МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В.Г.ШУХОВА»

(БГТУ им. В.Г.Шухова)

**Лабораторная работа № 0**

Дисциплина: Теория надежности

Выполнил: студент группы ВТ-31

Ковалёв И. Д.

Проверил:

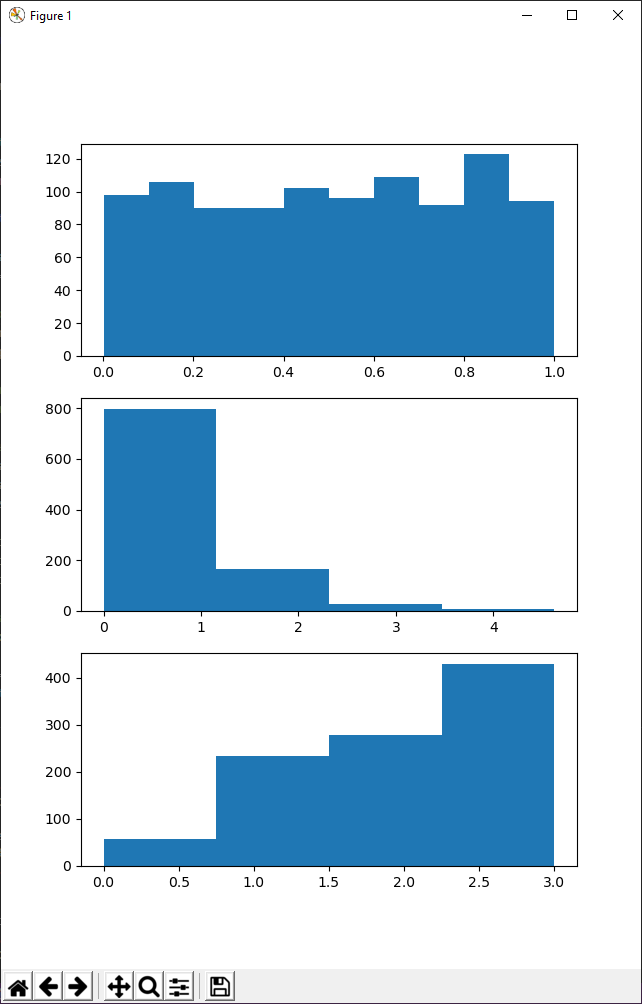
Кабалянц П. С

Белгород 2020

**Вариант №7**

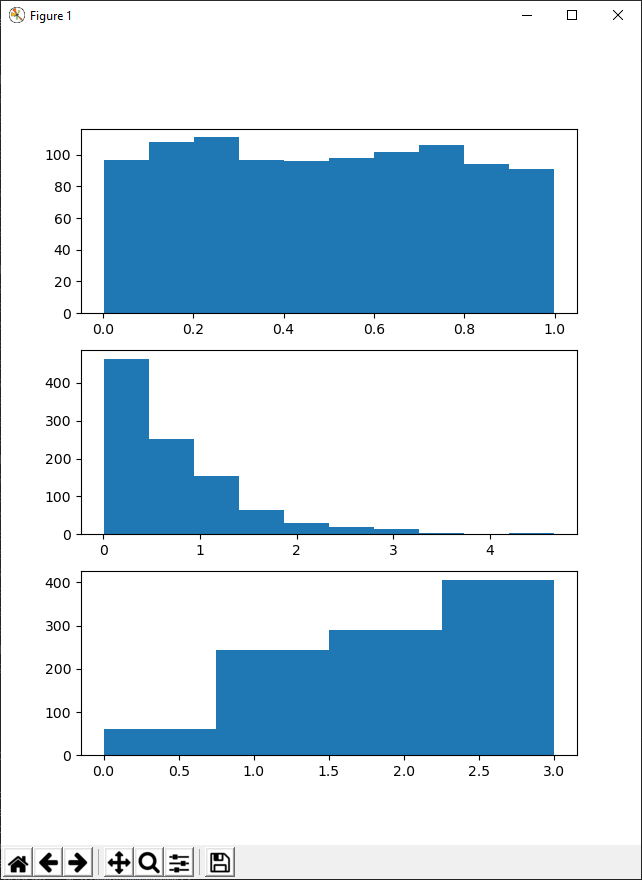
**Непрерывная случайная величина**

Сгенерируем 1000 случайных величин с равномерным распределением. Получим:



;

Получим по данной формуле набор случайных величин показательного распределения.



Вычислим среднеарифметическое и дисперсию:

Среднее: 0.718

Дисперсия: 0.537

Проверим гипотезу о том, что полученная выборка соответствует показательному закону распределения. Используем критерий Пирсона ()

= 1.752 <= 6, следовательно, гипотеза верна.

**Дискретная случайная величина**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 |
|  |  | 0.227 | 0.295 | 0.409 |

Вычислим среднеарифметическое и дисперсию:

Среднее: 2.043

Дисперсия: 0.910

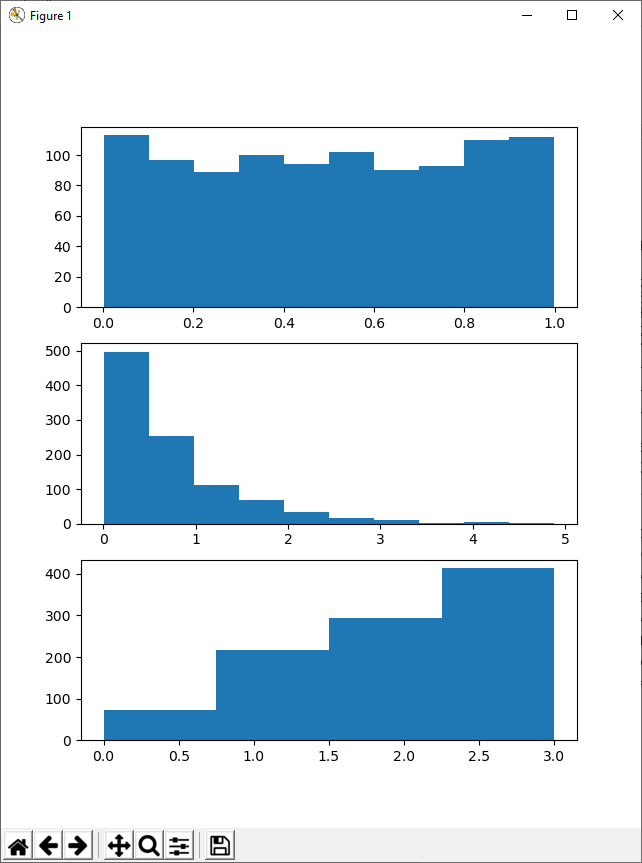
Вычислим теоретические и фактические частоты:

Теоретические частоты:

[68, 227, 295, 409]

Полученные частоты:

[74, 218, 295, 413]



Проверим гипотезу о том, что полученная выборка соответствует показательному закону распределения. Используем критерий Пирсона ()

= 0.925 <= 6, следовательно, гипотеза верна.

# This Python file uses the following encoding: utf-8

import random

import math

import numpy

from prettytable import PrettyTable

import matplotlib.pyplot as pyplot

def getRandomNumbers(sz):

    result = []

    for i in range (0, sz):

        result.append(random.random())

    return result

if \_\_name\_\_ == "\_\_main\_\_":

    l = 7/5

    sz = 1000

    #Получаем случайные числа

    randomNumbers = getRandomNumbers(sz)

    expRandomNumbers = numpy.multiply(numpy.log(randomNumbers), - 1/l)

    #print("Полученные числа: ")

    #print(numpy.round(expRandomNumbers, 3))

    #Вычисляем мат ожидание и дисперсию

    mean = numpy.mean(expRandomNumbers)

    mean2 = numpy.mean(numpy.square(expRandomNumbers))

    disp = mean2 - mean\*\*2

    print("Непрерывная СВ")

    print("Среднее: ", numpy.mean(expRandomNumbers))

    print("Дисперсия: ", disp)

    #Находим интервалы разбиения

    max = numpy.max(expRandomNumbers)

    x = numpy.linspace(0, max, num = 5)

    intervals = [0, 0, 0, 0, 0] # n

    n = [0, 0, 0, 0]

    for X in expRandomNumbers:

        i = 0

        while not (x[i] < X <= x[i+1]):

            i+=1

        n[i]+=1

    p = [0, 0, 0, 0]

    for i in range (0, 4):

        p[i] = math.exp(-l \* x[i]) - math.exp(-l \* x[i+1])

    nt = numpy.multiply(p, sz)

    k = 0

    for i in range (0, 4):

        k += (n[i] - nt[i]) \*\* 2 / n[i]

    if (k <= 6):

        print ("K\_в = ", round(k, 3), "<= 6, ==> гипотеза верна")

    else :

        print ("K\_в = ", round(k, 3), ">= 6, ==> гипотеза не верна")

    # ДИСКРЕТНАЯ СВ

    i = 3

    j = 10

    k = 9

    p[0] = round(i / (2 \* (i + j + k)), 3)

    p[1] = round(j / (2 \* (i + j + k)), 3)

    p[2] = round((i + j) / (2 \* (i + j + k)), 3)

    p[3] = round((2 \* k) / (2 \* (i + j + k)), 3)

    n = [0, 0, 0, 0]

    for i in range(0, sz):

        if (randomNumbers[i] <= p[0]):                                               n[0] += 1

        elif (p[0] < randomNumbers[i] <= p[0] + p[1]):                               n[1] += 1

        elif (p[0] + p[1] < randomNumbers[i] <= p[0] + p[1] + p[2]):                 n[2] += 1

        elif (p[0] + p[1] + p[2] <= randomNumbers[i]) :                              n[3] += 1

    print("Дискретная СВ")

    print("Закон распределения СВ согласно варианту:", p)

    avg = 0

    for i in range (0, 4):

        avg += i \* p[i]

    dispersion = 0

    for i in range (0, 4):

        dispersion += i\*\*2 \* p[i]

    dispersion -= avg\*\*2

    print("Среднее:", avg)

    print("Дисперсия: ", dispersion)

    n1 = [0, 0, 0, 0]

    for i in range (0, 4):

        n1[i] = sz \* p[i]

    print("Теоретические частоты: ")

    print(n1)

    print("Полученные частоты: ")

    print(n)

    k = 0

    for i in range(0, 4):

        k += ((n[i] - n1[i])\*\*2)/n1[i]

    if (k <= 6):

        print ("K\_в = ", round(k, 3), "<= 6, ==> гипотеза верна")

    else :

        print ("K\_в = ", round(k, 3), ">= 6, ==> гипотеза не верна")

    dExprandomNumbers = []

    for i in range (len(n)):

        for k in range(int(n[i])):

            dExprandomNumbers.append(i)

    title, p = pyplot.subplots(3)

    p[0].hist(randomNumbers)

    p[1].hist(expRandomNumbers, 10)

    p[2].hist(dExprandomNumbers, 4)

    pyplot.show()