Sales Data

June 22, 2024

0.1 Uploading Necessary Modules

[50]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[51]:
```
df=pd.read_csv('Sales Data.csv',encoding = 'unicode_escape')
df.head(5)
```

[51]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount	Status	unnamed1
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharashtra	Western	Healthcare	Auto	1	23952.0	NaN	NaN
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Pradesh	Southern	Govt	Auto	3	23934.0	NaN	NaN
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Pradesh	Central	Automobile	Auto	3	23924.0	NaN	NaN
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnataka	Southern	Construction	Auto	2	23912.0	NaN	NaN
4	1000588	Joni	P00057942	M	26-35	28	1	Gujarat	Western	Food Processing	Auto	2	23877.0	NaN	NaN

[52]:
```
df.tail()
```

[52]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status
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  \
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
11246     NaN       NaN
11247     NaN       NaN
11248     NaN       NaN
11249     NaN       NaN
11250     NaN       NaN
```

[53]:
```
df["Status"].empty  #Since the Status column is not empty
```

[53]:
```
False
```

[54]:
```
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
```

[55]:
```
df.shape
```

[55]:
```
(11251, 15)
```

[56]:
```
df.size
```

[56]:
```
168765
```

[57]:
```
df.index
```

[57]:
```
RangeIndex(start=0, stop=11251, step=1)
```

[58]:
```
df.columns.value_counts()
```

[58]:
```
User_ID             1
Cust_name           1
Product_ID          1
Gender              1
Age Group           1
Age                 1
Marital_Status      1
State               1
Zone                1
Occupation          1
Product_Category    1
Orders              1
Amount              1
Status              1
unnamed1            1
Name: count, dtype: int64
```

[59]:
```
df.drop(['Status','unnamed1'],axis=1 ,inplace=True)
```

[60]:
```
df.sample(3)
```

[60]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	State	Zone	Occupation	Product_Category	Orders	Amount
7433	1001268	Baptist	P00278642	M	51-55	53	0	Delhi	Central	Healthcare	Clothing & Apparel	4	6946.0
2281	1002837	Ordway	P00148642	F	18-25	19	0	Haryana	Northern	Media	Food	2	15298.0
5319	1004318	Carlisle	P00221442	F	26-35	29	1	Madhya Pradesh	Central	Banking	Footwear & Shoes	3	8510.0

[61]:
```
df.isnull().sum()
```

[61]:
```
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
```

[14]:
```
df["Amount"].fillna(df["Amount"].mean(),inplace=True)
```

[15]:
```
df["Amount"].describe().to_frame()
```

[15]:

	Amount
count	11251.000000
mean	9453.610858
std	5219.569870
min	188.000000
25%	5443.500000
50%	8110.000000
75%	12671.000000
max	23952.000000

[16]:
```
df.describe()
```

[16]:

	User_ID	Age	Marital_Status	Orders	Amount
count	1.125100e+04	11251.000000	11251.000000	11251.000000	11251.000000
mean	1.003004e+06	35.421207	0.420318	2.489290	9453.610858
std	1.716125e+03	12.754122	0.493632	1.115047	5219.569870
min	1.000001e+06	12.000000	0.000000	1.000000	188.000000
25%	1.001492e+06	27.000000	0.000000	1.500000	5443.500000
50%	1.003065e+06	33.000000	0.000000	2.000000	8110.000000
75%	1.004430e+06	43.000000	1.000000	3.000000	12671.000000
max	1.006040e+06	92.000000	1.000000	4.000000	23952.000000

[17]:
```
df.isna().sum() #All NULL values are removed from the Data Set
```

[17]:
```
User_ID             0
Cust_name           0
Product_ID          0
Gender              0
Age Group           0
```

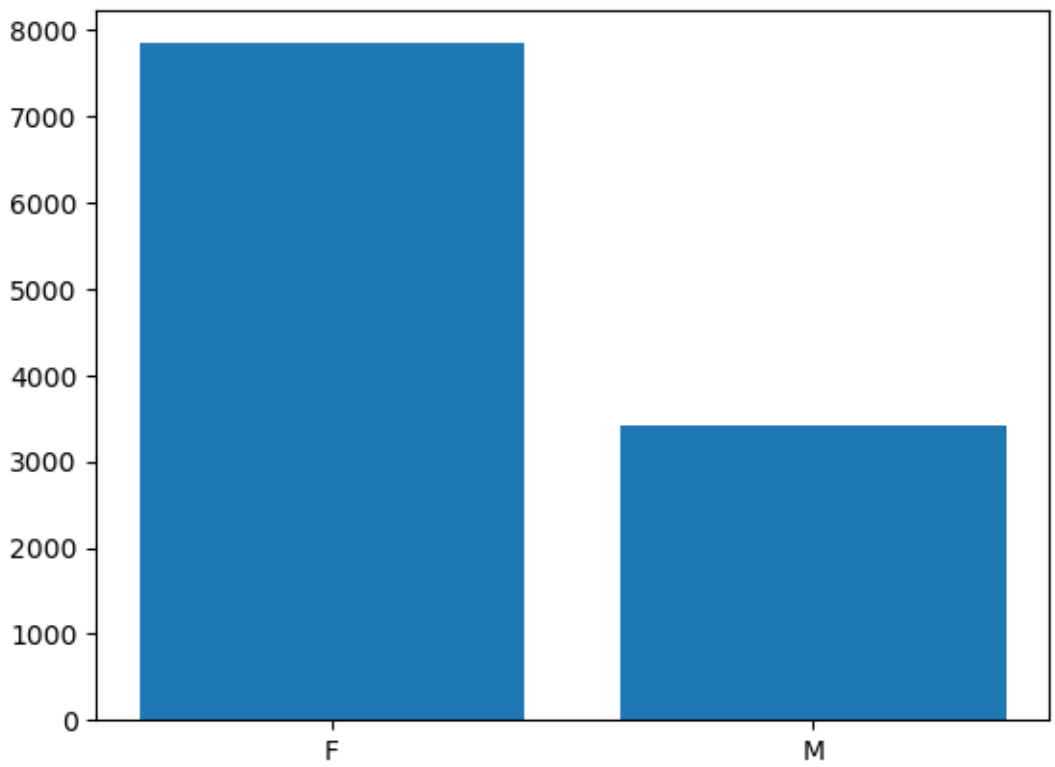
```
Age                 0
Marital_Status      0
State               0
Zone                0
Occupation          0
Product_Category    0
Orders              0
Amount              0
dtype: int64
```

[18]:
```
df.dtypes
```

[18]:
```
User_ID              int64
Cust_name           object
Product_ID          object
Gender              object
Age Group           object
Age                  int64
Marital_Status       int64
State               object
Zone                object
Occupation          object
Product_Category    object
Orders               int64
Amount             float64
dtype: object
```

0.2 Gender Count in Data Set

[19]:
```
gender_count=df["Gender"].value_counts()
plt.bar(gender_count.index,gender_count.values)
plt.show()
```



[20]:
```
df.sample()
```

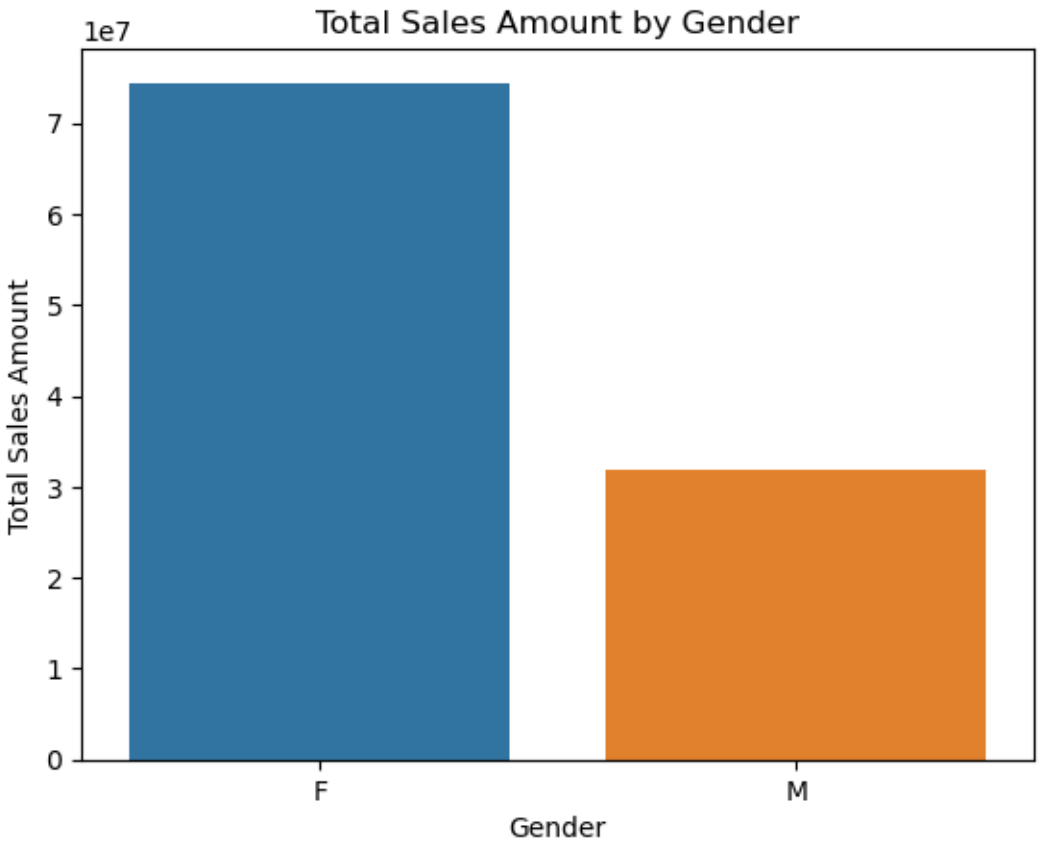
[20]:
```
      User_ID  Cust_name Product_ID Gender Age Group  Age  Marital_Status  \
1981  1004156  Christine  P00066542      F       55+   67               0

        State      Zone Occupation Product_Category  Orders   Amount
1981  Haryana  Northern      Media             Food       3  15571.0
```

[21]:
```
group_data=df.groupby("Gender")["Amount"].sum()

sorted=group_data.sort_values(ascending=False)
sorted.to_frame()
```

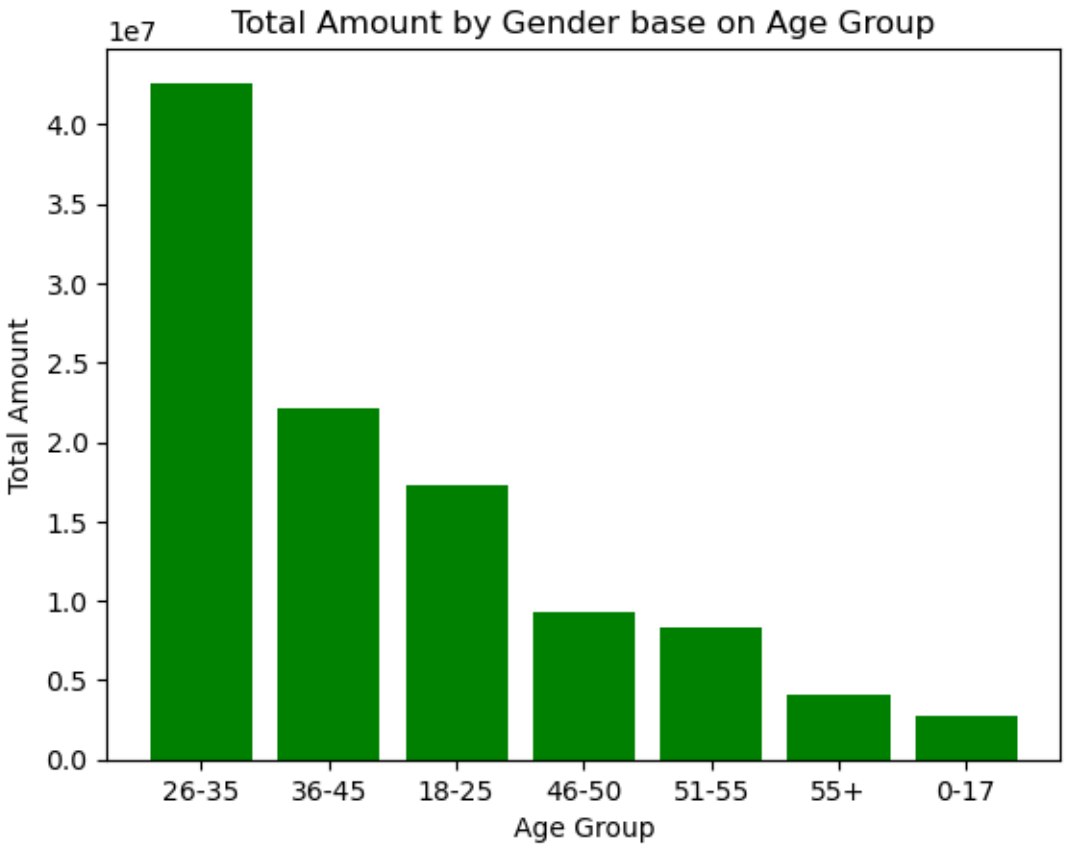
[21]:
```
              Amount
Gender
F       7.443039e+07
M       3.193218e+07
```

[22]:
```
sorted_data = df.groupby(['Gender'])['Amount'].sum().reset_index().
 ↪sort_values(by='Amount', ascending=False)
sns.barplot(x = 'Gender',y= 'Amount' ,data = sorted_data)
plt.xlabel('Gender')
```

```
plt.ylabel('Total Sales Amount')
plt.title('Total Sales Amount by Gender')
plt.show()
```



[23]:
```
sorted_group=df.groupby(["Age Group"])["Amount"].sum().reset_index().
↪sort_values(by='Amount',ascending=False)

plt.bar(sorted_group["Age Group"],sorted_group["Amount"],color="Green")

plt.xlabel("Age Group")

plt.ylabel("Total Amount")

plt.title("Total Amount by Gender base on Age Group")

plt.show()
```



[24]:
```
first_row=sorted_group.iloc[0]

first_row.to_frame()
```

[24]:

	2
Age Group	26-35
Amount	42632351.161715

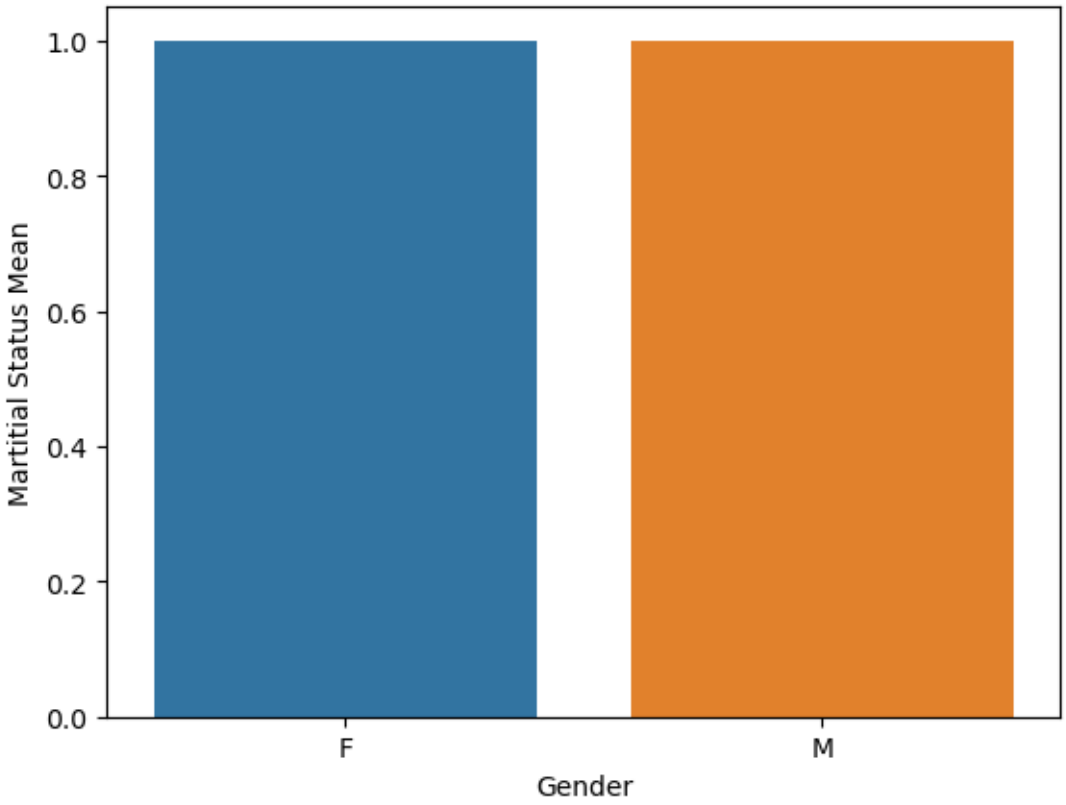
[25]:
```
df1=df.groupby(["Gender"],as_index=False)["Marital_Status"].mean()
df1
```

[25]:

	Gender	Marital_Status
0	F	0.416475
1	M	0.429158

[26]:
```
sns.countplot(data=df1,x="Gender")

plt.xlabel("Gender")

plt.ylabel("Martitial Status Mean")
```

```
plt.show()
```



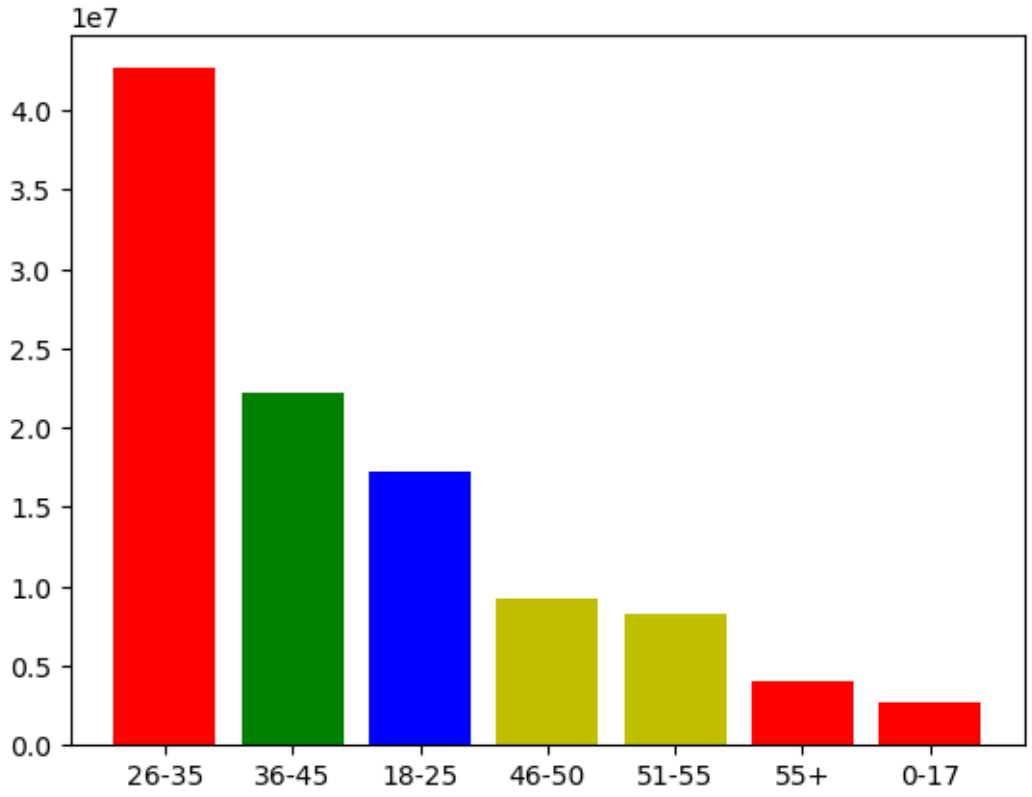
[27]:
```
df_amount=df.groupby(["Age Group"],as_index=False)["Amount"].sum().
↪sort_values(by='Amount',ascending=False)

df_amount
```

[27]:

	Age Group	Amount
2	26-35	4.263235e+07
3	36-45	2.217336e+07
1	18-25	1.724073e+07
4	46-50	9.245658e+06
5	51-55	8.280384e+06
6	55+	4.090441e+06
0	0-17	2.699653e+06

[28]:
```
plt.bar(df_amount["Age␣
↪Group"],df_amount["Amount"],color=["r","g","b","y","y","r","r"])
```

[28]:
```
<BarContainer object of 7 artists>
```



[29]:
```
df.columns
```

[29]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

0.3 Top Three States Orders

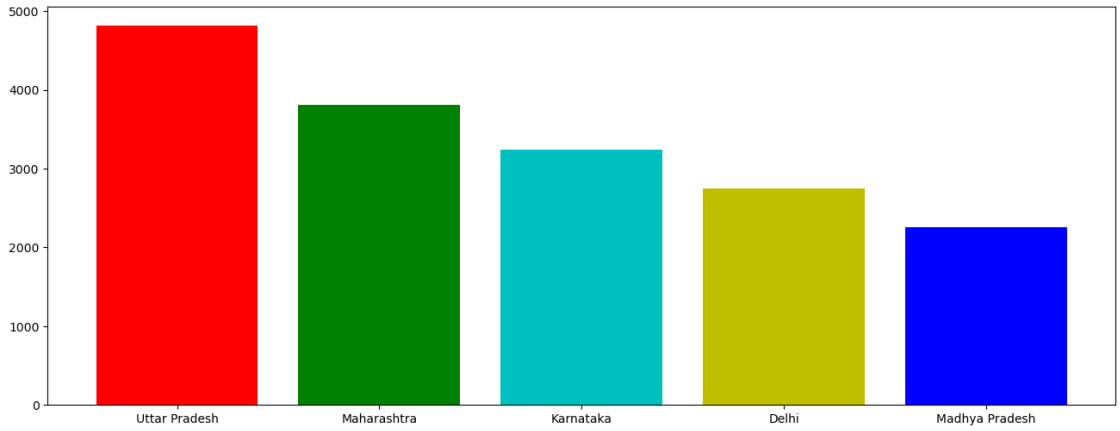
[30]:
```
order_sort=df.groupby(["State"])["Orders"].sum().reset_index().
 ↪sort_values(by="Orders",ascending=False)
order_sort.head(3)
```

[30]:

	State	Orders
14	Uttar Pradesh	4813
10	Maharashtra	3811
7	Karnataka	3241

[31]:
```
plt.figure(figsize=(16,6))
```

```
plt.bar(order_sort["State"].head(5),order_sort["Orders"].
  ↪head(5),color=["r","g","c","y","b"])

plt.show()
```



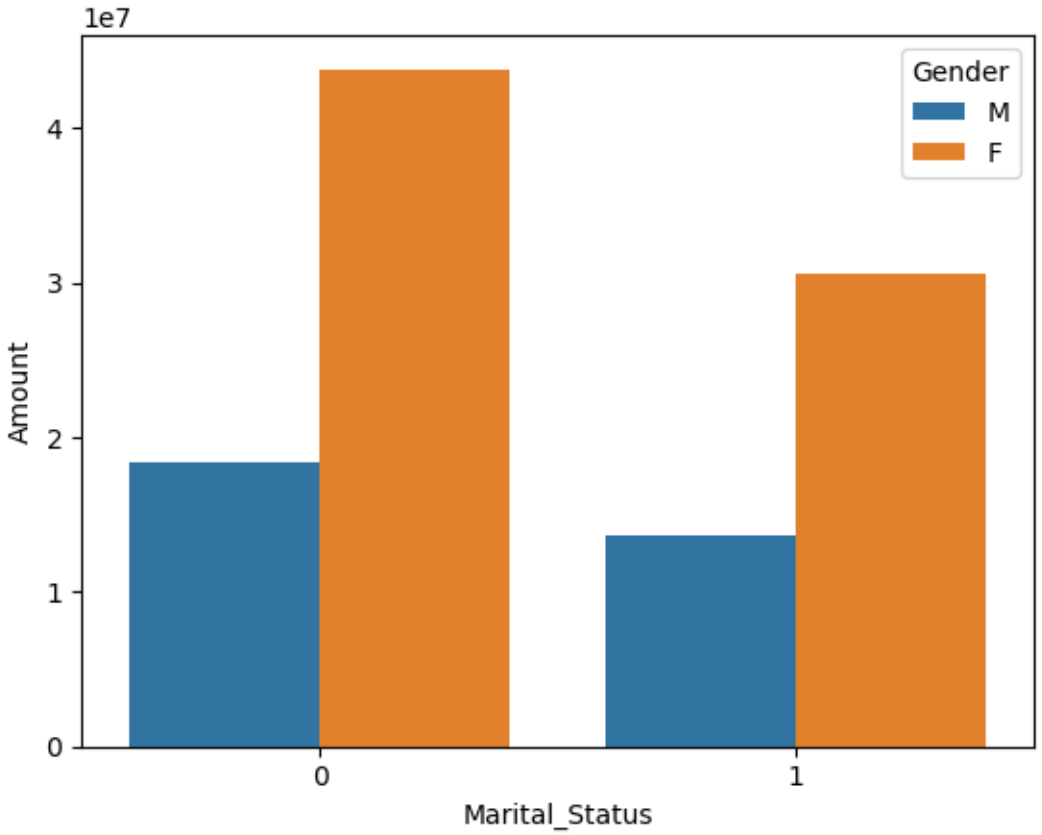
[32]:
```
maritial_sort=df.groupby(["Gender","Marital_Status"])["Amount"].sum().
  ↪reset_index().sort_values(by="Amount",ascending=True)

maritial_sort
```

[32]:

	Gender	Marital_Status	Amount
3	M	1	1.358399e+07
2	M	0	1.834819e+07
1	F	1	3.061538e+07
0	F	0	4.381501e+07

[33]:
```
sns.barplot(x=maritial_sort['Marital_Status'],y=maritial_sort["Amount"],␣
  ↪hue="Gender",data=maritial_sort)
```

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



- Married Female have more Purchasing Power than Married

[34]:
```
df["Product_Category"].duplicated(keep="last").info()
```

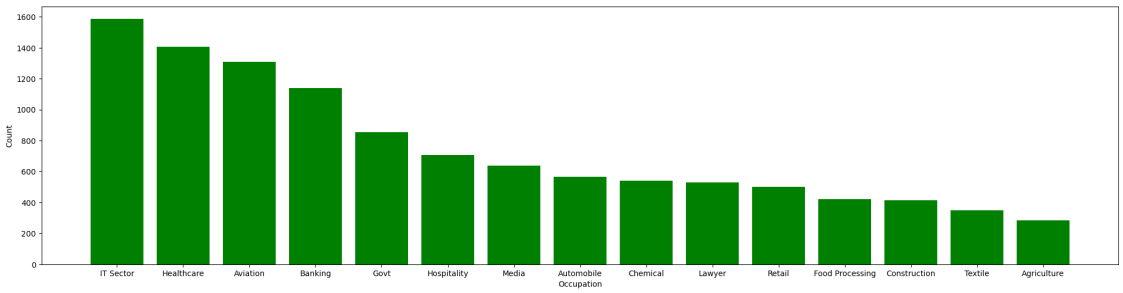
```
<class 'pandas.core.series.Series'>
RangeIndex: 11251 entries, 0 to 11250
Series name: Product_Category
Non-Null Count  Dtype
--------------  -----
11251 non-null  bool
dtypes: bool(1)
memory usage: 11.1 KB
```

0.4 Occupation

[38]:
```
new_occupation=df.groupby(["Occupation","Age␣
↪Group","Gender"],as_index=False)["Amount"].sum().
↪sort_values(by="Amount",ascending=False)
new_occupation.head(4)
```

[38]:
```
     Occupation Age Group Gender      Amount
144   IT Sector     26-35      F  4392550.00
116  Healthcare     26-35      F  3896548.50
32     Aviation     26-35      F  3853727.00
46      Banking     26-35      F  3516138.45
```

[36]:
```
plt.figure(figsize=(25,6))
counts=df["Occupation"].value_counts()
plt.bar(counts.index,counts.values,color="green")
plt.xlabel("Occupation")
plt.ylabel("Count")
plt.show()
```



- From the graph it can be visually seen that F(26-35) work in IT-Sector have more purchasing then other age categories woman

[37]:
```
df.columns
```

[37]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

[40]:
```
count=df["Product_Category"].value_counts()

count
```

[40]:
```
Product_Category
Clothing & Apparel       2655
Food                     2493
Electronics & Gadgets    2087
Footwear & Shoes         1064
Household items           520
Beauty                    422
Games & Toys              386
Sports Products           356
```

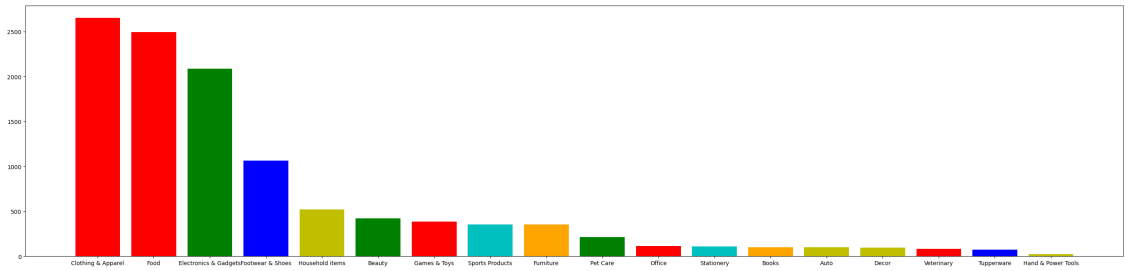
```
Furniture                353
Pet Care                 212
Office                   113
Stationery               112
Books                    103
Auto                     100
Decor                     96
Veterinary                81
Tupperware                72
Hand & Power Tools        26
Name: count, dtype: int64
```

[41]:
```
count.size
```

[41]:
```
18
```

[42]:
```
plt.figure(figsize=(35,8))

plt.bar(count.index,count.
  ↪values,color=["r","r","g","b","y","g","r","c","orange","g","r","c","orange","y","y","r","b"
```

[42]:
```
<BarContainer object of 18 artists>
```



- From the Above Graph the Married(26-35) Woman working in IT-Sector Shows great intrest in Clothing and Apparel

[43]:
```
df.columns
```

[43]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

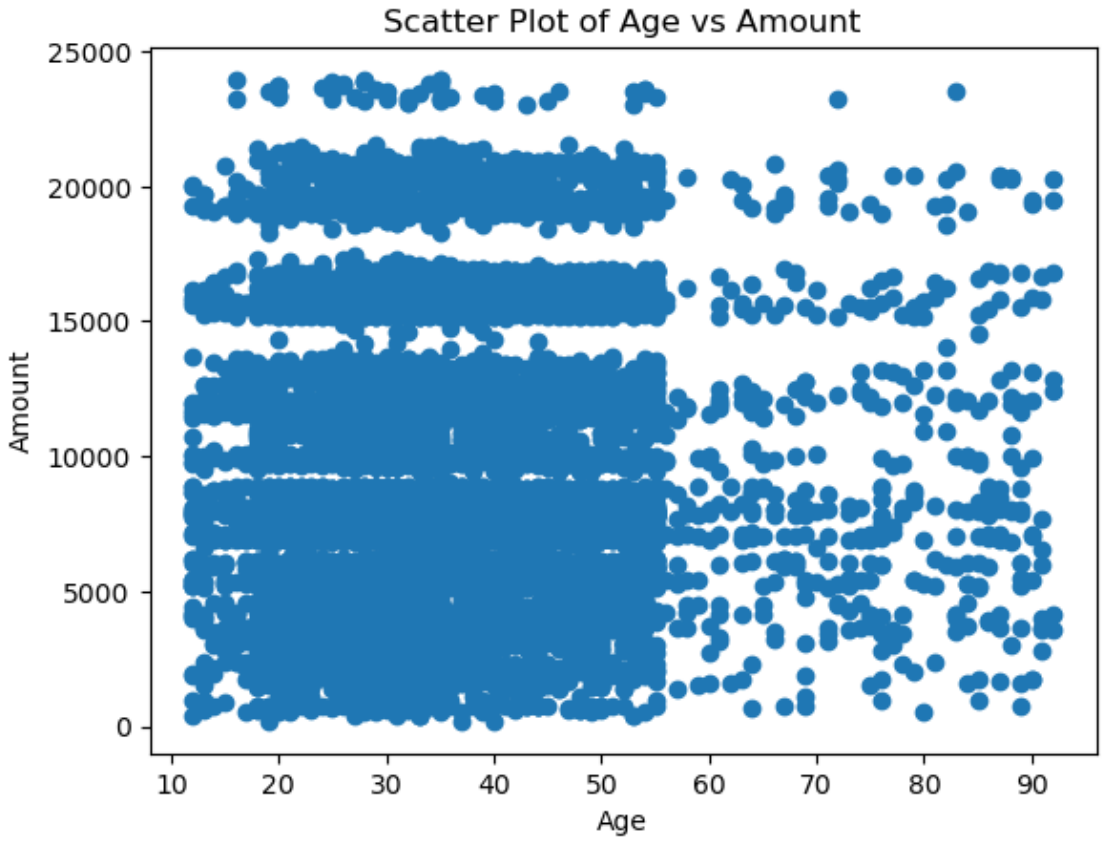
[44]:
```
correlation=df[["Age","Amount"]].corr()
```

[45]:
```
correlation
```

[45]:

	Age	Amount
Age	1.000000	0.030924
Amount	0.030924	1.000000

[46]:
```
plt.scatter(df["Age"],df["Amount"])
plt.xlabel('Age')
plt.ylabel('Amount')
plt.title('Scatter Plot of Age vs Amount')
plt.show()
```



[47]:
```
grouped = df.groupby(['Gender', 'Age Group', 'Product_Category']).size().
  ↪reset_index(name='Count')
most_purchased = grouped.loc[grouped.groupby(['Gender', 'Age Group'])['Count'].
  ↪idxmax()].reset_index(drop=True)
most_purchased
```

[47]:

	Gender	Age Group	Product_Category	Count
0	F	0-17	Food	42
1	F	18-25	Food	344
2	F	26-35	Clothing & Apparel	746

3	F	36-45	Clothing & Apparel	379
4	F	46-50	Clothing & Apparel	166
5	F	51-55	Electronics & Gadgets	126
6	F	55+	Clothing & Apparel	81
7	M	0-17	Food	41
8	M	18-25	Clothing & Apparel	146
9	M	26-35	Clothing & Apparel	311
10	M	36-45	Clothing & Apparel	153
11	M	46-50	Clothing & Apparel	68
12	M	51-55	Clothing & Apparel	66
13	M	55+	Footwear & Shoes	51

- The insight is Married women of age group (26-35) years from UP working in IT Sector, have more purchase of Clothing and Apparel Products