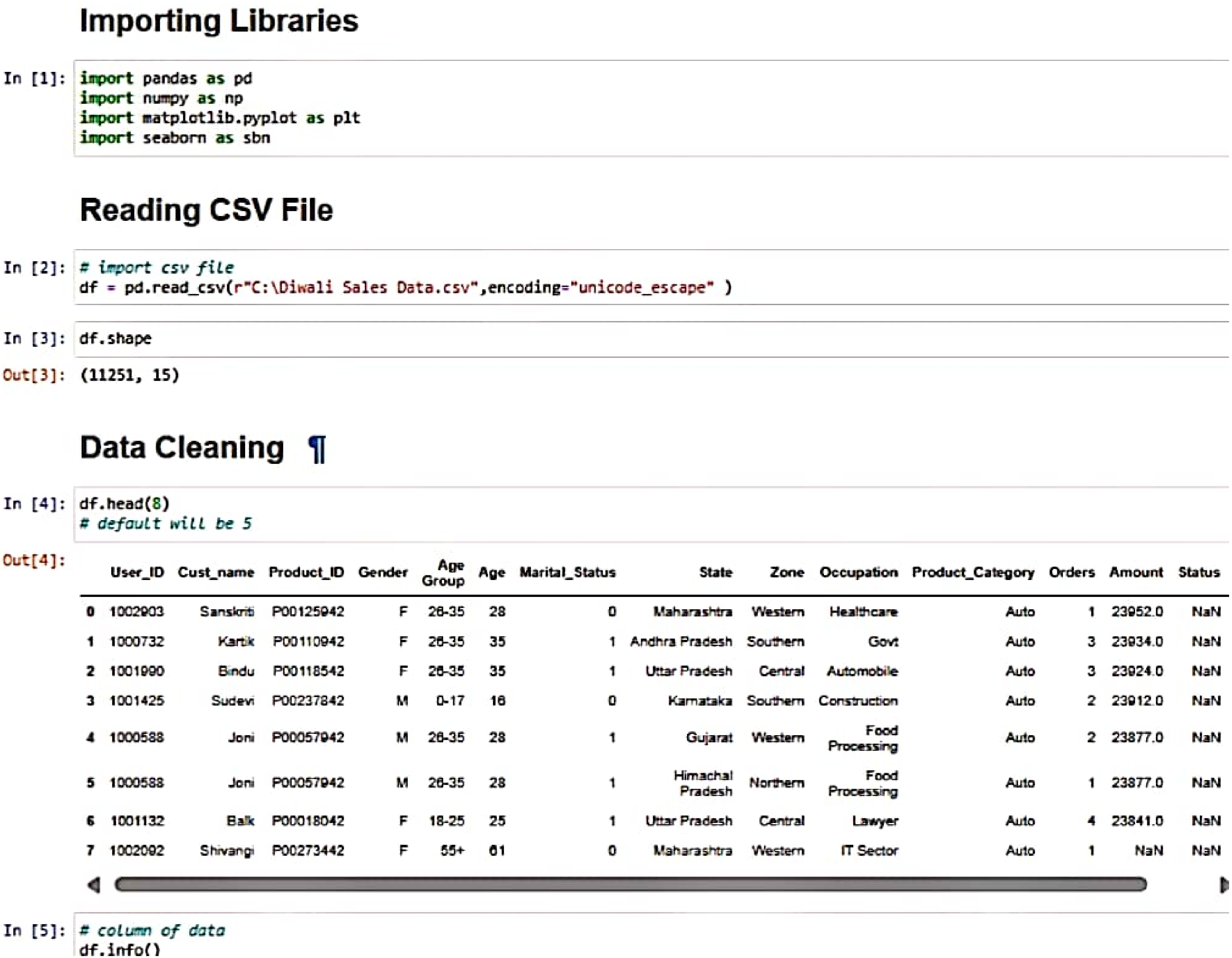


## Importing Libraries

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

## Reading CSV File

```
In [2]: # import csv file
        df = pd.read_csv(r"C:\Diwali Sales Data.csv",encoding="unicode_escape" )
```

```
In [3]: df.shape

Out[3]: (11251, 15)
```

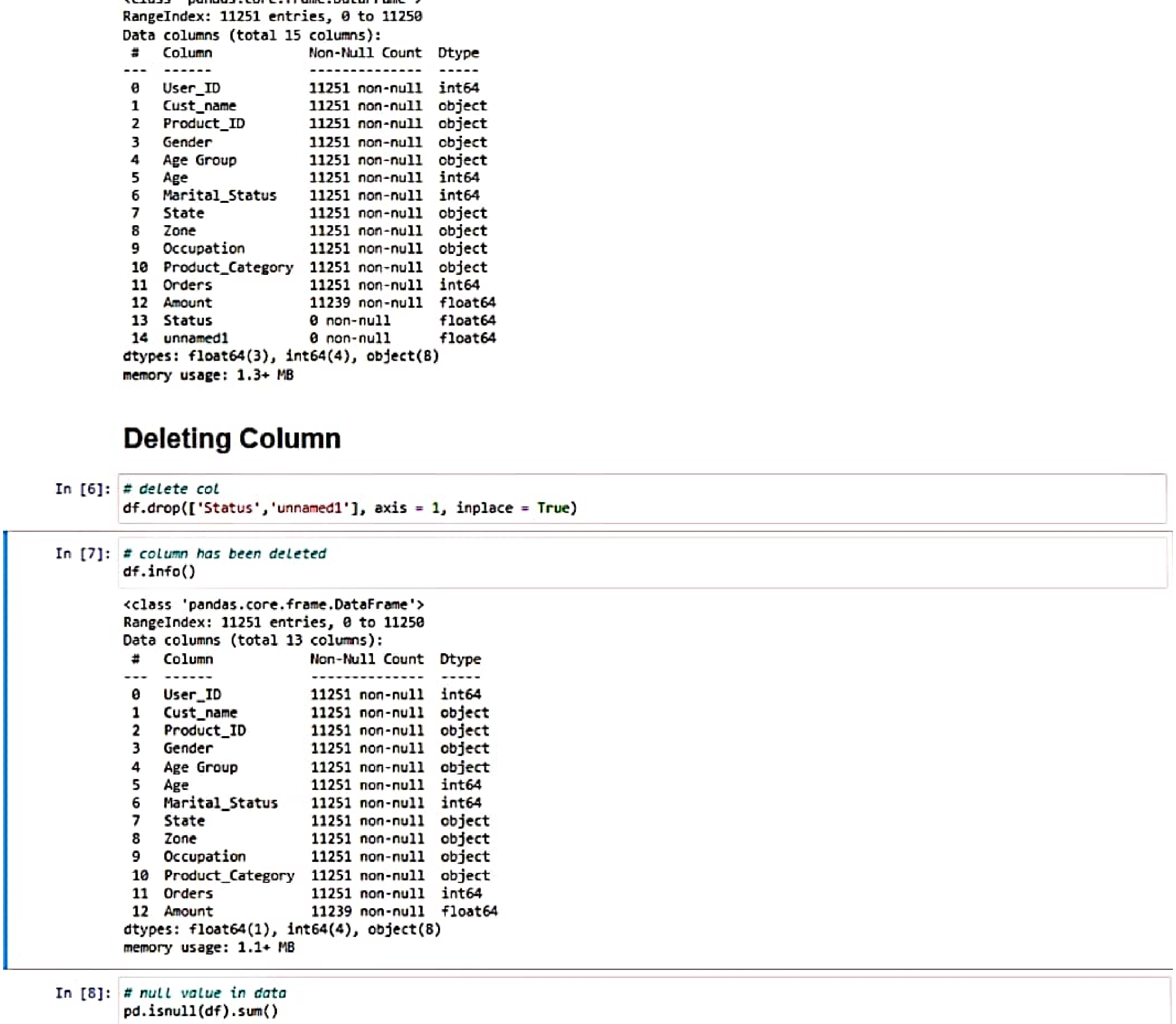
## Data Cleaning ¶

```
In [4]: df.head(8)
        # default will be 5
```

Out[4]:

| | User_ID | Cust_name | Product_ID | Gender | Age Group | Age | Marital_Status | State | Zone | Occupation | Product_Category | Orders | Amount | Status |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1002903 | Sanskriti | P00125942 | F | 26-35 | 28 | 0 | Maharashtra | Western | Healthcare | Auto | 1 | 23952.0 | NaN |
| 1 | 1000732 | Kartik | P00110942 | F | 26-35 | 35 | 1 | Andhra Pradesh | Southern | Govt | Auto | 3 | 23934.0 | NaN |
| 2 | 1001990 | Bindu | P00118542 | F | 26-35 | 35 | 1 | Uttar Pradesh | Central | Automobile | Auto | 3 | 23924.0 | NaN |
| 3 | 1001425 | Sudevi | P00237842 | M | 0-17 | 16 | 0 | Karnataka | Southern | Construction | Auto | 2 | 23912.0 | NaN |
| 4 | 1000588 | Joni | P00057942 | M | 26-35 | 28 | 1 | Gujarat | Western | Food Processing | Auto | 2 | 23877.0 | NaN |
| 5 | 1000588 | Joni | P00057942 | M | 26-35 | 28 | 1 | Himachal Pradesh | Northern | Food Processing | Auto | 1 | 23877.0 | NaN |
| 6 | 1001132 | Balk | P00018042 | F | 18-25 | 25 | 1 | Uttar Pradesh | Central | Lawyer | Auto | 4 | 23841.0 | NaN |
| 7 | 1002092 | Shivangi | P00273442 | F | 55+ | 61 | 0 | Maharashtra | Western | IT Sector | Auto | 1 | NaN | NaN |

```
In [5]: # column of data
        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
```

## Deleting Column

```
In [6]: # delete col
        df.drop(['Status','unnamed1'], axis = 1, inplace = True)
```

```
In [7]: # column has been deleted
        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 [8]: # null value in data
        pd.isnull(df).sum()
```

```
Out[8]: 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
```

## Deleting null data

```
In [9]: # delete null
        df.dropna(inplace=True)
```

```
In [10]: df.shape
         # earlier it was (11251, 15)
         # 12 null values has been deleted

Out[10]: (11239, 13)
```

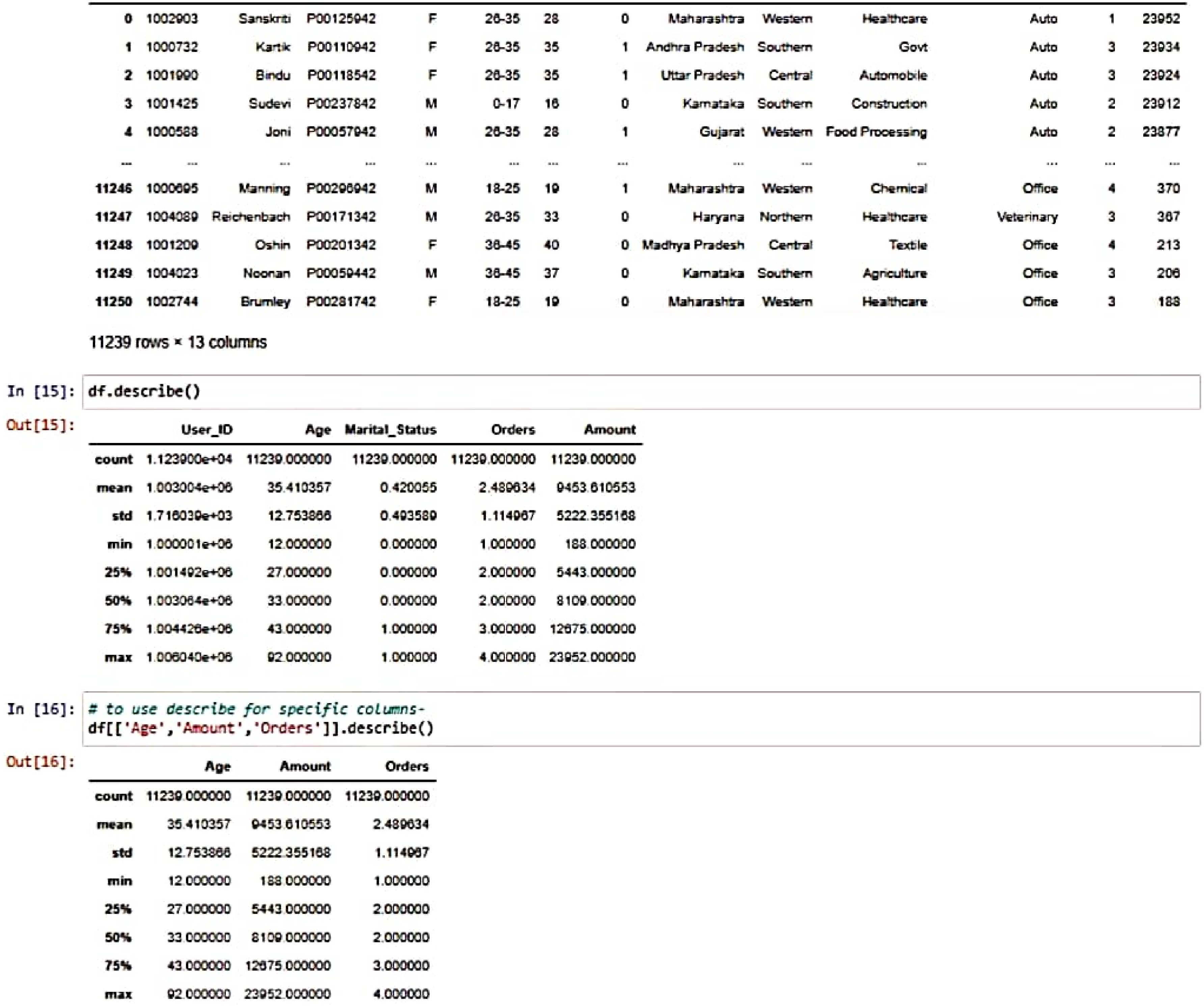
```
In [11]: # changing datatype
         df['Amount'] = df['Amount'].astype('int')
```

```
In [12]: df['Amount'].dtypes

Out[12]: dtype('int32')
```

```
In [13]: df.columns

Out[13]: Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
               'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
               'Orders', 'Amount'],
              dtype='object')
```

## Renaming

```
In [14]: 
         # rename
         df.rename(columns={'Marital_Status': 'Marriage'},inplace=False)
```



```
0      1002903  Sanskriti     P00125942  F  26-35  28  0  Maharashtra     Western   Healthcare       Auto        1  23952
1      1000732  Kartik        P00110942  F  26-35  35  1  Andhra Pradesh  Southern  Govt             Auto        3  23934
2      1001990  Bindu         P00118542  F  26-35  35  1  Uttar Pradesh   Central   Automobile       Auto        3  23924
3      1001425  Sudevi        P00237842  M  0-17   16  0  Karnataka       Southern  Construction     Auto        2  23912
4      1000588  Joni          P00057942  M  26-35  28  1  Gujarat         Western   Food Processing  Auto        2  23877
...    ...      ...           ...        ...  ...  ...  ...  ...          ...       ...              ...         ... ...
11246  1000695  Manning       P00296942  M  18-25  19  1  Maharashtra     Western   Chemical         Office      4  370
11247  1004089  Reichenbach   P00171342  M  26-35  33  0  Haryana         Northern  Healthcare       Veterinary  3  367
11248  1001209  Oshin         P00201342  F  36-45  40  0  Madhya Pradesh  Central   Textile          Office      4  213
11249  1004023  Noonan        P00059442  M  36-45  37  0  Karnataka       Southern  Agriculture      Office      3  206
11250  1002744  Brumley       P00281742  F  18-25  19  0  Maharashtra     Western   Healthcare       Office      3  188

11239 rows × 13 columns
```

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

```
Out[15]:
              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
```

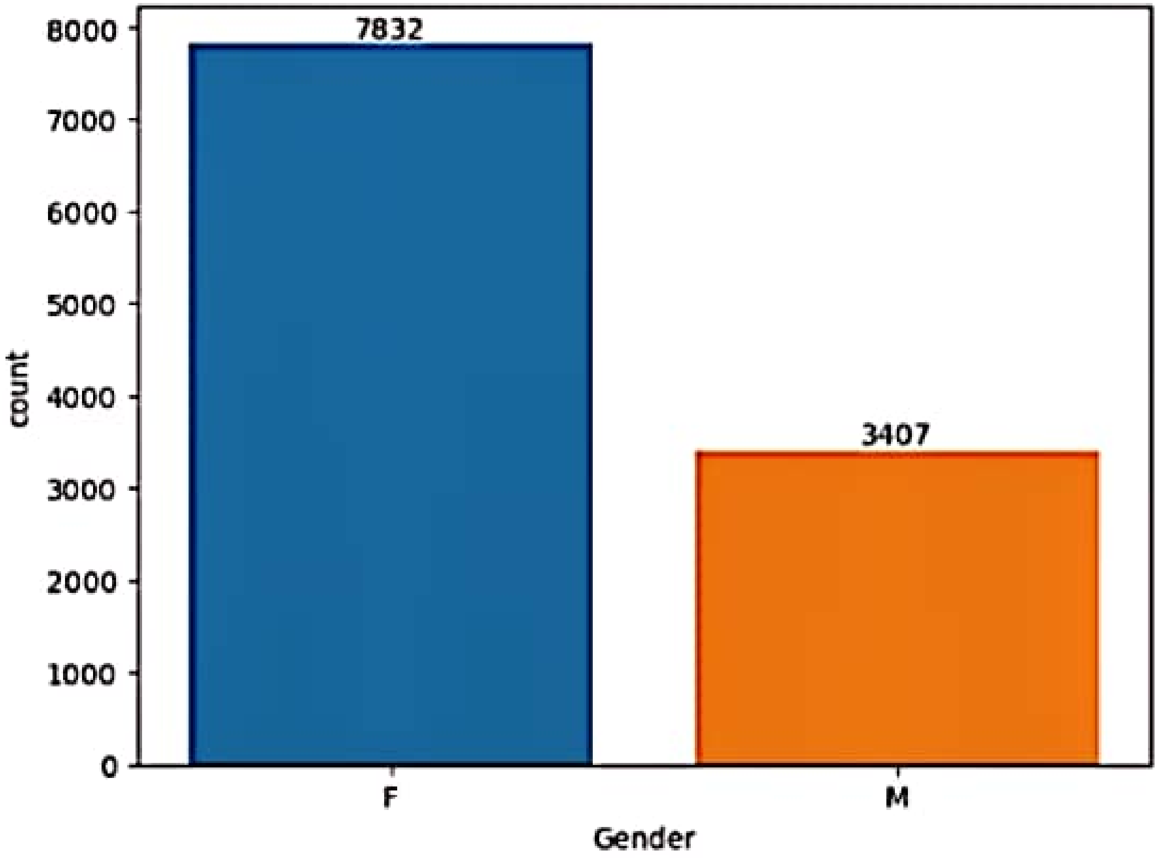
```
In [16]: # to use describe for specific columns-
         df[['Age','Amount','Orders']].describe()
```

```
Out[16]:
                Age        Amount        Orders
count  11239.000000  11239.000000  11239.000000
mean      35.410357   9453.610553      2.489634
std       12.753866   5222.355168      1.114967
min       12.000000    188.000000      1.000000
25%       27.000000   5443.000000      2.000000
50%       33.000000   8109.000000      2.000000
75%       43.000000  12675.000000      3.000000
max       92.000000  23952.000000      4.000000
```

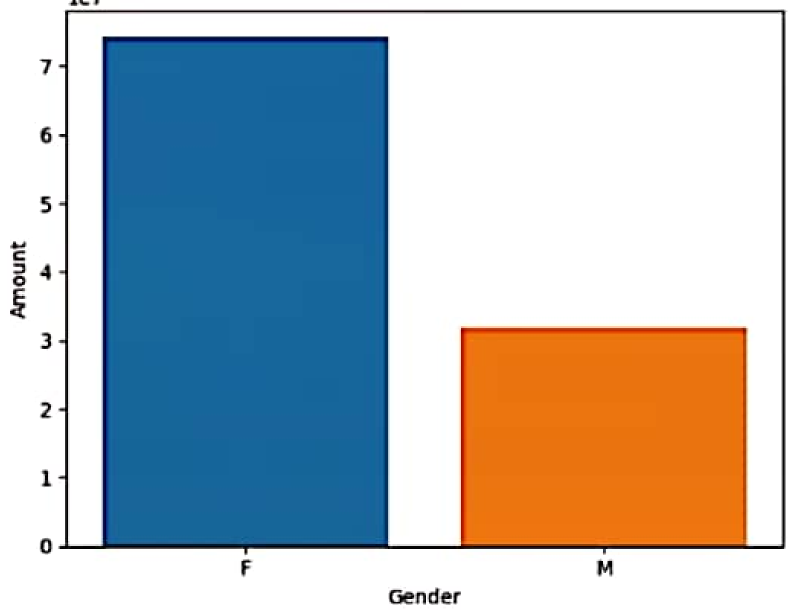
# EDA/ EXPLORATORY DATA ANALYSIS

## Gender

```
In [17]: v = sbn.countplot(x ='Gender',data = df)
         for i in v.containers:
             v.bar_label(i)
```



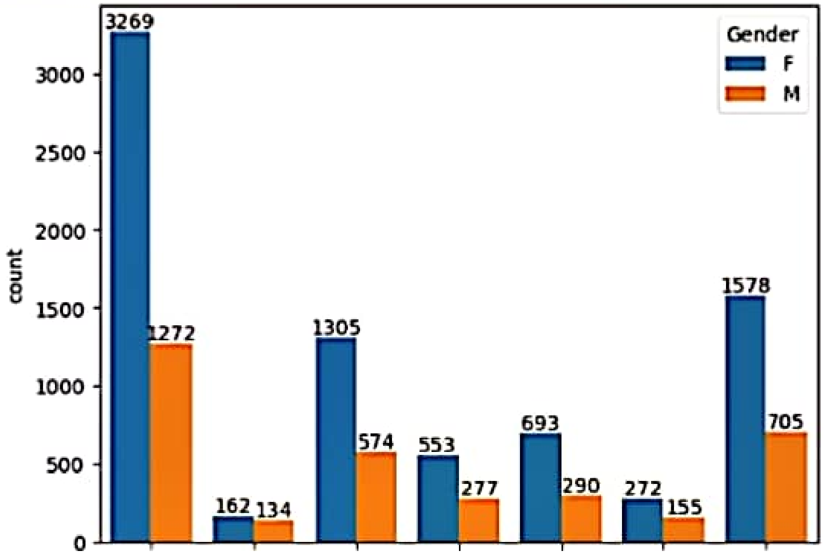
```
In [18]: s= df.groupby(['Gender'],as_index=False)['Amount'].sum().sort_values(by= 'Amount', ascending = False)
         sbn.barplot(x='Gender' , y = 'Amount' , data = s)

Out[18]: <AxesSubplot:xlabel='Gender', ylabel='Amount'>
```

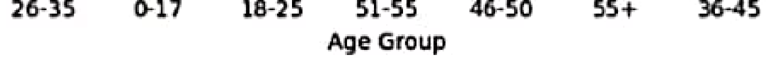


From above graphs we can see that the most of the buyers are females and even the purchasing power of females are greater than men

## Age

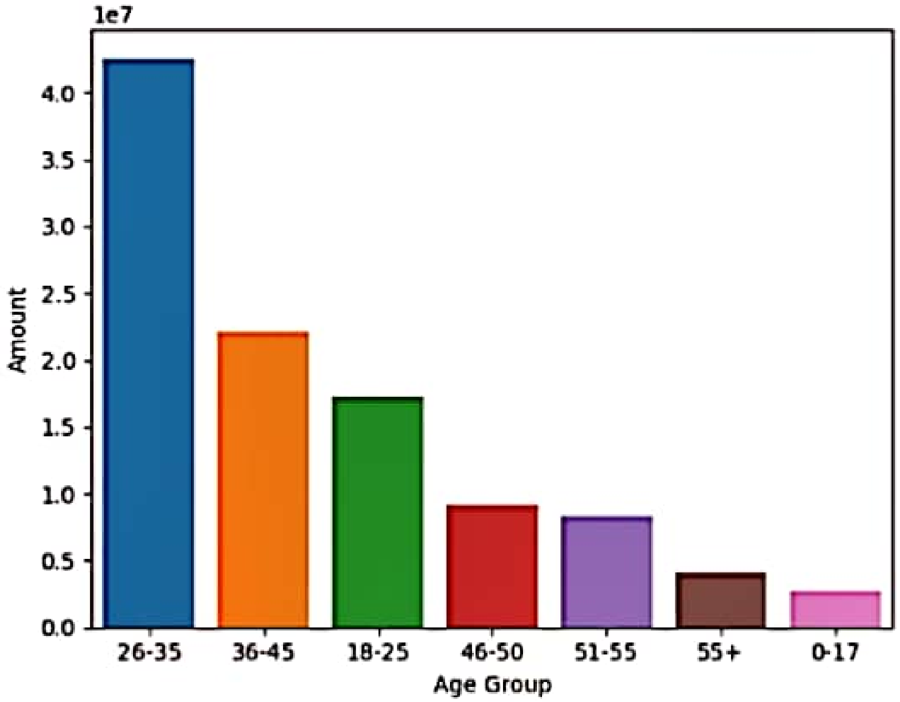
```
In [19]: v = sbn.countplot(data= df ,x ='Age Group', hue = 'Gender')
         for i in v.containers:
             v.bar_label(i)
```





```
In [20]: # total amount vs age group
         s= df.groupby(['Age Group'],as_index=False)['Amount'].sum().sort_values(by= 'Amount', ascending = False)
         sbn.barplot(x='Age Group' , y = 'Amount' , data = s)
```

```
Out[20]: <AxesSubplot:xlabel='Age Group', ylabel='Amount'>
```

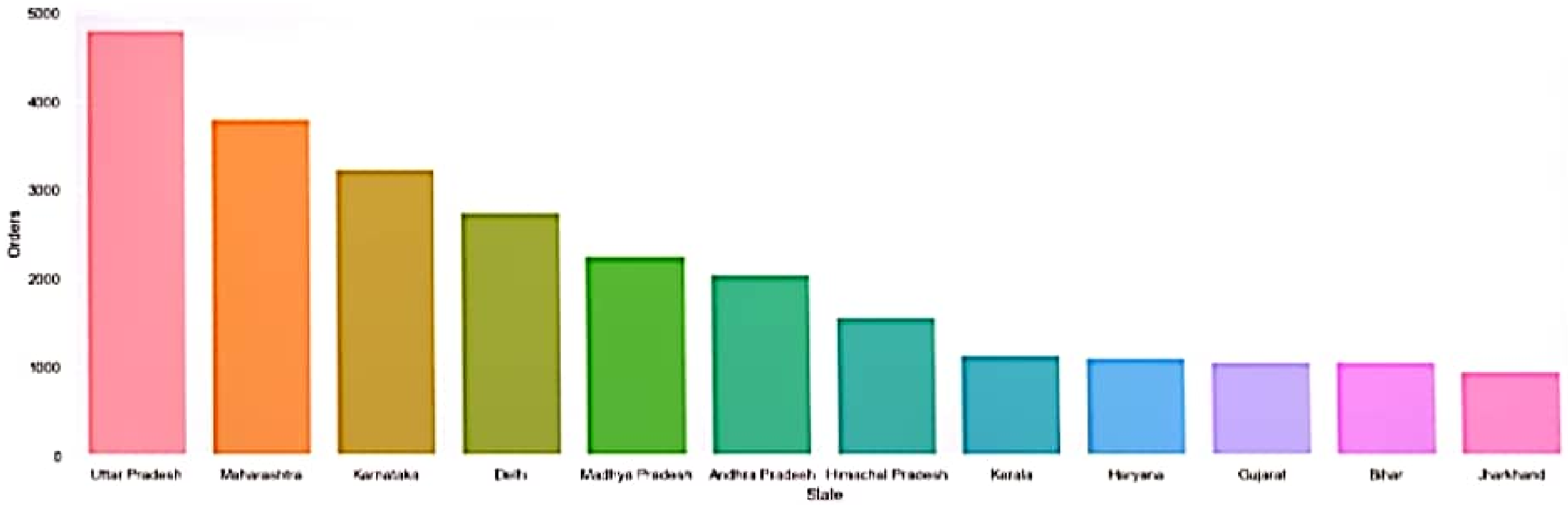


```
From above graphs we can see that most of the buyers are of age group between 26-35 years female
```

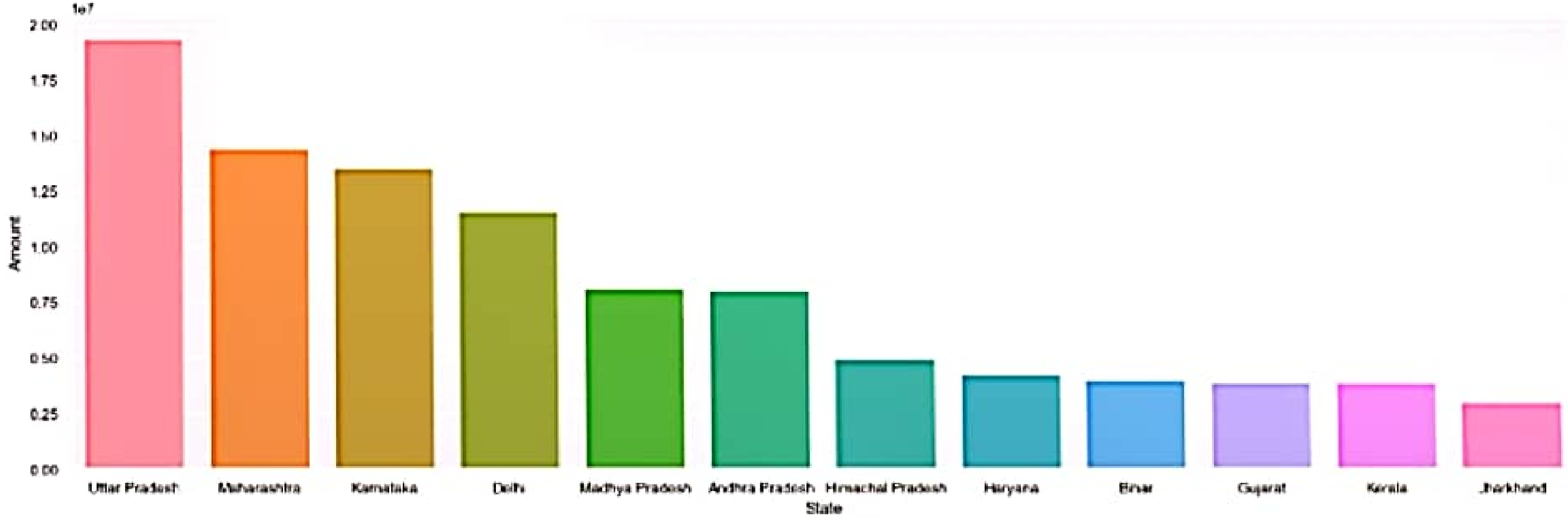
## State

```
In [21]:
         # total number of orders from top 12 states
         s= df.groupby(['State'],as_index=False)['Orders'].sum().sort_values(by= 'Orders', ascending = False).head(12)

         sbn.set(rc={'figure.figsize':(20,6)})

         sbn.barplot(x='State' , y = 'Orders' , data = s)
```

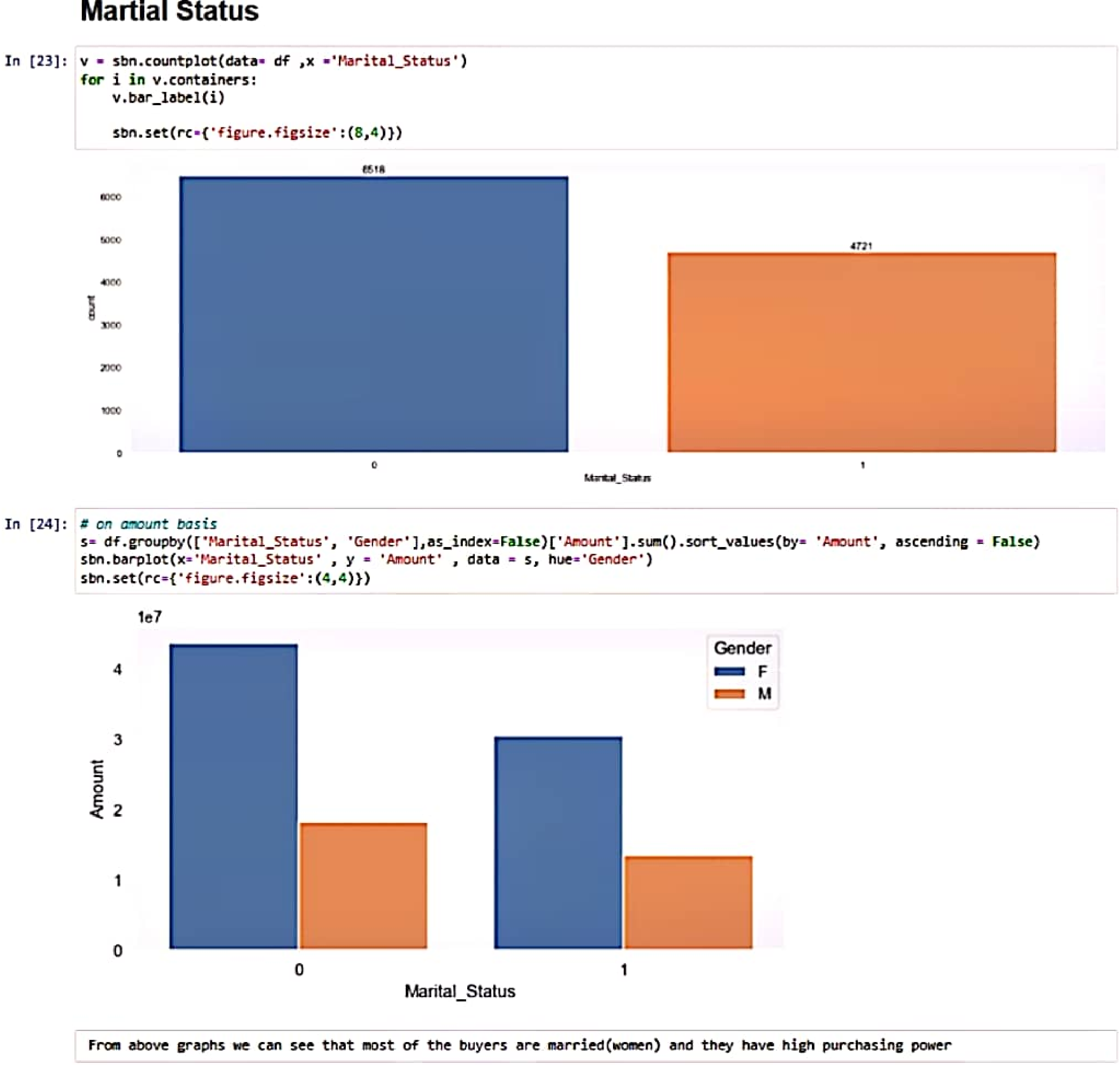
```
Out[21]: <AxesSubplot:xlabel='State', ylabel='Orders'>
```



```
In [22]: # total amount from top 12 states
         s= df.groupby(['State'],as_index=False)['Amount'].sum().sort_values(by= 'Amount', ascending = False).head(12)

         sbn.set(rc={'figure.figsize':(20,6)})

         sbn.barplot(x='State' , y = 'Amount' , data = s)

Out[22]: <AxesSubplot:xlabel='State', ylabel='Amount'>
```



```
From above graphs we can see that most of the orders & total sales/amount are from uttar pradesh, maharashtra
 and karnatka respectively
```



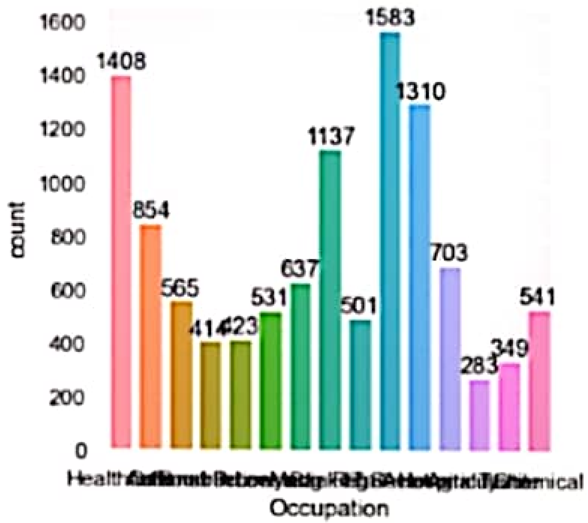
## Martial Status

```
In [23]: v = sbn.countplot(data= df ,x ='Marital_Status')
         for i in v.containers:
             v.bar_label(i)

             sbn.set(rc={'figure.figsize':(8,4)})
```

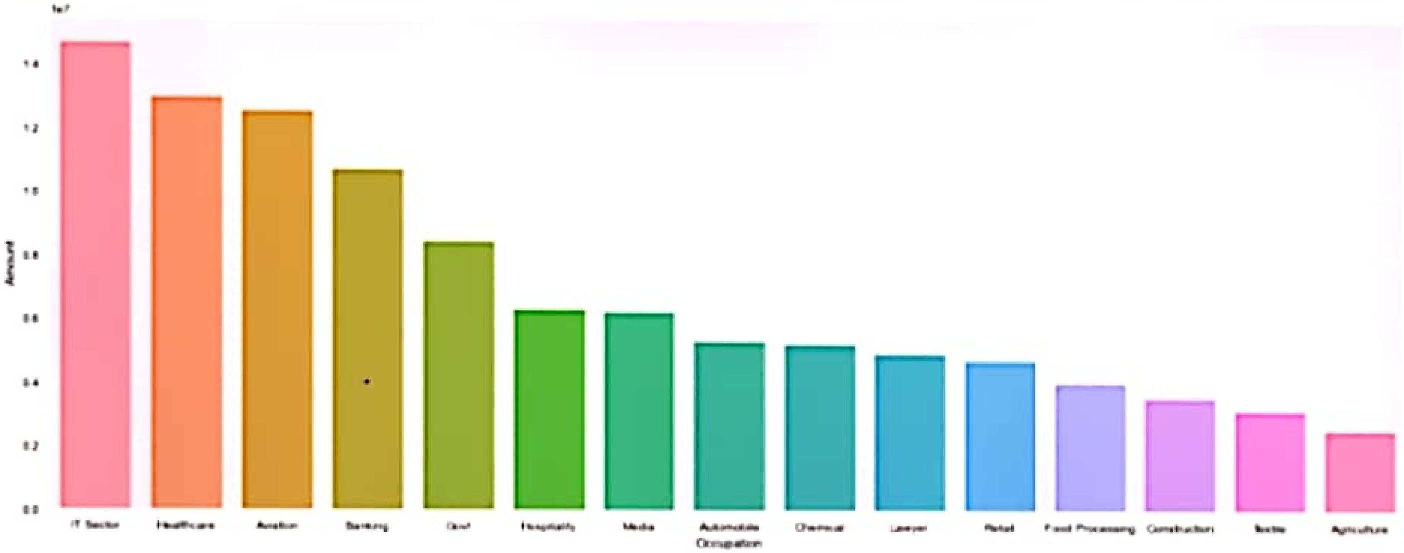
```
In [24]: # on amount basis
         s= df.groupby(['Marital_Status', 'Gender'],as_index=False)['Amount'].sum().sort_values(by= 'Amount', ascending = False)
         sbn.barplot(x='Marital_Status' , y = 'Amount' , data = s, hue='Gender')
         sbn.set(rc={'figure.figsize':(4,4)})
```

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

In [25]:
```
v = sbn.countplot(data= df ,x ='Occupation')
for i in v.containers:
    v.bar_label(i)

    sbn.set(rc={'figure.figsize':(22,8)})
```



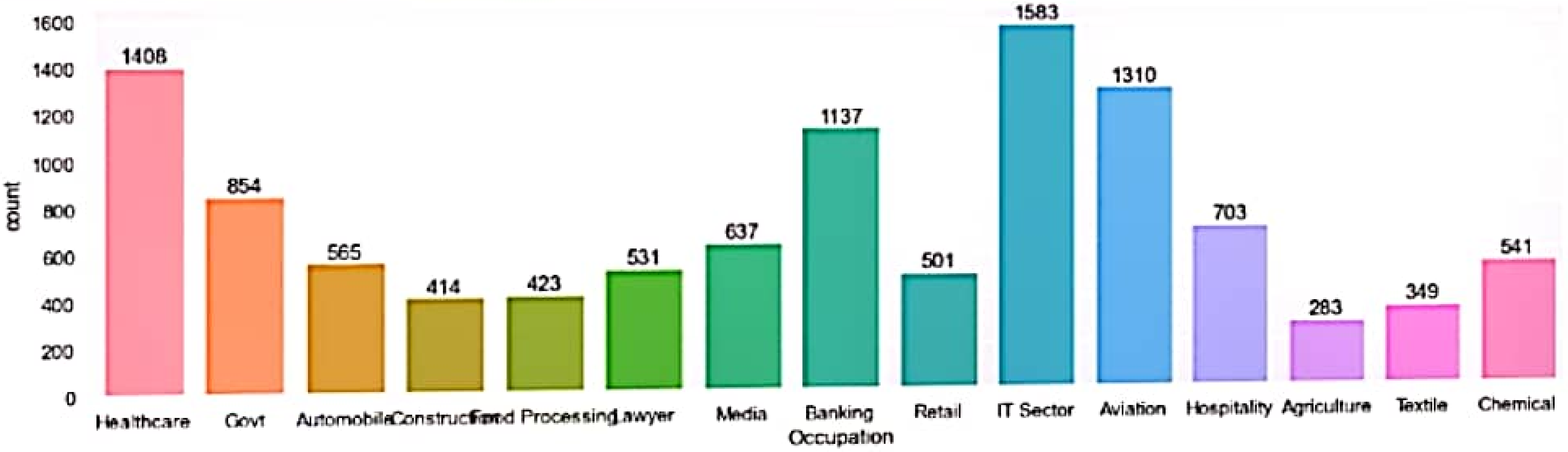
In [26]:
```
s= df.groupby(['Occupation'],as_index=False)['Amount'].sum().sort_values(by= 'Amount', ascending = False)
sbn.barplot(x='Occupation' , y = 'Amount' , data = s)
sbn.set(rc={'figure.figsize':(15,4)})
```



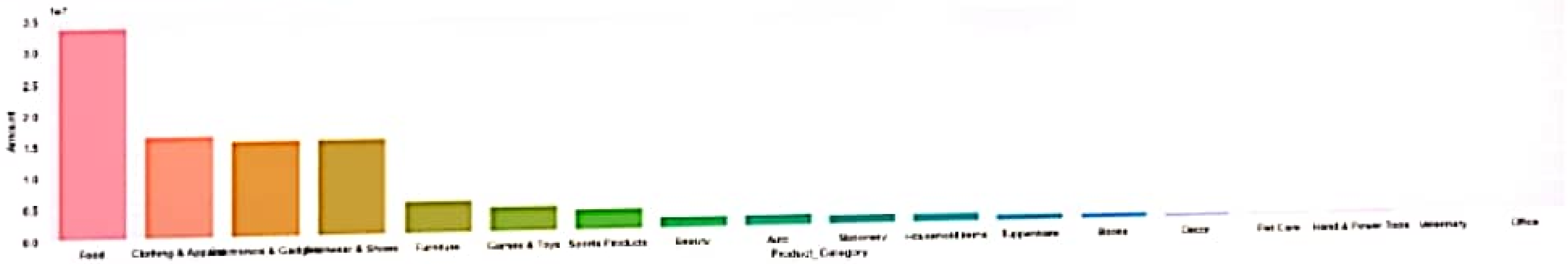
From above graphs we can see that most of the buyers are working in IT , Healthcare and aviation sector

## product category

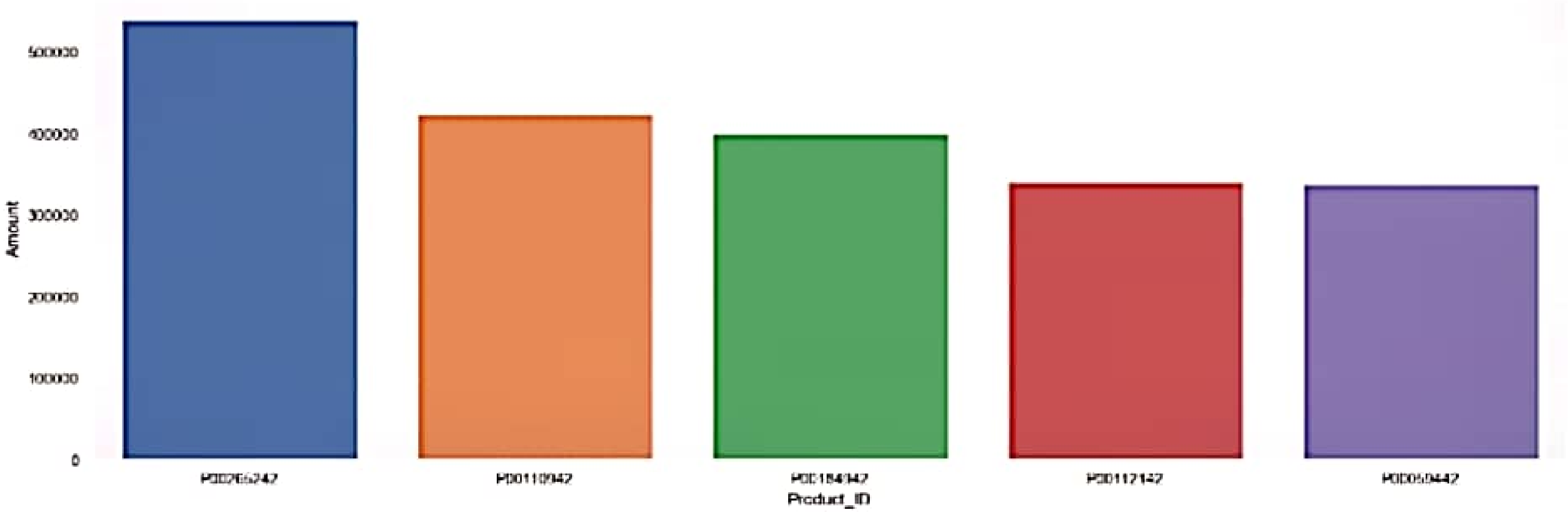
```
In [27]: v = sbn.countplot(data= df ,x ='Occupation')
         for i in v.containers:
             v.bar_label(i)

             sbn.set(rc={'figure.figsize':(27,4)})
```



```
In [28]: s= df.groupby(['Product_Category'],as_index=False)['Amount'].sum().sort_values(by= 'Amount', ascending = False)
         sbn.barplot(x='Product_Category' , y = 'Amount' , data = s)
         sbn.set(rc={'figure.figsize':(19,6)})
```



```
In [29]: # top 5 product sold
         s= df.groupby(['Product_ID'],as_index=False)['Amount'].sum().sort_values(by= 'Amount', ascending = False).head(5)
         sbn.barplot(x='Product_ID' , y = 'Amount' , data = s)
         sbn.set(rc={'figure.figsize':(15,8)})
```



# conclusion

```
From the above insights i can say that the married woman age group of 26-35 years from up,maharashtra and karnatka working in
IT,healthcare and aviation are more likely to buy products from food , clothing and electronics category.
```

```
GITHUB LINK:- https://github.com/KESHAV2006
```

```
  THANK YOU:)
```