Import Libraries

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

In [8]:
```
cd J:\Python\Diwali Sales
```

```
J:\Python\Diwali Sales
```

Import CSV file

In [12]:
```
ds=pd.read_csv('Diwali Sales Data.csv',encoding='unicode_escape')
```

In [13]:
```
ds
```

Out[13]:

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

11251 rows × 15 columns

Quality Data Cheak

In [14]:
```
ds.shape
```

Out[14]:
```
(11251, 15)
```

In [15]:
```
ds.info() # change "age group" data type
```

```
<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 [16]:
```
ds.describe()            #no outlier
```

Out[16]:

	User_ID	Age	Marital_Status	Orders	Amount	Status	unnamed1
count	1.125100e+04	11251.000000	11251.000000	11251.000000	11239.000000	0.0	0.0
mean	1.003004e+06	35.421207	0.420318	2.489290	9453.610858	NaN	NaN
std	1.716125e+03	12.754122	0.493632	1.115047	5222.355869	NaN	NaN
min	1.000001e+06	12.000000	0.000000	1.000000	188.000000	NaN	NaN
25%	1.001492e+06	27.000000	0.000000	1.500000	5443.000000	NaN	NaN
50%	1.003065e+06	33.000000	0.000000	2.000000	8109.000000	NaN	NaN
75%	1.004430e+06	43.000000	1.000000	3.000000	12675.000000	NaN	NaN
max	1.006040e+06	92.000000	1.000000	4.000000	23952.000000	NaN	NaN

In [21]:
```
ds.isnull().sum()       # "Amount ,Status ,unnamed1" has null value
```

Out[21]:
```
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
```

Data Cleaning

In [26]:
```
# drop unrelvant columns

ds.drop(["Status" ,"unnamed1"] , axis=1, inplace =True)
```

In [27]:
```
ds
```

Out[27]:

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

11251 rows × 13 columns

In [29]:
```
# drop the nun value

ds.dropna(inplace=True)
```

In [30]:
```
pd.isnull(ds).sum()
```

Out[30]:
```
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 [23]:
```
pd.isnull(ds).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 [31]:
```
ds.shape
```

Out[31]:
```
(11239, 13)
```

In [46]:
```
# change datatype

ds['Amount']=ds['Amount'].astype(int)
```

In [48]:
```
ds['Amount'].dtype
```

Out[48]:
```
dtype('int32')
```

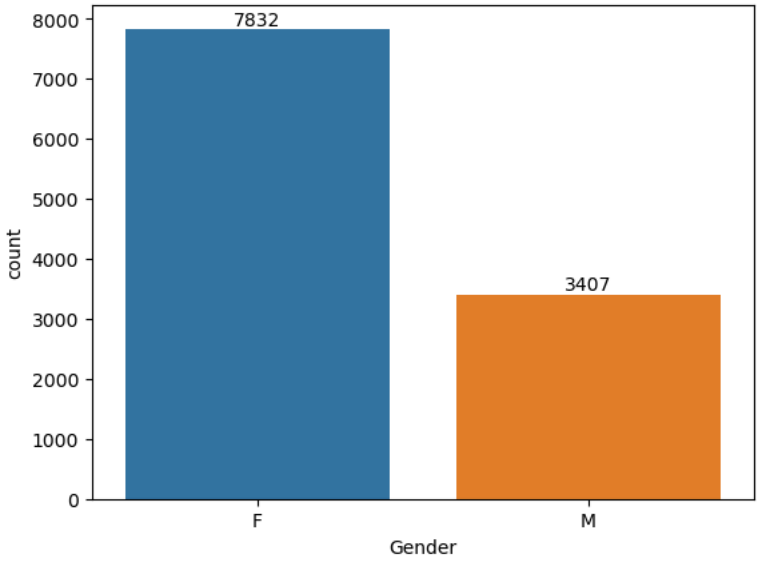
In [49]:
```
ds.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 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
```

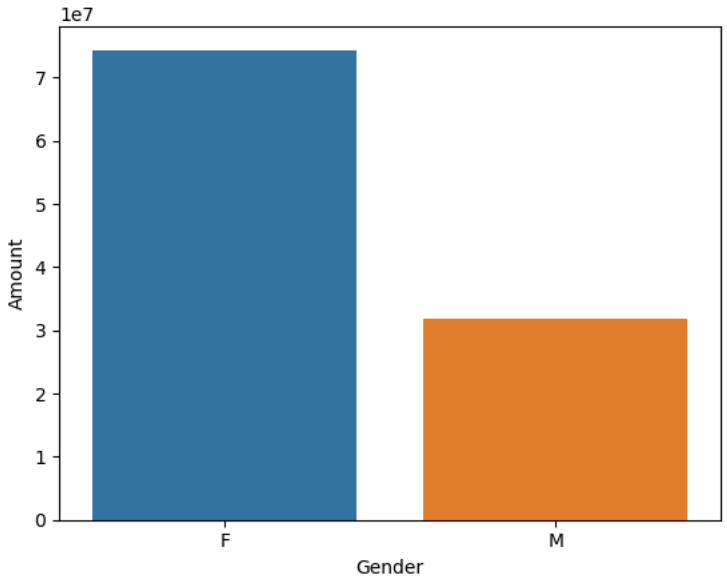
Exploratory Data Analysis

Gender

In [53]:
```
# plotting a bar chart for Gender and it's count

ax = sns.countplot(x = 'Gender',data = ds)

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



In [82]:
```
# plotting a bar chart for gender vs total amount

sales=ds.groupby('Gender', as_index=False )['Amount'].sum().sort_values(by='Amount',  ascending=False)

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




```
From above graphs we can see that most of the buyers are females and even the purchasing power of females are greater than men
```

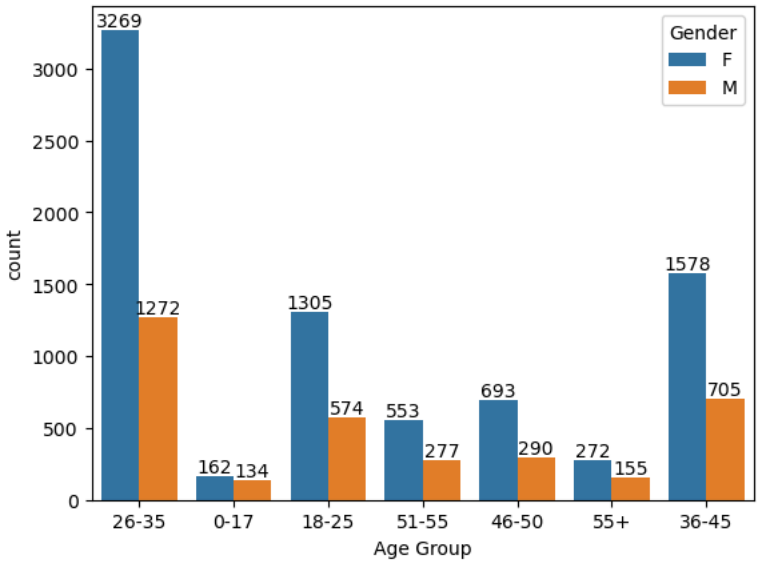
Age

In [97]:
```
# which age group are orders the most

ds.groupby('Age Group', as_index=False)['Orders'].count().sort_values(by='Orders' , ascending=False)


```

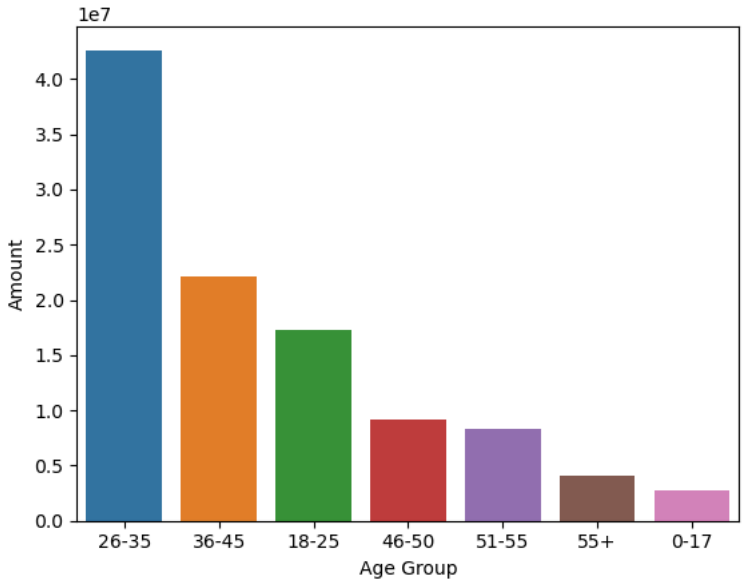
Out[97]:

	Age Group	Orders
2	26-35	4541
3	36-45	2283
1	18-25	1879
4	46-50	983
5	51-55	830
6	55+	427
0	0-17	296

In [105]:
```

ax=sns.countplot(x='Age Group' , data = ds , hue='Gender' );

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



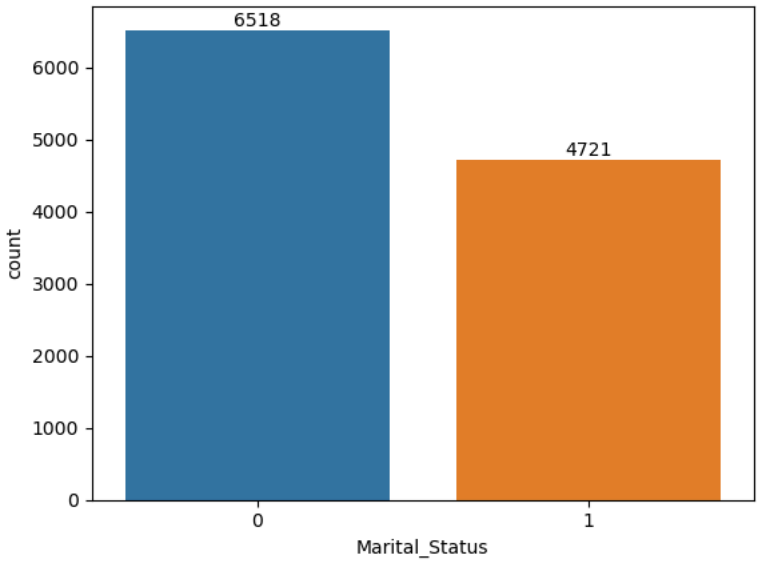
In [129]:
```
# Total amount vs Age Group

sales_age=ds.groupby('Age Group' , as_index=False)["Amount"].sum().sort_values(by='Amount',ascending=False)

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


plt.show()
```



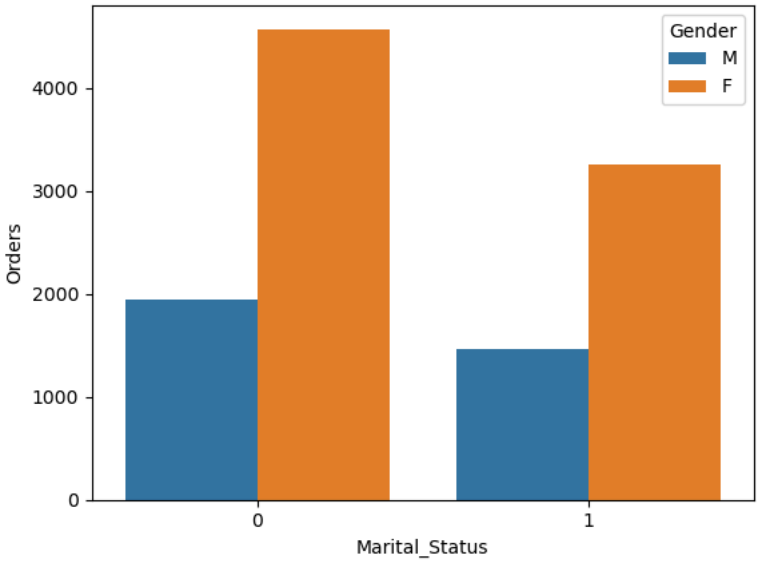
```
From above graphs we can see that most of the buyers are of age group between 26-35 yrs females
```

Marital_Status

In [157]:
```
ax=sns.countplot(x='Marital_Status' , data=ds);

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

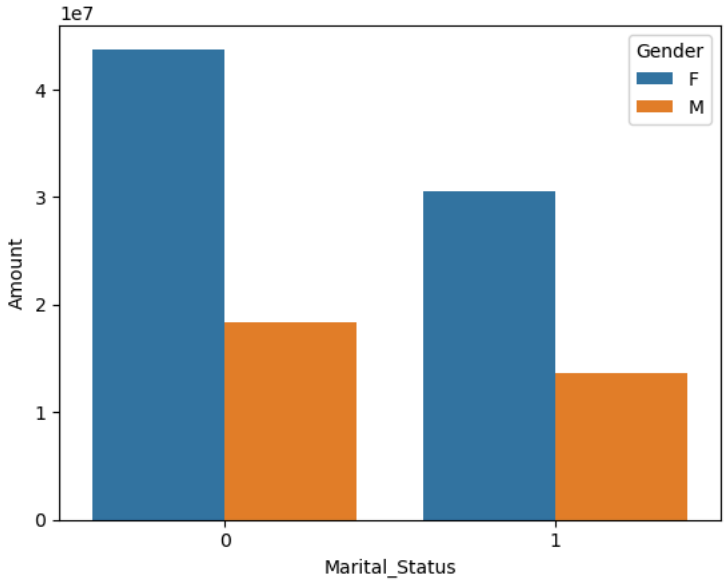


In [137]:
```
# total num of orders by marital status

order_marital=ds.groupby(['Marital_Status' , 'Gender'] , as_index=False)['Orders'].count().sort_values(by='Gender' , asc

sns.barplot(x='Marital_Status' , y='Orders' , data=order_marital , hue='Gender' );
```



In [144]:
```
# total num of amounts by marital status

sales_marital=ds.groupby(['Marital_Status' , 'Gender'],as_index=False)['Amount'].sum().sort_values(by='Amount' , ascendi

sns.barplot(x='Marital_Status' , y='Amount' , data=sales_marital , hue='Gender')
```

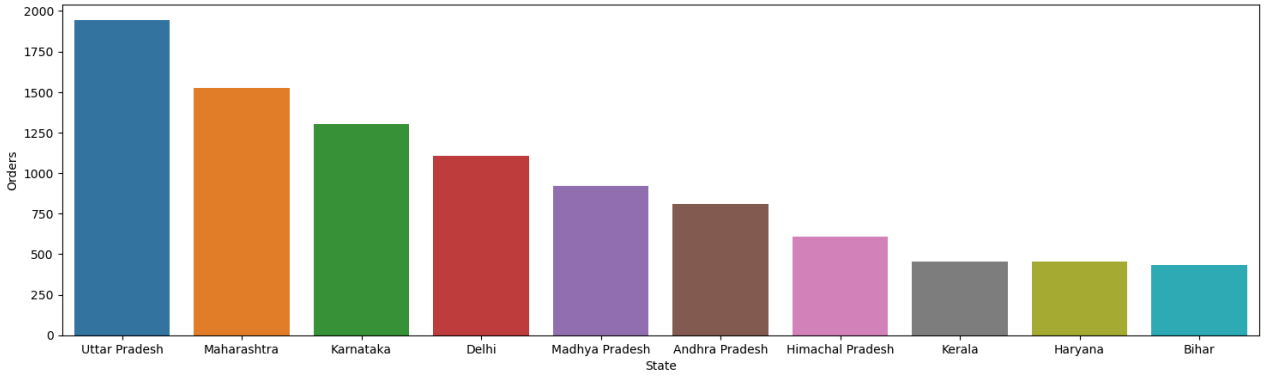
Out[144]: <Axes: xlabel='Marital_Status', ylabel='Amount'>



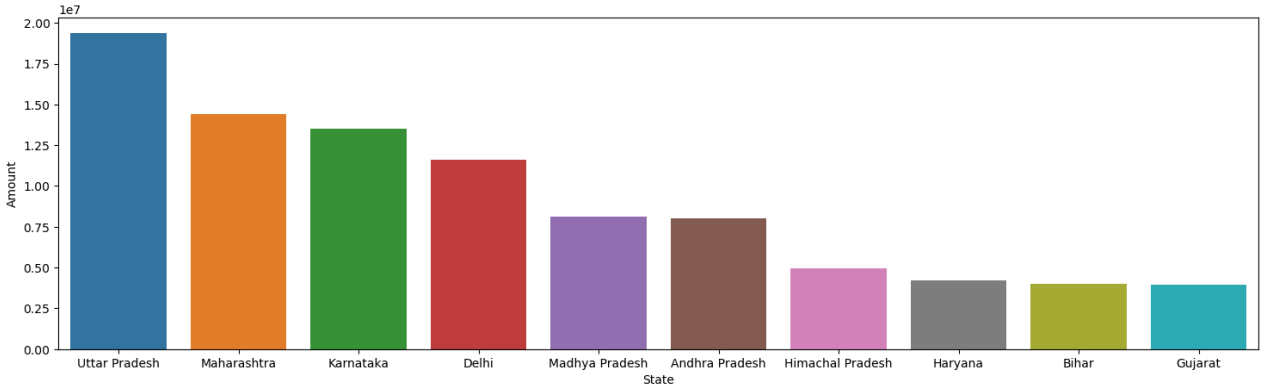

```
From above graphs we can see that most of the buyers are married (women) and they have high purchasing power
```

State

In [176]:
```
# total num of orders by top 10 state

plt.figure(figsize=(18,5))
state_ord=ds.groupby('State' , as_index=False)['Orders'].count().sort_values(by='Orders',ascending=False).head(10)

sns.barplot(x='State'  , y='Orders', data=state_ord );
```



In [179]:
```
# total numbers of amount by top 5 state
plt.figure(figsize=(18,5))
sales_sta=ds.groupby('State' , as_index=False)['Amount'].sum().sort_values(by="Amount",ascending =False).head(10)

sns.barplot(x='State' , y='Amount' , data=sales_sta);


```

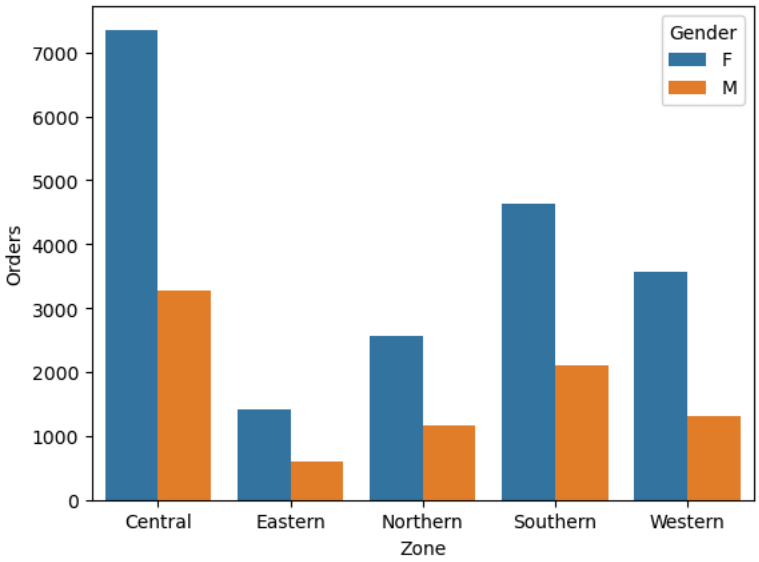



```
From above graphs we can see that most of the orders & total sales/amount are from Uttar Pradesh, Maharashtra and
Karnataka respectively
```

Zone

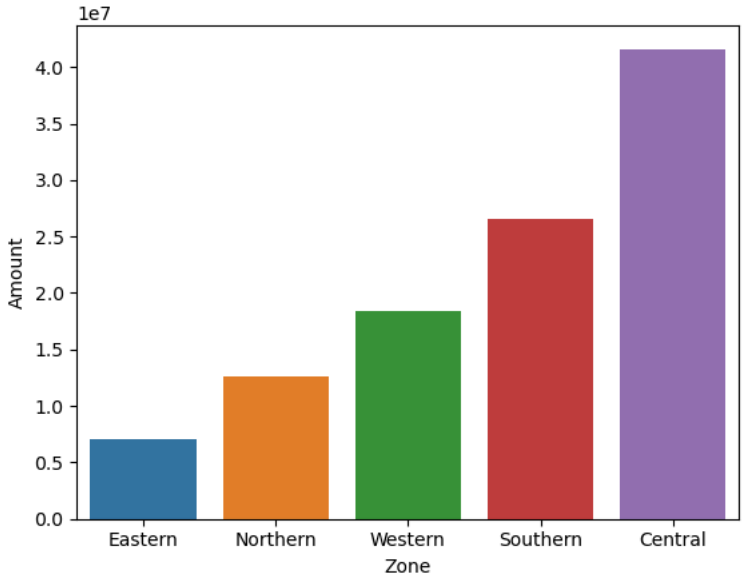
In [205]:
```
# num of total orders by zone with gender distribution

sns.barplot(x='Zone' , y='Orders',data=ds.groupby(['Zone','Gender'] , as_index=False)['Orders'].sum().sort_values(by='Zo
```

Out[205]: <Axes: xlabel='Zone', ylabel='Orders'>



In [208]:
```
# num of total Amount by zone with gender distribution

sns.barplot(x='Zone',y='Amount',data=ds.groupby('Zone' , as_index=False)['Amount'].sum().sort_values(by='Amount'))
```

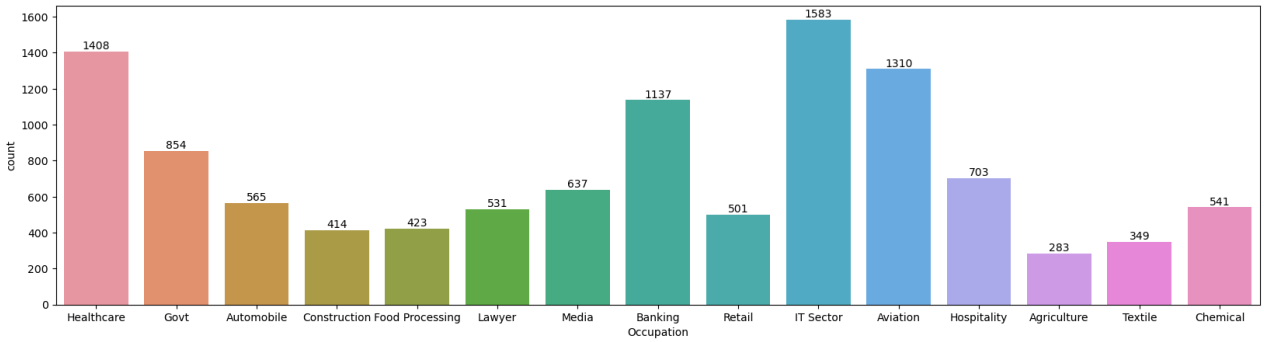
Out[208]: <Axes: xlabel='Zone', ylabel='Amount'>




```
from above the charts we can say that the central zone has females are most buying the products
```

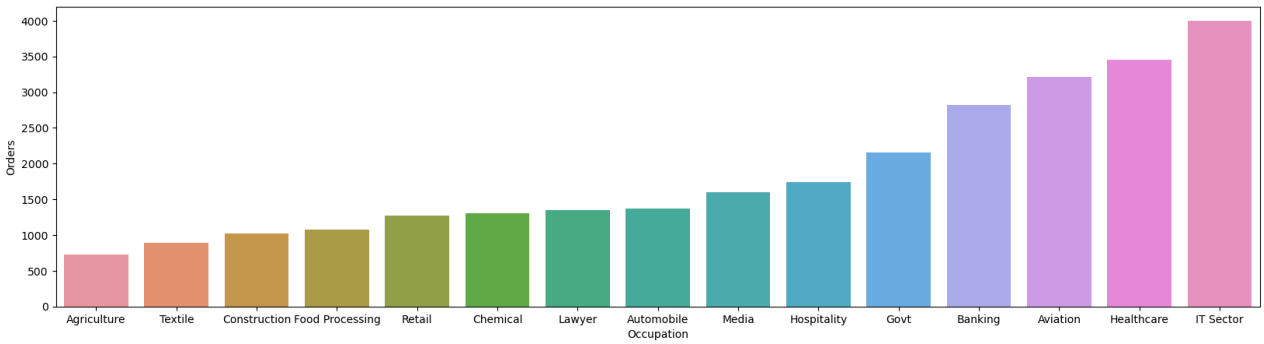
Occupation

In [215]:
```
plt.figure(figsize=(20,5))
ax=sns.countplot(x='Occupation',data=ds);

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



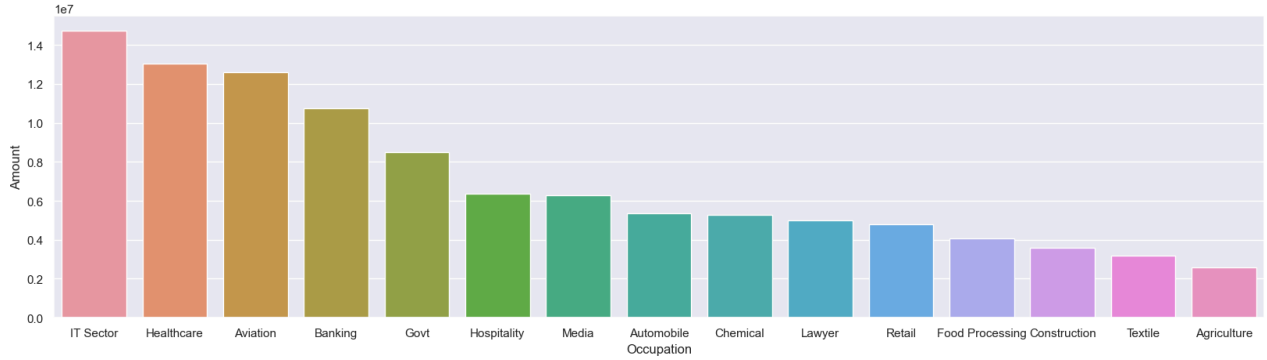
In [221]:
```
# occupation / orders

plt.figure(figsize=(20,5))
occ_ord=ds.groupby('Occupation',as_index=False)['Orders'].sum().sort_values(by='Orders')

sns.barplot(x='Occupation' , y='Orders' , data =occ_ord);
```

Out[221]: <Axes: xlabel='Occupation', ylabel='Orders'>



In [237]:
```
sales_state = ds.groupby(['Occupation'], as_index=False)['Amount'].sum().sort_values(by='Amount', ascending=False)

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Occupation',y= 'Amount')
```

Out[237]: <Axes: xlabel='Occupation', ylabel='Amount'>



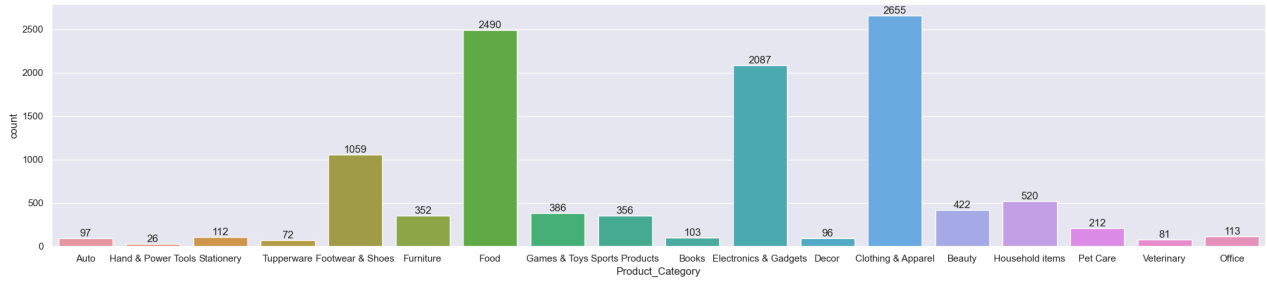

```
From above graphs we can see that most of the buyers are working in IT, Healthcare and Aviation sector
```

In [240]:
```
# here top 3 customers are who more than 3 times order

ds.groupby(['Cust_name'])['Orders'].count().sort_values(ascending=False).head(3)
```

Out[240]:
```
Cust_name
Vishakha    42
Shreyshi    32
Sudevi      30
Name: Orders, dtype: int64
```

Product_Category

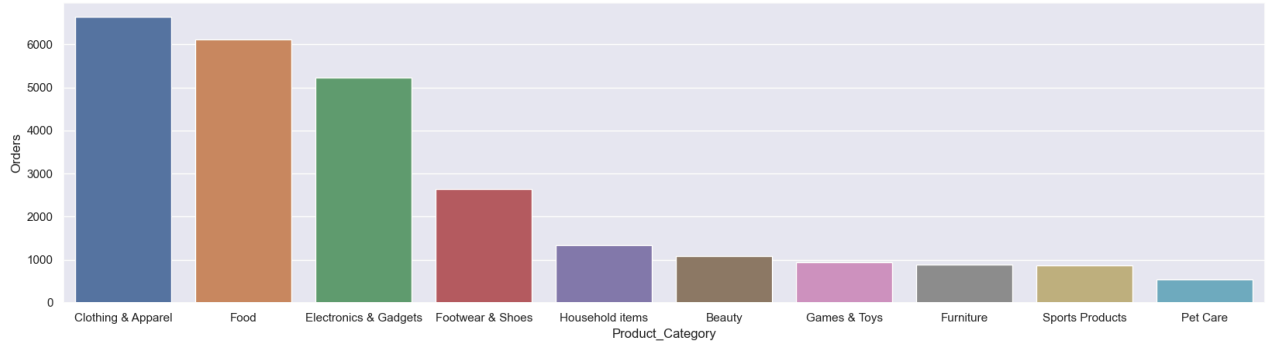
In [258]:
```
plt.figure(figsize=(25,5))
ax=sns.countplot(x='Product_Category' , data=ds)

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



In [265]:
```
pro_ord=ds.groupby('Product_Category' , as_index=False)['Orders'].sum().sort_values(by='Orders' , ascending=False).head(
```

In [266]:
```
plt.figure(figsize=(20,5))
sns.barplot(x='Product_Category', y="Orders" , data=pro_ord)
```

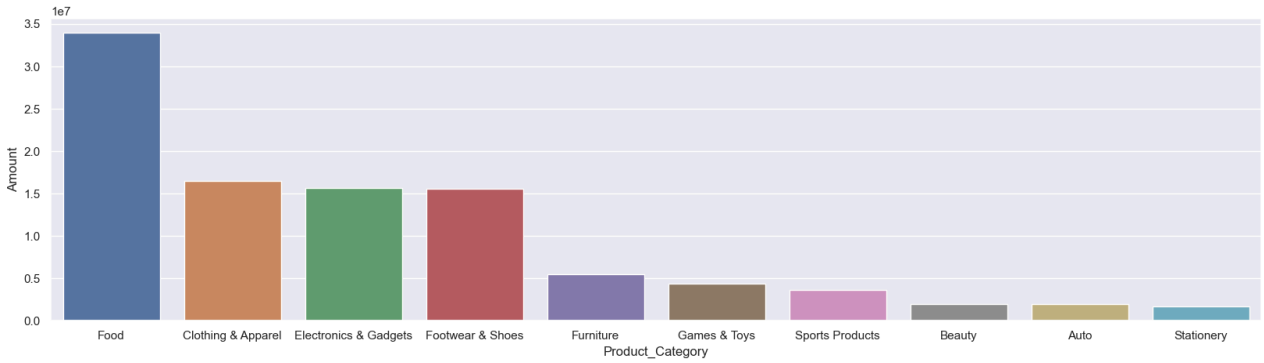
Out[266]: <Axes: xlabel='Product_Category', ylabel='Orders'>



In [267]:
```
pro_sales=ds.groupby('Product_Category' , as_index=False)['Amount'].sum().sort_values(by='Amount' , ascending=False).head
```

In [269]:
```
plt.figure(figsize=(20,5))
sns.barplot(x='Product_Category', y="Amount" , data=pro_sales)
```

Out[269]:
```
<Axes: xlabel='Product_Category', ylabel='Amount'>
```




```
From above graphs we can see that most of the sold products are from Food, Clothing and Electronics category
```

Conclusion:

Married women age group 26-35 yrs from UP, Maharastra and Karnataka working in IT, Healthcare and Aviation are more likely to buy products from Food, Clothing and Electronics category

In [ ]:

In [ ]: