

```
#import python libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt #visualizing data
%matplotlib inline
import seaborn as sns

import plotly.graph_objects as go
import plotly.express as px

#import csv file
df = pd.read_csv('Sales Data.csv',encoding = 'unicode_escape')

df.shape

(11251, 15)

print(df)
```

	User_ID	Cust_name	Product_ID	Gender	Age	Group	Age
0	1002903	Sanskriti	P00125942	F	26-35	28	
1	1000732	Kartik	P00110942	F	26-35	35	
2	1001990	Bindu	P00118542	F	26-35	35	
3	1001425	Sudevi	P00237842	M	0-17	16	
4	1000588	Joni	P00057942	M	26-35	28	
...
11246	1000695	Manning	P00296942	M	18-25	19	
11247	1004089	Reichenbach	P00171342	M	26-35	33	
11248	1001209	Oshin	P00201342	F	36-45	40	
11249	1004023	Noonan	P00059442	M	36-45	37	
11250	1002744	Brumley	P00281742	F	18-25	19	

	State	Zone	Occupation	Product_Category
0	Maharashtra	Western	Healthcare	Auto
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  Status  unnamed1
0      23952.0     NaN      NaN
1      23934.0     NaN      NaN
2      23924.0     NaN      NaN
3      23912.0     NaN      NaN
4      23877.0     NaN      NaN
...      ...      ...      ...
11246      370.0     NaN      NaN
11247      367.0     NaN      NaN
11248      213.0     NaN      NaN
11249      206.0     NaN      NaN
11250      188.0     NaN      NaN

```

[11251 rows x 15 columns]

df.head

```

<bound method NDFrame.head of      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
...      ...      ...      ...      ...      ...
...

```

```

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      Status      unnamed1
0      23952.0      NaN      NaN
1      23934.0      NaN      NaN
2      23924.0      NaN      NaN
3      23912.0      NaN      NaN
4      23877.0      NaN      NaN
...      ...      ...      ...
11246      370.0      NaN      NaN
11247      367.0      NaN      NaN
11248      213.0      NaN      NaN
11249      206.0      NaN      NaN
11250      188.0      NaN      NaN

```

[11251 rows x 15 columns]>

```
df.head(50)
```

```
{
  "summary": {
    "name": "df",
    "rows": 11251,
    "fields": [
      {
        "column": "User_ID",
        "properties": {
          "dtype": "number",
          "std": 1716,
          "min": 1000001,
          "max": 1006040,
          "num_unique_values": 3755,
          "samples": [
            1005905,
            1003730,
            1005326
          ],
          "semantic_type": ""
        },
        "description": "",
        "column": "Cust_name",
        "properties": {
          "dtype": "category",
          "num_unique_values": 1250,
          "samples": [
            "Nida",
            "Lacy",
            "Caudle"
          ],
          "semantic_type": ""
        },
        "description": "",
        "column": "Product_ID",
        "properties": {
          "dtype": "category",
          "num_unique_values": 2351,
          "samples": [
            "P00224442",
            "P00205242",
            "P00347442"
          ],
          "semantic_type": ""
        },
        "description": "",
        "column": "Gender",
        "properties": {
          "dtype": "category",
          "num_unique_values": 2,
          "samples": [
            "M",
            "F"
          ],
          "semantic_type": ""
        },
        "description": "",
        "column": "Age Group",
        "properties": {
          "dtype": "category",
          "num_unique_values": 7,
          "samples": [
            "26-35",
            "0-17"
          ],
          "semantic_type": ""
        },
        "description": "",
        "column": "Age",
        "properties": {
          "dtype": "number",
          "std": 12,
          "min": 12,
          "max": 92,
          "num_unique_values": 81,
          "samples": [
            18,
            28
          ],
          "semantic_type": ""
        },
        "description": "",
        "column": "Marital_Status",
        "properties": {
          "dtype": "number",
          "std": 0,
          "min": 0,
          "max": 1,
          "num_unique_values": 2,
          "samples": [
            1,
            0
          ],
          "semantic_type": ""
        },
        "description": "",
        "column": "State",
        "properties": {
          "dtype": "category",
          "num_unique_values": 16,
          "samples": [
            "Maharashtra",
            "Andhra Pradesh"
          ],
          "semantic_type": ""
        },
        "description": "",
        "column": "Zone",
        "properties": {
          "dtype": "category",
          "num_unique_values": 5,
          "samples": [
            "Southern",
            "Eastern"
          ],
          "semantic_type": ""
        },
        "description": "",
        "column": "Occupation",
        "properties": {
          "dtype": "category",
          "num_unique_values": 15,
          "samples": [

```

```

\"Retail\\",\n          \"Aviation\\\"\n          ],\n\"semantic_type\": \"\", \n          \"description\": \"\"\n          }\n          },\n          {\n          \"column\": \"Product_Category\", \n          \"properties\": {\n          \"dtype\": \"category\", \n          \"num_unique_values\": 18, \n          \"samples\": [\n          \"Auto\\", \n          \"Hand & Power Tools\\\"\n          ], \n          \"semantic_type\": \"\", \n          \"description\": \"\"\n          }\n          },\n          {\n          \"column\": \"Orders\", \n          \"properties\": {\n          \"dtype\": \"number\", \n          \"std\": 1, \n          \"min\": 1, \n          \"max\": 4, \n          \"num_unique_values\": 4, \n          \"samples\": [\n          3, \n          4\n          ], \n          \"semantic_type\": \"\", \n          \"description\": \"\"\n          }\n          },\n          {\n          \"column\": \"Amount\", \n          \"properties\": {\n          \"dtype\": \"number\", \n          \"std\": 5222.355869186444, \n          \"min\": 188.0, \n          \"max\": 23952.0, \n          \"num_unique_values\": 6584, \n          \"samples\": [\n          19249.0, \n          13184.0\n          ], \n          \"semantic_type\": \"\", \n          \"description\": \"\"\n          }\n          },\n          {\n          \"column\": \"Status\", \n          \"properties\": {\n          \"dtype\": \"number\", \n          \"std\": null, \n          \"min\": null, \n          \"max\": null, \n          \"num_unique_values\": 0, \n          \"samples\": [], \n          \"semantic_type\": \"\", \n          \"description\": \"\"\n          }\n          },\n          {\n          \"column\": \"unnamed1\", \n          \"properties\": {\n          \"dtype\": \"number\", \n          \"std\": null, \n          \"min\": null, \n          \"max\": null, \n          \"num_unique_values\": 0, \n          \"samples\": [], \n          \"semantic_type\": \"\", \n          \"description\": \"\"\n          }\n          }\n          ], \n          \"type\": \"dataframe\", \"variable_name\": \"df\"}

```

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

```
df
```

```

{\"summary\": { \n  \"name\": \"df\", \n  \"rows\": 11251, \n  \"fields\": [\n    {\n      \"column\": \"User_ID\", \n      \"properties\": {\n        \"dtype\": \"number\", \n        \"std\": 1716, \n        \"min\": 1000001, \n        \"max\": 1006040, \n        \"num_unique_values\": 3755, \n        \"samples\": [\n          1005905, \n          1003730, \n          1005326\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\"\n      }, \n      \"column\": \"Cust_name\", \n      \"properties\": {\n        \"dtype\": \"category\", \n        \"num_unique_values\": 1250, \n        \"samples\": [\n          \"Nida\", \n          \"Lacy\", \n          \"Caudle\"\n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\"\n      }, \n      \"column\": \"Product_ID\", \n      \"properties\": {\n        \"dtype\": \"category\", \n        \"num_unique_values\": 2351, \n

```

```

\"samples\": [\n          \"P00224442\", \n          \"P00205242\", \n          \"P00347442\" \n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\", \n        \"column\": \"Gender\", \n        \"properties\": { \n          \"dtype\": \"category\", \n          \"num_unique_values\": 2, \n          \"samples\": [\n            \"M\", \n            \"F\" \n          ], \n          \"semantic_type\": \"\", \n          \"description\": \"\", \n          \"column\": \"Age Group\", \n          \"properties\": { \n            \"dtype\": \"category\", \n            \"num_unique_values\": 7, \n            \"samples\": [\n              \"26-35\", \n              \"0-17\" \n            ], \n            \"semantic_type\": \"\", \n            \"description\": \"\", \n            \"column\": \"Age\", \n            \"properties\": { \n              \"dtype\": \"number\", \n              \"std\": 12, \n              \"min\": 12, \n              \"max\": 92, \n              \"num_unique_values\": 81, \n              \"samples\": [\n                18, \n                28 \n              ], \n              \"semantic_type\": \"\", \n              \"description\": \"\", \n              \"column\": \"Marital_Status\", \n              \"properties\": { \n                \"dtype\": \"number\", \n                \"std\": 0, \n                \"min\": 0, \n                \"max\": 1, \n                \"num_unique_values\": 2, \n                \"samples\": [\n                  1, \n                  0 \n                ], \n                \"semantic_type\": \"\", \n                \"description\": \"\", \n                \"column\": \"State\", \n                \"properties\": { \n                  \"dtype\": \"category\", \n                  \"num_unique_values\": 16, \n                  \"samples\": [\n                    \"Maharashtra\", \n                    \"Andhra Pradesh\", \n                    \"Zone\", \n                    \"properties\": { \n                      \"dtype\": \"category\", \n                      \"num_unique_values\": 5, \n                      \"samples\": [\n                        \"Southern\", \n                        \"Eastern\" \n                      ], \n                      \"semantic_type\": \"\", \n                      \"description\": \"\", \n                      \"column\": \"Occupation\", \n                      \"properties\": { \n                        \"dtype\": \"category\", \n                        \"num_unique_values\": 15, \n                        \"samples\": [\n                          \"Retail\", \n                          \"Aviation\" \n                        ], \n                        \"semantic_type\": \"\", \n                        \"description\": \"\", \n                        \"column\": \"Product_Category\", \n                        \"properties\": { \n                          \"dtype\": \"category\", \n                          \"num_unique_values\": 18, \n                          \"samples\": [\n                            \"Auto\", \n                            \"Hand & Power Tools\" \n                          ], \n                          \"semantic_type\": \"\", \n                          \"description\": \"\", \n                          \"column\": \"Orders\", \n                          \"properties\": { \n                            \"dtype\": \"number\", \n                            \"std\": 1, \n                            \"min\": 1, \n                            \"max\": 4, \n                            \"num_unique_values\": 4, \n                            \"samples\": [\n                              3, \n                              4 \n                            ], \n                            \"semantic_type\": \"\", \n                            \"description\": \"\", \n                            \"column\": \"Amount\", \n                            \"properties\": { \n                              \"dtype\": \"number\", \n                              \"std\": 5222.355869186444, \n                              \"min\": 188.0, \n                              \"max\":

```

```
23952.0,\n          \"num_unique_values\": 6584,\n          \"samples\": [\n          19249.0,\n          13184.0\n          ],\n          \"semantic_type\": \"\",,\n          \"description\": \"\"\n          }\n          }\n          ],\"type\":\"dataframe\",\"variable_name\":\"df\"}
```

```
df.tail()
```

```
{\"summary\":{\"name\": \"df\", \"rows\": 5, \"fields\": [\n  {\n    \"column\": \"User_ID\", \"properties\": {\n      \"dtype\": \"number\", \"std\": 1566, \"min\": 1000695,\n      \"max\": 1004089, \"num_unique_values\": 5,\n      \"samples\": [\n        1004089,\n        1002744,\n        1001209\n      ],\n      \"semantic_type\": \"\",,\n      \"description\": \"\"\n    },\n    {\n      \"column\": \"Cust_name\", \"properties\": {\n        \"dtype\": \"string\", \"num_unique_values\": 5,\n        \"samples\": [\n          \"Reichenbach\", \"Brumley\", \"Oshin\"\n        ],\n        \"semantic_type\": \"\",,\n        \"description\": \"\"\n      },\n      {\n        \"column\": \"Product_ID\", \"properties\": {\n          \"dtype\": \"string\", \"num_unique_values\": 5,\n          \"samples\": [\n            \"P00171342\", \"P00281742\", \"P00201342\"\n          ],\n          \"semantic_type\": \"\",,\n          \"description\": \"\"\n        },\n        {\n          \"column\": \"Gender\", \"properties\": {\n            \"dtype\": \"category\", \"num_unique_values\": 2,\n            \"samples\": [\n              \"F\", \"M\"\n            ],\n            \"semantic_type\": \"\",,\n            \"description\": \"\"\n          },\n          {\n            \"column\": \"Age Group\", \"properties\": {\n              \"dtype\": \"string\", \"num_unique_values\": 3,\n              \"samples\": [\n                \"18-25\", \"26-35\"\n              ],\n              \"semantic_type\": \"\",,\n              \"description\": \"\"\n            },\n            {\n              \"column\": \"Age\", \"properties\": {\n                \"dtype\": \"number\", \"std\": 9, \"min\": 19, \"max\": 40,\n                \"num_unique_values\": 4,\n                \"samples\": [\n                  33, 37\n                ],\n                \"semantic_type\": \"\",,\n                \"description\": \"\"\n              },\n              {\n                \"column\": \"Marital_Status\", \"properties\": {\n                  \"dtype\": \"number\", \"std\": 0, \"min\": 0, \"max\": 1,\n                  \"num_unique_values\": 2,\n                  \"samples\": [\n                    0, 1\n                  ],\n                  \"semantic_type\": \"\",,\n                  \"description\": \"\"\n                },\n                {\n                  \"column\": \"State\", \"properties\": {\n                    \"dtype\": \"string\", \"num_unique_values\": 4,\n                    \"samples\": [\n                      \"Haryana\", \"Karnataka\"\n                    ],\n                    \"semantic_type\": \"\",,\n                    \"description\": \"\"\n                  },\n                  {\n                    \"column\": \"Zone\", \"properties\": {\n                      \"dtype\": \"string\", \"num_unique_values\": 4,\n                      \"samples\": [\n                        \"Northern\", \"Southern\"\n                      ]\n                    }\n                  }\n                }\n              }\n            }\n          }\n        ]\n      }\n    }\n  }\n}
```

```

],\n      \"semantic_type\": \"\", \n      \"description\": \"\"\n}\n  },\n  {\n    \"column\": \"Occupation\", \n    \"properties\": {\n      \"dtype\": \"string\", \n      \"num_unique_values\": 4, \n      \"samples\": [\n        \"Healthcare\", \n        \"Agriculture\" \n      ], \n      \"semantic_type\": \"\", \n      \"description\": \"\" \n    }, \n    {\n      \"column\": \"Product_Category\", \n      \"properties\": {\n        \"dtype\": \"category\", \n        \"num_unique_values\": 2, \n        \"samples\": [\n          \"Veterinary\", \n          \"Office\" \n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      }, \n      {\n        \"column\": \"Orders\", \n        \"properties\": {\n          \"dtype\": \"number\", \n          \"std\": 0, \n          \"min\": 3, \n          \"max\": 4, \n          \"num_unique_values\": 2, \n          \"samples\": [\n            3, \n            4 \n          ], \n          \"semantic_type\": \"\", \n          \"description\": \"\" \n        }, \n        {\n          \"column\": \"Amount\", \n          \"properties\": {\n            \"dtype\": \"number\", \n            \"std\": \n91.47513323302677, \n            \"min\": 188.0, \n            \"max\": 370.0, \n            \"num_unique_values\": 5, \n            \"samples\": [\n              367.0, \n              188.0 \n            ], \n            \"semantic_type\": \"\", \n            \"description\": \"\" \n          } \n        } \n      ], \n      \"type\": \"dataframe\"}

```

```
df.dropna(inplace=True)
```

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

```
df['Amount'].dtypes
```

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

```
df.columns
```

```

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

```

```
#rename column
```

```
df.rename(columns = {'Marital_Status': 'Shaadi'})
```

```

{"summary": "{\n  \"name\": \"df\", \n  \"rows\": 11251, \n  \"fields\": \n  [\n    {\n      \"column\": \"User_ID\", \n      \"properties\": {\n        \"dtype\": \"number\", \n        \"std\": 1716, \n        \"min\": \n1000001, \n        \"max\": 1006040, \n        \"num_unique_values\": \n3755, \n        \"samples\": [\n          1005905, \n          1003730, \n          1005326 \n        ], \n        \"semantic_type\": \"\", \n        \"description\": \"\" \n      }, \n      {\n        \"column\": \n\"Cust_name\", \n        \"properties\": {\n          \"dtype\": \n

```



```

\"category\", \n          \"num_unique_values\": 1250, \n
\"samples\": [ \n          \"Nida\", \n          \"Lacy\", \n
\"Caudle\", \n          ], \n          \"semantic_type\": \"\", \n
\"description\": \"\", \n          }, \n          { \n          \"column\":
\"Product_ID\", \n          \"properties\": { \n          \"dtype\":
\"category\", \n          \"num_unique_values\": 2351, \n
\"samples\": [ \n          \"P00224442\", \n          \"P00205242\", \n
\"P00347442\", \n          ], \n          \"semantic_type\": \"\", \n
\"description\": \"\", \n          }, \n          { \n          \"column\":
\"Gender\", \n          \"properties\": { \n          \"dtype\":
\"category\", \n          \"num_unique_values\": 2, \n          \"samples\":
[ \n          \"M\", \n          \"F\", \n          ], \n          \"semantic_type\": \"\", \n
\"description\": \"\", \n          }, \n          { \n          \"column\": \"Age Group\", \n
\"properties\": { \n          \"dtype\": \"category\", \n
\"num_unique_values\": 7, \n          \"samples\": [ \n          \"26-35\", \n
\"0-17\", \n          ], \n          \"semantic_type\":
\"\", \n          \"description\": \"\", \n          }, \n          { \n
\"column\": \"Age\", \n          \"properties\": { \n          \"dtype\":
\"number\", \n          \"std\": 12, \n          \"min\": 12, \n
\"max\": 92, \n          \"num_unique_values\": 81, \n          \"samples\":
[ \n          18, \n          28, \n          ], \n          \"semantic_type\":
\"\", \n          \"description\": \"\", \n          }, \n          { \n
\"column\": \"Shaadi\", \n          \"properties\": { \n          \"dtype\":
\"number\", \n          \"std\": 0, \n          \"min\": 0, \n
\"max\": 1, \n          \"num_unique_values\": 2, \n          \"samples\":
[ \n          1, \n          0, \n          ], \n          \"semantic_type\":
\"\", \n          \"description\": \"\", \n          }, \n          { \n
\"column\": \"State\", \n          \"properties\": { \n          \"dtype\":
\"category\", \n          \"num_unique_values\": 16, \n
\"samples\": [ \n          \"Maharashtra\", \n          \"Andhra\", \n
\"u00a0Pradesh\", \n          ], \n          \"semantic_type\": \"\", \n
\"description\": \"\", \n          }, \n          { \n          \"column\":
\"Zone\", \n          \"properties\": { \n          \"dtype\": \"category\", \n
\"num_unique_values\": 5, \n          \"samples\": [ \n
\"Southern\", \n          \"Eastern\", \n          ], \n          \"semantic_type\": \"\", \n
\"description\": \"\", \n          }, \n          { \n          \"column\":
\"Occupation\", \n          \"properties\": { \n          \"dtype\": \"category\", \n
\"num_unique_values\": 15, \n          \"samples\": [ \n
\"Retail\", \n          \"Aviation\", \n          ], \n          \"semantic_type\": \"\", \n
\"description\": \"\", \n          }, \n          { \n          \"column\":
\"Product_Category\", \n          \"properties\": { \n          \"dtype\": \"category\", \n
\"num_unique_values\": 18, \n          \"samples\": [ \n
\"Auto\", \n          \"Hand & Power Tools\", \n          ], \n          \"semantic_type\": \"\", \n
\"description\": \"\", \n          }, \n          { \n          \"column\":
\"Orders\", \n          \"properties\":
{ \n          \"dtype\": \"number\", \n          \"std\": 1, \n

```

```

{"min": 1, "max": 4, "num_unique_values": 4, "samples": [3, 4], "semantic_type": "", "description": "", "column": "Amount", "properties": {"dtype": "number", "std": 5222.355869186444, "min": 188.0, "max": 23952.0, "num_unique_values": 6584, "samples": [19249.0, 13184.0]}, "semantic_type": "", "description": ""}
{"type": "dataframe"}

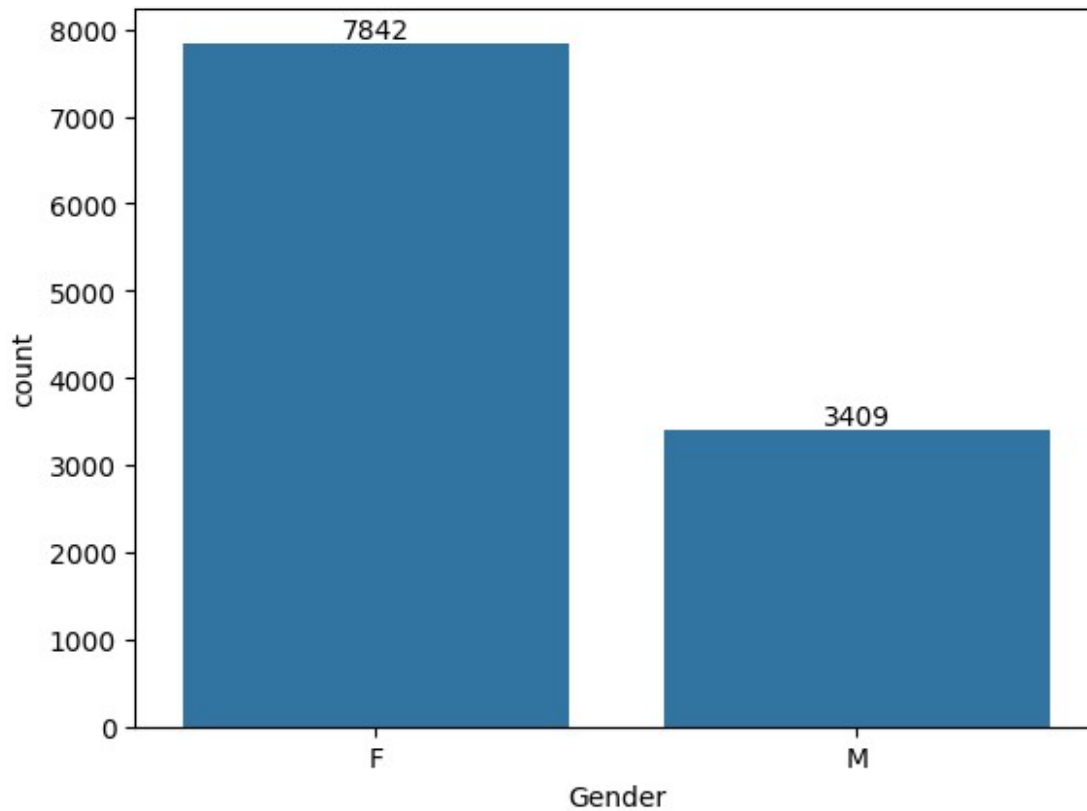
#use describe() for specific cloumns
df[['Age', 'Orders', 'Amount']].describe()

{"summary": {"name": "df[['Age', 'Orders', 'Amount']]", "rows": 8, "fields": [{"column": "Age", "properties": {"dtype": "number", "std": 3965.0199871765367, "min": 12.0, "max": 11251.0, "num_unique_values": 8, "samples": [35.421207003821884, 33.0, 11251.0]}, "semantic_type": "", "description": ""}, {"column": "Orders", "properties": {"dtype": "number", "std": 3977.0664408388902, "min": 1.0, "max": 11251.0, "num_unique_values": 8, "samples": [2.4892898409030306, 2.0, 11251.0]}, "semantic_type": "", "description": ""}, {"column": "Amount", "properties": {"dtype": "number", "std": 7024.070625946779, "min": 188.0, "max": 23952.0, "num_unique_values": 8, "samples": [9453.610857727557, 8109.0, 11239.0]}, "semantic_type": "", "description": ""}], "type": "dataframe"}

# plotting a bar chart for Gender and It's count
ax = sns.countplot(x = 'Gender', data = df)

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

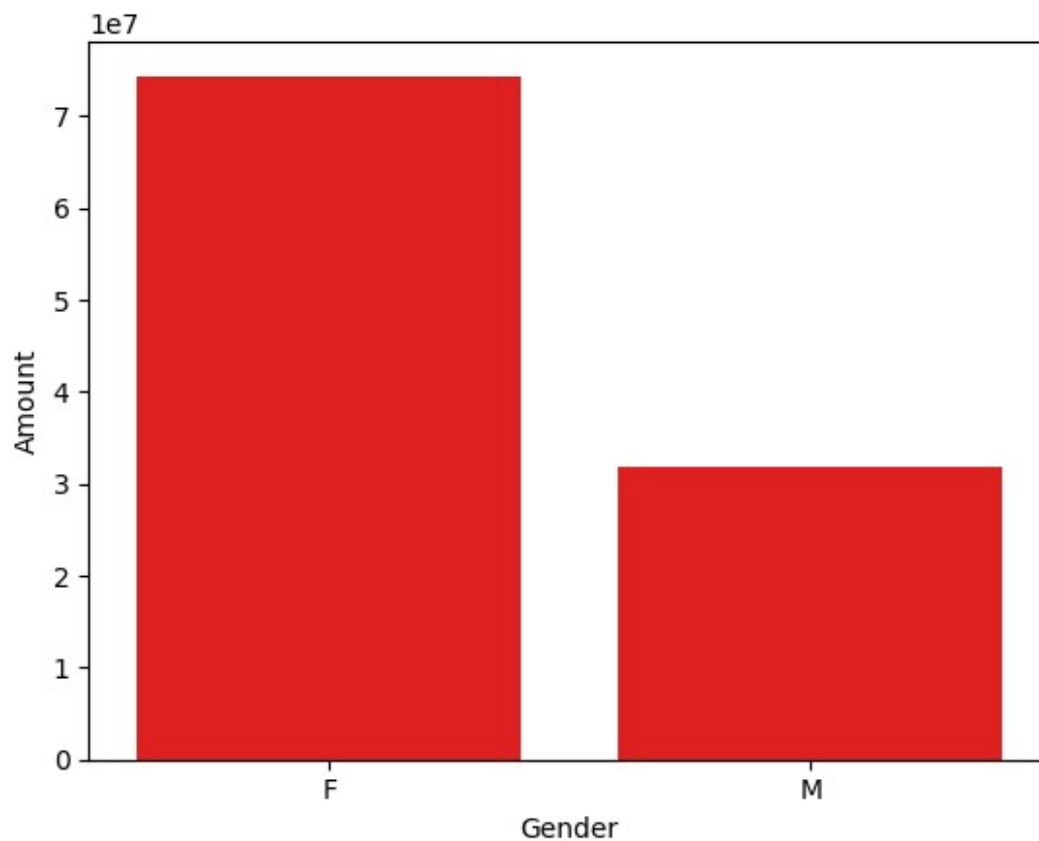
```



In this graph there is more count of women than men

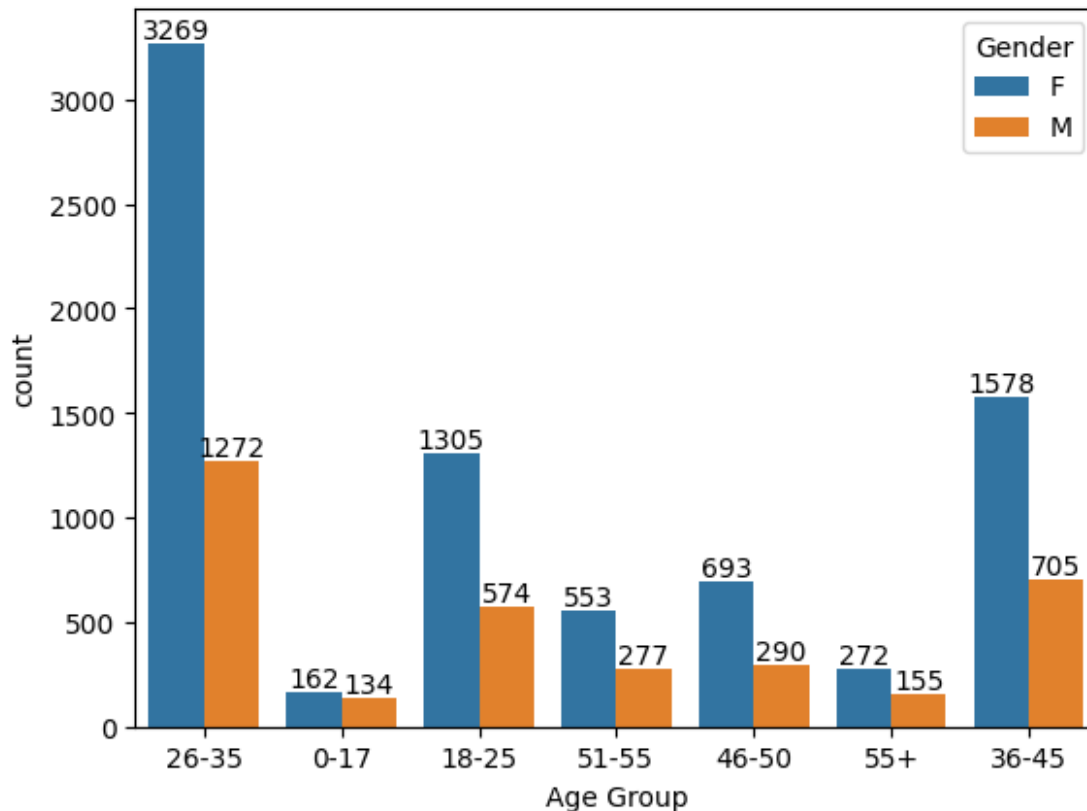
```
# plotting a bar chart fort gender vs total amount  
sales_gen = df.groupby(['Gender'], as_index=False)  
['Amount'].sum().sort_values(by='Amount', ascending=False)  
sns.barplot(x = 'Gender',y= 'Amount' ,data = sales_gen, color='red')
```

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



In this graph Female spent more money than men

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



there is a count of people according to thier age

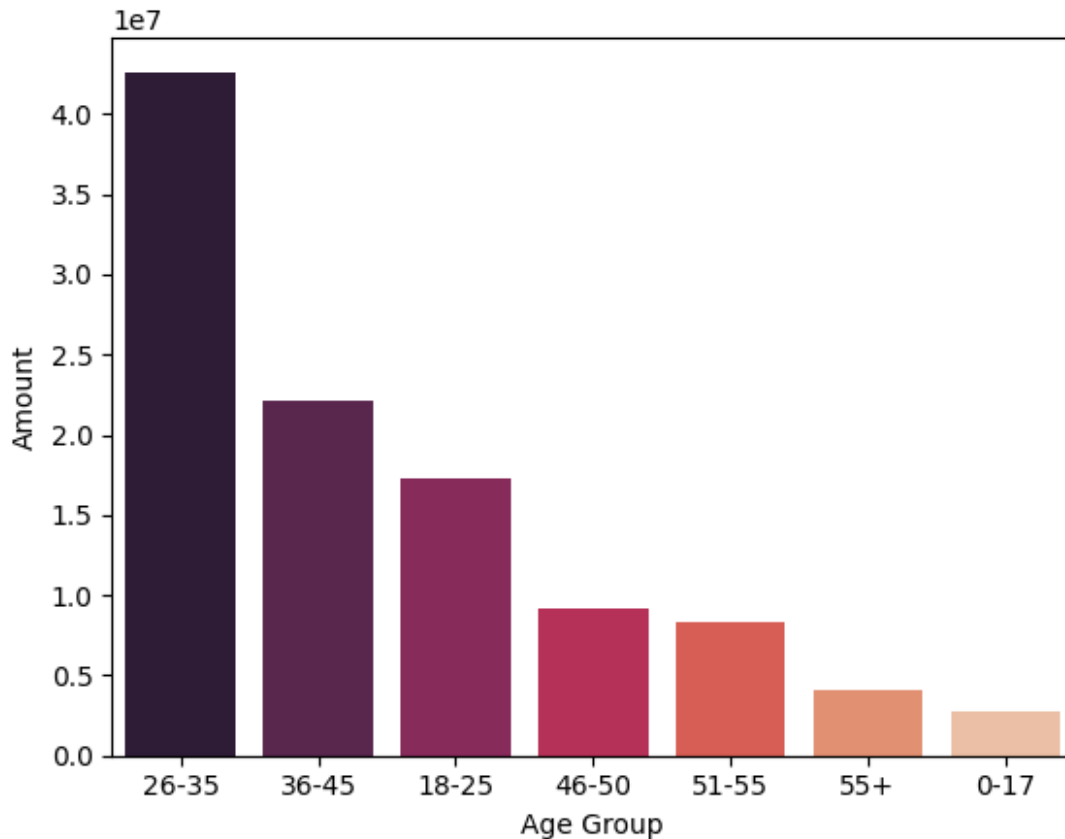
```
#total amount vs age group
sales_age = df.groupby(['Age Group'], as_index=False)
['Amount'].sum().sort_values(by='Amount', ascending=False)

sns.barplot(x = 'Age Group',y= 'Amount' ,data = sales_age,
color='red', palette='rocket')

<ipython-input-29-ea2b9d5bcad3>:4: FutureWarning:
Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

sns.barplot(x = 'Age Group',y= 'Amount' ,data = sales_age,
color='red', palette='rocket')

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



the age group of 26 and 35 sell more products

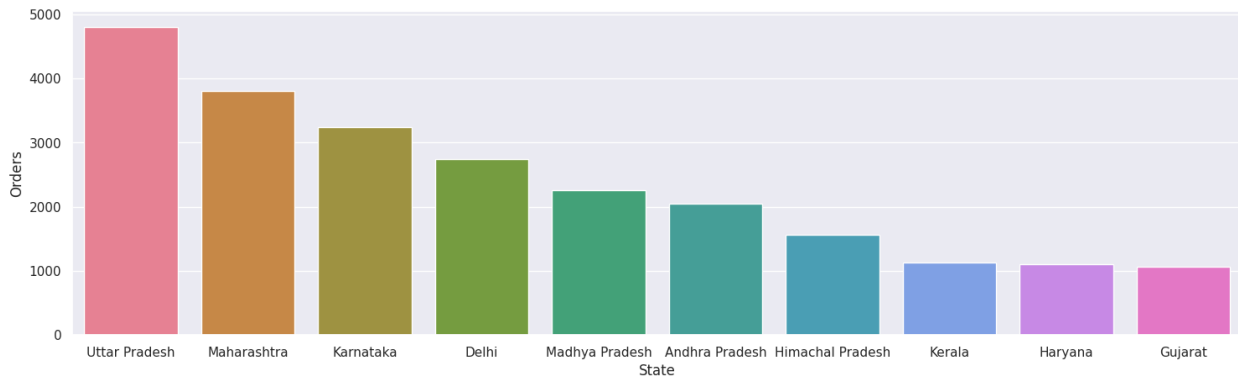
```
sales_state = df.groupby(['State'], as_index=False)
['Orders'].sum().sort_values(by='Orders', ascending=False).head(10)
custom_color = sns.color_palette('husl', 10)
sns.set(rc={'figure.figsize': (18, 5)})
sns.barplot(x = 'State', y= 'Orders' ,data = sales_state,
palette=custom_color)
```

<ipython-input-32-9db373daf273>:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x = 'State', y= 'Orders' ,data = sales_state,
palette=custom_color)
```

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



the maximum order is got from the uttarpradesh remains are poor Aura ∞

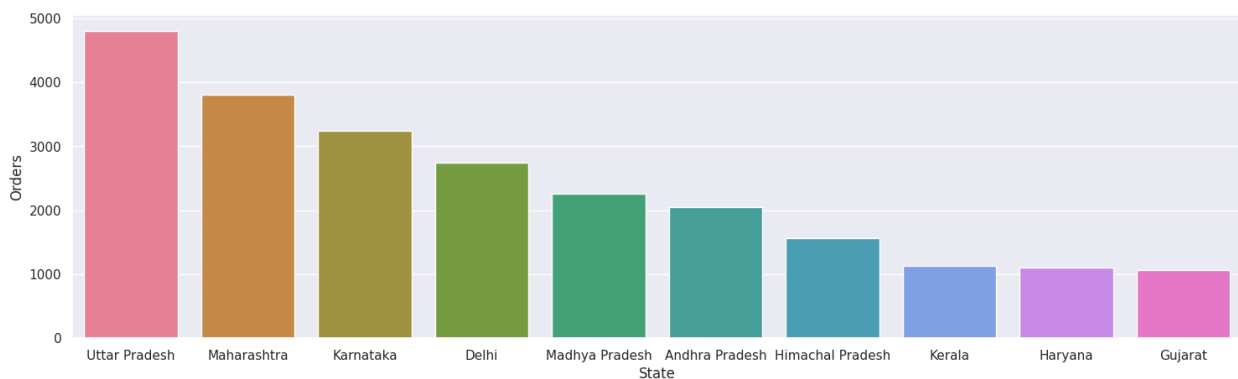
```
sales_state = df.groupby(['State'], as_index=False)
['Orders'].sum().sort_values(by='Orders', ascending=False).head(10)
custom_color = sns.color_palette('husl',10)
sns.set(rc={'figure.figsize':(18,5)})
sns.barplot(x = 'State',y= 'Orders' ,data = sales_state,
palette=custom_color)
```

<ipython-input-37-9db373daf273>:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x = 'State',y= 'Orders' ,data = sales_state,
palette=custom_color)
```

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



once again , Uttar pradesh is showing their status

```
# prompt: sales_state = df.groupby(['State'], as_index=False)
['Orders'].sum().sort_values(by='Orders', ascending=False).head(10)
# custom_color = sns.color_palette('husl',10)
# sns.set(rc={'figure.figsize':(18,5)})
# sns.barplot(x = 'State',y= 'Orders' ,data = sales_state,
palette=custom_color) --> i want sales like orders*amount
```

```
# Calculate total sales (Orders * Amount)
#df['Amount'] = df['Orders'] * df['Amount']
```

```
# Group by state and sum total sales
sales_state = df.groupby(['State'], as_index=False)
['Amount'].sum().sort_values(by='Amount', ascending=False).head(10)
```

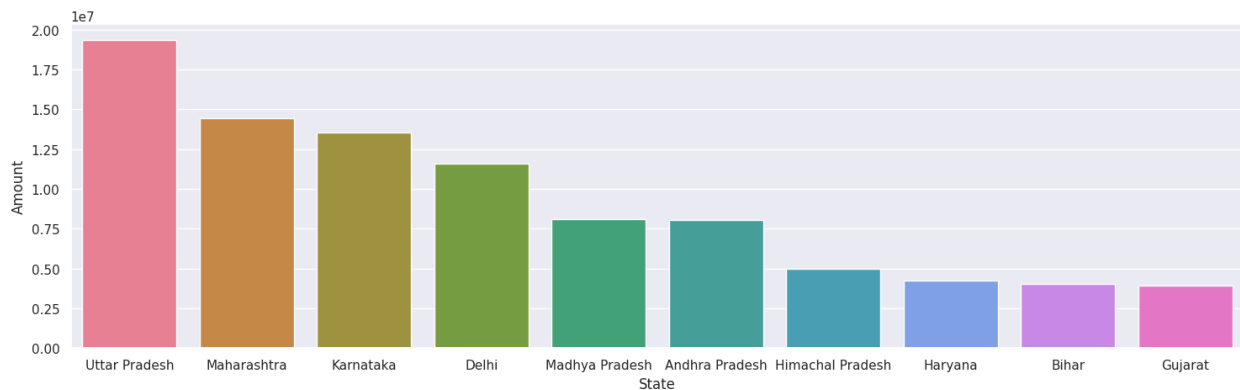
```
# Customize the plot
custom_color = sns.color_palette('husl', 10)
sns.set(rc={'figure.figsize':(18,5)})
sns.barplot(x='State', y='Amount', data=sales_state,
palette=custom_color)
```

<ipython-input-41-d2a11e0b5a32>:15: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(x='State', y='Amount', data=sales_state,
palette=custom_color)
```

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

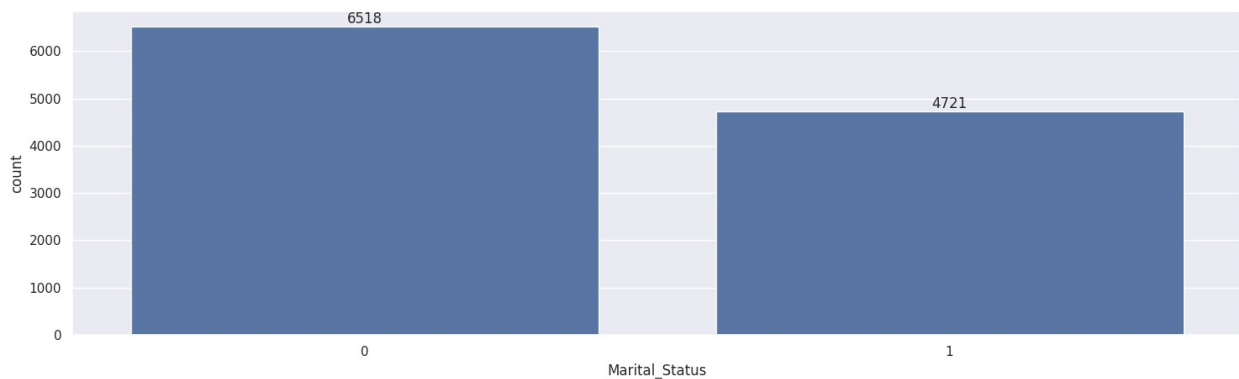


People of Uttar Pradesh earn more than other poor state(Rajasthan)

```
ax = sns.countplot(data = df, x = 'Marital_Status')
sns.set(rc={'figure.figsize':(7,5)})
```



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

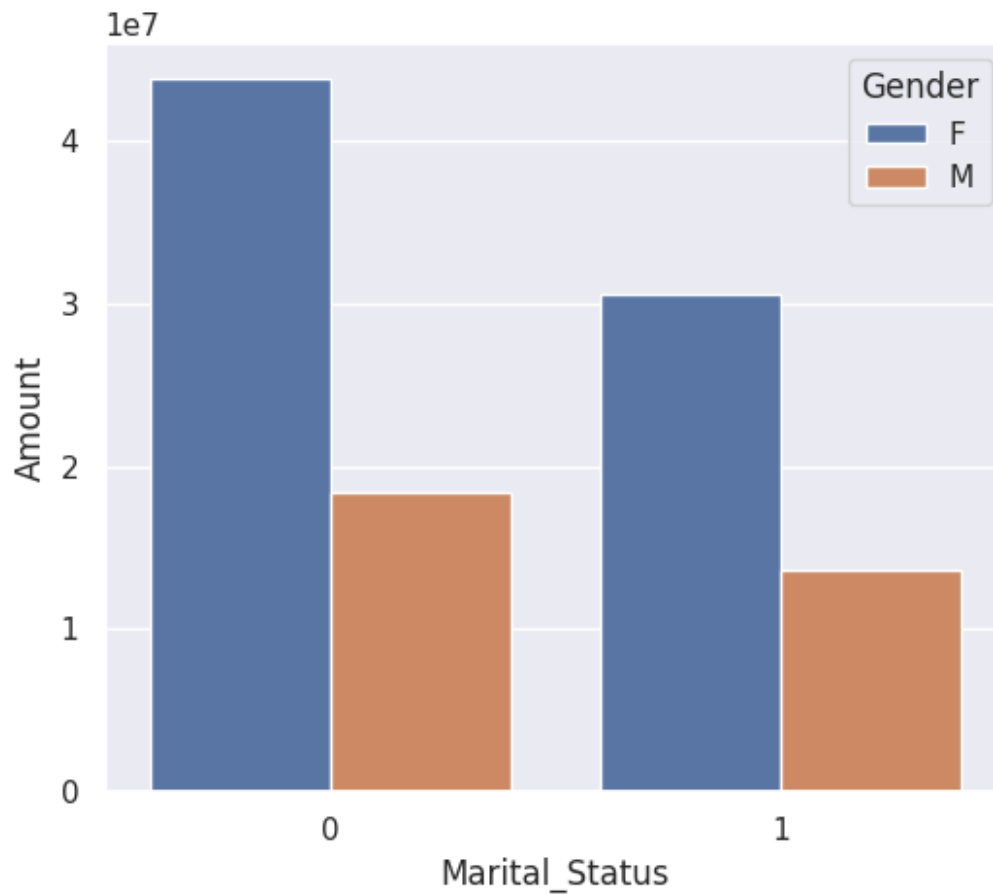


this figure shows that the unmarried people have high graph

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

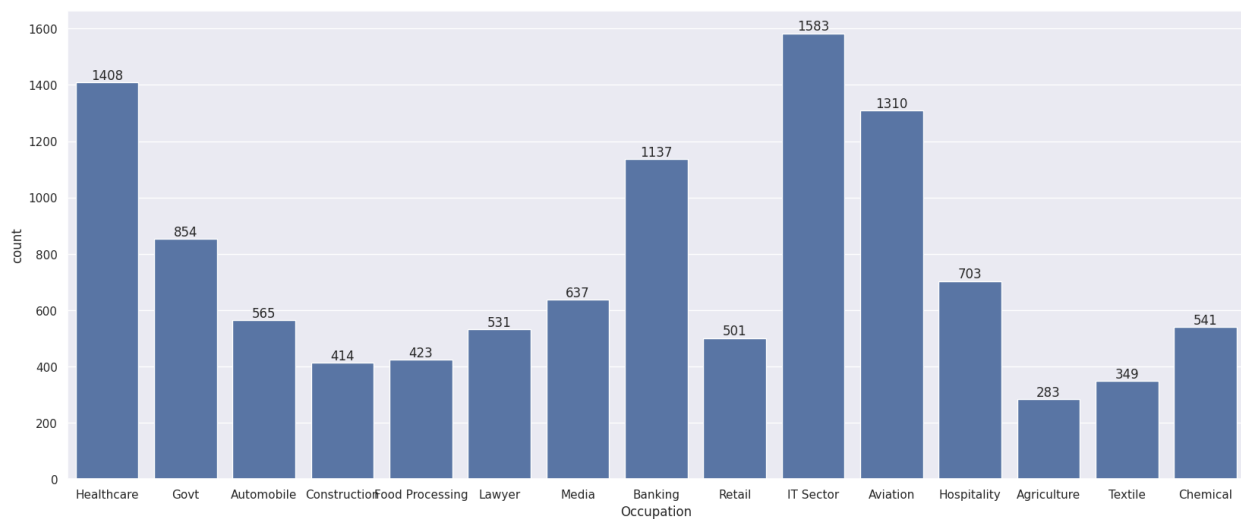
```
sns.set(rc={'figure.figsize':(6,5)})  
sns.barplot(data = sales_state, x = 'Marital_Status', y= 'Amount',  
hue='Gender')
```

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



```
sns.set(rc = {'figure.figsize':(20,8)})
ax = sns.countplot(data = df, x = 'Occupation')

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



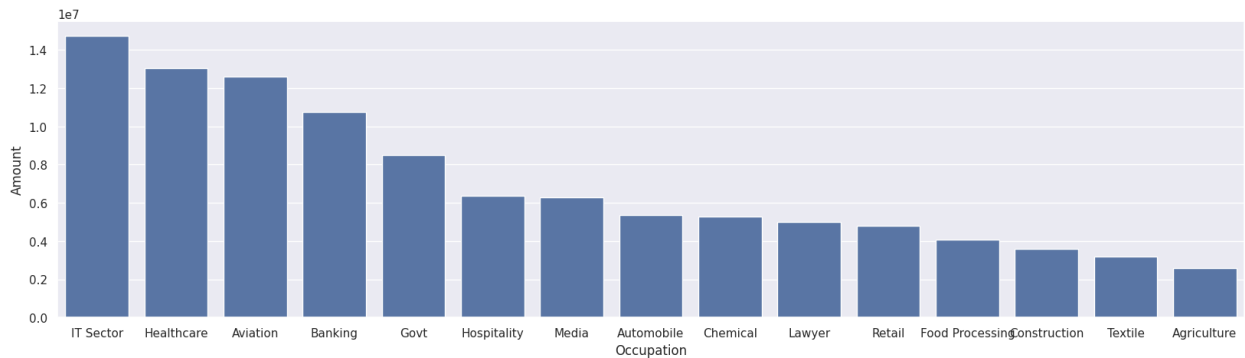
```

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')

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

```



```

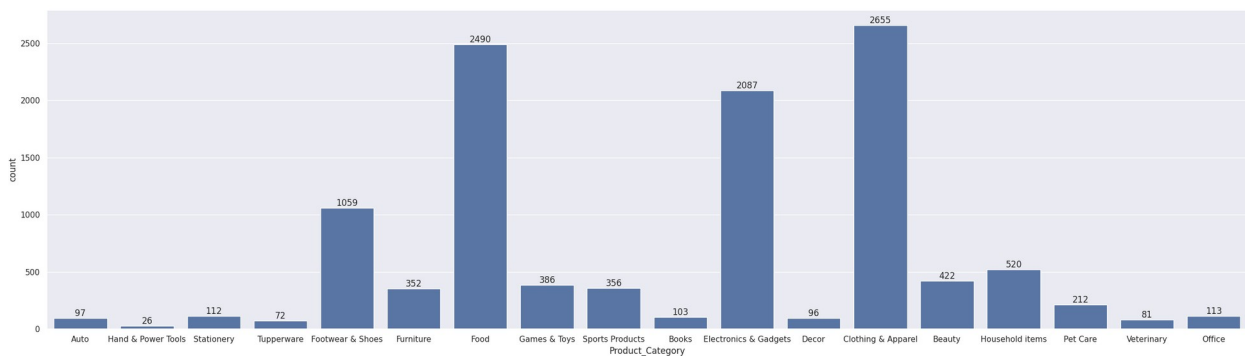
sns.set(rc = {'figure.figsize':(30,8)})
ax = sns.countplot(data = df, x = 'Product_Category')

```

```

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

```

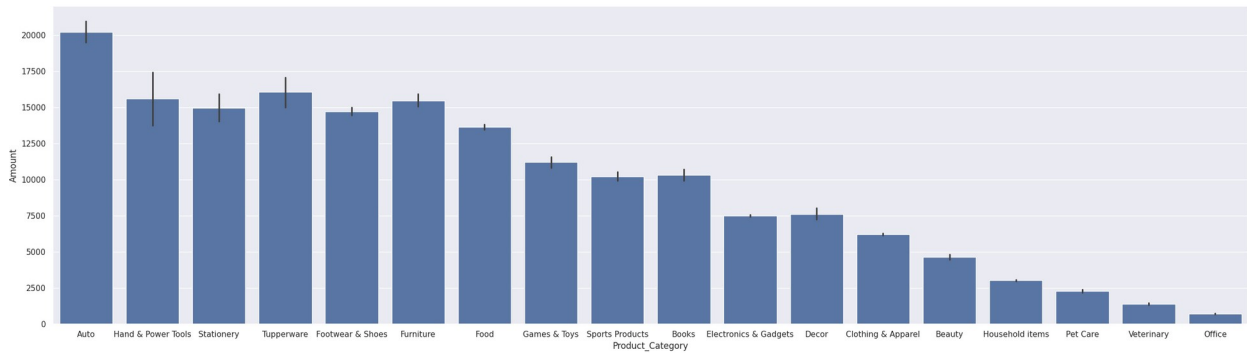


```

sns.set(rc = {'figure.figsize':(30,8)})
sns.barplot(data = df, x = 'Product_Category',y= 'Amount')

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

```



prompt: I want to conclude my analysis and want recommendation by analysing the all graphs above

Recommendations based on the provided analysis:

1. Target Female Customers:

- Females contribute significantly more to overall sales than males.

- Focus marketing and product development efforts towards female demographics.

2. Focus on Age Group 26-35:

- This age group exhibits the highest spending.

- Tailor product offerings and marketing campaigns to resonate with this demographic.

3. Prioritize Uttar Pradesh:

- Uttar Pradesh demonstrates the highest sales volume and revenue.

- Increase marketing efforts and inventory in this region.

- Explore reasons for high sales in this state and replicate successful strategies in other regions.

4. Consider Marital Status:

- Unmarried individuals constitute a larger customer base and promotions.

- Further investigation could examine if marital status influences

- Analyze purchasing patterns within this group to personalize offers and spending habits.

5. Product Category Optimization:

- Analyze the best-selling product categories and identify any correlations with other factors like age, gender, or location.

- Optimize inventory levels for top-performing product categories in high-demand areas.

6. Occupation-based Targeting:

```

# - Analyze the spending patterns based on occupation to tailor
marketing campaigns.
# - Certain occupations may have higher disposable incomes or
specific product needs.

# 7. Deeper Data Analysis:
# - Investigate the correlations between various factors such as
age, gender, marital status, occupation, product category, and
location to develop more nuanced customer segments.
# - Consider A/B testing different marketing strategies across
different segments.
# - Analyze seasonality and trends to identify periods of peak
demand and optimize stock levels.

# 8. Geographic Expansion:
# - Explore opportunities to expand into new markets or regions
with similar demographic profiles to Uttar Pradesh.

# 9. Customer Retention:
# - Implement customer loyalty programs and personalized
recommendations to retain high-value customers.
# - Analyze customer churn and identify factors that contribute to
it.

# Assuming df is already defined
sales_profit_by_segment = df.groupby('Segment').agg({'Sales': 'sum',
'Profit': 'sum'}).reset_index()

# Use plotly express color palette
color_palette = px.colors.qualitative.Pastel

# Create figure
fig = go.Figure()

# Add Sales Bar
fig.add_trace(go.Bar(
    x=sales_profit_by_segment['Segment'],
    y=sales_profit_by_segment['Sales'],
    name='Total Sales',
    marker_color=color_palette[0]
)))

# Add Profit Bar
fig.add_trace(go.Bar(
    x=sales_profit_by_segment['Segment'],
    y=sales_profit_by_segment['Profit'],
    name='Total Profit',
    marker_color=color_palette[1]
)))

```

```

))

# Update layout
fig.update_layout(
    barmode='stack',
    title='Total Sales and Profit by Segment',
    xaxis_title='Segment',
    yaxis_title='Amount'
)

# Show plot
fig.show()

```

```

-----
-----
KeyError                                Traceback (most recent call
last)
<ipython-input-17-31b683141270> in <cell line: 0>()
      1 # Assuming df is already defined
----> 2 sales_profit_by_segment = df.groupby('Segment').agg({'Sales':
'Sum', 'Profit': 'sum'}).reset_index()
      3
      4 # Use plotly express color palette
      5 color_palette = px.colors.qualitative.Pastel

/usr/local/lib/python3.11/dist-packages/pandas/core/frame.py in
groupby(self, by, axis, level, as_index, sort, group_keys, observed,
dropna)
    9181         raise TypeError("You have to supply one of 'by'
and 'level'")
    9182
-> 9183         return DataFrameGroupBy(
    9184             obj=self,
    9185             keys=by,

/usr/local/lib/python3.11/dist-packages/pandas/core/groupby/groupby.py
in __init__(self, obj, keys, axis, level, grouper, exclusions,
selection, as_index, sort, group_keys, observed, dropna)
    1327
    1328         if grouper is None:
-> 1329             grouper, exclusions, obj = get_grouper(
    1330                 obj,
    1331                 keys,

/usr/local/lib/python3.11/dist-packages/pandas/core/groupby/grouper.py
in get_grouper(obj, key, axis, level, sort, observed, validate,
dropna)
    1041         in_axis, level, gpr = False, gpr, None
    1042         else:
-> 1043             raise KeyError(gpr)

```

```
1044         elif isinstance(gpr, Grouper) and gpr.key is not None:
1045             # Add key to exclusions
```

```
KeyError: 'Segment'
```

```
# Assuming df is already defined
# Check if 'Segment' column exists, if not, create it based on
available data
if 'Segment' not in df.columns:
    # Example: Create 'Segment' based on 'Age Group' or any other
relevant column
    df['Segment'] = pd.cut(df['Age'], bins=[0, 25, 35, 45, 55, 100],
                           labels=['<25', '26-35', '36-45', '46-55',
                                   '>55'])
```

```
sales_profit_by_segment = df.groupby('Segment').agg({'Amount':
'sum'}).reset_index()
```

```
color_palette = px.colors.qualitative.Pastel
```

```
fig = go.Figure()
```

```
fig.add_trace(go.Bar(
    x=sales_profit_by_segment['Segment'],
    y=sales_profit_by_segment['Amount'],
    name='Total Sales',
    marker_color=color_palette[0]
))
```

```
fig.update_layout(
    title='Total Sales by Segment',
    xaxis_title='Segment',
    yaxis_title='Amount'
)
```

```
# Show plot
fig.show()
```

```
<ipython-input-30-d6a4afd8ecc3>:8: FutureWarning:
```

The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```
sales_profit_by_segment = df.groupby('Segment').agg({'Amount':
'sum'}).reset_index()
sales_profit_by_segment['Sales_to_Profit_Ratio'] =
sales_profit_by_segment['Amount'] / sales_profit_by_segment['Amount']
print(sales_profit_by_segment[['Segment', 'Sales_to_Profit_Ration']])
```

<ipython-input-33-4c664a7a0ecf>:1: FutureWarning:

The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```
-----
-----
KeyError                                Traceback (most recent call
last)
<ipython-input-33-4c664a7a0ecf> in <cell line: 0>()
      1 sales_profit_by_segment = df.groupby('Segment').agg({'Amount':
'sum'}).reset_index()
      2 sales_profit_by_segment['Sales_to_Profit_Ratio'] =
sales_profit_by_segment['Amount'] / sales_profit_by_segment['Amount']
----> 3
print(sales_profit_by_segment[['Segment', 'Sales_to_Profit_Ration']])

/usr/local/lib/python3.11/dist-packages/pandas/core/frame.py in
__getitem__(self, key)
    4106         if is_iterator(key):
    4107             key = list(key)
-> 4108         indexer = self.columns._get_indexer_strict(key,
"columns")[1]
    4109
    4110         # take() does not accept boolean indexers

/usr/local/lib/python3.11/dist-packages/pandas/core/indexes/base.py in
_get_indexer_strict(self, key, axis_name)
    6198         keyarr, indexer, new_indexer =
self._reindex_non_unique(keyarr)
    6199
-> 6200         self._raise_if_missing(keyarr, indexer, axis_name)
    6201
    6202         keyarr = self.take(indexer)

/usr/local/lib/python3.11/dist-packages/pandas/core/indexes/base.py in
_raise_if_missing(self, key, indexer, axis_name)
    6250
    6251         not_found = list(ensure_index(key)
[missing_mask.nonzero()[0]].unique())
-> 6252         raise KeyError(f"{not_found} not in index")
    6253
    6254     @overload

KeyError: "[ 'Sales_to_Profit_Ration' ] not in index"
```



```

sales_profit_by_segment = df.groupby('Segment').agg({'Amount':
'sum'}).reset_index()
sales_profit_by_segment['Sales_to_Profit_Ratio'] =
sales_profit_by_segment['Amount'] / sales_profit_by_segment['Amount']
print(sales_profit_by_segment[['Segment','Sales_to_Profit_Ratio']])

```

	Segment	Sales_to_Profit_Ratio
0	<25	1.0
1	26-35	1.0
2	36-45	1.0
3	46-55	1.0
4	>55	1.0

<ipython-input-35-98fb7dd2c6df>:1: FutureWarning:

The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.

```

fig = px.bar(sales_profit_by_segment,x = 'Segment',y =
'Sales_to_Profit_Ratio',color = 'Segment',title = 'Sales to Profit
Ratio by Segment')
fig.update_traces(textposition='inside')
fig.update_layout(title_text='Sales to Profit Ratio by
Segment',title_font = dict(size=24))
fig.show()

```

prompt: conclude the above graph

Recommendations based on the provided analysis:

1. Target Female Customers:

- Females contribute significantly more to overall sales than males.

- Focus marketing and product development efforts towards female demographics.

2. Focus on Age Group 26-35:

- This age group exhibits the highest spending.

- Tailor product offerings and marketing campaigns to resonate with this demographic.

3. Prioritize Uttar Pradesh:

- Uttar Pradesh demonstrates the highest sales volume and revenue.

- Increase marketing efforts and inventory in this region.

- Explore reasons for high sales in this state and replicate successful strategies in other regions.

4. Consider Marital Status:

- # - Unmarried individuals constitute a larger customer base.
- # - Analyze purchasing patterns within this group to personalize offers and promotions.

5. Product Category Optimization:

- # - Analyze the best-selling product categories and identify any correlations with other factors like age, gender, or location.
- # - Optimize inventory levels for top-performing product categories in high-demand areas.

6. Occupation-based Targeting:

- # - Analyze the spending patterns based on occupation to tailor marketing campaigns.
- # - Certain occupations may have higher disposable incomes or specific product needs.

7. Deeper Data Analysis:

- # - Investigate the correlations between various factors such as age, gender, marital status, occupation, product category, and location to develop more nuanced customer segments.
- # - Consider A/B testing different marketing strategies across different segments.
- # - Analyze seasonality and trends to identify periods of peak demand and optimize stock levels.

8. Geographic Expansion:

- # - Explore opportunities to expand into new markets or regions with similar demographic profiles to Uttar Pradesh.

9. Customer Retention:

- # - Implement customer loyalty programs and personalized recommendations to retain high-value customers.
- # - Analyze customer churn and identify factors that contribute to it.