import numpy as np *# linear algebra*

import pandas as pd *# data processing, CSV file I/O (e.g. pd.read\_csv)*

import os

for dirname, \_, filenames **in** os.walk('/kaggle/input'):

for filename **in** filenames:

print(os.path.join(dirname, filename))

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

df=pd.read\_csv('Mall\_Customers.csv’)

df.rename(columns={'Genre':'Gender'},inplace=True)

df.head()

df.describe()

df.isnull().sum()

df.drop(['CustomerID'],axis=1,inplace=True)

plt.figure(1,figsize=(15,6))

n = 0

for x **in** ['Age','Annual Income (k$)','Spending Score (1-100)']:

n +=1

plt.subplot(1,3,n)

plt.subplots\_adjust(hspace=0.5,wspace=0.5)

sns.distplot(df[x],bins=20)

plt.title('Distplot of **{}**'.format(x))

plt.show()

plt.figure(figsize=(15,5))

sns.countplot(y='Gender',data=df)

plt.show()

plt.figure(1,figsize=(15,6))

n = 0

for cols **in** ['Age','Annual Income (k$)','Spending Score (1-100)']:

n +=1

plt.subplot(1,3,n)

sns.set(style="whitegrid")

plt.subplots\_adjust(hspace=0.5,wspace=0.5)

sns.violinplot(x = cols,y = 'Gender',data=df)

plt.ylabel('Gender' if n== 1 else '')

plt.title('Violin Plot')

plt.show()

age\_18\_25 = df.Age[(df.Age >=18) & (df.Age <= 25)]

age\_26\_35 = df.Age[(df.Age >=26) & (df.Age <= 35)]

age\_36\_45 = df.Age[(df.Age >=36) & (df.Age <= 45)]

age\_46\_55 = df.Age[(df.Age >=46) & (df.Age <= 55)]

age\_55\_above = df.Age[(df.Age >= 56)]

age\_x =["18-25","26-35","36-45","46-55","55+"]

age\_y = [len(age\_18\_25.values),len(age\_26\_35.values),len(age\_36\_45),len(age\_46\_55),len(age\_55\_above)]

plt.figure(figsize = (15,6))

sns.barplot(x=age\_x, y=age\_y,palette = "mako")

plt.title("Number of Customer and Ages")

plt.xlabel("Age")

plt.ylabel("Number of Customer")

plt.show()

sns.relplot(x="Annual Income (k$)",y = "Spending Score (1-100)",data=df)

ss\_1\_20 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 1) & (df["Spending Score (1-100)"] <= 20)]

ss\_21\_40 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 21) & (df["Spending Score (1-100)"] <= 40)]

ss\_41\_60 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 41) & (df["Spending Score (1-100)"] <= 60)]

ss\_61\_80 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 61) & (df["Spending Score (1-100)"] <= 80)]

ss\_81\_100 = df["Spending Score (1-100)"][(df["Spending Score (1-100)"] >= 81) & (df["Spending Score (1-100)"] <= 100)]

ssx= ["1-20","21-40","41-60","61-80","81-100"]

ssy=[len(ss\_1\_20.values),len(ss\_21\_40.values),len(ss\_41\_60.values),len(ss\_61\_80.values),len(ss\_81\_100.values)]

plt.figure(figsize=(15,6))

sns.barplot(x=ssx,y=ssy, palette="rocket")

plt.title("Spending Scores")

plt.xlabel("Score")

plt.ylabel("Number of Customer having the Score")

plt.show()

ai\_0\_30 = df["Annual Income (k$)"][(df["Annual Income (k$)"] >= 0) & (df["Annual Income (k$)"] <= 30)]

ai\_31\_60= df["Annual Income (k$)"][(df["Annual Income (k$)"] >=31)& (df["Annual Income (k$)"] <=60)]

ai\_61\_90= df["Annual Income (k$)"][(df["Annual Income (k$)"] >=61)& (df["Annual Income (k$)"] <=90)]

ai\_61\_90=df["Annual Income (k$)"][(df["Annual Income (k$)"] >=91)& (df["Annual Income (k$)"] <=120)]

ai\_121\_150 = df["Annual Income (k$)"][(df["Annual Income (k$)"]>=121) & (df["Annual Income (k$)"] <=150)]

aix = ["$ 0 - 30,000","$ 30,001 - 60,000","$ 60,001 - 90,000","$ 90,001 - 120,000","$ 120,001 - 150,000"]

aiy = [len(ai\_0\_30.values),len(ai\_31\_60.values),len(ai\_61\_90.values),len(ai\_61\_90.values),len(ai\_121\_150.values)]

plt.figure(figsize=(15,6))

sns.barplot(x=aix,y=aiy,palette="Spectral")

plt.title("Annual Incomes")

plt.xlabel("Income")

plt.ylabel("Numer of Customer")

plt.show()

kmeans = KMeans(n\_clusters=4)

label = kmeans.fit\_predict(X1)

print(label)

print(kmeans.cluster\_centers\_)

plt.scatter(X1[:,0],X1[:,1],c=kmeans.labels\_,cmap='rainbow')

plt.scatter(kmeans.cluster\_centers\_[:,0],kmeans.cluster\_centers\_[:,1],color='black')

plt.title('Clusters of Customers')

plt.xlabel('Age')

plt.ylabel('Spending Score(1-100)')

plt.show

kmeans = KMeans(n\_clusters=5)

label = kmeans.fit\_predict(X2)

print(label)

print(kmeans.cluster\_centers\_)

plt.scatter(X2[:,0],X1[:,1],c=kmeans.labels\_,cmap='rainbow')

plt.scatter(kmeans.cluster\_centers\_[:,0],kmeans.cluster\_centers\_[:,1],color='black')

plt.title('Clusters of Customers')

plt.xlabel('Annual Income (k$)')

plt.ylabel('Spending Score(1-100)')

plt.show