**Московский авиационный институт**

**(Национальный исследовательский университет)**

Факультет прикладной математики и физики

Кафедра вычислительной математики и программирования

**Лабораторная работа № 2**

по курсу «Анализ и планирование эксперимента»

Тема: Сравнение критерия Вилкоксона и Стьюдента.

Студент Чекушкин Д.И.

Группа: 80-404Б

Преподаватель: Горяинов А.В.

Оценка:

Москва, 2020

Задание.

1) Необходимо реализовать генератор данных, который способен выдавать зашумленные данные.

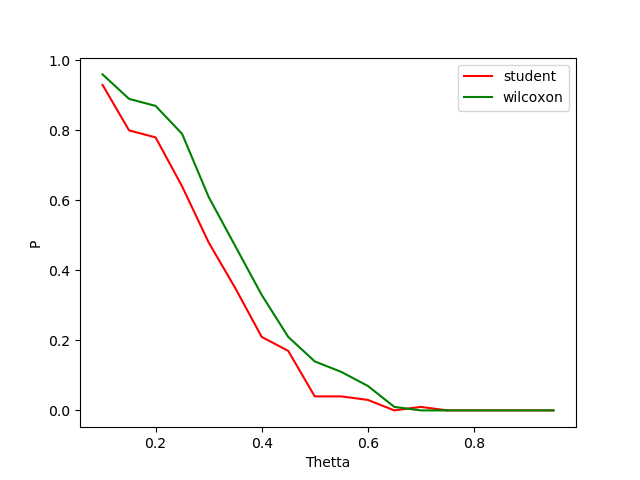
2) Реализовать критерий Вилкоксона и Стьюдента.

3) Сравнить полученные результаты.

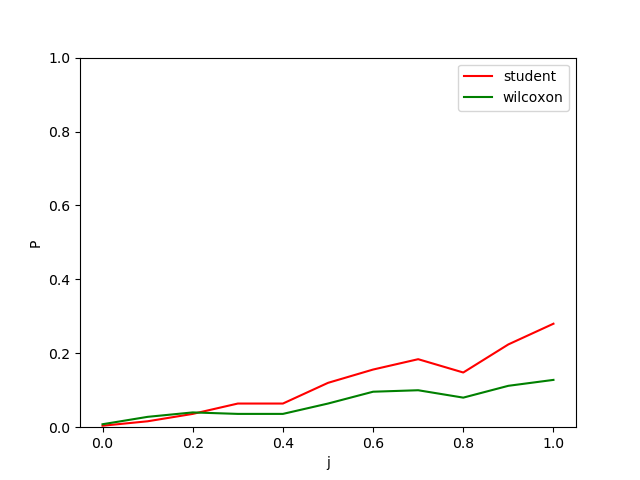
Ход работы.

Был реализован генератор. Далее проводилось сравнение точности двух методов при прибавлении коэффициента „thetta“ к мат. ожиданию. То есть сравнивались выборки X(M,D) и Y(M+thetta,D). Результаты получились следующие:

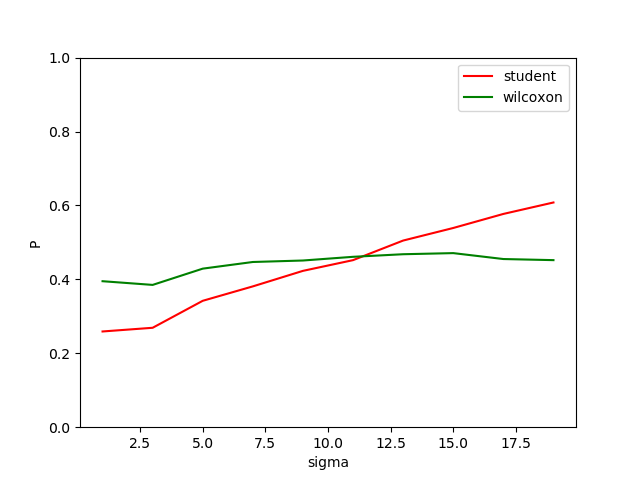
График зависит от вероятности попадания в доверительную область при различных тетта. Критерий Стюдента более точный. Чем больше тетта тем меньше ошибка, но Стьюдент точнее.



Второй график это зависимость зашумленности от вероятности попадание в ДО. МО для X и Y равны соответственно 0 и 2. Критерий Вилкоксона со временем становится точнее.



Третий график это зависимость силы выброса от вероятности попадание в ДО при зашумленности в 10%. Критерий Вилкоксона также со временем становится точнее.



Выводы.

В этой лабораторной работе были реализованы два метода «решения» задачи о сдвиге. При отсутствии шума критерий Стьюдента показал себя лучше. При сильных выбросах и наличии шума точнее оказался критерий Вилкоксона.

Код программы.

import random

import numpy as np

import math

import matplotlib.pyplot as plt

def gen3(thetta,j,sigma,n):

i=0

u = []

v = []

w = []

z = []

while i<n-1:

u.append(random.normalvariate(thetta, 1))

v.append(random.normalvariate(thetta, sigma))

w.append(random.uniform(thetta, 1))

i+=1

i = 0

for el in w:

if w[i]>j:

z.append(u[i])

i+=1

elif w[i]<j:

z.append(v[i])

i+=1

return(z)

def stud(X,Y):

Xm = np.mean(X)

Ym = np.mean(Y)

sum1 = 0

i = 0

m = len(X)-1

while i<=m:

sum1+=(X[i]-Xm)\*\*2

i+=1

sum2 = 0

i = 0

n = len(Y)-1

while i<=n:

sum2+=(Y[i]-Ym)\*\*2

i+=1

S = 1/(m+1+n+1-2)\*(sum1+sum2)

T = (Xm-Ym)/(math.sqrt(S\*(1/(m+1)+1/(n+1))))

print(Xm,Ym,'\n','sum1,2:',sum1,',',sum2,'\n','S:',S,'T:',T)

if (T>=-2.1) and (T<=2.1): # попадает ли в ДО

return(1)

else:

return(0)

def wilk(X,Y):

XX = []

for el in X:

XX.append([el,'X'])

YY = []

for el in Y:

YY.append([el,'Y'])

untd = XX + YY

untd = sorted(untd, key=lambda untd: untd[0])

W = 0

i = 1

for el in untd:

if el[1]=='Y':

W+=i

i+=1

else:

i+=1

m = len(X)

n = len(Y)

mw = n/2\*(m+n+1)

dw = (m\*n)/12\*(m+n+1)

w\_final = (W - mw)/(math.sqrt(dw))

print('m,n',m,n,'\n','mw,dw',mw,dw,'\n','w\_final',w\_final)

if (w\_final>=-2.4) and (w\_final<=2.4): # попадает ли в ДО

return(1)

else:

return(0)

P1 = [] # stud

P2 = [] # wilk

T = []

t = 0.1

while t<=1:

T.append(t)

res1 = 0

res2 = 0

counter = 1

while counter <=100:

X = gen3(0,0,1,101)

Y = gen3(0+t,0,1,101)

stud\_res = stud(X,Y)

#print('stud\_res',stud\_res)

wilk\_res = wilk(X,Y)

#print('wilk\_res',wilk\_res)

if stud\_res == 1:

res1+=1

if wilk\_res == 1:

res2+=1

counter+=1

print(res1,res1)

res1=res1/100

res2 = res2/100

P1.append(res1) # stud

P2.append(res2) # wilk

t+=0.05

plt.xlabel("Thetta")

plt.ylabel("P")

plt.plot(T, P1,'r', label='student')

plt.plot(T, P2,'g', label='wilcoxon')

plt.legend()

plt.show()

plt.close()

P1 = [] # stud

P2 = [] # wilk

T = []

t = 0.1

while t<=1:

T.append(t)

res1 = 0

res2 = 0

counter = 1

while counter <=100:

X = gen3(0,t,1,101)

Y = gen3(0,t,1,101)

stud\_res = stud(X,Y)

#print('stud\_res',stud\_res)

wilk\_res = wilk(X,Y)

#print('wilk\_res',wilk\_res)

if stud\_res == 1:

res1+=1

if wilk\_res == 1:

res2+=1

counter+=1

print(res1,res1)

res1=res1/100

res2 = res2/100

P1.append(res1) # stud

P2.append(res2) # wilk

t+=0.1

plt.xlabel("j")

plt.ylabel("P")

plt.plot(T, P1,'r', label='student') #

plt.plot(T, P2,'g', label='wilcoxon')

plt.legend()

plt.show()

plt.close()

P1 = [] # stud

P2 = [] # wilk

T = []

t = 1

while t<=10:

T.append(t)

res1 = 0

res2 = 0

counter = 1

while counter <=100:

X = gen3(0,0.1,1,101)

Y = gen3(0,0.1,1+t,101)

stud\_res = stud(X,Y)

#print('stud\_res',stud\_res)

wilk\_res = wilk(X,Y)

#print('wilk\_res',wilk\_res)

if stud\_res == 1:

res1+=1

if wilk\_res == 1:

res2+=1

counter+=1

print(res1,res1)

res1=res1/100

res2 = res2/100

P1.append(res1) # stud

P2.append(res2) # wilk

t+=1

plt.xlabel("sigma")

plt.ylabel("P")

plt.plot(T, P1,'r', label='student')

plt.plot(T, P2,'g', label='wilcoxon')

plt.legend()

plt.show()

plt.close()