In [ ]:
```
#!pip install openpyxl
#!pip install sqlalchemy
#!pip install mysql-connector-python

#importing the libraries, data and fetching the top 5 rows.
import matplotlib.pyplot as plt
import pandas as
from scipy.stats import pearsonr
df= pd.read_csv("C:/Users/admin/Downloads/amazon.csv")
df.head()

#To check for null values across the df
df.isnull().sum()

#Data cleaning, dividing the catgory column into main_category and sub_category
df['main_category']=df['category'].astype(str).str.split('|').str[0]
df['sub_category']=df['category'].astype(str).str.split('|').str[-1]
df.head()

#Dropping columns that wont be needed for analysis
df.drop(df.columns[8:16], axis=1, inplace=True)
df.drop('category', axis=1, inplace=True)
df.head()

#Converting values into correct dataype and reoving special characters such as (,%₹
df['discount_percentage']=df['discount_percentage'].replace('[%,]','', regex=True).
df['discounted_price']=df['discounted_price'].replace('[₹,]','', regex=True).astype
df['actual_price']=df['actual_price'].replace('[₹,]','', regex=True).astype(float).
df['rating']=pd.to_numeric(df['rating'], errors='coerce').astype(float)
df['rating_count']=df['rating_count'].replace('[,]','', regex=True).dropna().astype
df['product_name']=df['product_name'].replace('[~,:;]', " ", regex=True)
df.dropna()

#To remove special characters from the product_name column and to drop duplicates
df['product_name']=df['product_name'].replace('[~,:;]', " ", regex=True)
df=df.drop_duplicates(subset='product_id', keep=False)

#This helps to know how many products were sold in each category
df_count= df.groupby('main_category').size()
df_count.plot(kind='bar')

#Calculating Total Sales(disocunted_price) across all main categories
df_grouped=df.groupby('main_category')['discounted_price'].sum().reset_index()
df_grouped.sort_values(by='discounted_price', ascending=False)

#Calculating mean/average sale across the categories
df_average=df.groupby('main_category')['discounted_price'].mean().astype(int).reset
df_average.round(0).sort_values(by='discounted_price', ascending=False)

#Calculating maximum discounted_percentage across the main categories
df_percentage= df.groupby('main_category')['discount_percentage'].max().reset_index
df_percentage.sort_values(by='discount_percentage', ascending=False)

#Calculating the difference between the Total_actual_price & Total_Discounted_price
```

```
df_summary=pd.DataFrame({
    'actual_price': df.groupby('main_category')['actual_price'].sum(),
    'discouned_price': df.groupby('main_category')['discounted_price'].sum()})
df_summary['difference']= df_summary['discouned_price']-df_summary['actual_price']
df_summary
df_summary.reset_index(inplace=True)
df_summary.plot(x='main_category', y='difference', kind='bar') #plot

#Exporting data to sql to analyze the data and create a database
from sqlalchemy import create_engine
engine= create_engine('mysql+mysqlconnector://root:Yaalimadad10@127.0.0.1/practice1

df.to_sql('amazon_sales_data_pd', engine, index=False, if_exists='append')

#To check the correlation between rating and discount percent using pearson correla
correlation_co, p_value= pearsonr(df['rating'], df['discount_percentage'])
if p_value<=0.005:
    print('Reject Null Hypothesis, There is a correlation')
else:
    print('Fail to reject the null hypothesis, there is no significant avidence of

#To check whether the correlation between rating and discount percentage is strong
correlation_matrix = np.corrcoef(df['rating'], df['discount_percentage'])
correlation_coefficient = correlation_matrix[0, 1]

if correlation_coefficient == 1:
    print('Perfect Positive Correlation')
elif correlation_coefficient == -1:
    print('Perfect Negative Correlation')
elif correlation_coefficient == 0:
    print('No Correlation')
elif -1 < correlation_coefficient < 1:
    print('Strong Correlation')
else:
    print('Not within expected correlation range')
```

In [ ]:

In [ ]:

In [ ]: