```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

DATA CLEANING

```
df = pd.read_csv('Diwali Sales Data.csv', encoding= 'unicode_escape')
```

```
df.shape
```

```
(11251, 15)
```

```
df.head(10)
```

```
   User_ID  Cust_name Product_ID Gender Age Group  Age  Marital_Status
\
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

5  1000588       Joni  P00057942      M     26-35   28               1

6  1001132       Balk  P00018042      F     18-25   25               1

7  1002092   Shivangi  P00273442      F       55+   61               0

8  1003224     Kushal  P00205642      M     26-35   35               0

9  1003650      Ginny  P00031142      F     26-35   26               1


               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
5  Himachal Pradesh  Northern  Food Processing             Auto
1
6     Uttar Pradesh   Central           Lawyer             Auto
4
7       Maharashtra   Western        IT Sector             Auto
1
8     Uttar Pradesh   Central             Govt             Auto
2
9    Andhra Pradesh  Southern            Media             Auto
4

     Amount  Status  unnamed1
0  23952.00     NaN       NaN
1  23934.00     NaN       NaN
2  23924.00     NaN       NaN
3  23912.00     NaN       NaN
4  23877.00     NaN       NaN
5  23877.00     NaN       NaN
6  23841.00     NaN       NaN
7       NaN     NaN       NaN
8  23809.00     NaN       NaN
9  23799.99     NaN       NaN

df.tail(10)

       User_ID    Cust_name Product_ID Gender Age Group  Age
Marital_Status  \
11241  1003032     Matthias  P00058042      F     26-35   33
0
11242  1004344   Hildebrand  P00185442      F     26-35   27
1
11243  1005446      Sheetal  P00297742      M     51-55   53
0
11244  1005446      Sheetal  P00297742      M     51-55   53
0
11245  1004140    Bertelson  P00057442      F     26-35   31
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
Amount  \
11241           Delhi   Central  Hospitality           Office       3
384.0
11242           Delhi   Central   Healthcare           Office       2
382.0
11243         Gujarat   Western   Healthcare           Office       1
382.0
11244  Madhya Pradesh   Central   Healthcare           Office       2
382.0
11245           Delhi   Central     Aviation           Office       2
381.0
11246     Maharashtra   Western     Chemical           Office       4
370.0
11247         Haryana  Northern   Healthcare       Veterinary       3
367.0
11248  Madhya Pradesh   Central      Textile           Office       4
213.0
11249       Karnataka  Southern  Agriculture           Office       3
206.0
11250     Maharashtra   Western   Healthcare           Office       3
188.0

       Status  unnamed1
11241     NaN       NaN
11242     NaN       NaN
11243     NaN       NaN
11244     NaN       NaN
11245     NaN       NaN
11246     NaN       NaN
11247     NaN       NaN
11248     NaN       NaN
11249     NaN       NaN
11250     NaN       NaN
```

```
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
```

```
df.drop(['Status', 'unnamed1'], axis=1, inplace=True)  #delete null
column
```

```
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
```

```
pd.isnull(df)  #check null value with the help of true means have null
value false means no_null value   but it is not efficient to work
```

```
       User_ID  Cust_name  Product_ID  Gender  Age Group    Age  \
0        False      False       False   False      False  False
1        False      False       False   False      False  False
2        False      False       False   False      False  False
3        False      False       False   False      False  False
4        False      False       False   False      False  False
...        ...        ...         ...     ...        ...    ...
11246    False      False       False   False      False  False
11247    False      False       False   False      False  False
11248    False      False       False   False      False  False
11249    False      False       False   False      False  False
```

```
11250    False      False       False   False      False  False

       Marital_Status  State   Zone  Occupation  Product_Category
Orders  \
0               False  False  False       False             False
False
1               False  False  False       False             False
False
2               False  False  False       False             False
False
3               False  False  False       False             False
False
4               False  False  False       False             False
False
...               ...    ...    ...         ...               ...
...
11246           False  False  False       False             False
False
11247           False  False  False       False             False
False
11248           False  False  False       False             False
False
11249           False  False  False       False             False
False
11250           False  False  False       False             False
False

       Amount
0       False
1       False
2       False
3       False
4       False
...       ...
11246   False
11247   False
11248   False
11249   False
11250   False

[11251 rows x 13 columns]
```

```
pd.isnull(df).sum()    #here 0 means no null_values but 12 means there
12 null values on amount
```

```
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
```

```
# drop null values of amount
df.dropna(inplace=True)
```

```
pd.isnull(df).sum()    #check wheather it really delete data or not
check amount
```

```
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
```

```
# change data type
df['Amount'] = df['Amount'].astype('int')
```

```
df['Amount'].dtypes #check data_type of amount change or not
```

```
dtype('int64')
```

```
df.columns # check what are the columns in data_set
```

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

```
# describe() method returns ALL  (i.e. count, mean, std, etc)
df.describe()
```

```
            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
```

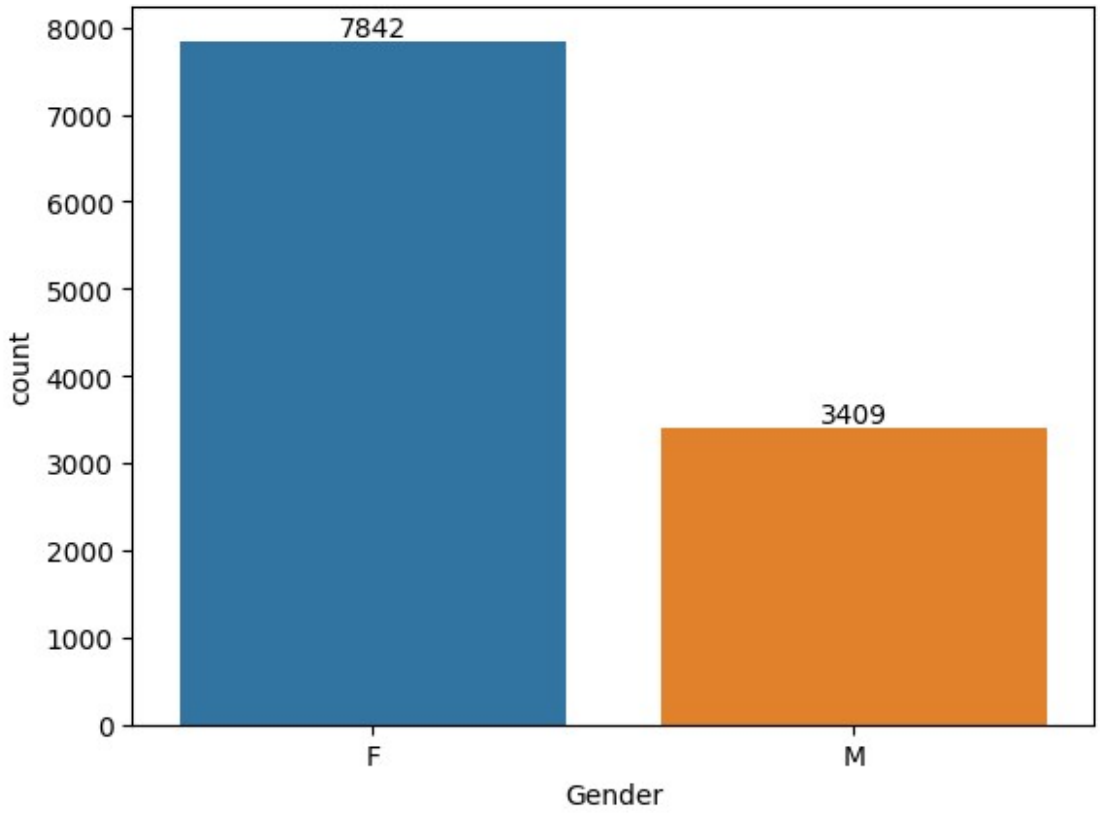
```
#FOR SPECIFIC DESCRIBE YOU WANT THEN
df[['Age', 'Orders', 'Amount']].describe()
```

```
                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
```

AFTER DATA CLEANING I MOVE TO DATA VISULIZATION

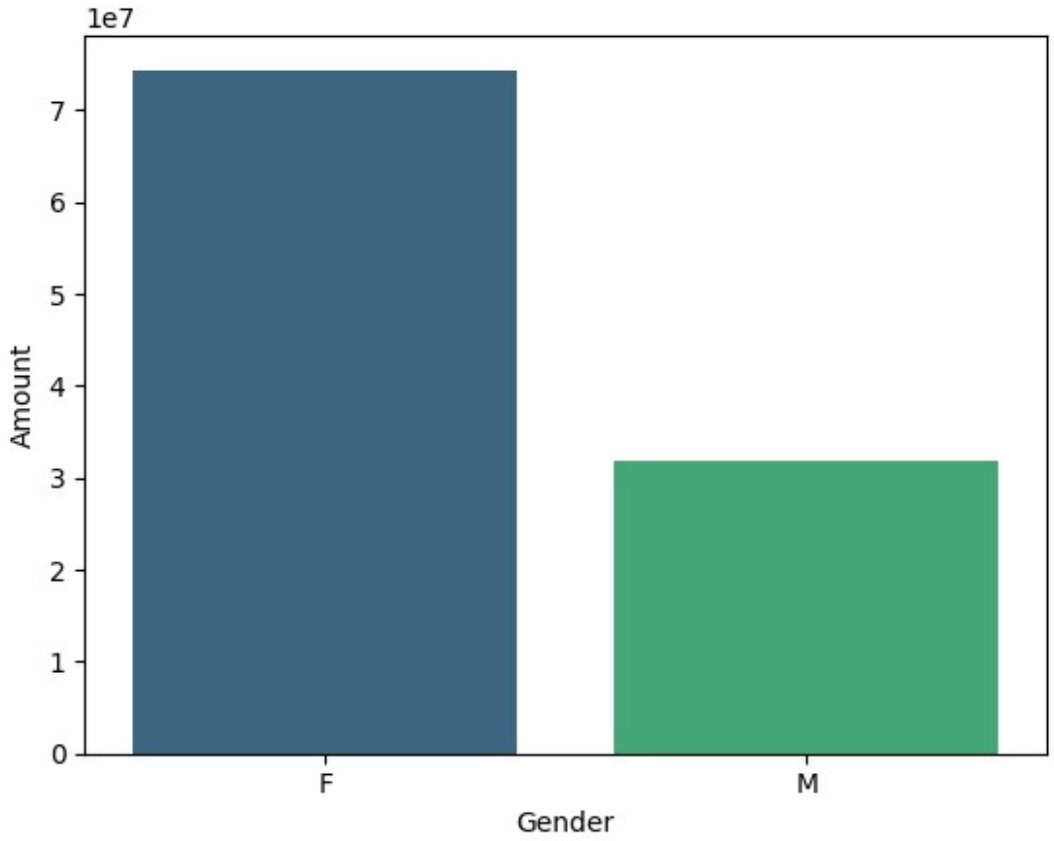
Gender

```
# plotting a bar chart for Gender and it's count
ax = sns.countplot(x='Gender', hue='Gender', data=df,
palette=['#1f77b4', '#ff7f0e'], legend=False)
for bars in ax.containers:
    ax.bar_label(bars)
```




```
# plotting a bar chart for gender vs total amount

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

sns.barplot(x='Gender', y='Amount', hue='Gender', data=sales_gen,
palette='viridis', dodge=False, legend=False)
```

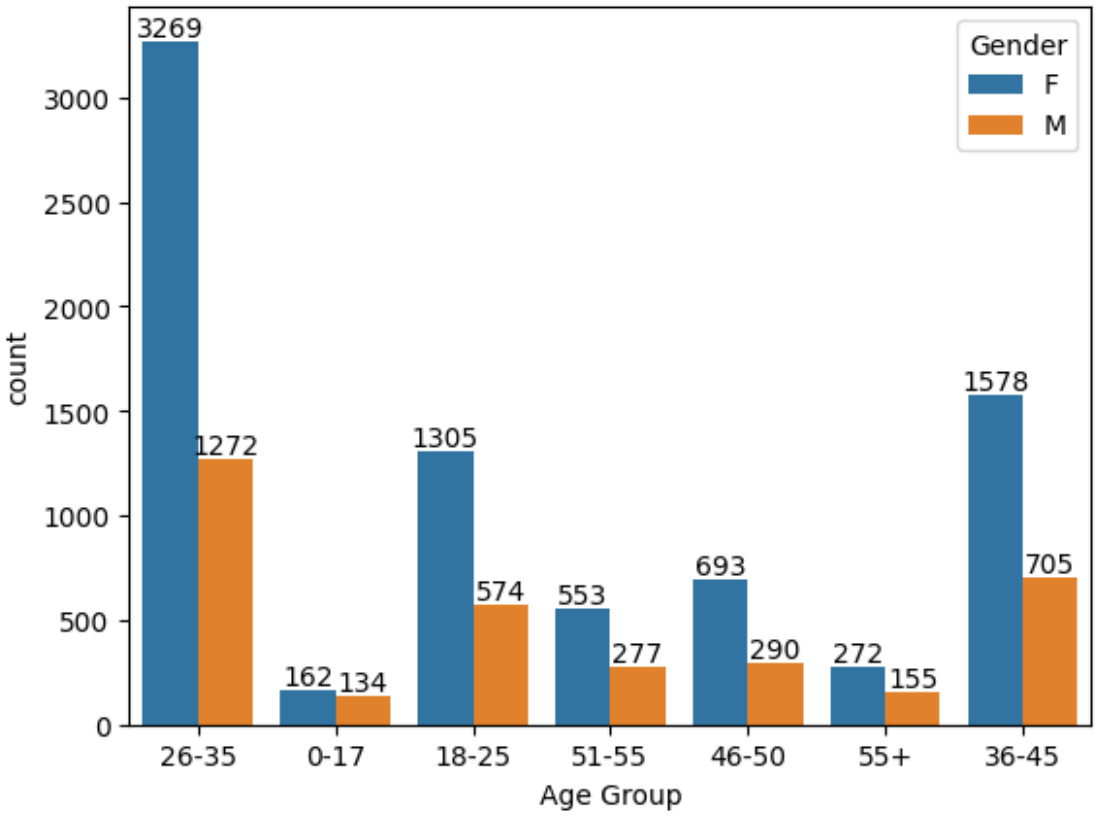
```
<Axes: xlabel='Gender', ylabel='Amount'>
```



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

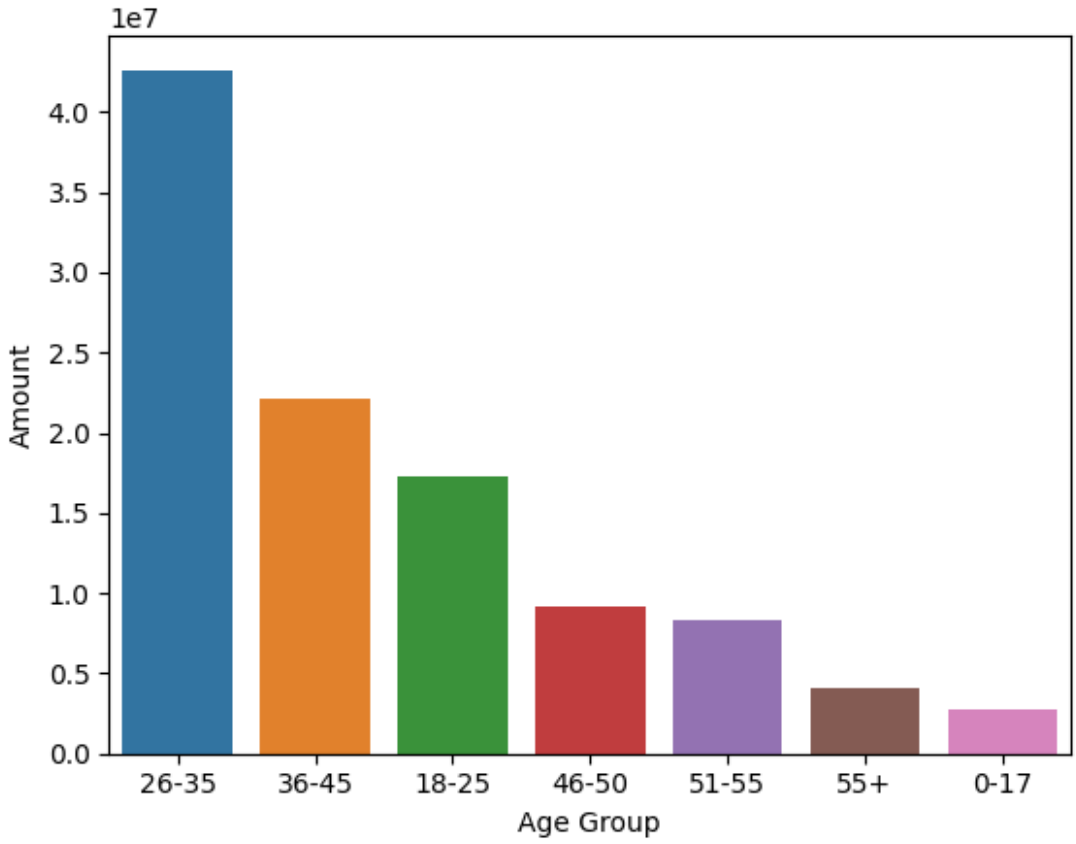
Age

```
ax = sns.countplot(data = df, x = 'Age Group', hue = 'Gender')

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




```
# Total Amount vs Age Group
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,
    hue='Age Group',
    palette=['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd',
'#8c564b', '#e377c2'],
    legend=False  # Adjust legend display as needed
)
```

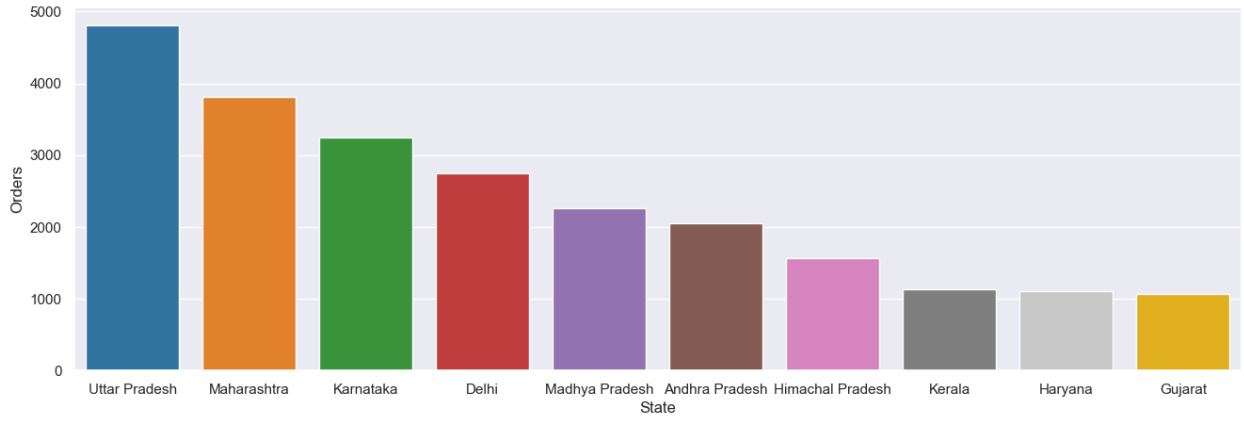
```
<Axes: xlabel='Age Group', ylabel='Amount'>
```



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

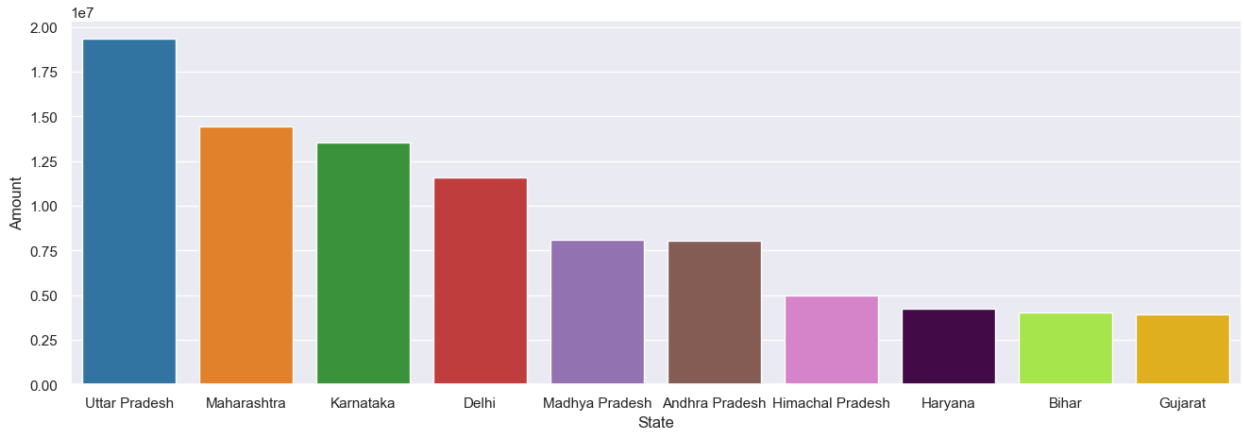
State

```
# total number of orders from top 10 states

sales_state = df.groupby(['State'], as_index=False)
['Orders'].sum().sort_values(by='Orders', ascending=False).head(10)

sns.set(rc={'figure.figsize':(16,5)})
sns.barplot(
    data=sales_state,
    x='State',
    y='Orders',
    hue='State',  # Assign 'State' to 'hue' to apply colors correctly
    palette=['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd',
'#8c564b', '#e377c2', '#7f7f7f', '#c7c7c7', '#ffbf00']
)

<Axes: xlabel='State', ylabel='Orders'>
```




```
# total amount/sales from top 10 states

sales_state = df.groupby(['State'], as_index=False)
['Amount'].sum().sort_values(by='Amount', ascending=False).head(10)

sns.set(rc={'figure.figsize':(16,5)})
sns.barplot(
    data=sales_state,
    x='State',
    y='Amount',
    hue='State',  # Assign 'State' to 'hue' for color application
    palette=['#1f77b4', '#ff7f0e', '#2ca02c', '#d62728', '#9467bd',
'#8d564b', '#e377d2', '#4b0052', '#aaff32', '#ffbf00']
  # Changed palette to 'viridis' (you can change it to any other
palette like 'Blues', 'coolwarm', etc.)
)
```

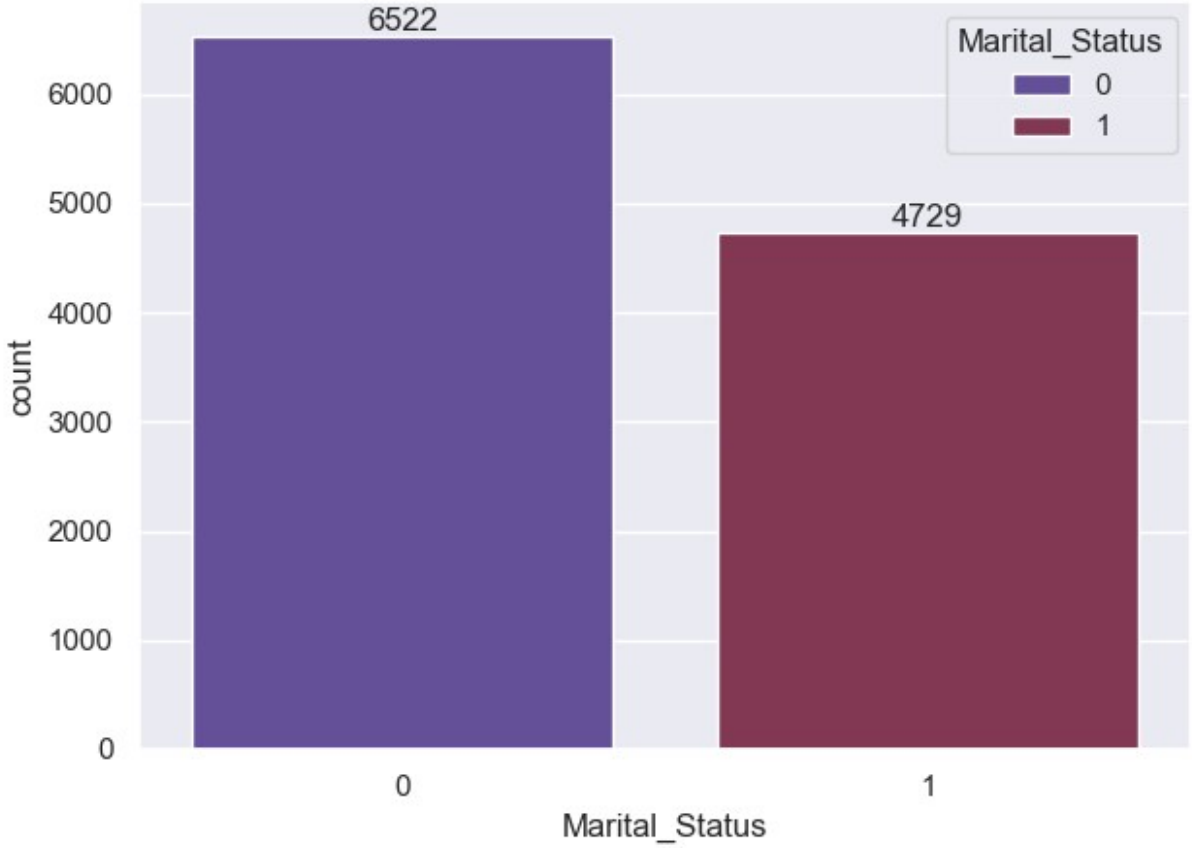
```
<Axes: xlabel='State', ylabel='Amount'>
```



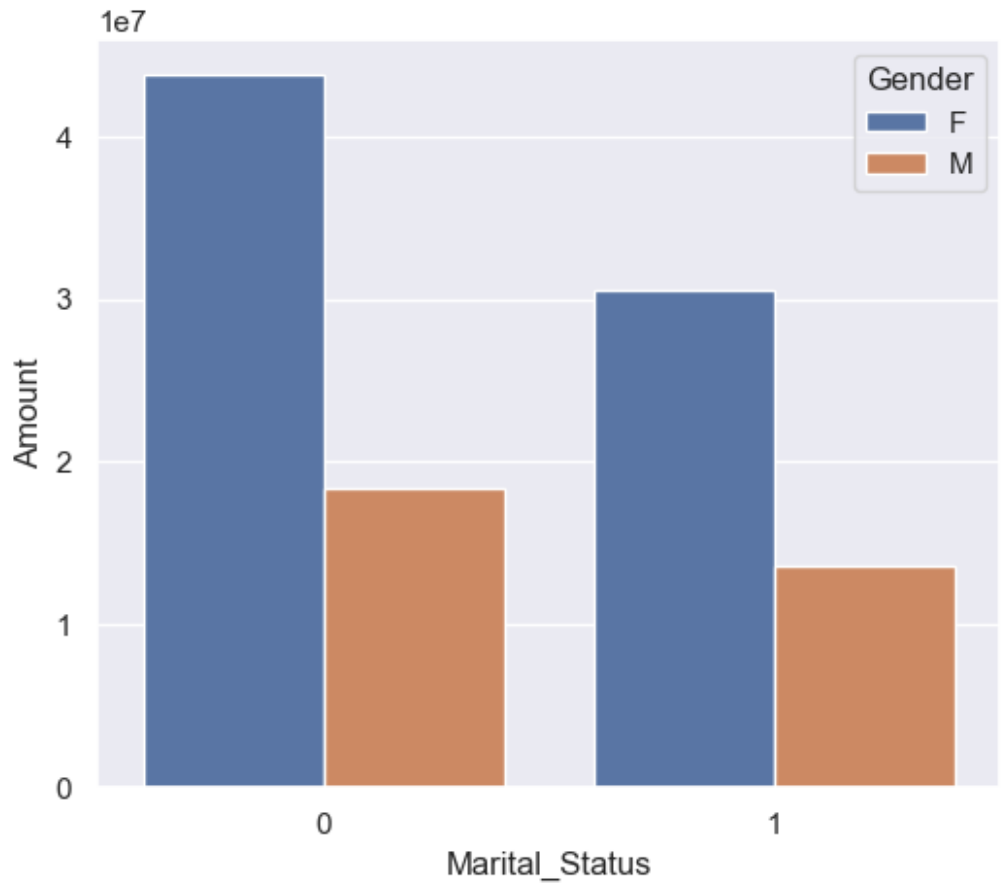
From above graphs we can see that most of the orders & total sales/amount are from Uttar Pradesh, Maharashtra and Karnataka respectively

Marital Status

```
ax = sns.countplot(data=df, x='Marital_Status',hue='Marital_Status',
palette='twilight')  # Change 'Set2' to any palette or list of colors

sns.set(rc={'figure.figsize':(7,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



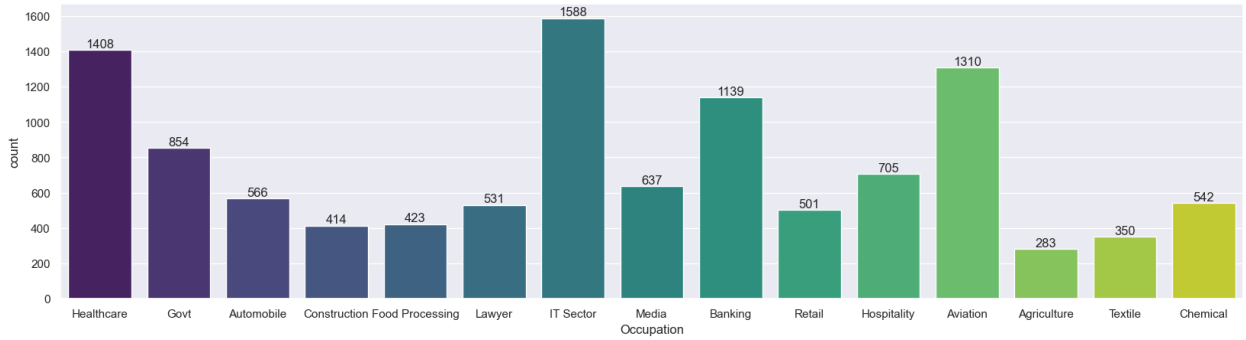

```
sales_state = df.groupby(['Marital_Status', 'Gender'], as_index=False)
['Amount'].sum().sort_values(by='Amount', ascending=False)

sns.set(rc={'figure.figsize':(6,5)})
sns.barplot(data = sales_state, x = 'Marital_Status',y= 'Amount',
hue='Gender')

<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

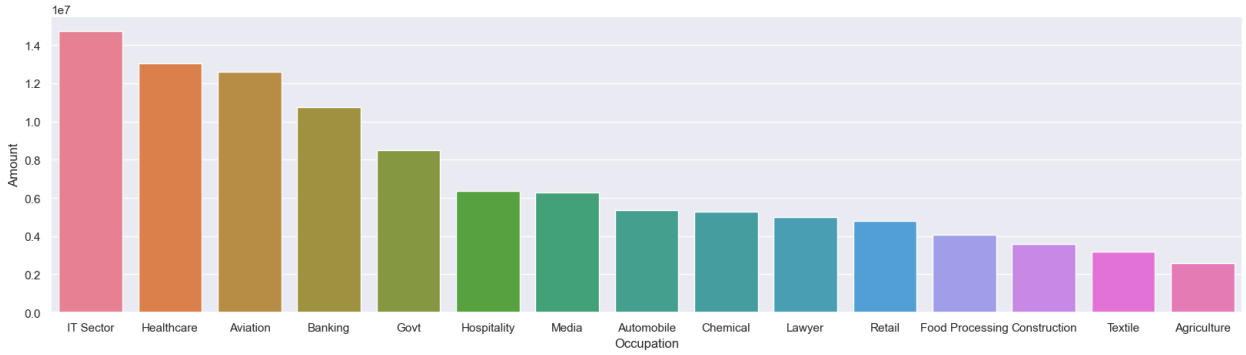
Occupation

```
sns.set(rc={'figure.figsize':(20,5)})
ax = sns.countplot(data=df, x='Occupation', hue='Occupation',
palette='viridis')  # Correct hue name and color palette

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



```
sales_state = df.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',hue='Occupation')
```

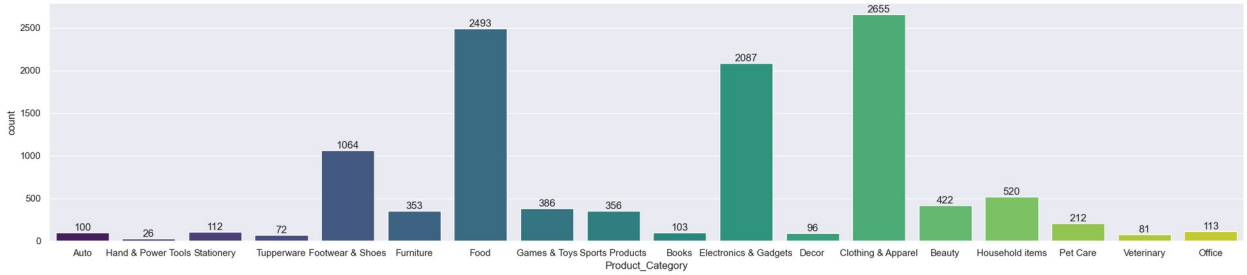
```
<Axes: xlabel='Occupation', ylabel='Amount'>
```



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

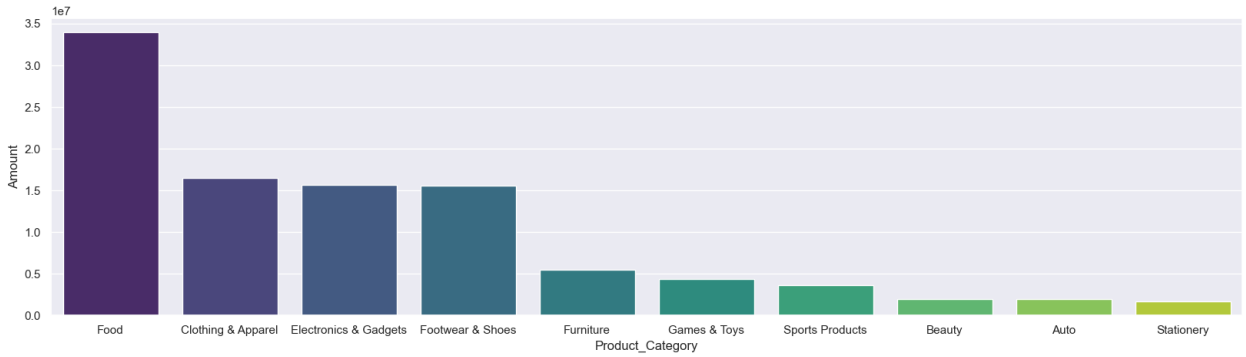
Product Category

```
sns.set(rc={'figure.figsize':(25,5)})
ax = sns.countplot(data=df, x='Product_Category',
hue='Product_Category', palette='viridis', legend=False)

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




```
sales_state = df.groupby(['Product_Category'], as_index=False)
['Amount'].sum().sort_values(by='Amount', ascending=False).head(10)

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Product_Category',y=
'Amount' ,hue='Product_Category', palette='viridis', legend=False)
```

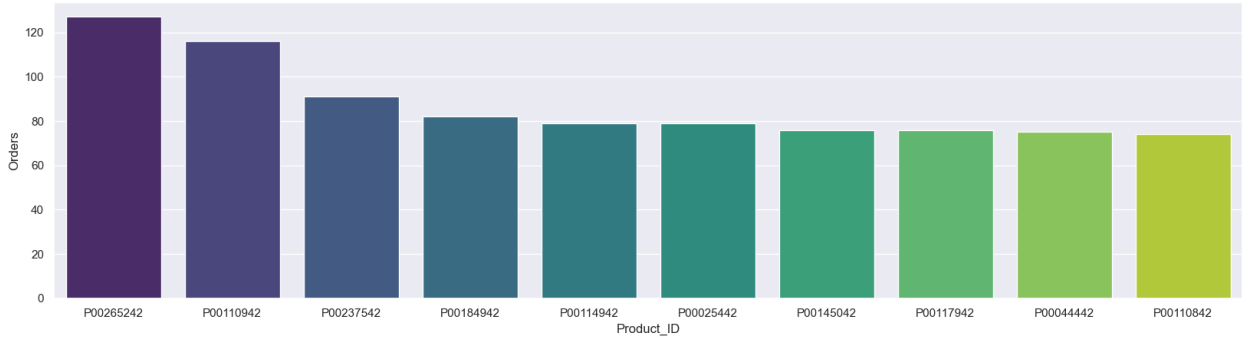
```
<Axes: xlabel='Product_Category', ylabel='Amount'>
```



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

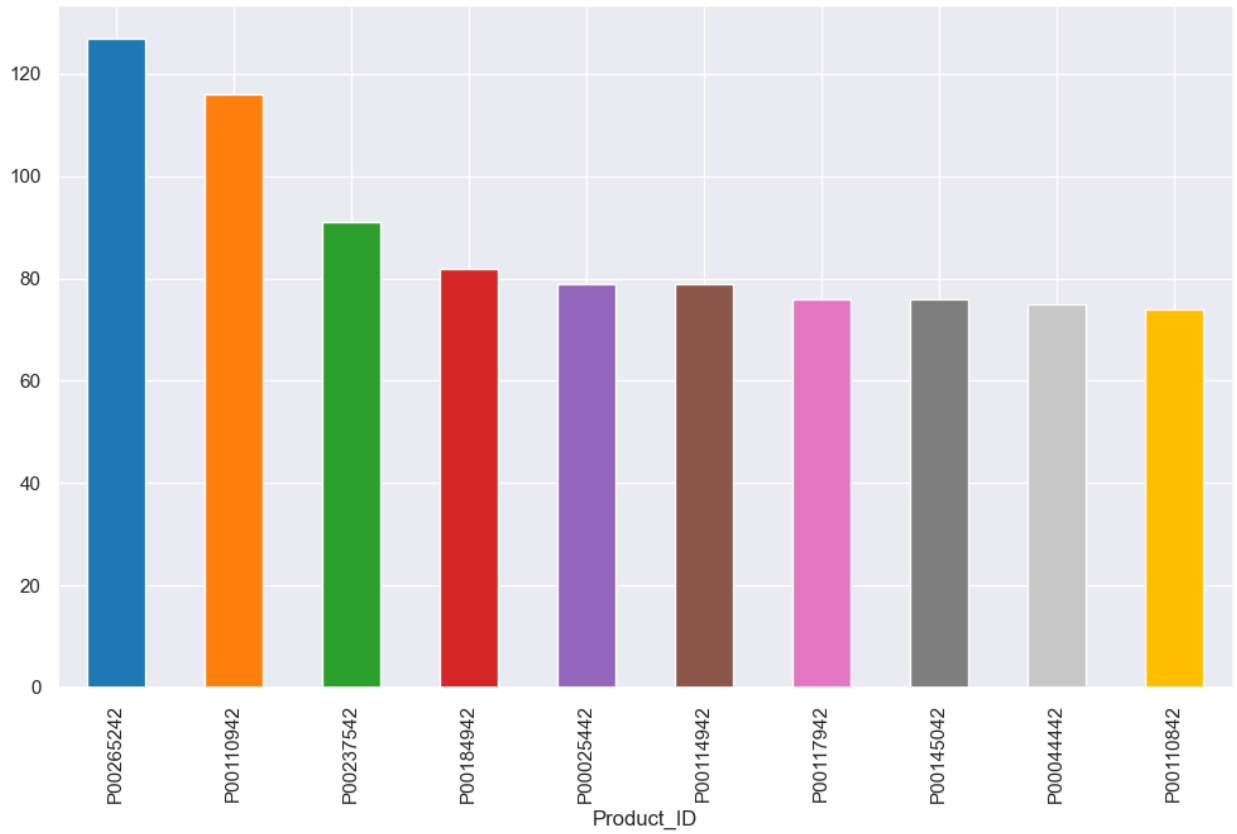
```
sales_state = df.groupby(['Product_ID'], as_index=False)
['Orders'].sum().sort_values(by='Orders', ascending=False).head(10)

sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data = sales_state, x = 'Product_ID', y =
'Orders',hue='Product_ID', palette='viridis')  # You can change
'coolwarm' to any other palette
```

```
<Axes: xlabel='Product_ID', ylabel='Orders'>
```




```
# Grouping and sorting data for top 10 most sold products
top_10_products = df.groupby('Product_ID')
['Orders'].sum().nlargest(10).sort_values(ascending=False)

# Create the figure and axis
fig1, ax1 = plt.subplots(figsize=(12,7))

# Plot the bar chart with custom colors
top_10_products.plot(kind='bar', color=['#1f77b4', '#ff7f0e',
'#2ca02c', '#d62728', '#9467bd', '#8c564b', '#e377c2', '#7f7f7f',
'#c7c7c7', '#ffbf00'], ax=ax1)
```

```
<Axes: xlabel='Product_ID'>
```



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