In [1]:
```
# Import Required Packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# Import .csv file
df = pd.read_csv("Diwali_Sales_Data.csv")
df
```

Out[2]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	
0	1000545	Aditya Yadav	5805	M	18-25	22	Unmarried	Maha
1	1003759	Suman Mishra	8611	F	18-25	20	Married	Andhra P
2	1004167	Anita Mishra	6349	F	46-55	46	Married	Uttar P
3	1004402	Amit Chopra	1267	M	36-45	41	Married	Kar
4	1001439	Kavya Iyer	4667	F	26-35	30	Married	(
...	...	...	...	...	...	...	...	
11246	1006176	Aditya Mehta	9161	M	36-45	42	Unmarried	Maha
11247	1003398	Rahul Joshi	3600	M	26-35	31	Married	H
11248	1005049	Anita Singh	9327	F	36-45	43	Married	M P
11249	1006495	Pooja Reddy	5801	F	36-45	43	Married	Kar
11250	1002811	Suman Patel	9821	F	26-35	35	Married	Maha

11251 rows × 15 columns

In [3]:
```
# Too See Top 15 Data
df.head(15)
```

Out[3]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	Sta
0	1000545	Aditya Yadav	5805	M	18-25	22	Unmarried	Maharash
1	1003759	Suman Mishra	8611	F	18-25	20	Married	Andhra Prade
2	1004167	Anita Mishra	6349	F	46-55	46	Married	Uttar Prade
3	1004402	Amit Chopra	1267	M	36-45	41	Married	Karnata
4	1001439	Kavya Iyer	4667	F	26-35	30	Married	Guja
5	1000831	Karan Nair	3710	M	18-25	24	Married	Himac Prade
6	1006417	Riya Mishra	7071	F	36-45	45	Unmarried	Uttar Prade
7	1002895	Rahul Reddy	7689	M	46-55	46	Married	Maharash
8	1002675	Amit Kapoor	4932	M	18-25	25	Unmarried	Uttar Prade
9	1001467	Mohit Mishra	5318	M	36-45	41	Unmarried	Andhra Prade
10	1006767	Pooja Patel	5486	F	46-55	47	Married	De
11	1002772	Rahul Iyer	1100	M	36-45	37	Married	Andhra Prade
12	1004996	Vikas Mehta	5008	M	46-55	53	Married	Andhra Prade
13	1005313	Vikas Mishra	3277	M	26-35	35	Married	Andhra Prade
14	1006370	Raj Kapoor	8755	M	26-35	34	Unmarried	Madh Prade

In [4]:
```
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  int64 
 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  object
 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            11251 non-null  int64 
 13  Status            11251 non-null  object
 14  unnamed1          11251 non-null  int64 
dtypes: int64(6), object(9)
memory usage: 1.3+ MB
```

In [6]:
```
# Drop Column Here axis=1 Means Remove Entire Rows Of The Column
df.drop(['unnamed1'],axis=1,inplace = True)
```

In [11]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11251 entries, 0 to 11250
Data columns (total 14 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  int64 
 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  object
 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            11251 non-null  int64 
 13  Status            11251 non-null  object
dtypes: int64(5), object(9)
memory usage: 1.2+ MB
```

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

Out[13]:
```
User_ID             0
Cust_name           0
Product_ID          0
Gender              0
Age Group           0
Age                 0
Marital_Status      0
State               0
Zone                0
Occupation          0
Product_Category    0
Orders              0
Amount              0
Status              0
dtype: int64
```

In [15]:
```
df.shape
```

Out[15]:
```
(11251, 14)
```

In [17]:
```
# Column Rename Or Change The Name Of Columns
df.rename(columns = {'Orders':'Total_Orders'})
```

Out[17]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	
0	1000545	Aditya Yadav	5805	M	18-25	22	Unmarried	Maha
1	1003759	Suman Mishra	8611	F	18-25	20	Married	Andhra P
2	1004167	Anita Mishra	6349	F	46-55	46	Married	Uttar P
3	1004402	Amit Chopra	1267	M	36-45	41	Married	Kar
4	1001439	Kavya Iyer	4667	F	26-35	30	Married	(
...	...	...	...	...	...	...	...	
11246	1006176	Aditya Mehta	9161	M	36-45	42	Unmarried	Maha
11247	1003398	Rahul Joshi	3600	M	26-35	31	Married	H
11248	1005049	Anita Singh	9327	F	36-45	43	Married	M P
11249	1006495	Pooja Reddy	5801	F	36-45	43	Married	Kar
11250	1002811	Suman Patel	9821	F	26-35	35	Married	Maha

11251 rows × 14 columns

In [19]:
```
df.describe()
```

Out[19]:

	User_ID	Product_ID	Age	Orders	Amount
count	1.125100e+04	11251.000000	11251.000000	11251.000000	11251.000000
mean	1.003530e+06	5502.008799	39.999911	4.995200	25108.864990
std	2.031925e+03	2581.794650	11.164933	2.590302	14260.396693
min	1.000001e+06	1001.000000	20.000000	1.000000	508.000000
25%	1.001760e+06	3261.500000	30.000000	3.000000	12822.000000
50%	1.003517e+06	5513.000000	40.000000	5.000000	24860.000000
75%	1.005292e+06	7719.500000	49.000000	7.000000	37487.000000
max	1.007040e+06	9997.000000	60.000000	9.000000	49996.000000

In [21]:
```
# Use describe() for specific columns
df[['Age','Orders','Amount']].describe()
```

Out[21]:

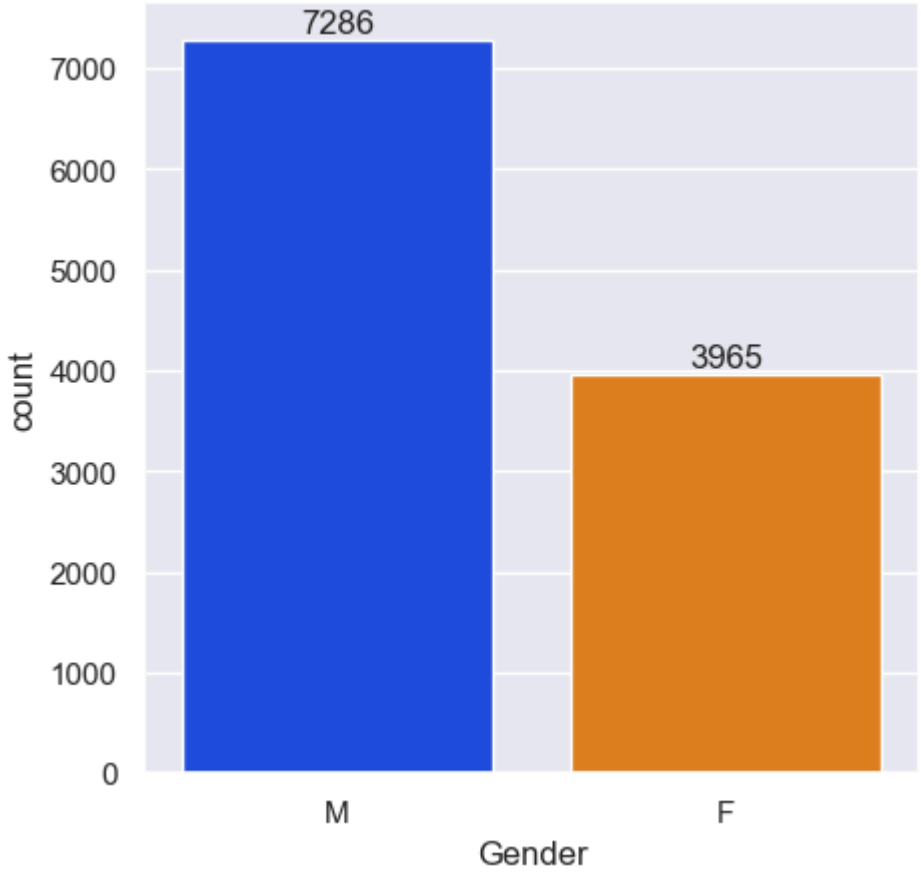
	Age	Orders	Amount
count	11251.000000	11251.000000	11251.000000
mean	39.999911	4.995200	25108.864990
std	11.164933	2.590302	14260.396693
min	20.000000	1.000000	508.000000
25%	30.000000	3.000000	12822.000000
50%	40.000000	5.000000	24860.000000
75%	49.000000	7.000000	37487.000000
max	60.000000	9.000000	49996.000000

In [23]:
```
# To See All Columns
df.columns
```

Out[23]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount', 'Status'],
      dtype='object')
```

Exploratory Data Analysis

In [203…
```
ax = sns.countplot(x='Gender', data=df,hue='Gender',palette='bright')

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

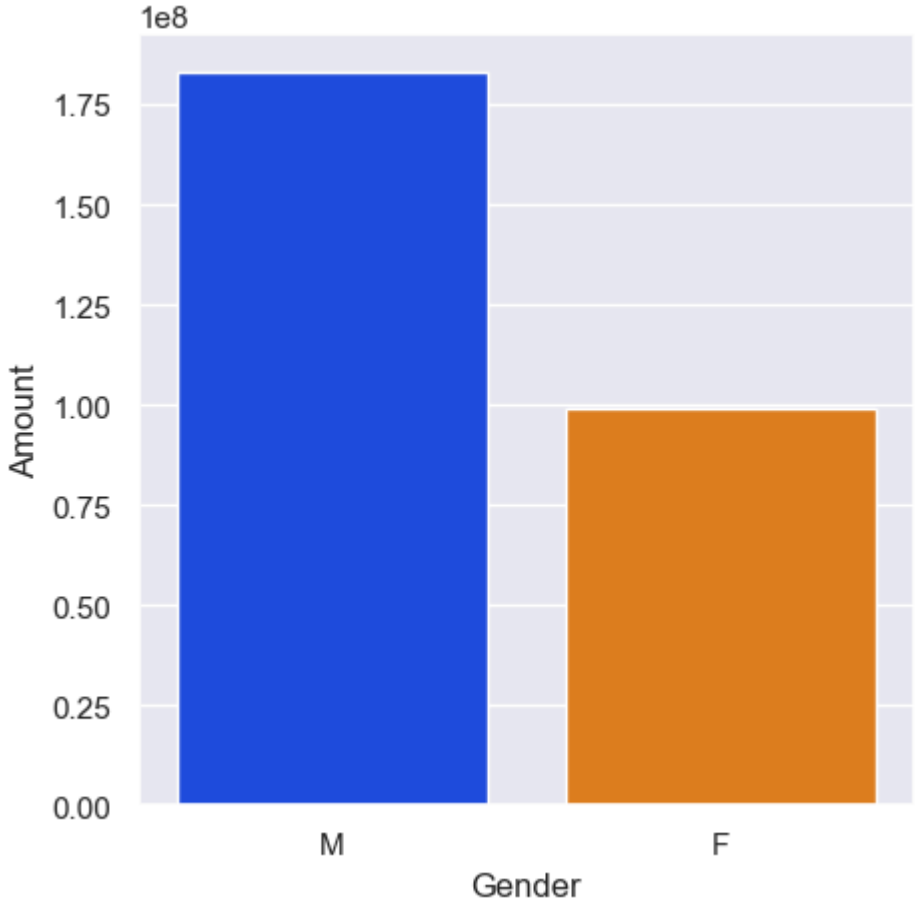


In [28]:
```
sales_gen = df.groupby(['Gender'],as_index=False)['Amount'].sum().sort_values(by
sales_gen
```

Out[28]:

	Gender	Amount
1	M	183234947
0	F	99264893

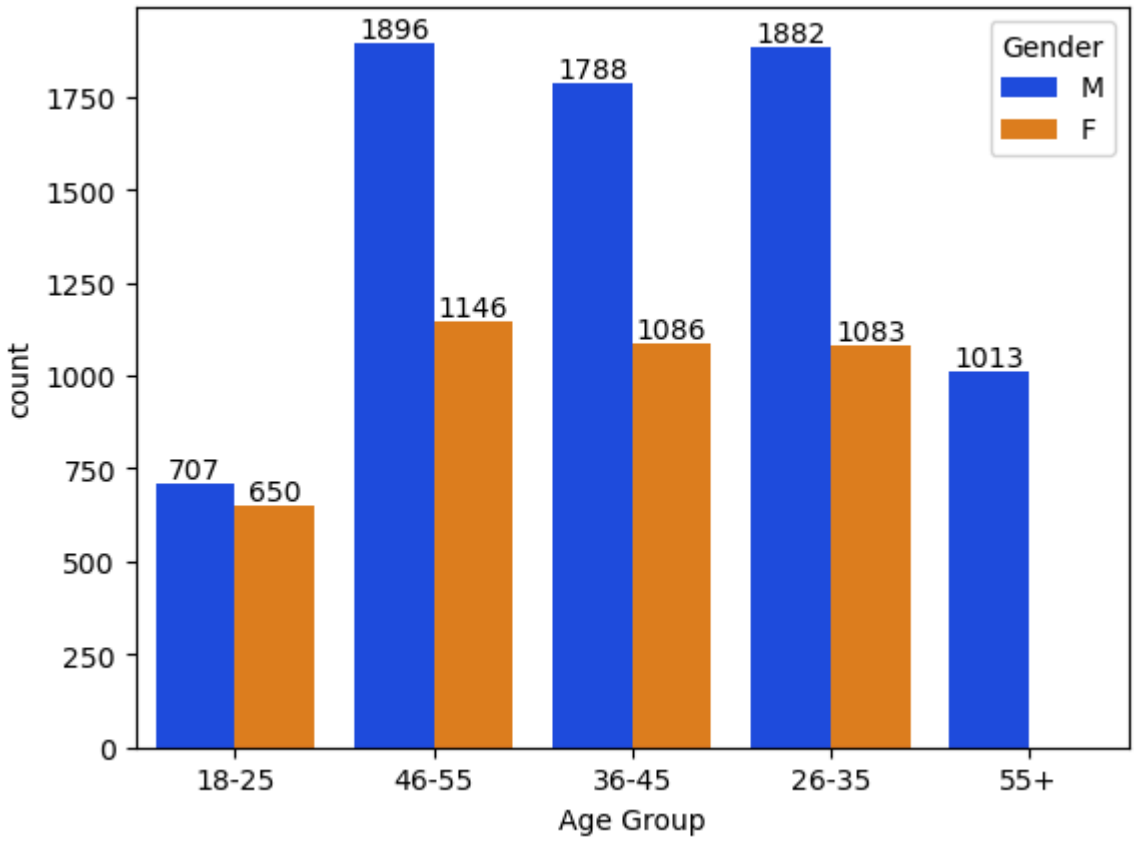
In [207…
```
sns.barplot(x='Gender',y='Amount',data=sales_gen,hue='Gender',palette='bright')
sns.set(rc={'figure.figsize':(5,5)})
```



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

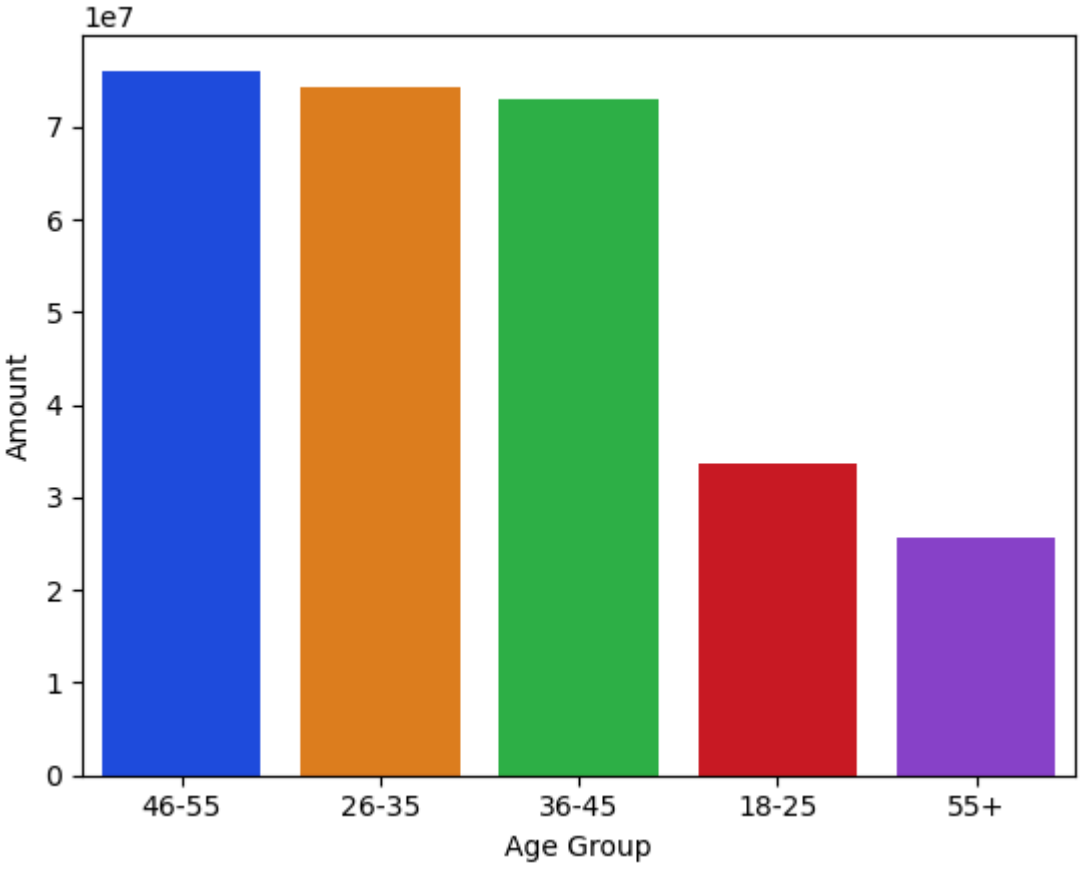
AGE

In [51]:
```
ax=sns.countplot(data=df, x='Age Group', hue='Gender',palette='bright')

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



In [53]:
```
sales_age=df.groupby(['Age Group'], as_index=False)['Amount'].sum().sort_values(

sns.barplot(x='Age Group',y='Amount' ,data=sales_age,hue='Age Group',palette='br
```

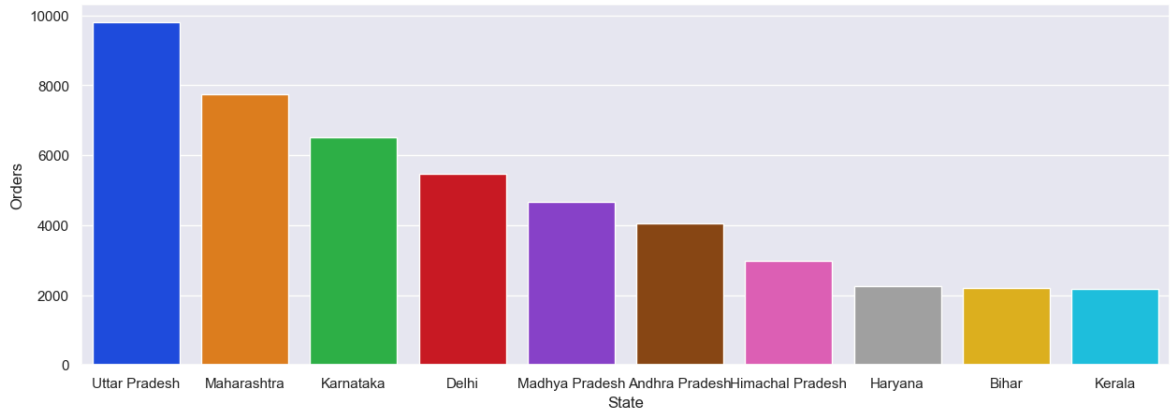
Out[53]: <Axes: xlabel='Age Group', ylabel='Amount'>



From above graphs we can see that most of the buyers are of age group between 46-55 yrs Male

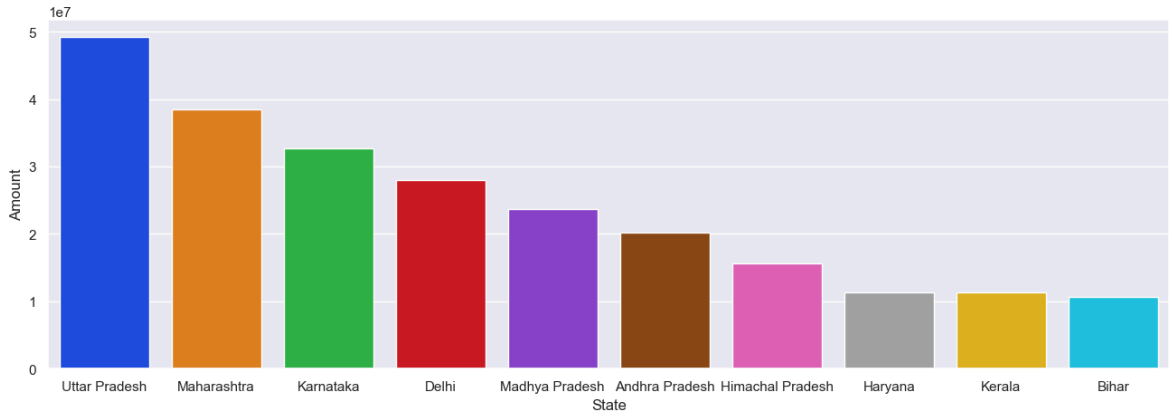
In [57]:
```
# Total Number of Orders from top 10 States
sales_state=df.groupby(['State'], as_index=False)['Orders'].sum().sort_values(by

sns.set(rc={'figure.figsize':(15,5)})
sns.barplot(data=sales_state, x='State',y='Orders',hue='State',palette='bright')
```

Out[57]: <Axes: xlabel='State', ylabel='Orders'>



In [59]:
```
# Total Amount/Sales from top 10 States
sales_state=df.groupby(['State'], as_index=False)['Amount'].sum().sort_values(by

sns.set(rc={'figure.figsize':(16,5)})
sns.barplot(data=sales_state, x='State',y='Amount',hue='State',palette='bright')
```

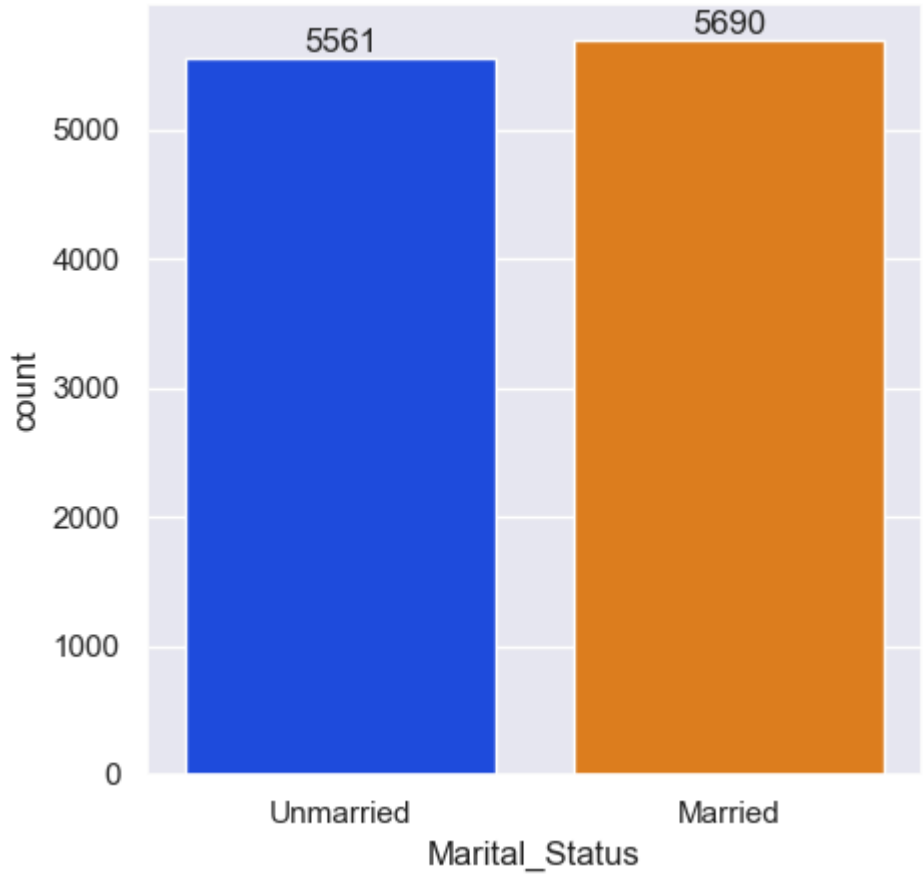
Out[59]: <Axes: xlabel='State', ylabel='Amount'>



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

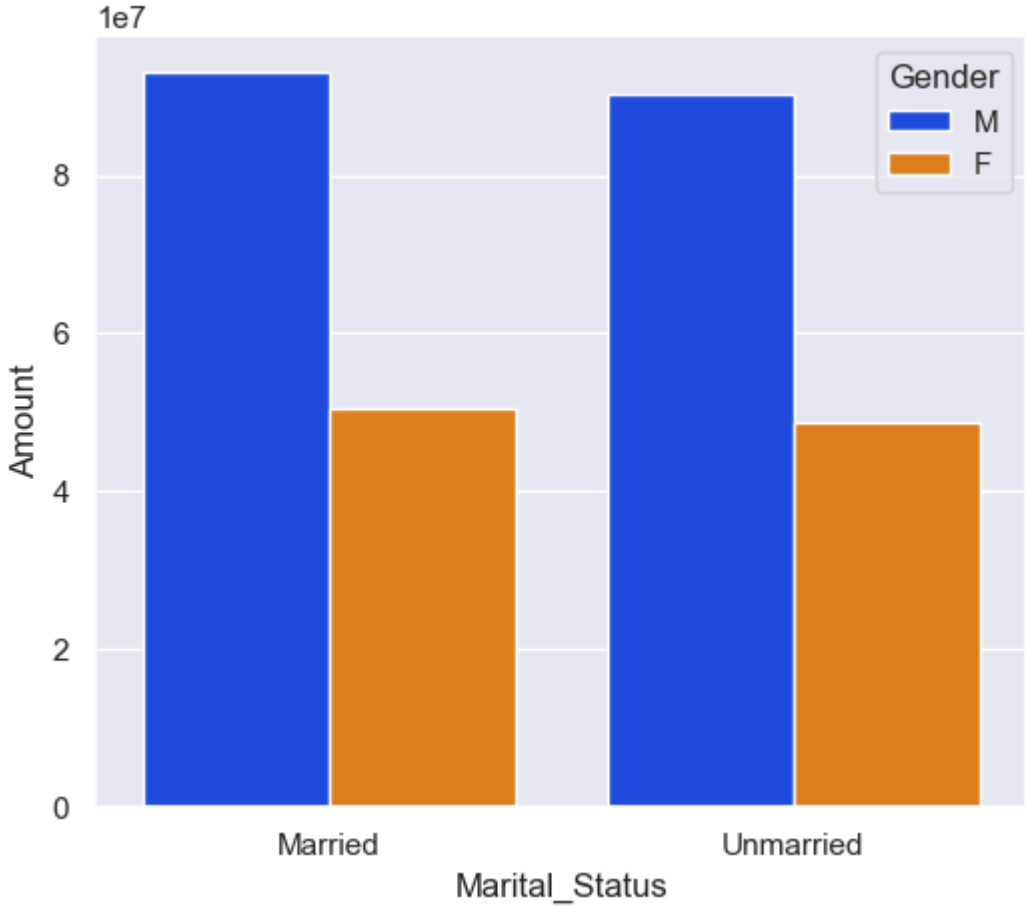
Marital Status

In [209…
```
ax = sns.countplot(data=df,x='Marital_Status',hue='Marital_Status',palette='brig

sns.set(rc={'figure.figsize':(4,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



In [78]:
```
# Total Amount/Sales from top 10 States
sales_state=df.groupby(['Marital_Status','Gender'], as_index=False)['Amount'].su

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

Out[78]: <Axes: xlabel='Marital_Status', ylabel='Amount'>

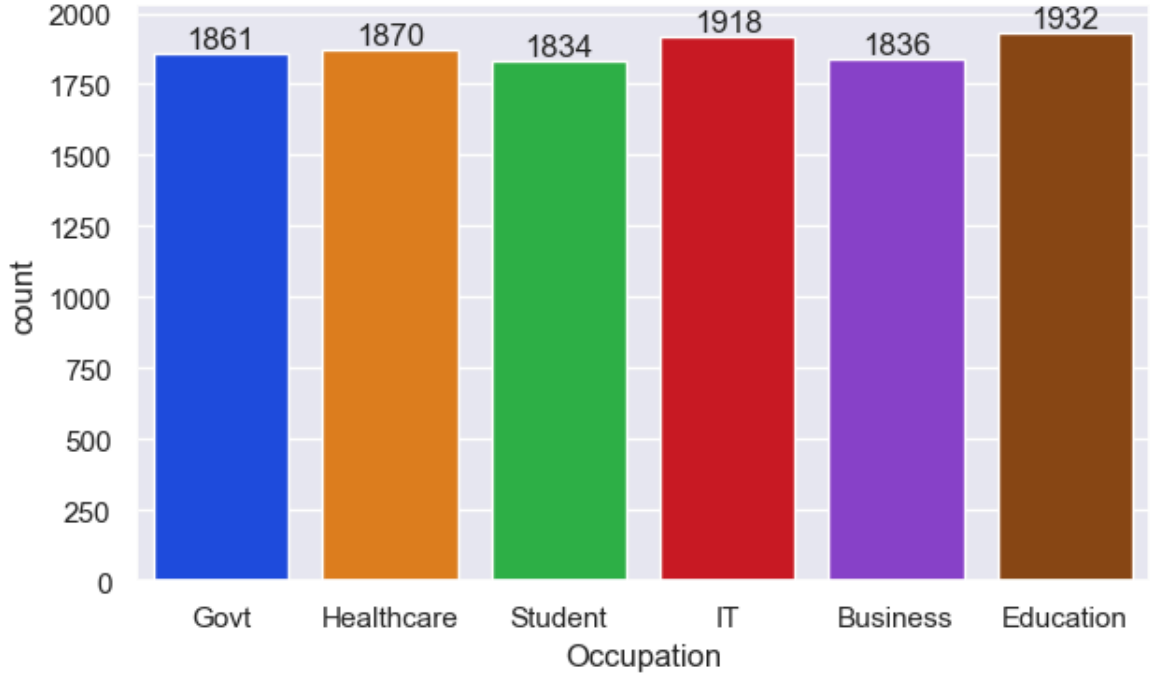


From above graphs we can see that most of the buyers are married (Male) and they have high purchasing power

Occupation

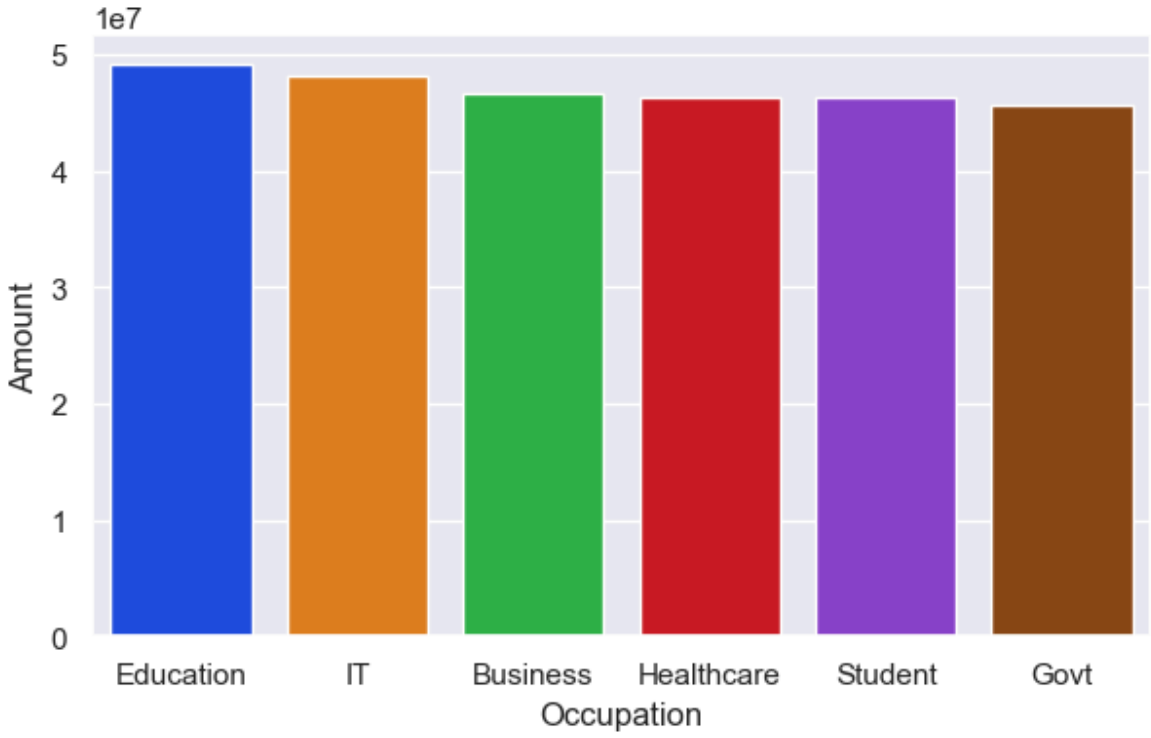
In [114…

```
sns.set(rc={'figure.figsize':(7,4)})
ax = sns.countplot(data = df, x = 'Occupation',hue='Occupation',palette='bright'

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



In [116…
```
sales_state = df.groupby(['Occupation'], as_index=False)['Amount'].sum().sort_va

sns.set(rc={'figure.figsize':(7,4)})
sns.barplot(data = sales_state, x = 'Occupation',y= 'Amount',hue='Occupation',pa
```

Out[116… `<Axes: xlabel='Occupation', ylabel='Amount'>`

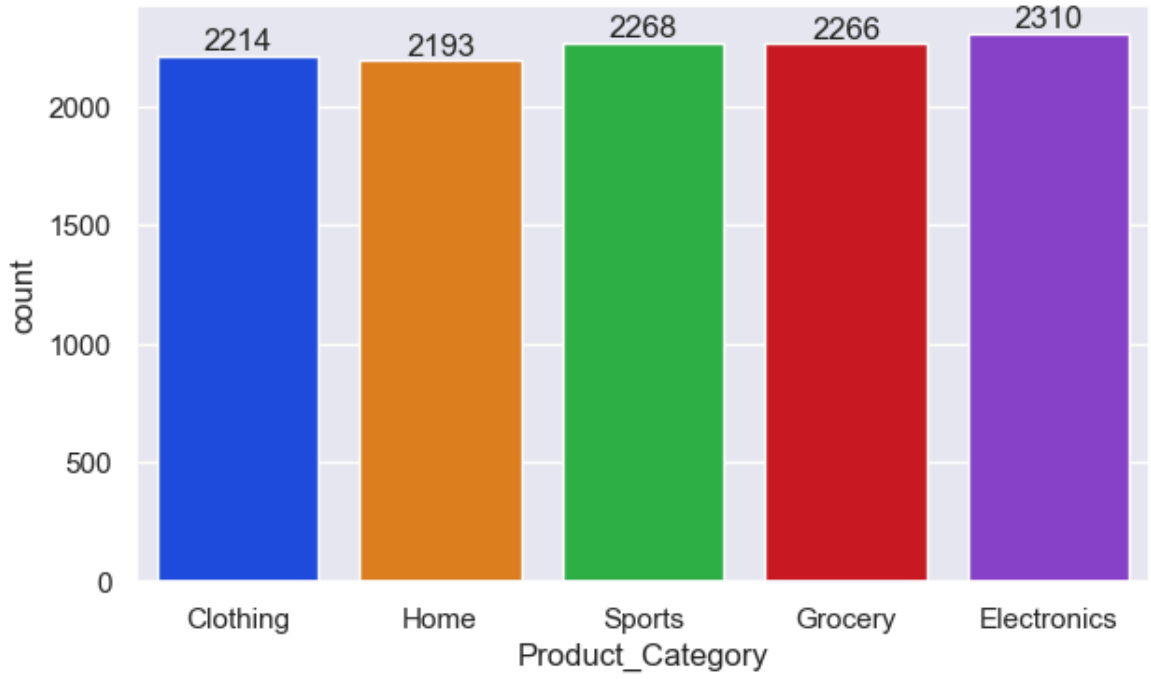


From above graphs we can see that most of the buyers are working in Education, IT and Business Sector

Product Category

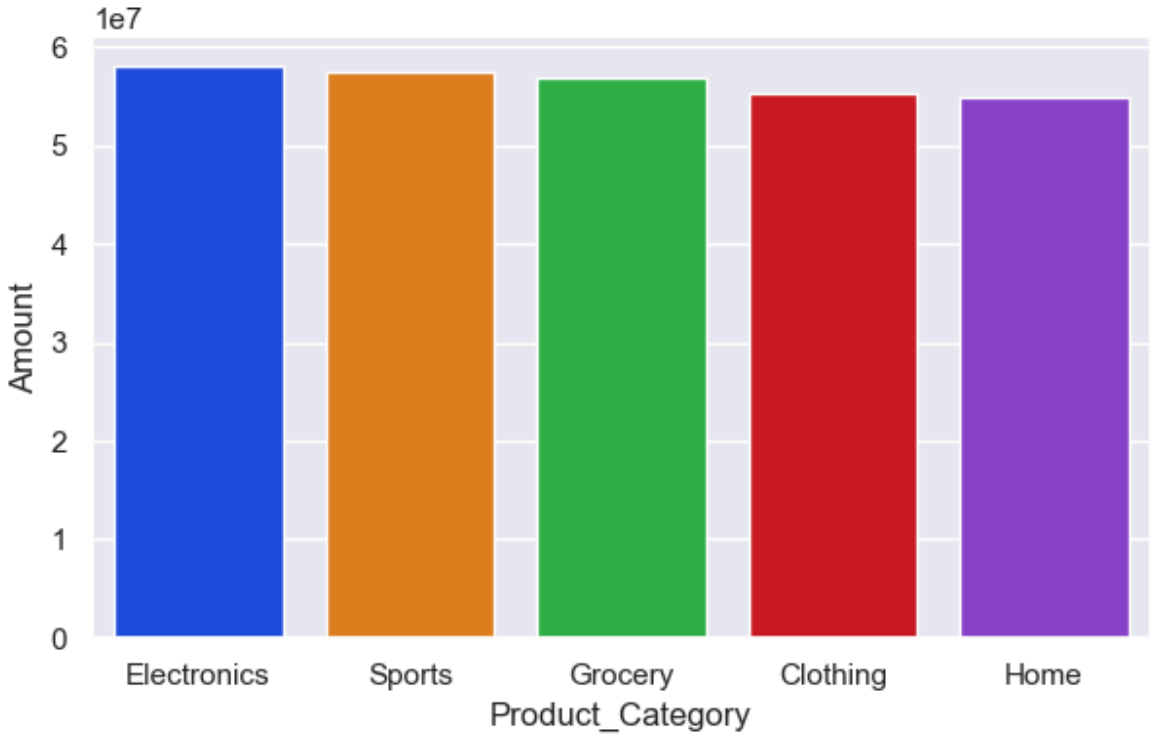
In [118…
```
sns.set(rc={'figure.figsize':(7,4)})
ax = sns.countplot(data = df, x = 'Product_Category',hue='Product_Category',pale
```

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



In [122…
```
sales_state = df.groupby(['Product_Category'], as_index=False)['Amount'].sum().s

sns.set(rc={'figure.figsize':(7,4)})
sns.barplot(data = sales_state, x = 'Product_Category',y= 'Amount',hue='Product_
```

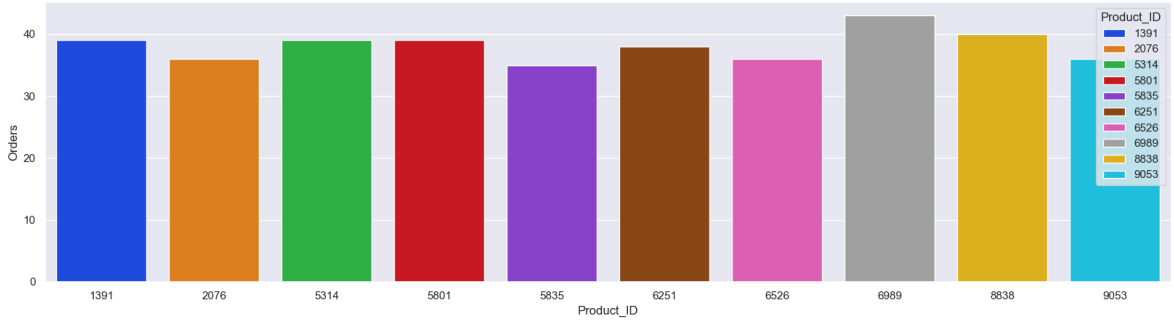
Out[122…
```
<Axes: xlabel='Product_Category', ylabel='Amount'>
```



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

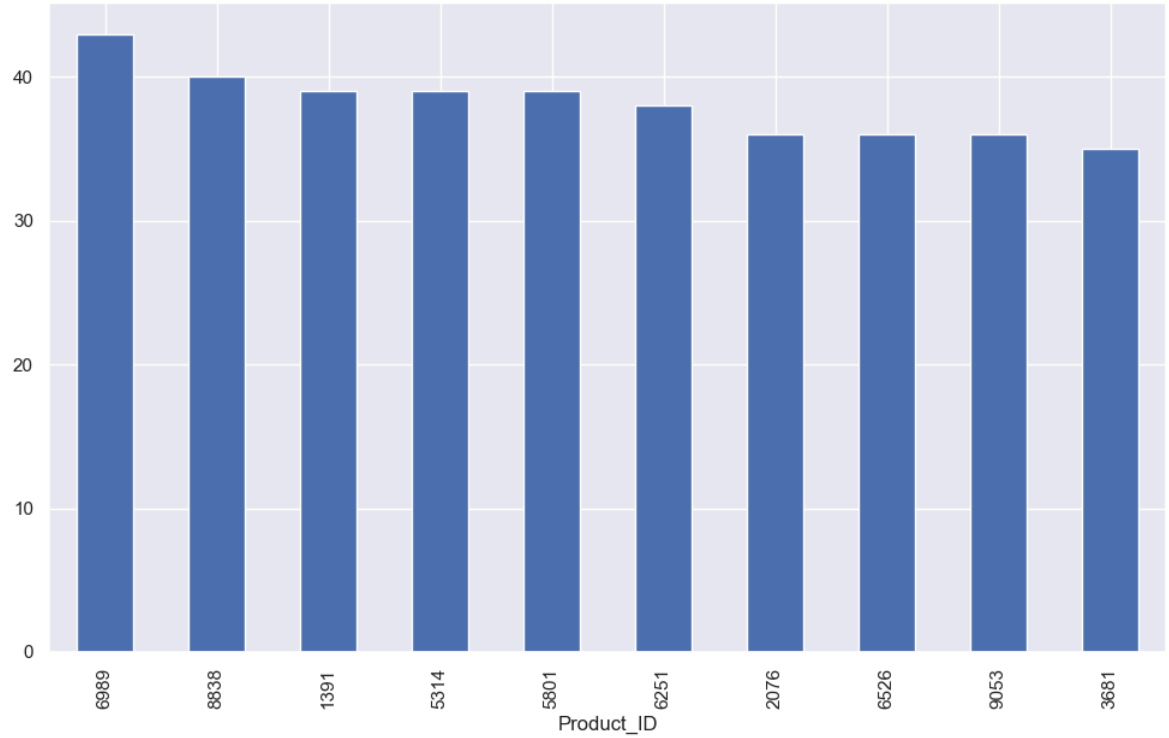
In [124…
```
sales_state = df.groupby(['Product_ID'], as_index=False)['Orders'].sum().sort_va

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

Out[124… <Axes: xlabel='Product_ID', ylabel='Orders'>



In [161…
```
# top 10 most sold products (same thing as above)

fig1, ax1 = plt.subplots(figsize=(12,7))
df.groupby('Product_ID')['Orders'].sum().nlargest(10).sort_values(ascending=Fals
```

Out[161… <Axes: xlabel='Product_ID'>



Conclusion:

Married Men age group 46-55 yrs from Uttar Pradesh, Maharashtra and Karnataka working in Education,IT and Business are more likely to buy products from Electronics, Sports and Grocery category.