

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

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

Отчет по лабораторной работе №6

Вариант 1

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

Проверила:
Шерстнева Л.В.

Нижний Новгород
2019

Задание

1. Определить форму связи между факторными и результативным признаками, построив корреляционные поля для каждой пары признаков.
2. Определить тесноту связи между факторами. Произвести отбор факторов, включаемых в модель. Оставить только два наиболее значимых фактора. Записать уравнение множественной регрессии.
3. Проверить адекватность полученного по критерию Фишера

Результат работы

Для $t_{\text{крит}} = 1.29$ значимыми являются X_{11} ($t_{\text{набл}} = 4.015837996433007$), X_{12} ($t_{\text{набл}} = 3.497108216671788$), X_8 ($t_{\text{набл}} = 1.6303307453989526$)
Остальные незначимы: $t_{\text{набл}} = X_{17}: 0.16426544648160085$, $X_6: 1.0617358709687235$

$r_{Y1X8} = 0.2225658694014811$, $r_{Y1X11} = 0.490148537943028$,
 $r_{Y1X12} = 0.4397929717659975$. Оставляем x_{11} и x_{12} для нахождения уравнения регрессии.

Уравнение регрессии:

$$y = 9.925466418825233e-05 x_{11} + 0.003981447458435705 x_{12} + 6.163429618208277$$

$F_{\text{набл}} = 8.238474047069282 > F_{\text{крит}} = 3.19$, а значит с вероятностью 0,95 гипотеза о статистической значимости эмпирических данных принимается, корреляционная модель может быть построена.

Исходный код

```
import numpy

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

def findXIJs(v1, v2):
    Xsr = 0
    n = len(v1)
    for i in range(n):
        Xsr += v1[i]*v2[i]/n
    return Xsr

def findDispers(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 += i
    return theSum

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

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

X6 = [0.4, 0.26, 0.4, 0.5, 0.4, 0.19, 0.25, 0.44, 0.17, 0.39, 0.33, 0.25, 0.32, 0.02, 0.06, 0.15, 0.08, 0.2, 0.2, 0.3, 0.24, 0.1, 0.11, 0.47, 0.53, 0.34, 0.2, 0.24, 0.54, 0.4, 0.2, 0.64, 0.42, 0.27, 0.37, 0.38, 0.35, 0.42, 0.32, 0.33, 0.29, 0.3, 0.56, 0.42, 0.26, 0.16, 0.45, 0.31, 0.08, 0.68, 0.03, 0.02, 0.22]

X8 = [1.23, 1.04, 1.8, 0.43, 0.88, 0.57, 1.72, 1.7, 0.84, 0.6, 0.82, 0.84, 0.67, 1.04, 0.66, 0.86, 0.79, 0.34, 1.6, 1.46, 1.27, 1.58, 0.68, 0.86, 1.98, 0.33, 0.45, 0.74, 0.03, 0.99, 0.24, 0.57, 1.22, 0.68, 1, 0.81, 1.27, 1.14, 1.89, 0.67, 0.96, 0.67, 0.98, 1.16, 0.54, 1.23, 0.78, 1.16, 4.44, 1.06, 2.13, 1.21, 2.2]

X11 = [26006, 23935, 22589, 21220, 7394, 11586, 26609, 7801, 11587, 9475, 10811, 6371, 26761, 4210, 3557, 14148, 9872, 5975, 16662, 9166, 15118, 11429, 6462, 24628, 49727, 11470, 19448, 18963, 9185, 17478, 6265, 8810, 17659, 10342, 8901, 8402, 32625, 31160, 46461, 13833, 6391, 11115, 6555, 11085, 9484, 3967, 15283, 20874, 19418, 3351, 6338, 9756, 11705]

X12 = [167.69, 186.1, 220.45, 169.3, 39.53, 40.41, 102.96, 37.02, 45.74, 40.07, 45.44, 41.08, 136.14, 42.39, 37.39, 101.78, 47.55, 32.61, 103.25, 38.95, 81.32, 67.26, 59.92, 107.34, 512.6, 53.81, 80.83, 59.42, 36.96, 91.43, 17.16, 27.29, 184.33, 58.42, 59.4, 49.63, 391.27, 258.62, 75.66, 123.68, 37.21, 53.37, 32.87, 45.63, 48.41, 13.58, 63.99, 104.55, 222.11, 25.76, 29.52, 41.99, 78.11]

X17 = [17.72, 18.39, 26.46, 22.37, 28.13, 17.55, 21.92, 19.52, 23.99, 21.76, 25.68, 18.13, 25.74, 21.21, 22.97, 16.38, 13.21, 14.48, 13.38, 13.69, 16.66, 15.06, 20.09, 15.98, 18.27, 14.42, 22.76, 15.41, 19.35, 16.83, 30.53, 17.98, 22.09, 18.29, 26.05, 26.2, 17.26, 18.83, 19.7, 16.87, 14.63, 22.17, 22.62, 26.44, 22.26, 19.13, 18.28, 28.23, 12.39, 11.64, 8.62, 20.1, 19.41]

Y1 = [9.26, 9.38, 12.11, 10.81, 9.35, 9.87, 8.17, 9.12, 5.88, 6.3, 6.2, 5.49, 6.5, 6.61, 4.32, 7.37, 7.02, 8.25, 8.15, 8.72, 6.64, 8.1, 5.52, 9.37, 13.17, 6.67, 6.68, 5.22, 10.02, 8.16, 3.78, 6.48, 10.44, 7.65, 8.77, 7, 11.06, 9.02, 13.28, 9.27, 6.7, 6.69, 9.42, 7.24, 5.39, 5.61, 5.59, 6.57, 6.54, 4.23, 5.22, 18, 11.03]

k=5 #количество факторов

n = len(X11)

X11cp = findXsr(X11)

X12cp = findXsr(X12)

X6cp = findXsr(X6)

X8cp = findXsr(X8)

X17cp = findXsr(X17)

Y1cp = findXsr(Y1)

#Среднеквадратическое отклонение

SX11 = findDispers(X11, X11cp)

SX12 = findDispers(X12, X12cp)

SX6 = findDispers(X6, X6cp)

SX8 = findDispers(X8, X8cp)

```

SX17 = findDispers(X17, X17cp)
SY1 = findDispers(Y1, Y1cp)
#Среднее произведений
X11X12cp = findXIJsр(X11, X12)
X11X6cp = findXIJsр(X11, X6)
X11X8cp = findXIJsр(X11, X8)
X11X17cp = findXIJsр(X11, X17)
X12X6cp = findXIJsр(X12, X6)
X12X8cp = findXIJsр(X12, X8)
X12X17cp = findXIJsр(X12, X17)
X6X8cp = findXIJsр(X6, X8)
X6X17cp = findXIJsр(X6, X17)
X8X17cp = findXIJsр(X8, X17)
Y1X11cp = findXIJsр(Y1, X11)
Y1X12cp = findXIJsр(Y1, X12)
Y1X6cp = findXIJsр(Y1, X6)
Y1X8cp = findXIJsр(Y1, X8)
Y1X17cp = findXIJsр(Y1, X17)
#коэффициенты парной корреляции
rX11X12 = (X11X12cp - X11cp*X12cp)/(SX11*SX12)
rX11X6 = (X11X6cp - X11cp*X6cp)/(SX11*SX6)
rX11X8 = (X11X8cp - X11cp*X8cp)/(SX11*SX8)
rX11X17 = (X11X17cp - X11cp*X17cp)/(SX11*SX17)
rX12X6 = (X12X6cp - X12cp*X6cp)/(SX12*SX6)
rX12X8 = (X12X8cp - X12cp*X8cp)/(SX12*SX8)
rX12X17 = (X12X17cp - X12cp*X17cp)/(SX12*SX17)
rX6X8 = (X6X8cp - X6cp*X8cp)/(SX6*SX8)
rX6X17 = (X6X17cp - X6cp*X17cp)/(SX6*SX17)
rX8X17 = (X8X17cp - X8cp*X17cp)/(SX8*SX17)
rY1X11 = (Y1X11cp - Y1cp*X11cp)/(SY1*SX11)
rY1X12 = (Y1X12cp - Y1cp*X12cp)/(SY1*SX12)
rY1X6 = (Y1X6cp - Y1cp*X6cp)/(SY1*SX6)
rY1X8 = (Y1X8cp - Y1cp*X8cp)/(SY1*SX8)
rY1X17 = (Y1X17cp - Y1cp*X17cp)/(SY1*SX17)

```

```

tnYX11 = ((n - 2)* rY1X11**2 / (1 - rY1X11**2))**(0.5)
tnYX12 = ((n - 2)* rY1X12**2 / (1 - rY1X12**2))**(0.5)
tnYX17 = ((n - 2)* rY1X17**2 / (1 - rY1X17**2))**(0.5)
tnYX8 = ((n - 2)* rY1X8**2 / (1 - rY1X8**2))**(0.5)
tnYX6 = ((n - 2)* rY1X6**2 / (1 - rY1X6**2))**(0.5)
tkr = 1.29

```

```
print("Для ткрит = ", tkr, "значимыми являются X11 (тнабл =", tnYX11, "), X12 (тнабл =", tnYX12, "), X8 (тнабл =", tnYX8, ")")
print("Остальные незначимы: тнабл = ", tnYX17, tnYX6)
```

```
print(rY1X8, rY1X11, rY1X12)
```

```
A = numpy.array([[listsum(X11), listsum(X12), n],
                 [listssum(X11, X12), list2sum(X12), listsum(X12)],
                 [list2sum(X11), listssum(X11, X12), listsum(X11)]])
B = numpy.array([listsum(Y1), listssum(Y1, X12), listssum(Y1, X11)])
res = numpy.linalg.solve(A, B)
print("y = ", res[0], "x11 + ", res[1], "x12 + ", res[2])
```

```
RYX11X12 = ((rY1X11**2 + rY1X12**2 - 2*rY1X11*rY1X12*rX11X12)/(1-
rX11X12**2))**(0.5)
```

```
# Проверка адекватности модели множественной линейной корреляции
```

```
# 2 - число факторных признаков.
```

```
Fnabl = ((n-2-1)*(RYX11X12**2))/(2*(1-RYX11X12**2))
```

```
Fkrit = 3.19 # 0.05 и 50
```

```
print("Фнабл =", Fnabl, "Фкрит =", Fkrit)
```

```
if Fnabl >= Fkrit:
```

```
    print('с вероятностью 0,95 гипотеза о статистической значимости эмпирических
данных принимается, корреляционная модель может быть построена. \n')
```

```
else:
```

```
    print('гипотеза отвергается\n')
```