In [2]:
```
import pandas as pd
```

In [3]:
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [6]:
```
df = pd.read_csv('Diwali Sales Data.csv', encoding= 'unicode_escape')
```

In [6]:
```
df.head()
```

Out[6]:

	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	Sc
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	'
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Sc
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	W

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

Out[7]:

	User_ID	Age	Marital_Status	Orders	Amount	Status	unna
count	1.125100e+04	11251.000000	11251.000000	11251.000000	11239.000000	0.0	
mean	1.003004e+06	35.421207	0.420318	2.489290	9453.610858	NaN	
std	1.716125e+03	12.754122	0.493632	1.115047	5222.355869	NaN	
min	1.000001e+06	12.000000	0.000000	1.000000	188.000000	NaN	
25%	1.001492e+06	27.000000	0.000000	1.500000	5443.000000	NaN	
50%	1.003065e+06	33.000000	0.000000	2.000000	8109.000000	NaN	
75%	1.004430e+06	43.000000	1.000000	3.000000	12675.000000	NaN	
max	1.006040e+06	92.000000	1.000000	4.000000	23952.000000	NaN	

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

Out[9]:
```
(11251, 15)
```

```
In [11]: df.head(20)
```

Out[11]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat
5	1000588	Joni	P00057942	M	26-35	28	1	Himachal Pradesh
6	1001132	Balk	P00018042	F	18-25	25	1	Uttar Pradesh
7	1002092	Shivangi	P00273442	F	55+	61	0	Maharashtra
8	1003224	Kushal	P00205642	M	26-35	35	0	Uttar Pradesh
9	1003650	Ginny	P00031142	F	26-35	26	1	Andhra Pradesh
10	1003829	Harshita	P00200842	M	26-35	34	0	Delhi
11	1000214	Kargatis	P00119142	F	18-25	20	0	Andhra Pradesh
12	1004035	Elijah	P00080342	F	18-25	20	1	Andhra Pradesh
13	1001680	Vasudev	P00324942	M	26-35	26	1	Andhra Pradesh
14	1003858	Cano	P00293742	M	46-50	46	1	Madhya Pradesh
15	1000813	Lauren	P00289942	F	18-25	24	0	Andhra Pradesh
16	1005447	Amy	P00275642	F	46-50	48	1	Andhra Pradesh
17	1001193	Mick	P00004842	F	26-35	29	0	Andhra Pradesh
18	1001883	Praneet	P00029842	M	51-55	54	1	Uttar Pradesh
19	1001883	Praneet	P00029842	M	51-55	54	1	Uttar Pradesh

In [12]:
```
df.info()
#to check the datatypes of the columns
```

```
<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 [14]:
```
#Data cleaning
#to drop a column that has missing, blank or irrelavant values
df.drop(['Status', 'unnamed1'], axis = 1, inplace = True)
```

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

Out[15]:

	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	Sc
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	'
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Sc
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	W

In [17]:
```
pd.isnull(df).sum()
#To check null values
```

Out[17]:
```
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 [18]:
```
df.dropna(inplace=True)
#to drop null values
```

In [19]:
```
pd.isnull(df).sum()
```

Out[19]:
```
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              0
dtype: int64
```

In [20]:

```
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  float64
dtypes: float64(1), int64(4), object(8)
memory usage: 1.2+ MB
```

In [24]:

```
df['Amount'] = df['Amount'].astype('int')
#to change the datatype of a column
```

In [25]:

```
df['Amount'].dtypes
```

Out[25]:

```
dtype('int32')
```

In [5]:

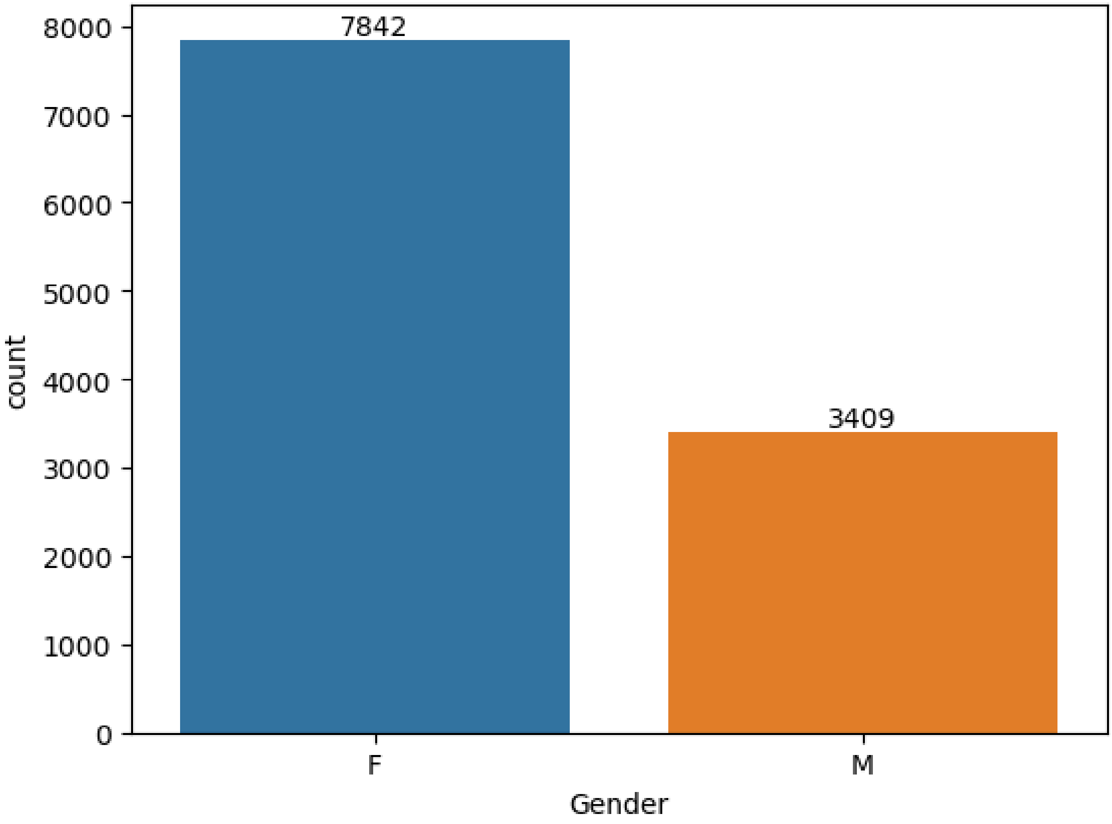
```
df.columns
```

Out[5]:

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

In [7]:
```
ax = sns.countplot(x = 'Gender', data =df)

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



In [5]:
```
#To check sales per gender
sales_gen = df.groupby(['Gender'], as_index = False)["Amount"].sum().sort_v
```

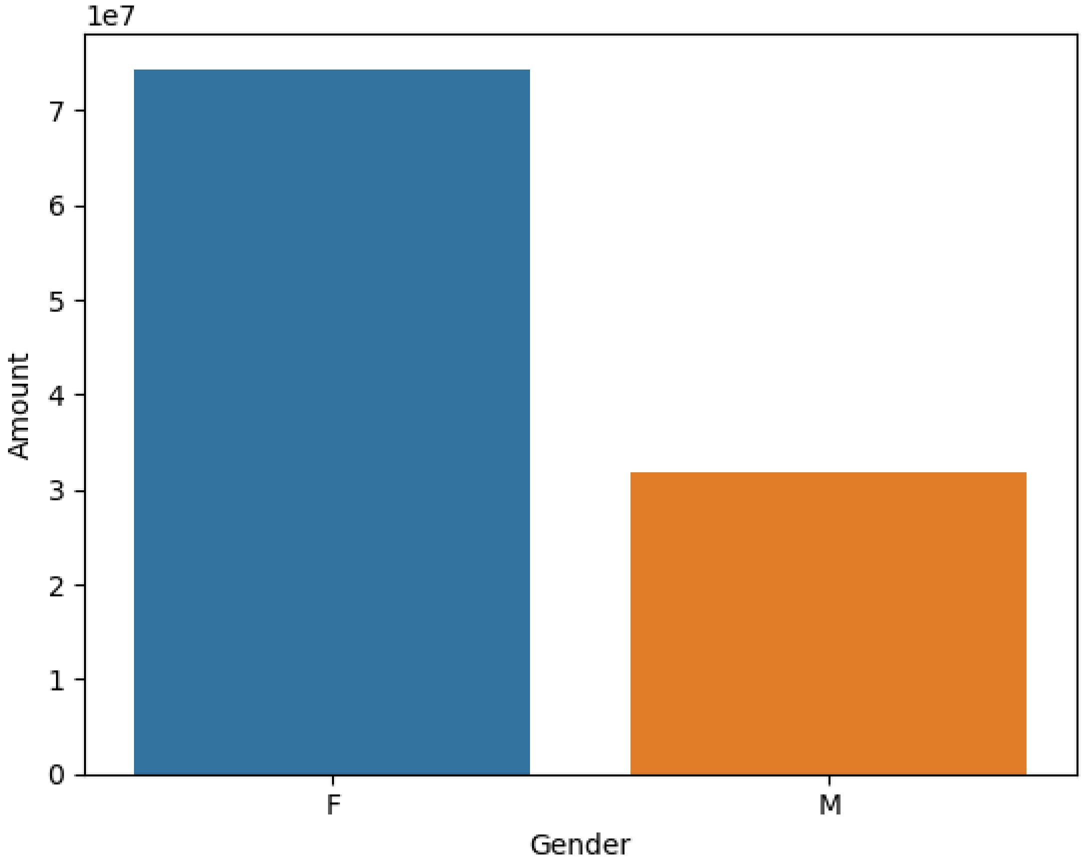
In [20]:
```
sales_gen
```

Out[20]:

	Gender	Amount
0	F	74335856.43
1	M	31913276.00

In [6]:
```
sns.barplot(x = 'Gender', y = 'Amount', data = sales_gen)
```

Out[6]:
```
<AxesSubplot:xlabel='Gender', ylabel='Amount'>
```



In [8]:
```
##from above graphs, most of the buyers are Females and even the purchasing
```

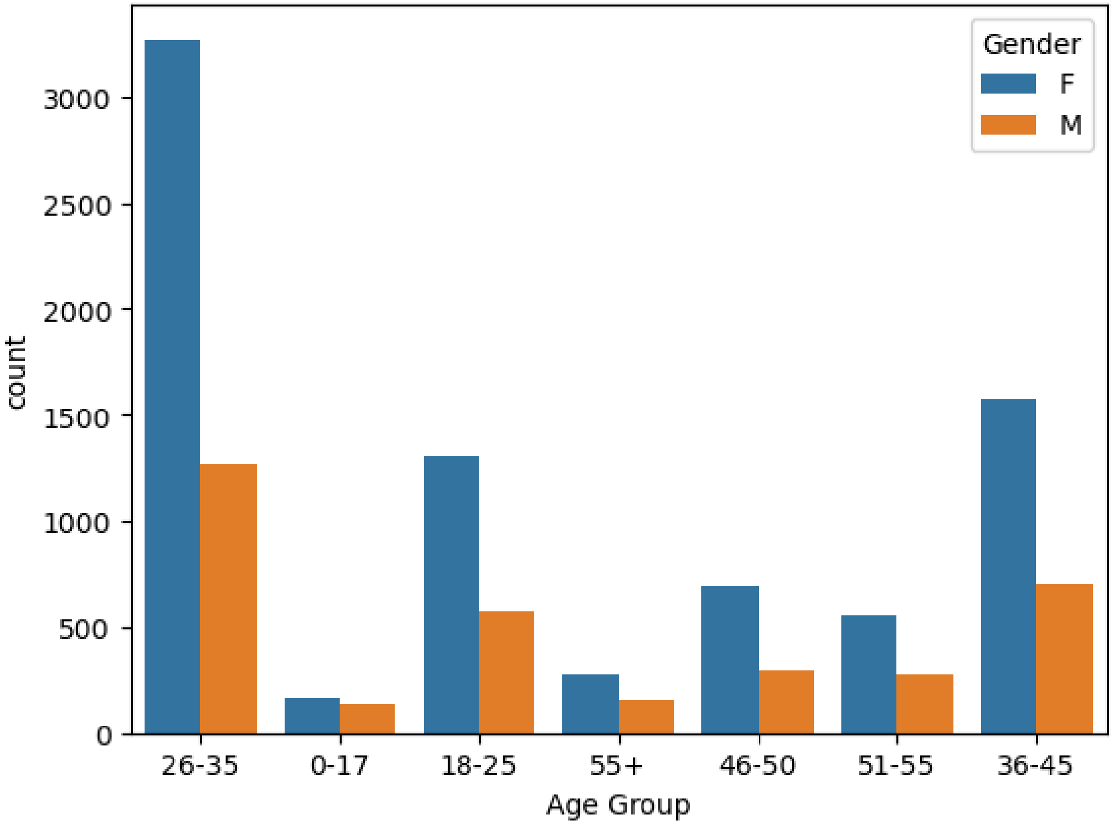
In [9]:
```
#AGE
```

In [10]:
```
df.columns
```

Out[10]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Categor
y',
       'Orders', 'Amount', 'Status', 'unnamed1'],
      dtype='object')
```

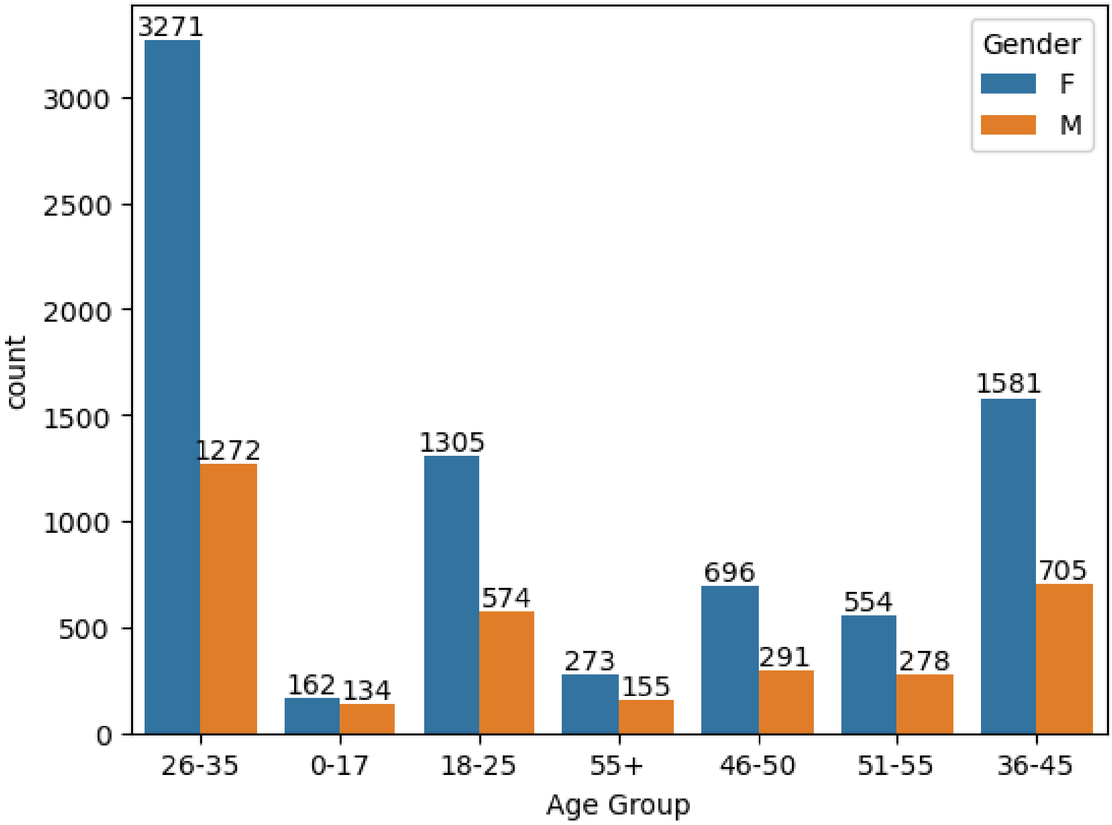
```
In [11]: sns.countplot(data = df, x = 'Age Group', hue = 'Gender')
```

```
Out[11]: <AxesSubplot:xlabel='Age Group', ylabel='count'>
```



```
In [12]: #from above graphs, we can determing that women of the age group 26-35
```

In [14]:
```
ax = sns.countplot(data = df, x = 'Age Group', hue ='Gender')
for bar in ax.containers:
    ax.bar_label(bar)
```



In [15]:
```
#Sales Amount as per age group
sales_age = df.groupby(['Age Group'], as_index=False)['Amount'].sum().sort_
```

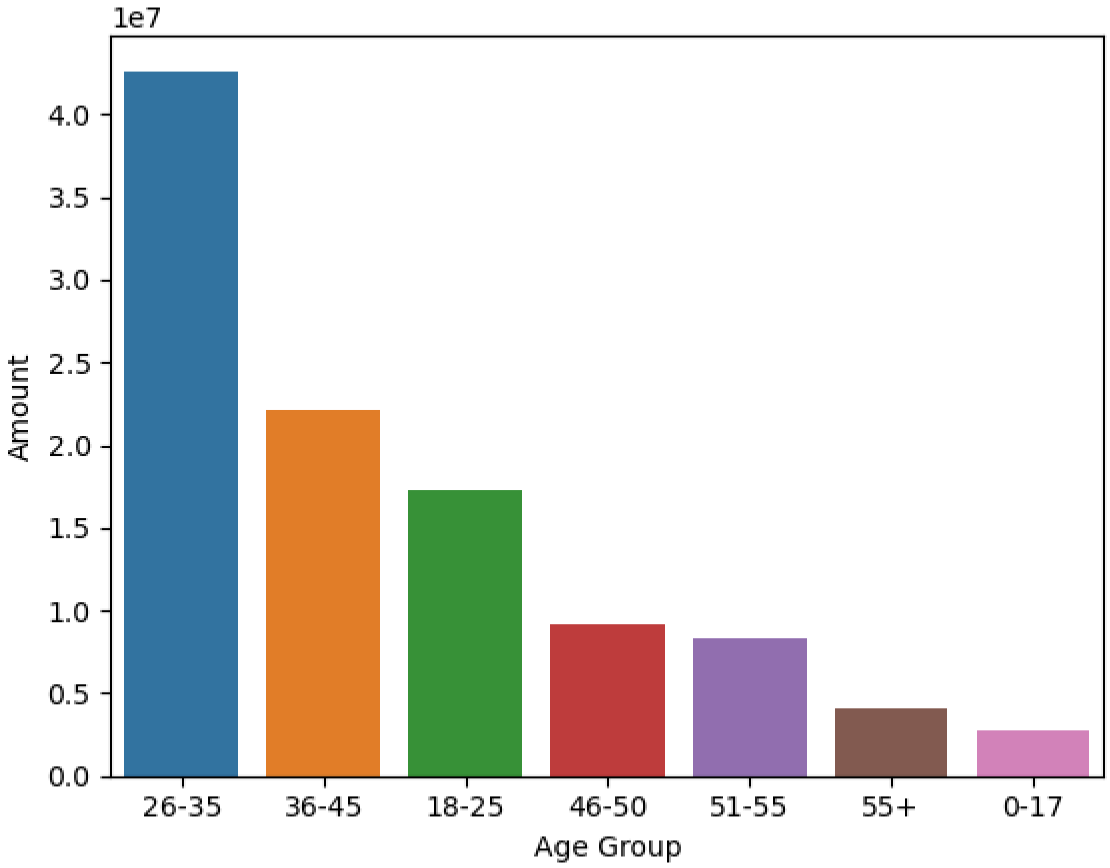
In [16]:
```
sales_age
```

Out[16]:

	Age Group	Amount
2	26-35	42613443.94
3	36-45	22144995.49
1	18-25	17240732.00
4	46-50	9207844.00
5	51-55	8261477.00
6	55+	4080987.00
0	0-17	2699653.00

```
In [17]: sns.barplot(x= 'Age Group', y = 'Amount', data = sales_age)
```

```
Out[17]: <AxesSubplot:xlabel='Age Group', ylabel='Amount'>
```

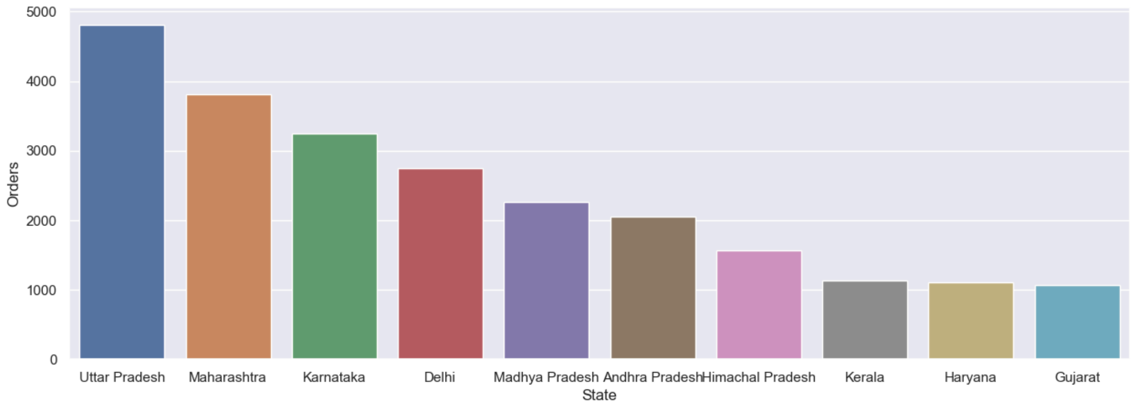


```
In [21]: #State
         #Total numbers of orders from top 10 states

         sales_state = df.groupby(['State'], as_index=False)['Orders'].sum().sort_va
         sns.set(rc={'figure.figsize':(15,5)})
```

```
In [24]: sns.barplot(data = sales_state, x = 'State', y = 'Orders')
```

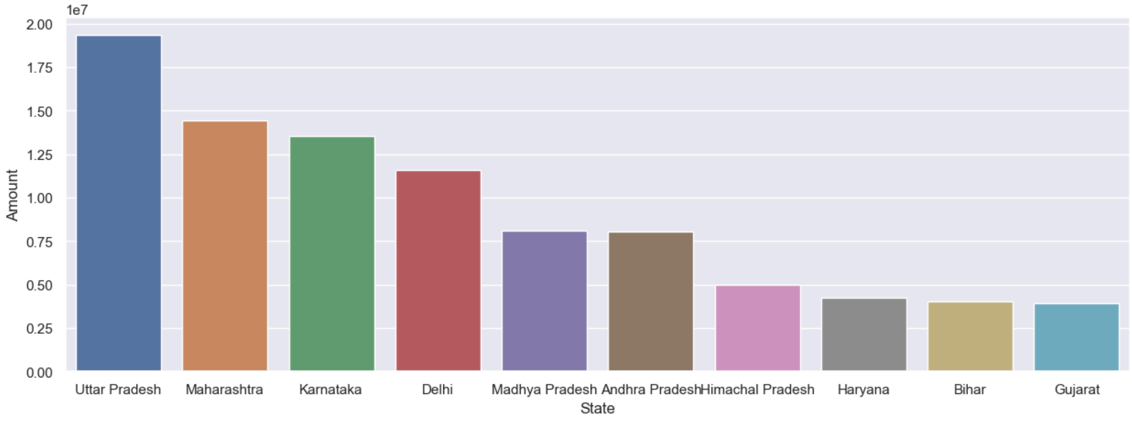
```
Out[24]: <AxesSubplot:xlabel='State', ylabel='Orders'>
```



```
In [ ]: #Total amount/sales from 10 states
```

In [27]:
```
sales_state = df.groupby(['State'], as_index= False)['Amount'].sum().sort_v
sns.set(rc={'figure.figsize': (15,5)})
sns.barplot(data = sales_state, x= 'State', y = 'Amount')
```

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



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

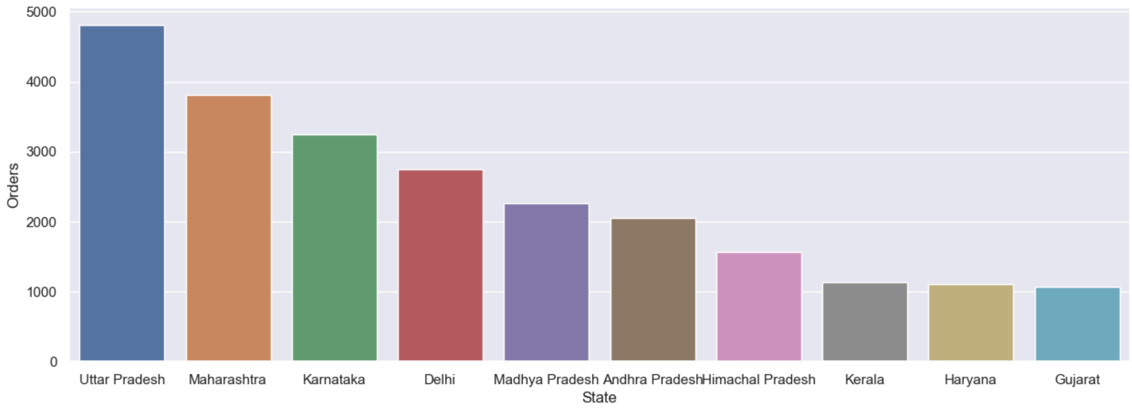
Out[12]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Categor
y',
       'Orders', 'Amount', 'Status', 'unnamed1'],
      dtype='object')
```

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

Out[14]:
```
9453.610857727557
```

In [4]:
```
sales_state = df.groupby(['State'], as_index= False)['Orders'].sum().sort_v
sns.set(rc={'figure.figsize': (15,5)})
sns.barplot(data = sales_state, x= 'State', y = 'Orders')
```

Out[4]:
```
<AxesSubplot:xlabel='State', ylabel='Orders'>
```

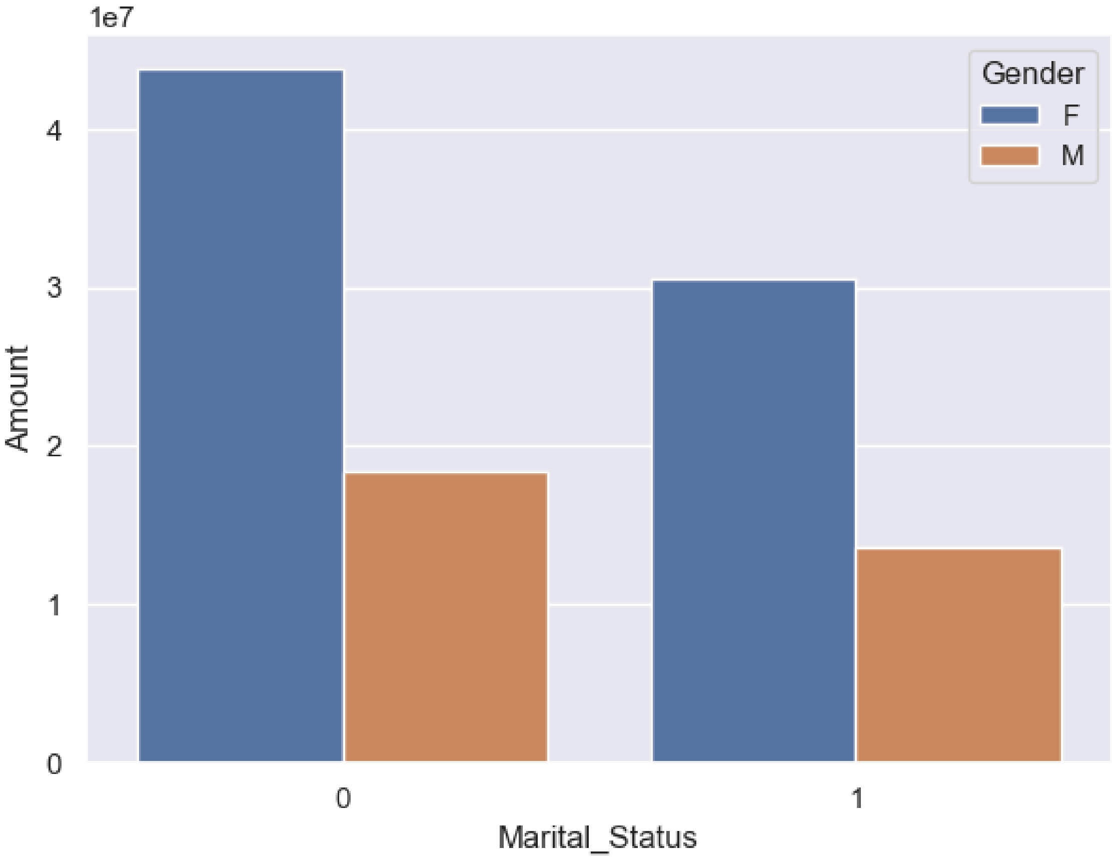


In [6]:
```
##From above graph, we can see most number of the orders are from Uttar Pra
```

In [7]:
```
##Martial Status
```

In [10]:
```
sales_state = df.groupby(['Marital_Status', 'Gender'], as_index= False)['Am
sns.set(rc={'figure.figsize': (7,5)})
sns.barplot(data = sales_state, x= 'Marital_Status', y = 'Amount', hue = 'G
```

Out[10]:
```
<AxesSubplot:xlabel='Marital_Status', ylabel='Amount'>
```



In [11]:
```
sales_state
```

Out[11]:

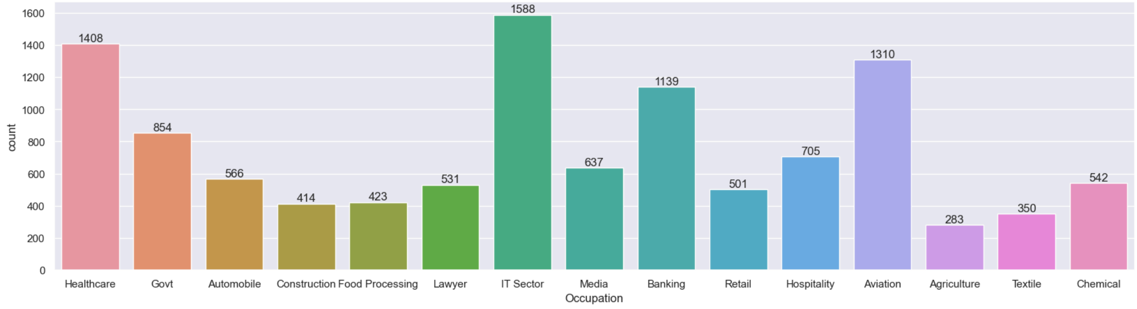
	Marital_Status	Gender	Amount
0	0	F	43786648.44
2	1	F	30549207.99
1	0	M	18338738.00
3	1	M	13574538.00

In [12]:
```
#Occupation
```

In [16]:

```
sns.set(rc={'figure.figsize': (20,5)})
ax = sns.countplot(data =df, x ='Occupation')

#to get count in numbers we need to write following code
for bars in ax.containers:
    ax.bar_label(bars)
```



In [ ]:

```
#From above graph, we can see the highest number of buyers are from IT, hea
```