In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [2]:
```
df = pd.read_csv(r'C:\Users\ankit\Downloads\Diwali_Sales_Dataset\Diwali Sale
```

In [3]:
```
df.shape
```

Out[3]:
```
(11251, 15)
```

In [4]:
```
df.head()
```

Out[4]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	W
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	So
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	C
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	So
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	W

In [5]:
```
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 [6]:
```
# drop blank columns
df.drop(['Status','unnamed1'], axis = 1, inplace = True)
```

In [7]:
```
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 [8]:
```
pd.isnull(df)
```

Out[8]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...
11246	False	False	False	False	False	False	False	False	False
11247	False	False	False	False	False	False	False	False	False
11248	False	False	False	False	False	False	False	False	False
11249	False	False	False	False	False	False	False	False	False
11250	False	False	False	False	False	False	False	False	False

11251 rows × 13 columns

In [9]:
```
pd.isnull(df).sum()
```

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

Out[10]:
```
(11251, 13)
```

In [11]:
```
# drop null values
df.dropna(inplace = True)
```

In [12]:
```
df.shape
```

Out[12]:
```
(11239, 13)
```

In [13]:
```
# Change data types
df['Amount'] = df['Amount'].astype('int')
```

In [14]:
```
df['Amount'].dtype
```

Out[14]:
```
dtype('int32')
```

In [ ]:

In [ ]:

In [15]:
```
df.describe()
```

Out[15]:

	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

In [16]:
```
df[['Age','Orders','Amount']].describe()
```

Out[16]:

	Age	Orders	Amount
count	11239.000000	11239.000000	11239.000000
mean	35.410357	2.489634	9453.610553
std	12.753866	1.114967	5222.355168
min	12.000000	1.000000	188.000000
25%	27.000000	2.000000	5443.000000
50%	33.000000	2.000000	8109.000000
75%	43.000000	3.000000	12675.000000
max	92.000000	4.000000	23952.000000

In [ ]:
```

```

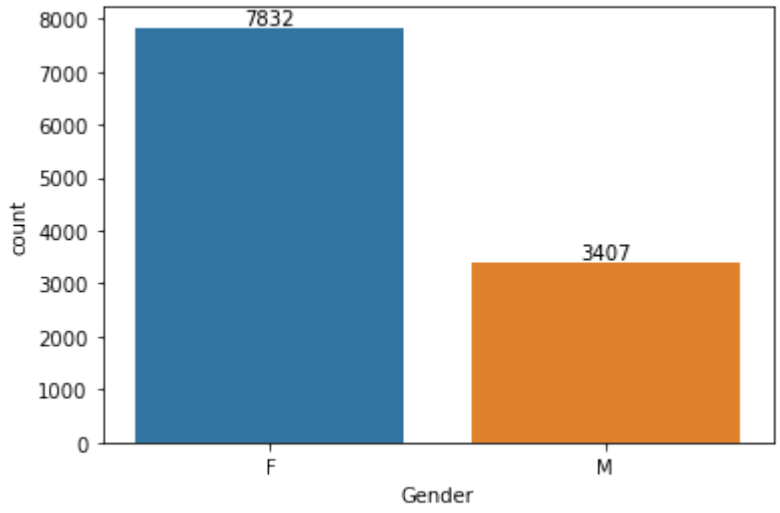
In [ ]:
```

```

Exploratory Data Analysis

In [17]:
```
df.columns
```

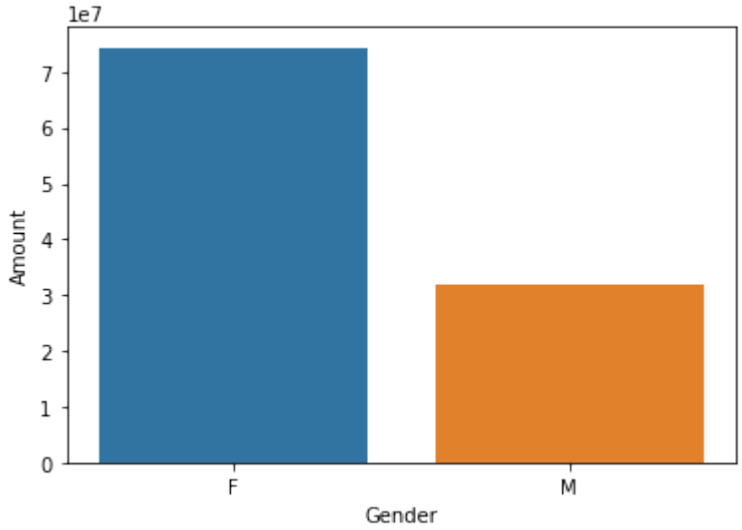
Out[17]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Categor
y',
       'Orders', 'Amount'],
      dtype='object')
```

In [18]:
```
ax = sns.countplot(x = 'Gender', data = df)

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



In [19]:
```
sales_gen = df.groupby(['Gender'], as_index = False)['Amount'].sum().sort_va

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

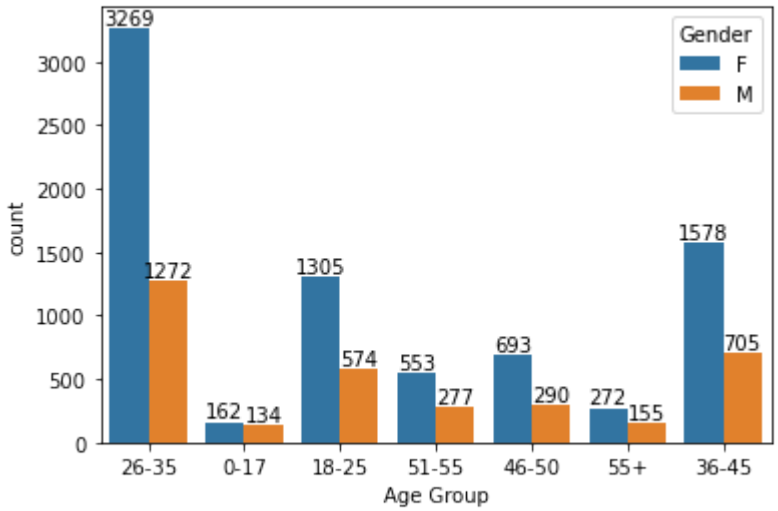
Out[19]: <AxesSubplot:xlabel='Gender', ylabel='Amount'>



In [ ]:

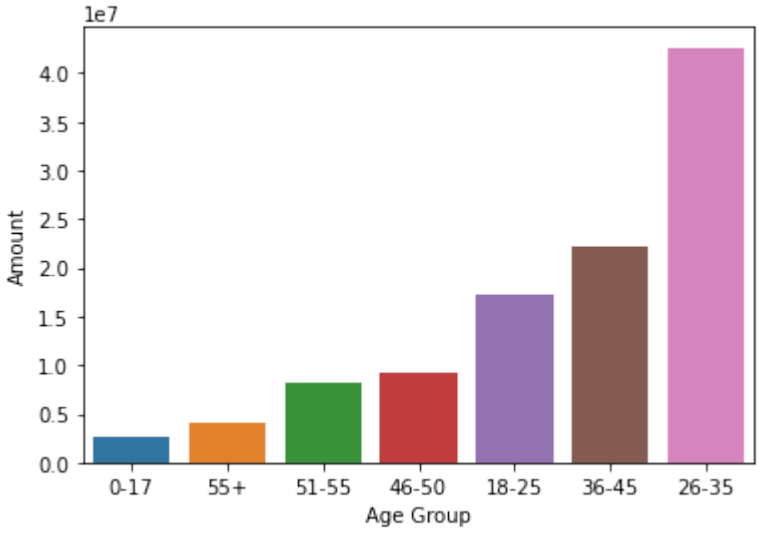
In [ ]:

In [20]:
```
ax = sns.countplot(data = df, x= 'Age Group', hue = 'Gender')

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



In [21]:
```
sales_age = df.groupby(['Age Group'], as_index = False)['Amount'].sum().sort

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

Out[21]: <AxesSubplot:xlabel='Age Group', ylabel='Amount'>

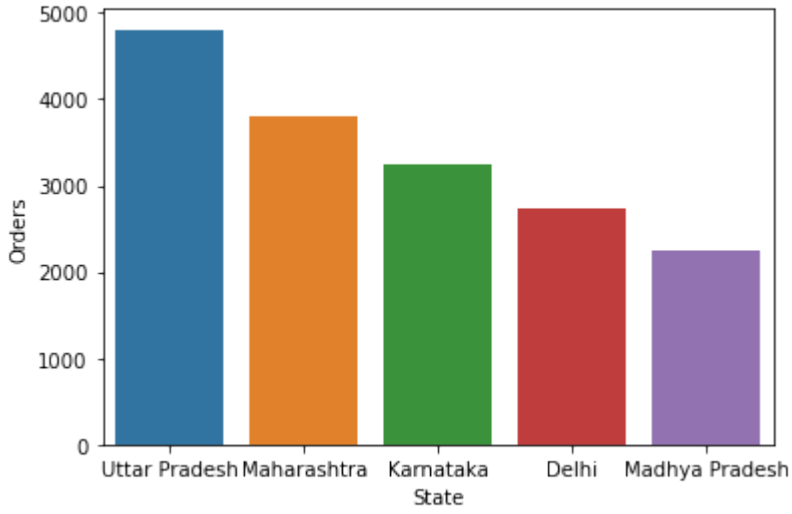


In [ ]:

In [ ]:

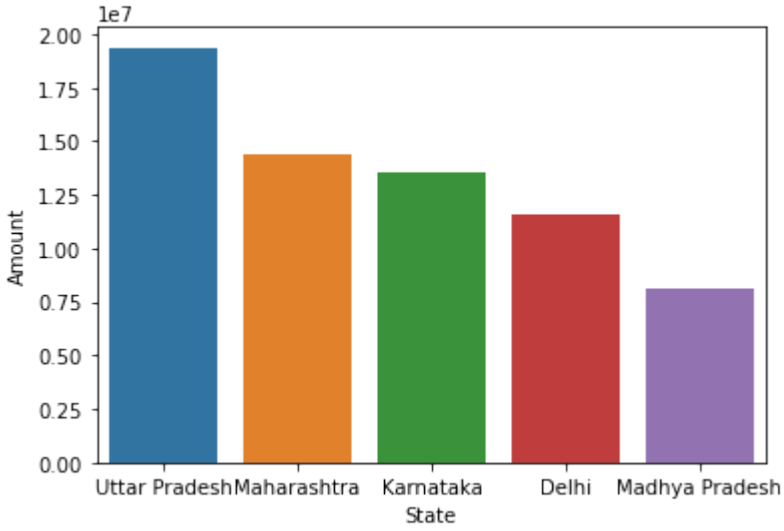
In [22]:
```
sales_state = df.groupby(['State'], as_index = False)['Orders'].sum().sort_v

sns.barplot(data = sales_state, x= "State", y = 'Orders')
```

Out[22]: <AxesSubplot:xlabel='State', ylabel='Orders'>

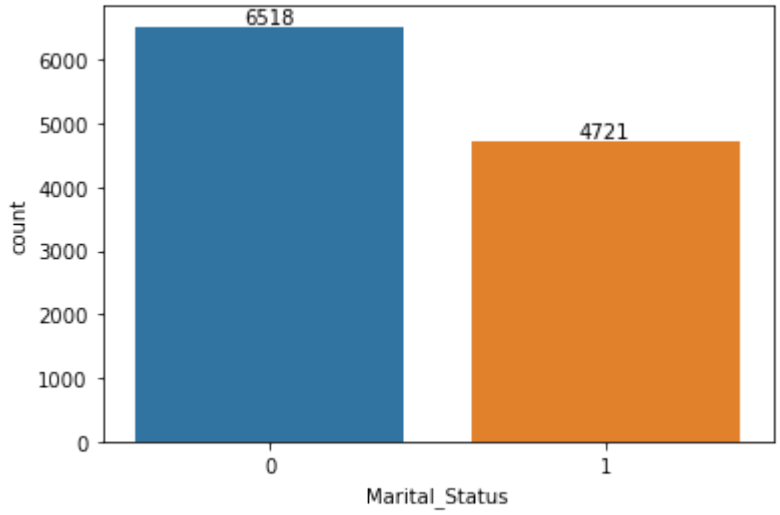


In [23]:
```
sales_state = df.groupby(['State'], as_index = False)['Amount'].sum().sort_v

sns.barplot(data = sales_state, x= "State", y = 'Amount')
```

Out[23]: <AxesSubplot:xlabel='State', ylabel='Amount'>

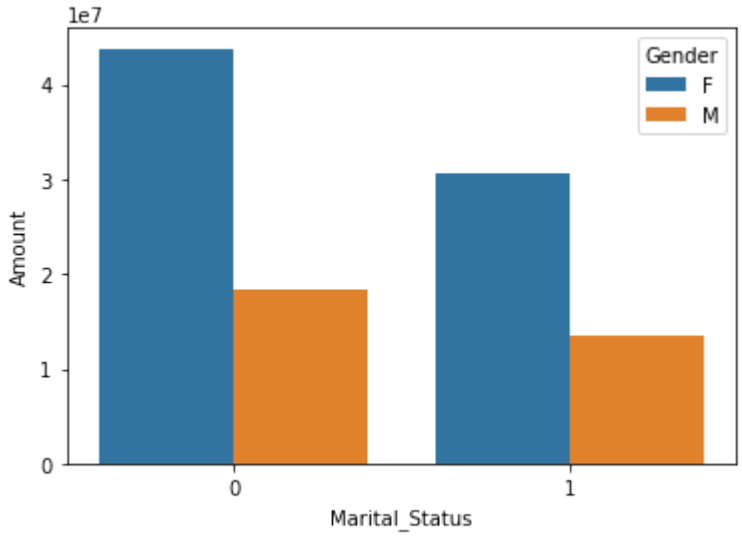


In [24]:
```
ax = sns.countplot(data = df, x = 'Marital_Status')

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



In [25]:
```
married_or_not = df.groupby(['Marital_Status','Gender'], as_index = False)['

sns.barplot(data = married_or_not, x= 'Marital_Status', y = 'Amount', hue =
```

Out[25]: <AxesSubplot:xlabel='Marital_Status', ylabel='Amount'>

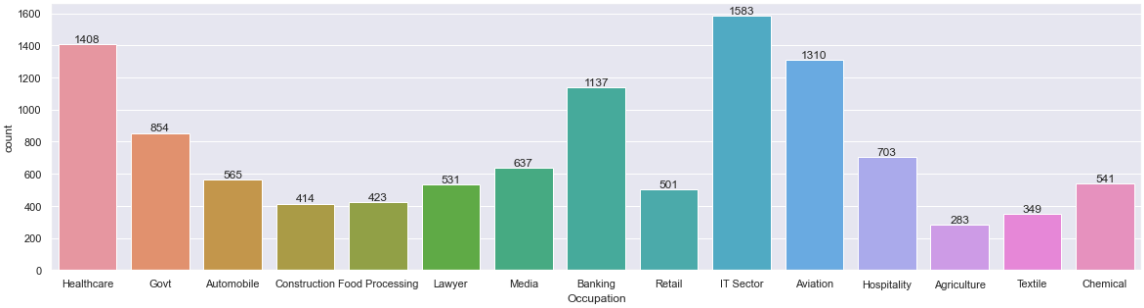


In [ ]:

In [ ]:

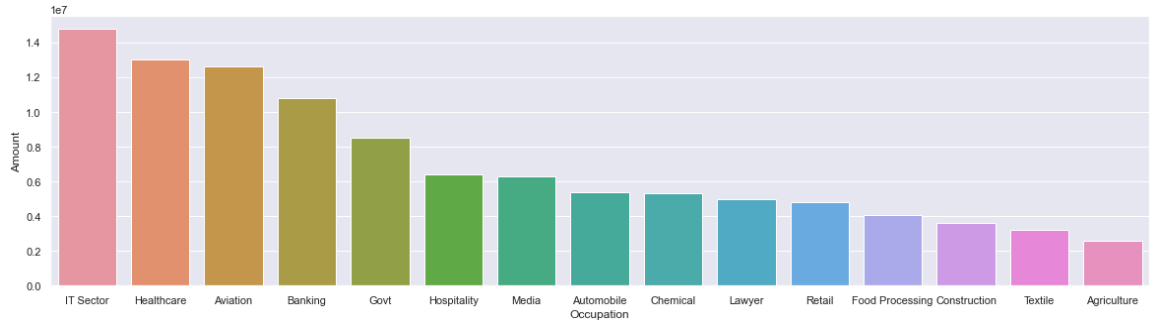
In [ ]:

In [26]:
```
sns.set(rc = {'figure.figsize':(20,5)})

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

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



In [27]:
```
purchase_by_occupation = df.groupby(['Occupation'], as_index = False)['Amour

sns.barplot(data = purchase_by_occupation, x= 'Occupation', y = 'Amount')
```

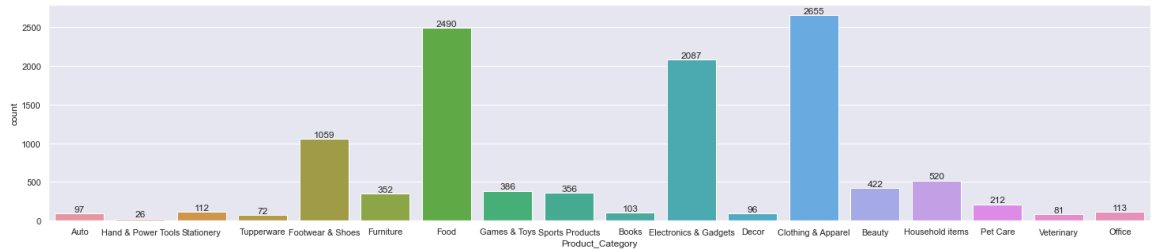
Out[27]: <AxesSubplot:xlabel='Occupation', ylabel='Amount'>



In [ ]:

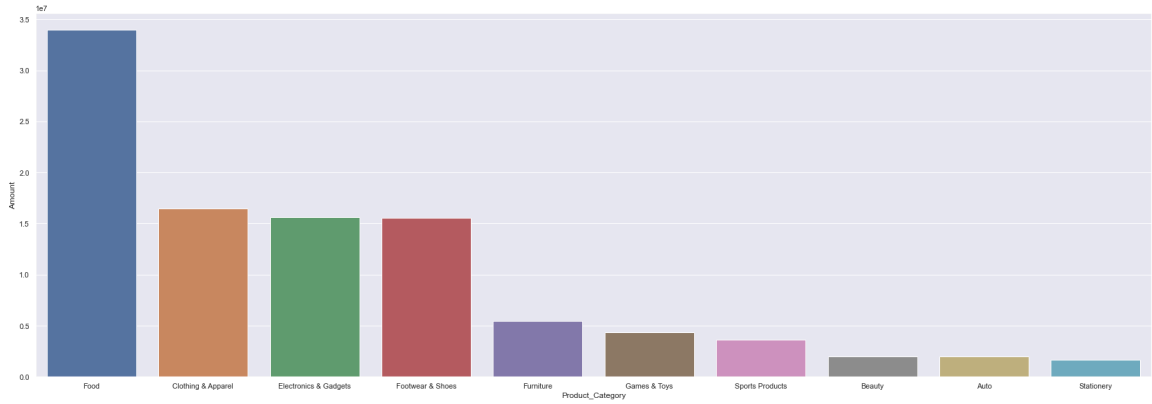
In [ ]:

In [28]:
```
sns.set(rc = {'figure.figsize':(25,5)})

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

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



In [29]:
```
purchase_by_product_category = df.groupby(['Product_Category'], as_index = F

sns.set(rc = {'figure.figsize':(30,10)})
sns.barplot(data = purchase_by_product_category, x= 'Product_Category', y =
```

Out[29]: <AxesSubplot:xlabel='Product_Category', ylabel='Amount'>



In [ ]:

In [ ]:

In [ ]: