1. Даны значения величины заработной платы заемщиков банка (zp) и значения их поведенческого кредитного скоринга (ks): zp = [35, 45, 190, 200, 40, 70, 54, 150, 120, 110], ks = [401, 574, 874, 919, 459, 739, 653, 902, 746, 832]. Используя математические операции, посчитать коэффициенты линейной регрессии, приняв за X заработную плату (то есть, zp - признак), а за y - значения скорингового балла (то есть, ks - целевая переменная). Произвести расчет как с использованием intercept,

n = 10  
  
x = np.array([35, 45, 190, 200, 40, 70, 54, 150, 120, 110], dtype=np.float64)  
y = np.array([401, 574, 874, 919, 459, 739, 653, 902, 746, 832], dtype=np.float64)  
  
b = (np.mean(x \* y) - np.mean(x) \* np.mean(y)) / (np.mean(x\*\*2) - np.mean(x) \*\* 2)

b = 2.620538882402765

a = np.mean(y) - b \* np.mean(x)

a = 444.1773573243596

mse = 6470.414201176658

так и без.

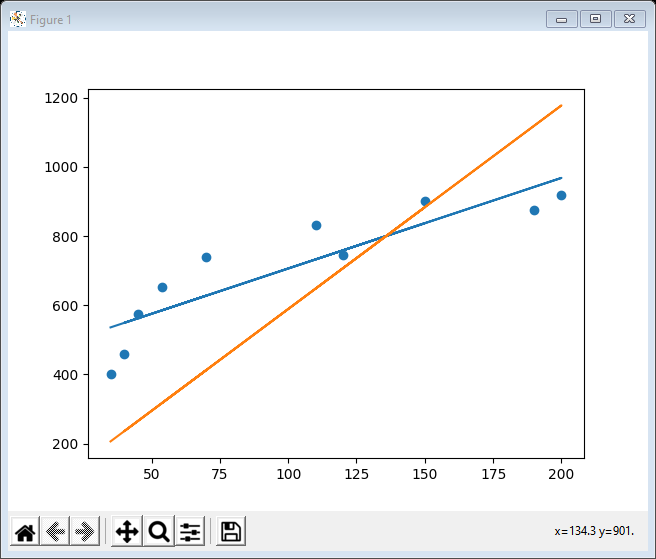
Sum (b\*x-y)^2 -> min сумма квадратов отклонений стремится к минимуму, производная = 0

Sum (2\*(b\*x-y)\*x) = sum (2b\*x^2 –2x\*y) = b\* sum(2\* x^2) –sum (2\*x\*y) = 0

B= sum (x\*y)/ \* sum( x^2)

b1 = np.sum(x\*y)/np.sum(x\*x)

без intercept b = 5.889820420132689



1. Посчитать коэффициент линейной регрессии при заработной плате (zp), используя градиентный спуск (без intercept).

iteration 500, B1 = 5.889815458885515, mse = 56516.858416058494

iteration 1000, B1 = 5.889820420128437, mse = 56516.8584157194

iteration 1500, B1 = 5.889820420132673, mse = 56516.85841571943

iteration 2000, B1 = 5.889820420132673, mse = 56516.85841571943

1. В каких случаях для вычисления доверительных интервалов и проверки статистических гипотез используется таблица значений функции Лапласа, а в каких - таблица критических точек распределения Стьюдента?

Таблица значений функции Лапласа используется при неизвестном математическом ожидании и известной дисперсии нормально распределенной случайной величины

Таблица критических точек распределения Стьюдента используется при неизвестном математическом ожидании и дисперсии нормально распределенной случайной величины

1. Посчитать коэффициент линейной регрессии при заработной плате (zp), используя градиентный спуск (c intercept).  Учесть, что изменение коэффициентов должно производиться на каждом шаге одновременно (то есть изменение одного коэффициента не должно влиять на изменение другого во время одной итерации).

Sum (a+b\*x-y)^2/n -> производная по а: Sum (a+b\*x-y)\*(2/n)

Sum (a+b\*x-y)^2/n -> производная по b: Sum (a+b\*x-y)\*(2/n)\*x

alpha = 1e-5  
a\_1 = 0.1  
b\_1 = 0.1  
for i in range(3000000):  
 a = a\_1  
 a\_1 -= alpha \* (2 / n) \* np.sum((b\_1 \* x + a\_1 - y))  
 b\_1 -= alpha \* (2 / n) \* np.sum((b\_1 \* x + a - y) \* x)  
 if (i + 1) % 500000 == 0:  
 print(f'iteration {i + 1}, A = {a\_1}, B = {b\_1}, mse = {np.sum((a\_1 + b\_1 \* x - y) \*\* 2) / n}')

iteration 500000, A = 409.03578929710056, B = 2.8791963600631103, mse = 6783.672559008695

iteration 1000000, A = 441.3962015560758, B = 2.6410094184769326, mse = 6472.3762554772975

iteration 1500000, A = 443.9572525355978, B = 2.6221589509619694, mse = 6470.426490256849

iteration 2000000, A = 444.15993790522555, B = 2.6206670970291595, mse = 6470.414278147766

iteration 2500000, A = 444.17597872572384, B = 2.620549029498238, mse = 6470.41420165876

iteration 3000000, A = 444.1772482200196, B = 2.6205396854589456, mse = 6470.4142011796785