CS 498 AML HW6

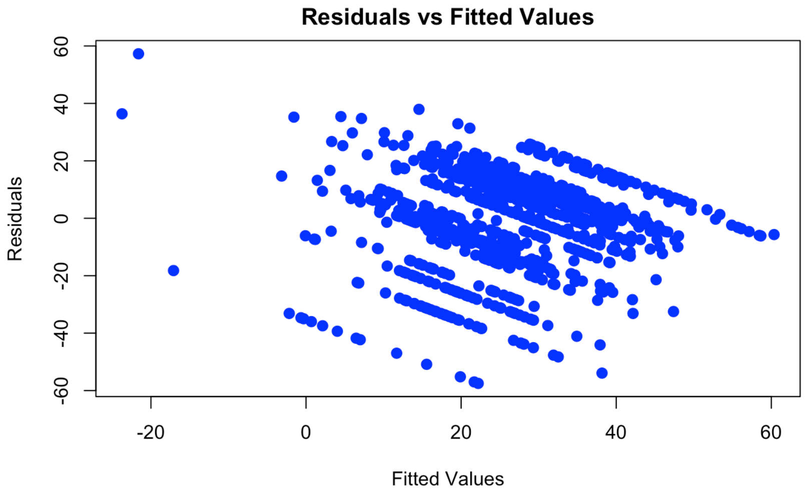
Songwei Feng(songwei3)
Yuxi Gu(yuxigu2)
Xuanyi Zhu(xzhu42)

We used R as the programming language for this homework.

Problem 1

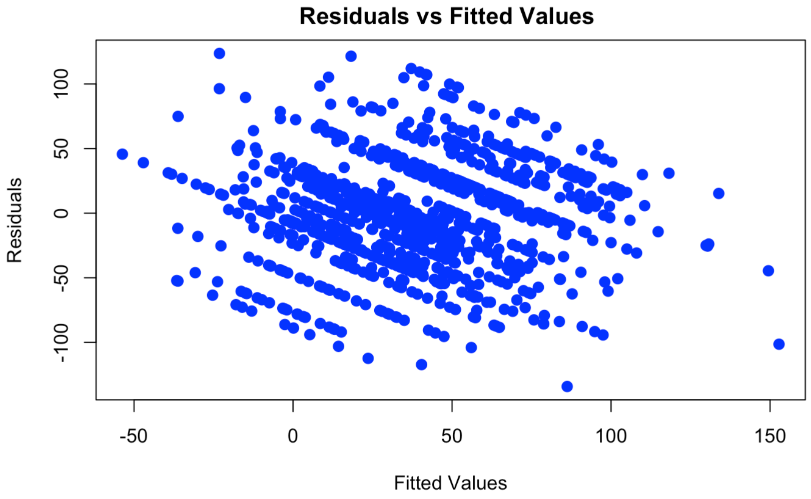
For this problem, we used the data with 116 independent variables, and took 2 other variables(latitude and longitude) as dependent variables.

1. First, we built a straightforward linear regression of latitude against all other independent variables, and plot the residuals against the fitted values:



Here we can get the **R-squared value as 0.2928092** from the summary() function and observe a non-random pattern in residuals from the plot above, which means the regression doesn't fit the data well.

Then we built another linear regression of longitude against all other independent variables, and plot the residuals against the fitted values:



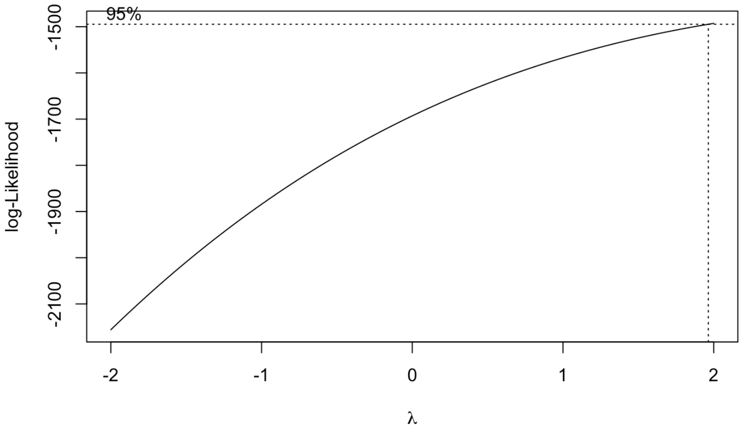
The **R-squared value is 0.3645767**, which is larger than the regression on latitude, but is still not good enough to explain the variances and it also got the non-random pattern in residuals vs fitted values plot.

2. Then we performed Box-Cox transformation to see whether it can improve the regressions.

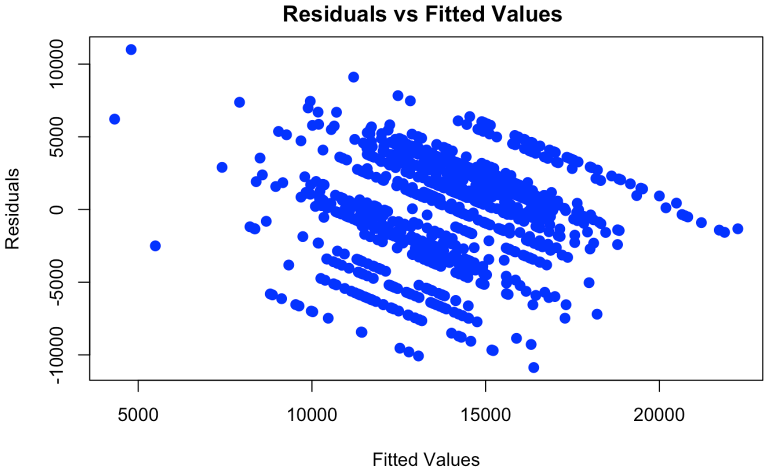
First we transform the negative values in dependent variables to fit into the Box-Cox, since these are angles, the range of the latitude is [-90,90], and the range of the longitude is [-180,180], we deal with the negative values in the following ways:

1) new latitude = latitude + 90, thus the range of new latitude is **[0,180]**;
2) new longitude = longitude + 180, thus the range of new longitude is **[0,360]**.

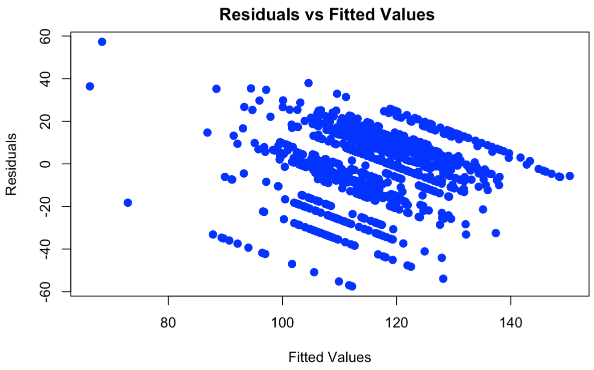
After the transformation, we did the Box-Cox transformation on the latitude against 116 features using the boxcox() function from the MASS package and took the **lambda = 2** which results to the largest log-likelihood.



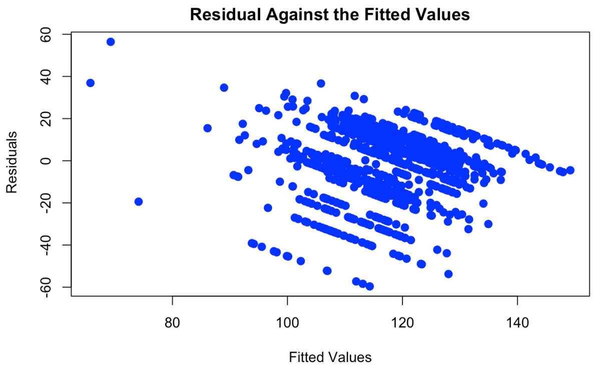
Then we rebuilt the linear regression model by taking transformed latitude = new latitude ^ 2. And we plot the residuals:



In order to compare the two regression, we plot the residuals vs fitted values and computed the R-squared value as well as the MSE in the same scale, and got the following results:



Original

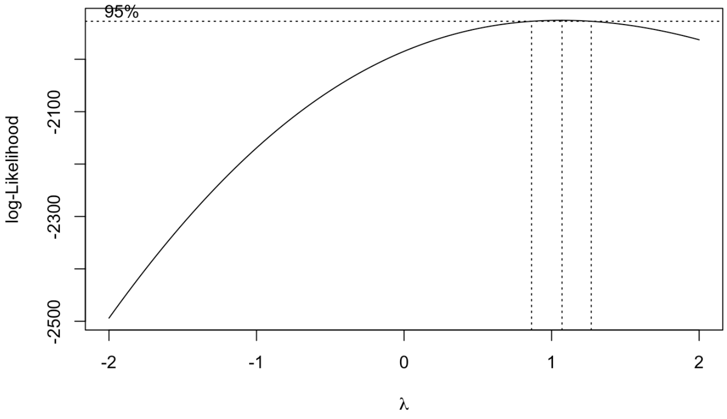


Rebuilt (same scale)

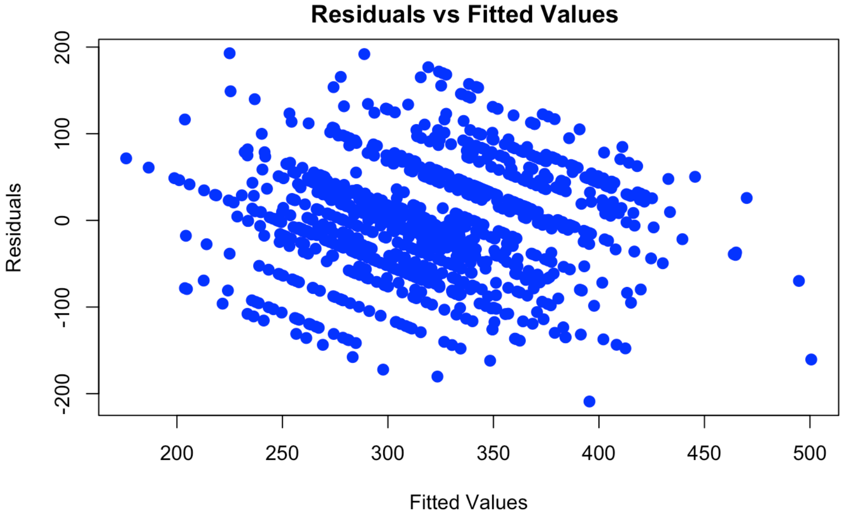
	MSE	R-Squared Value
Original	240.7481	0.2928092
After Box-Cox	243.9985	0.2832615

From the above results we can see that even though the residuals plots only have small differences, the R-Squared Value decreased after the Box-Cox transformation, and the MSE increased, which means the Box-Cox transformation cannot improve the regression.

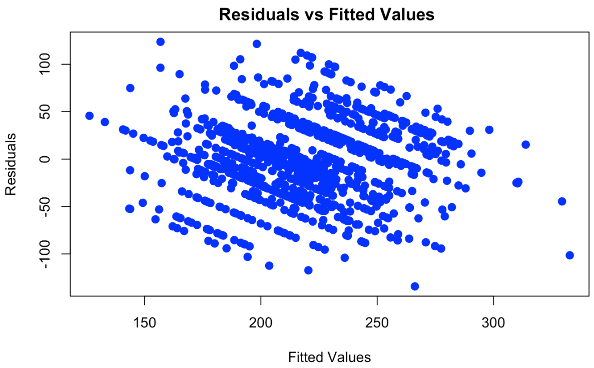
Then we performed the Box-Cox transformation on the longitude against 116 features using the same function and got the **lambda = 1.07070707070707**.



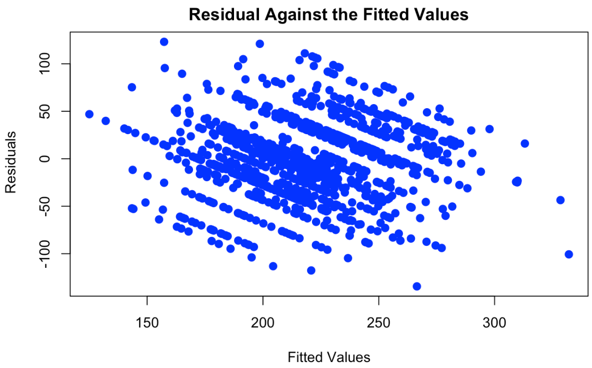
Then we rebuilt the linear regression model by taking transformed longitude = new longitude ^ 1.07070707070707. And we plot the residuals:



In order to compare the two regression, we plot the residuals vs fitted values and computed the R-squared value as well as the MSE in the same scale, and got the following results:



Original



Rebuilt (same scale)

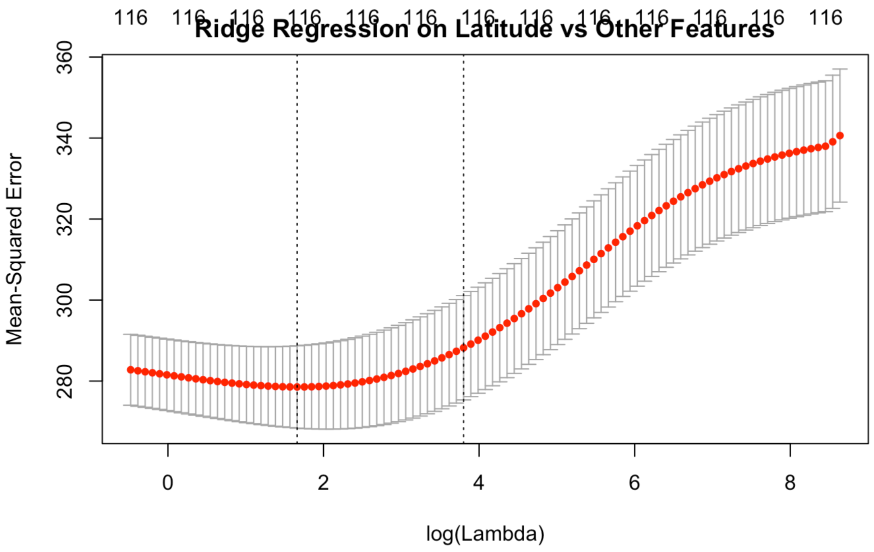
	MSE	R-Squared Value
Original	1613.819	0.3645767
After Box-Cox	1613.802	0.3645836

From the above results we can see that two residuals plots have small differences, the R-Squared Value increased a bit after the Box-Cox transformation, and the MSE decreased, which means the Box-Cox transformation did improve the regression.

Thus, we concluded with that even though there exists improvement when regress on longitude data, the Box-Cox transformation didn't improve the model when apply to the latitude data, and we decided to use the raw data to do the rest of exercise.

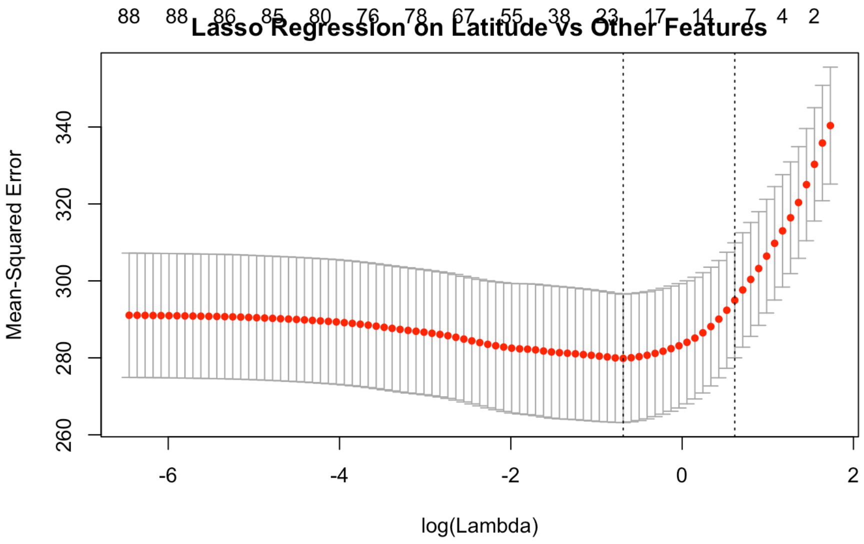
3. For this part, we first used latitude against independent variables to do the un-regularized and regularized regression by using the cv.lm() function in DAAG package and cv.glmnet() function in glmnet package. We performed the regressions with 10-folds cross-validation.

1) Ridge regression



The minimum mean-squared error we got is **279** when lambda = **5.26**, which is smaller than the original mean-squared error **288** as we calculated before, thus the regularized regression is better than the un-regularized regression.

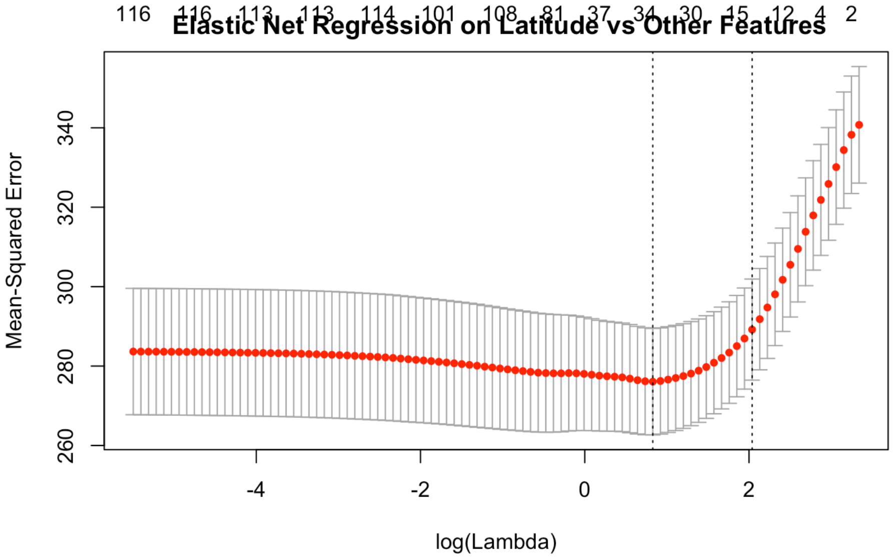
2) Lasso regression



The minimum mean-squared error we got is **280** when lambda = **0.503**, and **22** variables were used by this regression. By comparing to the mean-squared error of un-regularized regression, we can see that this regression regularized model is better than the un-regularized one.

3) Elastic Net regression

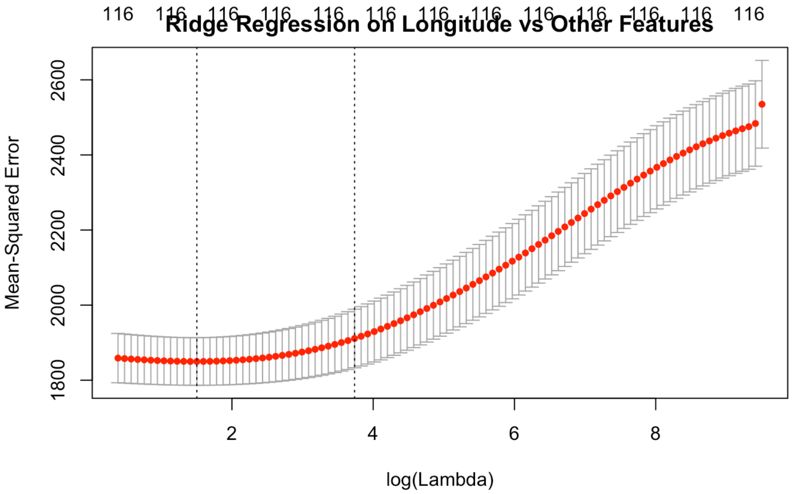
Here we tried alpha = **0.2, 0.5, 0.8**, and got the minimum mean-squared error = **276** when alpha = **0.2** and lambda = **2.29**.



The number of used variables is **34**. Since the minimum mean-squared error much smaller than the un-regularized regression, we can conclude that the regularized regression is better than the un-regularized regression.

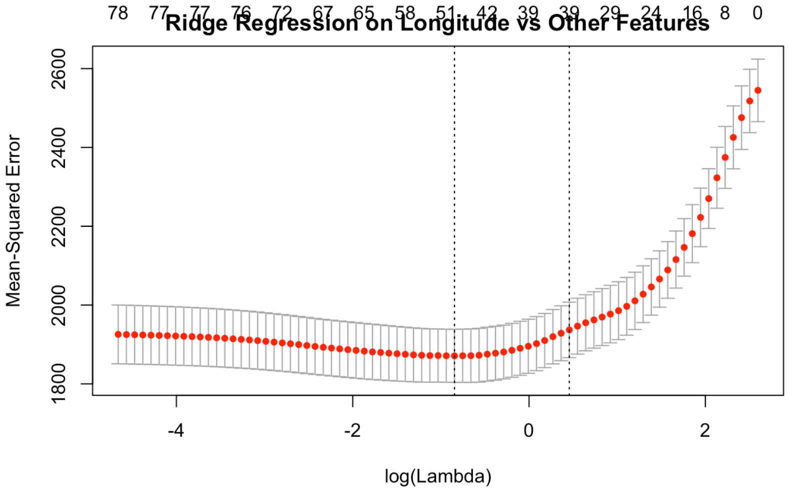
Then we performed the un-regularized regression and three regularized regression on longitude against independent variables, and got the following results:

1) Ridge regression



The minimum mean-squared error we got is **1850** when lambda = **4.5**, which is smaller than the original mean-squared error **1937** as we calculated before, thus the regularized regression is better than the un-regularized regression.

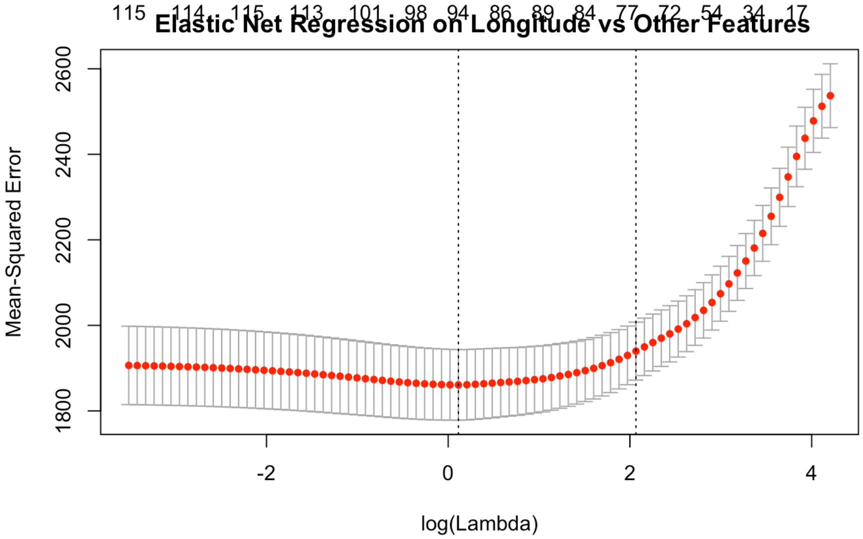
2) Lasso regression



The minimum mean-squared error we got is **1871** when lambda = **0.429**, and **51** variables were used by this regression. By comparing to the mean-squared error of un-regularized regression, we can see that this regression regularized model is better than the un-regularized one.

3) Elastic Net regression

Here we tried alpha = **0.2, 0.5, 0.8**, and got the minimum mean-squared error = **1861** when alpha = **0.2** and lambda = **1.12**.



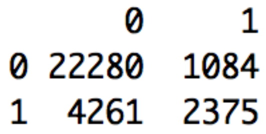
The number of used variables is **94**. Since the minimum mean-squared error much smaller than the un-regularized regression, we can conclude that the regularized regression is better than the un-regularized regression.

Note: Problem 2 is in the next page.

Problem 2

After reading all data into a data frame, we first use a function called **factor** to transform some columns of the data frame, such as SEX and EDUCATION, from numerical data into categorical data. Then we built a logistic model using **glm** function in R. As for the family type parameter in the glm function, we used "binomial" here. And we used a threshold of 0.5 here because that seemed to give us a better result after several trials. And using a threshold of 0.5 here means that any predicted value greater than 0.5 will be classified as 1, otherwise it will be classified as 0.

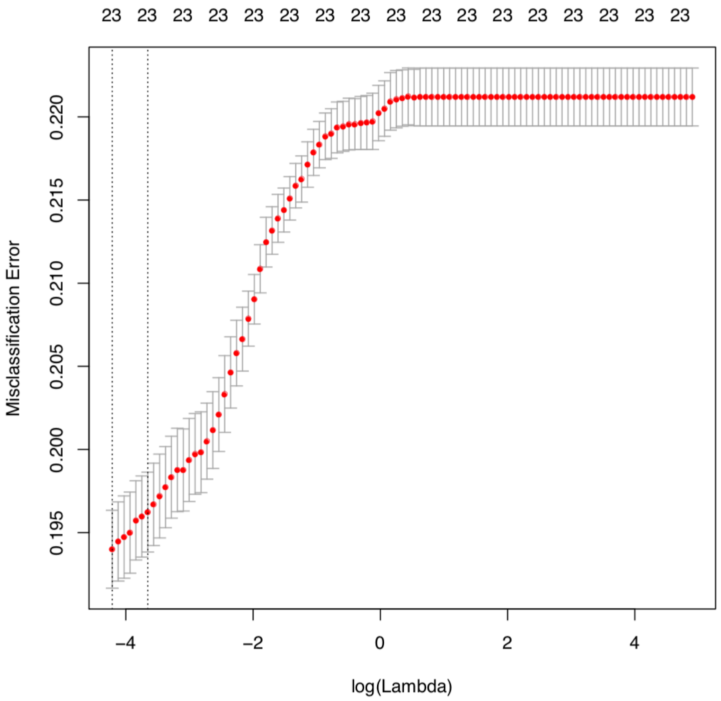
The accuracy of the logistic model we built was **82.18333%**. And the confusion matrix looks like the following:



```
      0     1
0 22280  1084
1  4261  2375
```

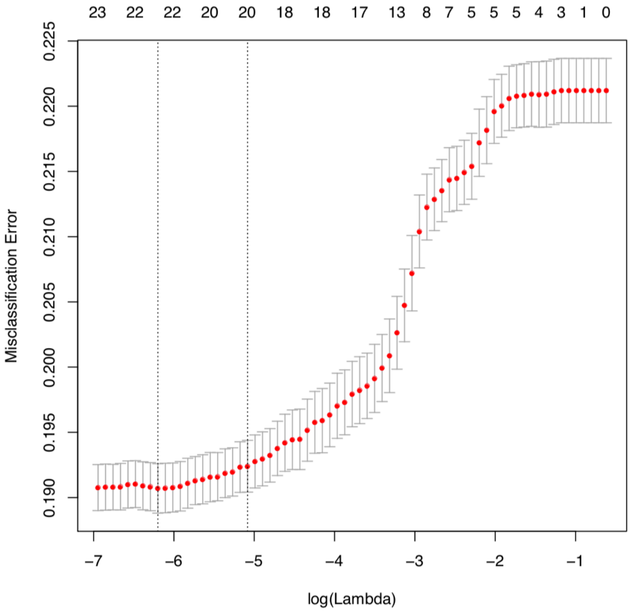
We then tried the various regularization schemes discussed in lecture. Here we tried ridge, lasso and elastic net regularization method by using different values of alpha. And we used the **cv.glmnet** function in R to build models with different regularization schemes by passing different values into the alpha parameter. In addition, we assigned the nfolds parameter in the cv.glmnet function to be 10.

Ridge Regression (alpha = 0)
We got an accuracy of **80.6%** in the ridge regression model. And the Misclassification Error vs Log(Lambda) plot is showed in the following figure:



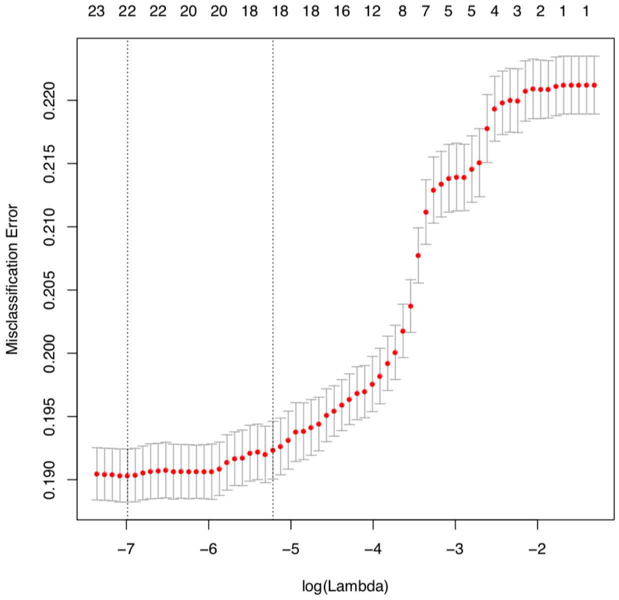
Elastic Net Regression (alpha = 0.25)

We got an accuracy of **80.93%** in the elastic net model, with alpha=0.25. And the Misclassification Error vs Log(Lambda) plot is showed in the following figure:



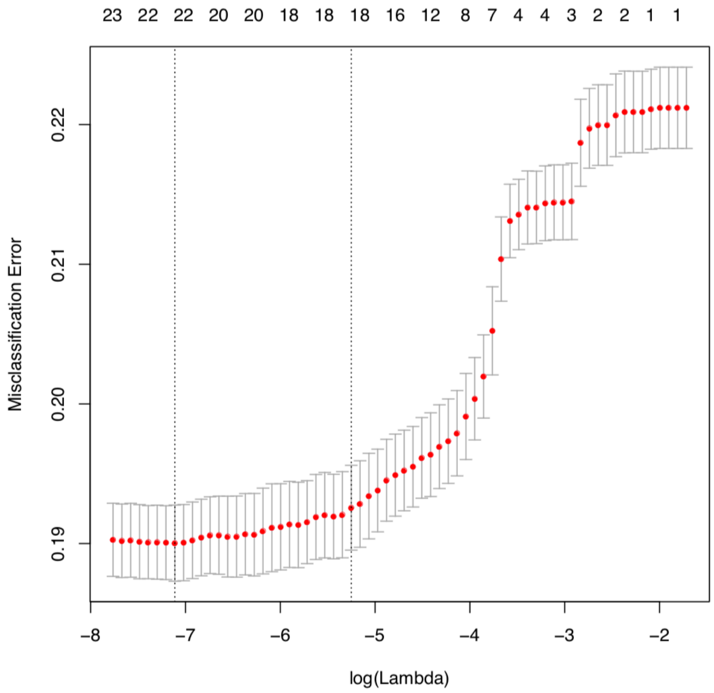
Elastic Net Regression (alpha = 0.5)

We got an accuracy of **80.96667%** in the elastic net model, with alpha=0.5. And the Misclassification Error vs Log(Lambda) plot is showed in the following figure:



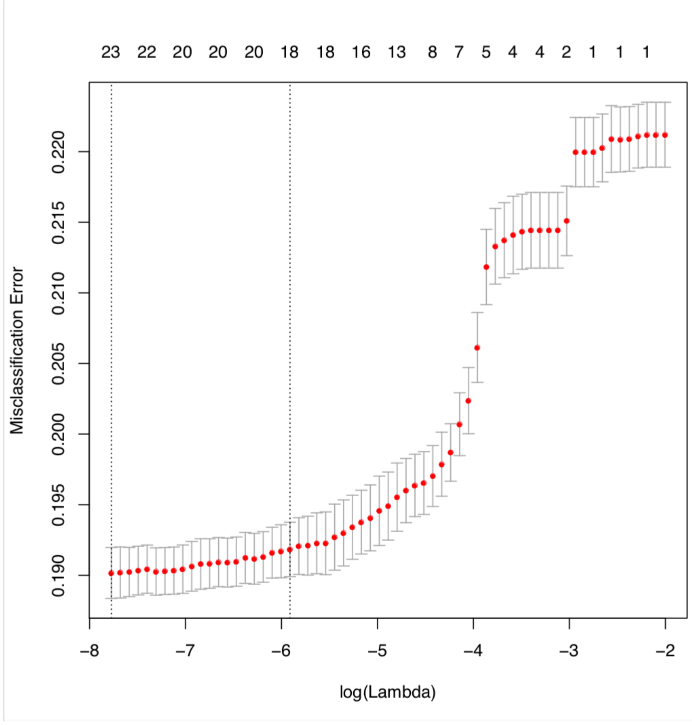
Elastic Net Regression (alpha = 0.75)

We got an accuracy of **80.99667%** in the elastic net model, with alpha=0.5. And the Misclassification Error vs Log(Lambda) plot is showed in the following figure:



Lasso Regression (alpha = 1)

We got an accuracy of **80.98333%** in the elastic net model, with alpha=0.5. And the Misclassification Error vs Log(Lambda) plot is showed in the following figure:



Overall, among all regression models with regularization schemas, elastic net regression and lasso regression seemed to have better accuracy than ridge regression.

However, we found out that logistic regression without any regularization seemed to have the best accuracy in this problem. And the differences of accuracy between the unregularized model and regularized models are significant compared to the differences of accuracy between all different regularized models.