# Python program to get variance of a list

# Importing the NumPy module

import numpy as np

# Taking a list of elements

listu = [2, 4, 4, 4, 5, 5, 7, 9]

listu1 = [0.7798, 0.9334, 0.9302, 0.9396, 0.9227, 0.9349] # Brainstem

listu2 = [0.6368, 0.0819, 0.5929, 0.5924,0.6871, 0.7077] # 2/chiasm

listu3=[0.8041, 0.6837, 0.6526, 0.7479, 0.4636, 0.6907] # 3/ pituitary

listu4 = [0.8217, 0.7196, 0.7239, 0.7625, 0.7973, 0.8543] #4 / hippocapus \_r

listu5 =[ 0.8229, 0.7648, 0.7776, 0.7352, 0.6482, 0.8020] # 5 Hippocampus L

listu6 = [ 0.7404, 0.7741, 0.6551, 0.5811, 0.5926, 0.7602] # opticnerve R

listu7 = [0.7104, 0.5395, 0.5734, 0.5250, 0.6465, 0.6668] # Opticnerve \_L

listu8 = [0.5199, 0.0017, 0.5526, 0.7483, 0.6391, 0.6547] # 8 optictract R

listu9 = [0.7598, 0.6458, 0.6670, 0.5404, 0.0, 0.6321] # optictract\_l

'''nnUNet down below'''

# Calculating variance using var()

listf1 =[0.8669, 0.9334, 0.9304, 0.9421, 0.9286, 0.9277] # BRIANSTEM 1

listf2 = [0.6448, 0.0973, 0.6389, 0.5955, 0.7064, 0.7004] # 2 chiasm

listf3 = [0.5967, 0.7395, 0.6909, 0.6486, 0.5382,0.6410] # 3

listf4 = [ 0.7901, 0.7281, 0.7284, 0.7963, 0.8552, 0.8245] # 4

listf5 =[0.7822, 0.7921, 0.7753, 0.7423, 0.7002, 0.7884] # 5

listf6 =[0.7433, 0.7849, 0.6841, 0.6152, 0.6057, 0.7409] #6

listf7 = [0.7162, 0.5897, 0.5775, 0.5075, 0.6606, 0.6457] # 7

listf8 = [0.5228, 0.0247, 0.5307, 0.7518, 0.6754, 0.5311] # 8

listf9 = [0.7218, 0.6262, 0.6893, 0.5623, 0.0, 0.5803]

# print(np.var(list))

'''

import json

# Load JSON file into a variable

with open('//onerm.dk/nfpdata/Begraenset/AUHDCENP\_BRAIN\_RT\_QA/DCPT/nnFormer\_predictions/summary.json', 'r') as json\_file:

data = json.load(json\_file)

# Extract specific field

print(data["results"]["all"])

'''

''''''

# Wilcoxon signed-rank test

from numpy.random import seed

from numpy.random import randn

from scipy.stats import wilcoxon

# seed the random number generator

# generate two independent samples

# data1 = [0.00062, 0.044, 0.0412, 0.0021, 0.001, 0.0044, 0.0049, 0.0537, 0.0059] # nnunet w. 54

# data2 = [0.0032, 0.0456, 0.011, 0.0024, 0.0031, 0.0061, 0.0046, 0.059, 0.062] # w. 54

# data1 = [ 0.0088] # nnunet w. 54

# data2 = [ 0.0065] # w. 54

# compare samples

'''

stat, p = wilcoxon(data1, data2)

print('Statistics=%.3f, p=%.3f' % (stat, p))

# interpret

alpha = 0.05

if p > alpha:

print('Same distribution (fail to reject H0)')

else:

print('Different distribution (reject H0)')

'''

# Kruskal-Wallis H-test

from numpy.random import seed

from numpy.random import randn

from scipy.stats import kruskal

# seed the random number generator

seed(1)

# generate three independent samples

# data1 = 5 \* randn(100) + 50

# data2 = 5 \* randn(100) + 50

# compare samples

'''

stat, p = kruskal(data1, data2)

print('Statistics=%.3f, p=%.3f' % (stat, p))

# interpret

alpha = 0.05

if p > alpha:

print('Same distributions (fail to reject H0)')

else:

print('Different distributions (reject H0)')

'''

# haus95 nnunet

list1 = [9.0, 1.9433, 1.7212, 1.4731, 2.0, 2.10241]

list2 = [1.6581, 1.5556, 1.8551, 1.5878, 4.5276, 1.5878]

list3 = [1.8262, 1.5, 1.8551, 2.5671, 2.8879, 1.8262]

list4 = [2.1024, 1.1646, 2.1424, 2.9017, 2.4034, 2.4034]

list5 = [2.6062, 2.4839, 2.3688, 2.7704, 4.6667, 3.0778]

list6 = [1.1646, 2.3143, 2.2592, 1.1275, 1.1646, 1.5350]

list7 = [1.4439, 2.2277, 2.8879, 2.0667, 1.6710, 1.89057]

list8 = [2.0, 1.5, 0.8902, 1.5494, 1.9556]

list9 = [1.6405, 1.1330, 1.4439, 2.0, 7.8330, 2.2579]

# haus 95 nnFormer

list11 = [15.5076, 1.8262, 1.6710, 1.5, 2.1313, 1.7804]

list22 = [1.6470, 2.1275, 2.0, 1.7804, 6.2706, 1.8262]

list33 = [1.5350, 1.2419, 2.1737, 1.5, 3.1756, 1.7212]

list44 = [1.5, 2.1659, 3.5052, 2.8879, 2.5908, 1.8262]

list55 = [1.8262, 2.9732, 2.1275, 2.9345, 6.5605, 2.3688]

list66 = [1.1554, 2.1275, 2.3143, 1.4439, 1.1554, 1.5624]

list77 = [1.2419, 3.0, 2.9481, 2.2550, 2.0667, 1.7804]

list88 = [1.5, 1.7212, 1.0, 1.6710, 8.2085, 2.6075]

list99 = [1.5624, 1.5556, 1.4439, 2.3143, 9.1915, 1.8262]

#print(np.var(list8))

#data1 = [7.1465, 1.1607, 0.2342, 0.2764, 0.6093, 0.2585, 0.2113, 6.5881, 5.3645] ## nnunet

#data2 = [26.2023, 2.7058, 0.4091, 0.4481, 2.5142, 0.2008, 0.3855, 6.109, 7.7925] # foremr

#data1 = [0.1583]

#data2 = [0.0980]

stat, p = kruskal(listu9, listf9)

print('Statistics=%.3f, p=%.3f' % (stat, p))

# interpret

alpha = 0.05

if p > alpha:

print('Same distributions (fail to reject H0)')

else:

print('Different distributions (reject H0)')