Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №3

З дисципліни «Методи наукових досліджень»

За темою:

«ПРОВЕДЕННЯ ТРЬОХФАКТОРНОГО ЕКСПЕРИМЕНТУ З ВИКОРИСТАННЯМ

ЛІНІЙНОГО РІВНЯННЯ РЕГРЕСІЇ»

ВИКОНАВ:

Студент ІІ курсу ФІОТ

Групи ІВ-91

Гришин О.С.

Номер у списку - 07

ПЕРЕВІРИВ:

асистент

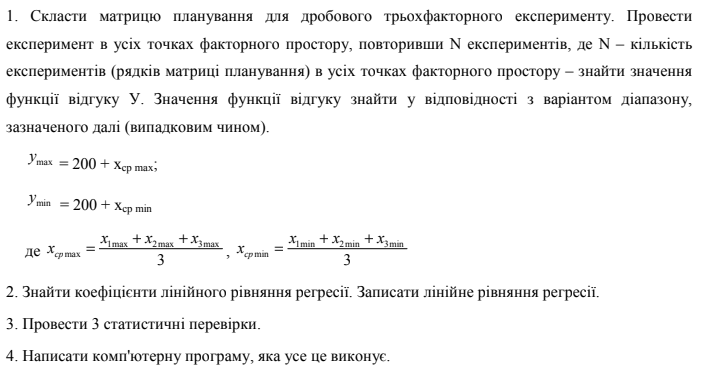
Регіда П.Г.

Київ 2021 р.

**Мета:** провести дробовий трьохфакторний експеримент. Скласти матрицю планування, знайти

коефіцієнти рівняння регресії, провести 3 статистичні перевірки.

**Завдання:**



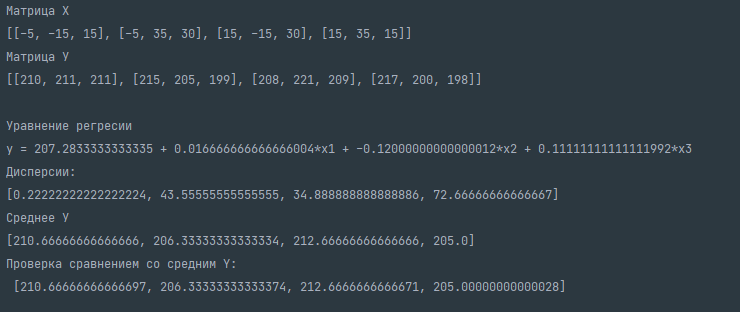
****

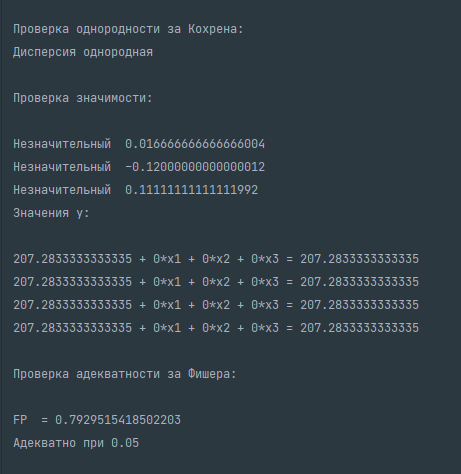
**Програмний код**

*from* script *import* Experiment  
  
*if* \_\_name\_\_ == '\_\_main\_\_':  
 exp = Experiment([[-5, 15], [- 15, 35], [15, 30]])  
 exp.show\_info = *True* exp.make\_experiment()

*from* random *import* \*  
*from* pprint *import* pprint  
*import* numpy *as* np  
*from* math *import* sqrt  
  
  
*class* Experiment:  
 m, N, d = 3, 4, 4  
 mat\_1X = [[1, -1, -1, -1],  
 [1, -1, 1, 1],  
 [1, 1, -1, 1],  
 [1, 1, 1, -1]]  
 show\_info = *False  
  
 def \_\_init\_\_*(*self*, x\_list):  
 *"""* ***:param*** *x\_list: список иксов  
 """* """матрица иксов"""  
 *self*.mat\_sX = x\_list  
 *self*.mat\_X = [[*self*.mat\_sX[0][0], *self*.mat\_sX[1][0], *self*.mat\_sX[2][0]],  
 [*self*.mat\_sX[0][0], *self*.mat\_sX[1][1], *self*.mat\_sX[2][1]],  
 [*self*.mat\_sX[0][1], *self*.mat\_sX[1][0], *self*.mat\_sX[2][1]],  
 [*self*.mat\_sX[0][1], *self*.mat\_sX[1][1], *self*.mat\_sX[2][0]]]  
  
 """Т-матрица иксов"""  
 *self*.trans\_sx = [*list*(i) *for* i *in zip*(\**self*.mat\_1X)]  
 *self*.trans\_x = [*list*(i) *for* i *in zip*(\**self*.mat\_X)]  
  
 *self*.x\_min\_max\_av = [*sum*(*self*.mat\_sX[i][k] *for* i *in range*(3)) / 3 *for* k *in range*(2)]  
 *self*.y\_min\_max = [*int*(200 + *self*.x\_min\_max\_av[i]) *for* i *in range*(2)]  
  
 """матрица игреков"""  
 *self*.mat\_Y = [[randint(*self*.y\_min\_max[0], *self*.y\_min\_max[1]) *for* \_ *in range*(*self*.m)] *for* \_ *in range*(*self*.N)]  
  
 *def* get\_average\_y(*self*):  
 *"""середні значення функцій"""  
 return* [*sum*(*self*.mat\_Y[k1]) / *self*.m *for* k1 *in range*(*self*.N)]  
  
 *def* get\_dispersion(*self*):  
 *return* [*sum*([((k1 - *self*.get\_average\_y()[j]) \*\* 2) *for* k1 *in self*.mat\_Y[j]]) / *self*.m *for* j *in range*(*self*.N)]  
  
 *def* show\_dets(*self*, check, b\_list):  
 *print*('Матрица Х')  
 pprint(*self*.mat\_X)  
 *print*('Матрица У')  
 pprint(*self*.mat\_Y)  
  
 *print*(f"\nУравнение регресии\ny = {b\_list[0]} + {b\_list[1]}\*x1 + {b\_list[2]}\*x2 + {b\_list[3]}\*x3")  
 *print*("Дисперсии:")  
 *print*(*self*.get\_dispersion())  
  
 *print*('Среднее У')  
 *print*(*self*.get\_average\_y())  
 *print*("Проверка сравнением со средним Y:\n", check)  
 *print*()  
  
 *def* get\_mx\_my(*self*):  
 *"""Знайдемо коефіцієнти рівняння регресії"""* mx = [*sum*(*self*.mat\_X[i][k] *for* i *in range*(*self*.N)) / *self*.N *for* k *in range*(*self*.m)]  
 my = *sum*(*self*.get\_average\_y()) / *self*.N  
 *return* mx, my  
  
 *def* get\_ai\_aii(*self*):  
 ai = [*sum*(*self*.trans\_x[k][i] \* *self*.get\_average\_y()[i] *for* i *in range*(*self*.N)) / *self*.N *for* k *in range*(*self*.m)]  
 aii = [*sum*(*self*.trans\_x[k][i] \*\* 2 *for* i *in range*(*self*.N)) / *self*.N *for* k *in range*(*self*.m)]  
 *return* ai, aii  
  
 *def* find\_cf(*self*):  
 tran = *self*.trans\_x  
 mx, my = *self*.get\_mx\_my()  
 ai, aii = *self*.get\_ai\_aii()  
  
 a12 = a21 = (tran[0][0] \* tran[1][0] + tran[0][1] \* tran[1][1] + tran[0][2] \* tran[1][2] + tran[0][3] \* tran[1][3]) / *self*.N  
 a13 = a31 = (tran[0][0] \* tran[2][0] + tran[0][1] \* tran[2][1] + tran[0][2] \* tran[2][2] + tran[0][3] \* tran[2][3]) / *self*.N  
 a23 = a32 = (tran[1][0] \* tran[2][0] + tran[1][1] \* tran[2][1] + tran[1][2] \* tran[2][2] + tran[1][3] \* tran[2][3]) / *self*.N  
 znamen = np.linalg.det(np.matrix(  
 [[1, mx[0], mx[1], mx[2]],  
 [mx[0], aii[0], a12, a13],  
 [mx[1], a12, aii[1], a32],  
 [mx[2], a13, a23, aii[2]]]))  
  
 b0 = np.linalg.det(np.matrix([[my, mx[0], mx[1], mx[2]],  
 [ai[0], aii[0], a12, a13],  
 [ai[1], a12, aii[1], a32],  
 [ai[2], a13, a23, aii[2]]])) / znamen  
  
 b1 = np.linalg.det(np.matrix([[1, my, mx[1], mx[2]],  
 [mx[0], ai[0], a12, a13],  
 [mx[1], ai[1], aii[1], a32],  
 [mx[2], ai[2], a23, aii[2]]])) / znamen  
  
 b2 = np.linalg.det(np.matrix([[1, mx[0], my, mx[2]],  
 [mx[0], aii[0], ai[0], a13],  
 [mx[1], a12, ai[1], a32],  
 [mx[2], a13, ai[2], aii[2]]])) / znamen  
  
 b3 = np.linalg.det(np.matrix([[1, mx[0], mx[1], my],  
 [mx[0], aii[0], a12, ai[0]],  
 [mx[1], a12, aii[1], ai[1]],  
 [mx[2], a13, a23, ai[2]]])) / znamen  
  
 check = [b0 + b1 \* tran[0][i] + b2 \* tran[1][i] + b3 \* tran[2][i] *for* i *in range*(4)]  
 b\_list = [b0, b1, b2, b3]  
 *return* check, b\_list  
  
 *def* make\_experiment(*self*):  
 mat\_disY = *self*.get\_dispersion()  
 check, b\_list = *self*.find\_cf()  
 *if self*.show\_info:  
 *self*.show\_dets(check, b\_list)  
  
 *print*('\nПроверка однородности за Кохрена:')  
 *if max*(mat\_disY) / *sum*(mat\_disY) < 0.7679:  
 *print*('Дисперсия однородная')  
 *else*:  
 *print*('Дисперсия неоднородная')  
  
 *print*('\nПроверка значимости:\n')  
 S2b = *sum*(mat\_disY) / *self*.N  
 S2bs = S2b / (*self*.m \* *self*.N)  
 Sbs = sqrt(S2bs)  
 bb = [*sum*(*self*.get\_average\_y()[k] \* *self*.trans\_sx[i][k] *for* k *in range*(*self*.N))/*self*.N *for* i *in range*(*self*.N)]  
 t = [*abs*(bb[i])/Sbs *for* i *in range*(*self*.N)]  
 *if self*.show\_info:  
 *print*('Sbs:\n', Sbs)  
 *print*('bi:\n', bb, '\nti:\n', t)  
 *for* i *in range*(*self*.N):  
 *if* t[i] < 2.306:  
 *print*('Незначительный ', b\_list[i])  
 b\_list[i] = 0  
 *self*.d -= 1  
  
 y\_reg = [b\_list[0] + b\_list[1] \* *self*.mat\_X[i][0] + b\_list[2] \* *self*.mat\_X[i][1] + b\_list[3] \* *self*.mat\_X[i][2] *for* i *in range*(*self*.N)]  
 *print*('Значения у:\n')  
 [*print*(f"{b\_list[0]} + {b\_list[1]}\*x1 + {b\_list[2]}\*x2 + {b\_list[3]}\*x3 = {b\_list[0] + b\_list[1] \* *self*.mat\_X[i][0] + b\_list[2] \* *self*.mat\_X[i][1] + b\_list[3] \* *self*.mat\_X[i][2]}") *for* i *in range*(*self*.N)]  
  
 *print*('\nПроверка адекватности за Фишера:\n')  
 Sad = (*self*.m / (*self*.N - *self*.d)) \* *int*(*sum*(y\_reg[i] - *self*.get\_average\_y()[i] *for* i *in range*(*self*.N)) \*\* 2)  
 Fp = Sad / S2b  
 *print*('FP =', Fp)  
 *if* Fp > 4.5:  
 *print*('Неадекватно при 0.05')  
 *else*:  
 *print*('Адекватно при 0.05')

**Результати роботи програми**





**Контрольні запитання:**

1. **Дробовий факторний експеримент – частина ПФЕ, який мінімізує число дослідів, за рахунок тієї інформації, яка не дуже істотна для побудови моделі**
2. **Значення Кохрена використовують для перевірки однорідності дисперсії**
3. **Критерій Стьюдента перевіряє значущість коефіцієнтів рівняння**
4. **Критерій Фішера використовують при перевірці отриманого рівняння регресії досліджуваному об’єкту**