In [6]:

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

In [8]:

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

In [10]:

```python
df.shape #rows and columns return
```

Out[10]:

```
(11251, 15)
```

In [12]:

```python
df.head(10) #returns given numbers of rows
```

Out[12]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	
0	1002903	Sanskriti	P00125942	F	26-35	28	0	M
1	1000732	Kartik	P00110942	F	26-35	35	1	Andh
2	1001990	Bindu	P00118542	F	26-35	35	1	Ut
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	Ut
7	1002092	Shivangi	P00273442	F	55+	61	0	M
8	1003224	Kushal	P00205642	M	26-35	35	0	Ut
9	1003650	Ginny	P00031142	F	26-35	26	1	Andh

In [13]:

```python
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
```

In [14]:
```
df.drop(['Status','unnamed1'],axis=1,inplace=True)
```

In [15]:
```
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 [16]:
```
pd.isnull(df)
```

Out[16]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...
11246	False	False	False	False	False	False	False
11247	False	False	False	False	False	False	False
11248	False	False	False	False	False	False	False
11249	False	False	False	False	False	False	False
11250	False	False	False	False	False	False	False

11251 rows × 13 columns

In [17]:

```python
pd.isnull(df).sum()
```

Out[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              12
dtype: int64
```

In [21]:

```python
df.dropna(inplace=True) #inplace are used to save the changes permenetly
```

In [22]:

```python
pd.isnull(df).sum()
```

Loading [MathJax]/extensions/Safe.js

```
Out[22]:  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
```

```
In [29]:  df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 11239 entries, 0 to 11250
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   User_ID           11239 non-null  int64 
 1   Cust_name         11239 non-null  object
 2   Product_ID        11239 non-null  object
 3   Gender            11239 non-null  object
 4   Age Group         11239 non-null  object
 5   Age               11239 non-null  int64 
 6   Marital_Status    11239 non-null  int64 
 7   State             11239 non-null  object
 8   Zone              11239 non-null  object
 9   Occupation        11239 non-null  object
 10  Product_Category  11239 non-null  object
 11  Orders            11239 non-null  int64 
 12  Amount            11239 non-null  int64 
dtypes: int64(5), object(8)
memory usage: 1.2+ MB
```

```
In [24]:  #change data type
          df['Amount'] = df['Amount'].astype('int')
```

```
In [25]:  df['Amount'].dtype
```

```
Out[25]:  dtype('int64')
```

```
In [26]:  df.columns
```

```
Out[26]:  Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
                 'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
                 'Orders', 'Amount'],
                dtype='object')
```

```
In [27]:  df.describe()
```

Out[27]:

	User_ID	Age	Marital_Status	Orders	Amoun
count	1.123900e+04	11239.000000	11239.000000	11239.000000	11239.00000
mean	1.003004e+06	35.410357	0.420055	2.489634	9453.61055
std	1.716039e+03	12.753866	0.493589	1.114967	5222.35516
min	1.000001e+06	12.000000	0.000000	1.000000	188.00000
25%	1.001492e+06	27.000000	0.000000	2.000000	5443.00000
50%	1.003064e+06	33.000000	0.000000	2.000000	8109.00000
75%	1.004426e+06	43.000000	1.000000	3.000000	12675.00000
max	1.006040e+06	92.000000	1.000000	4.000000	23952.00000

In [28]:

```
df[['Age','Orders','Amount']].describe()
```

Out[28]:

	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

Exploratory Data Analysis

- Gender

In [30]:

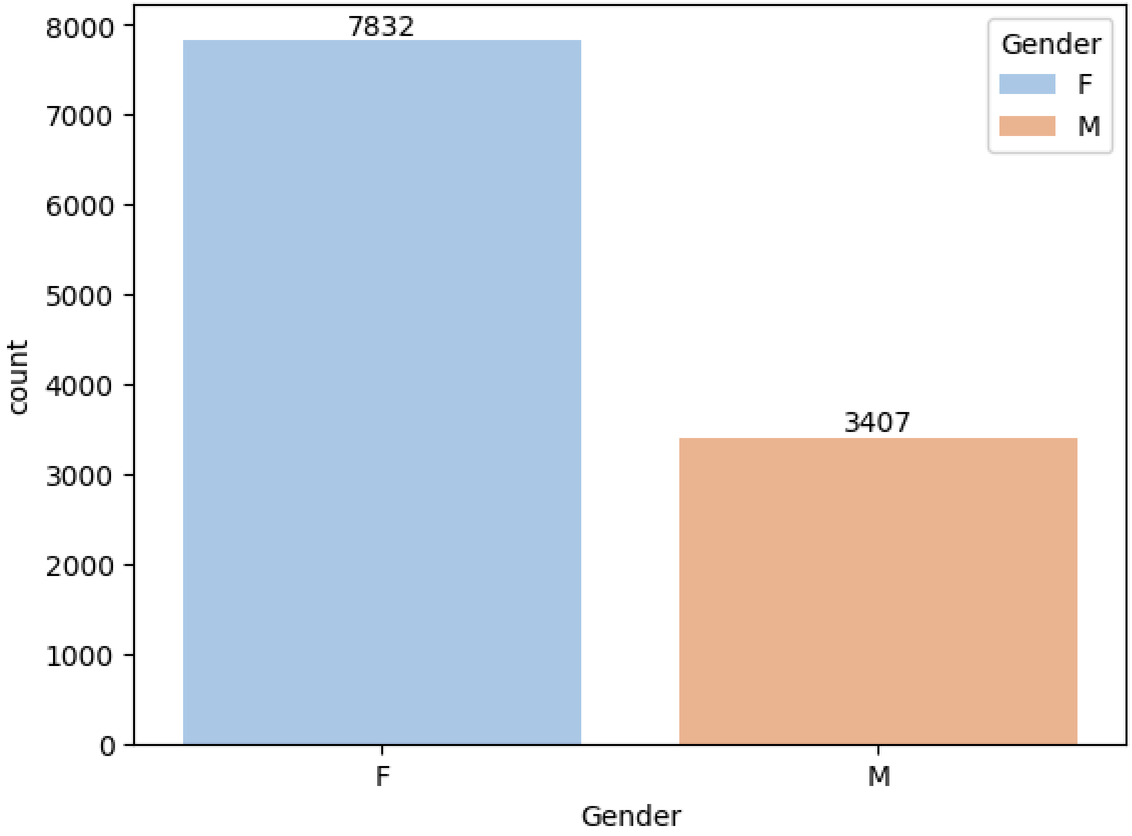
```
df.columns
```

Out[30]:

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

In [49]:

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

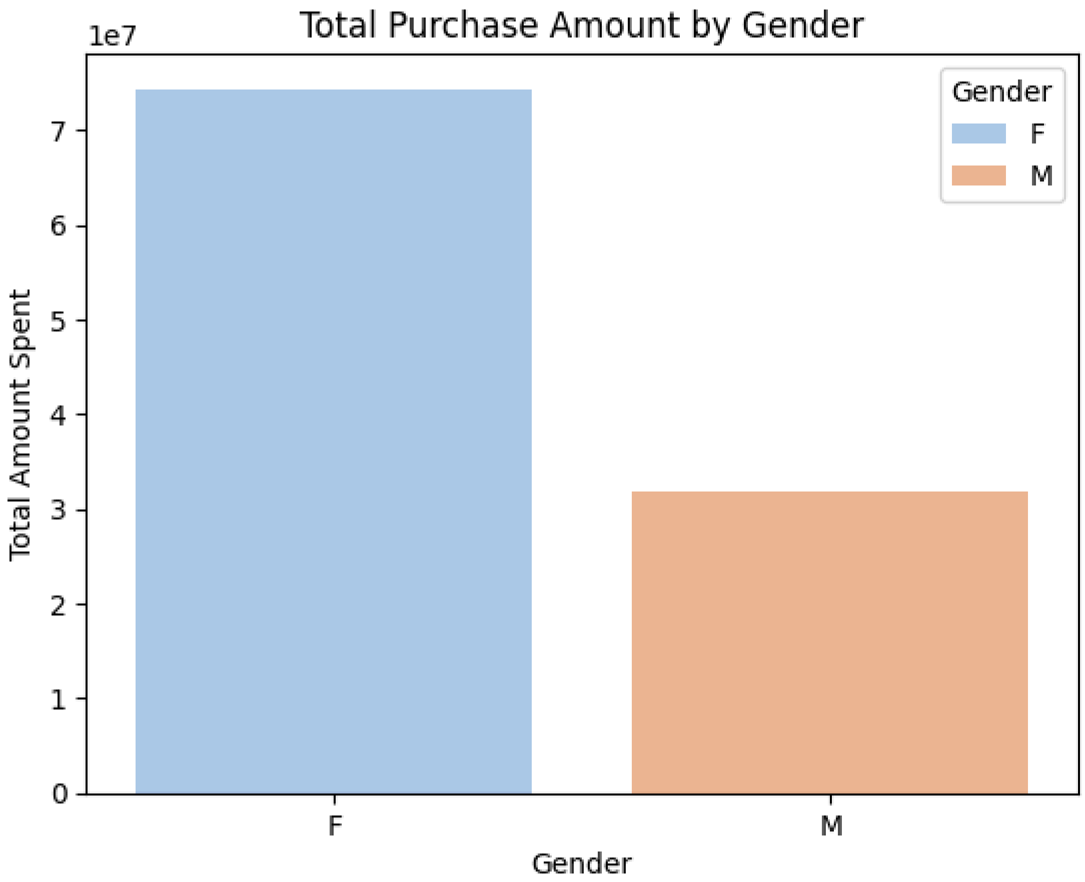


In [50]:

```
sales_gen = df.groupby(['Gender'],as_index=False)['Amount'].sum().sort_value
ax = sns.barplot(x='Gender',y='Amount',data=sales_gen,hue='Gender', palette=
plt.title('Total Purchase Amount by Gender')
plt.ylabel('Total Amount Spent')
plt.xlabel('Gender')
sales_gen
```

Out[50]:

	Gender	Amount
0	F	74335853
1	M	31913276



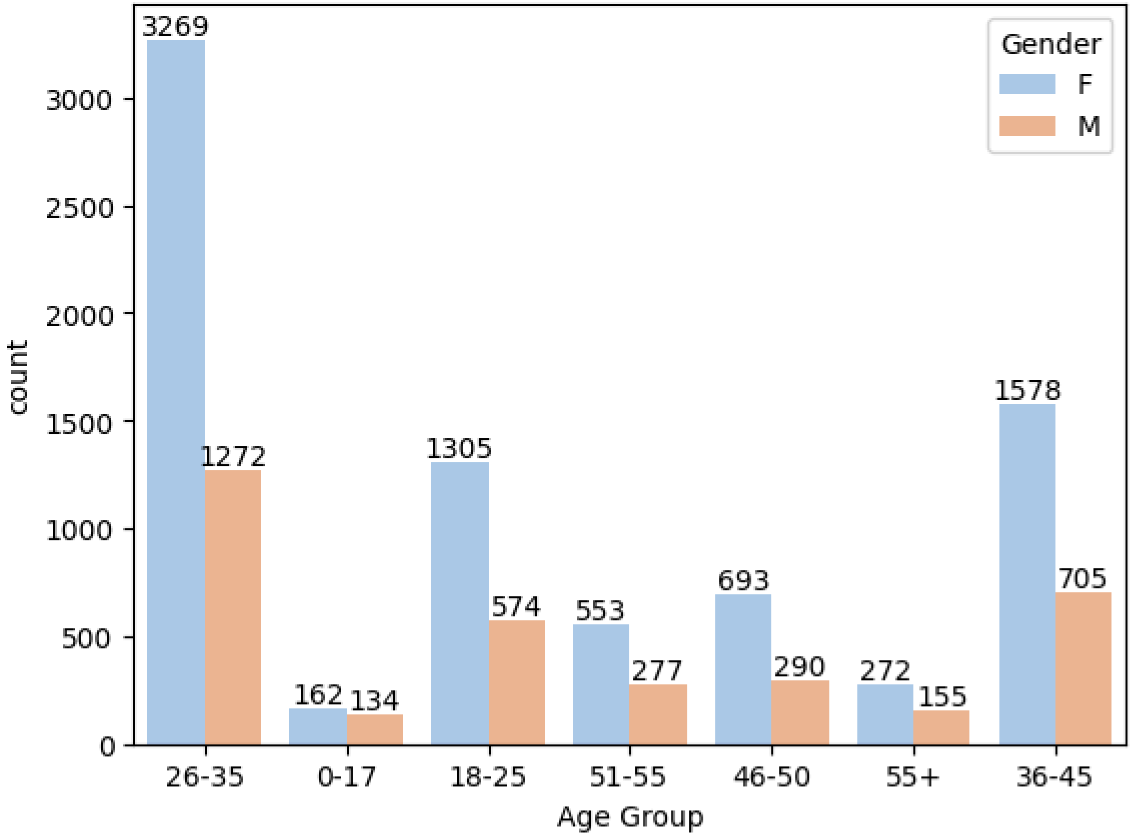
From above graphs we can see that most of the buyers are female and even the purchasisng power of females are greater than men

- Age

In [51]:
```
df.columns
```

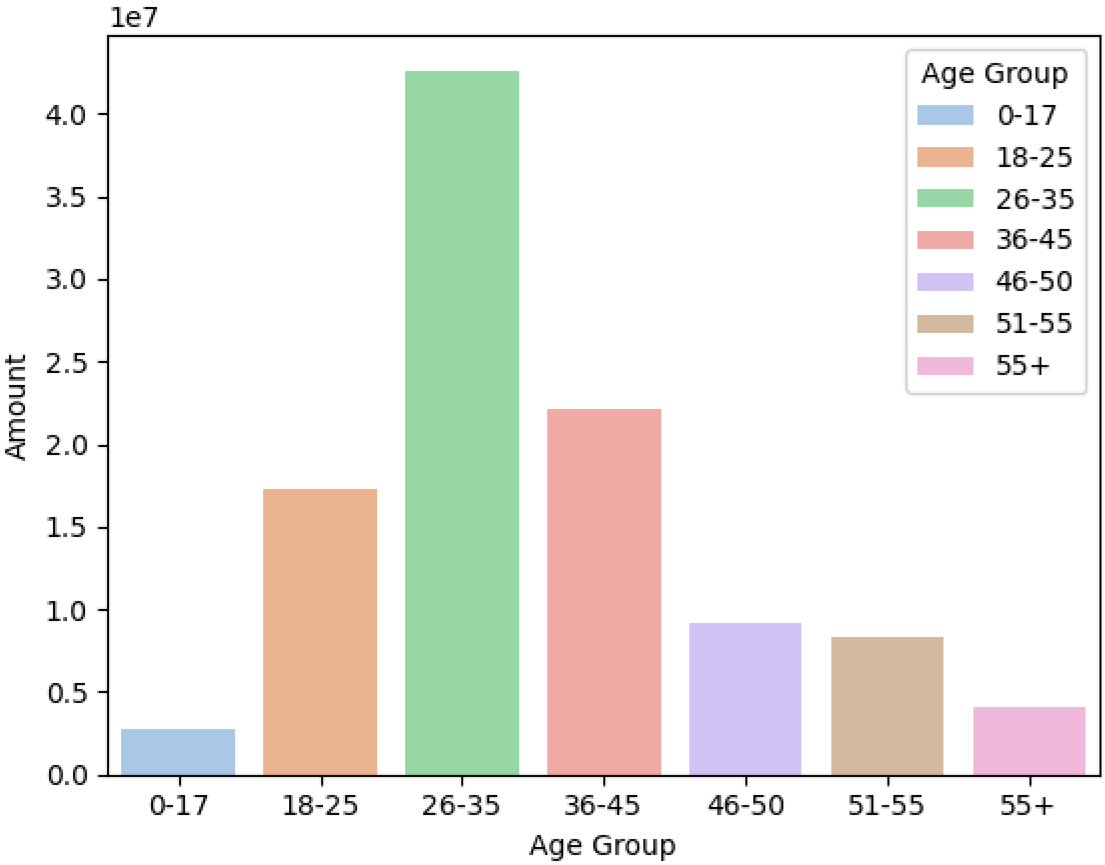
Out[51]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [55]:
```
ax = sns.countplot(data=df,x='Age Group',hue='Gender',palette='pastel')
for bars in ax.containers:
    ax.bar_label(bars)
```



In [66]:
```
sales_age = df.groupby(['Age Group'], as_index=False)['Amount'].sum()#.sort_
sns.barplot(x='Age Group', y='Amount', data=sales_age,palette='pastel',hue='
```

Out[66]:
```
<Axes: xlabel='Age Group', ylabel='Amount'>
```



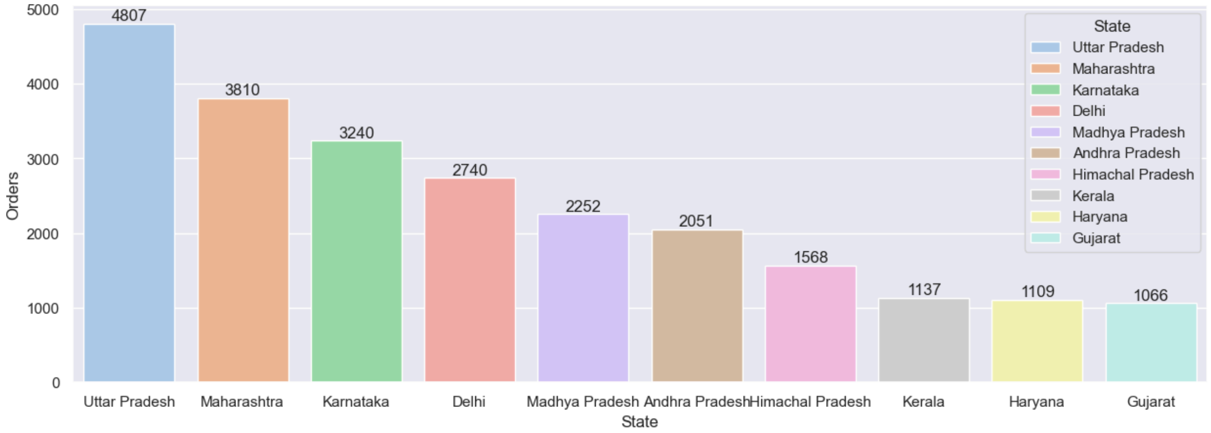
from above graphs we can see that most of the buyers are of age group between 26-35 year female

- State

In [67]:
```
df.columns
```

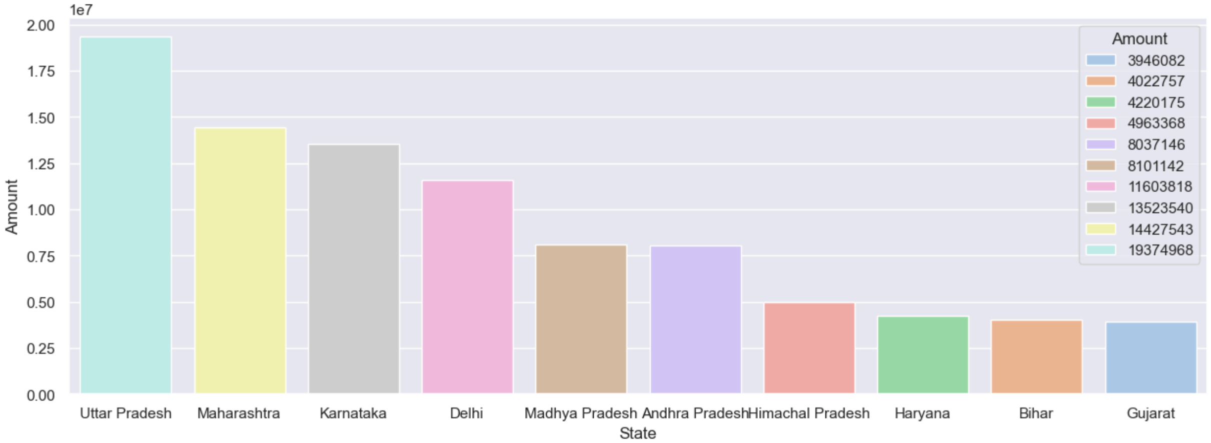
Out[67]:
```
Index(['User_ID', 'Cust_name', 'Product_ID', 'Gender', 'Age Group', 'Age',
       'Marital_Status', 'State', 'Zone', 'Occupation', 'Product_Category',
       'Orders', 'Amount'],
      dtype='object')
```

In [75]:
```
sale_state= df.groupby(['State'],as_index=False)['Orders'].sum().sort_values
sns.set(rc={'figure.figsize':(15,5)})
ax = sns.barplot(data=sale_state,x='State',y='Orders',palette='pastel',hue='
for bars in ax.containers:
    ax.bar_label(bars)
```



In [105…

```
sale_state= df.groupby(['State'],as_index=False)['Amount'].sum().sort_values
sns.set(rc={'figure.figsize':(15,5)})
ax = sns.barplot(data=sale_state,x='State',y='Amount',palette='pastel',hue='
```

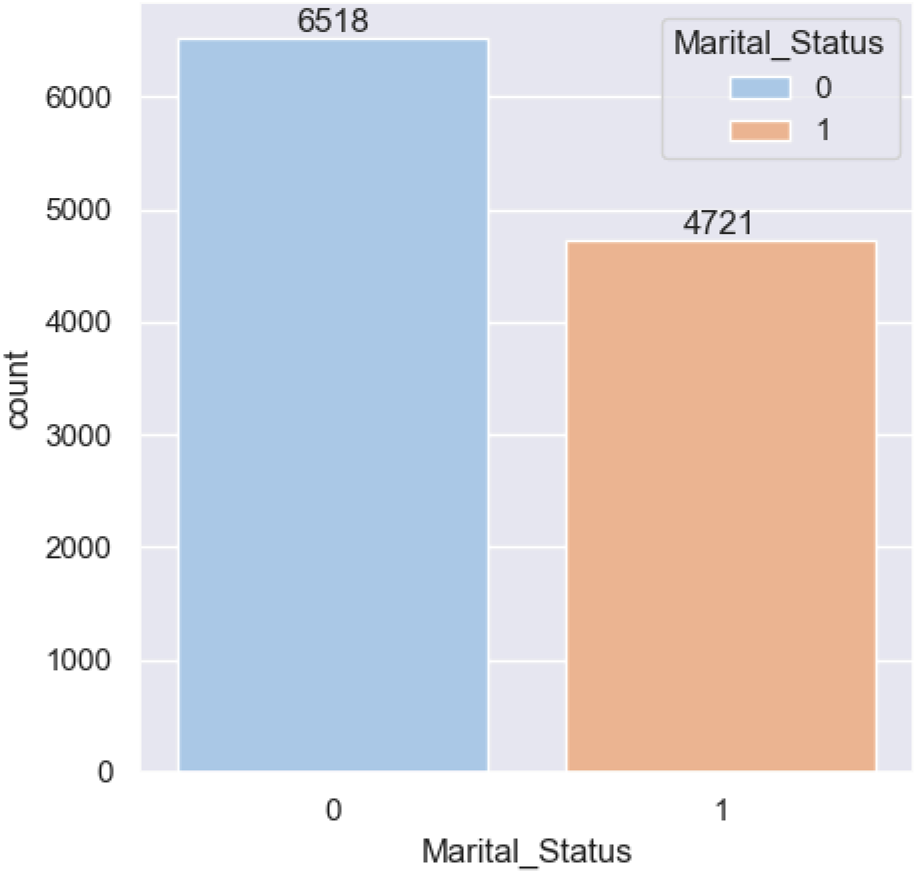


From above graphs we can see that most of the orders & total sales amount are from Uttar Pradesh

- Marital Status

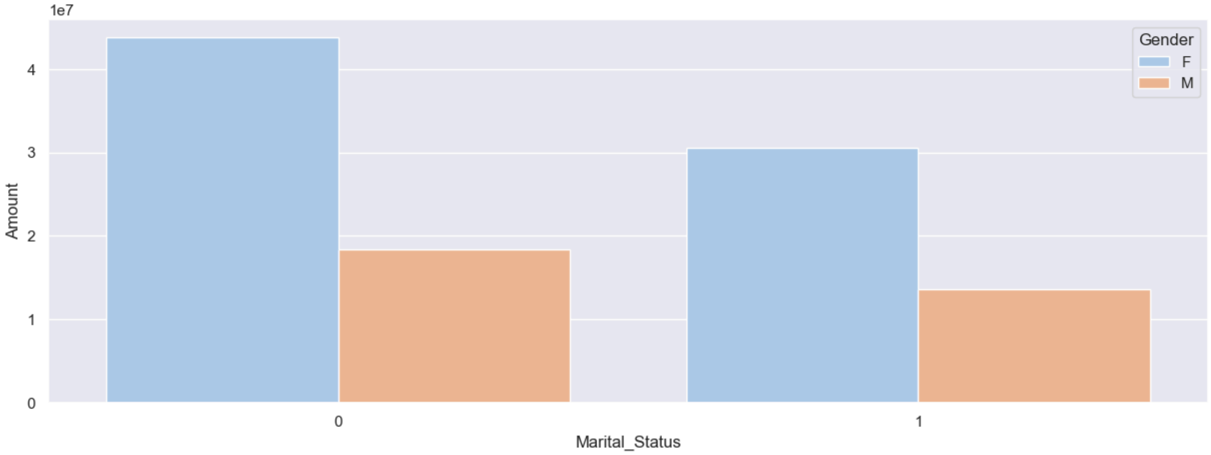
In [107…

```
ax = sns.countplot(data=df,x='Marital_Status',palette='pastel',hue='Marital_
sns.set(rc={'figure.figsize':(5,5)})
for bars in ax.containers:
    ax.bar_label(bars)
```



```
In [121…  sale_marital = df.groupby(['Marital_Status','Gender'], as_index=False)['Amou
          sns.barplot(data=sale_marital,x='Marital_Status',y='Amount',hue='Gender',pal
```

```
Out[121…  <Axes: xlabel='Marital_Status', ylabel='Amount'>
```

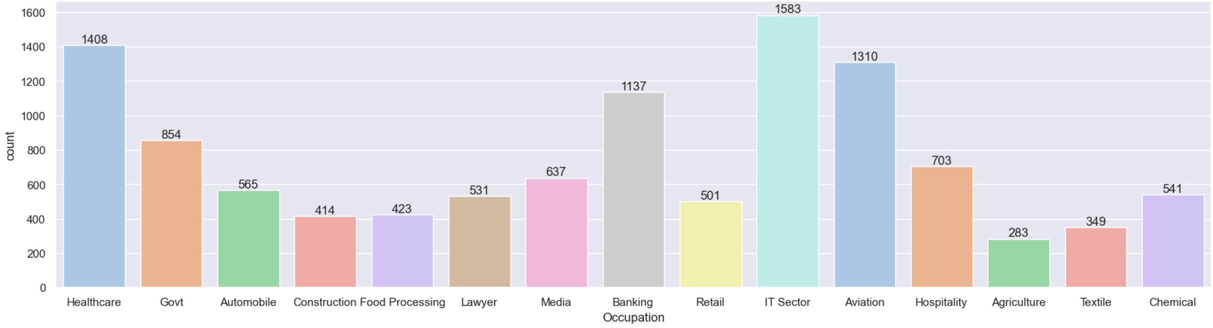


Frome Above graphs we can see that most of buyers are marride (women) and they have high purchasing power

- Occupation

```
In [125…  ax = sns.countplot(data=df,x='Occupation',palette='pastel',hue='Occupation')
          sns.set(rc={'figure.figsize':(20,5)})
```

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

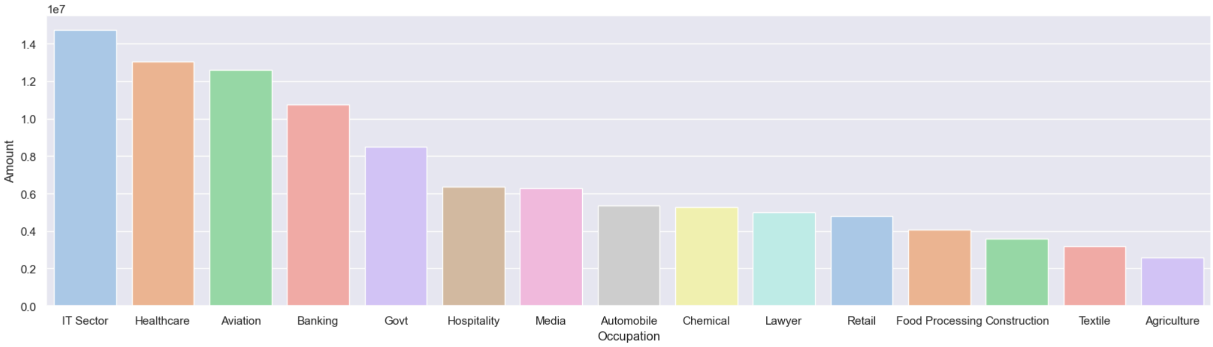


In [129…

```python
sales_occ = df.groupby(['Occupation'], as_index=False)['Amount'].sum().sort_
sns.set(rc={'figure.figsize':(20,5)})
sns.barplot(data=sales_occ,x='Occupation',y='Amount',palette='pastel',hue='O
```

Out[129…

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

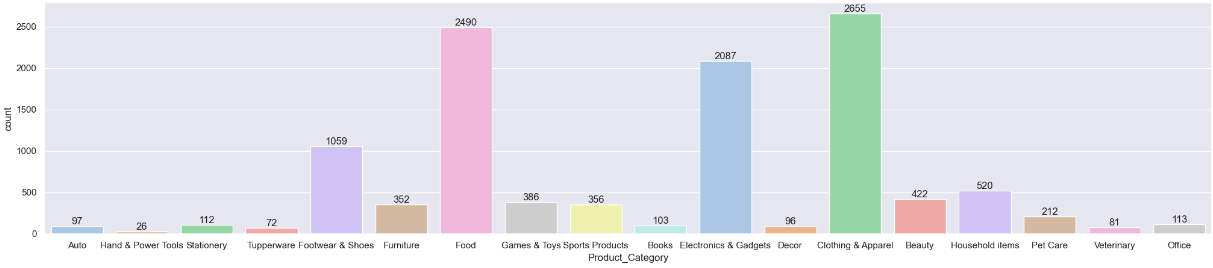


From above graphs we can see that most of the buyers are working in IT,Healthcare and Aviation sector

- Product Category

In [134…

```python
sns.set(rc={'figure.figsize':(25,5)})
ax = sns.countplot(data=df,x='Product_Category',palette='pastel',hue='Produc
for bars in ax.containers:
    ax.bar_label(bars)
```

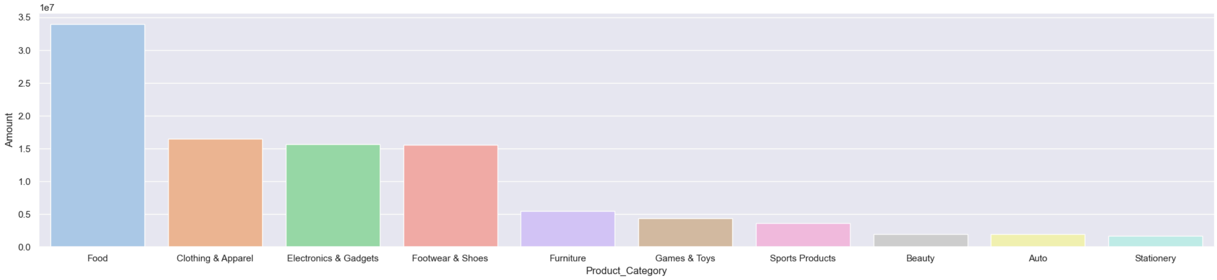


In [137…

```python
sales_occ = df.groupby(['Product_Category'], as_index=False)['Amount'].sum()
sns.set(rc={'figure.figsize':(25,5)})
sns.barplot(data=sales_occ,x='Product_Category',y='Amount',palette='pastel',
```

Loading [MathJax]/extensions/Safe.js

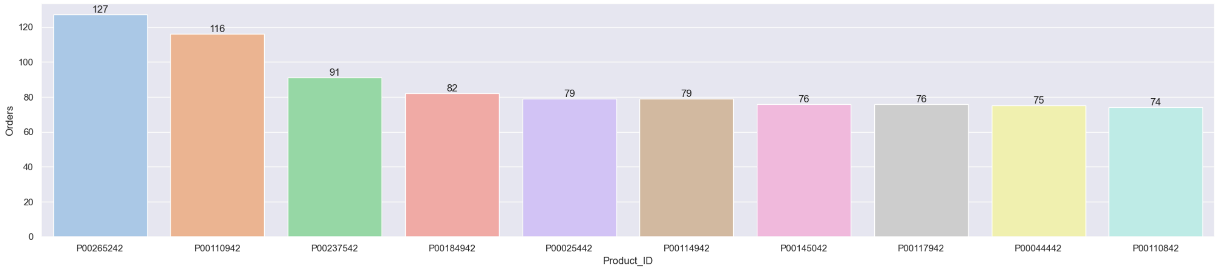
```
Out[137…  <Axes: xlabel='Product_Category', ylabel='Amount'>
```



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

- Product

```
In [138…  sale_state= df.groupby(['Product_ID'],as_index=False)['Orders'].sum().sort_v
          ax = sns.barplot(data=sale_state,x='Product_ID',y='Orders',palette='pastel',
          for bars in ax.containers:
              ax.bar_label(bars)
```



Conclusion :

Married women age group 26-35 year from UP, Maharashtra and Karnataka Working in IT, Healthcare and Aviation are more likely to buy products from Food, Clothings and Electronics category

```
In [ ]:
```

Loading [MathJax]/extensions/Safe.js