

**AI/ML Programming**

**MCA-475**

**Assignment – 02**

***BY***

**HIMANSHU HEDA (24225013)**

**SUBMITTED TO**

**Dr. Manjula Shannhog**

**SCHOOL OF SCIENCES**

**2024-25**

**Importing Libraries**

import pandas as pd

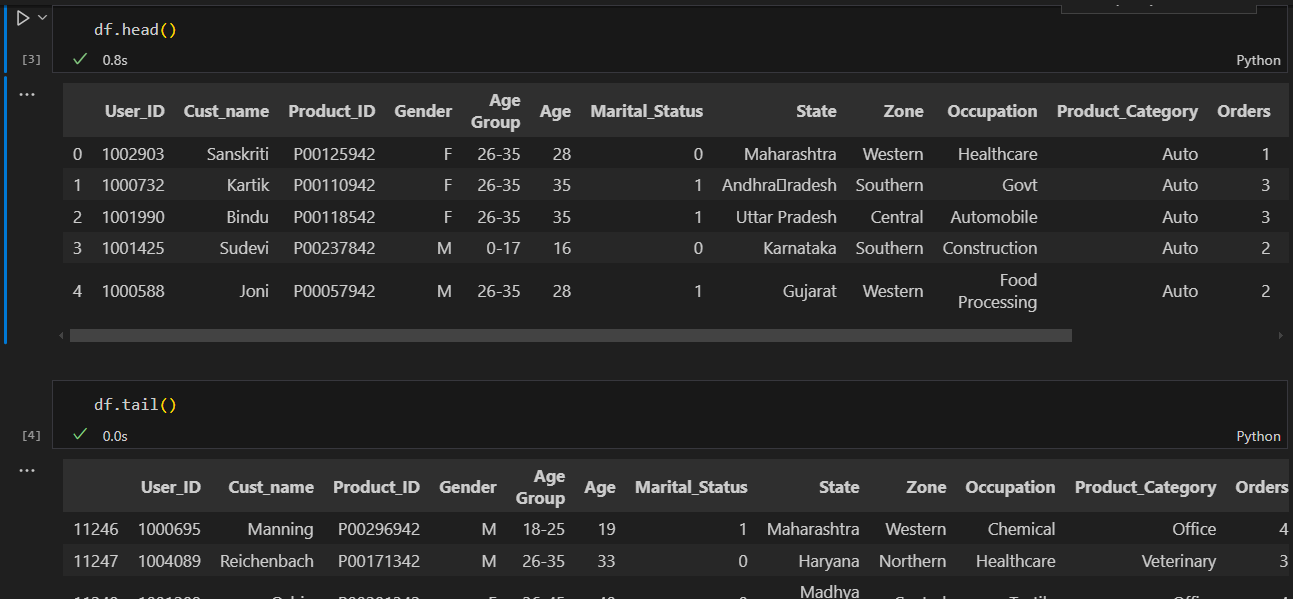
import numpy as np

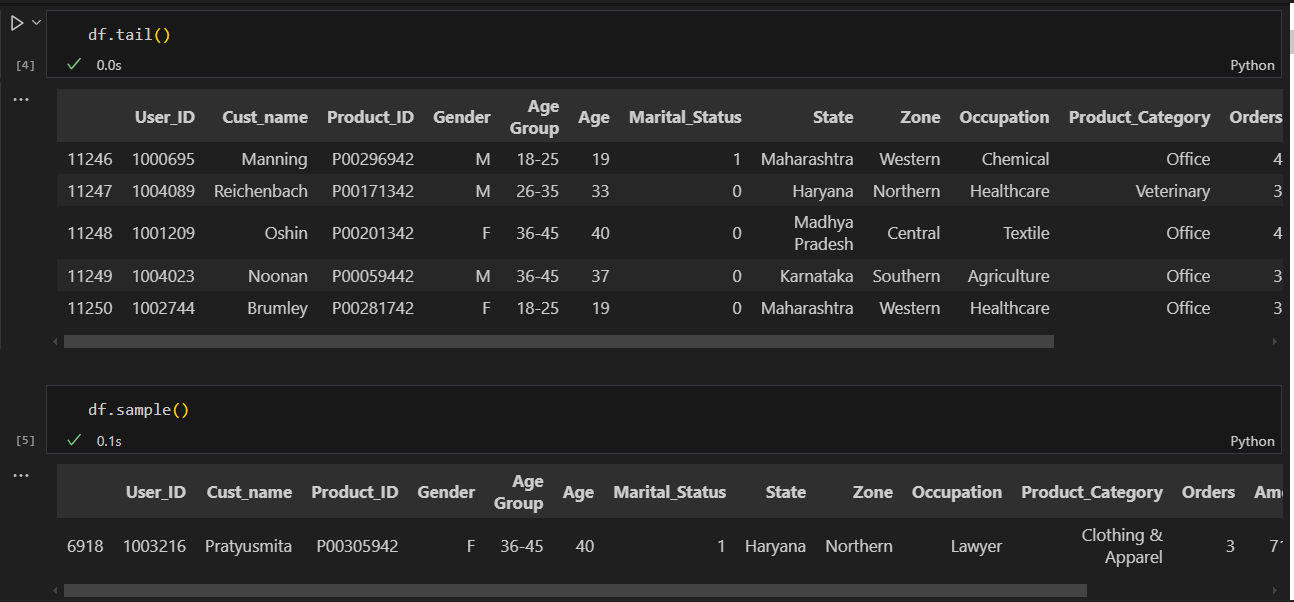
import matplotlib.pyplot as plt

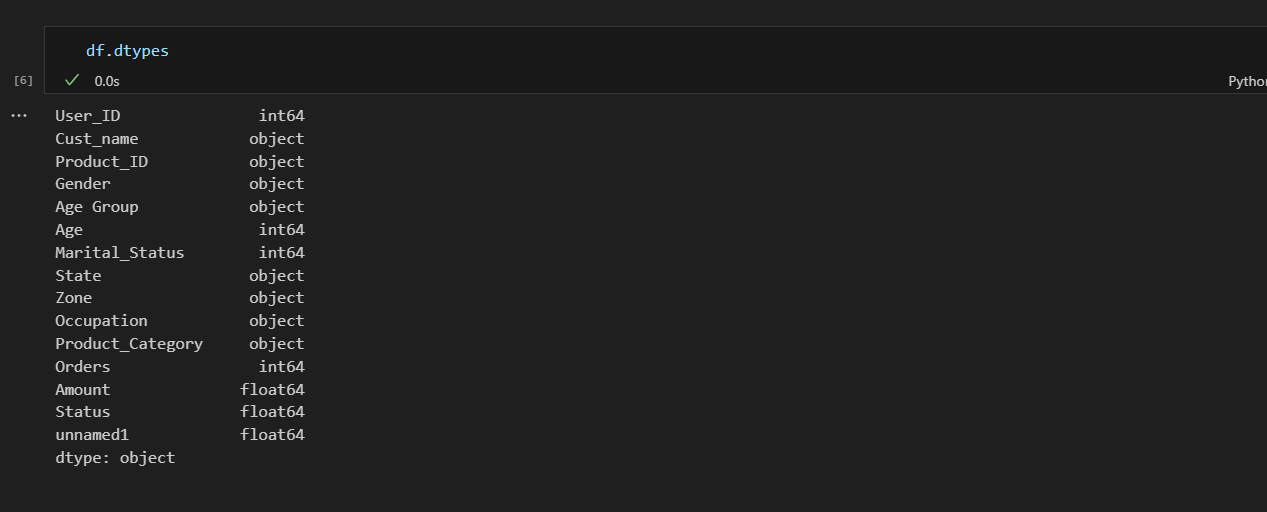
import seaborn as sns

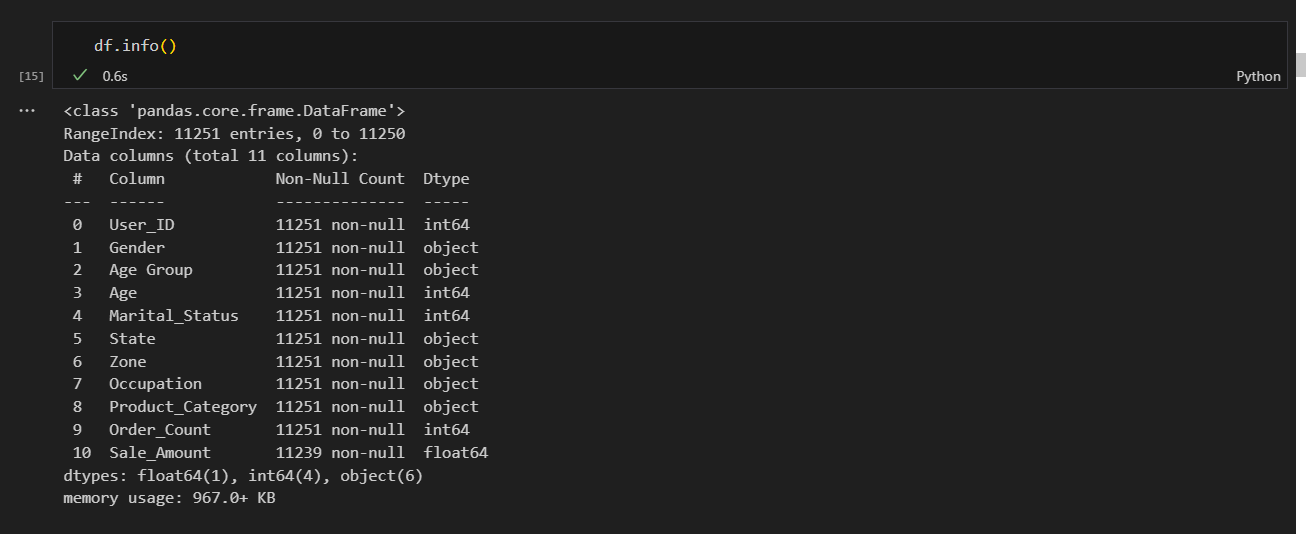
from datetime import date

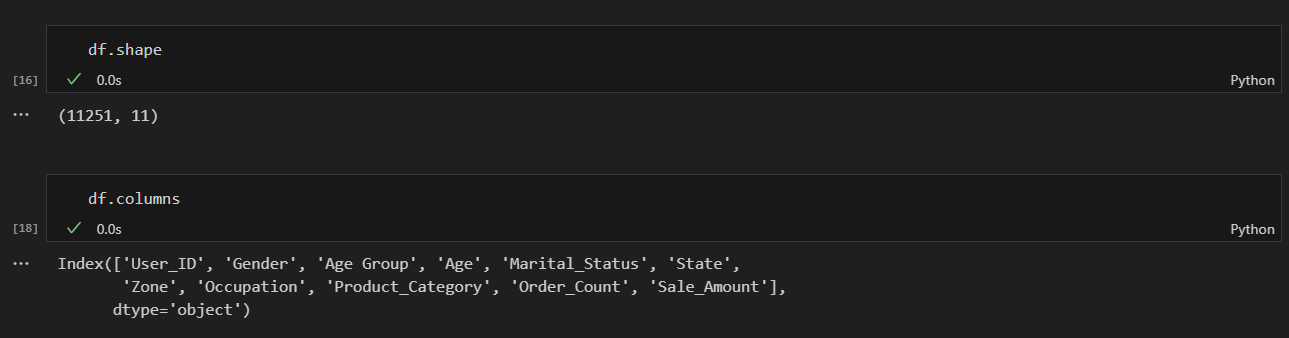
df = pd.read\_csv('./Dataset/Diwali\_Sales\_Data.csv')

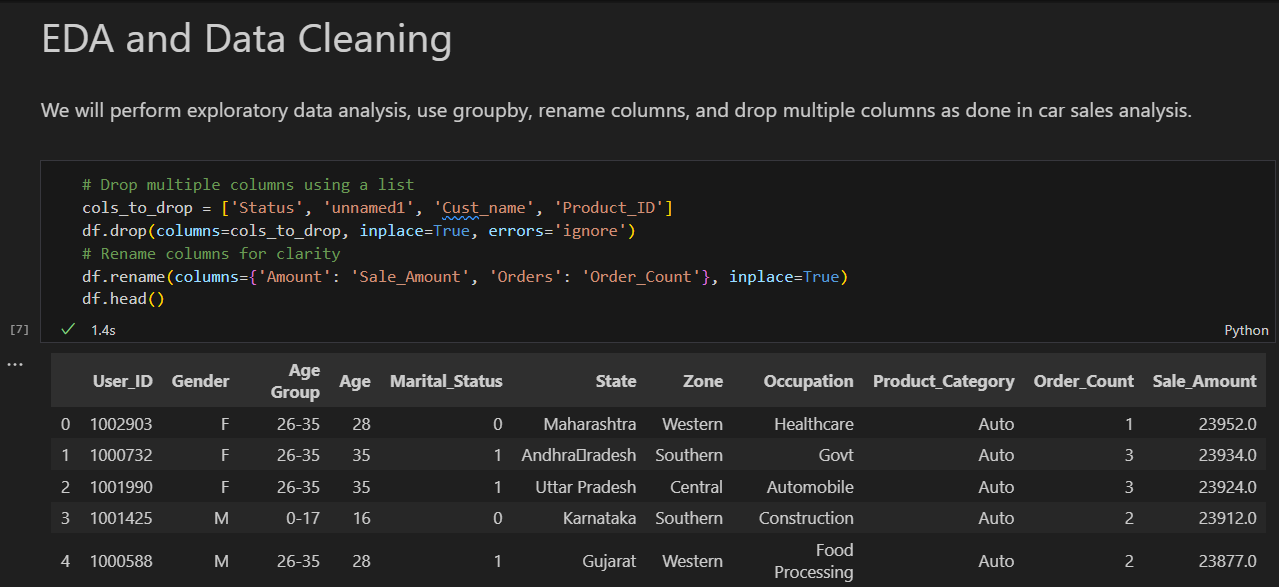
****

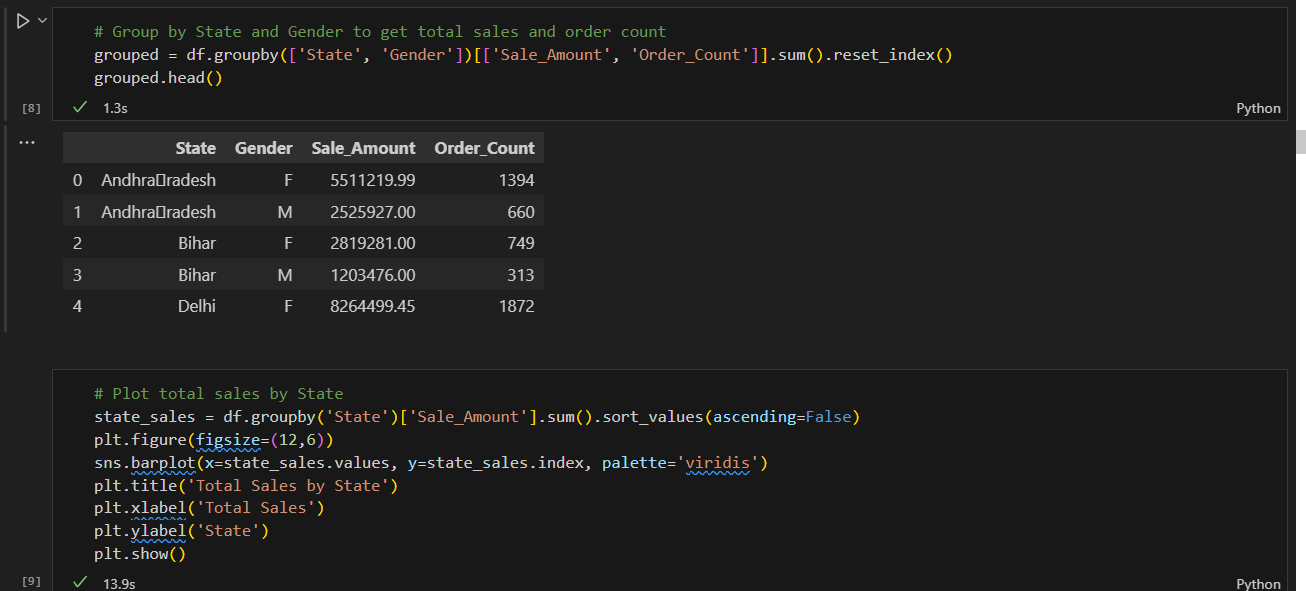
****

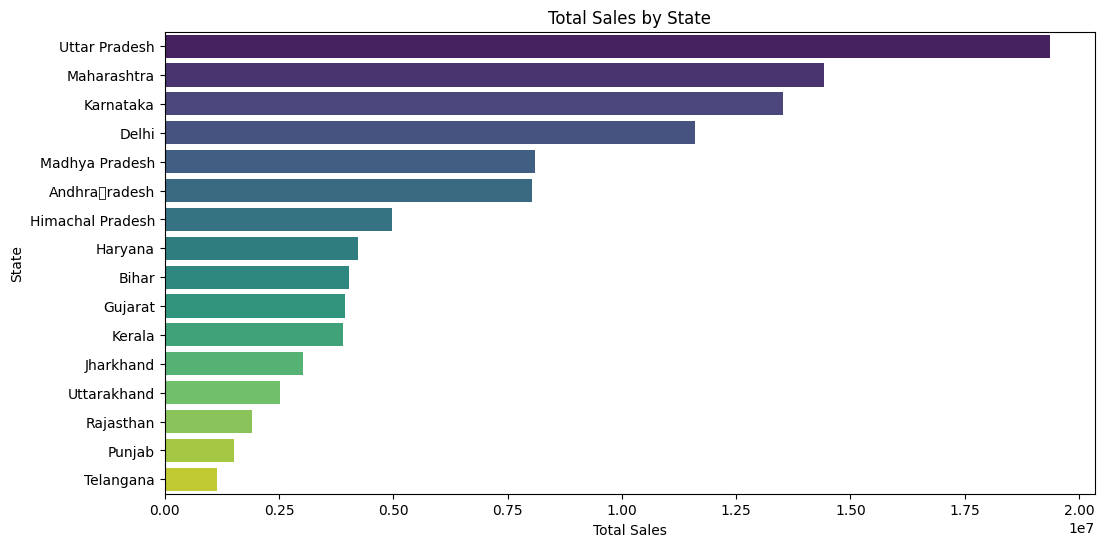
****

****

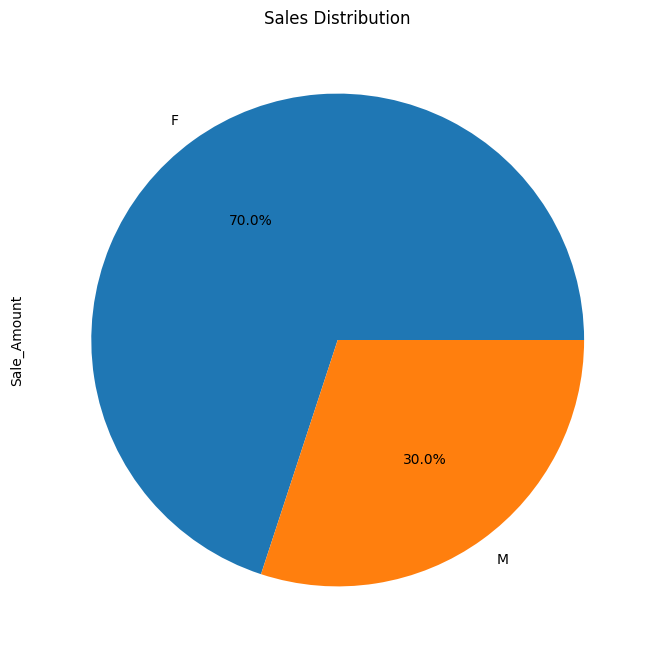
****

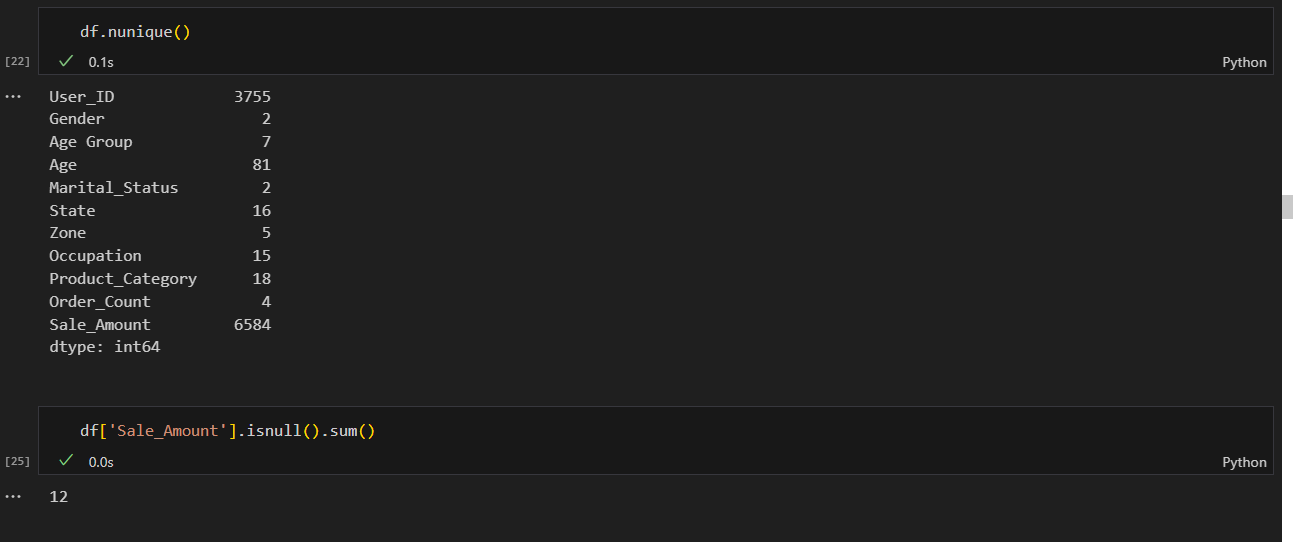
****

****

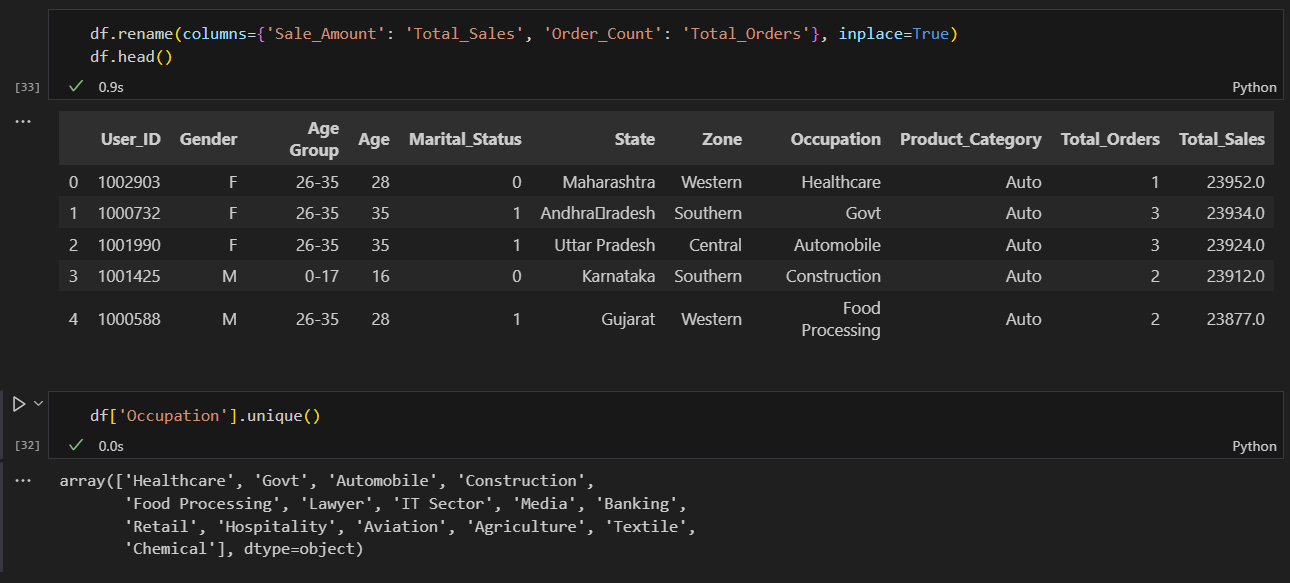
****

df.groupby('Gender')['Sale\_Amount'].sum().plot(kind='pie', autopct='%1.1f%%', figsize=(8, 8), title='Sales Distribution')

****

****

****

****

# Analyze unique User\_IDs and perform groupby operations

unique\_users = df['User\_ID'].nunique()

print(f'Total unique users: {unique\_users}')

# Group by User\_ID to get total orders and sales per user

user\_group = df.groupby('User\_ID').agg({

    'Total\_Orders': 'sum',

    'Total\_Sales': ['sum', 'mean', 'count']

}).reset\_index()

user\_group.columns = ['User\_ID', 'Total\_Orders', 'Total\_Sales\_Sum', 'Total\_Sales\_Mean', 'Order\_Count']

user\_group.head()

# Find top 10 users by total sales

top\_users = user\_group.sort\_values('Total\_Sales\_Sum', ascending=False).head(10)

print('Top 10 users by total sales:')

print(top\_users)

# Plot top 10 users by total sales

plt.figure(figsize=(10,5))

sns.barplot(data=top\_users, x='User\_ID', y='Total\_Sales\_Sum', palette='crest')

plt.title('Top 10 Users by Total Sales')

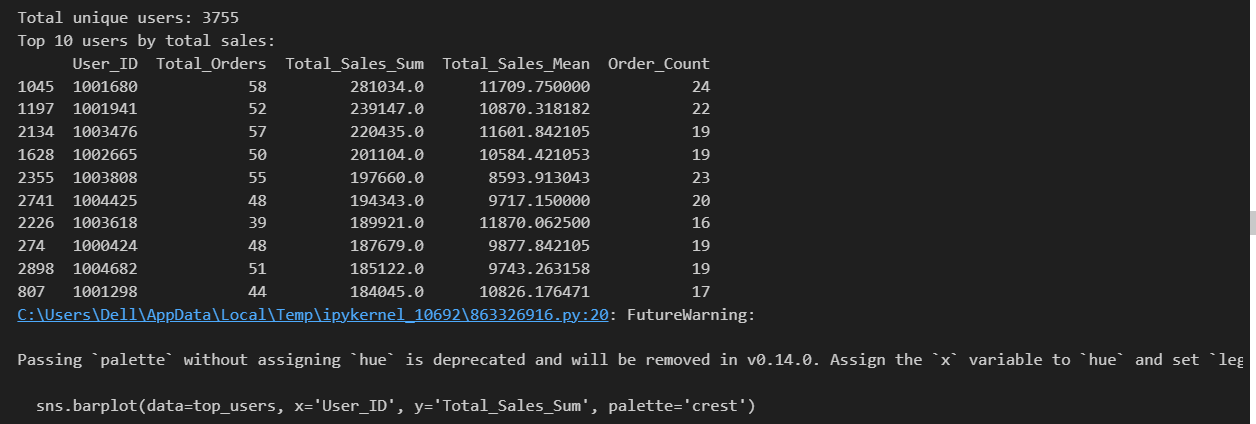
plt.xlabel('User ID')

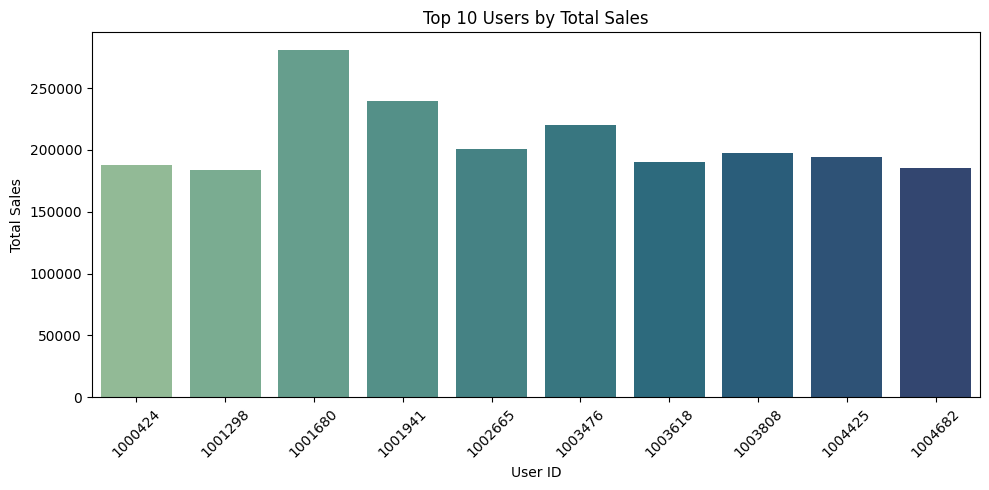
plt.ylabel('Total Sales')

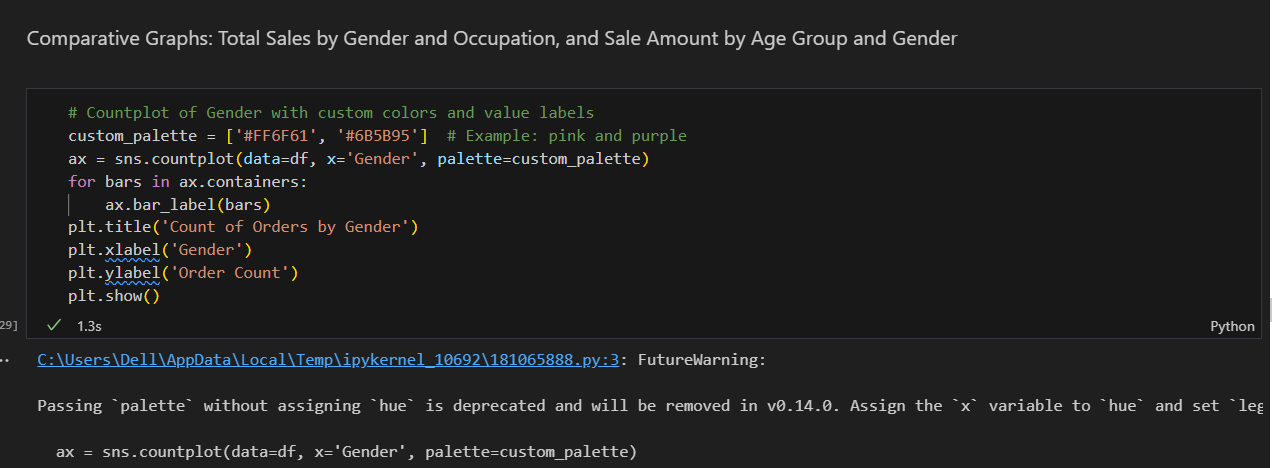
plt.xticks(rotation=45)

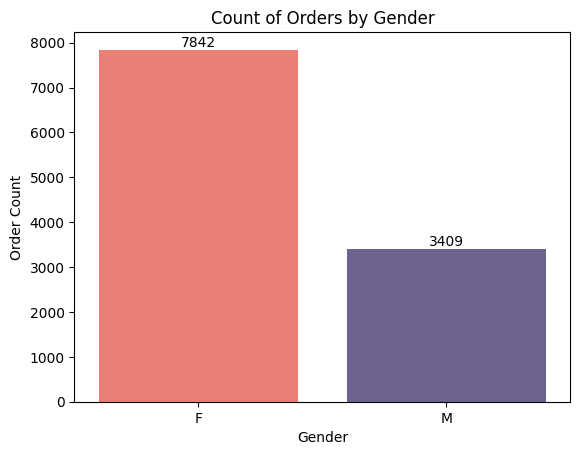
plt.tight\_layout()

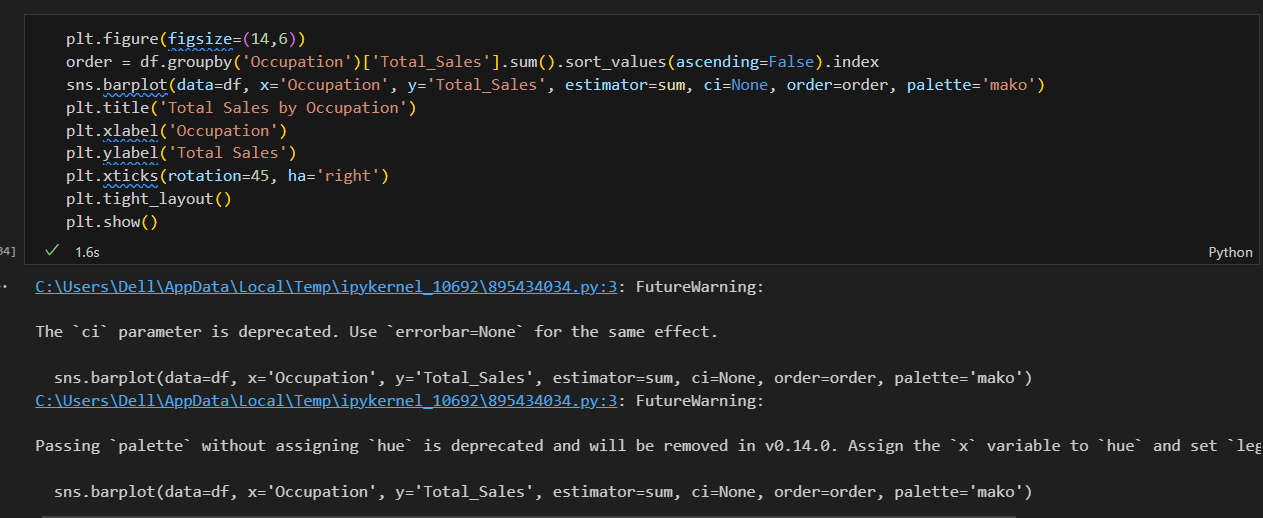
plt.show()

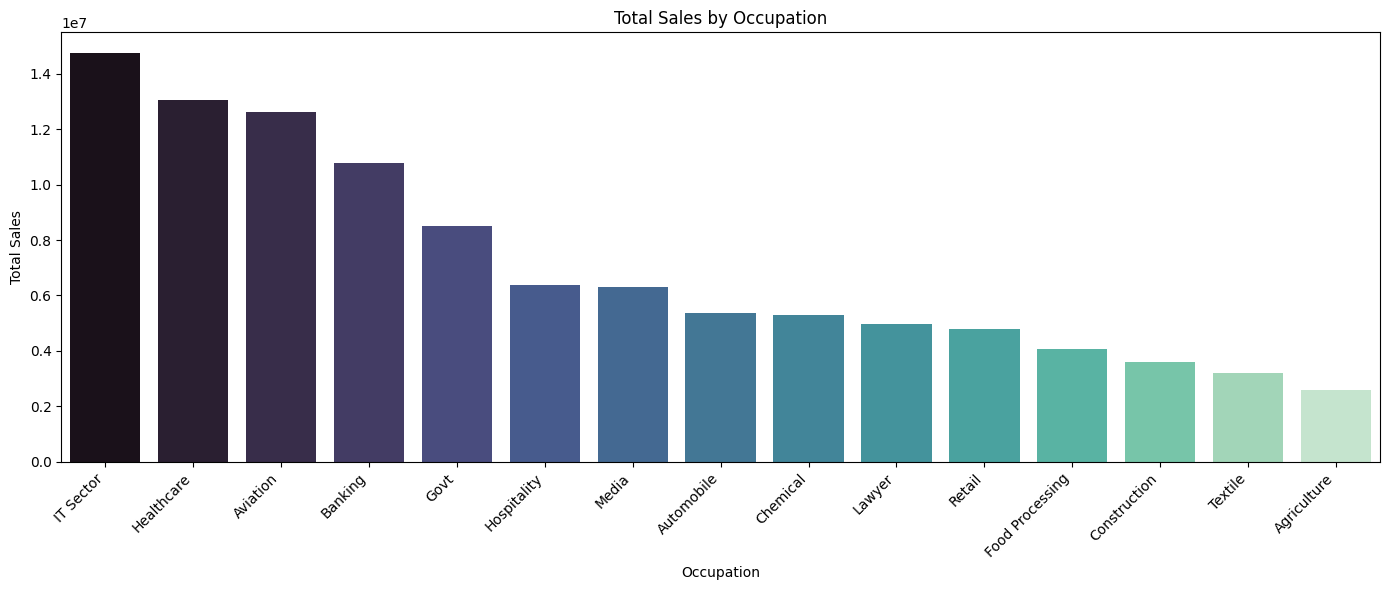
****

****

****

****

****

****

plt.figure(figsize=(12,6))

sns.boxplot(data=df, x='Age Group', y='Total\_Sales', hue='Gender', palette=custom\_palette)

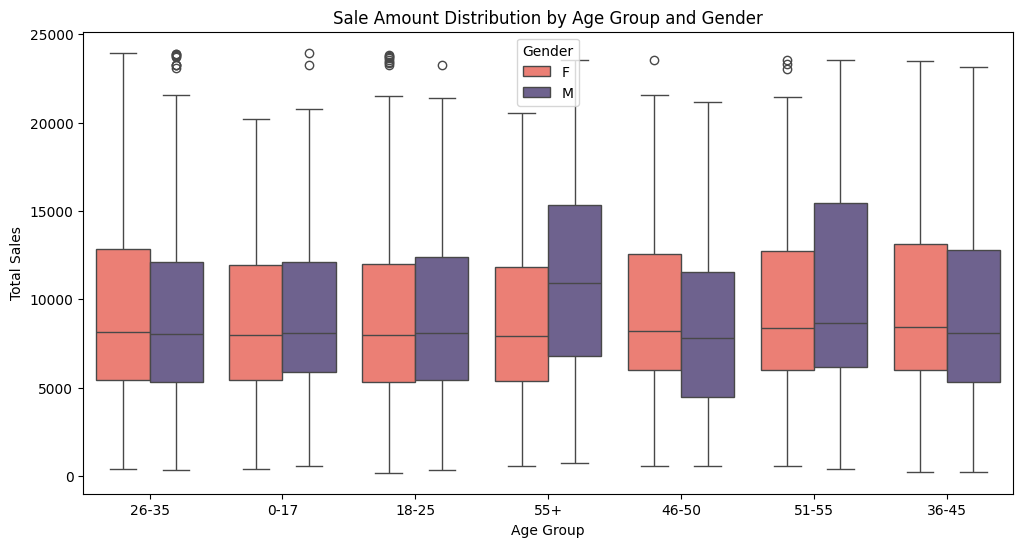
plt.title('Sale Amount Distribution by Age Group and Gender')

plt.xlabel('Age Group')

plt.ylabel('Total Sales')

plt.legend(title='Gender')

plt.show()

****

****

state\_gender\_group = df.groupby(['State', 'Gender']).agg({'Total\_Sales': 'sum', 'Total\_Orders': 'sum'}).reset\_index()

top\_states = state\_gender\_group.groupby('State')['Total\_Sales'].sum().sort\_values(ascending=False).head(10).index

filtered = state\_gender\_group[state\_gender\_group['State'].isin(top\_states)]

plt.figure(figsize=(16,7))

sns.barplot(data=filtered, x='State', y='Total\_Sales', hue='Gender', palette=custom\_palette)

plt.title('Total Sales by State and Gender (Top 10 States)')

plt.xlabel('State')

plt.ylabel('Total Sales')

plt.xticks(rotation=45, ha='right')

plt.legend(title='Gender')

plt.tight\_layout()

plt.show()

plt.figure(figsize=(16,7))

sns.barplot(data=filtered, x='State', y='Total\_Orders', hue='Gender', palette=custom\_palette)

plt.title('Total Orders by State and Gender (Top 10 States)')

plt.xlabel('State')

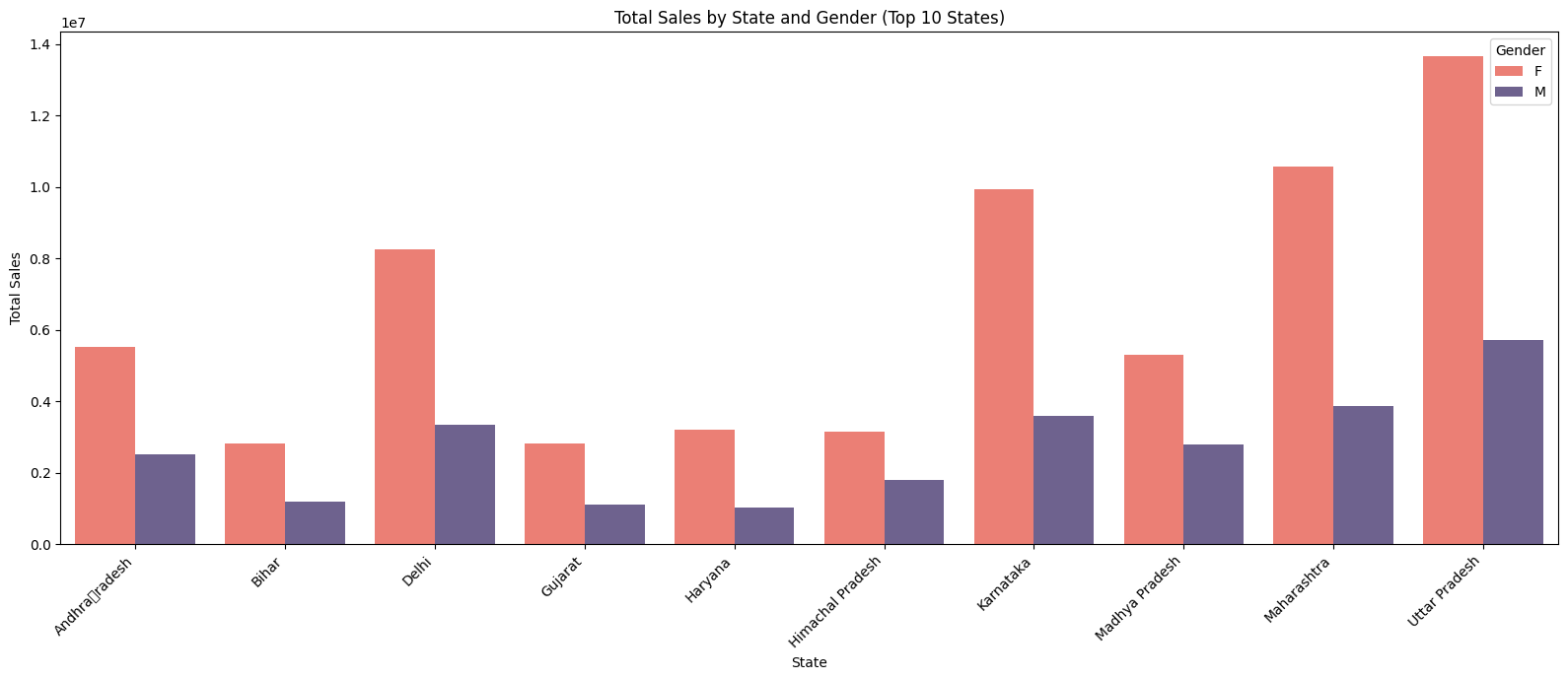
plt.ylabel('Total Orders')

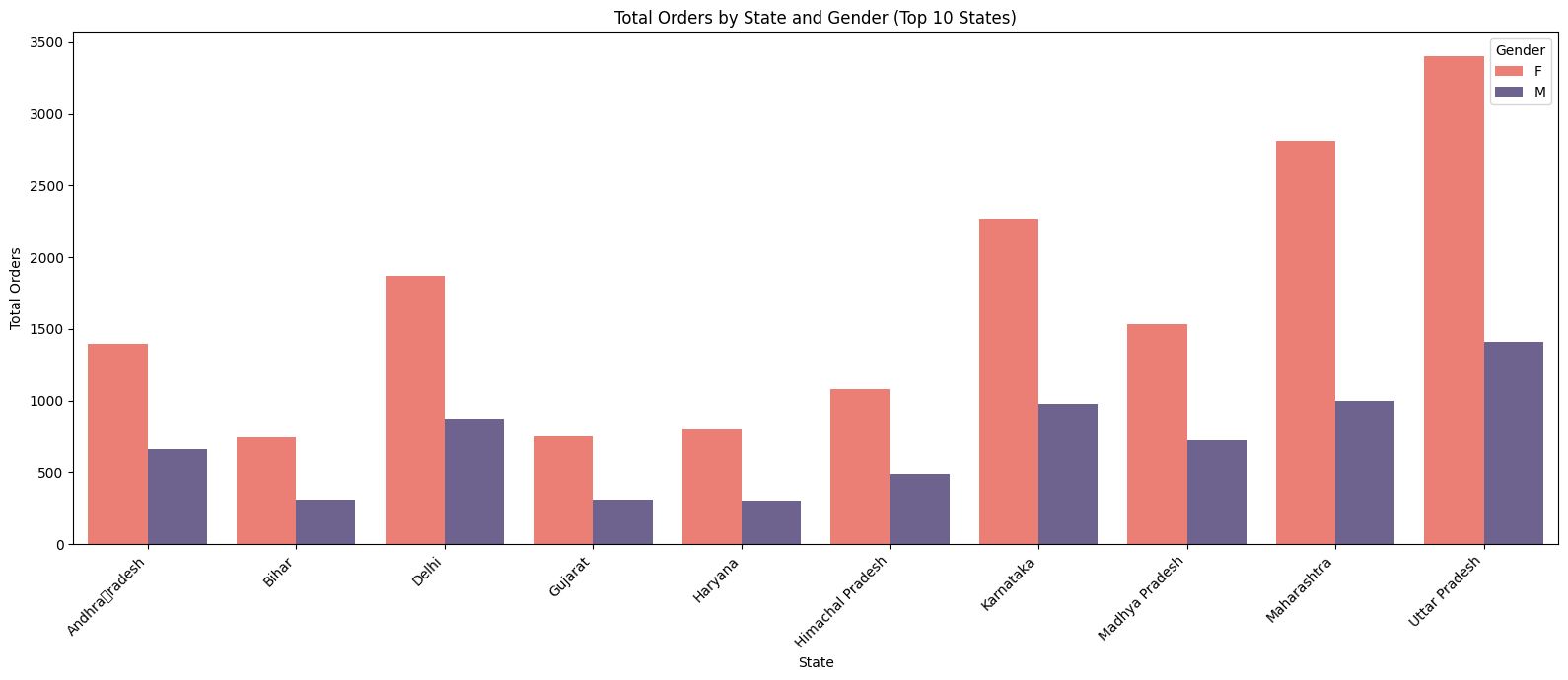
plt.xticks(rotation=45, ha='right')

plt.legend(title='Gender')

plt.tight\_layout()

plt.show()

****

****