HW6

Xiang Gao

1.

A. build a straightforward regression

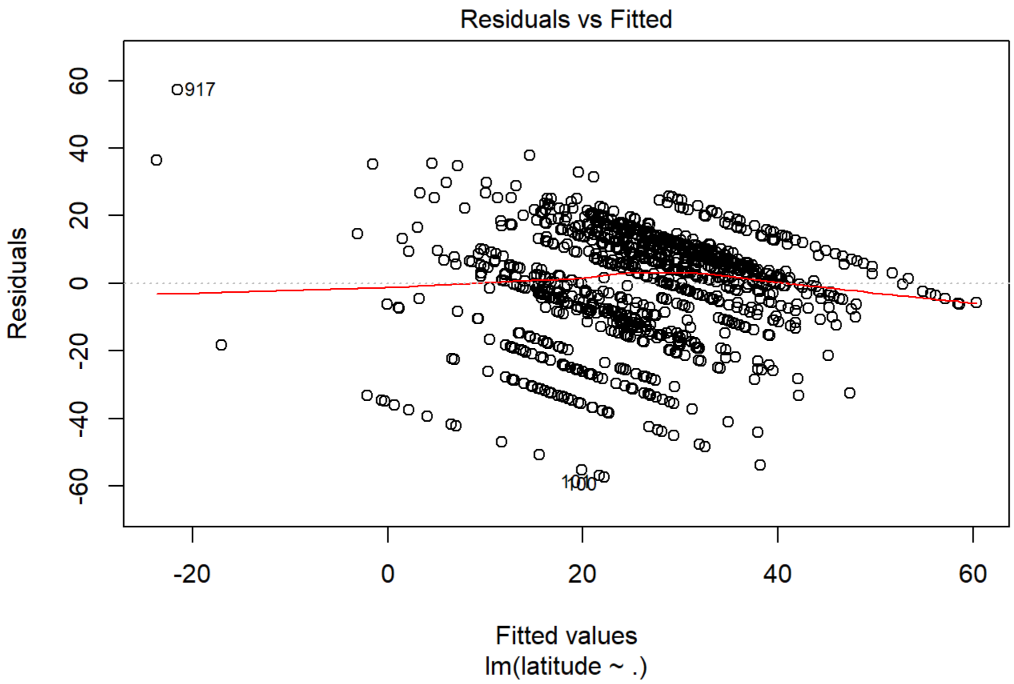
latitude model r squared

```
## [1] 0.2928092
```

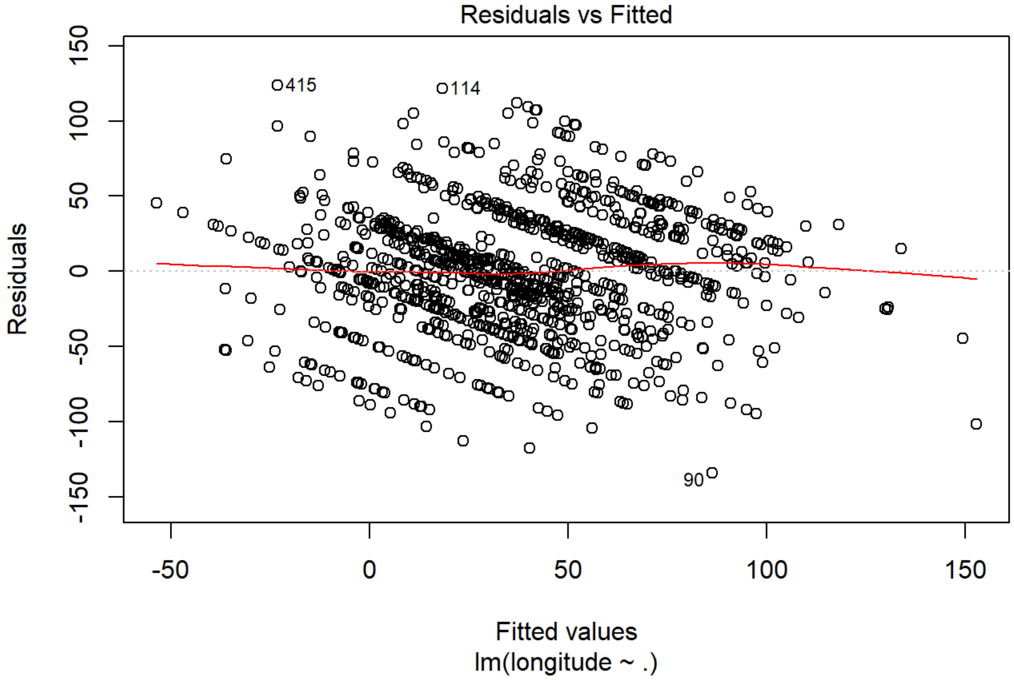
longitude model r squared

```
## [1] 0.3645767
```

latitude model residual plot

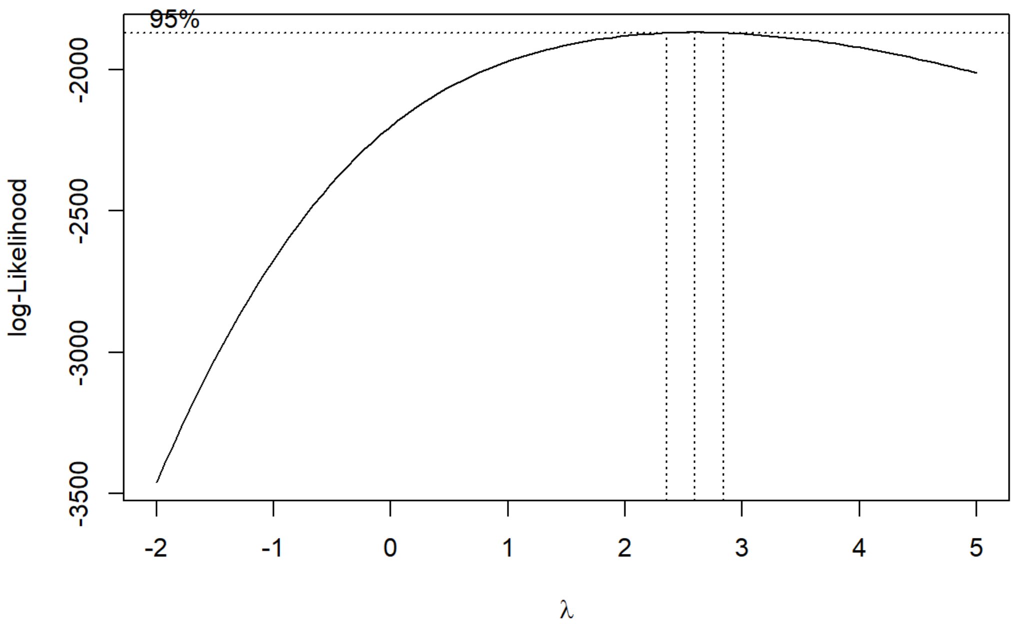


longitude model residual plot

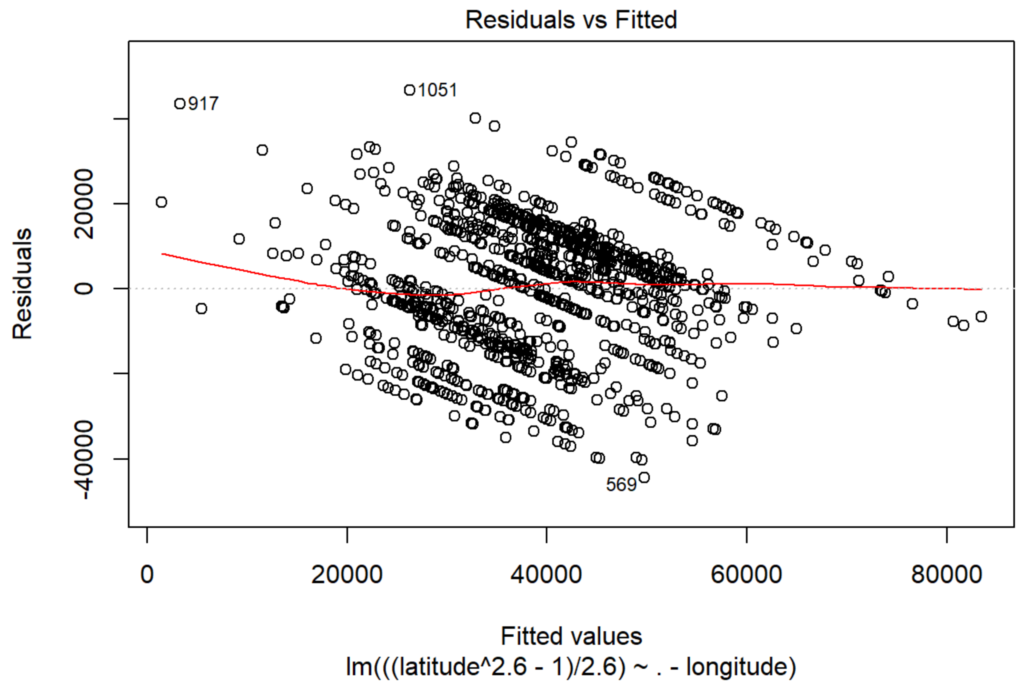


From the residual plot, the longitude model is more scattered and the r squared proved that it has better fit.

B. boxcox transformation for latitude model



This suggest lambda equals 2.6. I will fit the model again with this transformation. Thus we get the residual plot and r squared as follows.

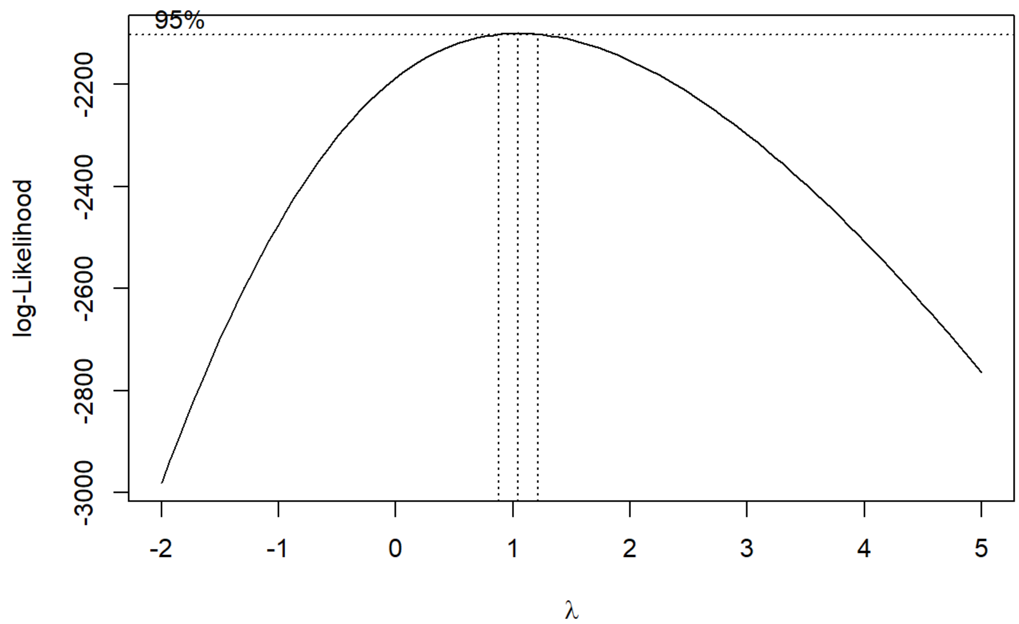


r-squared value

```
## [1] 0.3235732
```

Boxcox transformation for latitude model shows more scatter than straightfoward regression. So it helps improve latitude model. By checking the r squared also proves my conclusion.

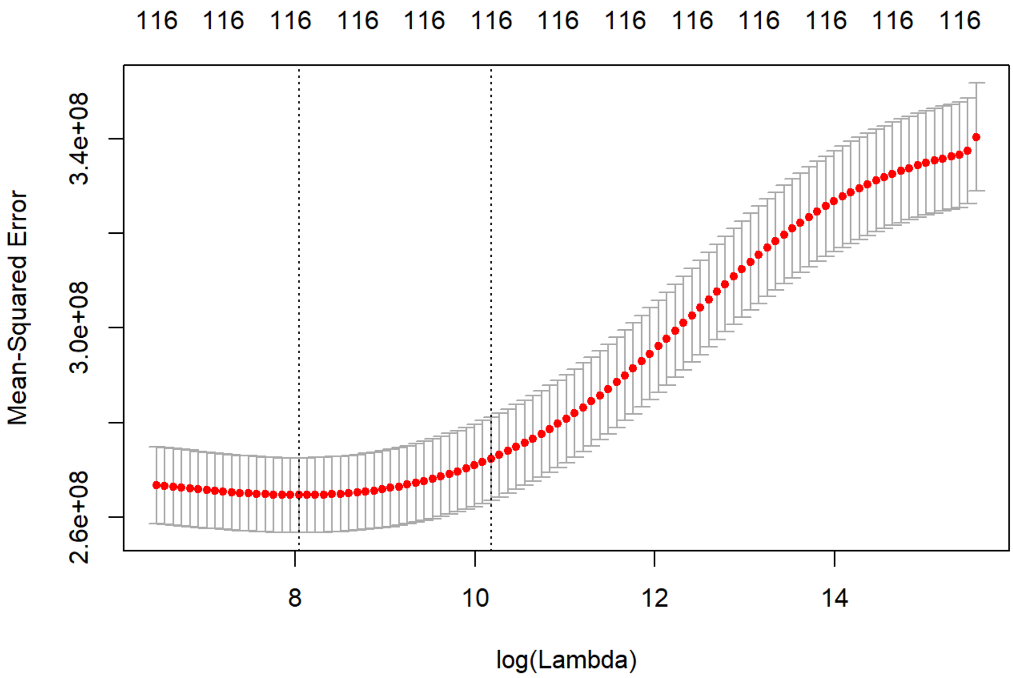
Boxcox for longitude model



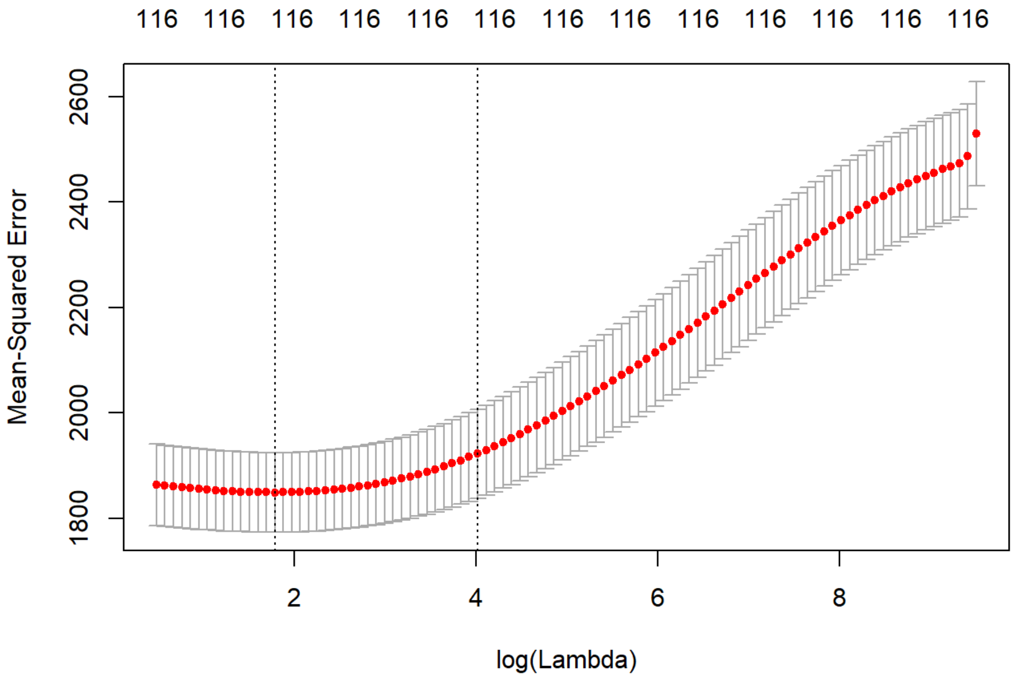
This suggest lambda equals to 1 which is not a transformation. It would not make the model fit better. The errors will remain the same. So it doesn't imporve the regressions for longitude model.

C Apply box-cox transformation for latitude model, longitude model will still use original data.

I.L2 ridge regression for latitude model



I.L2 ridge regression for longitude model



lambda that produces the minimun error for latitude and longitude model

```
## [1] 3104.568
```

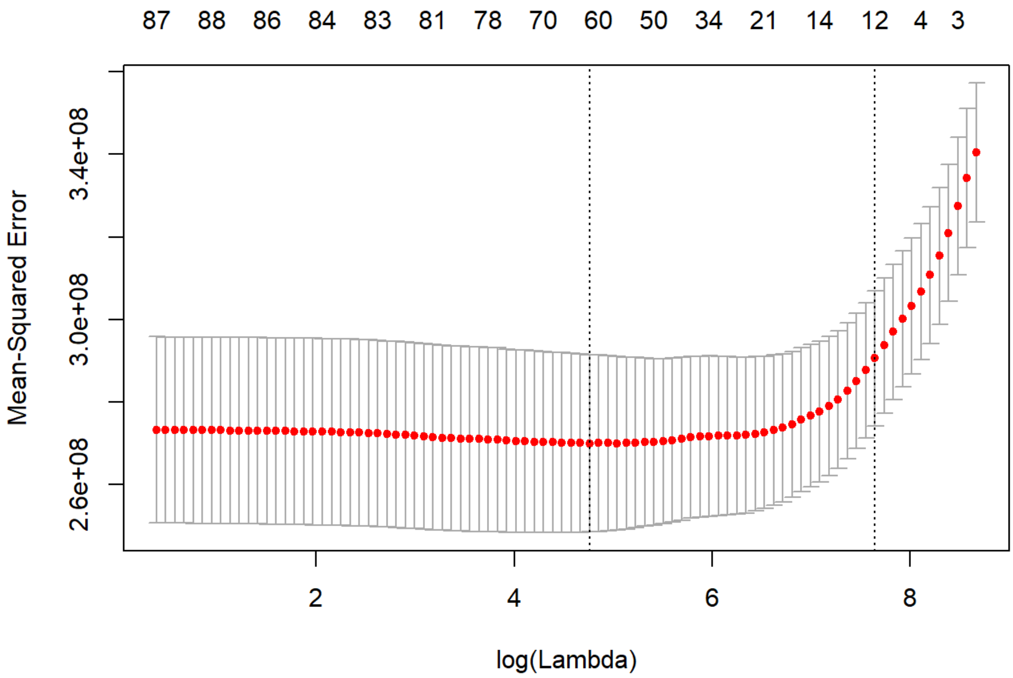
```
## [1] 5.944405
```

For each plot, imagin on the far left where all variables are used and the lambda closes to zero, can be regarded as unregularized. So the error of the unregularized model is closes to the left red dot of the plot.

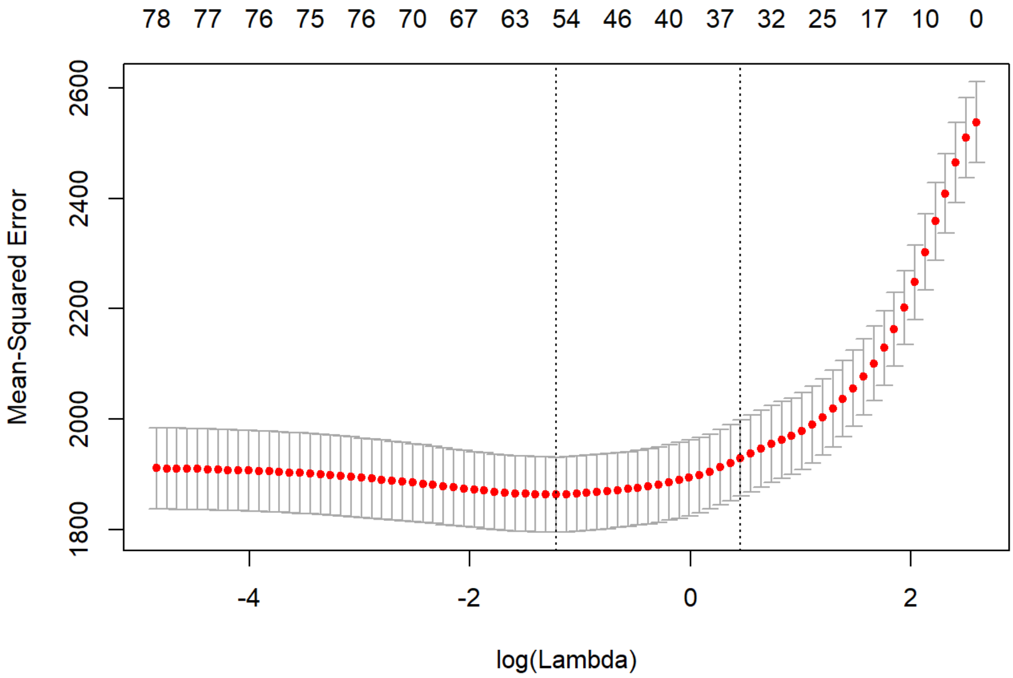
The left vertical line is where the lambda with the minimum deviance, I choose this value and check the error,it is lower than the unregularized model(the very left red dot).

This means if I choose this lambda to do regularization, the training error is lower. So it help improving the model.

II.L1 lasso regression for latitude



II.L1 lasso regression for longitude



lambda that produces the minimun error (latitude and longitude)

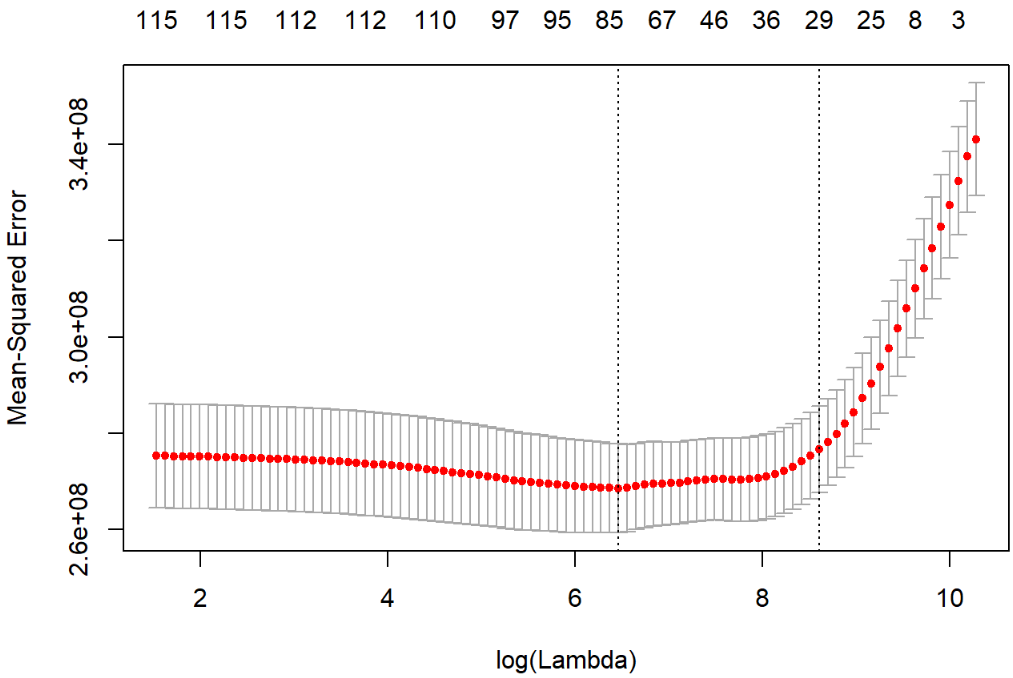
```
## [1] 116.885
```

```
## [1] 0.2958544
```

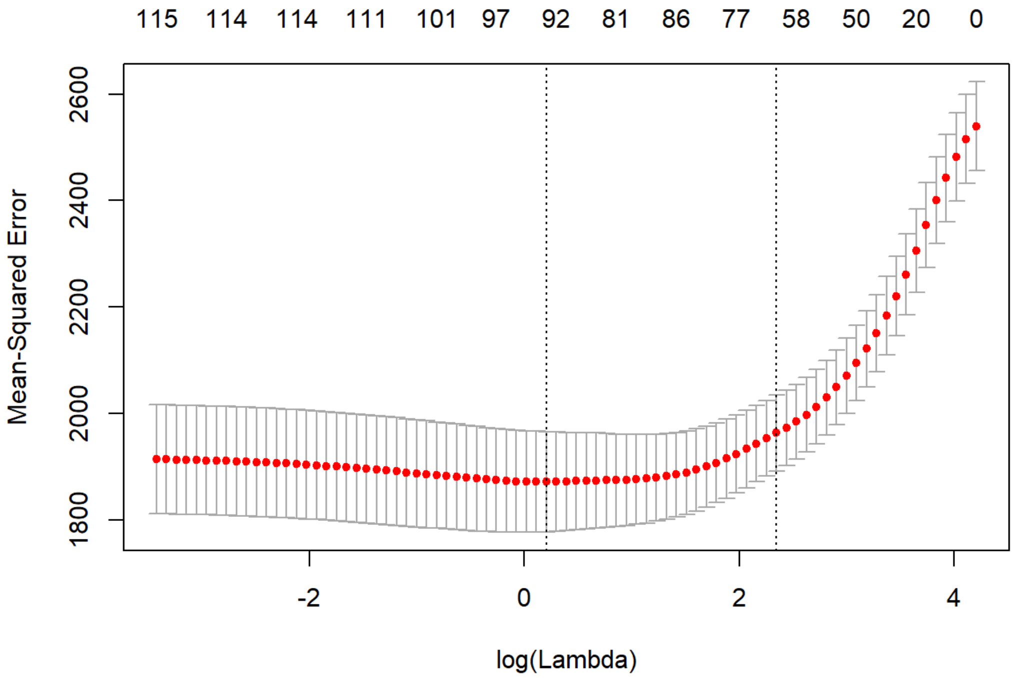
The number of variables is 62 and 56. The theory is the same as part I

I saw the left vertical line has lower error than the very left dot. This means regularization help improve the training error. So doing so is better.

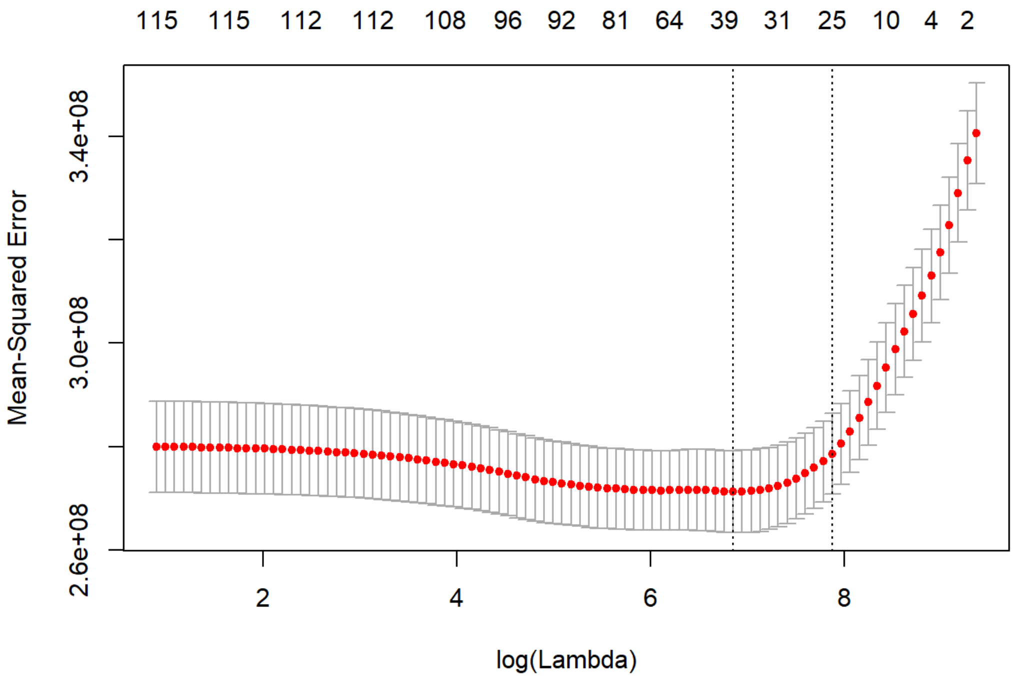
III I will use alpha= 0.2, 0.5, 0.8 to test for both models, when alpha =0.2 ,below shows latitude regression



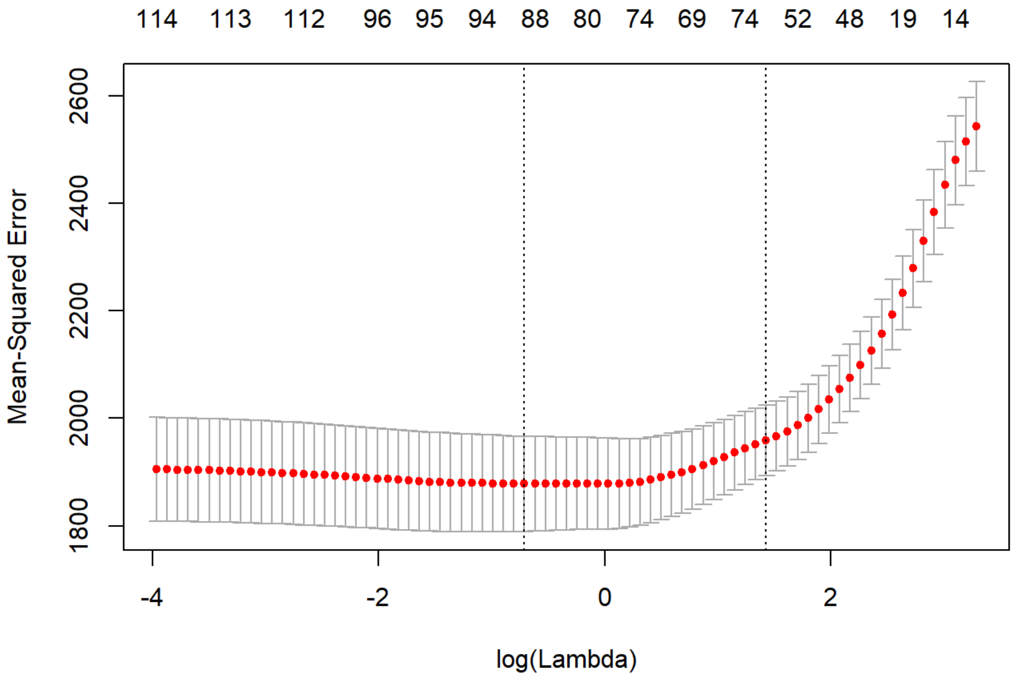
longitutde model at alpha =0.2



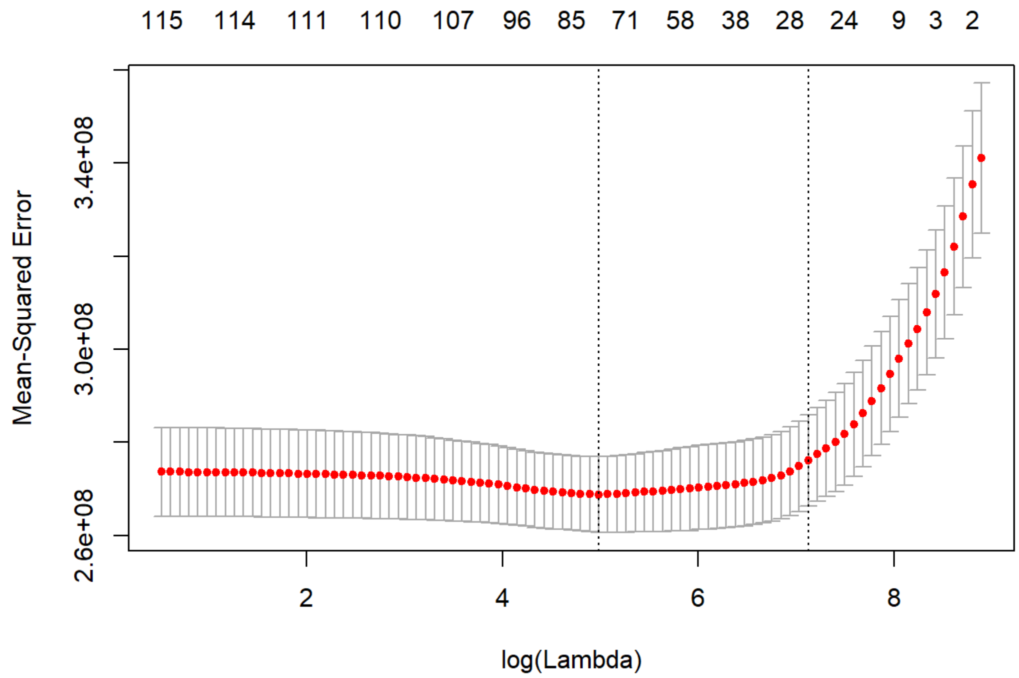
alpha =0.5 of latitude model



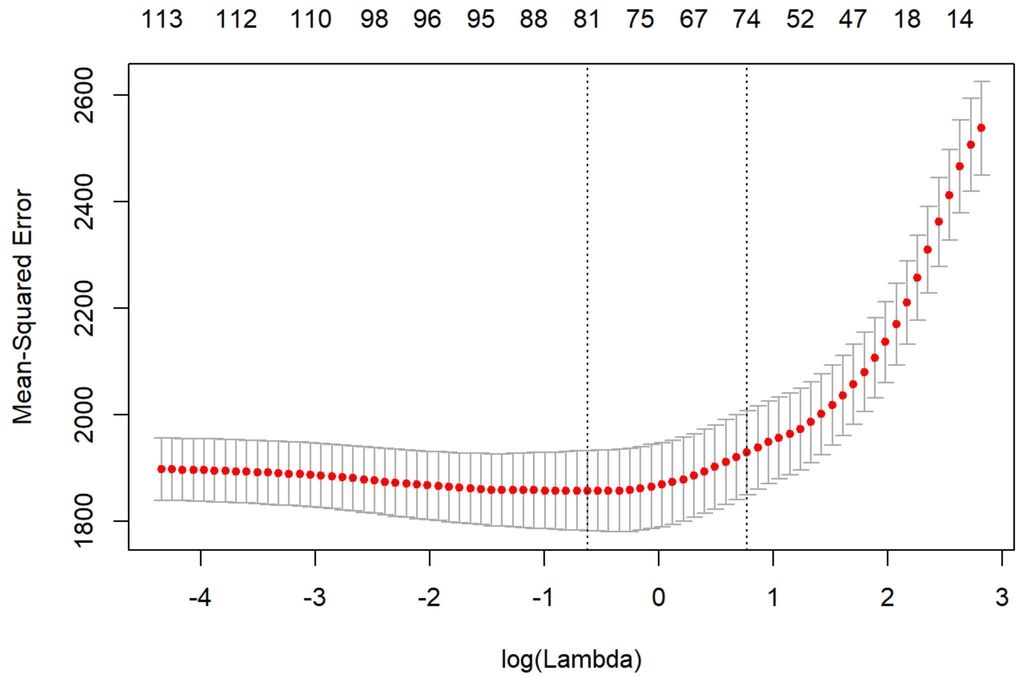
longitude model at alpha=0.5



latitdue model at alpha =0.8



longitude model at alpha =0.8



lambda for 0.2,0.5 and 0.8 elastic net variables for latitude and longitude model

```
## [1] 641.4058 943.7336 146.1063
```

```
## [1] 1.2281178 0.4912471 0.5365427
```

number of variables that are used for latitude and lonigtude

```
## [1] 85 37 81
```

```
## [1] 94 88 81
```

Theories are the same as part I and II. The error at the left line tends to be lower than the very left red dot. So doing regularization is better.

Problem 2

Split the data by 80% training set and test it on the 20% hold-out set,then apply ridge,lasso and elastic net regularization on training set. Use lambda.min to predict the hold-out set and produce the arrucracy.

Below shows accuracy using ridge regularization with lambda.min and lambda.1se. We can see lambdamin shows better accuracy

```
## [1] 0.804134
```

```
## [1] 0.7966328
```

Run lasso regularization and get this accuracy with lambda.min and lambda.1se. We can see lambdamin shows better accuracy

```
## [1] 0.8096349
```

```
## [1] 0.8024671
```

Apply elastic net on alpha from 0.1 to 0.9 with 0.1 interval. We can see for every alpha,lambda.min has higher accuracy than lambda.1se (the first two value are using alpha=0.1 with lambda.min and lambda1se and so on). Overall, Using alpha=0.6 with lambda.min will give us the highest accuracy, thus the best model is alpha=0.6 using elastic net regualrization.

```
##  [1] 0.8091349 0.7971329 0.8093016 0.7992999 0.8094682 0.7979663 0.8093016
##  [8] 0.7992999 0.8094682 0.8001334 0.8098016 0.8019670 0.8096349 0.8021337
## [15] 0.8096349 0.8021337 0.8096349 0.8018003
```