import numpy as np

import pandas as pd

import io

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.cluster import KMeans

from sklearn.linear\_model import LinearRegression

from google.colab import files

uploaded = files.upload()

data = pd.read\_csv(io . BytesIO(uploaded['books\_review&rating.csv']))

print(data)

print(data.head(10))

df=pd.DataFrame(data)

data.shape

data.info()

\*\*slicing\*\*

tb=data[0:200]

print(tb)

# Data cleaning

print(data.isnull().sum())

print("after data\_cleaning")

print(data.mean())

data.fillna(data.mean(),inplace=True)

print(data.isnull().sum())

print(data)

#'Name' Top 30 books that have highest sold\*\*

Top\_sold = data['Name'].value\_counts().iloc[:20]

plt.figure(figsize=(7,14))

sns.barplot(x = Top\_sold.values , y = Top\_sold.index , color='pink', alpha=.7)

plt.title('Top 30 books that have highest sold', fontsize=16)

plt.xlabel('Count' , fontsize=15)

plt.ylabel('NAME OF THE BOOKS' , fontsize=15)

plt.xticks(fontsize=10 )

plt.yticks(fontsize=10)

plt.show()

#histogram

bins = 100

data.hist(bins=bins,figsize=(20,15))

plt.show()

# Bar Graph

x=(tb['Year'])

y=(tb['Reviews'])

y1=(tb['Price'])

plt.xlabel("Year")

plt.ylabel(" Review & Price")

plt.title("Yearwise Review of book")

plt.bar(x,y,color='orange',align='center',edgecolor='green',label=('Reviews'))

plt.bar(x,y1,color='blue',align='center',edgecolor='green',label=('Price'))

plt.legend()

plt.show()

#Scatter plot

plt.scatter(data['Author'],data['User Rating'],s=15,c="orange",alpha=1)

plt.title('Scatter plot AUTHOR vs USER RATING')

plt.xlabel('Author')

plt.ylabel('User Rating')

plt.show()

#Planline

x = np.linspace(0, 5, 100)

plt.plot(x, x, label='Name')

plt.plot(x, x\*\*2, label='Reviews')

plt.plot(x, x\*\*3, label='User Rating')

plt.xlabel('x label')

plt.ylabel('y label')

plt.title("Simple Plot")

plt.legend()

plt.show()

#BOXPLOT

#for individual attribute of dataframe

plt.figure(figsize=(7,5))

sns.boxplot(y=data['User Rating'])

plt.title('User Rating')

plt.show()

#For each numeric attribute of dataframe

data.plot.box()

#KMeans Algorithm

print('4 centroid for Rating & year')

df=pd.DataFrame(data[['Year','User Rating']])

tb=df[0:200]

kmeans = KMeans(n\_clusters=4).fit(tb)

centroids = kmeans.cluster\_centers\_

print(centroids)

plt.xlabel("Year")

plt.ylabel("User Rating")

plt.title("Yearwise Ratings of books")

plt.scatter(tb['Year'], tb['User Rating'], c=kmeans.labels\_.astype(float), s=60, alpha=0.5)

plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=60)

plt.show()

#Linear Regression

plt.xlabel("Year")

plt.ylabel("User Rating")

plt.title("Yearwise Rating of books")

x = tb.iloc[:, 0].values.reshape(-1, 1) # values converts it into a numpy array

y = tb.iloc[:, 1].values.reshape(-1, 1) # -1 means that calculate the dimension of rows, but have 1 column

linear\_regressor = LinearRegression() # create object for the class

linear\_regressor.fit(x, y) # perform linear regression

y\_pred = linear\_regressor.predict(x)

plt.scatter(x, y)

plt.plot(x, y\_pred, color='red',label=('Linear Regression'))

plt.legend()

plt.show()