In [1]:
```python
import pandas as pd
import numpy as np
import seaborn as sms
import matplotlib.pyplot as plt
```

In [12]:
```python
import os
```

In [14]:
```python
#import data
os.listdir (r"D:\Downloads\Python_Diwali_Sales_Analysis\Python_Diwali_Sales_Analysis
```

Out[14]:
```
['Diwali Sales Data.csv',
 'Diwali_Sales_Analysis.ipynb',
 'DIWALI_SALES_PROJECT.ipynb']
```

In [20]:
```python
sale_data = pd.read_csv (r"D:\Downloads\Python_Diwali_Sales_Analysis\Python_Diwali_S
```

In [21]:
```python
sale_data.shape
```

Out[21]:
```
(11251, 15)
```

In [22]:
```python
sale_data.head(10)
```

Out[22]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occu
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Hea
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Auto
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Const
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Proc
5	1000588	Joni	P00057942	M	26-35	28	1	Himachal Pradesh	Northern	Proc
6	1001132	Balk	P00018042	F	18-25	25	1	Uttar Pradesh	Central	L
7	1002092	Shivangi	P00273442	F	55+	61	0	Maharashtra	Western	IT
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 [23]:
```python
sale_data.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 [25]:
```python
#drop blank columns
sale_data.drop(['Status', 'unnamed1'], axis=1 , inplace=True)
```

In [26]:
```python
sale_data.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 [27]:
```python
#checking null values
pd.isnull(sale_data)
```

Out[27]:

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

11251 rows × 13 columns

In [28]:
```python
pd.isnull(sale_data).sum()
```

Out[28]:
```
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 [29]:
```python
#drop null values
sale_data.dropna(inplace= True)
```

In [30]:
```python
sale_data.shape
```

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

In [32]:
```python
#change dtype
sale_data['Amount'] = sale_data ['Amount'].astype('int')
```

In [34]:
```python
sale_data['Amount'].dtypes
```

Out[34]:
```
dtype('int32')
```

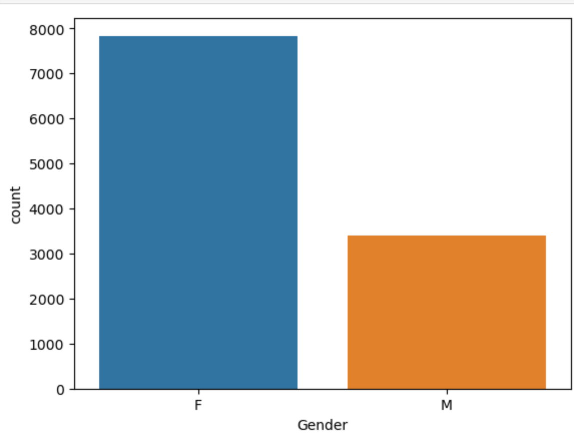
EDA

Gender

In [39]:
```python
sale_data.columns
```

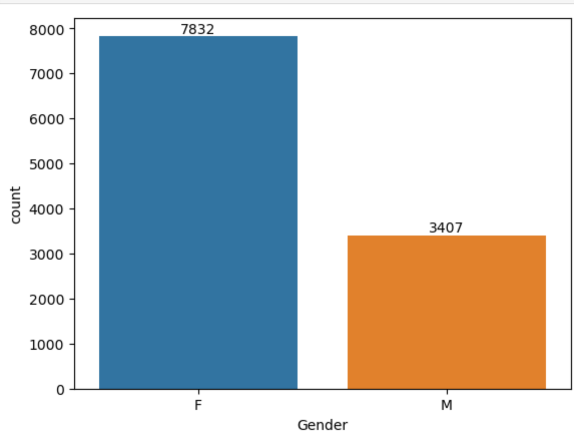
Out[39]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [45]:
```python
ax = sms.countplot(x= 'Gender' , data = sale_data)
```



In [46]:
```python
ax = sms.countplot(x= 'Gender' , data = sale_data)

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



In [47]:
```python
sale_data.groupby(['Gender'], as_index = False)['Amount'].sum().sort_values(by= 'Amo
```

Out[47]:

	Gender	Amount
0	F	74335853
1	M	31913276

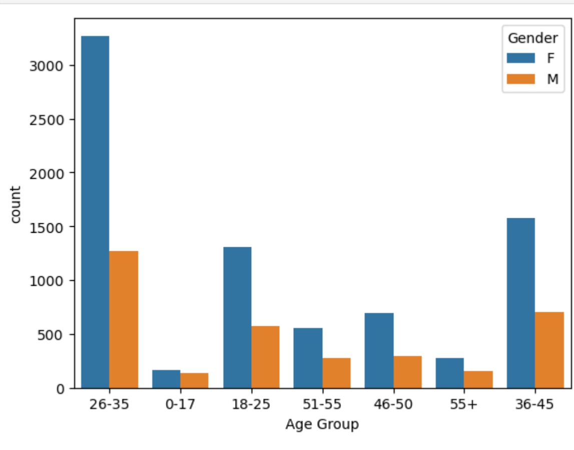
In [48]:
```python
sale_gender = sale_data.groupby(['Gender'], as_index = False)['Amount'].sum().sort_v
```

From the above plots, we can conclude that the number of female buyers is greater than male

Age

In [56]:
```python
age_data = sms.countplot(x= 'Age Group' , data = sale_data , hue= 'Gender')


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



In [58]:
```python
sale_data.groupby(['Age Group'], as_index = False)['Amount'].sum().sort_values(by=
```

Out[58]:

	Age Group	Amount
2	26-35	42613442
3	36-45	22144994
1	18-25	17240732
4	46-50	9207844
5	51-55	8261477
6	55+	4080987
0	0-17	2699653

In [59]:
```python
sale_age = sale_data.groupby(['Age Group'], as_index = False)['Amount'].sum().sort_v
```

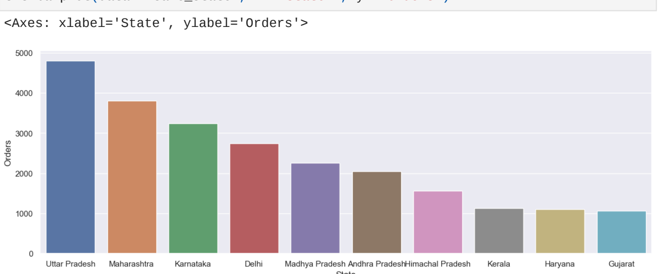
From the above plot, we can conclude that the maximum number of buyers belong to the age group between 26-35 female.

State

In [62]:
```python
#Orders from top 5 states
sale_state = sale_data.groupby(['State'], as_index = False)['Orders'].sum().sort_va

sms.set(rc={'figure.figsize' : (15,5)})
sms.barplot(data = sale_state , x= 'State' , y= 'Orders')
```

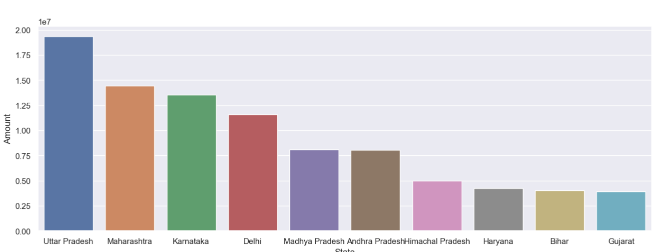
Out[62]:
```
<Axes: xlabel='State', ylabel='Orders'>
```



In [63]:
```python
#Total Amount from top 5 states
sale_state = sale_data.groupby(['State'], as_index = False)['Amount'].sum().sort_va

sms.set(rc={'figure.figsize' : (15,5)})
sms.barplot(data = sale_state , x= 'State' , y= 'Amount')
```

Out[63]:
```
<Axes: xlabel='State', ylabel='Amount'>
```



From the above plot, we can say that the maximum sale has been done in Uttar Pradesh. Also, we can notice that karela has made more orders compare to haryana and gujrat but amount spent is less.

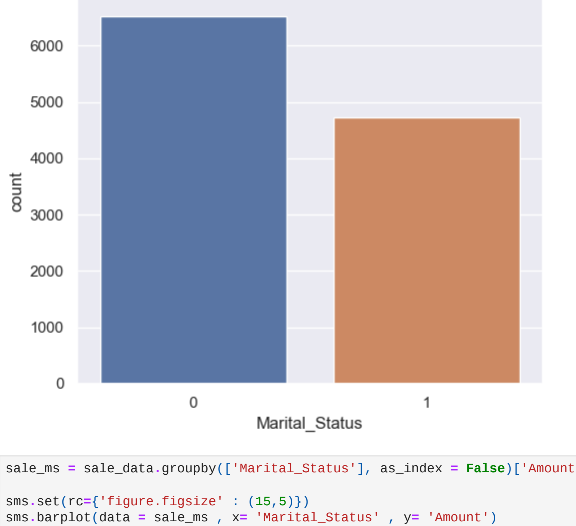
Marital Status

In [ ]:

In [71]:
```python
ms = sms.countplot(x= 'Marital_Status' , data = sale_data)

sms.set(rc={'figure.figsize' : (6,5)})

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



In [74]:
```python
sale_ms = sale_data.groupby(['Marital_Status'], as_index = False)['Amount'].sum().so

sms.set(rc={'figure.figsize' : (15,5)})
sms.barplot(data = sale_ms , x= 'Marital_Status' , y= 'Amount')
```

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

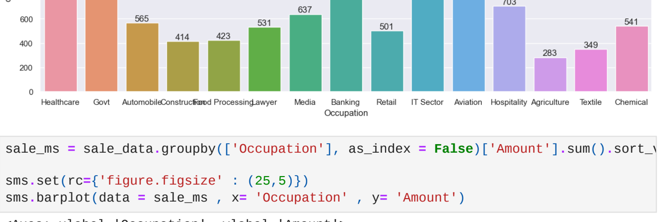
Married People has the highest purchasing power

Occupation

In [75]:
```python
ms = sms.countplot(x= 'Occupation' , data = sale_data)

sms.set(rc={'figure.figsize' : (25,5)})

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



In [76]:
```python
sale_ms = sale_data.groupby(['Occupation'], as_index = False)['Amount'].sum().sort_v

sms.set(rc={'figure.figsize' : (25,5)})
sms.barplot(data = sale_ms , x= 'Occupation' , y= 'Amount')
```

Out[76]:
```
<Axes: xlabel='Occupation', ylabel='Amount'>
```

From the above graph we can notice that the most of the buyers are working in IT, Healthcare and Aviation.

Product Category

In [77]:
```python
ms = sms.countplot(x= 'Product_Category' , data = sale_data)

sms.set(rc={'figure.figsize' : (25,5)})

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

In [78]:
```python
sale_ms = sale_data.groupby(['Product_Category'], as_index = False)['Amount'].sum()

sms.set(rc={'figure.figsize' : (25,5)})
sms.barplot(data = sale_ms , x= 'Product_Category' , y= 'Amount')
```

Out[78]:
```
<Axes: xlabel='Product_Category', ylabel='Amount'>
```

From the above plot we can conclude that the maximum sold product is Food

Conclusion

" It is possible that married women aged 26-35 from UP, Maharashtra, and Karnataka working in IT, Healthcare, and Aviation may prioritize purchasing food products followed by clothing."

In [ ]: