PRACTICAL 4

Aim: Implementation of Data Visualization and Statistical Data Analysis.

**Bar Chart**

import pandas as pd

import matplotlib.pyplot as plt

df1 = pd.read\_csv('Xyz.csv')

df1.plot(x="Name of the Contestant", y="Total", kind ="bar")

**Box Plot**

import numpy as np

import pandas as pd

from numpy import percentile

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read\_csv("Xyz.csv")

h = data

df = pd.DataFrame(data = h , columns = ['Round 1', 'Round 2', 'Round 3', 'Round 4', 'Round 5', 'Round 6', 'Round 7' , 'Round 8' , 'Round 9' , 'Round 10'])

my\_colors = []

for i in range(df.shape[1]):

  df.iloc[:, i]

  if i >= 8 :

    my\_colors.append("blue")

  elif i < 8 and i >= 6:

    my\_colors.append("green")

  elif i < 6:

    my\_colors.append("red")

plt.figure(figsize=(16,8))

sns.boxplot(data = h, palette=my\_colors)

plt.xlabel('Rounds')

plt.ylabel('Scores')

plt.show()

**Histogram**

[ ]

import matplotlib.pyplot as plt

import numpy as np

data=pd.read\_csv("Xyz.csv")

plt.hist(data['Total'])

plt.show()

**Scatter Plot**

[ ]

import pandas as pd

import matplotlib.pyplot as plt

df1=pd.read\_csv("Xyz.csv")

x = list(df1['Name of the Contestant'])

y = list(df1['Total'])

plt.scatter(x,y,marker=".",s=100,edgecolors="black",c="blue")

plt.title("Runs scored by players")

plt.show()

**Quantile-Quantile plot**

[ ]

import statsmodels.api as sm

import pylab as py

data=pd.read\_csv("Xyz.csv")

a=data['Round 10']

sm.qqplot(a,line='45')

py.xlabel('Round')

py.ylabel('Score')

py.show()

**Chi-Square Test**

[ ]

import pandas as pd

import numpy as np

df = pd.read\_excel(r'/content/Xyz\_newDataset.xlsx')

i = 0

rows = len(df)

columns = len(df.columns)

estimateArr = np.zeros((rows, columns))

while(i < len(df)-1):

    j = 1

    while(j < len(df.columns)-1):

      expect = (df.iloc[len(df)-1,j] \* df.iloc[i,len(df.columns)-1])// df.iloc[len(df)-1,len(df.columns)-1]

      estimateArr[i][j] = expect;

      j = j + 1

    i = i + 1

print(estimateArr)

import math

import pandas as pd

import numpy as np

chiSquare = 0

i = 0

while(i < len(df)-1):

    j = 1

    while(j < len(df.columns)-1):

      subelem = df.iloc[i][j] - estimateArr[i][j]

      chiSquare = chiSquare + (math.pow(subelem, 2) / estimateArr[i][j])

      j = j + 1

    i = i + 1

print("The value is:",chiSquare)

# Calculating Mean of the data

j = 1

meanArr = np.zeros(len(df.columns)-2)

count = 0;

while( j < len(df.columns)-1):

  i = 0

  sum = 0;

  mean = 0;

  while( i < len(df)-1):

     sum = sum + df.iloc[i,j]

     i = i + 1

  mean = sum / i

  meanArr[count] = mean

  count = count + 1

  j = j + 1

print(meanArr)

#Calculating the Standard Deviation of the data

j = 1

varianceArr = np.zeros(len(df.columns)-2)

count  = 0

while( j < len(df.columns)-1):

  i = 0

  variance = 0

  sd = 0

  while( i < len(df)-1):

     values = df.iloc[i,j] - meanArr[count]

     variance = variance + math.pow(values, 2)

     i = i + 1

  variance = variance / i

  sd = math.sqrt(variance)

  varianceArr[count] = sd

  count = count + 1

  j = j + 1

print(meanArr)

print(varianceArr)

**Pearson's Coefficient**

[ ]

i·=·0

count··=·0

prodSUM·=·0

while(·i·<·len(df)-1):

··j·=·1

··multi·=·1

··while(·j·<·len(df.columns)-1):

·····values·=·df.iloc[i][j]·-·meanArr[j-1]

·····multi·=·multi·\*·(values)

·····j·=·j·+·1

··

··prodSUM·=·prodSUM·+·multi

··i·=·i·+·1

varianceProd·=·1

for·i·in·varianceArr:

····varianceProd·=·varianceProd·\*·i

print(prodSUM)

print(varianceProd·\*·len(df)-1)·

rvalue·=·prodSUM·/·(varianceProd·\*·len(df)-1)·

print(rvalue)