

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий  
Кафедра “Прикладная математика”  
Дисциплина “Стохастический анализ”

**Отчет по лабораторной работе №4**

*Вариант 1*

Выполнила:  
студент гр. 16-ПМ  
Горенкова А. В.

Проверила:  
Шерстнева Л.В.

Нижний Новгород  
2019

## Задание

1. Построить линейную регрессионную модель;
2. Проверить на значимость коэффициент регрессии.

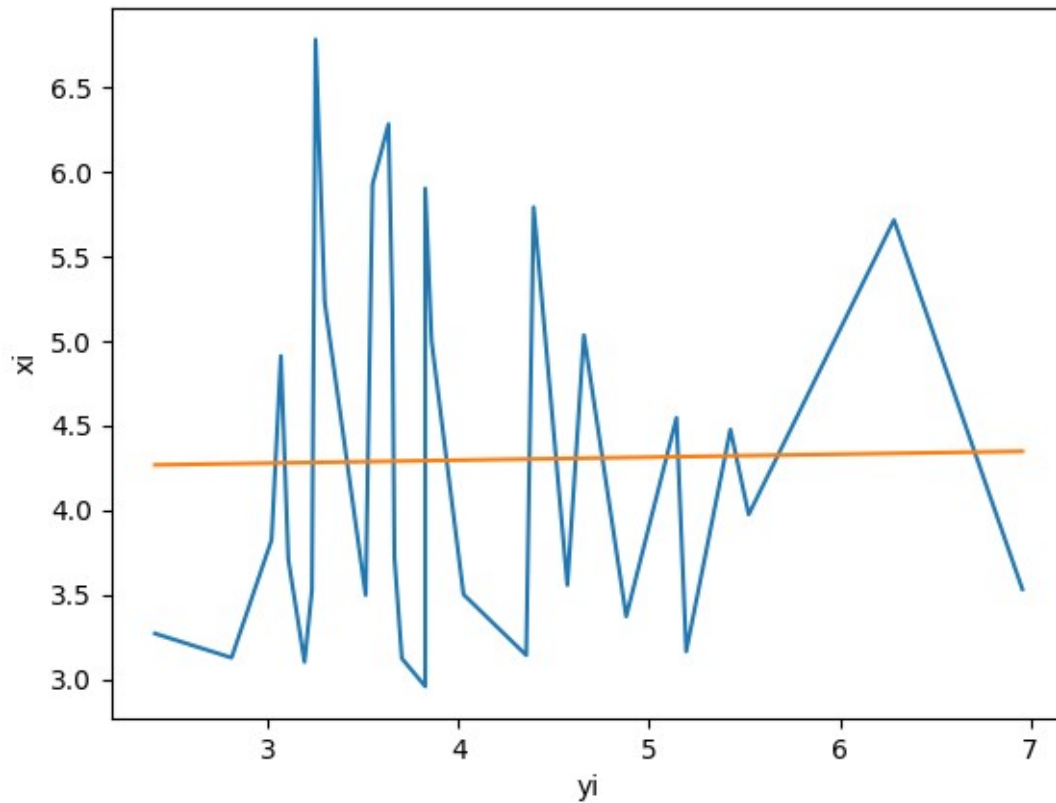
Исходная выборка:

1. 30 (4,570; 3,558), (3,017; 3,825), (3,511; 3,499), (4,393; 5,793),  
(5,522; 3,975), (3,066; 4,913), (4,657; 5,036), (5,143; 4,547), (3,824; 5,904),  
(3,248; 6,784), (3,105; 3, 708), (3,857; 5,002), (3,701; 3,124), (3,662 3,725),  
(5,194; 3,165), (3,190; 3,103), (2,405; 3,271), (2,807; 3,128), (3,824; 2,958),  
(3,631; 6,284), (4,879; 3,372), (6,959; 3,533), (4,354; 3,143), (3,651; 5,197),  
(5,426; 4,478), (3,229; 3,528), (3,547; 5,927), (3,296; 5,231), (4,025; 3,502),  
(6,285; 5,717).

## Результат работы

Коэффициент  $a$  не значимый.

```
witord@witordHP:~/Загрузки$ python3 task4.py  
[4.22600245 0.01762553]  
b = 4.226002450851829 - значимый коэффициент
```



## Исходный код

```
import matplotlib.pyplot as plt
import numpy

def xsr(vv):
    xsr = 0.0
    n = len(vv)
    for i in range(n):
        xsr += vv[i][0]
    return xsr/n

def regF(a, b, x):
    return float(a)*float(x) + float(b)

def znachim(vv, a, b):
    Xsr = xsr(vv)
    n = len(vv)
    Sx = 0
    S = 0
    for i in range(n):
        S += (vv[i][1] - regF(a, b, vv[i][0]))**2
        Sx += (vv[i][0] - Xsr)**2
    S = (S/(n-2)) ** (0.5)
    Sx = (Sx/(n-1)) ** (0.5)
    Sa = S/(Sx*(n-1)**0.5)
    Sb = S * (((1/n) + (Xsr**2)/((n-1) * Sx**2)) ** (0.5))

    tst095 = 2.04841 #28 step
    tst090 = 1.70113
    if abs(a) > tst095 * Sa:
        print("a = ", a, "- значимый коэффициент")
    if abs(b) > tst090 * Sb:
        print("b = ", b, "- значимый коэффициент")

def regressModel(vv):
    n = len(vv)
    xSum = 0.0
    ySum = 0.0
    x2Sum = 0.0
    xySum = 0.0
    for i in range(n):
        # vv = [[x1, y1], [x2, y2]...]
        xSum += vv[i][0]
        ySum += vv[i][1]
```

```
x2Sum += vv[i][0]**2
xySum += vv[i][0]*vv[i][1]
```

```
A = numpy.array([[n, xSum], [xSum, x2Sum]])
B = numpy.array([ySum, xySum])
res = numpy.linalg.solve(A, B)
print(res)
```

```
znachim(vv, res[1], res[0])
```

```
vv = sorted(vv)
vvYLinReg = []
vvX = []
vvY = []
for i in range(n):
    vvYLinReg.append(regF(res[1], res[0], float(vv[i][0])))
    vvX.append(float(vv[i][0]))
    vvY.append(float(vv[i][1]))
```

```
plt.plot(vvX, vvY)
plt.plot(vvX, vvYLinReg)
plt.ylabel('xi')
plt.xlabel('yi')
plt.show()
```

```
vv = [[4.570, 3.558], [3.017, 3.825], [3.511, 3.499], [4.393, 5.793],
[5.522, 3.975], [3.066, 4.913], [4.657, 5.036], [5.143, 4.547], [3.824, 5.904],
[3.248, 6.784], [3.105, 3.708], [3.857, 5.002], [3.701, 3.124], [3.662, 3.725],
[5.194, 3.165], [3.190, 3.103], [2.405, 3.271], [2.807, 3.128], [3.824, 2.958],
[3.631, 6.284], [4.879, 3.372], [6.959, 3.533], [4.354, 3.143], [3.651, 5.197],
[5.426, 4.478], [3.229, 3.528], [3.547, 5.927], [3.296, 5.231], [4.025, 3.502],
[6.285, 5.717]]
```

```
regressModel(vv)
```