# Problem 1

1. When running the thetas independently I ended up with the resulting thetas and final costs in the table below

|  |  |  |
| --- | --- | --- |
| # | Theta | cost |
| 0 | [ 0.92563782 0.92563782 ] | 3.81082769164781 |
| 1 | [ 5.29451128 -1.79179951 ] | 1.0361829849497886 |
| 2 | [ 0.68714072 0.57664762 ] | 3.5996713196049464 |
| 3 | [ 2.59189722 -0.40993238] | 3.6396084981267816 |

1. For the graphs kindly refer to the descent PDF file. I’m unsure of how to extract them.
2. X\_1 had the lowest cost function with the theta value of 1.0361829849497886.
3. The learning rate affected how quickly the training model got to its local minimum for the single explanatory variable. With a higher value it would jump to an acceptable value and then train on values adjusting somewhere close to 10-12 for the next few hundred iterations.

# Problem 2

1. The best linear model I found was
2. See plot sets (inclusive). Sets 0 thru 5 is starting theta at [ 0 0 0 0 ]
3. The learning rate affected how quickly the training model got to a minimum for the explanatory variable. The lower learning rates as expected changed how quickly it started to plateau its overall learning. However I suspect in actual training that we want to use the lower rates to not over-fit the data.
4. See output below

Y([1, 1, 1]) = 3.577409368656757

Y([2, 0, 4]) = 0.2443211714832545

Y([3, 2, 1]) = 0.10253417186972857