

```
In [1]: !pip install numpy
```

```
Defaulting to user installation because normal site-packages is not writeable  
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-packages (1.24.3)
```

```
In [45]: !pip install pandas
```

```
Defaulting to user installation because normal site-packages is not writeable  
Requirement already satisfied: pandas in c:\programdata\anaconda3\lib\site-packages (2.0.3)  
Requirement already satisfied: python-dateutil>=2.8.2 in c:\programdata\anaconda3\lib\site-packages (from pa  
ndas) (2.8.2)  
Requirement already satisfied: pytz>=2020.1 in c:\programdata\anaconda3\lib\site-packages (from pandas) (202  
3.3.post1)  
Requirement already satisfied: tzdata>=2022.1 in c:\programdata\anaconda3\lib\site-packages (from pandas) (2  
023.3)  
Requirement already satisfied: numpy>=1.21.0 in c:\programdata\anaconda3\lib\site-packages (from pandas) (1.  
24.3)  
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-packages (from python-dateutil>  
=2.8.2->pandas) (1.16.0)
```

```
In [46]: !pip install matplotlib
```

```
Out[46]: ['Defaulting to user installation because normal site-packages is not writeable',
'Requirement already satisfied: matplotlib in c:\\users\\shrut\\appdata\\roaming\\python\\python311\\site-p
ackages (3.8.2)',
'Requirement already satisfied: contourpy>=1.0.1 in c:\\programdata\\anaconda3\\lib\\site-packages (from ma
tplotlib) (1.0.5)',
'Requirement already satisfied: cycler>=0.10 in c:\\programdata\\anaconda3\\lib\\site-packages (from matplo
tlib) (0.11.0)',
'Requirement already satisfied: fonttools>=4.22.0 in c:\\programdata\\anaconda3\\lib\\site-packages (from m
atplotlib) (4.25.0)',
'Requirement already satisfied: kiwisolver>=1.3.1 in c:\\programdata\\anaconda3\\lib\\site-packages (from m
atplotlib) (1.4.4)',
'Requirement already satisfied: numpy<2,>=1.21 in c:\\programdata\\anaconda3\\lib\\site-packages (from matp
lotlib) (1.24.3)',
'Requirement already satisfied: packaging>=20.0 in c:\\programdata\\anaconda3\\lib\\site-packages (from mat
plotlib) (23.1)',
'Requirement already satisfied: pillow>=8 in c:\\programdata\\anaconda3\\lib\\site-packages (from matplotli
b) (9.4.0)',
'Requirement already satisfied: pyparsing>=2.3.1 in c:\\programdata\\anaconda3\\lib\\site-packages (from ma
tplotlib) (3.0.9)',
'Requirement already satisfied: python-dateutil>=2.7 in c:\\programdata\\anaconda3\\lib\\site-packages (fro
m matplotlib) (2.8.2)',
'Requirement already satisfied: six>=1.5 in c:\\programdata\\anaconda3\\lib\\site-packages (from python-dat
eutil>=2.7->matplotlib) (1.16.0)']
```

```
In [ ]: !pip install seaborn
```

```
In [ ]: #IMPORT PYTHON LABRARIES
import numpy as np
import pandas as pd
import matplotlib as plt
%matplotlib inline
import seaborn as sns
```

In [6]: `df=pd.read_csv(r'C:\Users\shrut\OneDrive\Desktop\DATA SETS\Python_Diwali_Sales_Analysis-main\Python_Diwali_Sa`

In [ ]: `df.shape`

In [8]: `df.head(10)`

Out[8]:

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

In [9]: 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 [10]: df.drop(['Status', 'unnamed1'], axis=1, inplace=True)

```
In [11]: 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 [12]: pd.isnull(df)
```

```
Out[12]:
```

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

11251 rows × 13 columns



```
In [13]: pd.isnull(df).sum()
```

```
Out[13]: 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 [14]: df.shape
```

```
Out[14]: (11251, 13)
```

```
In [15]: df.dropna(inplace=True)
```

```
In [16]: pd.isnull(df).sum()
```

```
Out[16]: 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 [17]: df['Amount']=df['Amount'].astype('int')
```

```
In [18]: df['Amount'].dtypes
```

```
Out[18]: dtype('int32')
```

```
In [19]: df.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 [20]: df.describe



```
Out[20]: <bound method NDFrame.describe of
us \
0      1002903      Sanskriti  P00125942      F      26-35      28      0
1      1000732      Kartik    P00110942      F      26-35      35      1
2      1001990      Bindu    P00118542      F      26-35      35      1
3      1001425      Sudevi   P00237842      M      0-17      16      0
4      1000588      Joni     P00057942      M      26-35      28      1
...      ...      ...      ...      ...      ...      ...
11246  1000695      Manning  P00296942      M      18-25      19      1
11247  1004089  Reichenbach  P00171342      M      26-35      33      0
11248  1001209      Oshin   P00201342      F      36-45      40      0
11249  1004023      Noonan  P00059442      M      36-45      37      0
11250  1002744      Brumley  P00281742      F      18-25      19      0
```

```

State      Zone      Occupation  Product_Category  Orders  \
0      Maharashtra  Western      Healthcare      Auto      1
1      Andhra Pradesh  Southern      Govt      Auto      3
2      Uttar Pradesh  Central      Automobile      Auto      3
3      Karnataka      Southern      Construction      Auto      2
4      Gujarat      Western      Food Processing      Auto      2
...      ...      ...      ...      ...      ...
11246  Maharashtra  Western      Chemical      Office      4
11247  Haryana      Northern      Healthcare      Veterinary  3
11248  Madhya Pradesh  Central      Textile      Office      4
11249  Karnataka      Southern      Agriculture      Office      3
11250  Maharashtra  Western      Healthcare      Office      3
```

```

Amount
0      23952
1      23934
2      23924
3      23912
4      23877
...      ...
11246      370
11247      367
11248      213
11249      206
11250      188
```

```
[11239 rows x 13 columns]>
```

```
In [21]: df[['Age', 'Orders', 'Amount']].describe
```

```
Out[21]: <bound method NDFrame.describe of
0      28      1  23952
1      35      3  23934
2      35      3  23924
3      16      2  23912
4      28      2  23877
...     ...     ...     ...
11246   19      4    370
11247   33      3    367
11248   40      4    213
11249   37      3    206
11250   19      3    188

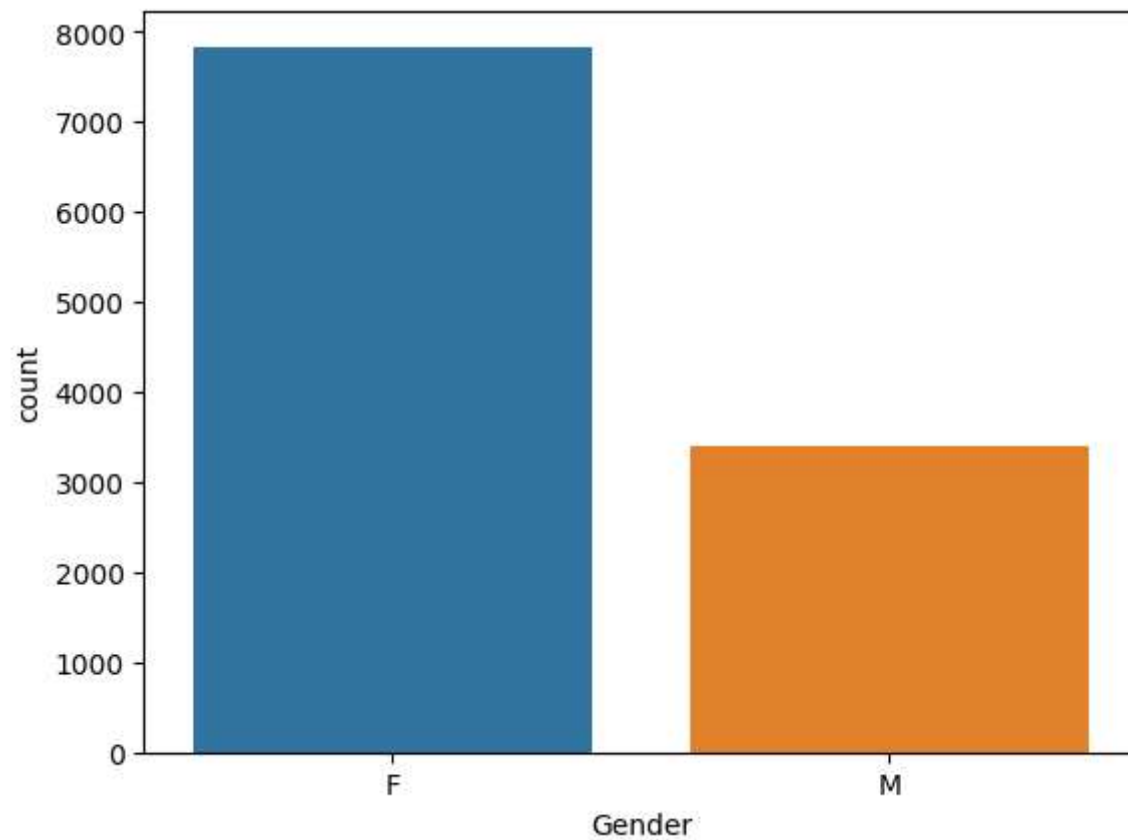
[11239 rows x 3 columns]>
```

```
In [22]: #GENDER AND AGE WISE
```

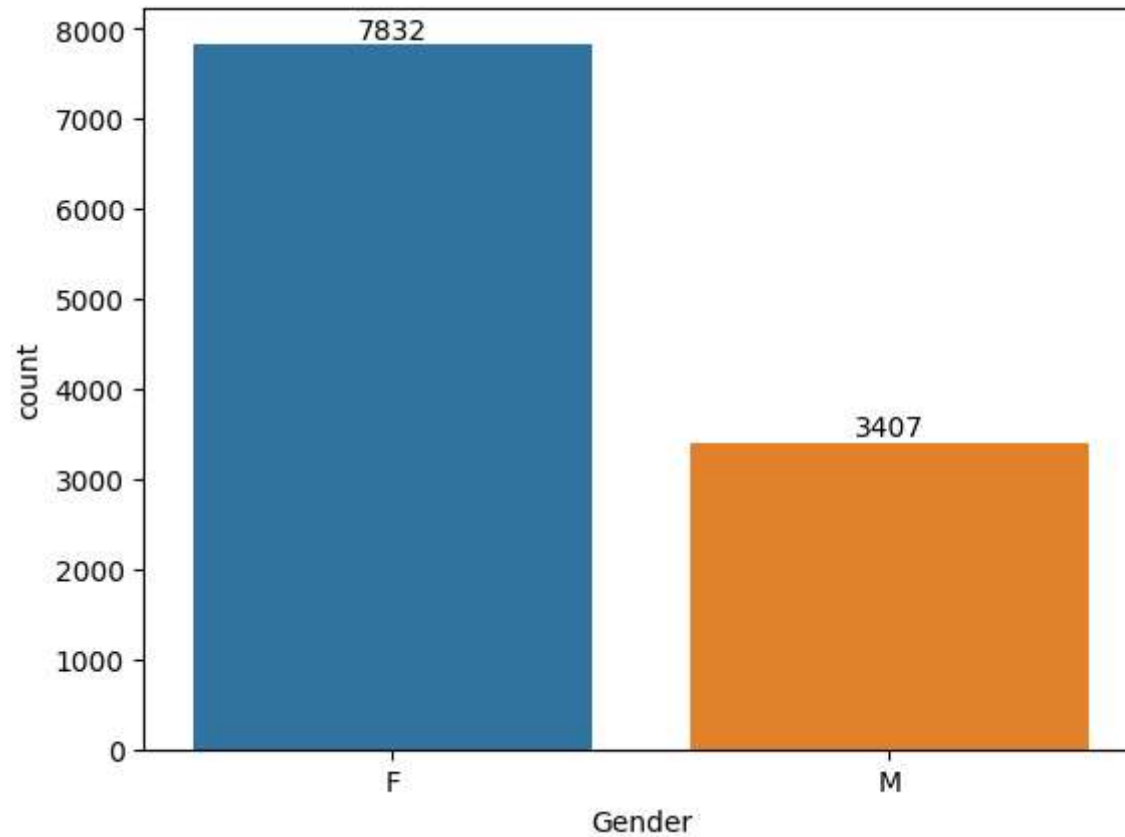
In [23]: *#graph create*

```
sns.countplot(x='Gender',data=df)
```

Out[23]: <Axes: xlabel='Gender', ylabel='count'>



```
In [24]: # ADD Lables  
#total shopping gender wise  
ax = sns.countplot(x='Gender',data=df)  
for bars in ax.containers:  
    ax.bar_label(bars)
```



```
In [25]: df.columns
```

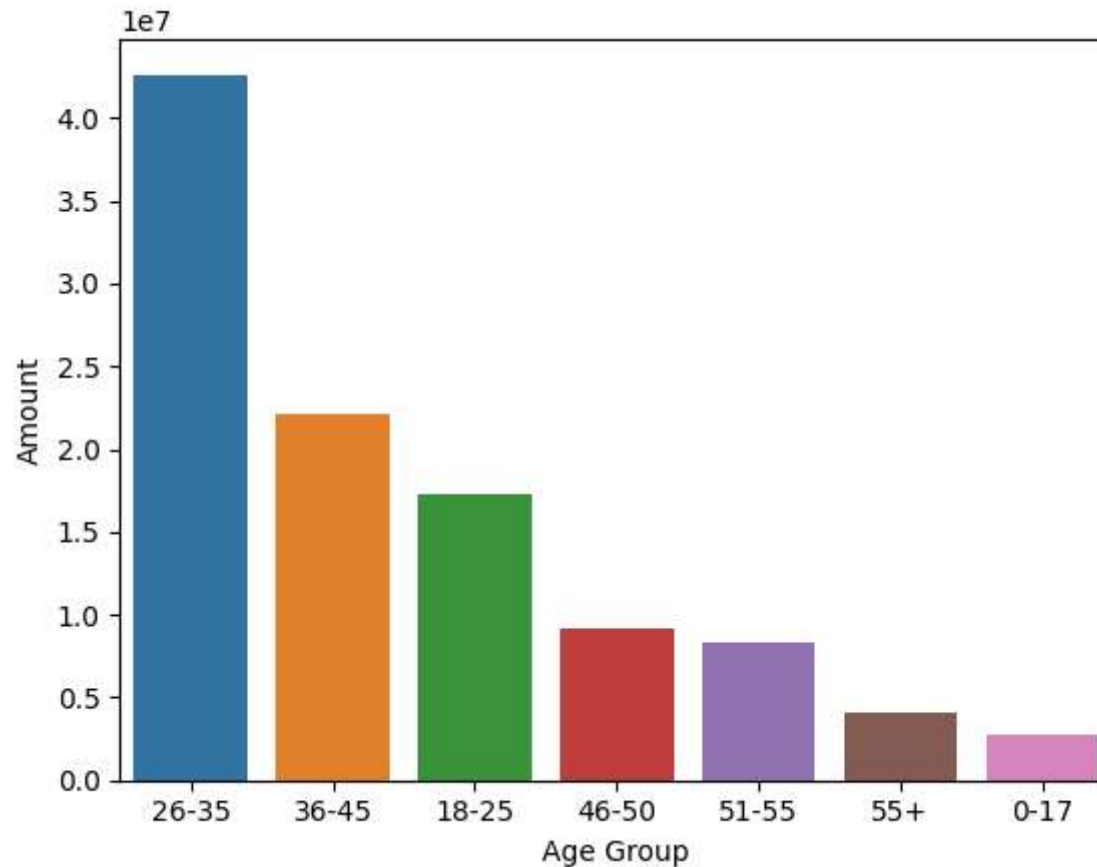
```
Out[25]: Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',  
               'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',  
               'Orders', 'Amount'],  
              dtype='object')
```

## AGE WISE SHOPPING MALE & FEMALE

ax=sns.countplot(data=df,x='Age Group',hue='Gender') for bars in ax.containers: ax.bar\_label(bars)

```
In [26]: Sales_age=df.groupby(['Age Group'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
sns.barplot(x='Age Group',y='Amount',data=Sales_age)
```

```
Out[26]: <Axes: xlabel='Age Group', ylabel='Amount'>
```



```
In [27]: #From above graphs we can see that most of the buyers are of age group between 26-35 yrs female
```

## STATE

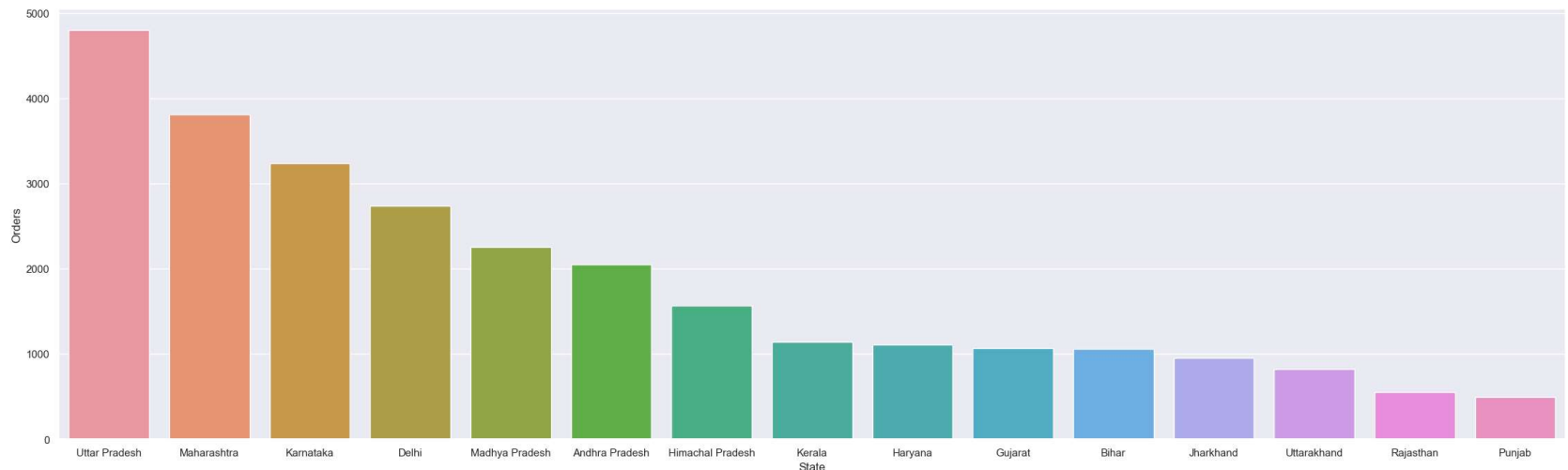
```
In [28]: df.columns
```

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

## STATE WISE SALES

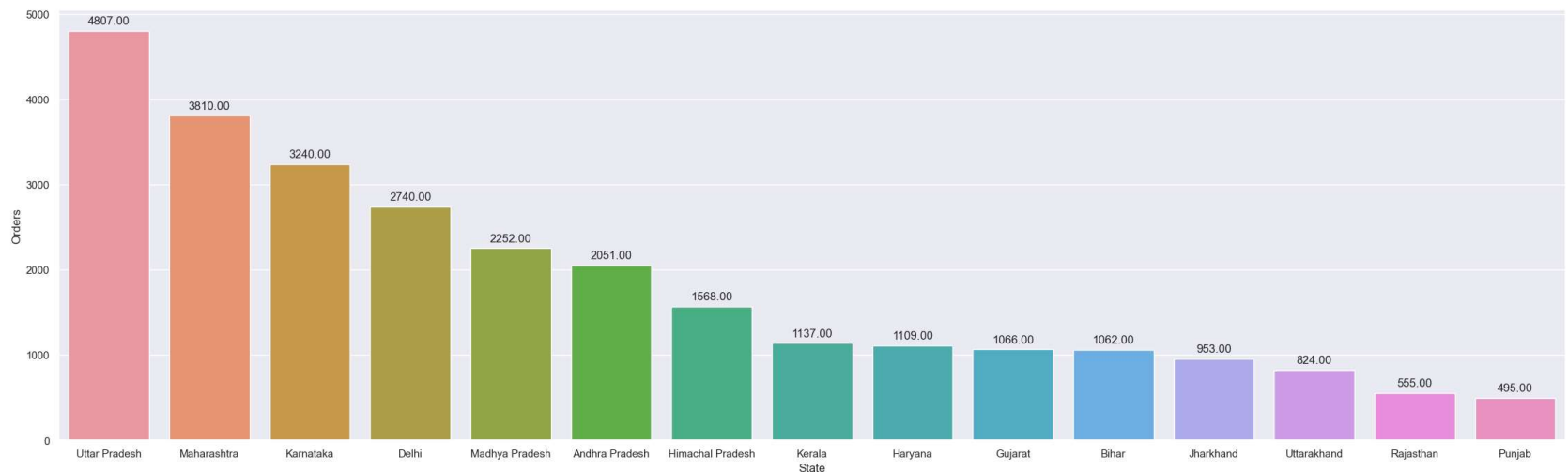
```
In [29]: order_state=df.groupby(['State'],as_index=False)['Orders'].sum().sort_values(by='Orders',ascending=False).head()  
sns.set(rc={'figure.figsize':(28,8)})  
sns.barplot(data=order_state,x='State',y='Orders')
```

```
Out[29]: <Axes: xlabel='State', ylabel='Orders'>
```



#From above graphs we can see that most of the orders from Uttar Pradesh, Maharashtra and Karnataka respectively

```
In [30]: #Label add;
order_state=df.groupby(['State'],as_index=False)['Orders'].sum().sort_values(by='Orders',ascending=False).head(15)
sns.set(rc={'figure.figsize':(28,8)})
ax=sns.barplot(data=order_state,x='State',y='Orders')
for p in ax.patches:
    ax.annotate(f'{p.get_height():.2f}', (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center', xytext=(0, 10), textcoords='offset points')
```



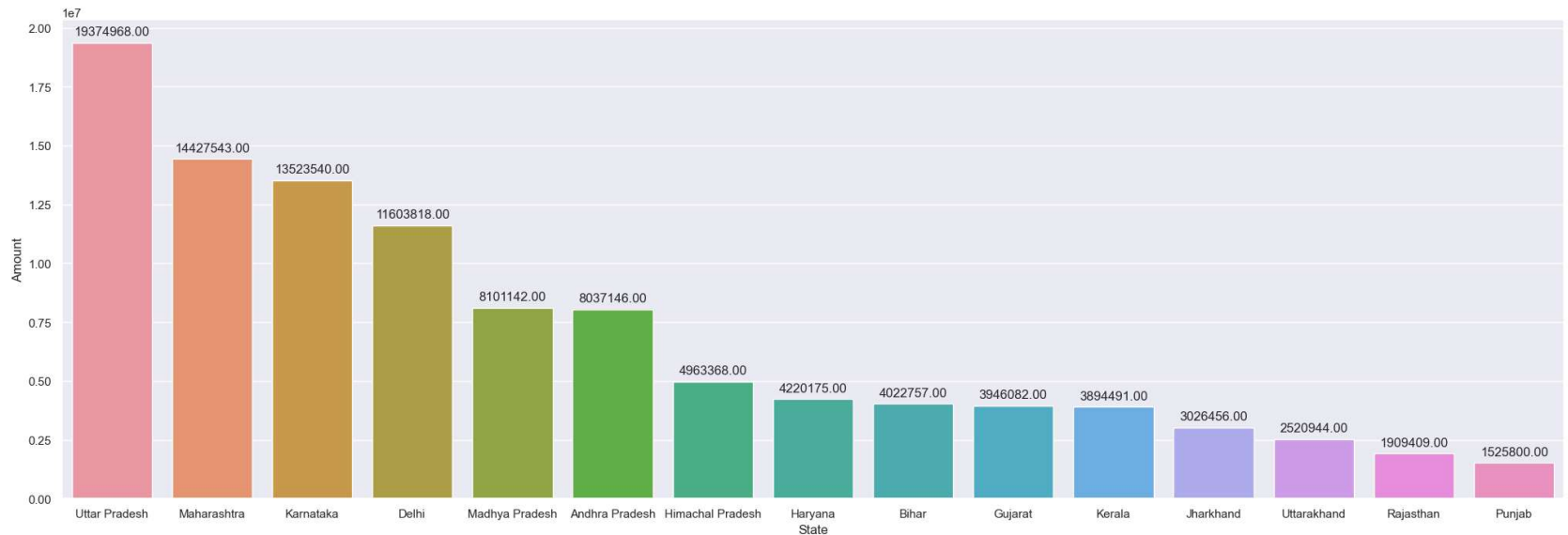
#From above graphs we can see that most of the orders from Uttar Pradesh, Maharashtra and Karnataka respectively

## STATE WISE SALES:

```
In [31]: sales_state=df.groupby(['State'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False).head(15)

sns.set(rc={'figure.figsize':(25,8)})

ax=sns.barplot(data=sales_state,x='State',y='Amount')
for p in ax.patches:
    ax.annotate(f'{p.get_height():.2f}', (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center', xytext=(0, 10), textcoords='offset points')
```

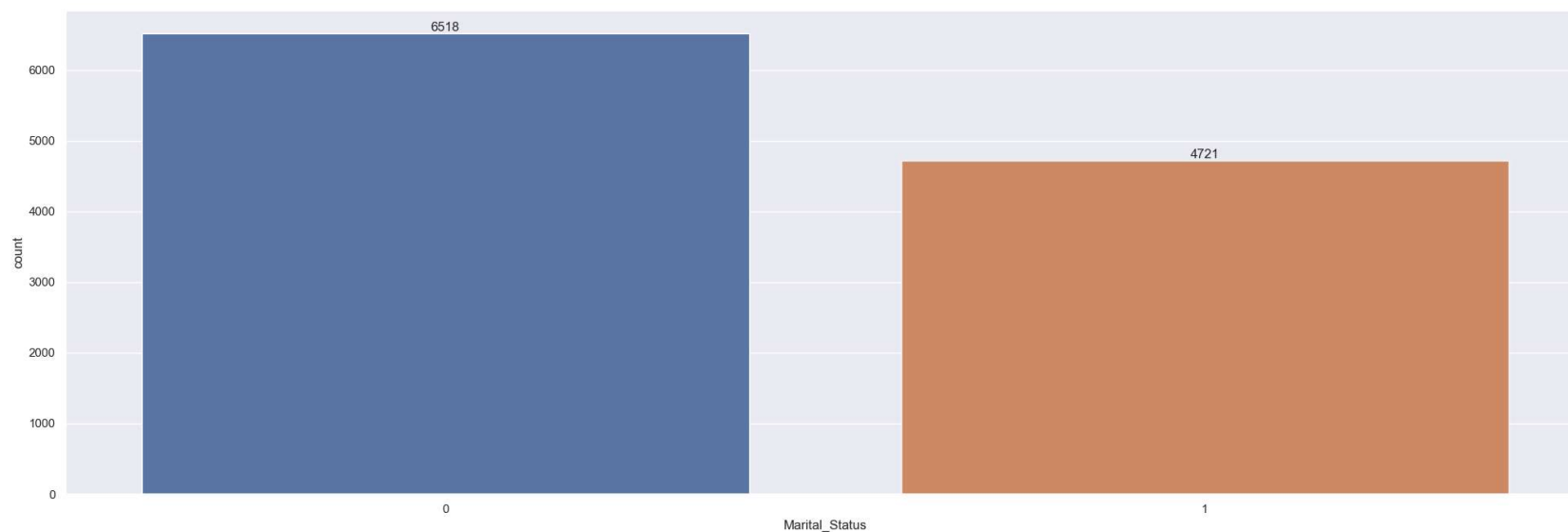


#From above graphs we can see that most revenue/amount from Uttar Pradesh, Maharashtra and Karnataka respectively



## Marital Status

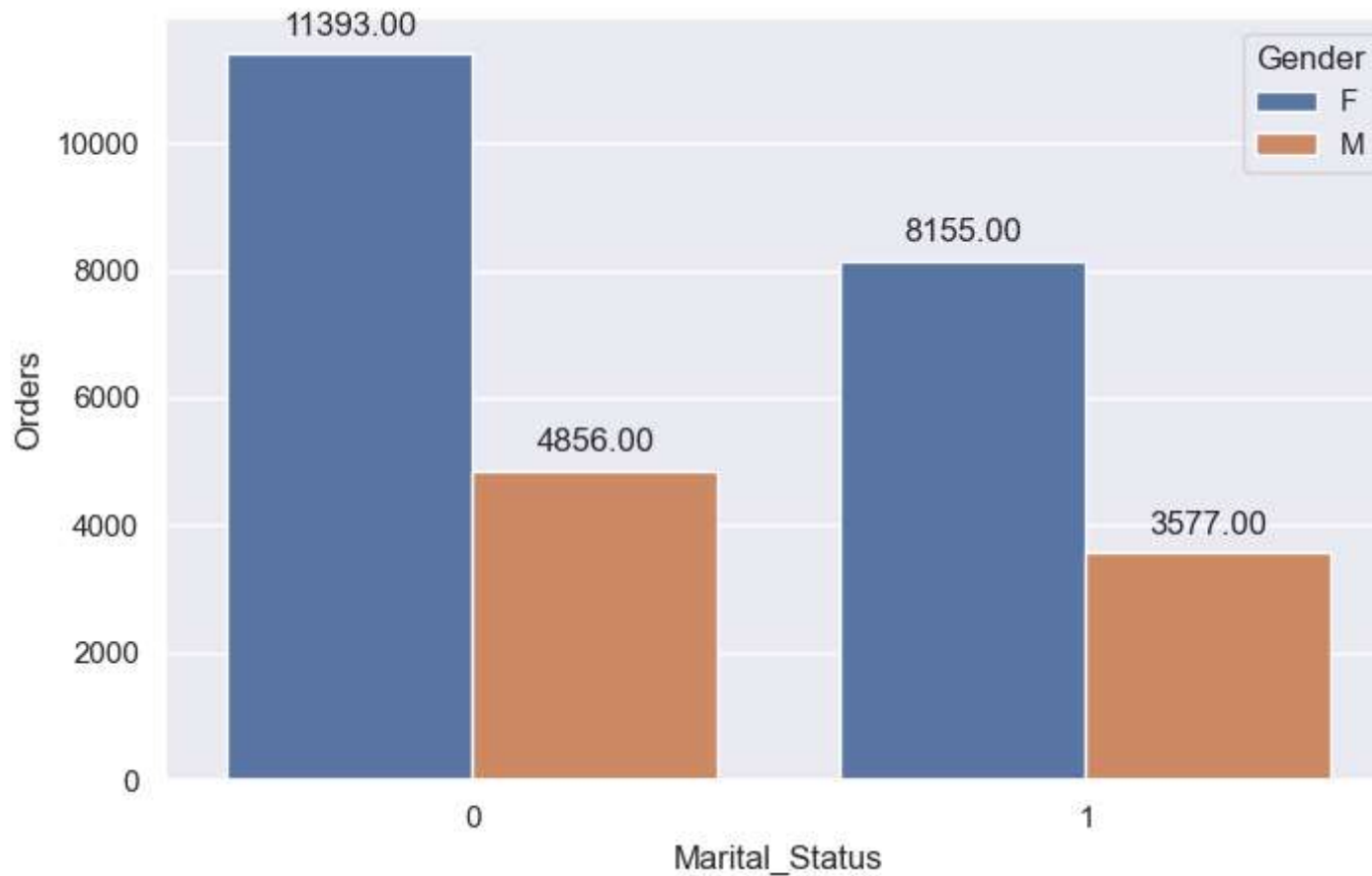
```
In [32]: ax=sns.countplot(data=df,x= 'Marital_Status')
sns.set(rc={'figure.figsize':(8,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



#From above graphs we can see that male & female marital\_status

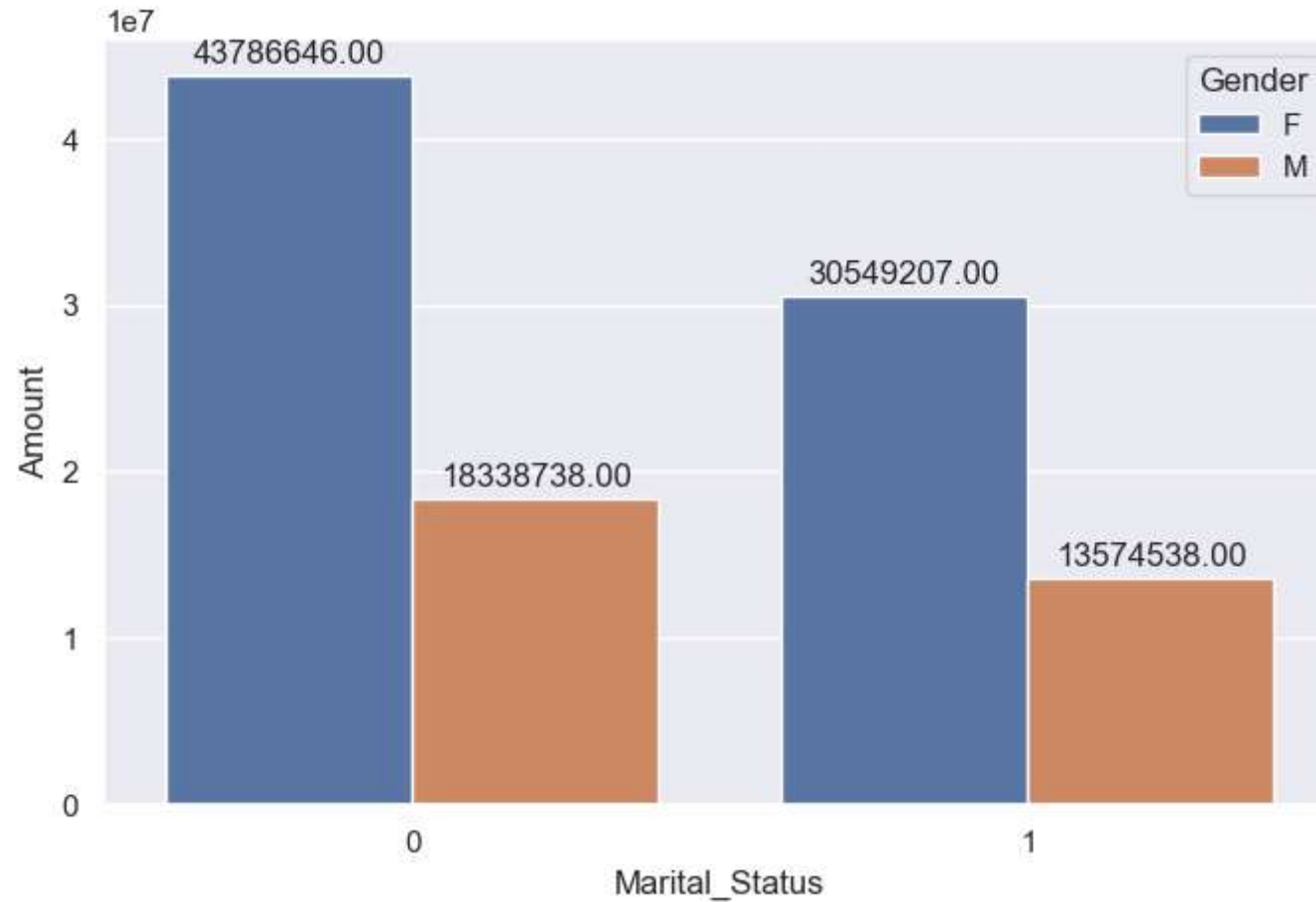
## male/ female order status.

```
In [33]: order_state=df.groupby(['Marital_Status', 'Gender'],as_index= False)['Orders'].sum().sort_values(by='Orders',a
ax=sns.barplot(data=order_state,x='Marital_Status',y='Orders',hue='Gender')
for p in ax.patches:
    ax.annotate(f'{p.get_height():.2f}', (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center', xytext=(.0, 10), textcoords='offset points')
```



```
#From above graphs we can see that most of the orders are married (women)
```

```
In [34]: state_sales=df.groupby(['Marital_Status', 'Gender'],as_index=False)['Amount'].sum().sort_values(by='Amount',as
ax=sns.barplot(data=state_sales,x='Marital_Status',y='Amount',hue='Gender')
for p in ax.patches:
    ax.annotate(f'{p.get_height():.2f}', (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center', xytext=(.0, 8), textcoords='offset points')
```



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

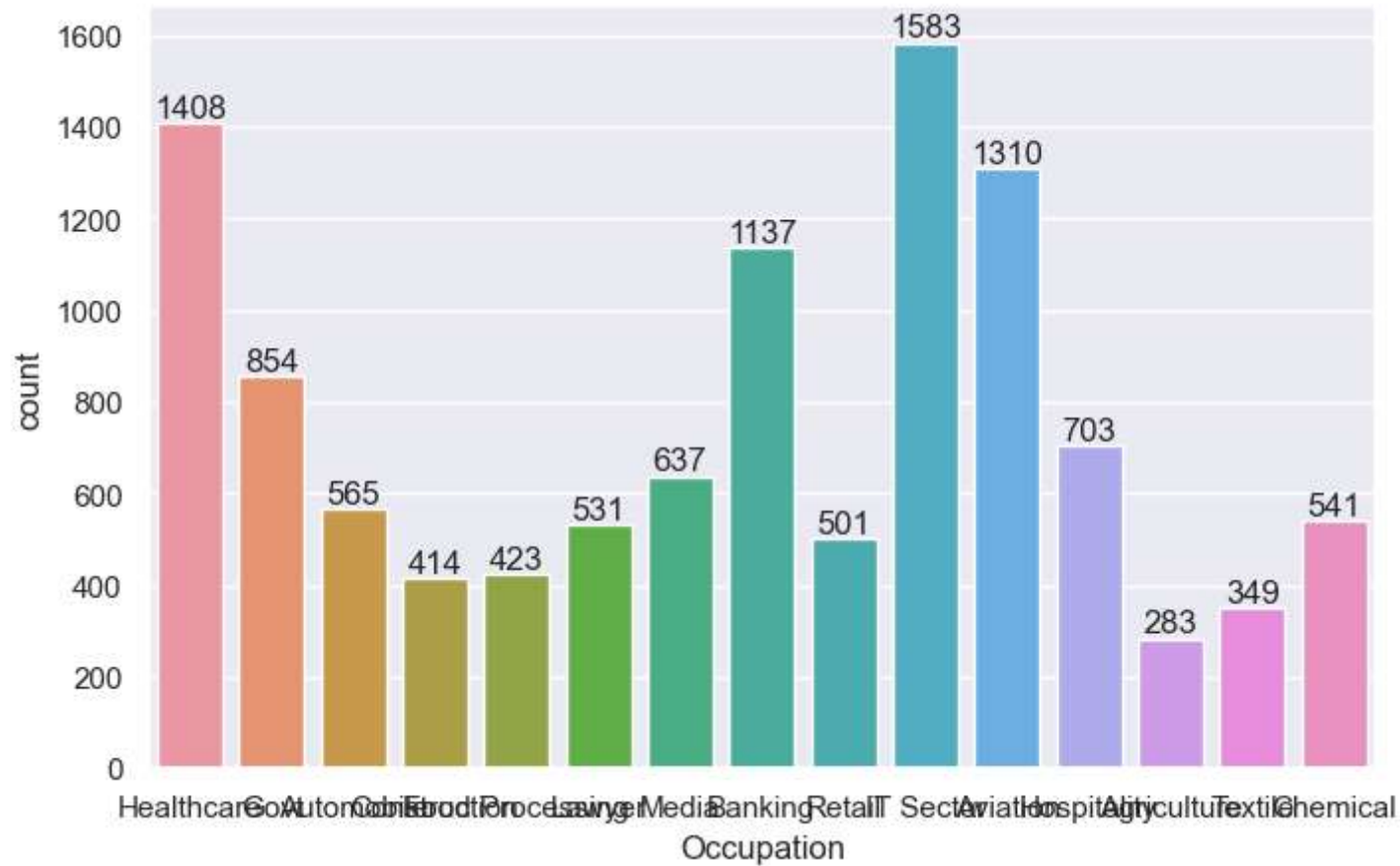
## Occupation

In [35]: `df.head(10)`

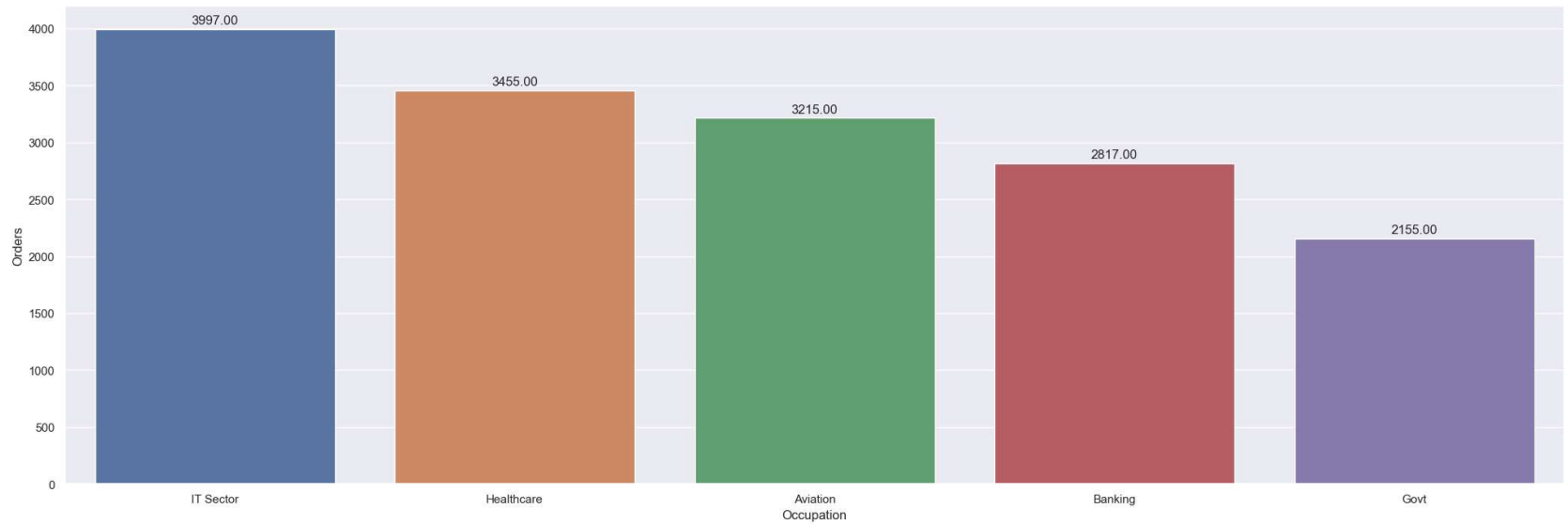
Out[35]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	On
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare		Auto
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt		Auto
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile		Auto
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction		Auto
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing		Auto
5	1000588	Joni	P00057942	M	26-35	28	1	Himachal Pradesh	Northern	Food Processing		Auto
6	1001132	Balk	P00018042	F	18-25	25	1	Uttar Pradesh	Central	Lawyer		Auto
8	1003224	Kushal	P00205642	M	26-35	35	0	Uttar Pradesh	Central	Govt		Auto
9	1003650	Ginny	P00031142	F	26-35	26	1	Andhra Pradesh	Southern	Media		Auto
10	1003829	Harshita	P00200842	M	26-35	34	0	Delhi	Central	Banking		Auto

```
In [36]: ax=sns.countplot(data=df,x='Occupation')
sns.set(rc={'figure.figsize':(30,15)})
for bars in ax.containers:
    ax.bar_label(bars)
```



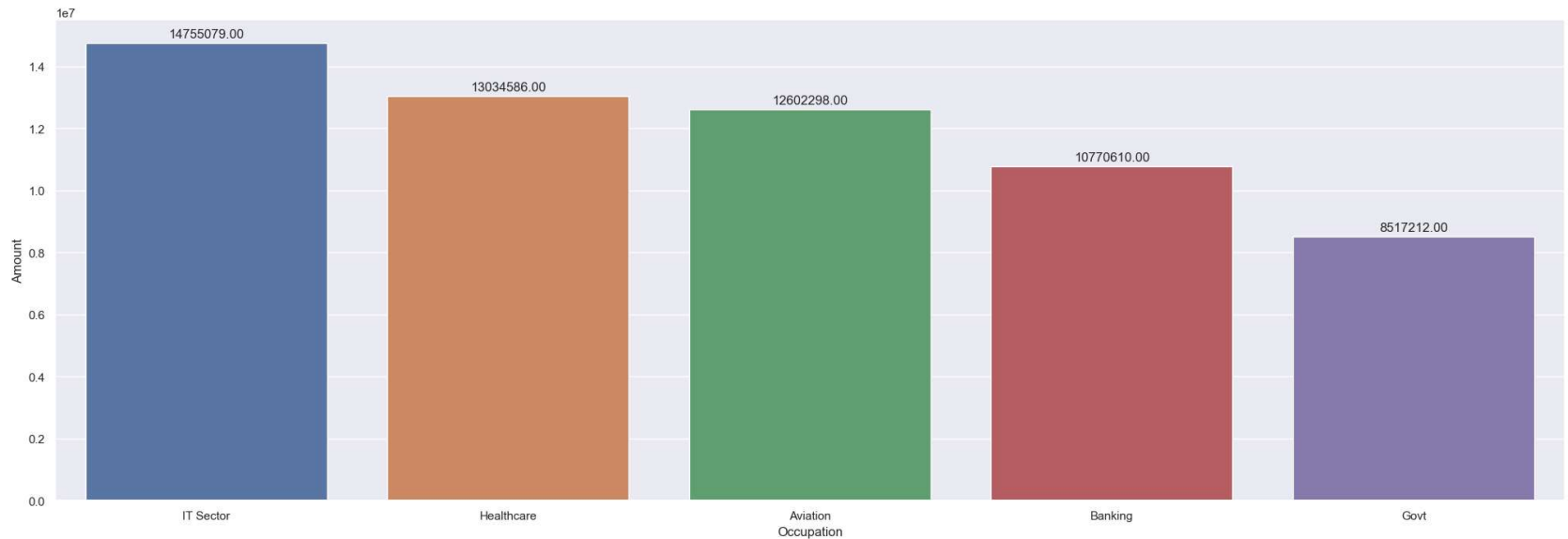
```
In [37]: state_order=df.groupby(['Occupation'],as_index=False)['Orders'].sum().sort_values(by='Orders',ascending=False)
sns.set(rc={'figure.figsize':(25,8)})
ax=sns.barplot(data=state_order,x='Occupation',y='Orders')
for p in ax.patches:
    ax.annotate(f'{p.get_height():.2f}', (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center', xytext=(.0, 8), textcoords='offset points')
```



***From above graphs we can see that most of the buyers are IT SECTOR ,HEALTHCARE,AVIATION OCCUPATION,BANKING,GOVT***

## AMOUNT

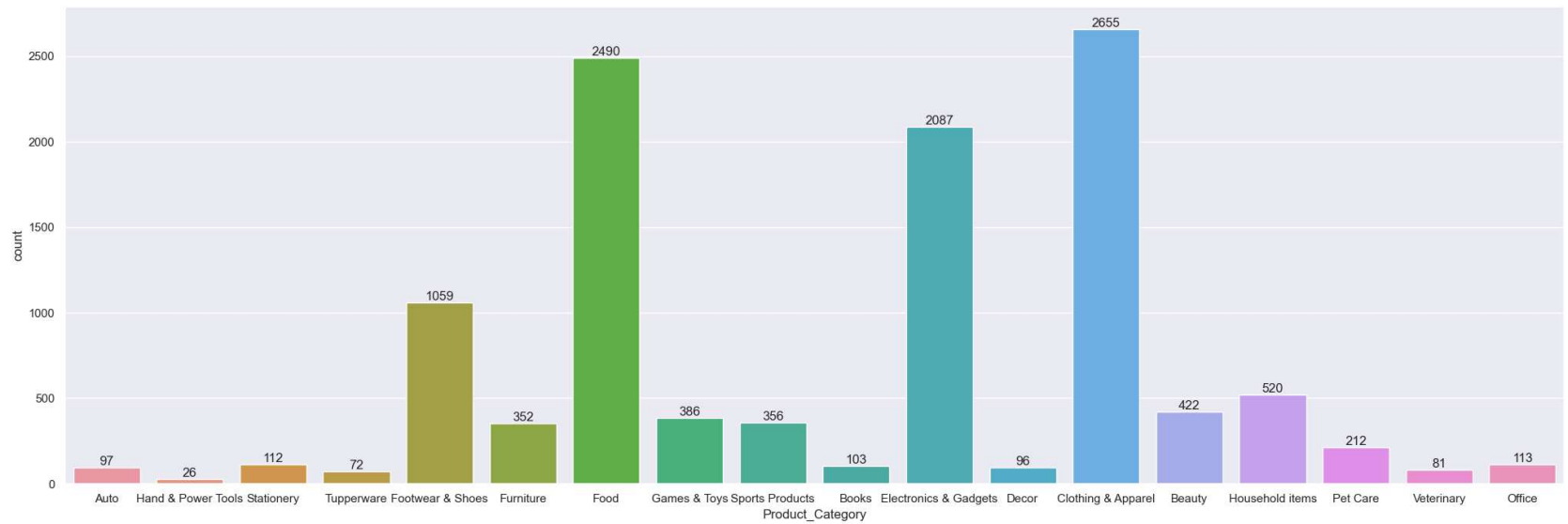
```
In [38]: state_order=df.groupby(['Occupation'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
sns.set(rc={'figure.figsize':(25,8)})
ax=sns.barplot(data=state_order,x='Occupation',y='Amount')
for p in ax.patches:
    ax.annotate(f'{p.get_height():.2f}', (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center', xytext=(.0, 8), textcoords='offset points')
```



From above graphs we can see that most of the buyers are IT SECTOR ,HEALTHCARE,AVIATION OCCUPATION,BANKING,GOVT

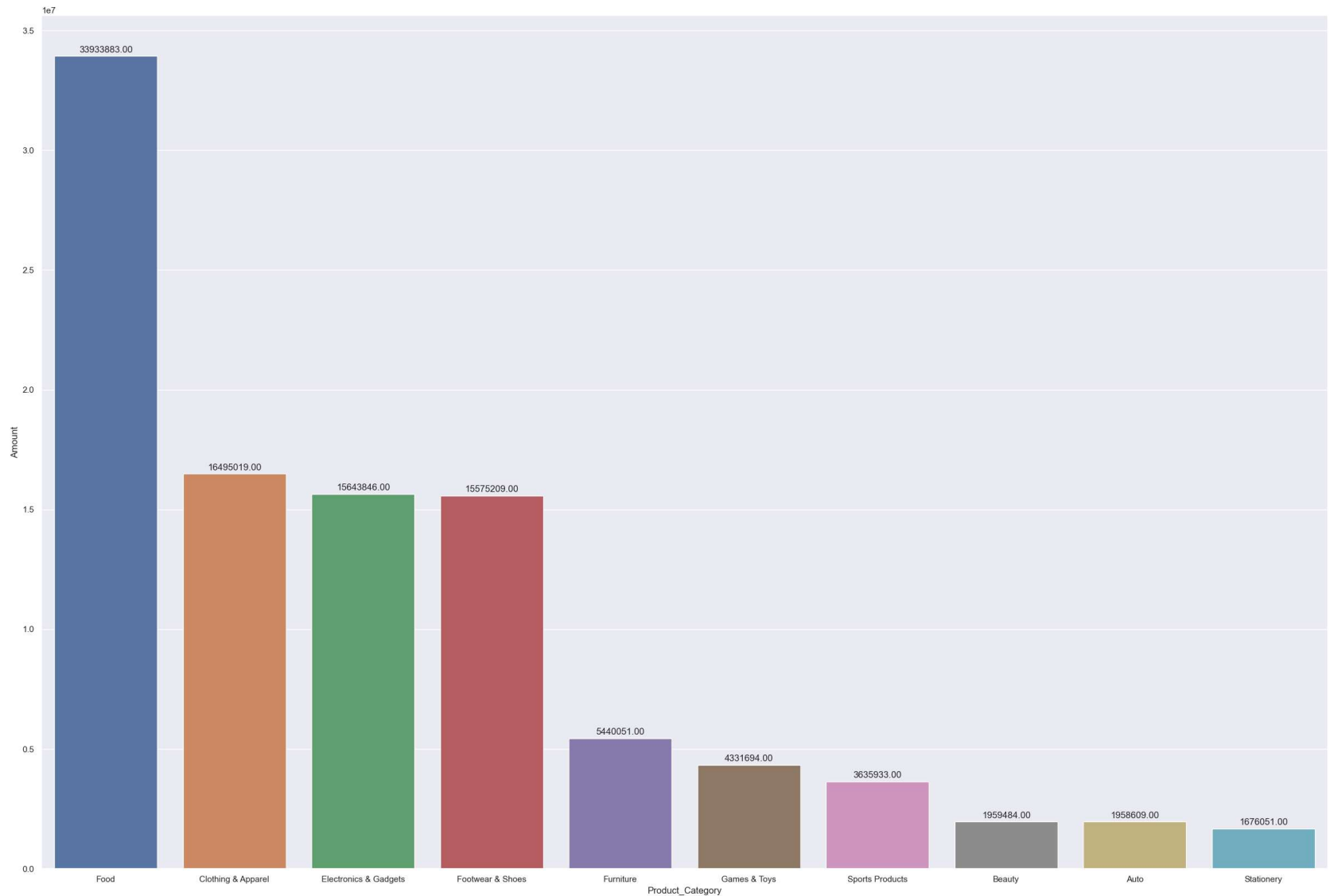
## Product\_Category

```
In [39]: ax=sns.countplot(data=df,x='Product_Category')
sns.set(rc={'figure.figsize':(30,15)})
for bars in ax.containers:
    ax.bar_label(bars)
```





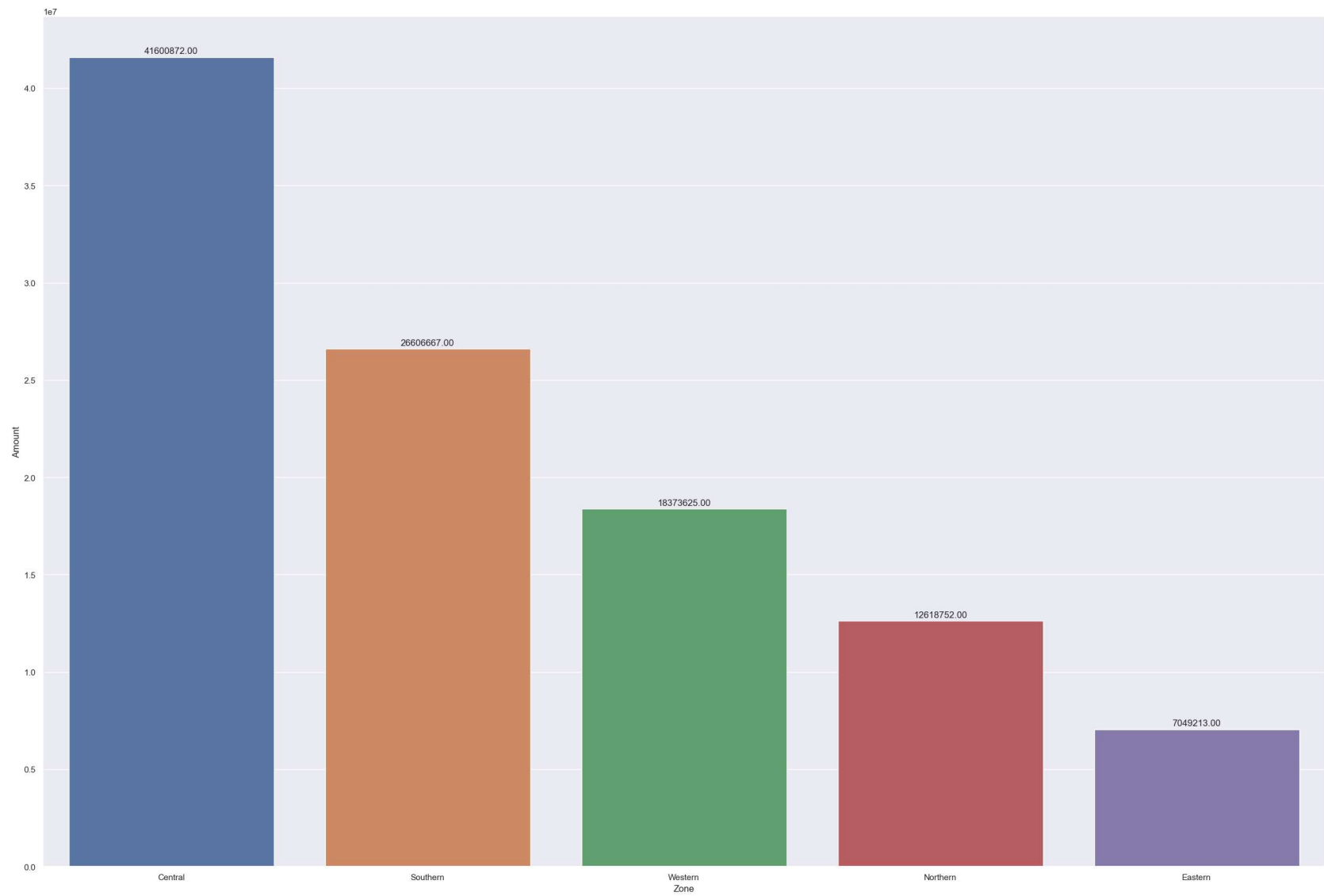
```
In [40]: state_amount=df.groupby(['Product_Category'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=True)
sns.set(rc={'figure.figsize':(30,20)})
ax=sns.barplot(data=state_amount,x='Product_Category',y='Amount')
for p in ax.patches:
    ax.annotate(f'{p.get_height():.2f}', (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center', xytext=(.0, 8), textcoords='offset points')
```



In [41]: *#From above graphs we can see that most of the sold products are from Food, Clothing and Electronics category*

## ZONE WISE SALES

```
In [42]: state_amount=df.groupby(['Zone'],as_index=False)['Amount'].sum().sort_values(by='Amount',ascending=False)
sns.set(rc={'figure.figsize':(30,20)})
ax=sns.barplot(data=state_amount,x='Zone',y='Amount')
for p in ax.patches:
    ax.annotate(f'{p.get_height():.2f}', (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center', xytext=(.0, 8), textcoords='offset points')
```

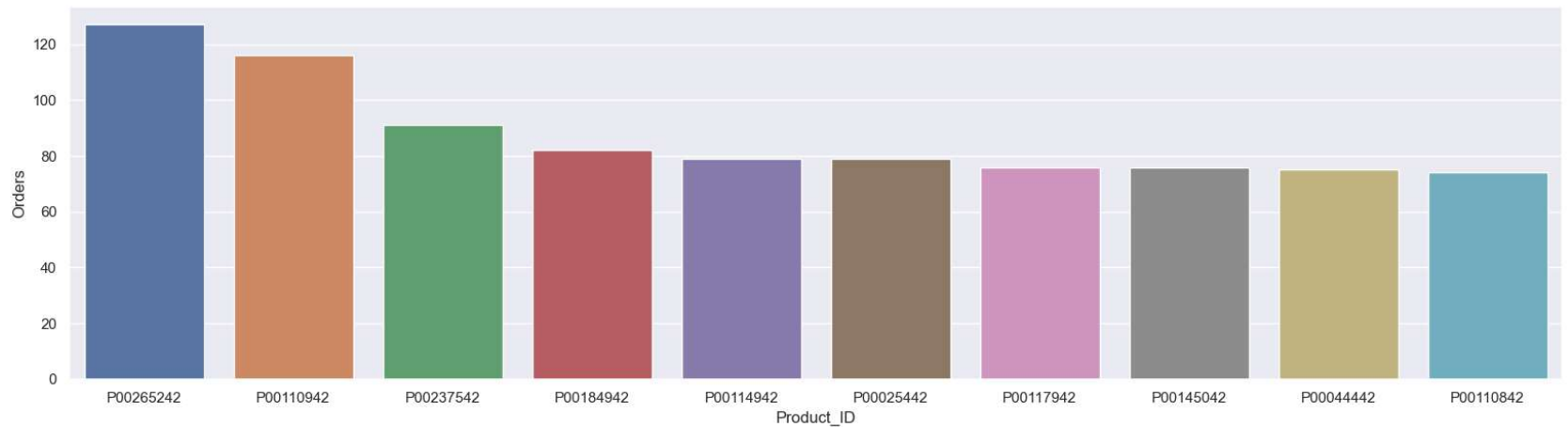


## Product\_ID

```
In [43]: sales_state = df.groupby(['Product_ID'], as_index=False)['Orders'].sum().sort_values(by='Orders', ascending=False)

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

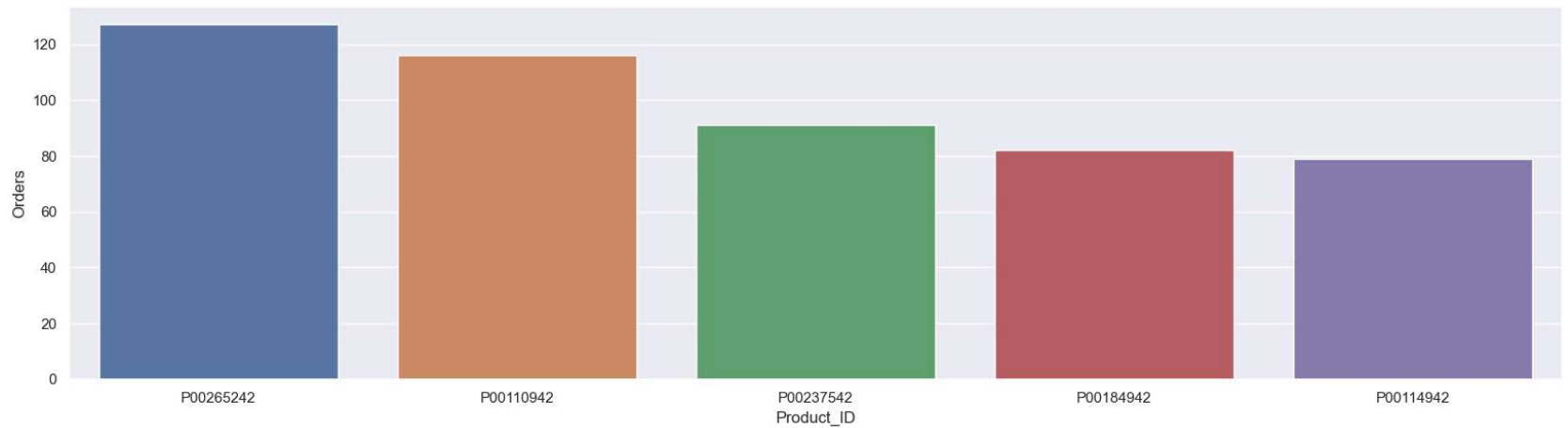
Out[43]: <Axes: xlabel='Product\_ID', ylabel='Orders'>



```
In [47]: sales_state = df.groupby(['Product_ID'], as_index=False)['Orders'].sum().sort_values(by='Orders', ascending=False)

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

Out[47]: <Axes: xlabel='Product\_ID', ylabel='Orders'>



In [ ]: