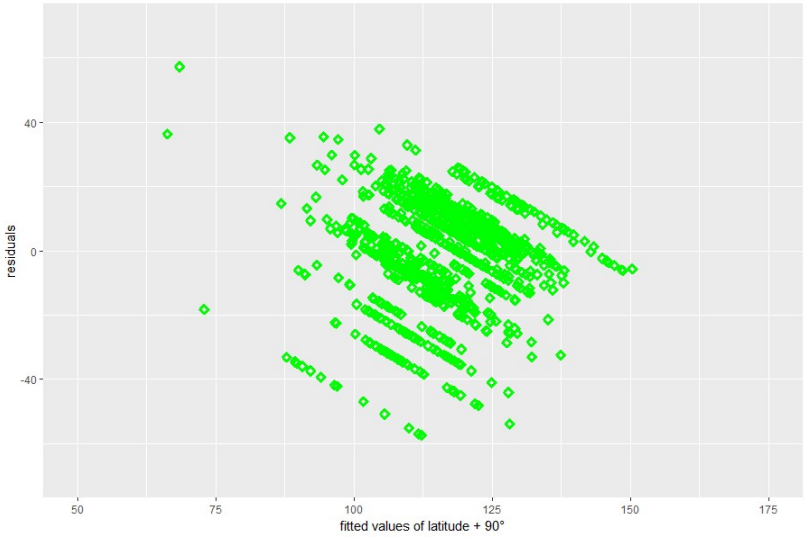
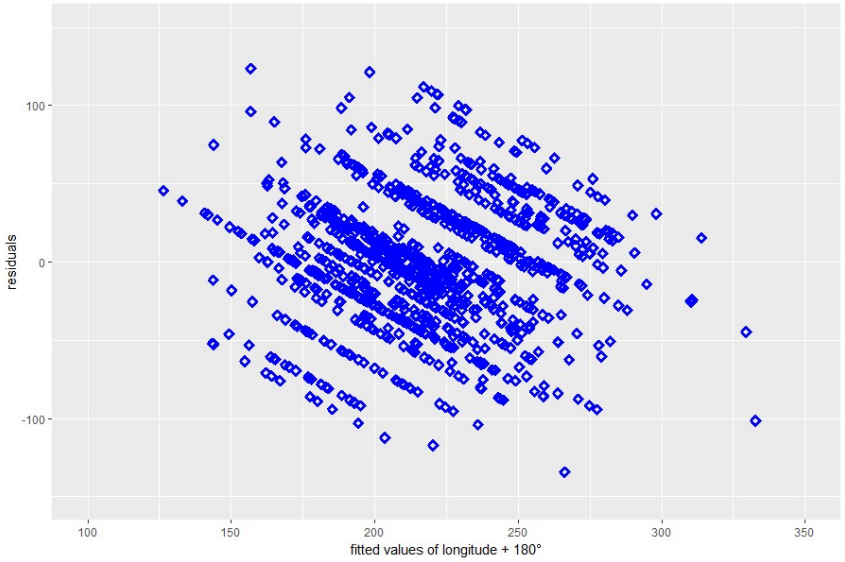
Problem 1. *Linear regression with various regularizers The UCI Machine Learning dataset repository hosts a dataset giving features of music, and the latitude and longitude from which that music originates here. Investigate methods to predict latitude and longitude from these features, as below. There are actually two versions of this dataset. Either one is OK by me, but I think you'll find the one with more independent variables more interesting. You should ignore outliers (by this I mean you should ignore the whole question; do not try to deal with them). You should regard latitude and longitude as entirely independent.*

First, build a straightforward linear regression of latitude (resp. longitude) against features. What is the R-squared? Plot a graph evaluating each regression.

I adjusted latitude values to avoid negative numbers by adding 90 to each latitude value. The graph below is the residuals for the adjusted latitude. The R^2 value of the regression was 0.2928.

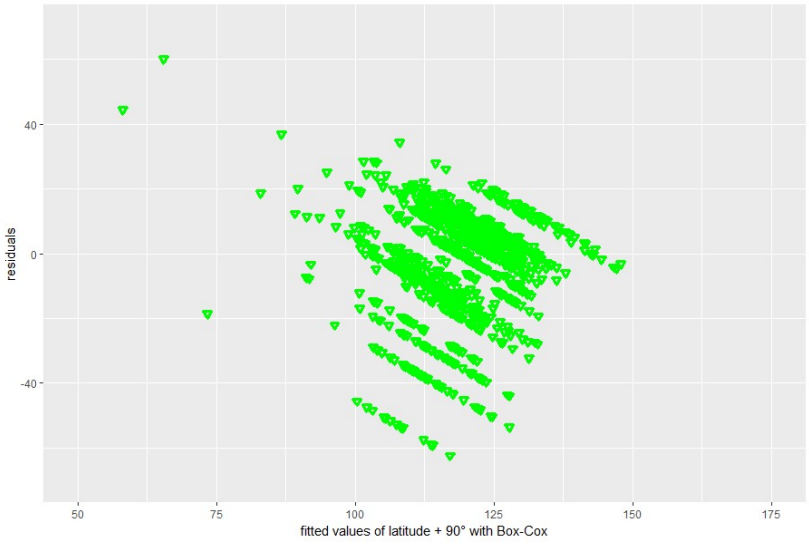


I adjusted longitude values to avoid negative numbers by adding 180 to each longitude value. The graph below is the residuals for the adjusted longitude. The R^2 value of the regression was 0.0091.

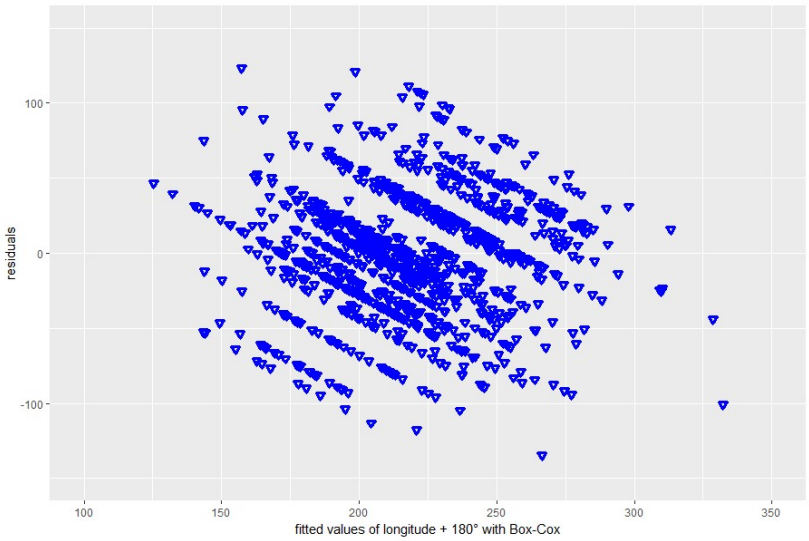


Does a Box-Cox transformation improve the regressions? Notice that the dependent variable has some negative values, which Box-Cox doesn't like. You can deal with this by remembering that these are angles, so you get to choose the origin. why do you say so?

The Box-Cox transformation did not improve the regressions much. For latitude, the transformation resulted in a slight reduction in the R^2 value to 0.2633. The Box-Cox transformation used $\lambda = 3.585859$. The residual plot for the Box-Cox transformed adjusted latitude regression is below.



For longitude, the transformation resulted in a slight reduction in the R^2 value to 0.0079. The Box-Cox transformation used $\lambda = 1.060606$. The residual plot for the Box-Cox transformed adjusted longitude regression is below.

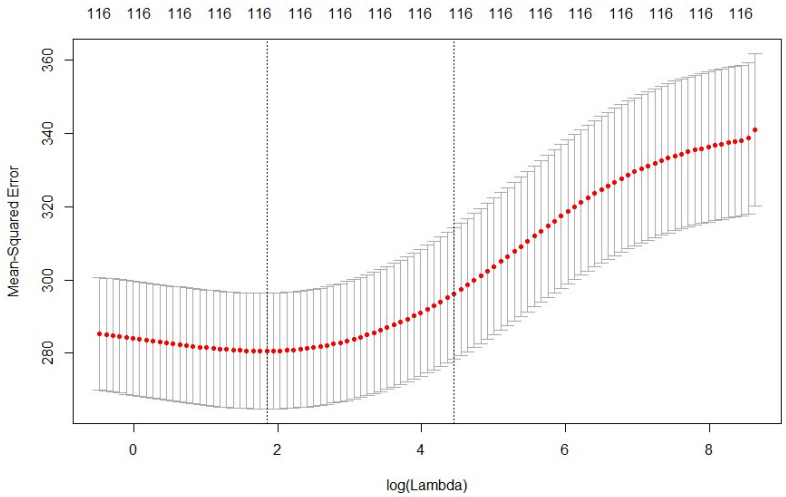


For the rest of the exercise, use the transformation if it does improve things, otherwise, use the raw data. Use glmnet to produce:

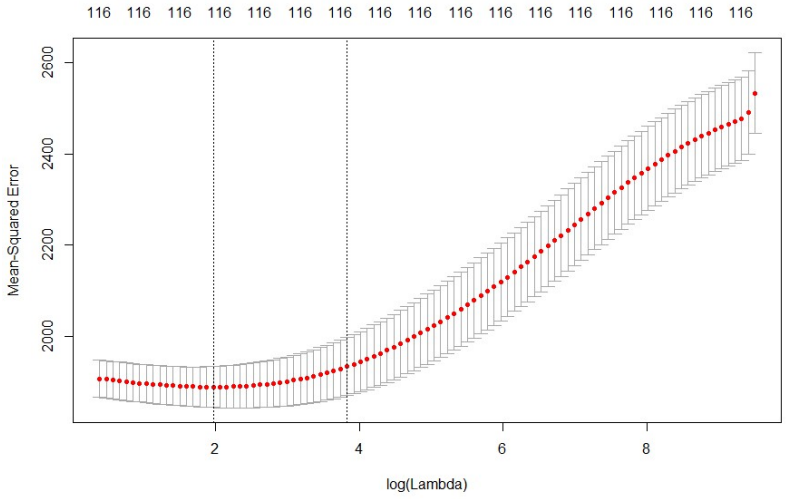
A regression regularized by L2 (equivalently, a ridge regression). You should estimate the regularization coefficient that produces the minimum error. Is the regularized regression better than the unregularized regression?

Because the Box-Cox transformation did not help, I used the raw data (but still adjusted the latitude and longitude to be positive).

Below is the cross-validated prediction error plot (with alpha set to 0 to enable **ridge** regression) for the adjusted **latitude**. The plot shows that regularized regression would be better with a regularization coefficient of 5.77799.



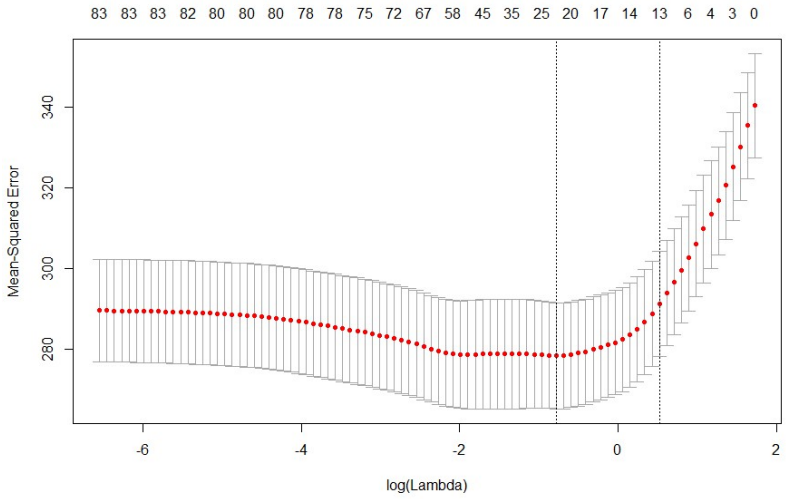
Below is the cross-validated prediction error plot (with alpha set to 0 to enable **ridge** regression) for the adjusted **longitude**. The plot shows that regularized regression would be better with a regularization coefficient of 7.160057.



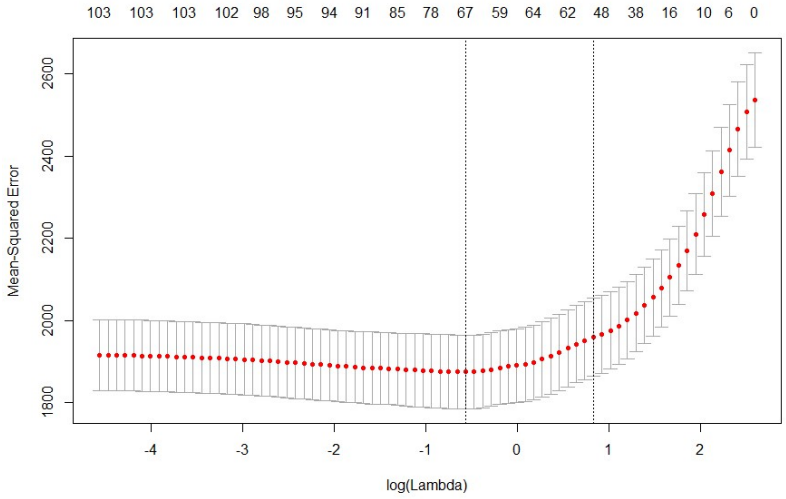
A regression regularized by L1 (equivalently, a lasso regression). You should estimate the regularization coefficient that produces the minimum error. How many variables are used by this regression? Is the regularized regression better than the unregularized regression?

Because the Box-Cox transformation did not help, I used the raw data (but still adjusted the latitude and longitude to be positive).

Below is the cross-validated prediction error plot (with alpha set to 0 to enable **lasso** regression) for the adjusted **latitude**. The plot shows that regularized regression would be better with a regularization coefficient of 0.5025401. The plot indicates that 21 variable are used by that regression.



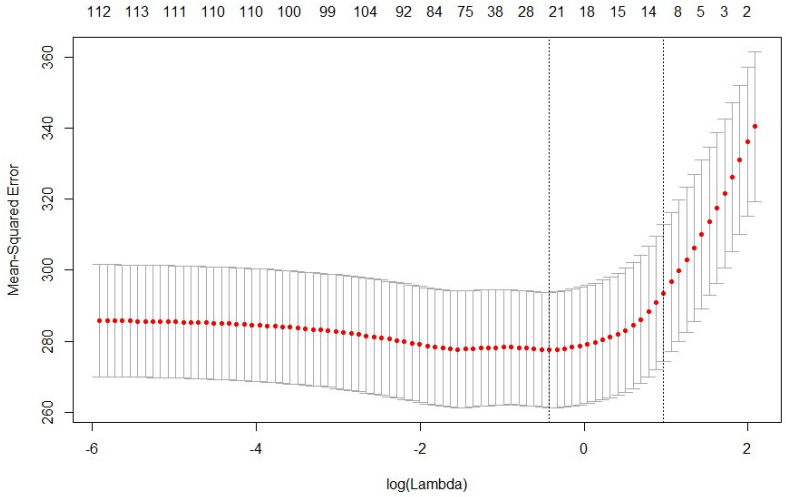
Below is the cross-validated prediction error plot (with alpha set to 0 to enable **lasso** regression) for the adjusted **longitude**. The plot shows that regularized regression would be better with a regularization coefficient of 0.5674223. The plot indicates that 67 variable are used by that regression.



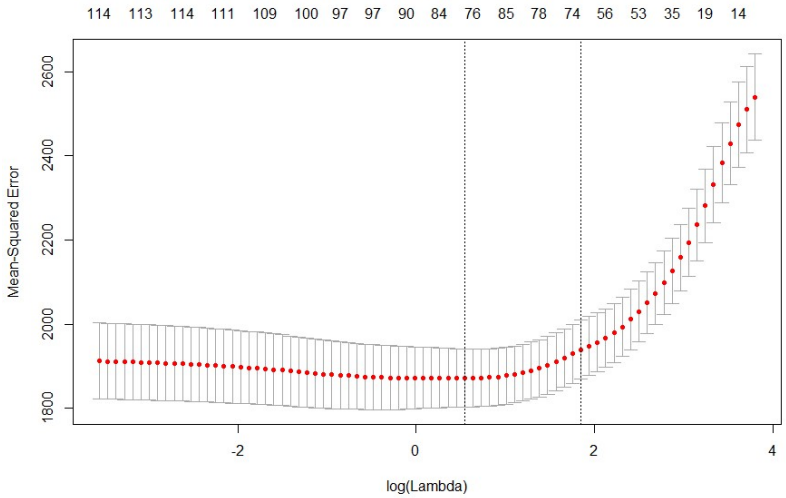
A regression regularized by elastic net (equivalently, a regression regularized by a convex combination of L1 and L2). Try three values of alpha, the weight setting how big L1 and L2 are. You should estimate the regularization coefficient that produces the minimum error. How many variables are used by this regression? Is the regularized regression better than the unregularized regression?

I used alpha values of 0.3, 0.5 and 0.7 in obtaining regressions regularized by elastic net.

For latitude, the regularization coefficient that produces the minimum error is with an alpha value of 0.7 (better than ridge or lasso). Below is the cross-validated prediction error plot (with alpha set to 0.7) for the adjusted **latitude**. The plot shows that regularized regression would be better with a regularization coefficient of 0.7179145. The plot indicates that 21 variable are used by that regression.

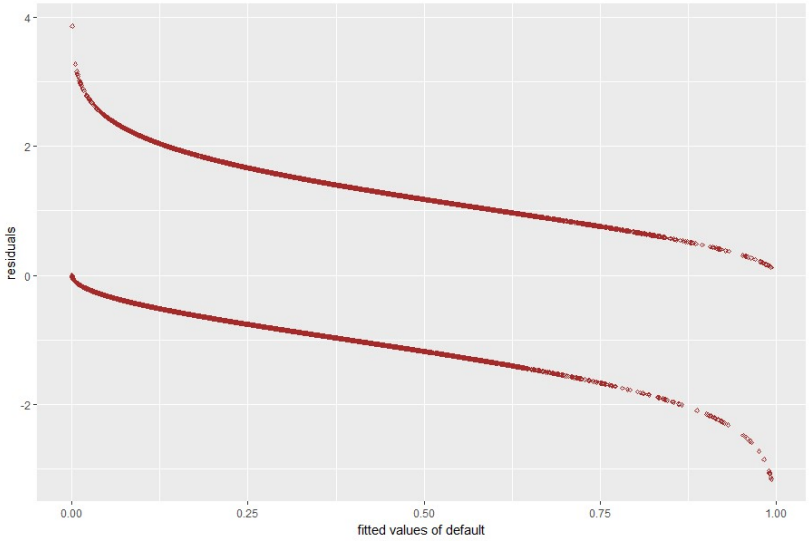


For longitude, the regularization coefficient that produces the minimum error is with an alpha value of 0.3. Below is the cross-validated prediction error plot (with alpha set to 0.3) for the adjusted **longitude**. The plot shows that regularized regression would be better with a regularization coefficient of 1.72338. The plot indicates that 78 variable are used by that regression.

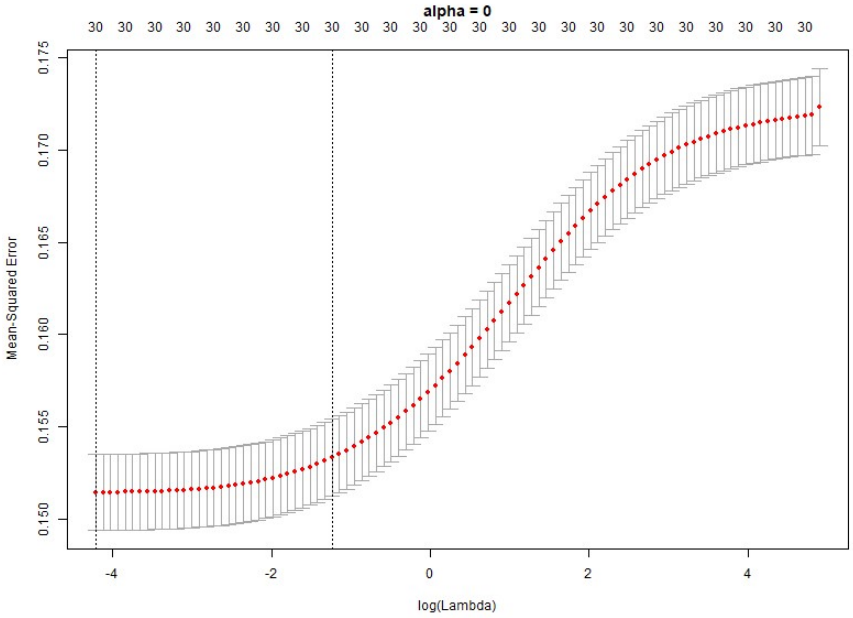


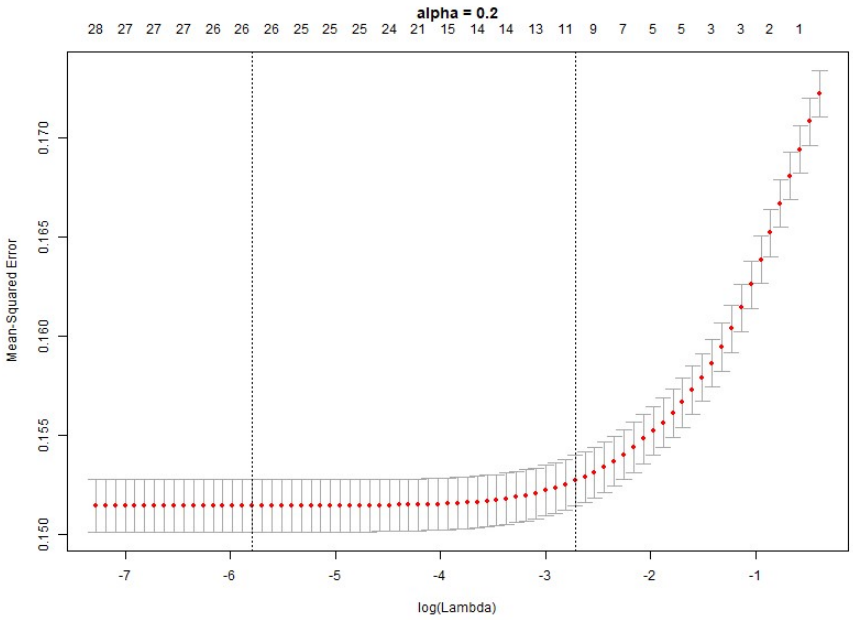
Problem 2. *Logistic regression The UCI Machine Learning dataset repository hosts a dataset giving whether a Taiwanese credit card user defaults against a variety of features here. Use logistic regression to predict whether the user defaults. You should ignore outliers, but you should try the various regularization schemes we have discussed.*

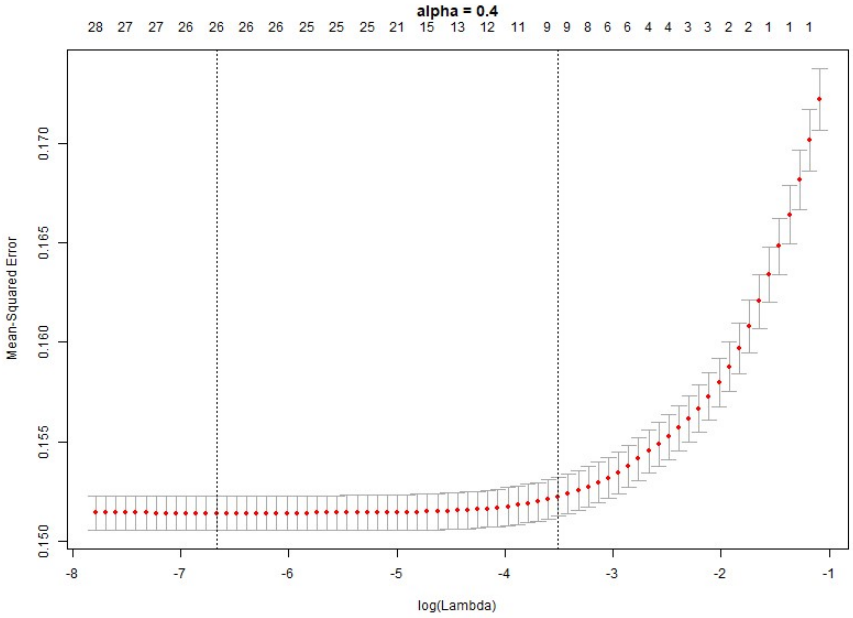
Below is the plot of residuals for the logistic regression without any regularizer. I used the Hosmer and Lemeshow goodness of fit (GOF) test that provided the results: X-squared = 847.72, df = 8, p-value < 2.2e-16 which indicates a poor regression. I used one hot encoding for the categorical variables like, sex, education and marriage.

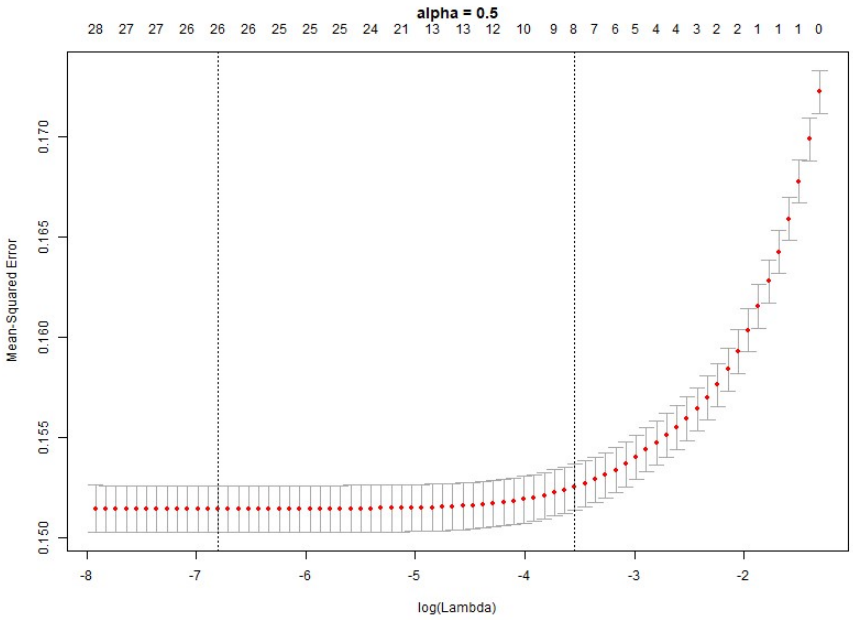


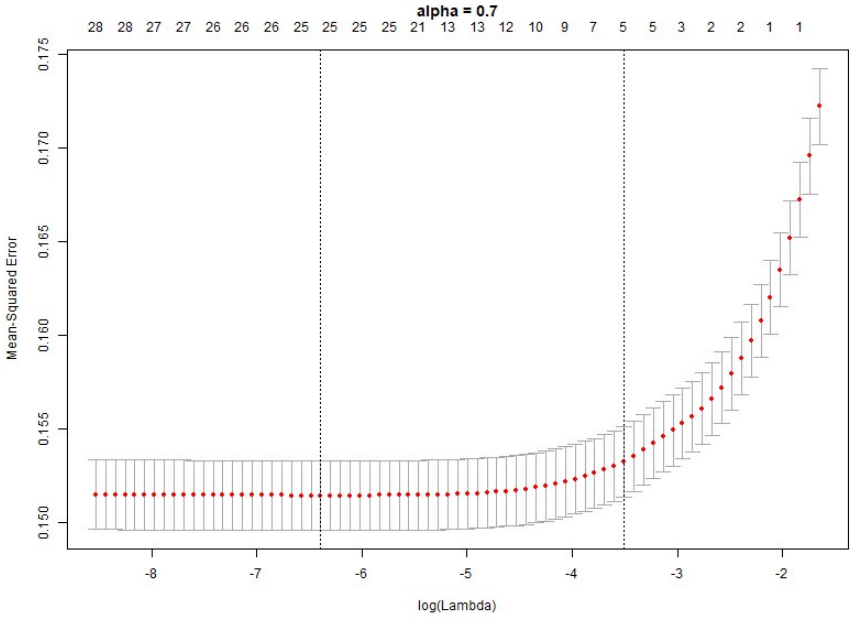
I then attempted to apply a regularizer (ridge, elastic net and lasso) by using cv.glmnet and adjusting alpha from 0 to 1 in increments of 0.1. The best regularizer regression occurred with alpha = 0.5. The accuracy was 0.8485904. The plot below shows that regularized regression would be better than the un-regularized regression with a regularization coefficient of 0.002136648. The following plots are example plots from evaluating the various cv.glmnet results. The plot with alpha = 0.5 (page 8) indicates that the number of coefficients in the regression would be 25.

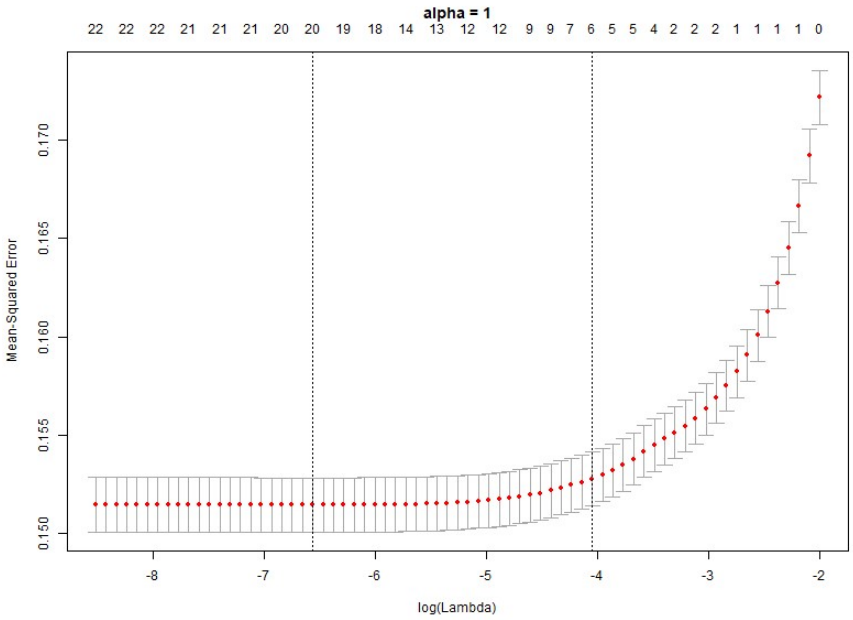

alpha = 0
30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30
Mean-Squared Error
0.150
0.155
0.160
0.165
0.170
0.175
-4
-2
0
2
4
log(Lambda)


alpha = 0.2
28 27 27 27 26 26 26 25 25 25 24 21 15 14 14 13 11 9 7 5 5 3 3 2 1
Mean-Squared Error
0.150
0.155
0.160
0.165
0.170
-7
-6
-5
-4
-3
-2
-1
log(Lambda)


alpha = 0.4
28 27 27 26 26 26 26 25 25 25 21 15 13 12 11 9 9 8 6 6 4 4 3 3 2 2 1 1 1
Mean-Squared Error
0.150
0.155
0.160
0.165
0.170
-8
-7
-6
-5
-4
-3
-2
-1
log(Lambda)


alpha = 0.5
28 27 27 26 26 26 25 25 25 24 21 13 13 12 10 9 8 7 6 5 4 4 3 2 2 1 1 1 0
Mean-Squared Error
0.150
0.155
0.160
0.165
0.170
-8
-7
-6
-5
-4
-3
-2
log(Lambda)


alpha = 0.7
28 28 27 27 26 26 26 25 25 25 25 21 13 13 12 10 9 7 5 5 3 2 2 1 1
Mean-Squared Error
0.150 0.155 0.160 0.165 0.170 0.175
-8 -7 -6 -5 -4 -3 -2
log(Lambda)


alpha = 1
22 22 22 21 21 21 20 20 19 18 14 13 12 12 9 9 7 6 5 5 4 2 2 2 1 1 1 1 0
Mean-Squared Error
0.150 0.155 0.160 0.165 0.170
-8 -7 -6 -5 -4 -3 -2
log(Lambda)