Homework 6 Report

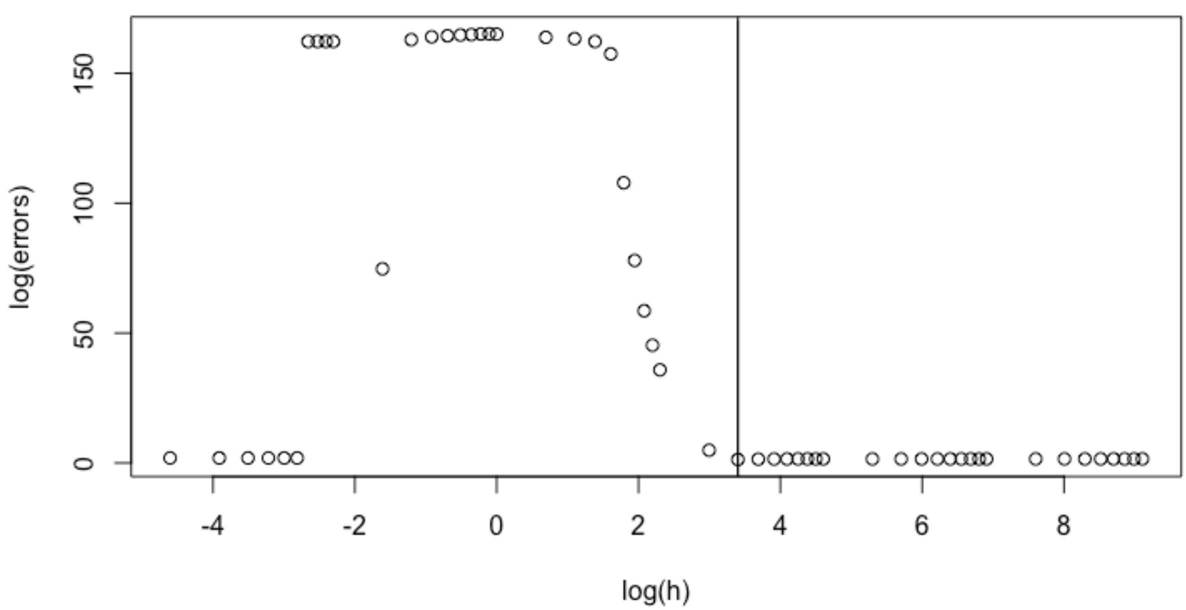
Part 1 : Non-Regularized Model

Preprocessing the Data

In order to preprocess the data, I divided the East and North UTM scales by 1000 to get them in units of kilometers instead of meters. The problem with having the scales in meters is that the squared distances between points were difficult to manage due to the high magnitudes. I also calculated the annual minimum temperatures average for each base point, and then calculated the average of the annual minimum temperatures averages for 2000-2005.

Cross-Validation and Training

I split my data into a 70%-30% train-test split. A cross-validated on the scales of the kernel function over several orders of magnitude ranging from 10^-2 to 10^3. I found that the best scales ended up being around 30 with a mean-squared error of 5.129 for the simple linear regression model. I trained the model using cv.glmnet with lambda equal to 0.



The figure above shows cross-validation for the scales. The best scale ended up being h = 30 with a mean-squared error of 3.80015. (Note: the scale is logarithmic on both axis). Multiple runs produced scales within the range {20, 30, 40, 50, 60, 70}.

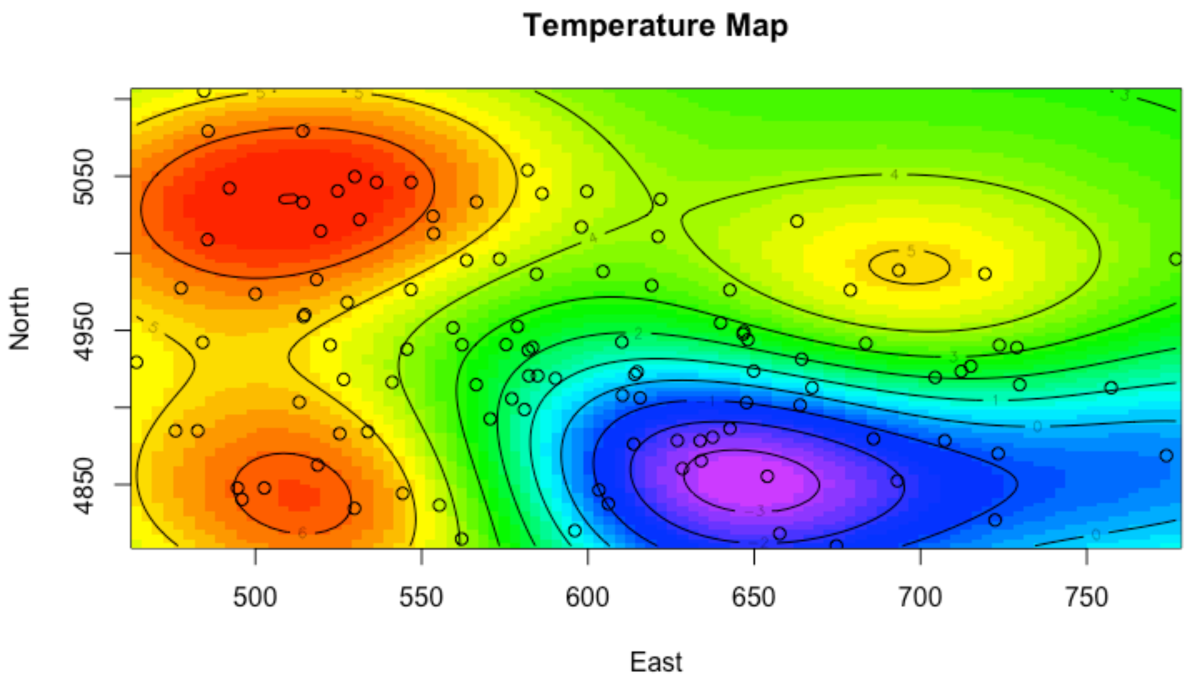
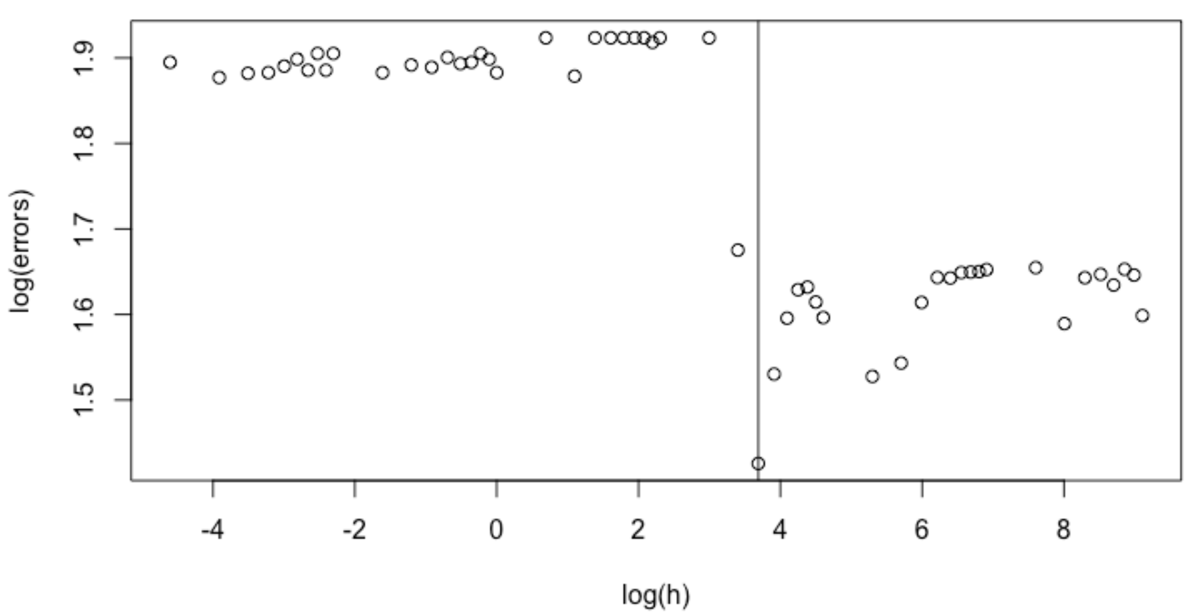
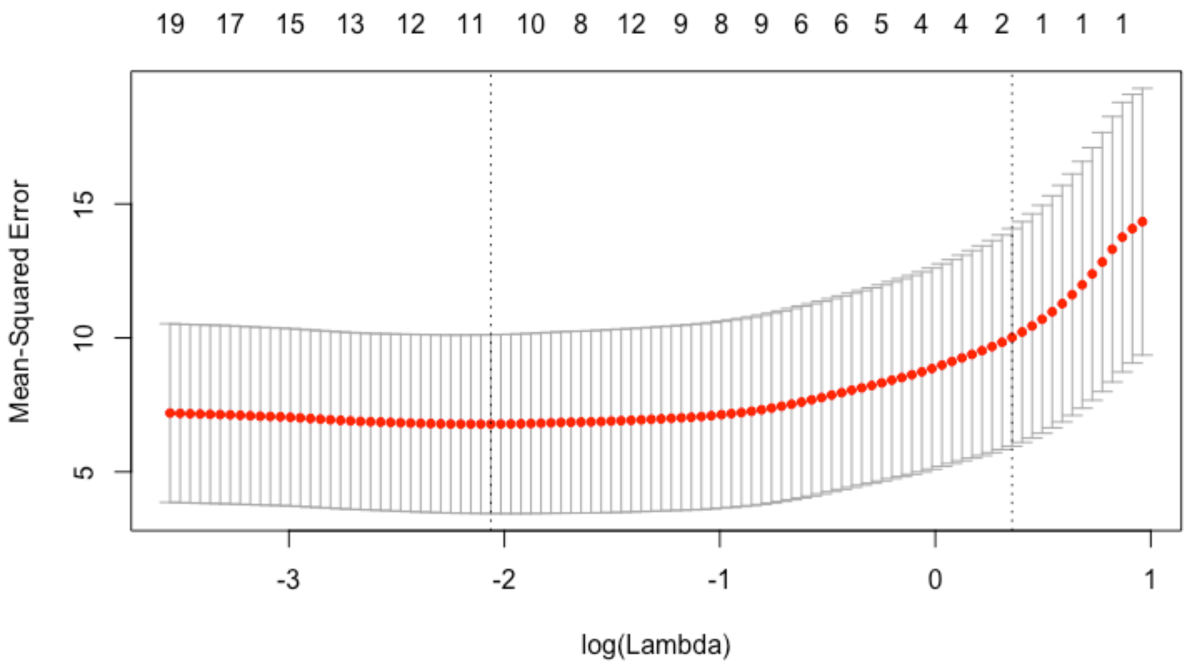


Figure above shows the plot for unregularized linear regression. The dots show the base points.

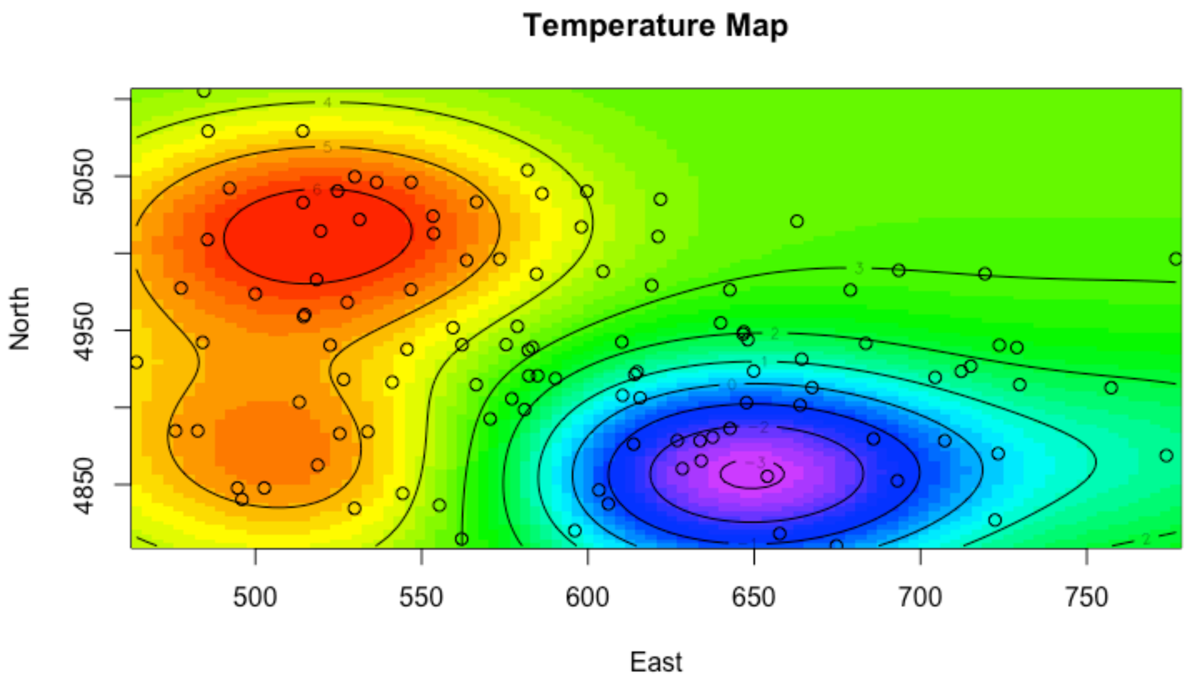
Part 2: Lasso Regression
I used cv.glmnet with alpha value of one, and cross-validated my scales on the best lambda for each model.



The figure above shows the cross-validation results for lasso regression. The best scale ended up being around 40. The optimal mean-squared error was about 4.160916. Multiple runs produced scales within the range {20, 30, 40, 50, 60, 70}.



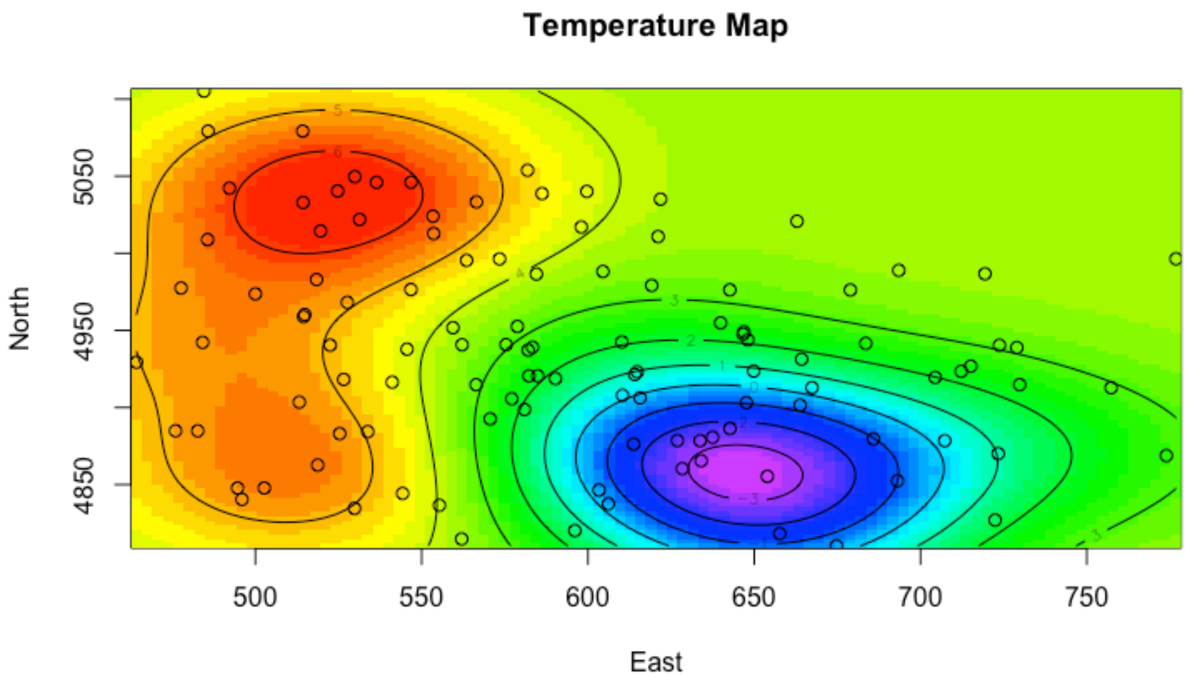
The figure above shows that the lasso regression learning algorithm selected 11 base points. The optimal lambda was 0.2221101. The plot shows that as the number of predictors decreases under 10, the mean-squared errors will exponentially increase.



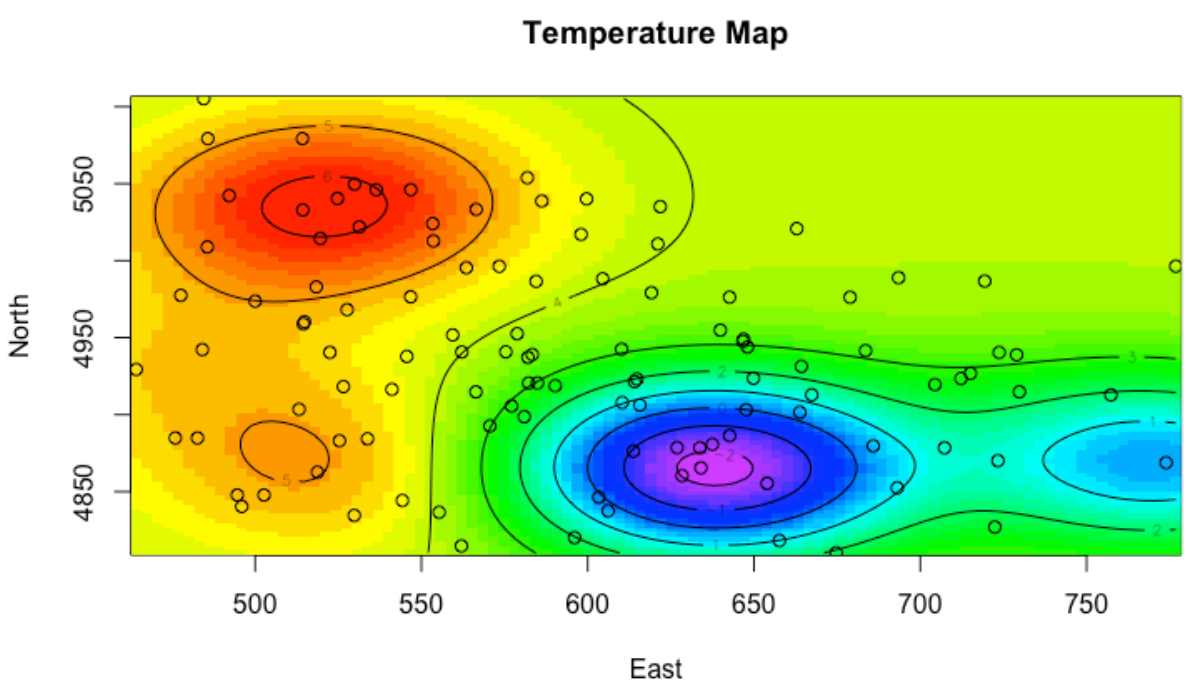
The figure shows considerable less detail than the non-regularized model. This can be attributed to the fact that the number of predictors in the model is considerably less. Temperature variations are more smooth that the non-regularized model so the dips and rises are not as tightly centered. It should also be noted that the mean-squared error increased from the non-regularized model from about 3.8 to 4.16, but running the learning algorithm several times suggest that the variation can be attributed to randomness. (Note: For all the temperature maps I plotted all the bases not just the bases used in the final model).

Part 3: Elastic Net Regression

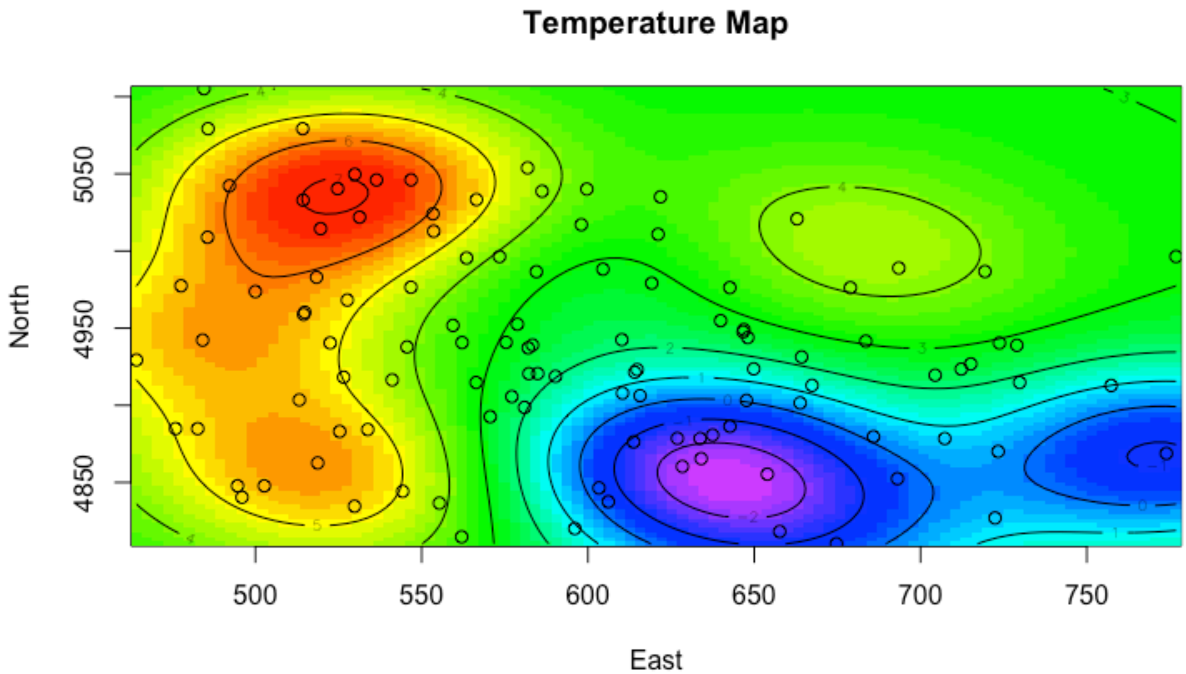
I cross-validated each of my models using alpha values 0.25, 0.50, and 0.75. I found the optimal scales using the same code I used for part 1 and part 2. Similar to part 2, I used cv.glmnet to generate the elastic net models and picked the minimum mean-square regularization parameter.



The figure above shows a temperature map grid produced with an alpha = 0.25 elastic net model. The mean-squared errors using the model I used to produce this grid on the held out dataset was 4.013168. The number of predictors was 35.



The figure above shows a temperature map grid produced with an alpha = 0.50 elastic net model. The mean-squared errors using the model I used to produce this grid on the held out dataset was 4.043463. The number of predictors was 32.



The figure above shows a temperature map grid produced with an alpha = 0.50 elastic net model. The mean-squared errors using the model I used to produce this grid on the held out dataset was 1.951237. The number of predictors was 23.

The different values of alpha did not seem to show significant variation in mean-squared error compared to noise. Increasing the alpha value decreased the the number of predictors that were used to predict the temperatures on the grid and at the base points. It also looks as if the kernel bump functions may be narrowing in scale as the value of alpha increase.