ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Старший преподаватель |  |  |  | М. Д. Поляк |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3 |
| «Однофакторный регрессионный анализ» |
| по курсу: Математические пакеты программ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4831 |  | 18.12.2020 |  | К.А.Корнющенков |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2020

1. **Задание на лабораторную работу**

На основе заданного массива данных:

* построить уравнение регрессии в виде алгебраического полинома третьей степени;
* проверить адекватность уравнения регрессии;
* проверить значимость коэффициентов регрессии;
* отобразить на графике алгебраический полином с найденными коэффициентами



1. **Код программы**

Lab3.py

from scipy import stats

import numpy

import math

import matplotlib.pyplot

x = numpy.array([-2, -1., 0, 1, 2, 3, 5])

y = numpy.array([14, 17, 10, 4, 6, 19, 101])

arraySize = x.size

keff\_count = 3

alpha = 0.05

x1 = numpy.array([

[1., x[0], x[0]\*\*2, x[0]\*\*3],

[1., x[1], x[1]\*\*2, x[1]\*\*3],

[1., x[2], x[2]\*\*2, x[2]\*\*3],

[1., x[3], x[3]\*\*2, x[3]\*\*3],

[1., x[4], x[4]\*\*2, x[4]\*\*3],

[1., x[5], x[5]\*\*2, x[5]\*\*3],

[1., x[6], x[6]\*\*2, x[6]\*\*3],

])

y1 = y.transpose()

keff = numpy.dot(numpy.dot(numpy.linalg.inv(numpy.dot(x1.transpose(), x1)), x1.transpose()), y1)

print(keff)

print(numpy.polyfit(x,y,keff\_count)[::-1])

ySktushkoy = keff[0] + keff[1] \* x + keff[2]\*(x\*\*2) + keff[3]\*(x\*\*3)

RSS = sum((y-ySktushkoy)\*\*2)

Ymiddle = sum(y) / arraySize

TSS = sum((y - Ymiddle)\*\*2)

sigma\_2 = TSS / (arraySize-1)

sigma1\_2 = RSS / (arraySize-keff\_count-1)

nabludaemoeF = sigma\_2 / sigma1\_2

kriticheskoeF = stats.f.ppf(1-alpha,arraySize-1,arraySize-keff\_count-1)

print(nabludaemoeF)

print(kriticheskoeF)

if nabludaemoeF > kriticheskoeF:

print("H1")

else:

print("Ho")

xtx1 = numpy.linalg.inv(numpy.dot(x1.transpose(), x1))

nabludaemoeT = [ abs(keff[i]) / (numpy.sqrt(RSS/(arraySize-keff\_count-1) \* xtx1[i][i])) for i in range(4)]

kriticheskoeT = stats.t.ppf(1-(alpha/2),arraySize-keff\_count-1)

print("")

print(nabludaemoeT)

print(kriticheskoeT)

ySktushkoy = keff[0] + keff[1] \* x + keff[3]\*(x\*\*3)

keff\_count = 2

RSS = sum((y-ySktushkoy)\*\*2)

Ymiddle = sum(y) / arraySize

TSS = sum((y - Ymiddle)\*\*2)

R2 = 1 - RSS / TSS

sigma\_2 = TSS / (arraySize-1)

sigma1\_2 = RSS / (arraySize-keff\_count-1)

nabludaemoeF = sigma\_2 / sigma1\_2

kriticheskoeF = stats.f.ppf(1-alpha,arraySize-1,arraySize-keff\_count-1)

print("")

print(nabludaemoeF)

print(kriticheskoeF)

if nabludaemoeF > kriticheskoeF:

print("H1")

else:

print("Ho")

xplot = numpy.arange(-3.,6,0.01)

matplotlib.pyplot.plot(x,y,'or')

yplot = keff[0] + keff[1]\*xplot + keff[2]\*(xplot\*\*2) + keff[3]\*(xplot\*\*3)

matplotlib.pyplot.plot(xplot,yplot,'k')

yplot = keff[0] + keff[1]\*xplot + keff[3]\*(xplot\*\*3)

matplotlib.pyplot.plot(xplot,yplot,'b')

matplotlib.pyplot.xlabel(r'$ось X$')

matplotlib.pyplot.ylabel(r'$ось Y$')

frst = "+" if keff[1]>0 else ""

scnd = "+" if keff[2]>0 else ""

frd = "+" if keff[3]>0 else ""

matplotlib.pyplot.title(r'$y = {{{}}} {{{}}} {{{}}}x {{{}}} {{{}}}x^2 {{{}}} {{{}}}x^3 $'.format(

round(keff[0],2),

frst,

round(keff[1],2),

scnd,

round(keff[2],2),

frd,

round(keff[3]),2),

color="b"

)

frst = "+" if keff[1]>0 else ""

frd = "+" if keff[3]>0 else ""

matplotlib.pyplot.suptitle(r'$y = {{{}}} {{{}}} {{{}}}x {{{}}} {{{}}}x^3 $'.format(

round(keff[0],2),

frst,

round(keff[1],2),

frd,

round(keff[3]),2),

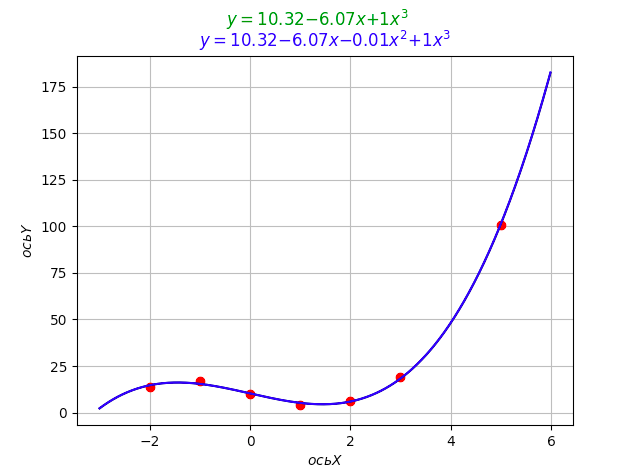
color="g"

)

matplotlib.pyplot.grid()

matplotlib.pyplot.show()

1. **Пример выполнения программы**

****

1. **Вывод**

В ходе выполнения лабораторной работы были получены навыки однофакторного регрессионного анализа с библиотек языка Python.