Madison Chamberlain

Ch9 Q2:

Below is a table demonstrating how test size affects optimal cost and degree when searching for the highest accuracy model.

Test size:	**Optimal Degree(s):**	**Optimal Cost(s):**
0.2	1, 2	0.2, 0.4
0.3	1	0.2, 0.6, 0.8, 1.0, 1.2
0.4	1	0.2, 0.6, 0.8, 1.0
0.5	1	1.0, 1.2, 1.4

As you can see in the table, changing the test size yielded different optimal selections for both cost and degree. In terms of cost, no two test sizes yielded the same set of optimal costs. There seems to be a slight trend in the direction of larger test sizes producing better models when larger costs are used. I would expect that this may be due to the fact that as the test size increases, the train set size decreases. In order to not overfit the model, and thus increase accuracy, we want to increase the cost of an incorrect prediction, or else our prediction may fit the train set well, but not actually be a good model for all of the data. In terms of degree, all test sizes were most accurate when degree was 1 with the exclusion of when test size was set to 0.2; in this case degree 2 produced an equally accurate prediction. I hypothesise that if each train set were an accurate representation of the population, each test size would have had the same degree. I believe that this difference in optimal degrees for different test sizes is a result of the test set generated when the test size was set to 0.2 randomly not being an accurate representation of the population.

Below is a table which demonstrates the effect of different parameters on runtime. Note not all degrees are equally represented, as I stopped running each cost for each degree because it was taking a very large span of time to run.

Degree	**Cost**	**Time**
1	0.2	0.0169
1	0.6	0.0383
1	1.0	0.0656
1	1.4	0.0910
1	1.8	0.1318
2	0.2	78.0151
2	0.6	74.01636
2	1.0	75.3699
2	1.4	72.2628
2	1.8	72.3227
3	0.2	276.1786
3	0.4	281.3812

As far as runtime goes, changing cost did not seem to significantly affect runtime, while changing degree greatly changed runtime. When the degree was set to 1, for all costs it took about 0.05 seconds to generate a polynomial function which was fit to the data. When the degree was changed to 2, for all costs, the time increased to about 75 seconds, and then when the degree was changed to 3, for all costs, the time increased to about 275 seconds. It makes sense that changing the cost would not change the runtime, since cost is a constant in the equation used to generate a fit; a different constant will not yield more calculation, so no extra time is needed. On

the other hand, the higher the degree, the more calculations the computer has to run, so it makes sense that a higher degree would take much more time to run.