In [16]:
```
import numpy as np
import pandas as ps
import matplotlib.pyplot as plt #importing the chart and graph part of matplotli
%matplotlib inline  #to show graphs in same page below the code line
import seaborn as sn
```

```
UsageError: unrecognized arguments: #to show graphs in same page below the code l
ine
```

In [23]:
```
df = ps.read_csv(r'C:\Users\vansh\Downloads\Python_Diwali_Sales_Analysis-main\Py
#To avoid any type of encoding error while reading the data, encoding is used
#We have to mention the full path of the csv file, if the location of jupyter py
#We have used r in front of the file location to resolve any error which might c
```

In [24]:
```
df.shape
```

Out[24]:
```
(11251, 15)
```

In [26]:
```
df.head(15) #Shows top 5 rows by generic, we can tell row number in the brackets
```

Out[26]:

	User_ID	Cust_name	Product_ID	Gender	Age Group	Age	Marital_Status	St
0	1002903	Sanskriti	P00125942	F	26-35	28	0	Maharash
1	1000732	Kartik	P00110942	F	26-35	35	1	Andhra Prade
2	1001990	Bindu	P00118542	F	26-35	35	1	Uttar Prade
3	1001425	Sudevi	P00237842	M	0-17	16	0	Karnata
4	1000588	Joni	P00057942	M	26-35	28	1	Guja
5	1000588	Joni	P00057942	M	26-35	28	1	Himac Prade
6	1001132	Balk	P00018042	F	18-25	25	1	Uttar Prade
7	1002092	Shivangi	P00273442	F	55+	61	0	Maharash
8	1003224	Kushal	P00205642	M	26-35	35	0	Uttar Prade
9	1003650	Ginny	P00031142	F	26-35	26	1	Andhra Prade
10	1003829	Harshita	P00200842	M	26-35	34	0	De
11	1000214	Kargatis	P00119142	F	18-25	20	0	Andhra Prade
12	1004035	Elijah	P00080342	F	18-25	20	1	Andhra Prade
13	1001680	Vasudev	P00324942	M	26-35	26	1	Andhra Prade
14	1003858	Cano	P00293742	M	46-50	46	1	Madh Prade

In [28]:
```
df.info()
#Can be used for data insights, empty rows etc.
```

```
<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 [30]:
```
#To drop columns
df.drop(['Status','unnamed1'], axis=1, inplace=True)
#axis=1 means full column, inplace=True means apply to the table
```

In [35]:
```
df.dropna(inplace=True)
#Drops null value
```

In [33]:
```
df.shape
```

Out[33]:
```
(11239, 13)
```

In [38]:
```
#Data type change
df['Amount']=df['Amount'].astype('int')
```

In [40]:
```
df['Amount'].dtypes
```

Out[40]:
```
dtype('int64')
```

In [140…
```
#returns arithmatic answers for columns with numberical data
df.describe()
```

Out[140…

	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 [126…

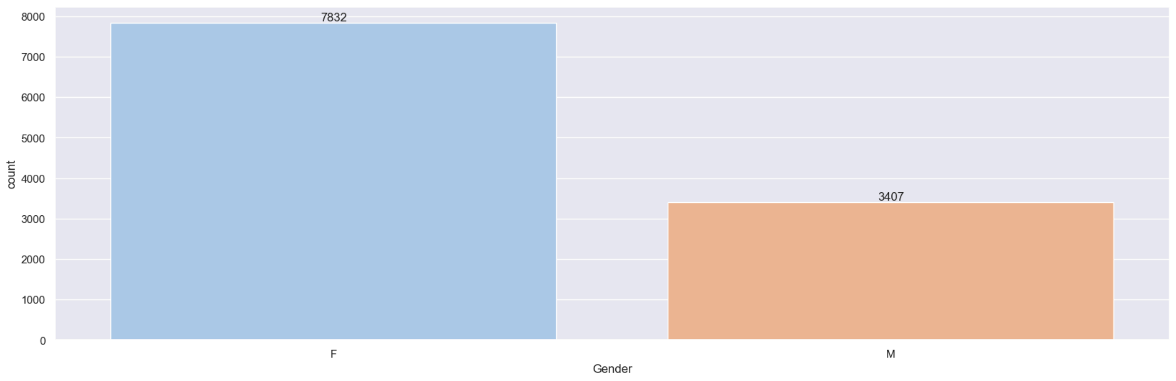
```
df[['Age','Orders','Amount']].describe()
```

Out[126…

	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

In [127…

```
# Gender Distribution of Buyers
sx=sn.countplot(x='Gender',hue='Gender',data=df,palette='pastel',legend=False)
for bars in sx.containers:
    sx.bar_label(bars)
```

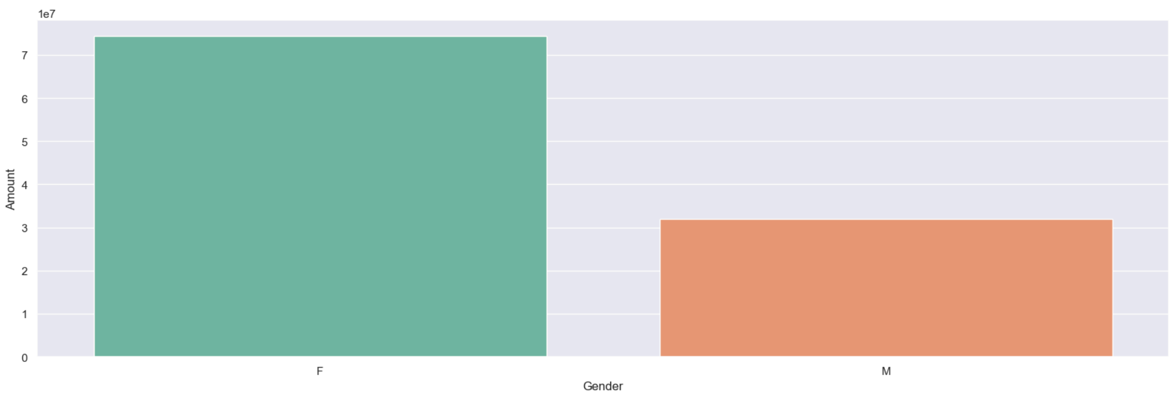


In [128…

```
sales_power = df.groupby(['Gender'], as_index=False)['Amount'].sum().sort_values
sn.barplot(x='Gender',hue='Gender',y='Amount',data=sales_power,palette='Set2', l
```

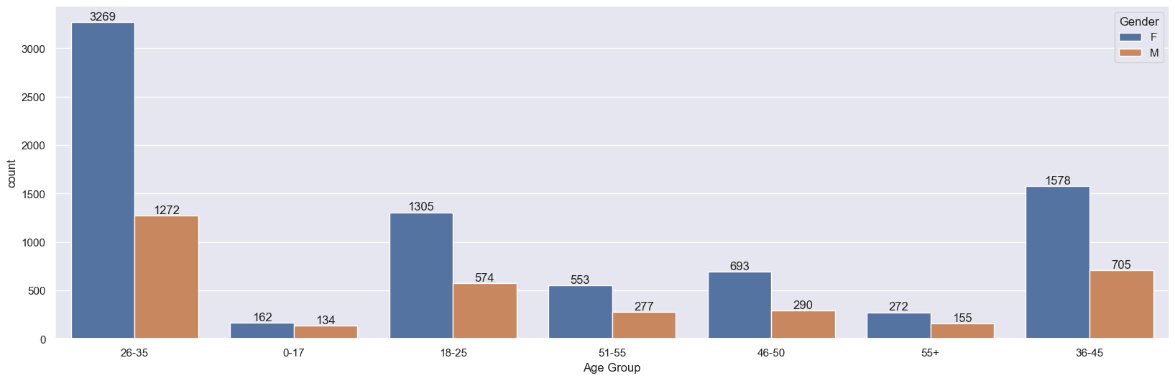
Out[128…

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



In [129…

```
# Gender Distribution Across Different Age Groups
ax=sn.countplot(data=df,x='Age Group',hue='Gender')
for bars in ax.containers:
    ax.bar_label(bars)
    #hue is used for categorisation of data furthermore
    #countplot is nothing but barplot with numbers
```

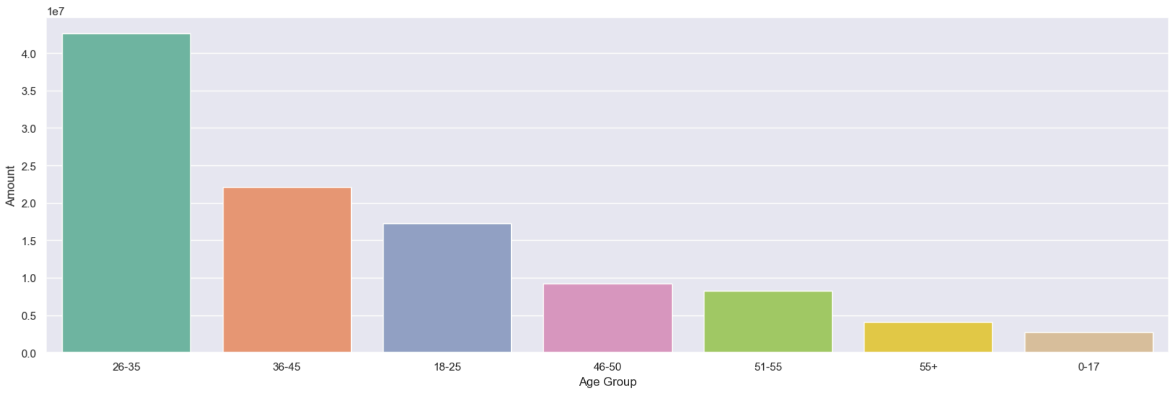


In [130…

```
#Total Amount vs Age Group Chart
sales_age= df.groupby(['Age Group'], as_index=False)['Amount'].sum().sort_values
sn.barplot(x='Age Group',y='Amount',hue='Age Group',palette='Set2',data=sales_ag
```

Out[130…

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

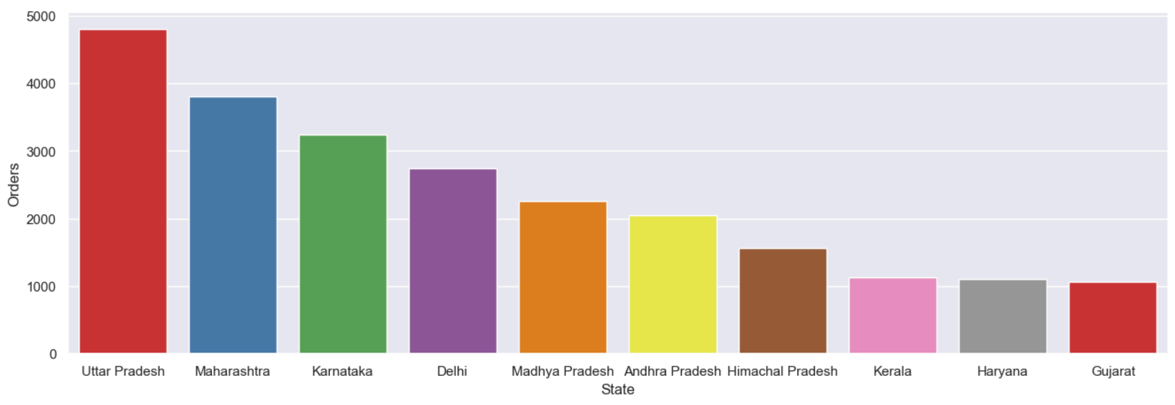


In [131…

```
sales_state= df.groupby(['State'], as_index=False)['Orders'].sum().sort_values(b
sn.set(rc={'figure.figsize':(16,5)})
sn.barplot(x='State',hue='State', data=sales_state, y='Orders',palette='Set1',le
```

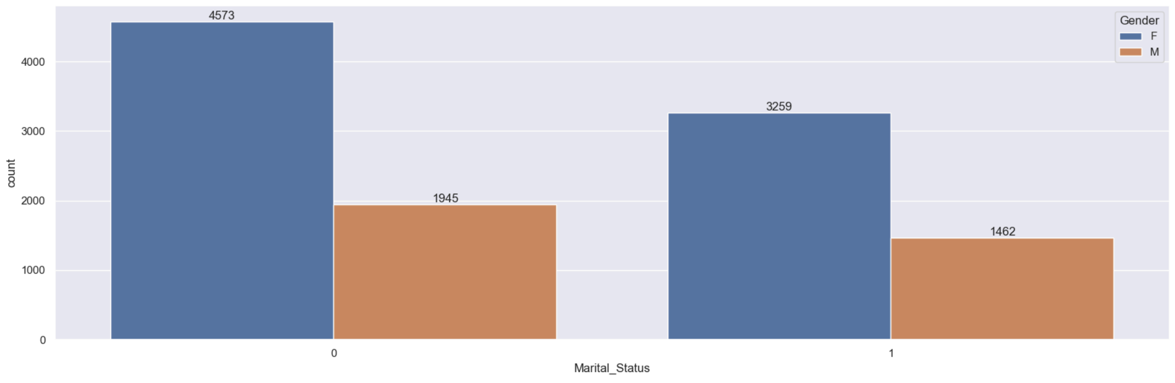
Out[131…

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



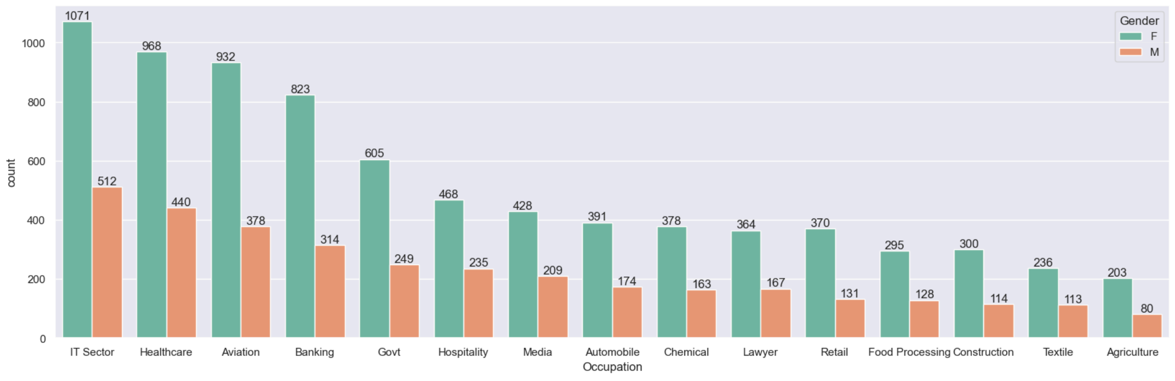
In [142…

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



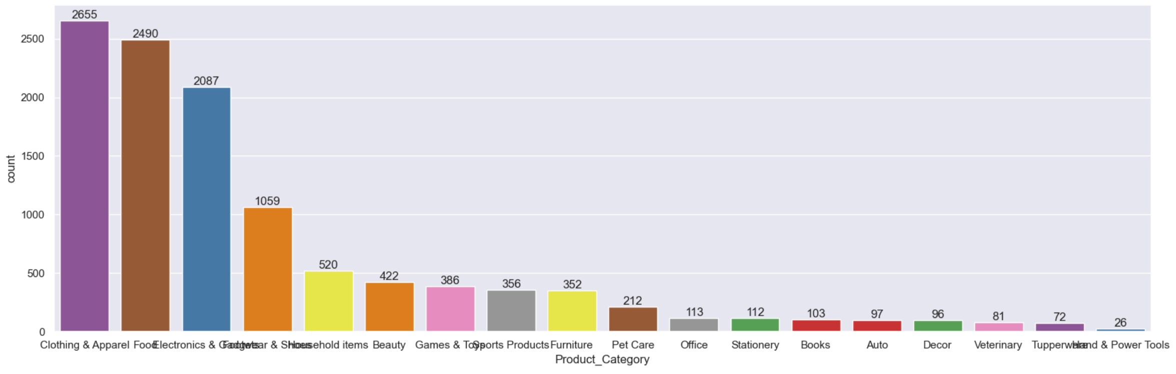
In [133…

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

order = df['Occupation'].value_counts().index
ax=sn.countplot(x='Occupation',data=df, hue='Gender',order=order,palette='Set2')

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



In [134…

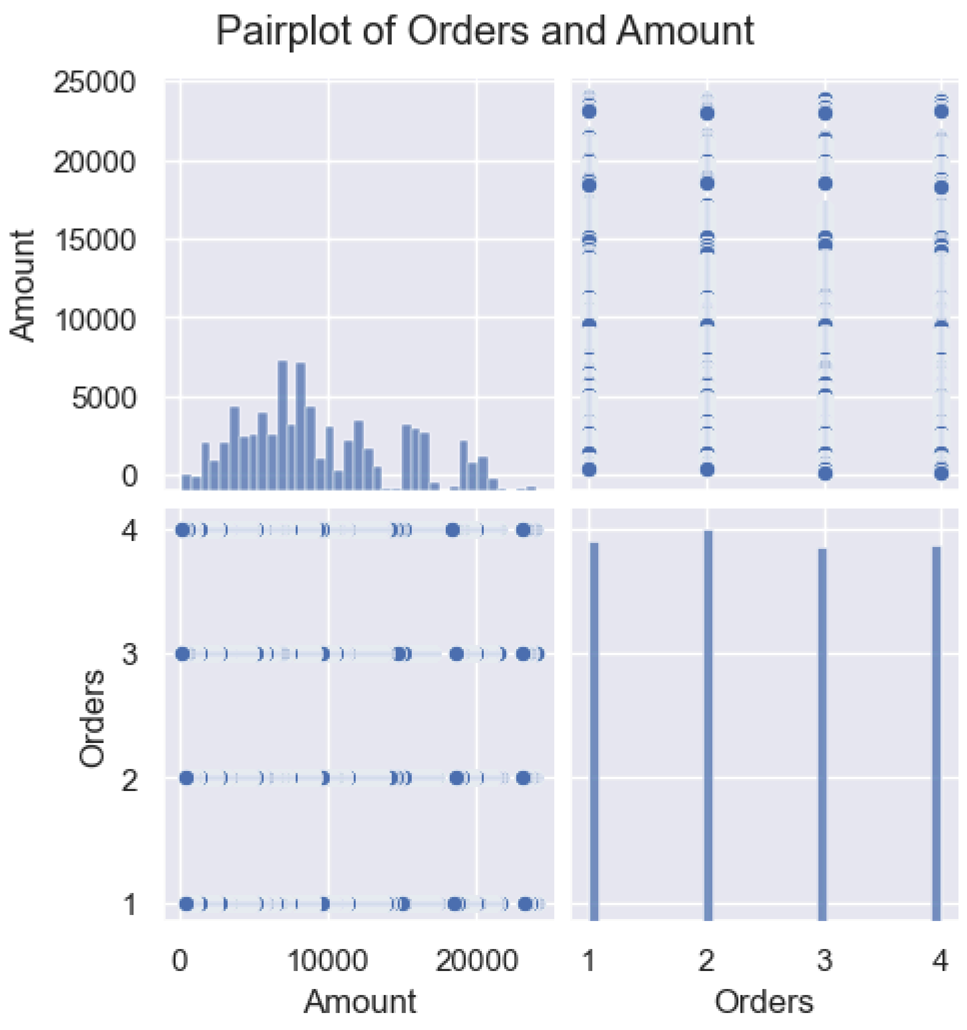
```
order= df['Product_Category'].value_counts().index
ax= sn.countplot(x='Product_Category', data=df, order=order, hue='Product_Catego
for bars in ax.containers:
    ax.bar_label(bars)
```



In [ ]:

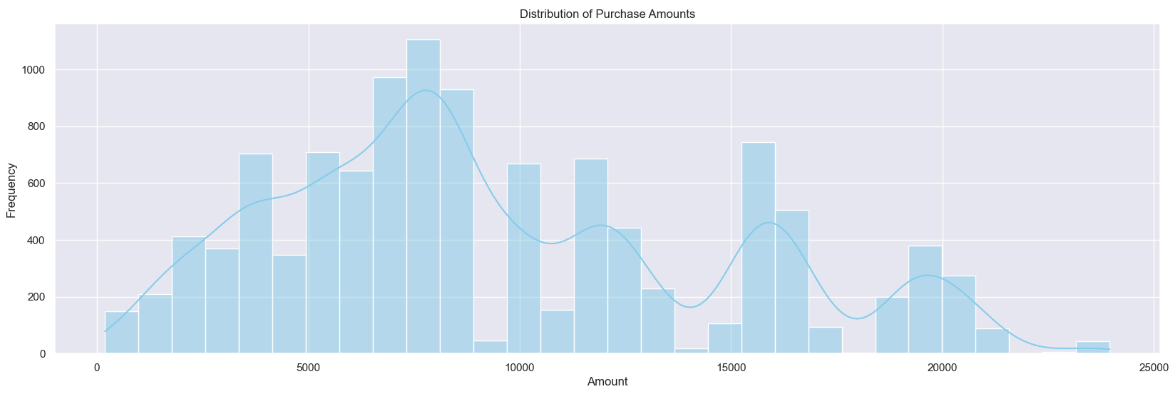
In [145…

```
ax=sn.pairplot(df[['Amount','Orders']])
plt.suptitle("Pairplot of Orders and Amount", y=1.03)
plt.show()
```



In [149…

```
ax= sn.histplot(df['Amount'], bins=30, kde=True, color='skyblue')
plt.title("Distribution of Purchase Amounts")
plt.ylabel("Frequency")
plt.show()
```



In [150…

```
# Conclusions:

#From above graph we can see that there is a steep growth in the frequency of sa
#From above graphs we can see that most of the buyers are females and even the p

#We can see that most of the buyers are of age group between 26-35 yrs female.

#We can also see that most of the orders & total sales/amount are from Uttar Pra

#We can see that most of the buyers are married (women) and they have high purch

#From above graphs we can see that most of the buyers are working in IT, Healthc

# Married women age group 26-35 yrs from UP, Maharastra and Karnataka working in
# Healthcare and Aviation are more likely to buy products from Food, Clothing an
```

In [ ]:

```
```