.5)
Requirement already satisfied: pytz>=2020.1 in c:\users\agaur\anaconda3\lib\site-p
ackages (from pandas) (2021.3)
Requirement already satisfied: six>=1.5 in c:\users\agaur\anaconda3\lib\site-packa
ges (from python-dateutil>=2.8.1->pandas) (1.16.0)
Requirement already satisfied: matplotlib in c:\users\agaur\anaconda3\lib\site-pac
kages (3.5.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\agaur\anaconda3\lib\site-
packages (from matplotlib) (9.0.1)
Requirement already satisfied: cycler>=0.10 in c:\users\agaur\anaconda3\lib\site-p
ackages (from matplotlib) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\agaur\anaconda3\lib\s
ite-packages (from matplotlib) (4.25.0)
Requirement already satisfied: numpy>=1.17 in c:\users\agaur\anaconda3\lib\site-pa
ckages (from matplotlib) (1.21.5)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\agaur\anaconda3\lib\s
ite-packages (from matplotlib) (1.3.2)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\agaur\anaconda3\lib\si
te-packages (from matplotlib) (3.0.4)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\agaur\anaconda3\li
b\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: packaging>=20.0 in c:\users\agaur\anaconda3\lib\sit
e-packages (from matplotlib) (21.3)
Requirement already satisfied: six>=1.5 in c:\users\agaur\anaconda3\lib\site-packa
ges (from python-dateutil>=2.7->matplotlib) (1.16.0)
Requirement already satisfied: seaborn in c:\users\agaur\anaconda3\lib\site-packag
es (0.11.2)
Requirement already satisfied: matplotlib>=2.2 in c:\users\agaur\anaconda3\lib\sit
e-packages (from seaborn) (3.5.1)
Requirement already satisfied: pandas>=0.23 in c:\users\agaur\anaconda3\lib\site-p
ackages (from seaborn) (1.4.2)
Requirement already satisfied: scipy>=1.0 in c:\users\agaur\anaconda3\lib\site-pac
kages (from seaborn) (1.7.3)
Requirement already satisfied: numpy>=1.15 in c:\users\agaur\anaconda3\lib\site-pa
ckages (from seaborn) (1.21.5)
Requirement already satisfied: packaging>=20.0 in c:\users\agaur\anaconda3\lib\sit
e-packages (from matplotlib>=2.2->seaborn) (21.3)
Requirement already satisfied: pillow>=6.2.0 in c:\users\agaur\anaconda3\lib\site-
packages (from matplotlib>=2.2->seaborn) (9.0.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\agaur\anaconda3\li
b\site-packages (from matplotlib>=2.2->seaborn) (2.8.2)
Requirement already satisfied: cycler>=0.10 in c:\users\agaur\anaconda3\lib\site-p
ackages (from matplotlib>=2.2->seaborn) (0.11.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\agaur\anaconda3\lib\si
te-packages (from matplotlib>=2.2->seaborn) (3.0.4)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\agaur\anaconda3\lib\s
ite-packages (from matplotlib>=2.2->seaborn) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\agaur\anaconda3\lib\s
ite-packages (from matplotlib>=2.2->seaborn) (1.3.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\agaur\anaconda3\lib\site-p
ackages (from pandas>=0.23->seaborn) (2021.3)
Requirement already satisfied: six>=1.5 in c:\users\agaur\anaconda3\lib\site-packa
ges (from python-dateutil>=2.7->matplotlib>=2.2->seaborn) (1.16.0)
Requirement already satisfied: numpy in c:\users\agaur\anaconda3\lib\site-packages
(1.21.5)
```

Importing Libraries

In [2]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline
```

Reading CSV File

In [18]:
```
df = pd.read_csv(r'C:\Users\agaur\OneDrive\Desktop\Diwali.csv',encoding='unicode_es
```

Data Cleaning

In [19]:
```
df.head(10)
```

Out[19]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western
5	1000588	Joni	P00057942	M	26-35	28	1	Himachal Pradesh	Northern
6	1001132	Balk	P00018042	F	18-25	25	1	Uttar Pradesh	Central
7	1002092	Shivangi	P00273442	F	55+	61	0	Maharashtra	Western
8	1003224	Kushal	P00205642	M	26-35	35	0	Uttar Pradesh	Central
9	1003650	Ginny	P00031142	F	26-35	26	1	Andhra Pradesh	Southern

In [20]:
```
df.shape
```

Out[20]:
```
(11251, 15)
```

In [21]:
```
df.columns
```

Out[21]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount', 'Status', 'unnamed1'],
      dtype='object')
```

In [22]:
```
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 [23]:
```
df.isnull().sum()
```

Out[23]:
```
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 [25]:
```
df.drop(['Status','unnamed1'],axis=1,inplace=True)
```

In [28]:
```
df.dropna(inplace=True)
```

In [29]:
```
df['Amount'] = df['Amount'].astype('int')
```

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

Out[30]:
```
(11239, 13)
```

In [33]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11239 entries, 0 to 11250
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   User_ID           11239 non-null  int64
 1   Cust_name         11239 non-null  object
 2   Product_ID        11239 non-null  object
 3   Gender            11239 non-null  object
 4   Age Group         11239 non-null  object
 5   Age               11239 non-null  int64
 6   Marital_Status    11239 non-null  int64
 7   State             11239 non-null  object
 8   Zone              11239 non-null  object
 9   Occupation        11239 non-null  object
 10  Product_Category  11239 non-null  object
 11  Orders            11239 non-null  int64
 12  Amount            11239 non-null  int32
dtypes: int32(1), int64(4), object(8)
memory usage: 1.2+ MB
```

In [34]:
```
df.rename(columns={'Marital_Status':'Marriage'})
```

Out[34]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marriage	State	Zor
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Weste
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southe
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Centr
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southe
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Weste
...	...	...	...	...	...	...	...	...	
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Weste
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northe
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Centr
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southe
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Weste

11239 rows × 13 columns

In [36]:
```
df.describe()
```

Out[36]:

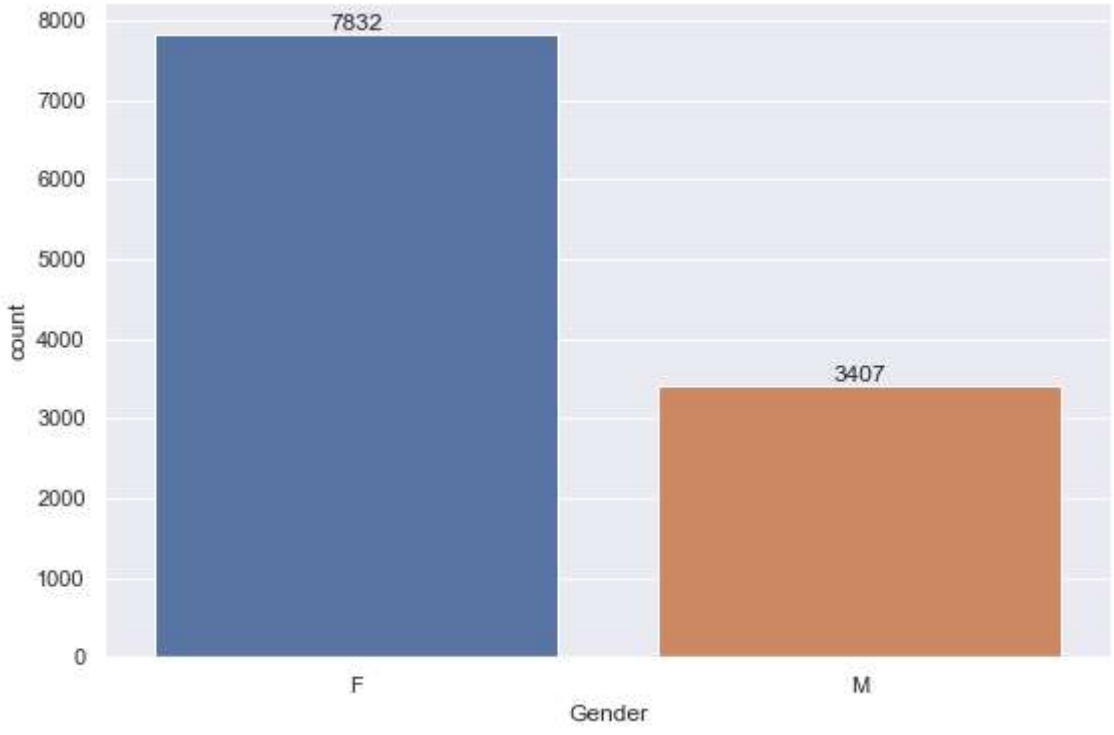
	User_ID	Age	Marital_Status	Orders	Amount
count	1.123900e+04	11239.000000	11239.000000	11239.000000	11239.000000
mean	1.003004e+06	35.410357	0.420055	2.489634	9453.610553
std	1.716039e+03	12.753866	0.493589	1.114967	5222.355168
min	1.000001e+06	12.000000	0.000000	1.000000	188.000000
25%	1.001492e+06	27.000000	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

EDA/ Exploratory Data Analysis

Gender

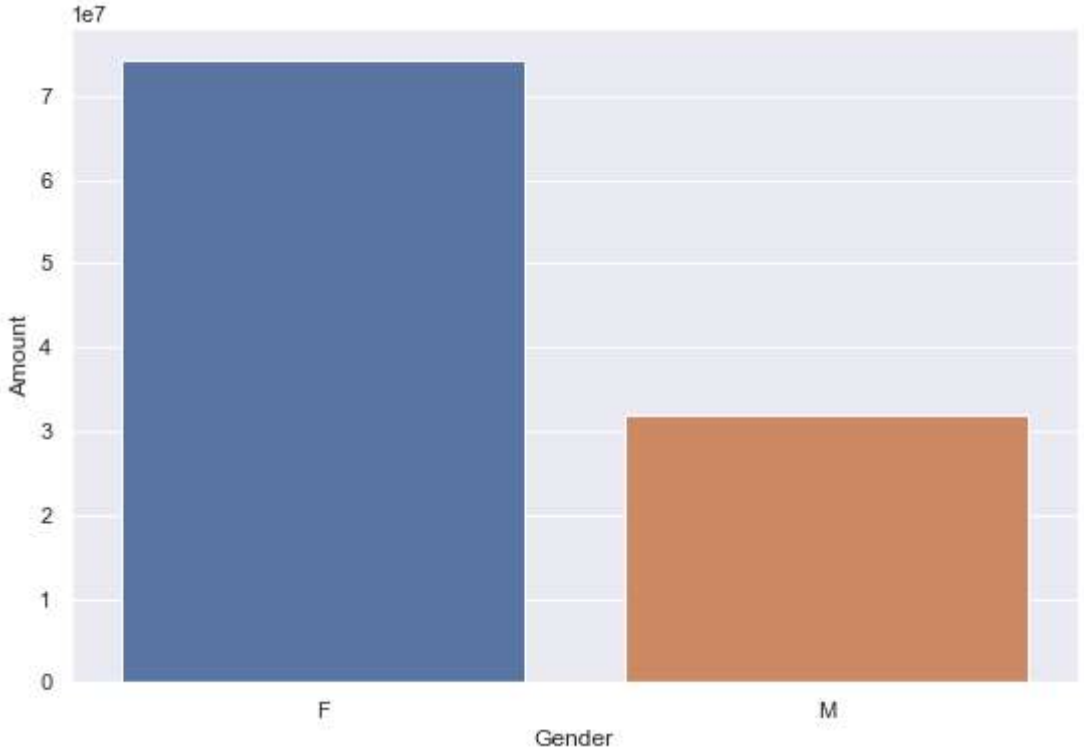
In [45]:
```
#Bar chart based on gender with count
```

In [44]:
```
ax = sns.countplot(x = 'Gender',data = df)
sns.set(rc={'figure.figsize':(9,6)})
for bar in ax.containers:
    ax.bar_label(bar)
```



In [48]:
```
# plot a bar char for Gender vs Amount
sales_amount = df.groupby(['Gender'],as_index=False)['Amount'].sum().sort_values(by
sns.barplot(x='Gender',y='Amount',data = sales_amount)
```

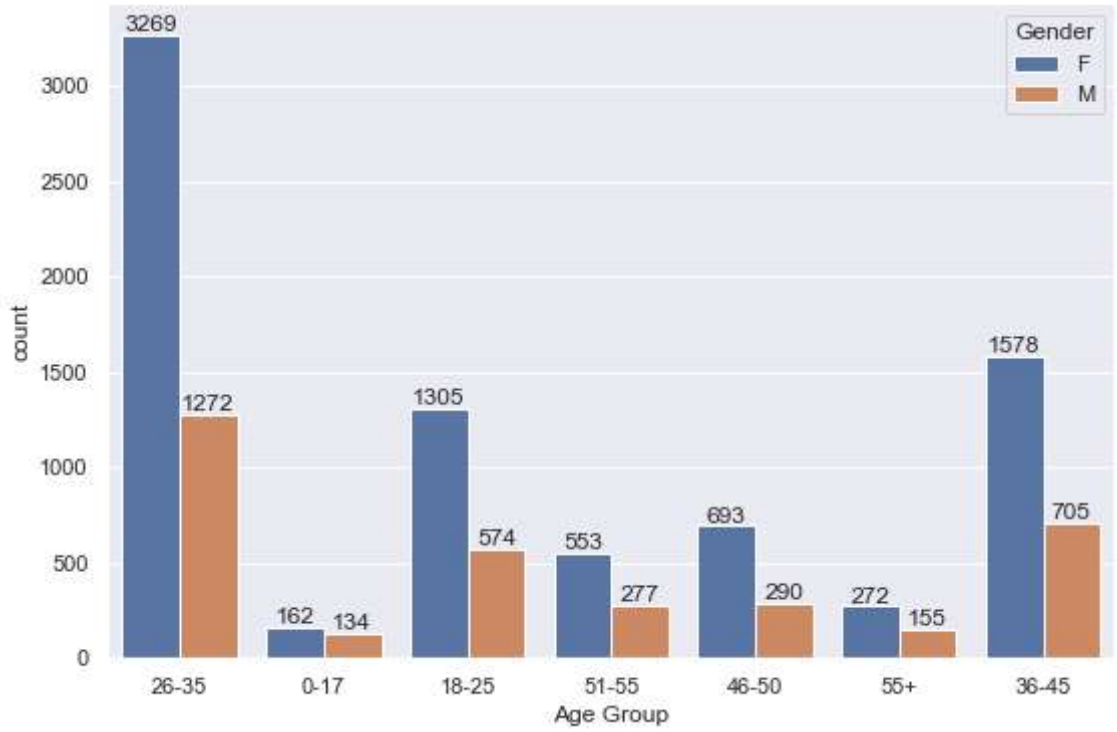
Out[48]:
```
<AxesSubplot:xlabel='Gender', ylabel='Amount'>
```



In [49]:
```
#Note The values on Y axis(Amount) is in (10^7)
```

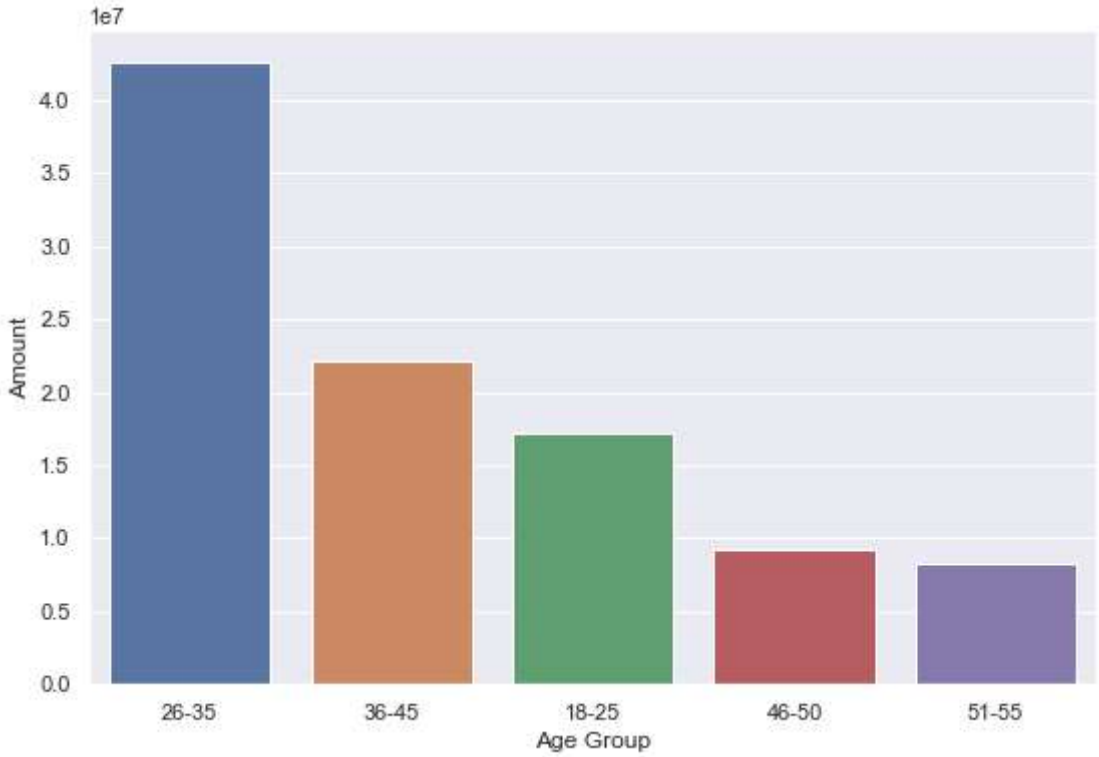
Age

In [52]:
```
ax = sns.countplot(x='Age Group',hue='Gender',data = df)
for bar in ax.containers:
    ax.bar_label(bar)
```



In [56]:
```
sales_age = df.groupby(['Age Group'],as_index=False)['Amount'].sum().sort_values(by
sns.barplot(x='Age Group',y='Amount',data = sales_age)
```

Out[56]:
```
<AxesSubplot:xlabel='Age Group', ylabel='Amount'>
```

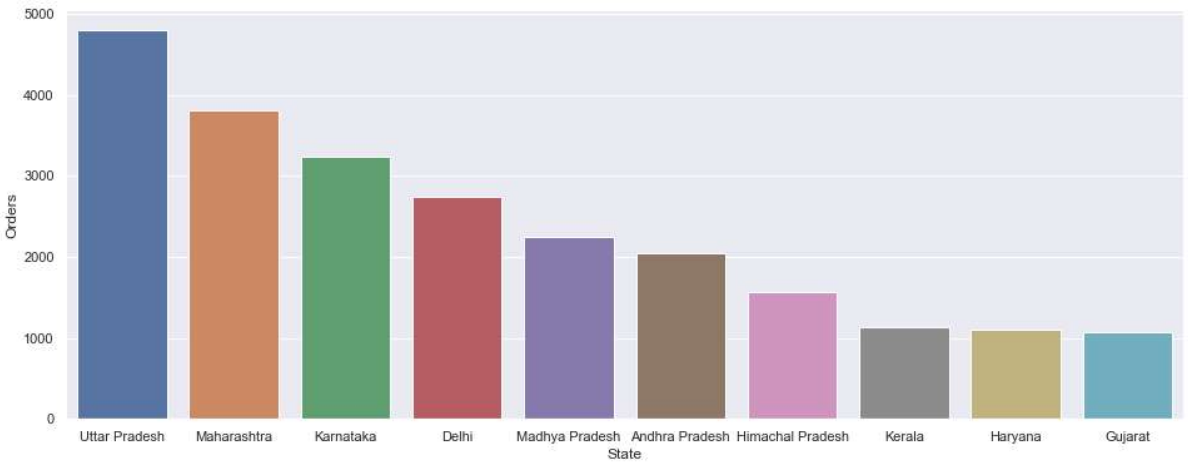


In [53]:
```
#we can clearly see that females with age group of 26-35 contributes more in overal
```

States

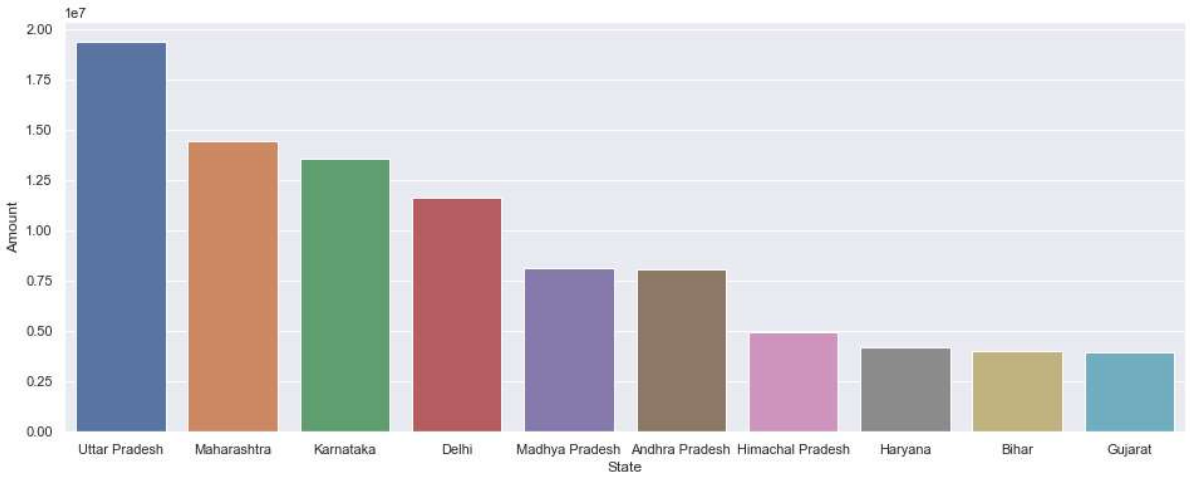
In [63]:
```
#total number of orders from the top 10 states
sales_state = df.groupby(['State'],as_index=False)['Orders'].sum().sort_values(by=
sns.set(rc={'figure.figsize':(16,6)})
sns.barplot(x='State',y='Orders',data = sales_state)
```

Out[63]:
```
<AxesSubplot:xlabel='State', ylabel='Orders'>
```



In [66]:
```
#total sales from the top 10 states
sales_state = df.groupby(['State'],as_index=False)['Amount'].sum().sort_values(by=
sns.set(rc={'figure.figsize':(16,6)})
sns.barplot(x='State',y='Amount',data=sales_state)
```

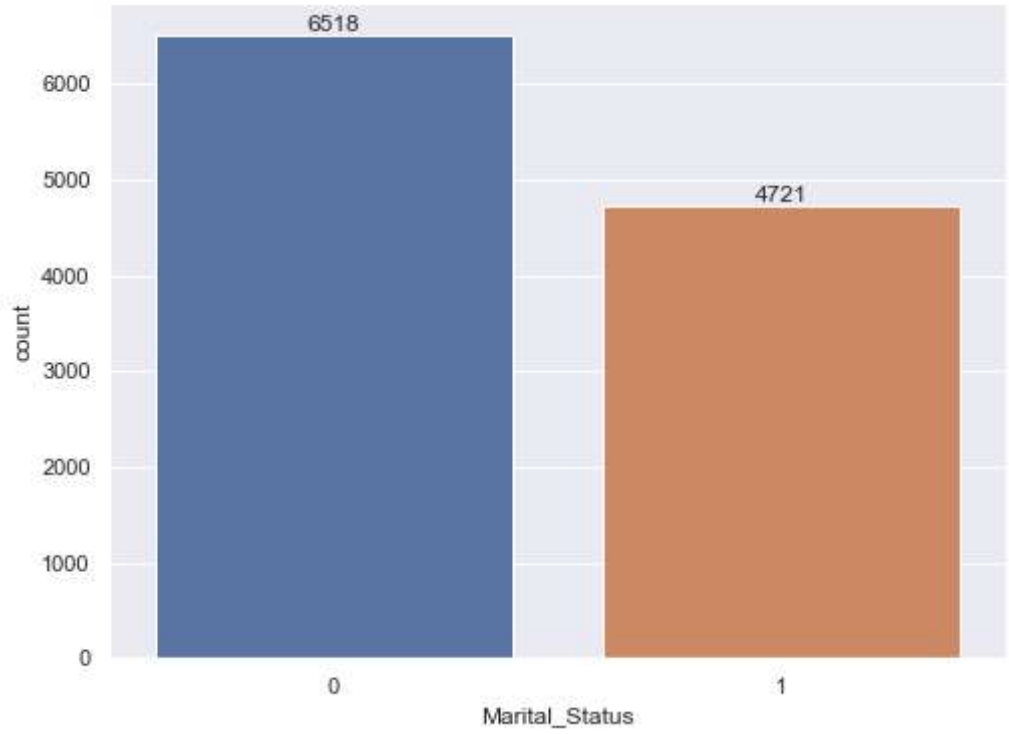
Out[66]:
```
<AxesSubplot:xlabel='State', ylabel='Amount'>
```



In [67]:
```
#From the above information it is very clear that Uttar Pradesh, Maharashtra and Ka
```

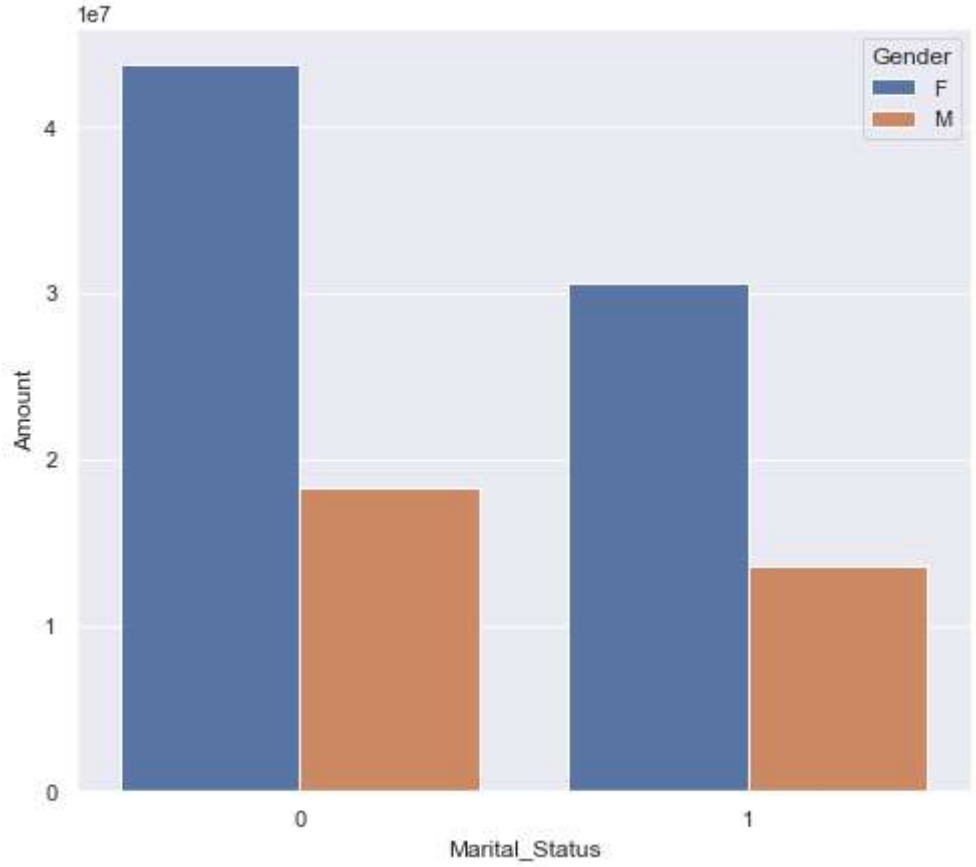
Marital Status

In [71]:
```
ax = sns.countplot(x = 'Marital_Status',data = df)
sns.set(rc={'figure.figsize':(7,6)})
for bar in ax.containers:
    ax.bar_label(bar)
```



In [74]:
```
sales_Marriage = df.groupby(['Marital_Status','Gender'],as_index=False)['Amount'].s
sns.set(rc={'figure.figsize':(8,7)})
sns.barplot(x='Marital_Status',y='Amount',hue='Gender',data = sales_Marriage)
```

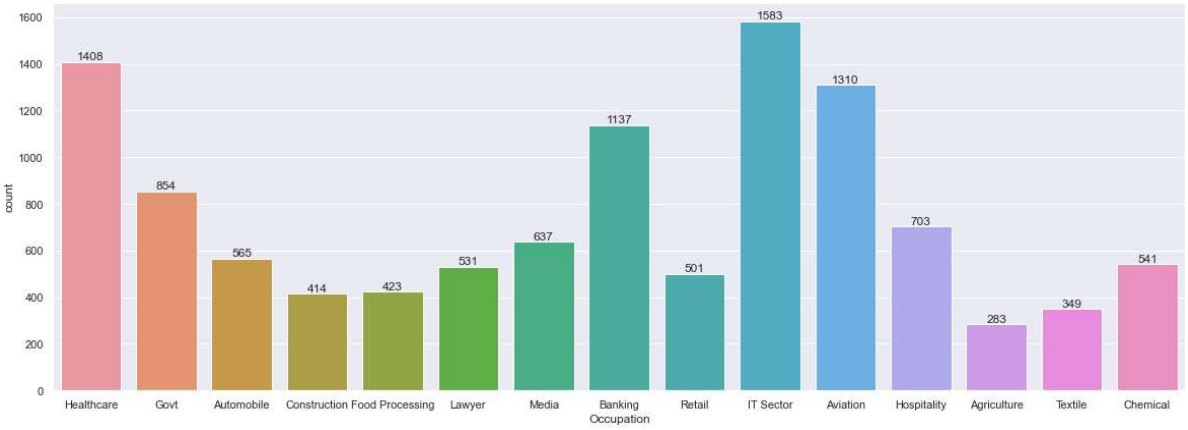
Out[74]:
```
<AxesSubplot:xlabel='Marital_Status', ylabel='Amount'>
```



In [75]:
```
#from the above information it is very clear that most of the purchase are from wor
```

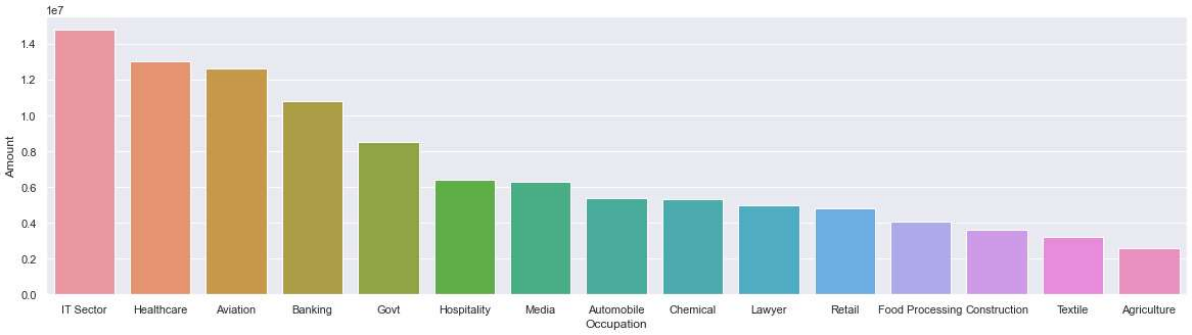
occupation

In [83]:
```
ax = sns.countplot(x = 'Occupation',data = df)
sns.set(rc={'figure.figsize':(20,5)})
for bar in ax.containers:
    ax.bar_label(bar)
```



In [85]:
```
sales_sector = df.groupby(['Occupation'],as_index=False)['Amount'].sum().sort_value
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(x = 'Occupation',y='Amount',data = sales_sector)
```

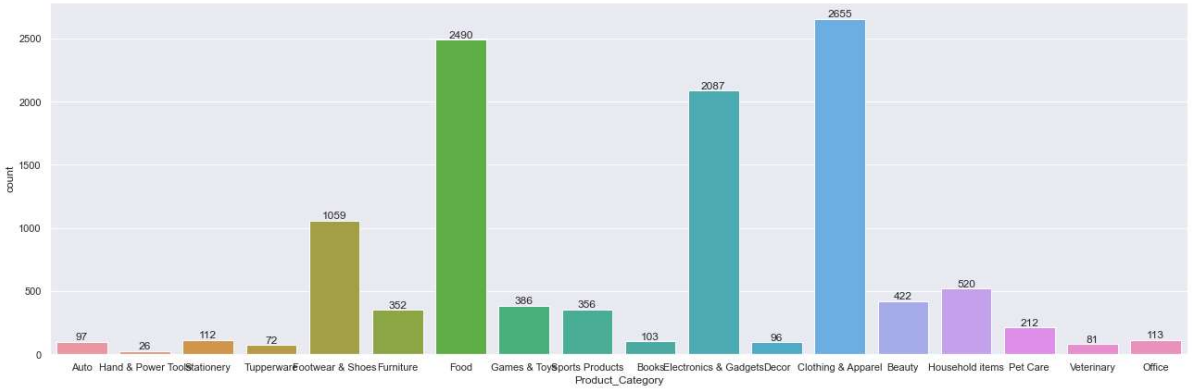
Out[85]:
```
<AxesSubplot:xlabel='Occupation', ylabel='Amount'>
```



In [86]:
```
#From the above information it is very clear that the people belong to IT Sector, H
```

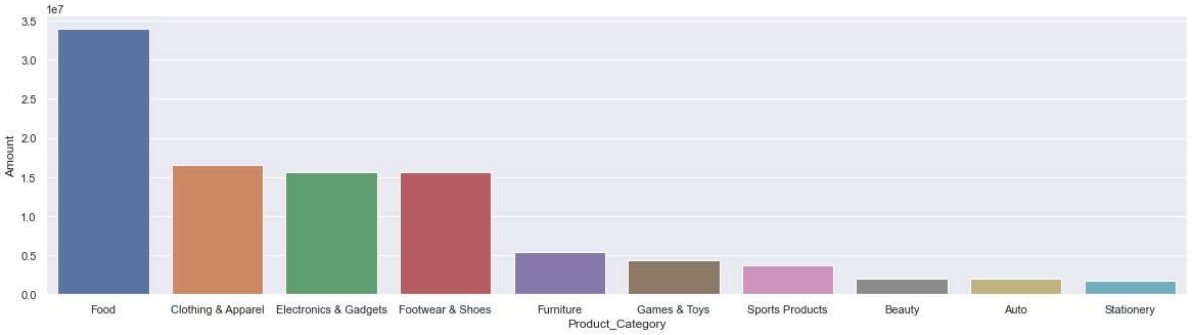
Product Category

In [90]:
```
ax = sns.countplot(x = 'Product_Category',data = df)
sns.set(rc={'figure.figsize':(20,5)})
for bar in ax.containers:
    ax.bar_label(bar)
```



In [91]:
```
sales_product = df.groupby(['Product_Category'],as_index=False)['Amount'].sum().sor
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(x='Product_Category',y='Amount',data = sales_product)
```

Out[91]:
```
<AxesSubplot:xlabel='Product_Category', ylabel='Amount'>
```

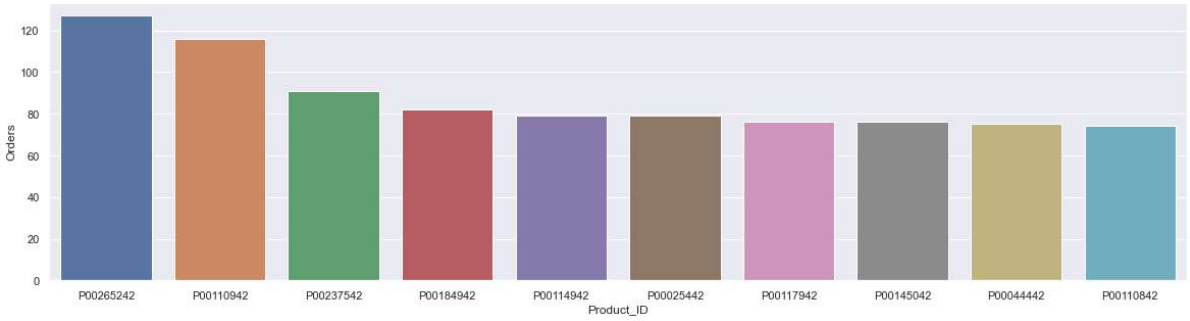


In [92]:
```
#we can clearly see that on the basis of info related to amount (food, clothing & A
```

Product Id

In [97]:
```
sales_id = df.groupby(['Product_ID'],as_index=False)['Orders'].sum().sort_values(by
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(x='Product_ID',y='Orders',data = sales_id)
```

Out[97]: <AxesSubplot:xlabel='Product_ID', ylabel='Orders'>



In [1]:
```
#Top 10 product id by orders.
```

Conclusion

it is very clear form the above exploratory Data Analysis that married women form the age group of 26-35 form the state of UP, Maharashtra and Karnataka working in IT-Sector, Healthcare and Aviation industry are more likely to buy products form the category Food, Clothing and Electronics.

Thank You :)