PROJECT 111 – SINGLE SAMPLE Z-TESTS

from google.colab import files

data\_to\_load=files.upload()

import plotly.figure\_factory as ff

import plotly.graph\_objects as go

import statistics

import random

import pandas as pd

import csv

df = pd.read\_csv("medium\_data.csv")

data = df["reading\_time"].tolist()

fig=ff.create\_distplot([data], ['reading time'], show\_hist=False)

fig.show()

pm=statistics.mean(data)

print("Mean of population is ", pm)

def rsom(counter):

  dataset=[]

  for i in range(0, 30):

    rindex=random.randint(0, len(data))

    value=data[rindex]

    dataset.append(value)

  mean=statistics.mean(dataset)

  return mean

def show\_fig(mlist):

    df = mlist

    smean = statistics.mean(mlist);

    std = statistics.stdev(mlist);

    fstds, fstde = smean - std, smean + std;

    sstds,sstde = smean - (2\*std),smean + (2\*std);

    tstds,tstde = smean - (3\*std),smean + (3\*std);

    print("Standard Deviation 1:",fstds,",",fstds);

    print("Standard Deviation 2:",sstds,",",sstde);

    print("Standard Deviation 3:",tstds,",",tstde);

    fig.add\_trace(go.Scatter(x = [fstds, fstds],y = [0,0.8]));

    fig.add\_trace(go.Scatter(x = [fstde, fstde],y = [0,0.8]));

    fig.add\_trace(go.Scatter(x = [sstds, sstds],y = [0,0.8]));

    fig.add\_trace(go.Scatter(x = [sstde, sstde],y = [0,0.8]));

    fig.add\_trace(go.Scatter(x = [tstds, tstds],y = [0,0.8]));

    fig.add\_trace(go.Scatter(x = [tstde, tstde],y = [0,0.8]));

    fig.add\_trace(go.Scatter(x = [smean,smean],y = [0,0.8]));

    sample\_mean = statistics.mean(data);

    fig.add\_trace(go.Scatter(x = [sample\_mean,sample\_mean],y = [0,0.8]));

    zscore = (sample\_mean - smean)/std;

    print("The z score is ",zscore);

    fig.show();

def setup():

  mlist=[]

  for i in range(0, 100):

    smeans=rsom(30)

    mlist.append(smeans)

  show\_fig(mlist)

  mean2=statistics.mean(mlist)

  std2=statistics.stdev(mlist)

  print("Sampling mean is ", mean2)

  print("Sampling Distribution Standard Distribution is ", mean2)

setup()

