

Федеральное государственное бюджетное образовательное учреждение высшего образования

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра теоретической и прикладной информатики Лабораторная работа №3

по дисциплине «Статистические методы анализа данных»

Студенты ИВАНОВ ВЛАДИСЛАВ (92)

ОБЕРШТ ЕЛЕНА (93)

Вариант 5

Преподаватель ПОПОВ АЛЕКСАНДР АЛЕКСАНДРОВИЧ

Новосибирск, 2022

# Постановка задачи

1. Изменить модель регрессии, добавив в нее дополнительный регрессор, ранее не вошедший в состав модели, порождающей данные. Не генерируя новых данных, найти точечные оценки всех параметров расширенной модели. В дальнейшем при рассмотрении этой расширенной модели анализе должно быть показано, что параметр при дополнительном регрессоре незначим.

2. Построить доверительные интервалы для каждого параметра модели регрессии.

3. Проверить гипотезу о незначимости каждого параметра модели.

4. Проверить гипотезу о незначимости самой регрессии.

5. Рассчитать прогнозные значения для математического ожидания f x для всего интервала действия одного из факторов, зафиксировав значения других факторов на границе функции отклика или в центре области их определения.

6. По полученным в п. 5 прогнозным значениям построить графики прогнозных значений и доверительной полосы для математического ожидания функции отклика и для самого отклика.

7. Заново смоделировать исходные данные (см. лаб. работу № 1), увеличив мощность случайной помехи до 50…70 % от мощности полезного сигнала, и провести оценку параметров. Повторить пункты 3, 4 с новыми данными.

# Описание объекта

## Добавим новый регрессор x1^3:

## *θ* = (1*,* 2*,* 0*.*01*,* −0*.*5, 0)*T*

*x*1 ∈ {−1*,* −0*.*5*,* 0*.*5*,* 1}

*x*2 ∈ {−1*,* −0*.*5*,* 0*,* 0*.*5*,* 1}

f(x1, x2)=1+x1+x1^2+x2^2+x1^3

*u* = *η*(*x, θ*) = *θT f* (*x*1*, x*2) = *θ*0 + *θ*1*x*1 + *θ*2*x*2 + *θ*3*x*2

1 2

= 1 + 2*x*1 + 0*.*01*x*2 − 0*.*5*x*2

1

2

Точечные оценки предыдущей модели:

Θ**~** = (1.03, 1.95, 0.05 −0.41)T

θ = (1, 2, 0.01, −0.5)T

*FT* = 2.01

*F* = = 0*.*95 *< FT*

Модель адекватная.

Точечные оценки измененной модели:

Θ**~** = (0.7 2.13 0.18 −0.13 -0.18)T

θ = (1, 2, 0.01, −0.5, 0)T

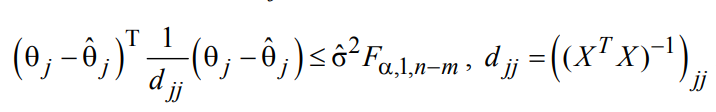
*FT* = 2.01

*F* = = 0*.*9 *< FT*

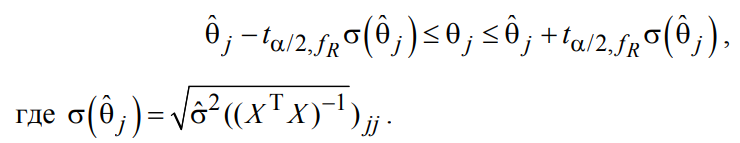
Модель адекватная.

**Доверительные интервалы для параметров регрессии**

Из формулы

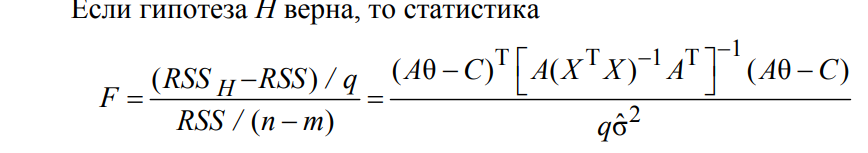


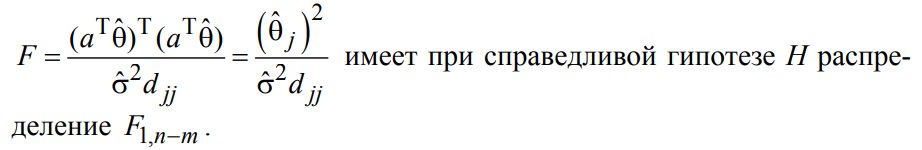
Получаем двустороннее неравенство:

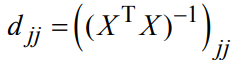
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| j | Оценка θ | Получ оценка Θ**~** | Мин грань | Макс грань |
| 1 | 1 | 0.79556379 | -1.27190645 | 2.86303404 |
| 2 | 2 | 1.99895313 | -1.44472888 | 5.44263515 |
| 3 | 0.01 | 0.12630703 | -2.28994011 | 2.54255417 |
| 4 | −0.5 | 0.19179642 | -2.3577722 | 1.97417936 |
| 5 | 0 | 0.00941988 | -3.8110023 | 3.82984205 |

**Гипотеза о незначимости каждого параметра регрессии**





где aT – вектор строка с нулями всюду, кроме j-го элемента, на нем стоит 1.

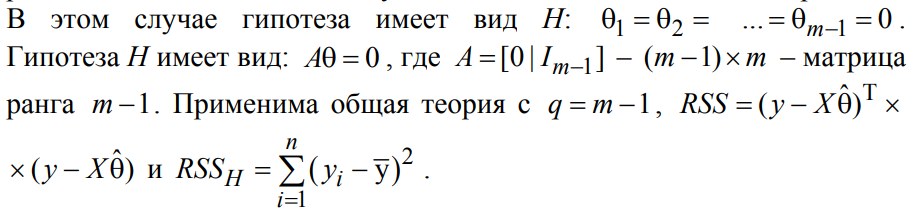


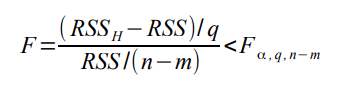
F = 4.54

|  |  |  |  |
| --- | --- | --- | --- |
| j | F | F ? Fф | Гипотеза о незначимости |
| 1 | 21.20186965 | > | Отвергается |
| 2 | 71.13650934 | > | Отвергается |
| 3 | 1.03399348 | < | Не отвергается |
| 4 | 0.73429135 | < | Не отвергается |
| 5 | 0.42909029 | < | Не отвергается |

Была отвергнута значимость параметров регрессии с коэффициентами близкими или равными 0, то есть имеющим наименьший вес в уравнении.

**Гипотеза о незначимости уравнения регрессии**



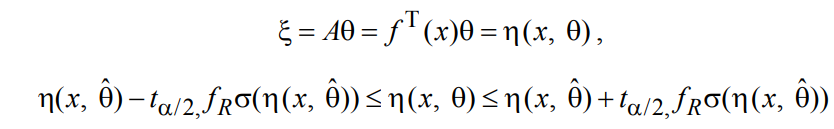


F 0.05, 4, 15  = 3.06

F = 163.76127004

Fфиш < F -> Гипотеза о незначимости уравнения регрессии отвергается.

**Доверительное оценивание мат ожидания**







Зафиксируем x1 = 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1 | X2 | Лев граница |  | Прав граница |
| 1 | -1 | 2.3803937276255773 | 2.55841323 | 2.6396062723744222 |
| 1 | -0.8 | 2.5771998194775825 | 2.74335703 | 2.8028001805224165 |
| 1 | -0.6 | 2.7171998194775826 | 2.88720221 | 2.9428001805224167 |
| 1 | -0.4 | 2.8099111112983404 | 2.98994877 | 3.050088888701659 |
| 1 | -0.2 | 2.8630401573200834 | 3.0515967 | 3.116959842679916 |
| 1 | 0 | 2.8803937276255773 | 3.07214601 | 3.1396062723744222 |
| 1 | 0.2 | 2.8630401573200834 | 3.0515967 | 3.116959842679916 |
| 1 | 0.4 | 2.8099111112983404 | 2.98994877 | 3.050088888701659 |
| 1 | 0.6 | 2.7171998194775826 | 2.88720221 | 2.9428001805224167 |
| 1 | 0.8 | 2.5771998194775825 | 2.74335703 | 2.8028001805224165 |
| 1 | 1 | 2.3803937276255773 | 2.55841323 | 2.6396062723744222 |

График для мат ожидания

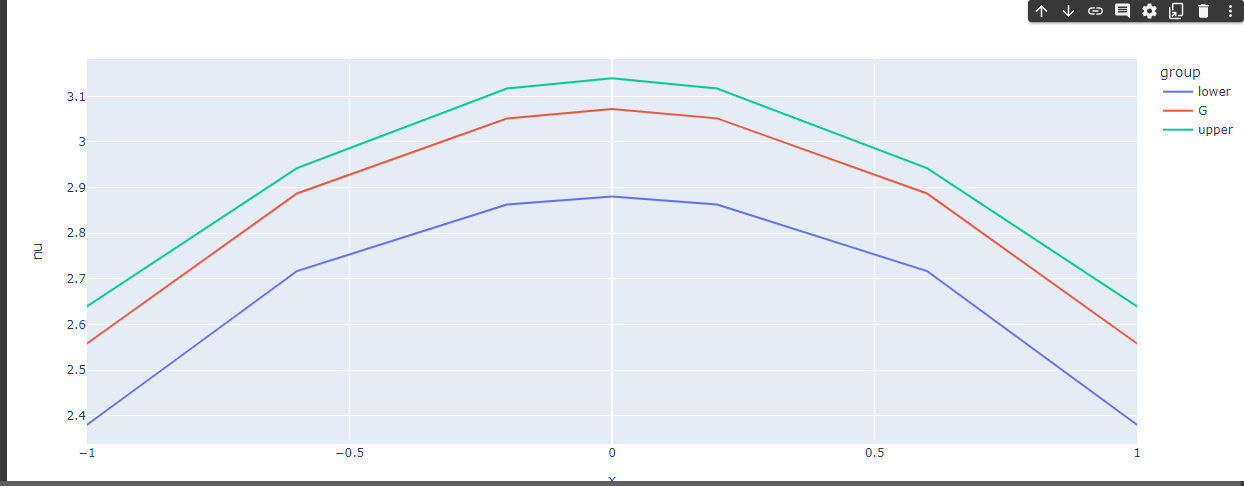
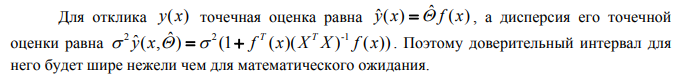
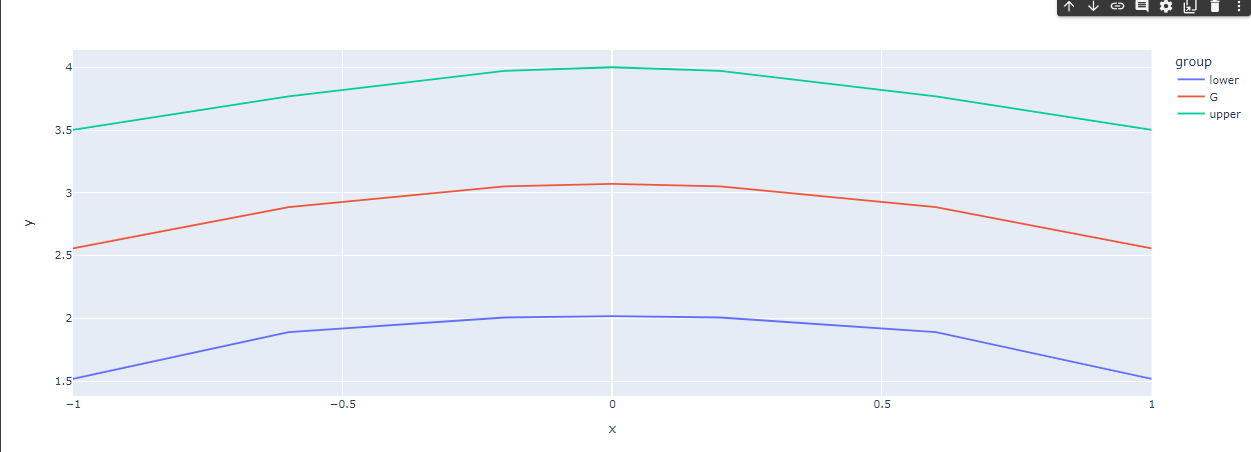


График для отклика





Оба графика симметричны относительно оси OX, так как для функции f(x1, x2) при фиксированном x1 зависимость от x2 квадратичная.

Увеличиваем мощность помехи до 70%.

sigma: 1.369069927763101

New sigma^2: 2.8487040841496585

Theta = [1, 2, 0.01, −0.5, 0]T

New theta: [1.20921204 0.70848653 0.22766134 -0.92599909 0.99100456]T

F = 1.5198337207884798

FT = 4.543077165266973

F < FT: модель адекватная

**Гипотеза о незначимости каждого параметра регрессии**

F Фишера = 4.543077165266973

|  |  |  |  |
| --- | --- | --- | --- |
| j | F | F ? Fф | Гипотеза о незначимости |
| 1 | 1.97176107 | < | Не отвергается |
| 2 | 0.24397483 | < | Не отвергается |
| 3 | 0.05117099 | < | Не отвергается |
| 4 | 1.05351767 | < | Не отвергается |
| 5 | 0.38784347 | < | Не отвергается |

**Гипотеза о незначимости уравнения регрессии**

F фиш = 4. 543077165266973

F = 10.5841074

Fфиш < F -> Гипотеза о незначимости уравнения регрессии отвергается.

# Код программы

import  pandas as pd

import  numpy as np

import  random

import  scipy.stats

from matplotlib import pyplot as plt

import plotly.express as px

random.seed(42)

def u(x1,x2):

    return 1 + 2\*x1 + 0.01\*x1\*\*2 - 0.5\*x2\*\*2 + 0\*x1\*x1\*x1

def f(x1, x2):

  return 1 + x1 + x1\*\*2 + x2\*\*2 + x1\*\*3

tetaj = np.array([1, 2, 0.01, -0.5, 0])    #basic theta

tetaj2 = [1, 2, 0.01, -0.5, 0]

#print("check  tetaj as list : ", tetaj2)

#print("check tetaj as array : ", tetaj)

x1j, x2j, uj = [], [], []

for x1 in [-1,-0.5,0.5,1]:

    for x2 in [-1,-0.5,0,0.5,1]:

        x1j.append(x1)

        x2j.append(x2)

        uj.append(u(x1,x2))

n = 20

m = 5

q = m - 1

p = 0.05

u\_avg = np.full((20,), np.mean(uj))

omega\_sq = (np.dot((uj-u\_avg),(uj-u\_avg)))/(n-1)

sigma = np.sqrt(p\*omega\_sq)                              #

ej = np.random.normal(0, sigma, n)

yj = uj + ej

df = pd.DataFrame(list(zip(x1j, x2j, uj, ej, yj)), columns=['x1', 'x2','u', 'e', 'y'])

df.index += 1

x1 = np.ones(n)                       #?????????

x2 = np.reshape(x1j, (n, ))

x3 = pow(np.reshape(x1j, (n, )), 2)

x4 = pow(np.reshape(x2j, (n, )), 2)

x5 = pow(np.reshape(x1j, (n, )), 3)

X = np.array([x1, x2, x3, x4, x5]).T

#print("check X reshaped : ", X)

y = np.reshape(yj, (n, 1))

#print("check y from X : ", y)

theta = np.dot(np.linalg.inv(np.dot(X.T, X)), np.dot(X.T, y))          #theta new recounted

y\_hat = np.dot(X, theta)

e\_tilde = y - y\_hat

sigma\_sq\_new = np.dot(e\_tilde.T, e\_tilde) / (n - theta.shape[0])    #

F = sigma\_sq\_new[0][0] / sigma\*\*2

FT = scipy.stats.f.ppf(q=1-0.7, dfn=1, dfd=15)  #!! need 15 insted 16

print("check 1st F = ", FT)

F\_scipy = scipy.stats.f.cdf(FT, dfn=1, dfd=15)

#print('TETA')

i=0

for par in theta:

  teta1 = par - FT \* pow(theta \* np.linalg.inv(  np.dot(X.T, X)  )[i][i], 0.5)

  teta2 = par + FT \* pow(theta \* np.linalg.inv(  np.dot(X.T, X)  )[i][i], 0.5)

  print('teta1 = ', teta1[0], 'teta2 = ', teta2[0])

  i = i+1

#print()

#print("check!  theta[j]: ", theta)

print('незначимость параметров регрессии')

j = 0

for par in theta:

  Ff = pow(theta[j], 2) / (sigma\_sq\_new[0][0] \* np.linalg.inv(np.dot(X.T, X))[j][j])

  j = j + 1

  print('F = ', Ff)

rssh = 0

for w in range(n):

  rssh = rssh + pow((y[w] - y\_hat), 2)

#print('rssh = ', rssh)

rss = np.dot((y - np.dot(X, theta)).T, (y - np.dot(X, theta)))

FfF = ((rssh - rss) / q) / (rss / (n-m))

print('незначимость регресии: ', FfF)

#print('u\_avg:', u\_avg[0])

#print('omega\_sq:', omega\_sq)

print('sigma:', sigma)

print('New theta:', theta.T[0])

print('New sigma^2:', sigma\_sq\_new[0][0])

print('F =', F)

#print('F (SciPy) =', F\_scipy)

print('FT =', FT)

print('F < FT:', F < FT)

#print('ch3ck: FT 15, FT 16')

#print(scipy.stats.f.ppf(q=1-0.7, dfn=1, dfd=15))

#print(scipy.stats.f.ppf(q=1-0.7, dfn=1, dfd=16))

#1

print('1 мат ожидание функции отклика')

X1 = x1[3]

X2 = x2[0]

print("check F-stat = ", scipy.stats.f.ppf(q=1-0.7, dfn=1, dfd=15))

t = pow(scipy.stats.f.ppf(q=1-0.7, dfn=1, dfd=15), 0.5)

print()

print('nu left    -    nu    -    nu right')

while (X2 <= 1):

  func = np.array([1, X1, X1\*\*2, X2\*\*2, X1\*\*3])

  #print('test: func = ', func)

  #sigmaa = sigma \* pow( np.dot( np.dot(func.T, np.linalg.inv(np.dot(X.T, X)) ), func), 0.5)

  sigmaa = sigma\_sq\_new[0][0] \* pow( np.dot( np.dot(func.T, np.linalg.inv(np.dot(X.T, X)) ), func), 0.5) #correct

  #print('check  sigmaa for мат ожидание ', sigmaa)

  nu = np.dot(func.T, tetaj2) # was theta

  nu\_min = nu - t \* sigmaa

  nu\_max = nu + t \* sigmaa

  nu\_orig = np.dot(func.T, theta)  #!!?(

  print(X1, ' ', X2, ':')

  X2 = X2 + 0.2

  print(nu\_min, '  ', nu\_orig, '  ', nu\_max)

#scipy.stats.f.ppf(q=1-0.05, dfn=1, dfd=15) FT для

df1 = pd.DataFrame({'x': [-1, -0.6, -0.2, 0, 0.2, 0.6, 1, -1, -0.6, -0.2, 0, 0.2, 0.6, 1, -1, -0.6, -0.2, 0, 0.2, 0.6, 1],

                   'nu': [2.3803937276255773, 2.7171998194775826, 2.8630401573200834, 2.8803937276255773, 2.8630401573200834, 2.7171998194775826, 2.3803937276255773,

                          2.55841323, 2.88720221, 3.0515967, 3.07214601, 3.0515967, 2.88720221, 2.55841323,

                          2.6396062723744222, 2.9428001805224167, 3.116959842679916, 3.1396062723744222, 3.116959842679916, 2.9428001805224167, 2.6396062723744222],

                   'group': ['lower', 'lower', 'lower', 'lower', 'lower', 'lower', 'lower', 'G', 'G', 'G', 'G', 'G', 'G', 'G', 'upper', 'upper', 'upper', 'upper',

                             'upper', 'upper', 'upper']})

fig = px.line(df1, x = 'x', y = 'nu', color = 'group')

fig.show()

#2

print('2 мат ожидание отклика')

X1 = x1[3]

X2 = x2[0]

t = pow(scipy.stats.f.ppf(q=1-0.7, dfn=1, dfd=15), 0.5)

print('y left    -    y   -    y right')

while (X2 <= 1):

  func = np.array([1, X1, X1\*\*2, X2\*\*2, X1\*\*3])

  sigmaa = sigma \* (1 +  np.dot( np.dot(func.T, np.linalg.inv(np.dot(X.T, X)) ), func))

  #print('sigmaa for отклик ', sigmaa)

  y\_ = np.dot(tetaj, func)

  y\_min = y\_ - t \* sigmaa

  y\_max = y\_ + t \* sigmaa

  y\_orig = np.dot(func.T, theta)

  print(X1, ' ', X2, ':')

  X2 = X2 + 0.2

  print(y\_min, '  ', y\_orig, '  ', y\_max)

df2 = pd.DataFrame({'x': [-1, -0.6, -0.2, 0, 0.2, 0.6, 1, -1, -0.6, -0.2, 0, 0.2, 0.6, 1, -1, -0.6, -0.2, 0, 0.2, 0.6, 1],

                   'y': [1.5184179951682837, 1.8897574320027166, 2.0069745679740225, 2.0184179951682837, 2.0069745679740225, 1.8897574320027166, 1.5184179951682837,

                         2.55841323, 2.88720221, 3.0515967, 3.07214601, 3.0515967, 2.88720221,  2.55841323,

                         3.501582004831716, 3.7702425679972826, 3.973025432025977, 4.0015820048317154, 3.973025432025977, 3.7702425679972826, 3.501582004831716],

                   'group': ['lower', 'lower', 'lower', 'lower', 'lower', 'lower', 'lower', 'G', 'G', 'G', 'G', 'G', 'G', 'G',

                             'upper', 'upper', 'upper', 'upper', 'upper', 'upper', 'upper']})

fig = px.line(df2, x = 'x', y = 'y', color = 'group')

fig.show()

# 