ГУАП

КАФЕДРА № 43

ОТЧЁТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| к.т.н., доцент |  |  |  | В. В. Мышко |
| должность, уч. Степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №3 | | | | | |
| проверка гипотез о параметрах законов распределения | | | | | |
| по дисциплине: ОБРАБОТКА ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ | | | | | |
|  | | | | | |
| РАБОТУ ВЫПОЛНИЛ | | | | | |
| СТУДЕНТ ГР. | 4931 |  |  |  | А.А. Кинько |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2023

**Текст задания**

Согласно варианту №4931-12:

Для случайных величин X и Y проверить гипотезу о равенстве математических ожиданий на основе заданных массивом экспериментальных данных.

Порядок выполнения задания:

1. Найти оценки математических ожиданий по заданным массивам экспериментальных данных (таблица №1)

2. Проверить нулевую гипотезу о равенстве математических ожиданий при конкурирующей гипотезе, что математическое ожидание случайной величины X больше математического ожидания случайной величины Y.

*Таблица №1. Экспериментальные данные варианта №4931-12*

| **Массив экспериментальных данных** | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| x | 0,2 | 1,4 | 5,3 | 6,9 | 5,8 | 3,3 | 3,2 | 2,8 | 2,9 | 0,9 |
| y | 0,6 | 1,1 | 2 | 3,4 | 4,5 | 6,1 | 5,2 | 4,7 | 1,3 | 1,3 |

**Расчетные формулы**

1. Оценка математического ожидания:

где – количество элементов массива; – -ый элемент массива.

2. Показатель согласованности гипотезы:

Введем в рассмотрение случайную величину , которая распределена по нормальному закону (как и и ), и, нормируя ее, получим

3. Правило проверки гипотезы, где

В этом случае строится правосторонняя критическая область, чтобы вероятность попадания в нее показателя согласованности в предположении о справедливости нулевой гипотезы была равна :

Для того, чтобы найти критическую точку с помощью функции Лапласа, перепишем выражение в виде:

Следовательно,

Для данной лабораторной работы уровень значимости

Проверяется условие . Если оно выполняется, гипотеза отвергается, в противном случае принимается.

**Результаты работы**

В ходе выполнения данной лабораторной работы была написана программа на языке Python 3.10, решающая задачу в общем виде для четных вариантов – проверяет только гипотезы : математические ожидания равны; - мат. ожидание с.в. больше мат. ожидания с.в. . Считается, что обе случайные величины распределены по нормальному закону. На вход программе подается таблица массивов случайных величин, а также значение уровня значимости и таблицу. Так, для варианта №4931-12 были получены следующие результаты:

Загружаем данные для варианта № 12 ...

Массив данных X: [0.2, 1.4, 5.3, 6.9, 5.8, 3.3, 3.2, 2.8, 2.9, 0.9]

Массив данных Y: [0.6, 1.1, 2.0, 3.4, 4.5, 6.1, 5.2, 4.7, 1.3, 1.3]

Оценка математического ожидания массива X: 3.2700000000000005

Оценка математического ожидания массива Y: 3.02

---- Проверка гипотез ----

Пусть нулевая гипотеза - математические ожидания случайных величин равны

Конкурирующая гипотеза - мат. ожидание с.в. X больше мат. ожидания с.в. Y

Вычисленный показатель согласованности гипотезы u: 0.26769159401796605

Уровень значимости: 0.05

Критическая точка: 1.643

Так как u <= u\_alpha, гипотеза H\_0 принимается - математические ожидания случайных величин равны

**Выводы**

В ходе данной лабораторной работы были получены навыки проверки гипотез равенства математических ожиданий двух случайных величин с конкурирующей гипотезой – математическое ожидание случайных величин X больше математического ожидания случайной величины Y.

**ПРИЛОЖЕНИЕ А. ЛИСТИНГ ПРОГРАММЫ**

Листинг программы доступен по ссылке: <https://github.com/ArtemKinko/SUAI-labs-spring-2023/tree/main/EDP/LABS/LAB-3-Statistic>

Файл lab3-statistic.py:

import math  
  
  
def upload\_matrix(data\_file\_name, task\_file\_name):  
 with open(data\_file\_name) as data\_file:  
 with open(task\_file\_name) as task\_file:  
 task\_num = int(task\_file.readline())  
 print("Загружаем данные для варианта №", task\_num, "...")  
 data = [[], []]  
 ready\_data = [[], []]  
 for \_ in range(task\_num):  
 data[0] = data\_file.readline().rsplit()  
 data[1] = data\_file.readline().rsplit()  
 ready\_data[0] = [float(x.replace(',', '.')) for x in data[0]]  
 ready\_data[1] = [float(x.replace(',', '.')) for x in data[1]]  
 return ready\_data, task\_num  
  
  
def find\_t\_from\_table(table\_file\_name, x):  
 with open(table\_file\_name) as l\_table\_file:  
 line = [-1, -1]  
 while line[0] != x:  
 line = [float(param.replace(',', '.')) for param in l\_table\_file.readline().rsplit()]  
 return line[1]  
  
  
def get\_expected\_value(x\_matrix):  
 return sum(x\_matrix) / len(x\_matrix)  
  
  
def get\_expectation\_dispersion(x\_matrix):  
 exp\_estimate = get\_expected\_value(x\_matrix)  
 summary = 0  
 for x in x\_matrix:  
 summary += (x - exp\_estimate) \*\* 2  
 return summary / (len(x\_matrix) - 1)  
  
  
def get\_u\_value(x\_matrix, y\_matrix):  
 m\_x = get\_expected\_value(x\_matrix)  
 m\_y = get\_expected\_value(y\_matrix)  
 d\_x = get\_expectation\_dispersion(x\_matrix)  
 d\_y = get\_expectation\_dispersion(y\_matrix)  
  
 return (m\_x - m\_y) / math.sqrt(d\_x / len(x\_matrix) + d\_y / len(y\_matrix))  
  
  
def upload\_alpha(alpha\_file\_name):  
 with open(alpha\_file\_name) as alpha\_file:  
 return float(alpha\_file.readline())  
  
  
def solve\_task():  
 data, task = upload\_matrix("data.txt", "task.txt")  
 print("Массив данных X:", data[0])  
 print("Массив данных Y:", data[1])  
 exp\_values = [get\_expected\_value(data[0]), get\_expected\_value(data[1])]  
 print("Оценка математического ожидания массива X:", exp\_values[0])  
 print("Оценка математического ожидания массива Y:", exp\_values[1])  
 print("---- Проверка гипотез ----")  
 print("Пусть нулевая гипотеза - математические ожидания случайных величин равны")  
 print("Конкурирующая гипотеза - мат. ожидание с.в. X больше мат. ожидания с.в. Y")  
 u\_value = get\_u\_value(data[0], data[1])  
 print("Вычисленный показатель согласованности гипотезы u:", u\_value)  
 alpha = upload\_alpha("alpha.txt")  
 print("Уровень значимости:", alpha)  
 t\_function = find\_t\_from\_table("t\_function.txt", 1 - 2 \* alpha)  
 print("Критическая точка:", t\_function)  
 if u\_value > t\_function:  
 print("Так как u > u\_alpha, гипотеза H\_0 отвергается,"  
 "принимается гипотеза H\_1 - мат. ожидание с.в. X больше мат. ожидания с.в. Y")  
 else:  
 print("Так как u <= u\_alpha, гипотеза H\_0 принимается - математические ожидания случайных величин равны")  
  
  
solve\_task()

Файл data.txt:

1,2 2,9 6,6 11,1 23,8 19,7 16,1 11,5 5,2 3  
0,8 2,7 14,6 17,8 10,3 15,5 11,2 9,7 7,8 5,4  
1,7 2,9 4,3 4,3 16,9 18,7 18,2 9,1 5,5 1,3  
0,9 1,8 2,3 2,8 15,1 17,6 14,9 8,7 3,2 2,8  
1,1 1,6 2,6 4,9 7,8 4,1 3,9 2 2,3 0,7  
0,7 1,7 2,1 3,9 5,4 6,9 6,7 5 4,2 2,6  
0,5 1,4 3,3 8,5 5,9 5,1 4,2 4,1 3,4 3  
0,3 0,8 1,3 2,5 3,9 4,1 6,2 4,7 3,4 2,3  
1,1 2,5 3,6 11,1 6,8 6,7 6,1 6,5 5,2 4  
0,9 2,2 3,8 14,1 9,9 5,8 4,9 3,5 2,3 1,6  
0,2 2,2 2,6 6,3 13,8 14,7 9,1 3,5 1,2 0,7  
1,3 2,8 4,5 6,2 9,8 11,7 10,1 9,5 3,2 2  
0,9 1,7 2,6 4,1 6,8 9,7 8,1 7,5 7,9 2,9  
1,3 2,8 4,3 6,5 9,9 10,1 9,2 7,7 6,4 4,6  
1,5 2,2 3 4,9 6,7 7,7 8,4 7,9 7,4 2,3  
1,4 2,6 3,3 5,5 6,9 8,1 9,2 7,7 6,2 3,4  
1,1 1,8 4,5 5,5 7,9 8,1 7,2 4,6 3,4 2,3  
1,2 1 3,5 3,8 11,9 6,4 5,2 3,7 1,6 0,6  
1 1,8 3,3 5,5 11,8 7,3 6,2 5,7 5,4 3,3  
1,4 2,9 3,4 10,5 5,9 5,7 4,2 4,1 3,8 2,6  
0,8 1,4 2,1 3,9 13,9 7,1 6,3 5,7 3,4 2,3  
1,2 1,8 3,3 4,4 6,8 12,1 7,2 4,7 4,3 3,5  
0,2 1,4 5,3 6,9 5,8 3,3 3,2 2,8 2,9 0,9  
0,6 1,1 2 3,4 4,5 6,1 5,2 4,7 1,3 1,3  
5,3 4,8 6,2 9,5 16,7 18,1 17,8 15,9 9,3 6,4  
3,1 5,4 7,9 11,7 17,2 14,1 14 12,7 11,8 4,1  
0,7 1,6 1,7 3,1 3,9 4,8 3,2 2,7 1,7 0,8  
0,4 1,3 2,3 3,8 6,9 7,8 7,1 1,9 1,3 0,1  
0,9 4,7 4,5 5,6 6,7 7,3 7,1 5,4 4,4 3,2  
0,3 0,8 2,6 10,3 5,1 5,7 4,3 3,7 3,1 2,8  
4,4 4,5 6,3 9,5 16,9 19,1 19,2 17,7 11,2 3,4  
5,1 4,8 7,5 15,5 17,9 18,2 15,3 9,6 3,4 2,3  
0,3 0,8 1,3 2,5 6,9 6,1 6,2 5,7 3,4 2,3  
1,1 2,5 2,6 8,1 5,8 6,7 6,1 5,5 3,2 3  
1,3 3,9 7,6 18,1 23,8 19,7 16,1 11,5 7,2 3  
3,8 12,7 14,6 20,8 18,3 18,5 14,2 14,7 7,8 6,4  
2,8 5,8 5,5 5,6 14,7 10,3 8,1 6,4 4,4 3,2  
0,3 0,8 3,6 12,3 15,1 7,7 6,3 4,7 5 2,8  
0,6 1,3 3,5 8,3 5,6 5,4 3,9 4,1 3,6 3  
0,7 0,9 2,3 2,6 4,7 5,8 6,1 5,5 3,9 2,7  
2,1 3,3 3,1 5,2 7,7 10,3 9,2 4,3 4,1 2,5  
1,7 4,4 6,2 8 6,9 4,4 3,8 3,9 3,9 3,7  
2,8 3,6 4,3 5,4 7,2 10,1 9,2 9,7 7,4 3,5  
0,2 1,4 4,4 6,9 14,6 13,2 7,2 4,6 4,9 3,4  
6,3 8,9 9,4 21,2 23,6 19,7 18,3 16,4 13,2 5,1  
3,7 12,8 14,4 20,9 18,6 18,2 14 14,6 7,9 6,2  
0,5 1,6 2,2 4,1 5,3 6,8 6,8 5,1 4,1 2,3  
0,3 1,5 3,1 8,6 6,2 5,6 4,4 4,2 3,4 3,5  
7,9 8,6 9,3 9,2 10,7 13,4 11,3 10,4 8,4 6,2  
6,3 10,2 11,8 12,3 13,1 11,9 10,8 9,9 9,7 4,8  
3,7 3,9 5,2 5,8 6,7 6,8 7,1 6,8 5,9 4,7  
2,8 3,6 3,5 7,2 7,7 10,3 9,7 6,3 4,8 4,8  
5,3 16,9 17,6 22,1 23,8 19,7 20,1 18,5 14,2 5,1  
6,8 12,6 19,5 20,8 18,3 18,5 14,2 14,7 12,8 6,4  
3,7 8,1 10,4 11,9 11,6 12,4 10,3 9,6 4,9 4,2  
6,7 7,6 8,2 9,3 10,4 9,8 8,9 6,7 5,8 5,3  
1,3 1,8 2,3 3,5 6,9 6,1 6,2 5,7 3,4 2,3  
1,2 2,4 2,7 8 5,6 6,8 6,1 4,5 3 3,1  
6,3 8,9 9,4 20,2 20,6 16,7 11,3 9,4 9,2 8,1  
4,5 13,7 13,4 20,9 19,8 18,2 14,8 14,9 10,1 9,2  
11,1 23,8 19,7 16,1 11,5 5,2 3 1,2 2,9 6,6  
11,2 9,7 7,8 5,4 0,8 2,7 14,6 17,8 10,3 15,5  
4,3 16,9 18,7 18,2 9,1 5,5 1,3 1,7 2,9 4,3  
14,9 8,7 3,2 2,8 0,9 1,8 2,3 2,8 15,1 17,6  
4,9 7,8 4,1 3,9 2 2,3 0,7 1,1 1,6 2,6  
6,7 5 4,2 2,6 0,7 1,7 2,1 3,9 5,4 6,9  
8,5 5,9 5,1 4,2 4,1 3,4 3 0,5 1,4 3,3  
6,2 4,7 3,4 2,3 0,3 0,8 1,3 2,5 3,9 4,1  
11,1 6,8 6,7 6,1 6,5 5,2 4 1,1 2,5 3,6  
4,9 3,5 2,3 1,6 0,9 2,2 3,8 14,1 9,9 5,8  
6,3 13,8 14,7 9,1 3,5 1,2 0,7 0,2 2,2 2,6  
10,1 9,5 3,2 2 1,3 2,8 4,5 6,2 9,8 11,7  
4,1 6,8 9,7 8,1 7,5 7,9 2,9 0,9 1,7 2,6  
9,2 7,7 6,4 4,6 1,3 2,8 4,3 6,5 9,9 10,1  
4,9 6,7 7,7 8,4 7,9 7,4 2,3 1,5 2,2 3  
9,2 7,7 6,2 3,4 1,4 2,6 3,3 5,5 6,9 8,1  
5,5 7,9 8,1 7,2 4,6 3,4 2,3 1,1 1,8 4,5  
5,2 3,7 1,6 0,6 1,2 1 3,5 3,8 11,9 6,4  
5,5 11,8 7,3 6,2 5,7 5,4 3,3 1 1,8 3,3  
4,2 4,1 3,8 2,6 1,4 2,9 3,4 10,5 5,9 5,7  
3,9 13,9 7,1 6,3 5,7 3,4 2,3 0,8 1,4 2,1  
7,2 4,7 4,3 3,5 1,2 1,8 3,3 4,4 6,8 12,1  
6,9 5,8 3,3 3,2 2,8 2,9 0,9 0,2 1,4 5,3  
5,2 4,7 1,3 1,3 0,6 1,1 2 3,4 4,5 6,1  
9,5 16,7 18,1 17,8 15,9 9,3 6,4 5,3 4,8 6,2  
14 12,7 11,8 4,1 3,1 5,4 7,9 11,7 17,2 14,1  
3,1 3,9 4,8 3,2 2,7 1,7 0,8 0,7 1,6 1,7  
7,1 1,9 1,3 0,1 0,4 1,3 2,3 3,8 6,9 7,8  
5,6 6,7 7,3 7,1 5,4 4,4 3,2 0,9 4,7 4,5  
4,3 3,7 3,1 2,8 0,3 0,8 2,6 10,3 5,1 5,7  
9,5 16,9 19,1 19,2 17,7 11,2 3,4 4,4 4,5 6,3  
15,3 9,6 3,4 2,3 5,1 4,8 7,5 15,5 17,9 18,2  
2,5 6,9 6,1 6,2 5,7 3,4 2,3 0,3 0,8 1,3  
6,1 5,5 3,2 3 1,1 2,5 2,6 8,1 5,8 6,7  
18,1 23,8 19,7 16,1 11,5 7,2 3 1,3 3,9 7,6  
14,2 14,7 7,8 6,4 3,8 12,7 14,6 20,8 18,3 18,5  
5,6 14,7 10,3 8,1 6,4 4,4 3,2 2,8 5,8 5,5  
6,3 4,7 5 2,8 0,3 0,8 3,6 12,3 15,1 7,7  
8,3 5,6 5,4 3,9 4,1 3,6 3 0,6 1,3 3,5  
6,1 5,5 3,9 2,7 0,7 0,9 2,3 2,6 4,7 5,8  
5,2 7,7 10,3 9,2 4,3 4,1 2,5 2,1 3,3 3,1  
3,8 3,9 3,9 3,7 1,7 4,4 6,2 8 6,9 4,4  
5,4 7,2 10,1 9,2 9,7 7,4 3,5 2,8 3,6 4,3  
7,2 4,6 4,9 3,4 0,2 1,4 4,4 6,9 14,6 13,2  
21,2 23,6 19,7 18,3 16,4 13,2 5,1 6,3 8,9 9,4  
14 14,6 7,9 6,2 3,7 12,8 14,4 20,9 18,6 18,2  
4,1 5,3 6,8 6,8 5,1 4,1 2,3 0,5 1,6 2,2  
4,4 4,2 3,4 3,5 0,3 1,5 3,1 8,6 6,2 5,6  
9,2 10,7 13,4 11,3 10,4 8,4 6,2 7,9 8,6 9,3  
10,8 9,9 9,7 4,8 6,3 10,2 11,8 12,3 13,1 11,9  
5,8 6,7 6,8 7,1 6,8 5,9 4,7 3,7 3,9 5,2  
9,7 6,3 4,8 4,8 2,8 3,6 3,5 7,2 7,7 10,3  
22,1 23,8 19,7 20,1 18,5 14,2 5,1 5,3 16,9 17,6  
14,2 14,7 12,8 6,4 6,8 12,6 19,5 20,8 18,3 18,5  
11,9 11,6 12,4 10,3 9,6 4,9 4,2 3,7 8,1 10,4  
8,9 6,7 5,8 5,3 6,7 7,6 8,2 9,3 10,4 9,8  
3,5 6,9 6,1 6,2 5,7 3,4 2,3 1,3 1,8 2,3  
6,1 4,5 3 3,1 1,2 2,4 2,7 8 5,6 6,8  
20,2 20,6 16,7 11,3 9,4 9,2 8,1 6,3 8,9 9,4  
14,8 14,9 10,1 9,2 4,5 13,7 13,4 20,9 19,8 18,2  
6,7 7,6 8,2 9,3 10,4 9,8 8,9 6,7 5,8 5,3  
3,7 8,1 10,4 11,9 11,6 12,4 10,3 9,6 4,9 4,2  
1,2 2,4 2,7 8 5,6 6,8 6,1 4,5 3 3,1  
1,3 1,8 2,3 3,5 6,9 6,1 6,2 5,7 3,4 2,3  
4,5 13,7 13,4 20,9 19,8 18,2 14,8 14,9 10,1 9,2  
6,3 8,9 9,4 20,2 20,6 16,7 11,3 9,4 9,2 8,1

Файл t\_function.txt:

0,8 1,282  
0,81 1,31  
0,82 1,34  
0,83 1,371  
0,84 1,404  
0,85 1,439  
0,86 1,475  
0,87 1,513  
0,88 1,554  
0,89 1,597  
0,9 1,643  
0,91 1,694  
0,92 1,75  
0,93 1,81  
0,94 1,88  
0,95 1,96  
0,96 2,053  
0,97 2,169  
0,98 2,325  
0,99 2,576  
0,995 2,82  
0,999 3,29

Файл alpha.txt:

0.05

Файл task.txt:

12