**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по лабораторной работе №5**

**по дисциплине «Статистические методы обработки**

**экспериментальных данных»**

Тема: Элементы регрессионного анализа.

Выборочные прямые среднеквадратической регрессии.

Корреляционное отношение.

|  |  |  |
| --- | --- | --- |
| Студент гр. 6381 |  | Фиалковский М.С. |
| Преподаватель |  | Середа В.И. |

Санкт-Петербург

2020

**Цель работы.**

Ознакомление с основными положениями метода наименьших квадратов (МНК), со статистическими свойствами МНК оценок, с понятием функции регрессии и роли МНК в регрессионном анализе, с корреляционным отношением, как мерой тесноты произвольной (в том числе и линейной) корреляционной связи.

**Постановка задачи.**

Для заданной двумерной выборки:

1. Построить уравнения выборочных прямых среднеквадратической регрессии. Вычислить оценки остаточной дисперсии. Полученные функции регрессии отобразить графически (на одном графике).
2. Найти значение выборочного корреляционного отношения.
3. Построить уравнения выборочных кривых для параболической среднеквадратической регрессии. Полученные функции регрессии отобразить графически (на одном графике).

Полученные результаты содержательно проинтерпретировать.

**Ход работы.**

*Уравнения выборочных прямых среднеквадратической регрессии.*

Используя ранее найденные оценки математических ожиданий, среднеквадратических отклонений и коэффициента корреляции Пирсона, построим уравнения выборочных прямых среднеквадратической регрессии с помощью следующих формул:

Получим:

Вычислим оценку остаточной дисперсии X относительно Y:

Вычислим оценку остаточной дисперсии Y относительно X:

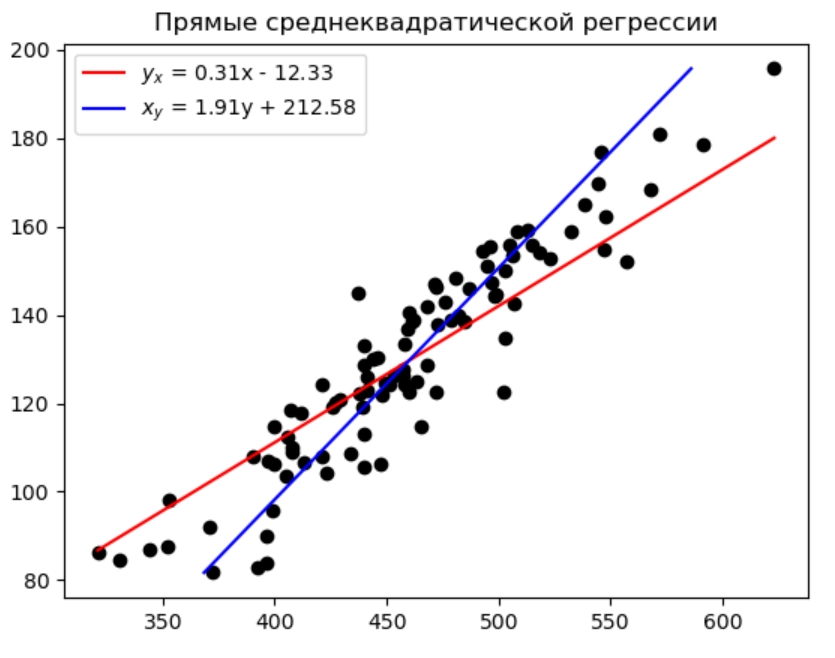
Полученные прямые изобразим на рис.1.

Рисунок 1 – Графическое представление

Рассмотрим корреляционную таблицу:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |  |

Найдем выборочное корреляционное отношение к :

;

;

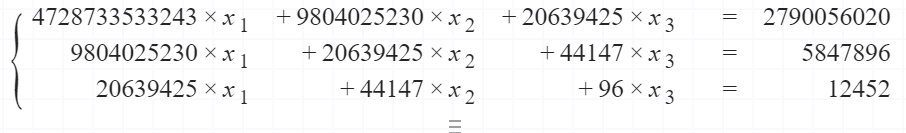
Теперь найдем корреляционное отношение к :

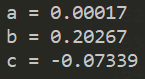
;

;

Получили, что и равны между собой и больше . На основании этого факта можно сделать вывод, что и в выборочных данные связаны корреляционной зависимостью, но характер этой зависимости определить пока не удаётся.

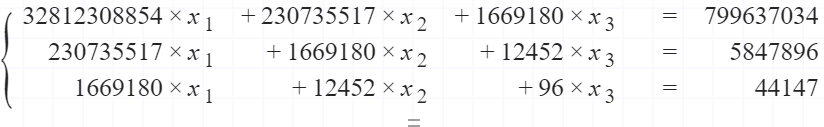
Теперь, используя корреляционную таблицу из предыдущего пункта, составим систему линейных уравнений третьего порядка для нахождения коэффициентов и в уравнениях и

Вычислим коэффициенты для СЛУ и найдём её решение:

Коэффициенты уравнения параболической регрессии на :

Тогда уравнение параболической регрессии на имеет вид:

Аналогично найдем и решим уравнение параболической регрессии на :

Решим систему уравнений:

Тогда уравнение параболической регрессии к имеет вид:

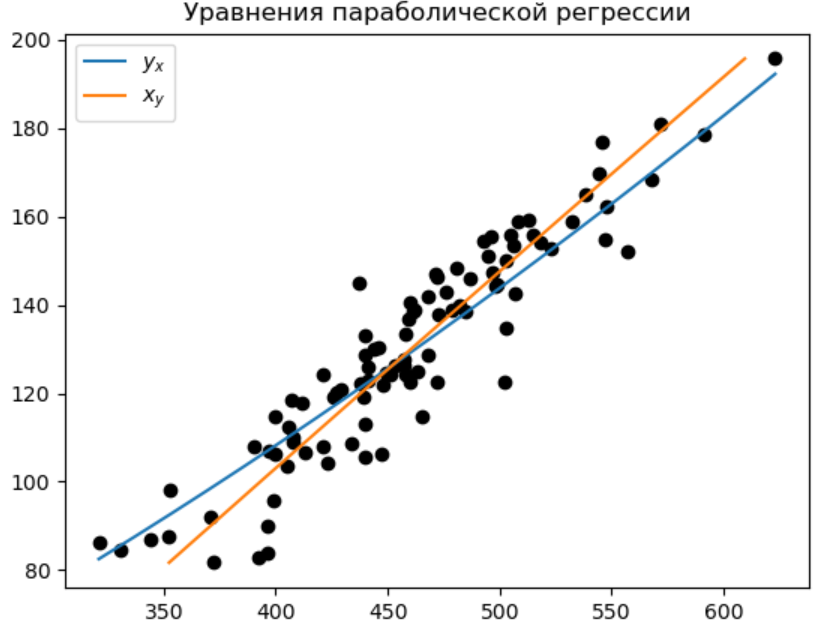
 Построим на рис. 2 графическое представление получившихся уравнений:

Рисунок 2 – Параболические уравнения

**Выводы.**

В ходе выполнения работы были найдены и визуализированы в виде графиков уравнения выборочных прямых среднеквадратической регрессии.

Далее были получены оценки корреляционного отношения . Они оказались равны и больше оценки коэффициента корреляции Пирсона. Это говорит о том, что выборочные данные согласованы с предположением о том, что и связаны корреляционной зависимостью. Но характер этой зависимости определить не удалось.

В конце работы были составлены и решены СЛУ в результате чего были получены уравнения выборочных кривых параболической среднеквадратической регрессии. Оценки коэффициентов при близки к нулю, что может говорить о том, что квадратичной зависимости скорее всего нет. После визуализации этих уравнений видно, что они близки к прямым, что косвенно подтверждает догадку об отсутствии квадратичной зависимости между исследуемыми параметрами.

**ПРИЛОЖЕНИЕ А**

**ИСХОДНЫЙ КОД**

from math import sqrt

import lab1

import lab2

import lab4

n = lab1.selection\_size

general\_population = lab1.read\_data(filename=lab1.data\_file\_name)

sample\_density = lab1.get\_sample\_first(general\_population, n)

sample\_elastic = lab1.get\_sample\_second(general\_population, n)

table\_x = lab2.build\_table(sample\_density)

mean\_x, S\_x = lab2.get\_main\_values\_from\_table(table\_x)

table\_y = lab2.build\_table(sample\_elastic)

mean\_y, S\_y = lab2.get\_main\_values\_from\_table(table\_y)

r = lab4.get\_correl\_coef(sample\_density, sample\_elastic)

print("Для X:\n\tmean = {0:.2f}\n\tS = {1:.2f}".format(mean\_x, S\_x))

print("Для Y:\n\tmean = {0:.2f}\n\tS = {1:.2f}".format(mean\_y, S\_y))

print("Коэффициент корреляции: {0:.3f}".format(r))

coef\_y\_x = r \* S\_y / S\_x

free\_number\_y\_x = mean\_y + coef\_y\_x\*mean\_x\*(-1)

print(free\_number\_y\_x)

coef\_x\_y = r \* S\_x / S\_y

free\_number\_x\_y = mean\_x + coef\_x\_y\*mean\_y\*(-1)

print(free\_number\_x\_y)

resudial\_disp\_x = pow(S\_x,2) \* pow(1-r, 2)

resudial\_disp\_y = pow(S\_y,2) \* pow(1-r, 2)

# print(resudial\_disp\_x, resudial\_disp\_y)

# import matplotlib.pyplot as plt

# fig, ax = plt.subplots()

# ax.plot(sample\_density, sample\_elastic, 'ok')

# min\_val, max\_val = min(sample\_density), max(sample\_density)

# y\_minval = mean\_y + coef\_y\_x\*(min\_val - mean\_x)

# y\_maxval = mean\_y + coef\_y\_x\*(max\_val - mean\_x)

# ax.plot([min\_val, max\_val], [y\_minval, y\_maxval], "-r", label="$y\_x$ = 0.31x - 12.33") # yx

# min\_val, max\_val = min(sample\_elastic), max(sample\_elastic)

# print(min\_val, max\_val)

# x\_minval = mean\_x + coef\_x\_y\*(min\_val - mean\_y)

# x\_maxval = mean\_x + coef\_x\_y\*(max\_val - mean\_y)

# ax.plot([x\_minval, x\_maxval], [min\_val, max\_val], "-b", label="$x\_y$ = 1.91y + 212.58") # xy

# ax.legend()

# # ax.set\_xlabel('Варианты')

# # ax.set\_ylabel('Абсолютная частота')

# ax.set\_title('Прямые среднеквадратической регрессии')

# # fig.tight\_layout()

# plt.show()

borders\_x, buckets\_x = lab1.get\_interval\_sample(sample\_density)

borders\_y, buckets\_y = lab1.get\_interval\_sample(sample\_elastic)

freqs\_x = [len(array) for array in buckets\_x]

freqs\_y = [len(array) for array in buckets\_y]

mid\_borders\_x = [(border[0] + border[1])/2 for border in borders\_x]

mid\_borders\_y = [(border[0] + border[1])/2 for border in borders\_y]

for elem in mid\_borders\_x:

print("{0:.2f}".format(elem), end=" ")

print()

# print(freqs\_x)

for elem in mid\_borders\_y:

print("{0:.2f}".format(elem), end=" ")

print()

# print(freqs\_y)

sample\_2D = list(zip(sample\_density, sample\_elastic))

table = lab4.build\_corr\_table(sample\_2D, borders\_x, borders\_y)

for row in table:

print(row)

print(end="\n")

n\_y = []

x\_y\_cherta = []

for i in range(len(table)):

summ = 0

num = 0

for j in range(len(table)):

num += table[j][i]

summ += table[j][i] \* mid\_borders\_x[j]

n\_y.append(num)

x\_y\_cherta.append(summ / num)

# n\_y[-1] +=1

n\_x = []

y\_x\_cherta = []

for i in range(len(table)):

summ = 0

num = 0

for j in range(len(table)):

num += table[i][j]

summ += table[i][j] \* mid\_borders\_y[j]

n\_x.append(num)

y\_x\_cherta.append(summ / num)

# n\_x[-1] += 1

D\_o\_xy = sum([n\_x[i] \* pow((mid\_borders\_x[i]-mean\_x),2) for i in range(len(table))]) / n

sigma\_x\_xy = sqrt(D\_o\_xy)

D\_m\_xy = sum([n\_y[i] \* pow((x\_y\_cherta[i]-mean\_x),2) for i in range(len(table))]) / n

sigma\_xy = sqrt(D\_m\_xy)

nu\_xy = sigma\_xy / sigma\_x\_xy

print("D(общ): {0:.2f}".format(D\_o\_xy))

print("σ(x): {0:.2f}".format(sigma\_x\_xy))

print("D(меж): {0:.2f}".format(D\_m\_xy))

print("σ(xy): {0:.2f}".format(sigma\_xy))

print("η(xy): {0:.3f}".format(nu\_xy))

D\_o\_xy = sum([n\_y[i] \* pow((mid\_borders\_y[i]-mean\_y),2) for i in range(len(table))]) / n

sigma\_x\_xy = sqrt(D\_o\_xy)

D\_m\_xy = sum([n\_x[i] \* pow((y\_x\_cherta[i]-mean\_y),2) for i in range(len(table))]) / n

sigma\_xy = sqrt(D\_m\_xy)

nu\_xy = sigma\_xy / sigma\_x\_xy

print()

print("D(общ): {0:.2f}".format(D\_o\_xy))

print("σ(x): {0:.2f}".format(sigma\_x\_xy))

print("D(меж): {0:.2f}".format(D\_m\_xy))

print("σ(yx): {0:.2f}".format(sigma\_xy))

print("η(yx): {0:.3f}".format(nu\_xy))

# 3 часть

import matplotlib.pyplot as plt

fig, ax = plt.subplots()

ax.plot(sample\_density, sample\_elastic, 'ok')

sum\_1 = sum([n\_x[i] \* mid\_borders\_x[i]\*\*1 for i in range(len(table))])

sum\_2 = sum([n\_x[i] \* mid\_borders\_x[i]\*\*2 for i in range(len(table))])

sum\_3 = sum([n\_x[i] \* mid\_borders\_x[i]\*\*3 for i in range(len(table))])

sum\_4 = sum([n\_x[i] \* mid\_borders\_x[i]\*\*4 for i in range(len(table))])

sum\_y\_1 = sum([n\_x[i] \* y\_x\_cherta[i] \* mid\_borders\_x[i]\*\*1 for i in range(len(table))])

sum\_y\_2 = sum([n\_x[i] \* y\_x\_cherta[i] \* mid\_borders\_x[i]\*\*2 for i in range(len(table))])

sum\_y\_3 = sum([n\_x[i] \* y\_x\_cherta[i] \* mid\_borders\_x[i]\*\*0 for i in range(len(table))])

print(round(sum\_1))

print(round(sum\_2))

print(round(sum\_3))

print(round(sum\_4))

print(round(sum\_y\_1))

print(round(sum\_y\_2))

print(round(sum\_y\_3))

a = 1330980770795 / 7822224861288736

b = 1585312202138487 / 7822224861288736

c = (-574079274021227) / 7822224861288736

print()

print("a = {:.5f}".format(a))

print("b = {:.5f}".format(b))

print("c = {:.5f}".format(c))

print()

min\_val, max\_val = min(sample\_density), max(sample\_density)

xs = [min\_val+((max\_val-min\_val)/10000\*i) for i in range(10000)]

ys = [a\*xs[i]\*\*2 + b\*xs[i] + c for i in range(10000)]

ax.plot(xs, ys, label="$y\_x$") # yx

sum\_1 = sum([n\_y[i] \* mid\_borders\_y[i]\*\*1 for i in range(len(table))])

sum\_2 = sum([n\_y[i] \* mid\_borders\_y[i]\*\*2 for i in range(len(table))])

sum\_3 = sum([n\_y[i] \* mid\_borders\_y[i]\*\*3 for i in range(len(table))])

sum\_4 = sum([n\_y[i] \* mid\_borders\_y[i]\*\*4 for i in range(len(table))])

sum\_y\_1 = sum([n\_y[i] \* x\_y\_cherta[i] \* mid\_borders\_y[i]\*\*2 for i in range(len(table))])

sum\_y\_2 = sum([n\_y[i] \* x\_y\_cherta[i] \* mid\_borders\_y[i]\*\*1 for i in range(len(table))])

sum\_y\_3 = sum([n\_y[i] \* x\_y\_cherta[i] \* mid\_borders\_y[i]\*\*0 for i in range(len(table))])

print(round(sum\_1))

print(round(sum\_2))

print(round(sum\_3))

print(round(sum\_4))

print(round(sum\_y\_1))

print(round(sum\_y\_2))

print(round(sum\_y\_3))

a = 3400468111 / 11432286831000

b = 24837124860883 / 11432286831000

c = 7906387274955121 / 45729147324000

print()

print("a = {:.5f}".format(a))

print("b = {:.5f}".format(b))

print("c = {:.5f}".format(c))

print()

min\_val, max\_val = min(sample\_elastic), max(sample\_elastic)

ys = [min\_val+((max\_val-min\_val)/10000\*i) for i in range(10000)]

xs = [a\*ys[i]\*\*2 + b\*ys[i] + c for i in range(10000)]

ax.plot(xs, ys, label="$x\_y$")

ax.legend()

ax.set\_title('Уравнения параболической регрессии')