

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

```
In [4]: data = pd.read_csv("D:\DATA ANALYST AND DATA SCIENCE\PYTHON\pandas project\Python_Diwali_Sales_Analysis-main\Python_Diwali_Sales_Analysis-main\Diwali Sales Data.csv",
encoding = 'unicode_escape')
data
```

```
Out[4]:
```

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount	Status	unnamed1
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952.0	NaN	NaN
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934.0	NaN	NaN
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.0	NaN	NaN
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.0	NaN	NaN
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877.0	NaN	NaN
...
11246	1000695	Manning	P00296942	M	18-25	19	1	Maharashtra	Western	Chemical	Office	4	370.0	NaN	NaN
11247	1004089	Reichenbach	P00171342	M	26-35	33	0	Haryana	Northern	Healthcare	Veterinary	3	367.0	NaN	NaN
11248	1001209	Oshin	P00201342	F	36-45	40	0	Madhya Pradesh	Central	Textile	Office	4	213.0	NaN	NaN
11249	1004023	Noonan	P00059442	M	36-45	37	0	Karnataka	Southern	Agriculture	Office	3	206.0	NaN	NaN
11250	1002744	Brumley	P00281742	F	18-25	19	0	Maharashtra	Western	Healthcare	Office	3	188.0	NaN	NaN

11251 rows × 15 columns

```
In [5]: data.shape
```

```
Out[5]: (11251, 15)
```



NoteGPT

In [6]: `data.head(10)`

Out[6]:	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount	Status	unnamed1
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952.00	NaN	NaN
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934.00	NaN	NaN
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.00	NaN	NaN
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.00	NaN	NaN
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877.00	NaN	NaN
5	1000588	Joni	P00057942	M	26-35	28	1	Himachal Pradesh	Northern	Food Processing	Auto	1	23877.00	NaN	NaN
6	1001132	Balk	P00018042	F	18-25	25	1	Uttar Pradesh	Central	Lawyer	Auto	4	23841.00	NaN	NaN
7	1002092	Shivangi	P00273442	F	55+	61	0	Maharashtra	Western	IT Sector	Auto	1	NaN	NaN	NaN
8	1003224	Kushal	P00205642	M	26-35	35	0	Uttar Pradesh	Central	Govt	Auto	2	23809.00	NaN	NaN
9	1003650	Ginny	P00031142	F	26-35	26	1	Andhra Pradesh	Southern	Media	Auto	4	23799.99	NaN	NaN



In [7]: `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 [8]: `data.drop(["Status", "unnamed1"], axis=1, inplace=True)`



```
In [8]: data.drop(["Status", "unnamed1"], axis=1, inplace=True)
```

```
In [9]: 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 [10]: pd.isnull(data).sum()
```

```
Out[10]: 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 [11]: data.dropna(inplace=True)
```

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

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

```
In [13]: data["Amount"] = data["Amount"].astype('int')
```

```
In [14]: data["Amount"].dtypes
```

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

```
In [15]: data.columns
```

```
Out[15]: Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',  
              'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',  
              'Orders', 'Amount'],  
              dtype='object')
```



NoteGPT

In [16]: `data.describe()`

Out[16]:

	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 [17]: `data[["Age", "Orders", "Amount"]].describe()`

Out[17]:

	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

EXPLORATORY DATA ANALYSIS

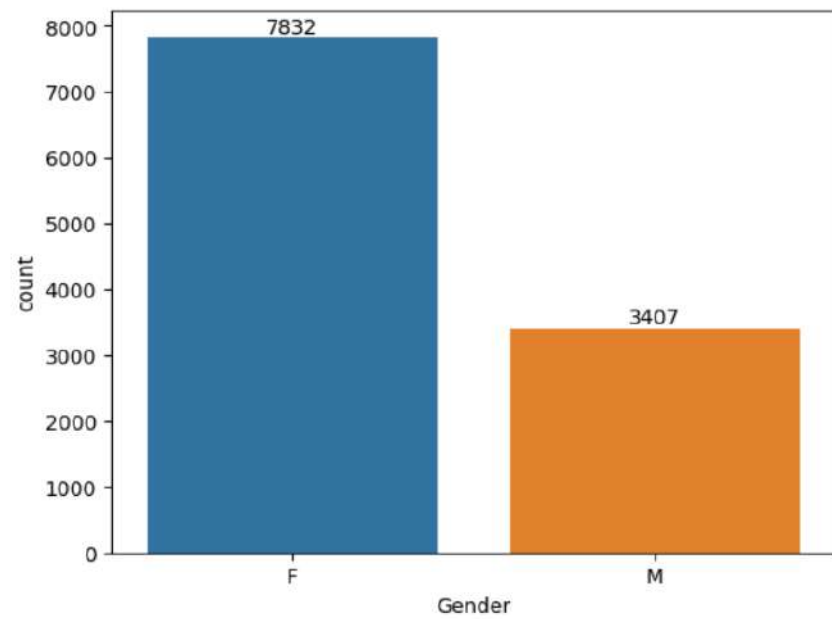
GENDER

```
In [19]: data.columns
```

```
Out[19]: Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',  
              'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',  
              'Orders', 'Amount'],  
              dtype='object')
```

```
In [24]: ax = sns.countplot(x="Gender", data=data)
```

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



NoteGPT

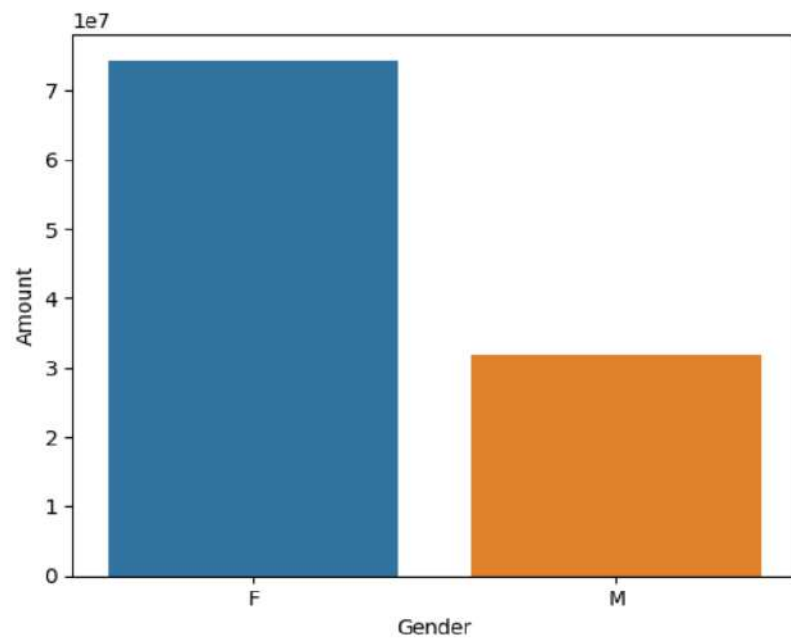

```
In [26]: data.groupby(["Gender"], as_index=False)["Amount"].sum().sort_values(by="Amount", ascending=False)
```

```
Out[26]:
```

	Gender	Amount
0	F	74335853
1	M	31913276

```
In [27]: sales_gen= data.groupby(["Gender"], as_index=False)["Amount"].sum().sort_values(by="Amount", ascending=False)  
sns.barplot(x="Gender", y="Amount", data=sales_gen)
```

```
Out[27]: <Axes: xlabel='Gender', ylabel='Amount'>
```



NoteGPT

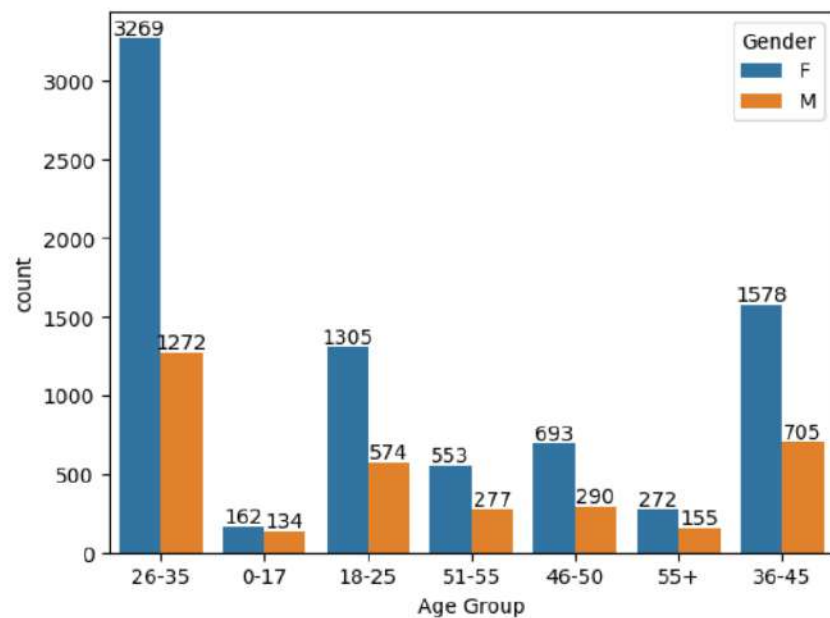
AGE

```
In [28]: data.columns
```

```
Out[28]: Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',  
            'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',  
            'Orders', 'Amount'],  
            dtype='object')
```

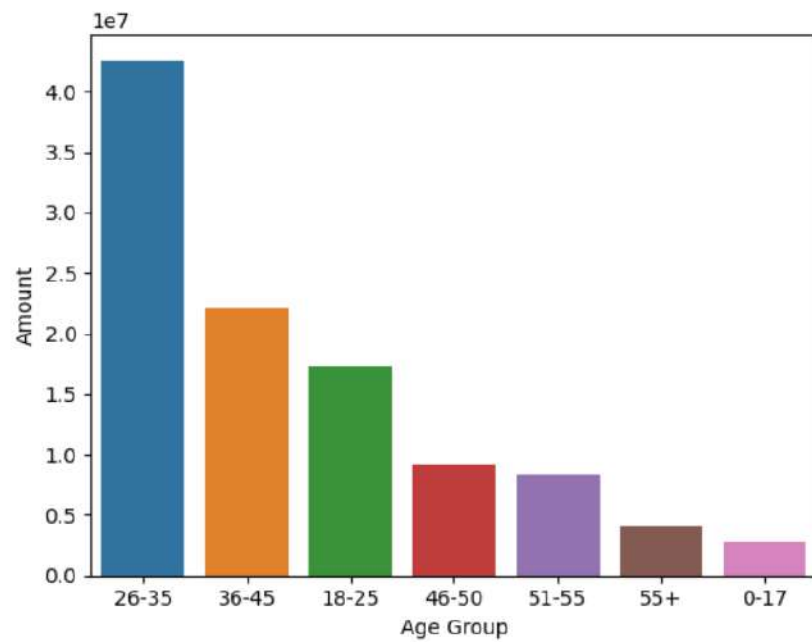
```
In [30]: ax = sns.countplot(x="Age Group", hue="Gender", data=data)
```

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



```
In [31]: sales_gen= data.groupby(["Age Group"], as_index=False)["Amount"].sum().sort_values(by="Amount", ascending=False)
sns.barplot(x="Age Group", y="Amount", data=sales_gen)
```

```
Out[31]: <Axes: xlabel='Age Group', ylabel='Amount'>
```



NoteGPT

STATE

In [33]: `data.columns`

Out[33]: `Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
 'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
 'Orders', 'Amount'],
 dtype='object')`

In [34]: `sales_gen= data.groupby(["State"], as_index=False)["Orders"].sum().sort_values(by="Orders", ascending=False).head(10)`

`sns.set(rc={"figure.figsize":(15,5)})`

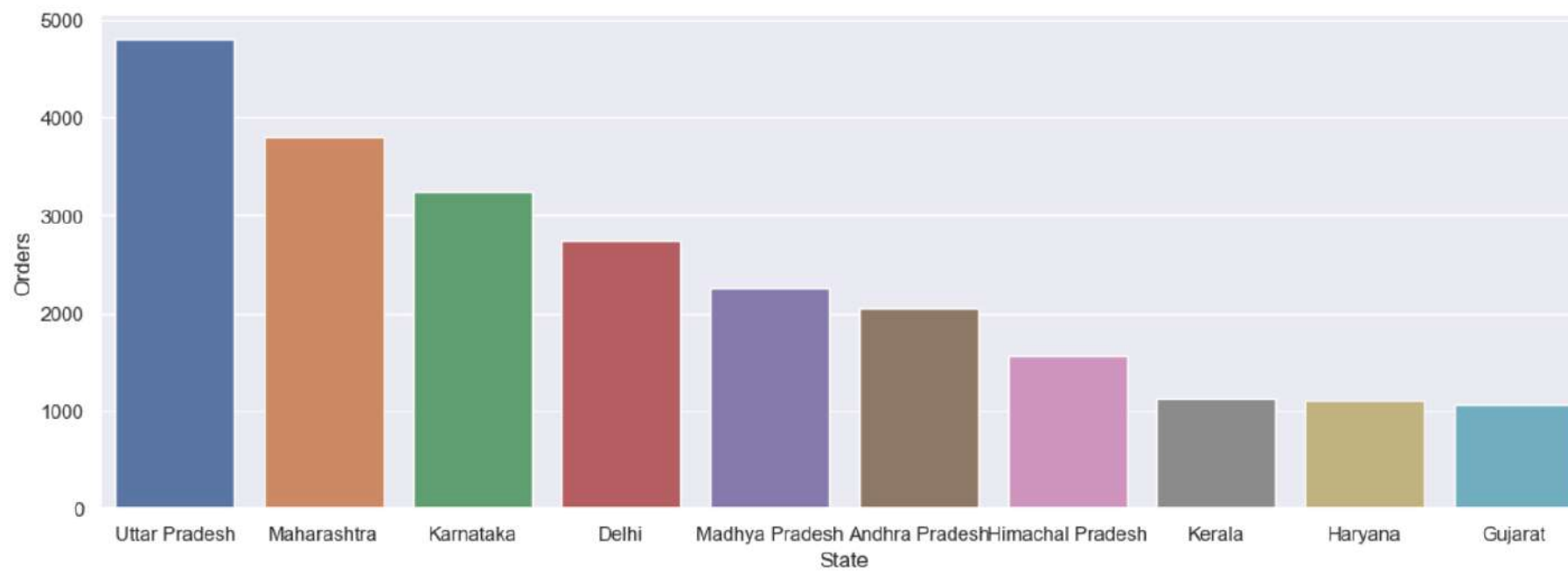
`sns.barplot(x="State", y="Orders", data=sales_gen)`

Out[34]: `<Axes: xlabel='State', ylabel='Orders'>`



NoteGPT

Out[34]: <Axes: xlabel='State', ylabel='Orders'>

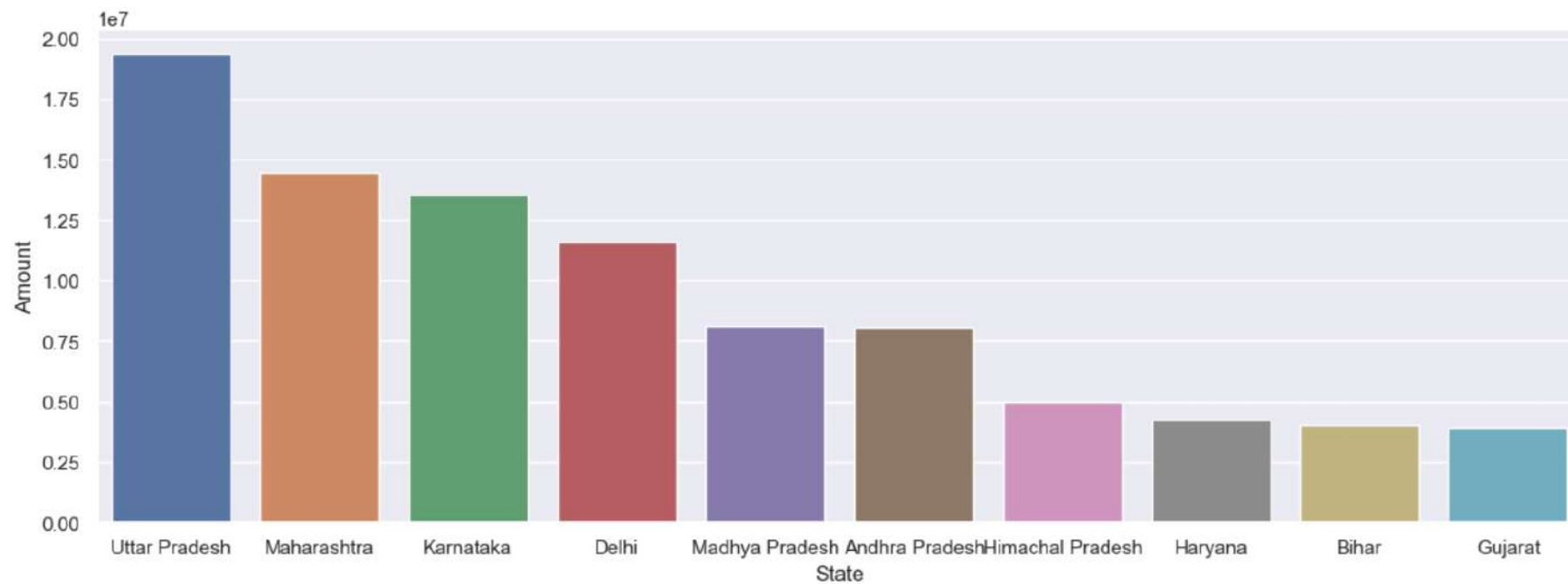


```
In [35]: sales_gen = data.groupby(["State"], as_index=False)["Amount"].sum().sort_values(by="Amount", ascending=False).head(10)

sns.set(rc={"figure.figsize":(15,5)})

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

```
Out[35]: <Axes: xlabel='State', ylabel='Amount'>
```



MARTIAL STATUS

In [36]: `data.columns`

Out[36]: `Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
 'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
 'Orders', 'Amount'],
 dtype='object')`

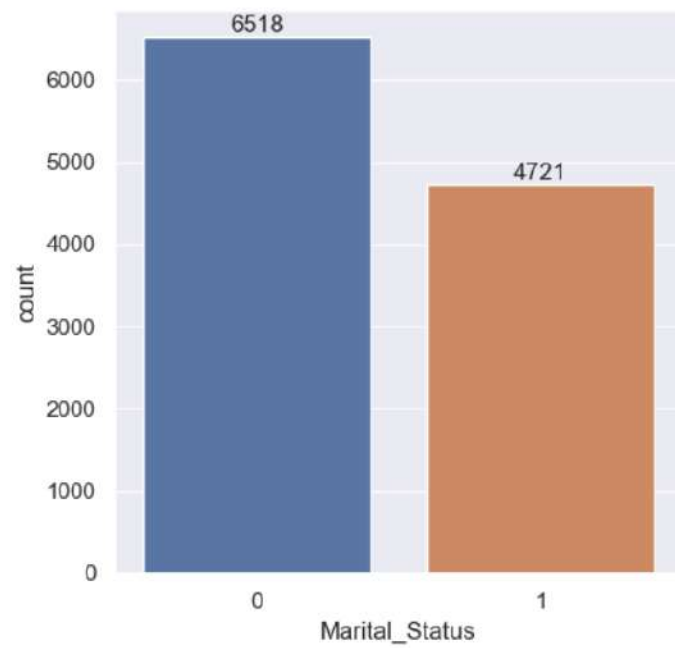
In [39]: `ax = sns.countplot(x= "Marital_Status", data=data)`

```
sns.set(rc={"figure.figsize":(5,5)})
```

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



NoteGPT

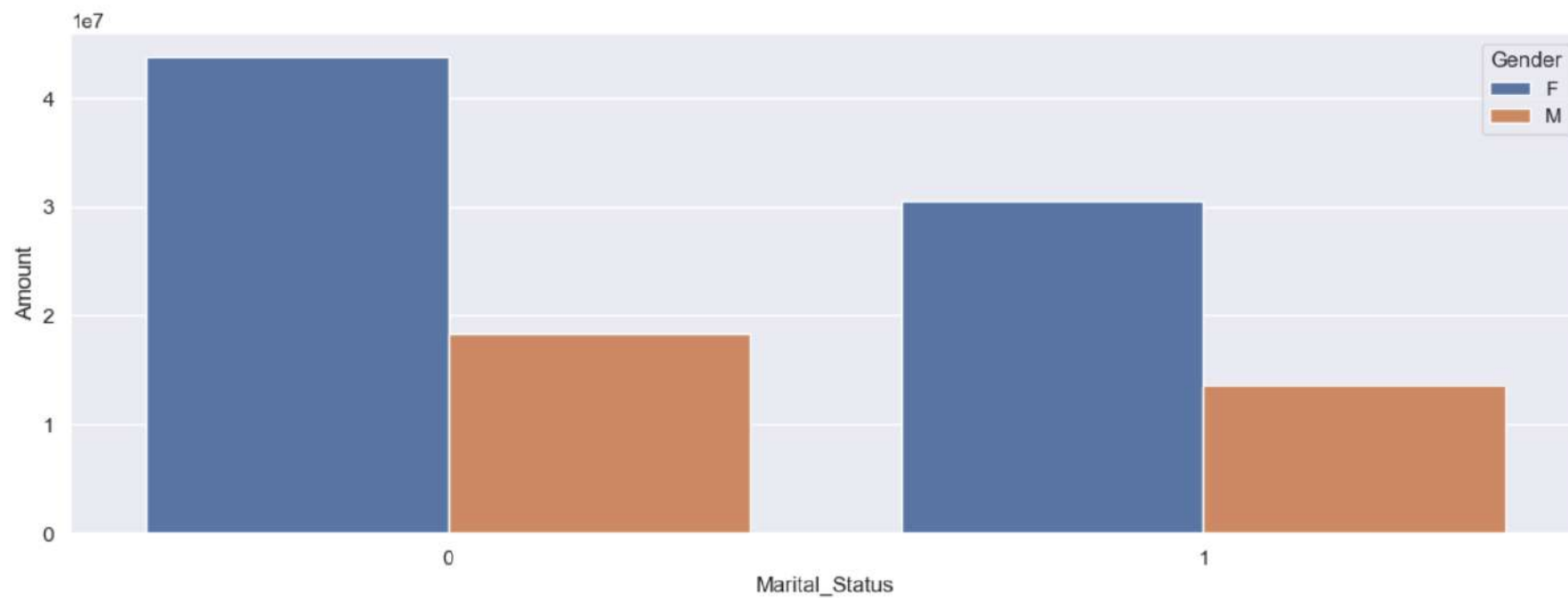



```
In [40]: sales_gen= data.groupby(["Marital_Status", "Gender"], as_index=False)["Amount"].sum().sort_values(by="Amount", ascending=False)

sns.set(rc={"figure.figsize":(15,5)})

sns.barplot(x="Marital_Status", y="Amount", data=sales_gen, hue="Gender")
```

Out[40]: <Axes: xlabel='Marital_Status', ylabel='Amount'>



OCCUPATION

In [41]: `data.columns`

Out[41]: `Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
 'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
 'Orders', 'Amount'],
 dtype='object')`

In [42]: `ax= sns.countplot(x= "Occupation", data=data)`

`sns.set(rc={"figure.figsize":(20,5)})`

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



NoteGPT