Objective :

Python project for beginners- Analyze Diwali sales data to improve customer experience and sales

Import the Library

In [ ]:
```
# import python libraries

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

In [ ]:
```
# import CSV Data

df = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/Diwali Sales Data.csv
df
```

Out[ ]:

	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	A
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	

11251 rows × 15 columns

Data Cleaning

In [ ]:
```
# Show the Number of Rows and column

df.shape
```

Out[ ]: (11251, 15)

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

Out[ ]:

	User_ID	Age	Marital_Status	Orders	Amount
count	1.125100e+04	11251.000000	11251.000000	11251.000000	11239.000000
mean	1.003004e+06	35.421207	0.420318	2.489290	9453.610858
std	1.716125e+03	12.754122	0.493632	1.115047	5222.355869
min	1.000001e+06	12.000000	0.000000	1.000000	188.000000
25%	1.001492e+06	27.000000	0.000000	1.500000	5443.000000
50%	1.003065e+06	33.000000	0.000000	2.000000	8109.000000
75%	1.004430e+06	43.000000	1.000000	3.000000	12675.000000
max	1.006040e+06	92.000000	1.000000	4.000000	23952.000000

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

Out[ ]:

	0
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
Status	11251
unnamed1	11251

dtype: int64

In [ ]:
```
#drop unrelated/blank columns

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

In [ ]:
```
df.isnull().sum()
df.dropna(inplace=True)
```

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

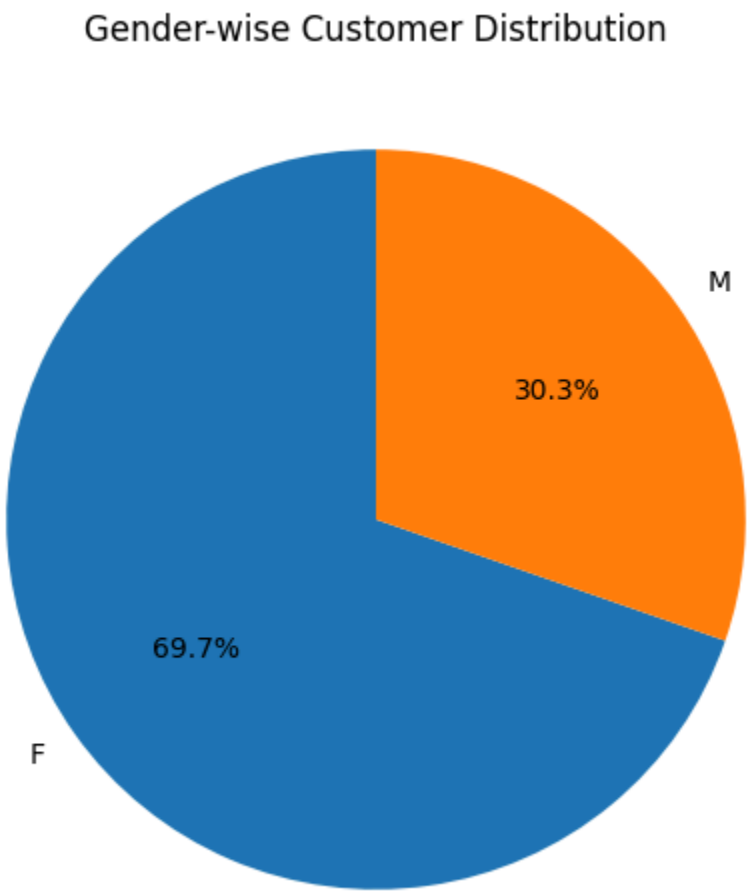
```
<class 'pandas.core.frame.DataFrame'>
Index: 11239 entries, 0 to 11250
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   User_ID           11239 non-null  int64  
 1   Cust_name         11239 non-null  object 
 2   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  float64
dtypes: float64(1), int64(4), object(8)
memory usage: 1.2+ MB
```

In [ ]:
```
df.drop_duplicates()
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  float64
dtypes: float64(1), int64(4), object(8)
memory usage: 1.2+ MB
```

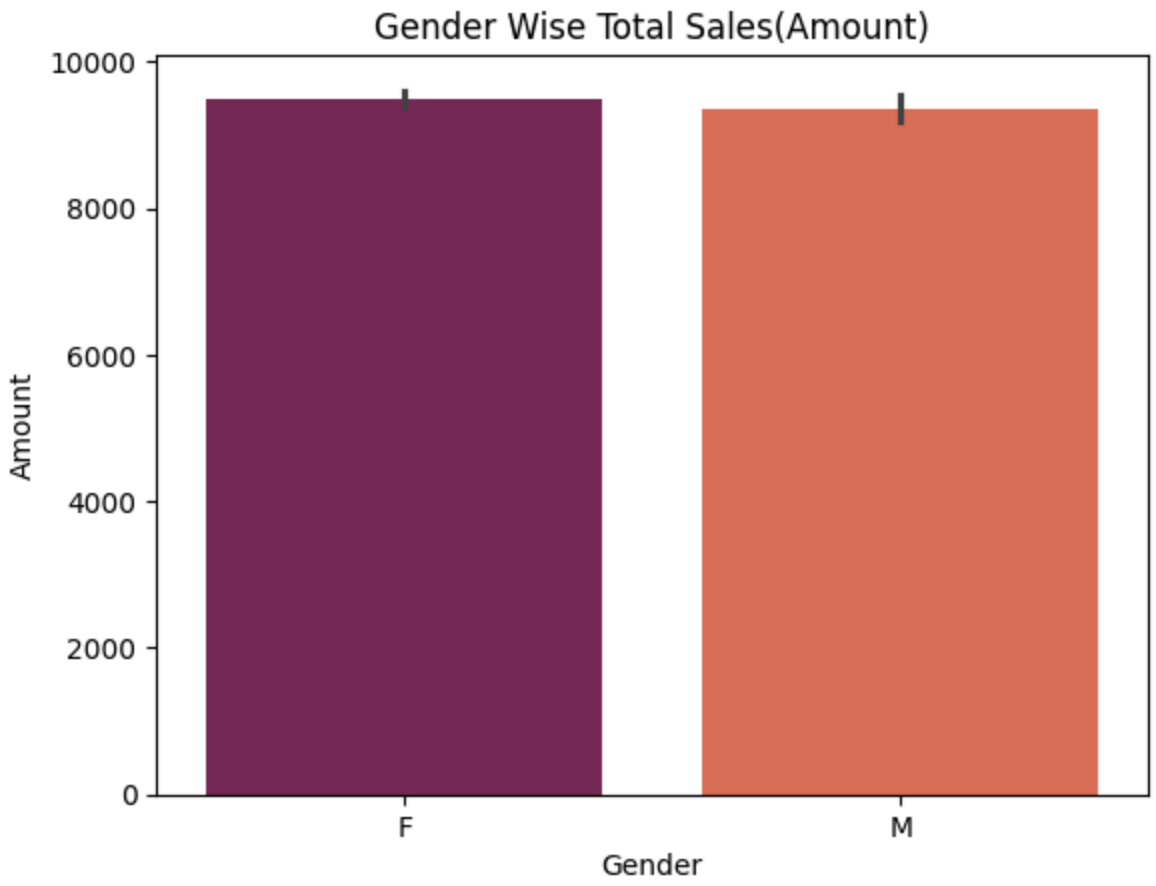
Exploratory Data Analysis

In [ ]:
```
# Count of Gender values
gender_counts = df['Gender'].value_counts()

# Plotting the pie chart
plt.figure(figsize=(6,6))
plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.1f%%', startang
plt.title('Gender-wise Customer Distribution')
plt.show()
```



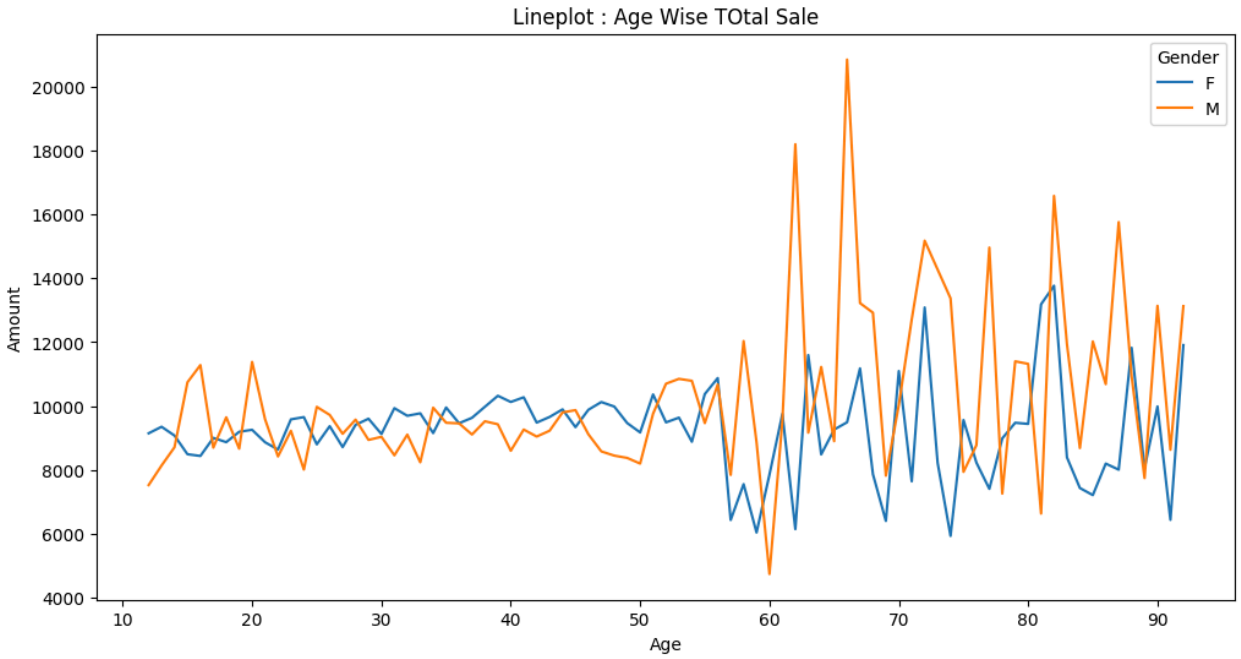
In [ ]:
```
# plotting a bar chart for gender vs total amount
sns.barplot(x=df['Gender'],y=df['Amount'],palette='rocket')
plt.title('Gender Wise Total Sales(Amount)')
plt.show()
```

```
/tmp/ipython-input-33-2896078806.py:2: 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 e
ffect.

  sns.barplot(x=df['Gender'],y=df['Amount'],palette='rocket')
```



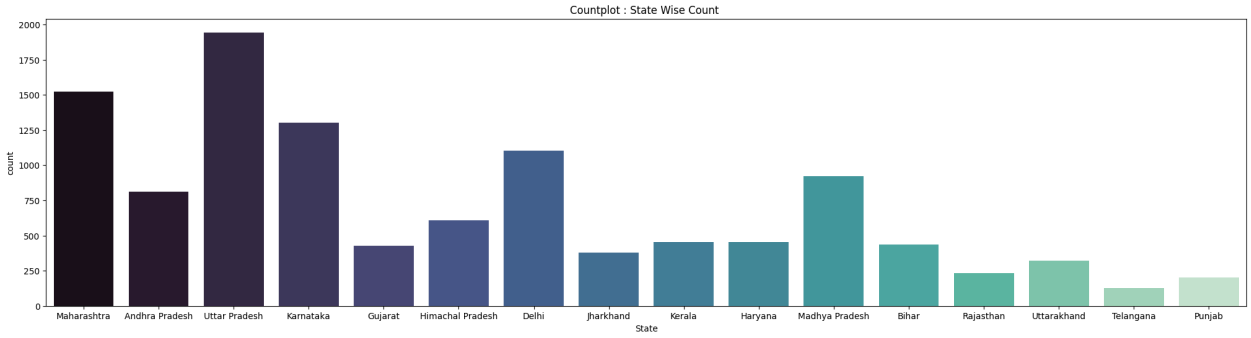
In [ ]:
```
plt.figure(figsize=(12,6))
sns.lineplot(x=df['Age'],y=df['Amount'],hue='Gender', data=df,ci=None)
plt.title('Lineplot : Age Wise TOtal Sale')
plt.show()
```

```
/tmp/ipython-input-34-2461244245.py:2: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(x=df['Age'],y=df['Amount'],hue='Gender', data=df,ci=None)
```



In [ ]:
```
# total number of orders from top 10 states
plt.figure(figsize=(25,6))
sns.countplot(x=df['State'],data=df,palette='mako')
plt.title('Countplot : State Wise Count ')
plt.show()
```

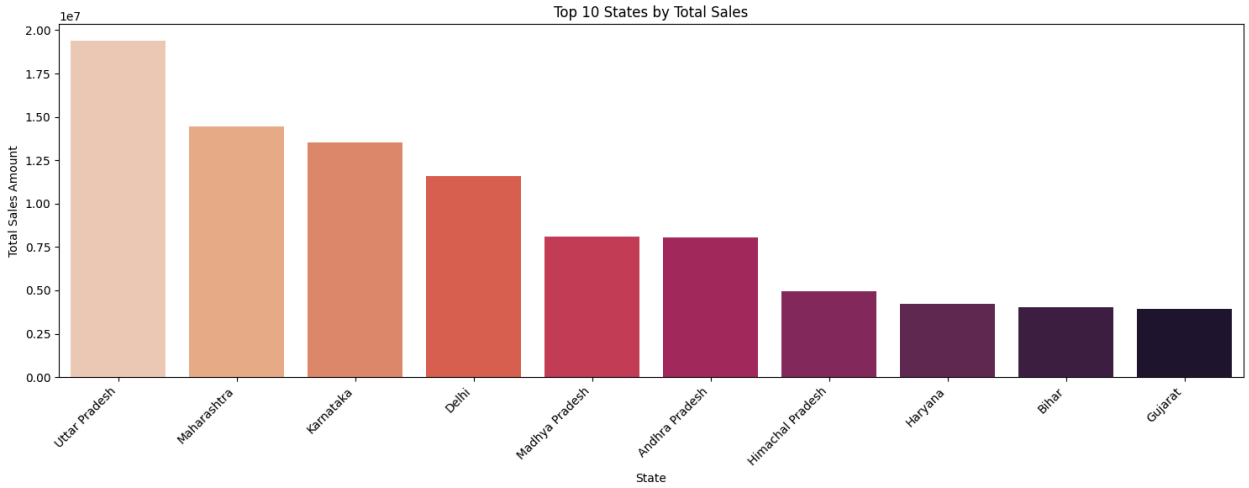
```
/tmp/ipython-input-49-3092893013.py:3: FutureWarning:

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 e
ffect.

  sns.countplot(x=df['State'],data=df,palette='mako')
```



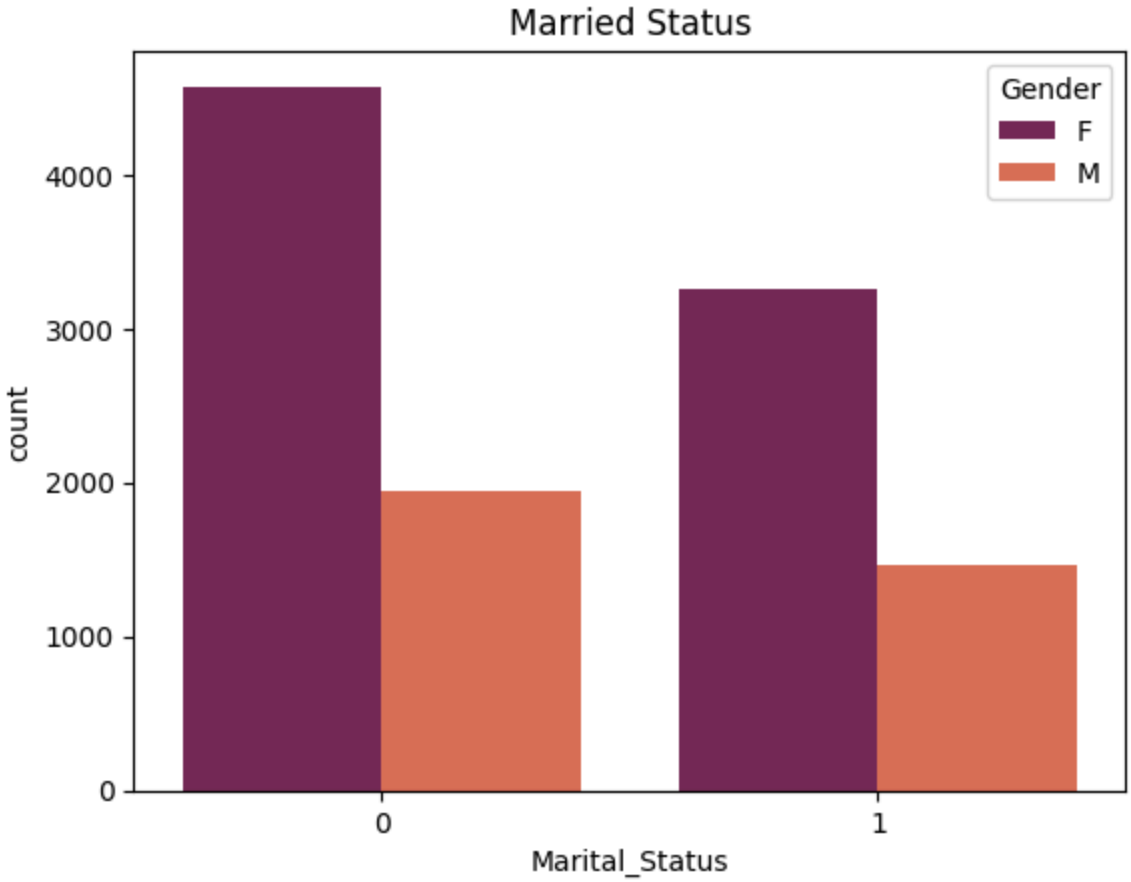
In [ ]:
```
# total amount/sales from top 10 states
sales_by_state = df.groupby('State')['Amount'].sum().sort_values(ascending=Fal

plt.figure(figsize=(15,6))
sns.barplot(x=sales_by_state.index, y=sales_by_state.values, palette="rocket_r
plt.title('Top 10 States by Total Sales')
plt.xlabel('State')
plt.ylabel('Total Sales Amount')
plt.xticks(rotation=45, ha='right') # Rotate labels for better readability
```

```
plt.tight_layout() # Adjust layout to prevent labels overlapping
plt.show()
```

```
/tmp/ipython-input-66-2385837598.py:5: 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 e
ffect.

  sns.barplot(x=sales_by_state.index, y=sales_by_state.values, palette="rocke
t_r")
```

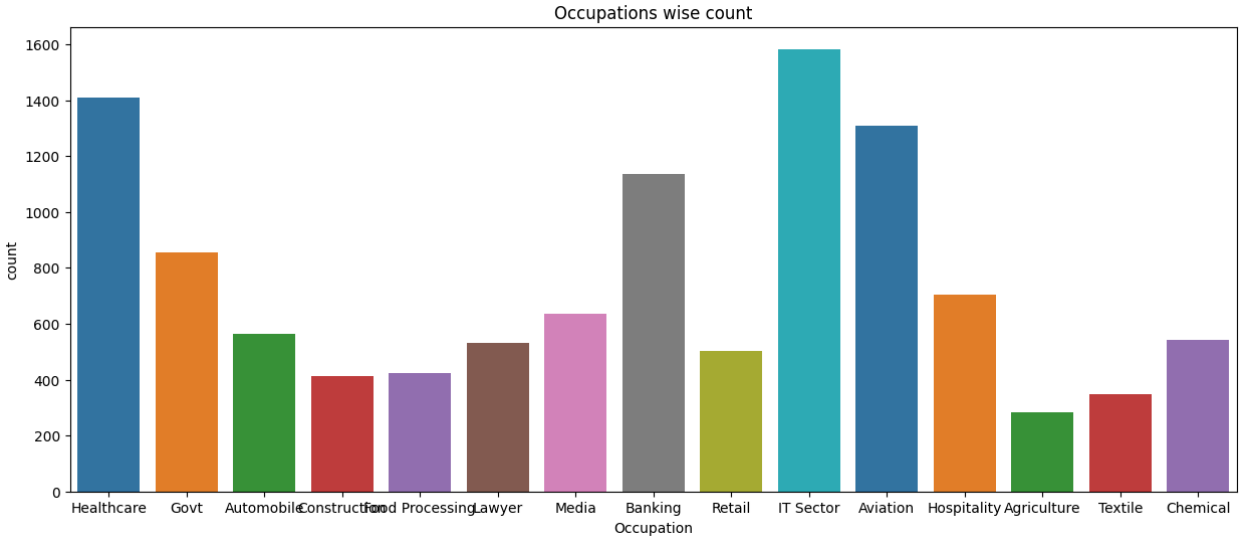


In [ ]:
```
# Married VS Single
sns.countplot(x=df['Marital_Status'],data=df,palette='rocket',hue='Gender')
plt.title('Married Status')
plt.grid=(True)
plt.show()
```



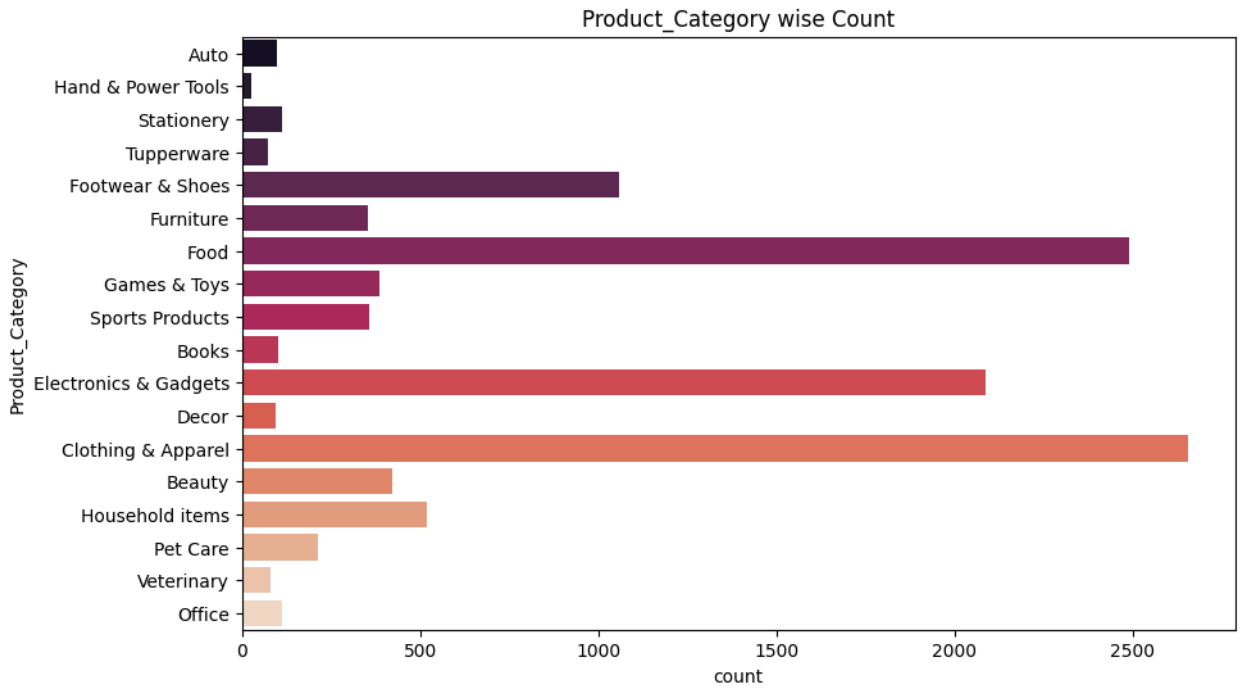
In [ ]:
```
# Count of Occupation
plt.figure(figsize=(15,6))
sns.countplot(x=df['Occupation'],data=df,palette='tab10')
plt.title('Occupations wise count')
plt.show()
```

```
/tmp/ipython-input-89-2553830485.py:3: 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 e
ffect.

  sns.countplot(x=df['Occupation'],data=df,palette='tab10')
```



In [ ]:
```
# Product_Category VS Count
plt.figure(figsize=(10,6))
sns.countplot(y=df['Product_Category'],data=df,palette='rocket')
plt.title('Product_Category wise Count')
plt.show()
```

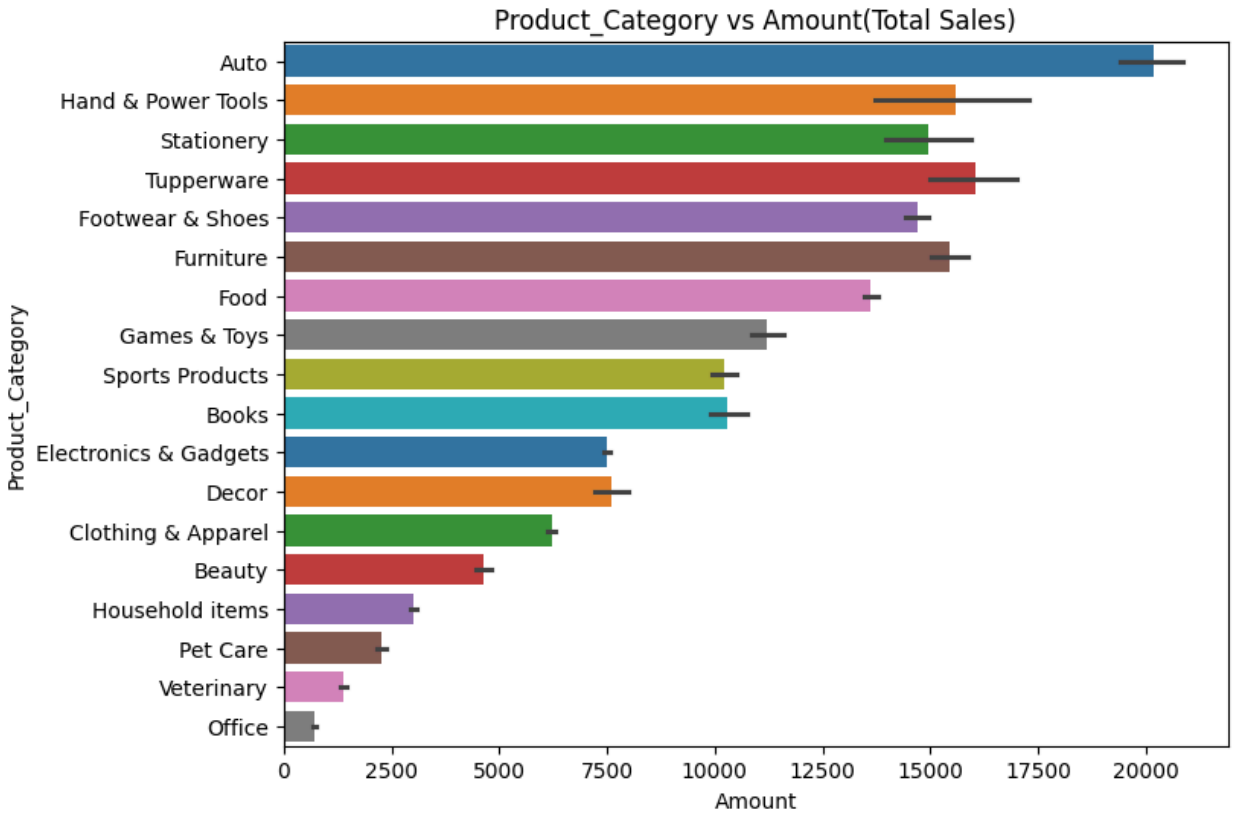
```
/tmp/ipython-input-95-900891140.py:3: FutureWarning:

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

  sns.countplot(y=df['Product_Category'],data=df,palette='rocket')
```



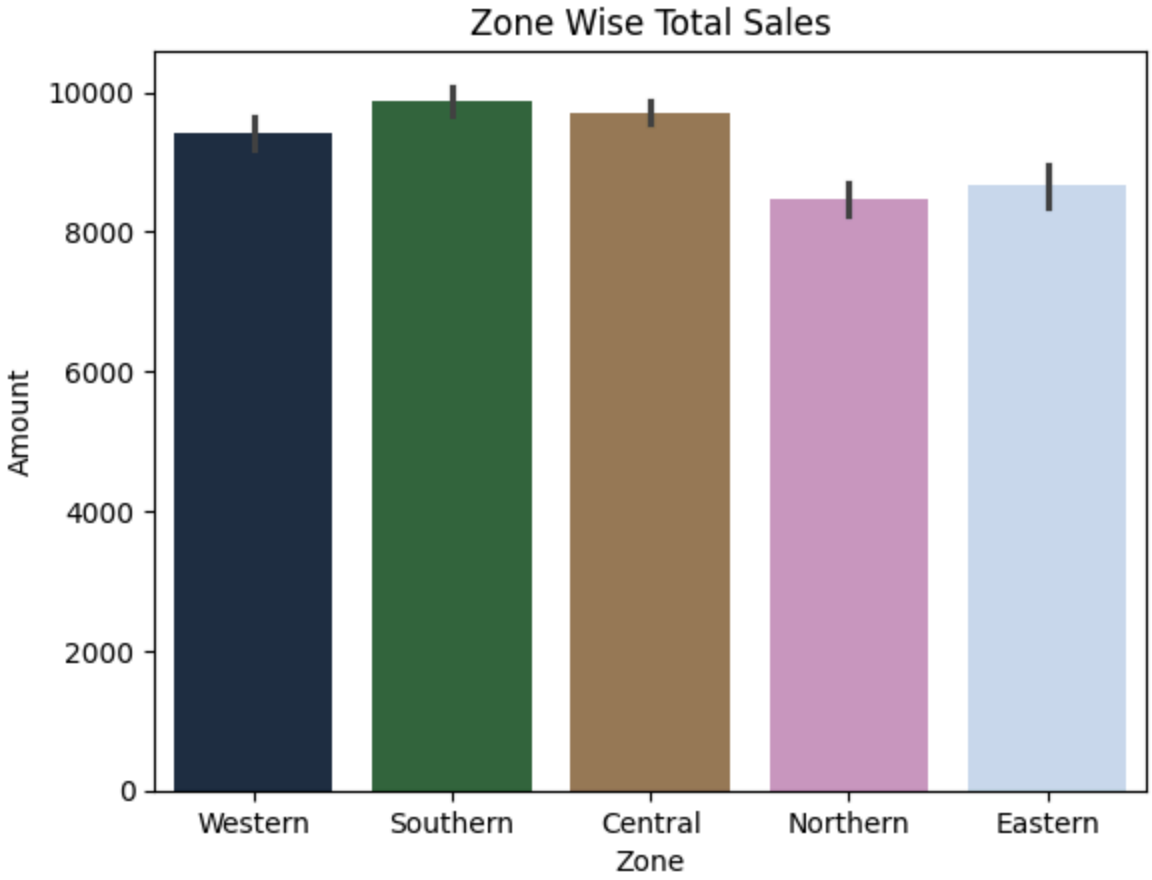
In [ ]:
```
# Product_Category vs Amount(Total Sales)
plt.figure(figsize=(8,6))
```

```
sns.barplot(y=df['Product_Category'],x=df['Amount'],palette='tab10')
plt.title('Product_Category vs Amount(Total Sales)')
plt.show()
```

```
/tmp/ipython-input-102-4006654722.py:3: FutureWarning:

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

  sns.barplot(y=df['Product_Category'],x=df['Amount'],palette='tab10')
```



In [ ]:
```
ax = sns.barplot(x=df['Zone'],y=df['Amount'],palette='cubehelix')
plt.title('Zone Wise Total Sales')
plt.show()
```


Zone Wise Total Sales
Amount
10000
8000
6000
4000
2000
0
Western
Southern
Central
Northern
Eastern
Zone