Coding Assignment #1

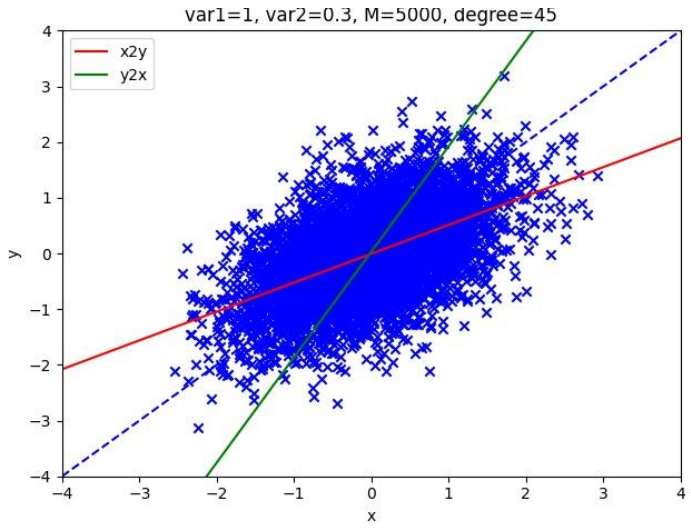
Negar Mirgati mirgati@ualberta.ca
SID: 1722860

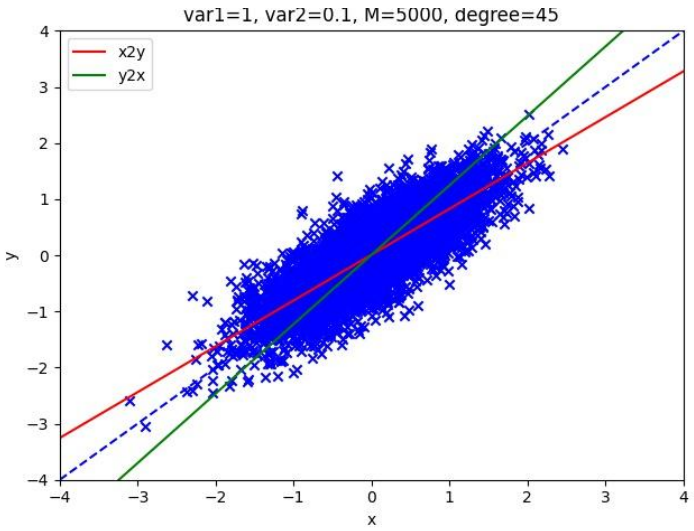
Part 1)

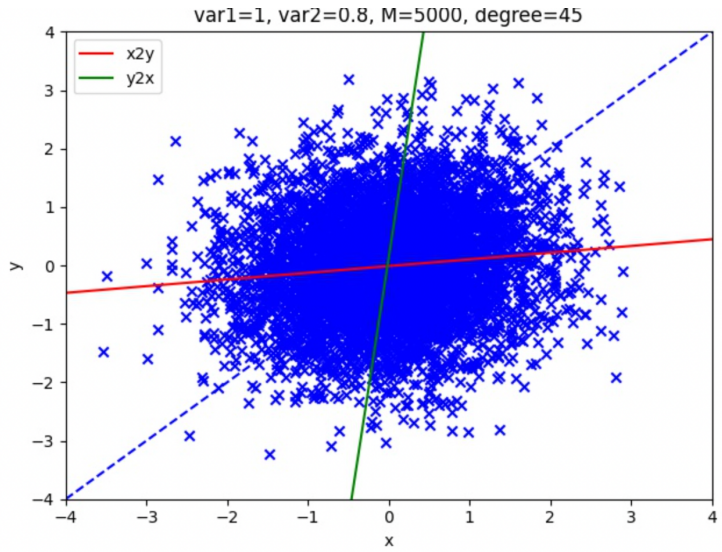
1) M = 5000, var1 = 1, var2 = 0.3, degree = 45

	weight	bias
x2y	0.529462183005872	0.0035926123150616437
y2x	0.5198990188826349	-0.017016549049388933

2) Three plots of regression in a row with var2 = 0.3, 0.1, 0.8,





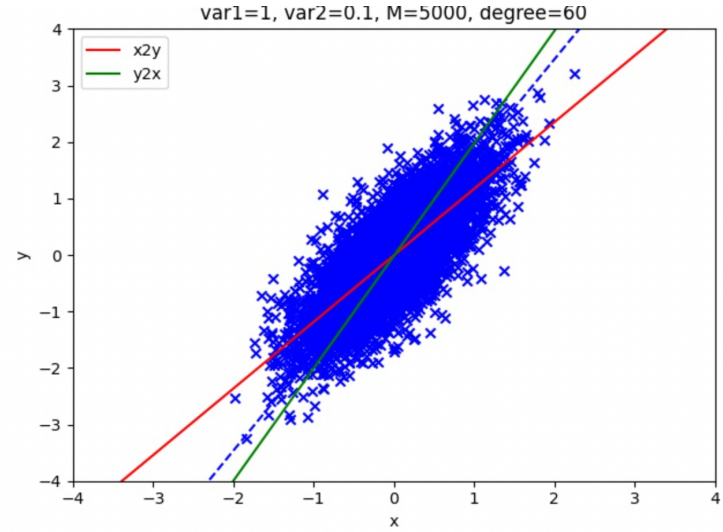


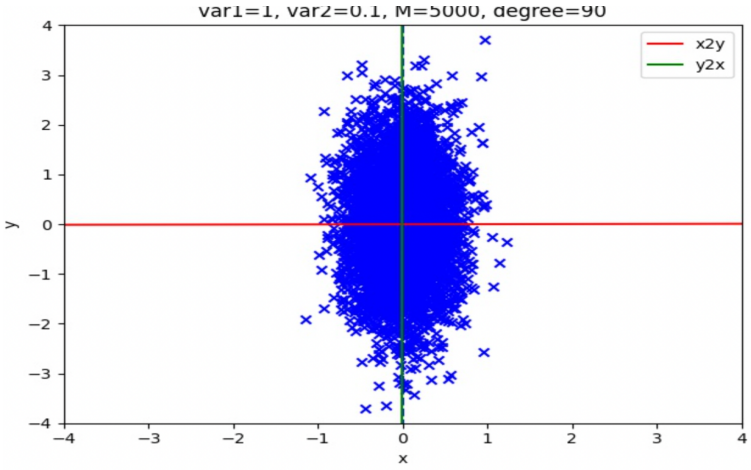
3) A description of what happened in 1 & 2

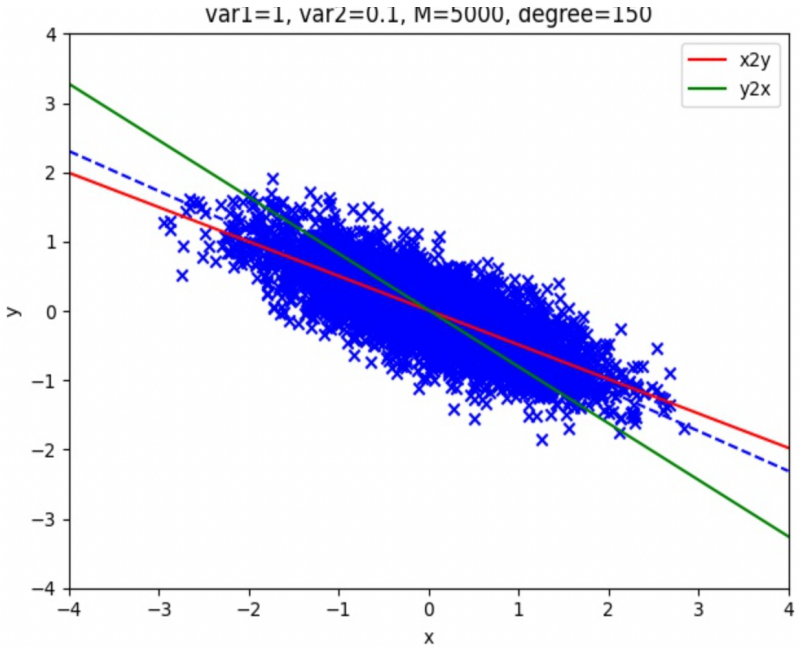
In part one, we are observing that the slope of the x2y and y2x are pretty close to each other. That is observable from the plot of data because the points show a trend between the variables x, y. We are seeing that when x is higher, y tends to be higher as well, and vice versa.

In part two, we are observing that as the var2 increases, data is more scattered in the 2D space, and it is harder to see a trend between x, y. In the case where var2=0.1, we see that the data points are more concentrated and the two prediction lines are closer together and to the blue line. However, in the third picture where var2 is greater than the other cases, prediction lines are farther away from each other and from the blue line. We can conclude that when correlation is higher between the variables, the prediction lines become closer to each other and when the correlation is lower, the prediction lines are farther away from each other.

4) I implemented a for loop to change the degree from 0 to 360 with step size 30. I observed that at the beginning of the process when degree is zero, the dashed line is at the same location as the red line, and the green line is approximately orthogonal to the other two lines. As we increase the degree, the dashed line moves farther away from the red line and becomes closer to the green line, until at degree=90, when the dashed line and the green line will be located at the same position and the red line is approximately orthogonal to the other two lines. Again, the dashed line goes farther away from the green line and this process continues in the same pattern until degree=360.


var1=1, var2=0.1, M=5000, degree=60
x2y
y2x
x
y

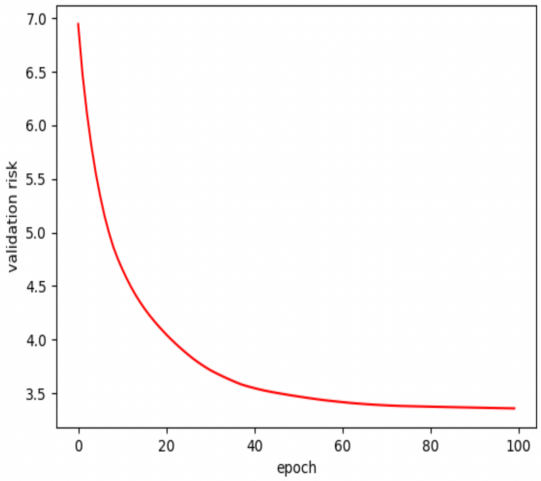

var1=1, var2=0.1, M=5000, degree=90
x2y
y2x
x
y

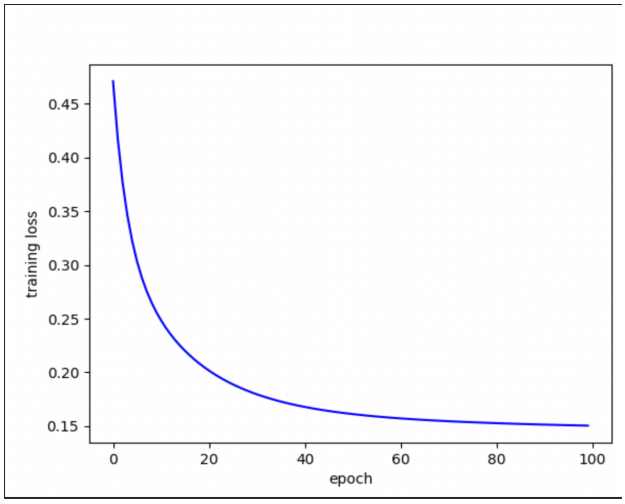

var1=1, var2=0.1, M=5000, degree=150
x2y
y2x
x
y

Part 2)

a)

Below is the validation risk/epoch and training loss/epoch. The best epoch is this part was 99.



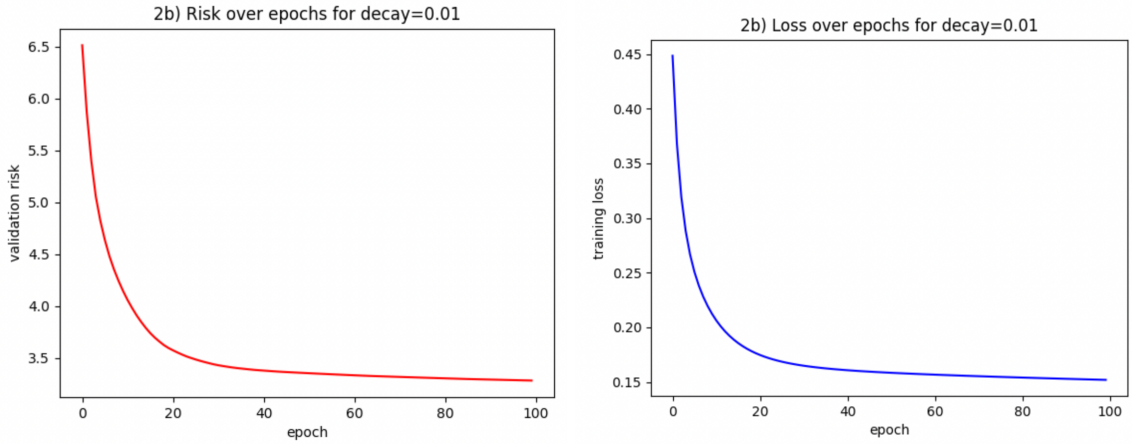


Furthermore, the best validation risk, best epoch, and test risk for the best decay(0.01) is presented in the below table:

Best epoch	Best epoch validation risk	Test data risk
99	3.3580	3.2370

b)

Below is the validation risk/epoch and training loss/epoch for the best decay which was **0.01**



Furthermore, the best validation risk, best epoch, and test risk for the best decay(0.01) is presented in the below table:

	Best epoch validation risk	Best epoch	Test data risk
Best decay=0.01	3.2804	99	3.1532

c) Questions: How do the values of learning_rate affect our results? What is the best learning rate for this task? Can we improve our results by finding a better learning rate?

To explore the consequences of different learning rate values, I set alpha to different values and recorded the results.

Below is the loss/risk values for different learning rates:

Learning Rate	Validation Risk	Test Risk
0.1	3.6702	3.5920
0.01	3.3253	3.3215
0.001	3.3215	3.2370
0.0001	4.7560	3.7112
0.00001	6.9472	5.4682

I added a for loop to change the learning rate between 0.00001, 0.1 with step size 0.001, and I plotted best validation risks vs learning rates. I also saved the validation risks of different learning rates, and found the minimum validation risk, which was **3.3215.** This is the same value as our default learning rate of part a, which was 0.001. Based on this observation, I conclude that we have chosen the best value for our learning rate, and we can't improve our results further by changing alpha.