

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

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

Отчет по лабораторной работе №5

Вариант 1

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

Проверила:
Шерстнева Л.В.

Нижний Новгород
2019

Задание

1. Построить корреляционное поле. По характеру расположения точек в корреляционном поле и на основе метода средних точек выбрать вид нелинейной регрессии.
2. Вычислить числовые характеристики: среднее у, среднее х, среднеквадратичное отклонение х и у.
3. Найти эмпирическое уравнение регрессий Y на X.
4. Вычислить коэффициент детерминации R^2 .
5. Проверить адекватность уравнения регрессии Y на X.

Результат работы

Корреляционное поле:

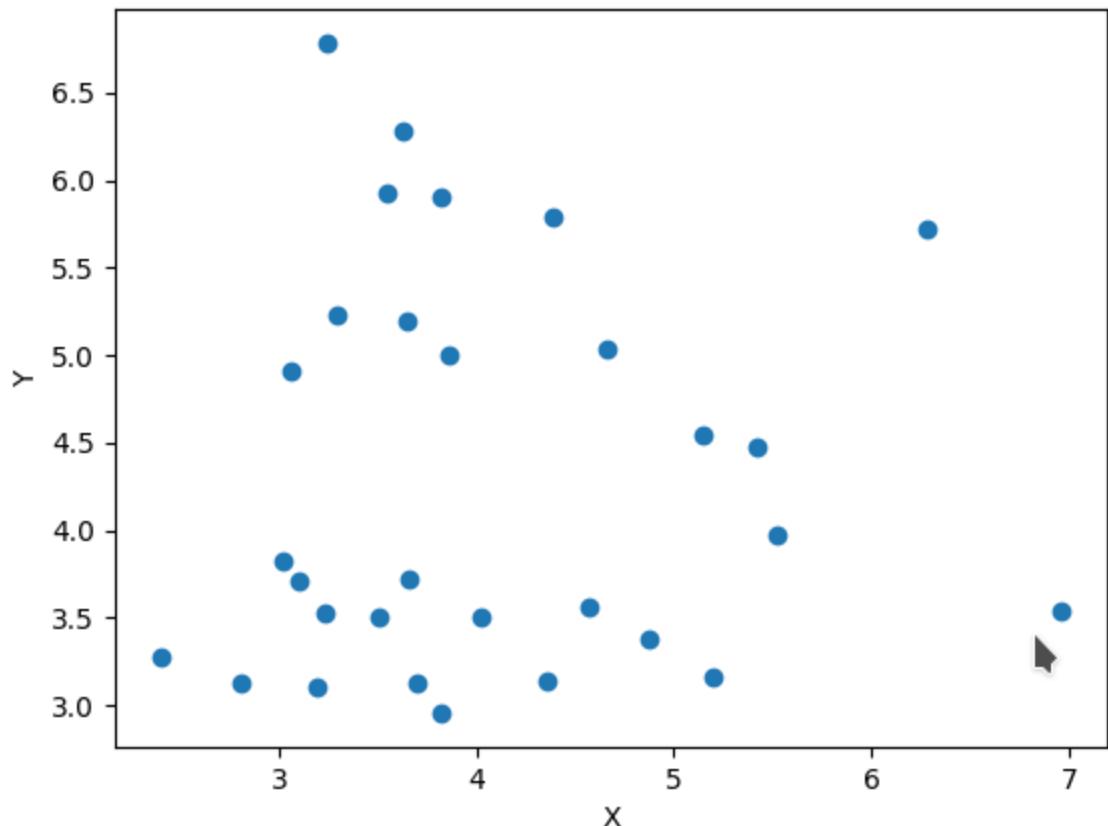
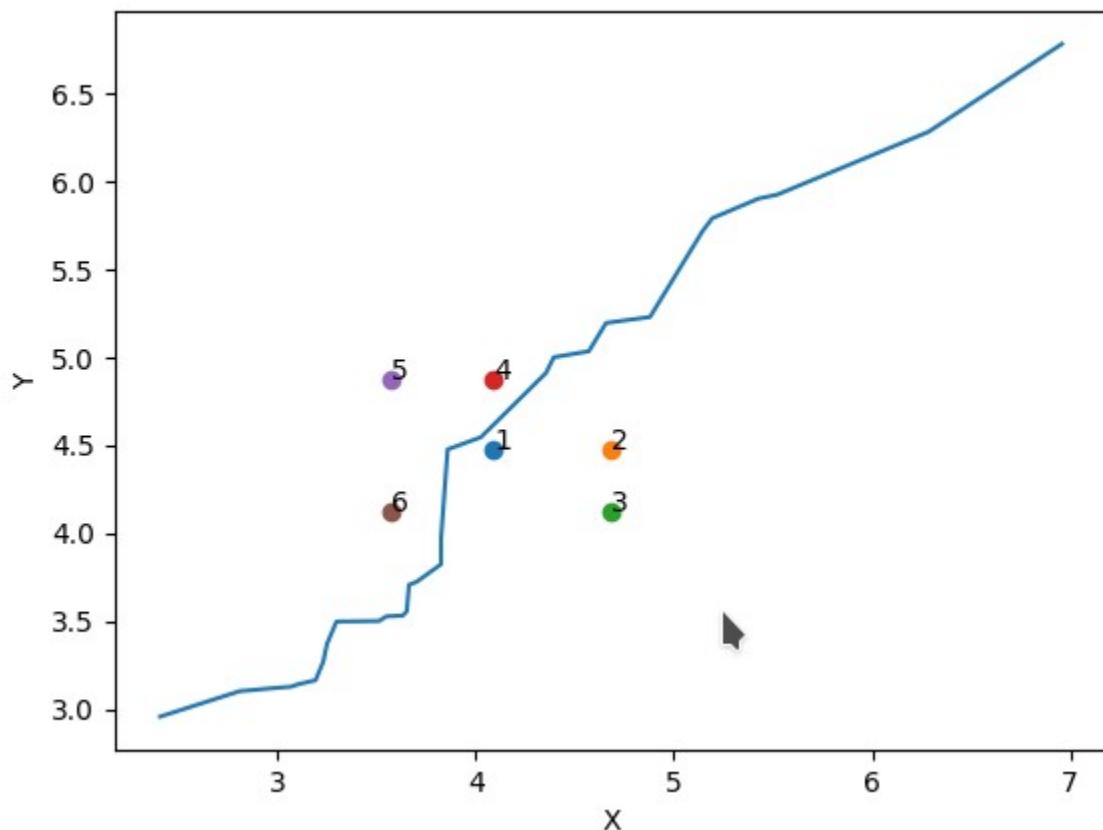


График упорядоченных данных. Наиболее близко к графику средняя точка для уравнения $y = ax^b$.



Среднее X = 4.06593333333334

Среднее Y = 4.297666666666666

Sx = 1.0431531985070819

Sy = 1.1150542089463134

y = 0.06583541124443597 x + 1.3357226370556237 -

эмпирическое уравнение регрессии

R² = 0.8689595263416539 – коэффициент детерминации

Fвыб = 185.67444132568306 > Fкрит=1.7, полученное уравнение регрессии статистически значимо описывает результаты эксперимента.

Исходный код

```
import matplotlib.pyplot as plt
import numpy
import math

def find_Xsr(v):
    Xsr = 0
    n = len(v)
    for i in range(n):
        Xsr += v[i]/n
    return Xsr

def find_Dispers(v, Xsr):
    S2 = 0
    n = len(v)
    for i in range(n):
        S2 += ((v[i] - Xsr)**2)/n
    return S2**(0.5)

def listsum(numList):
    theSum = 0
    for i in numList:
        theSum += math.log(i)
    return theSum

def list2sum(numList):
    theSum = 0
    for i in numList:
        theSum += math.log(i)**2
    return theSum

def listssum(numList1, numList2):
    theSum = 0
    for i in range(len(numList1)):
        theSum += math.log(numList1[i])*math.log(numList2[i])
    return theSum

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]]  
vv = sorted(vv)
```

```
X = []  
Y = []  
n = len(vv) # 40 элементов
```

```
for i in range(n):  
    X.append(vv[i][0])  
    Y.append(vv[i][1])
```

```
nX = len(X)  
nY = len(Y)
```

```
sortX = sorted(X)  
sortY = sorted(Y)
```

```
plt.scatter(X, Y)  
plt.ylabel('Y')  
plt.xlabel('X')  
# plt.show()
```

```
plt.plot(sortX, sortY)  
plt.ylabel('Y')  
plt.xlabel('X')
```

```
# y = a*x**b  
Xcp = (sortX[0]*sortX[n-1])**0.5  
Ycp = (sortY[0]*sortY[n-1])**0.5  
plt.scatter(Xcp, Ycp)  
plt.annotate(1, (Xcp, Ycp))
```

```
# y = ab**x  
Xcp = (sortX[0]+sortX[n-1])/2  
plt.scatter(Xcp, Ycp)  
plt.annotate(2, (Xcp, Ycp))
```

```
# y = 1/(a+bx)  
Ycp = 2*sortY[0]*sortY[n-1]/(sortY[0]+sortY[n-1])  
plt.scatter(Xcp, Ycp)  
plt.annotate(3, (Xcp, Ycp))
```

```

# y = a+b*lg(x)
Xcp = (sortX[0]*sortX[n-1])**0.5
Ycp = (sortY[0]+sortY[n-1])/2
plt.scatter(Xcp,Ycp)
plt.annotate(4, (Xcp, Ycp))

# y = a + b/x
Xcp = 2*sortX[0]*sortX[n-1]/(sortX[0]+sortX[n-1])
plt.scatter(Xcp,Ycp)
plt.annotate(5, (Xcp, Ycp))

# y = ax/(b+x)
Ycp = 2*sortY[0]*sortY[n-1]/(sortY[0]+sortY[n-1])
plt.scatter(Xcp,Ycp)
plt.annotate(6, (Xcp, Ycp))

# plt.show()

Xcp = find_Xsr(X)
print('Среднее X =', Xcp)

Ycp = find_Xsr(Y)
print('Среднее Ycp =', Ycp)

Sx = find_Dispers(X, Xcp)
Sy = find_Dispers(Y, Ycp)

print('Sx =', Sx)
print('Sy =', Sy)

#y = a*x**b -> Y = A+b*X, Y = lny, A = lna, X = lnX
A = [[n, listsum(X)],
      [listsum(X), list2sum(X)]]
B = [listsum(Y), listssum(X, Y)]
res = numpy.linalg.solve(A, B)

a = res[1]
b = res[0]

print('y =', a,'x +', b, ' - эмпирическое уравнение регрессии')

```

```
y = []
for i in range(n):
    y.append(a*math.log(X[i]) + b)
```

```
Xcp = find_Xsr(X)
Ycp = find_Xsr(Y)
```

```
res = 0
for i in range(n):
    res += (Y[i] - y[i])**2
```

```
resSr = 0
for i in range(n):
    resSr += (Y[i] - Ycp)**2
```

```
R2 = 1 - resSr/res
print('R^2 =', R2)
```

```
F = ((R2)*(n-2))/(1-R2)
print('Fвыб =', F, ' > Fкрит=1.7, полученное уравнение регрессии статистически
значимо описывает результаты эксперимента')
```