**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](http://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients). Use logistic regression to predict whether the user defaults. You should ignore outliers, but you should try the various regularization schemes we have discussed.

Raw data for Taiwanese credit card user defaults database are downloaded from the webpage and the headres are modified in the downloadd .xls file to remove header rows (more than one). The data file is then converted to csv format (also provided in the submitted homework). The .csv raw data file is read into the R and the databse is crearted. As preparation of the databse, the feature database is genertaed from the 23 feature coloumns and labels are converted to factors with two classes of 0 and 1. The label and factor datasets are then used for cv.glmnet() function of R with parameters of family="binomial", type.measure = 'class' which performes similar to a binary logistic regression analysis. Performance of the binary logistic regression is evaluated using the misclassification error reported by the cv.glmnet() function.

According to the documentation of the cv.glmnet() function, the fuction does a 10-fold cross validiation by splitting the data into 10 sets, fitting the moder on 9 sets and evaluating on the one, repaeating this process 10 times, changing the sets each round.

First an unregulrized linear model is fitted by assigning lmabda and alpha coeficients of zero (0) to cv.glmnet() function. According to the Eq. 1. mentioned below, by feeding zero values for lmabda and alpha coeficients to the cv.glmnet(), it will behave as if an unregulrized linear model is fitted.

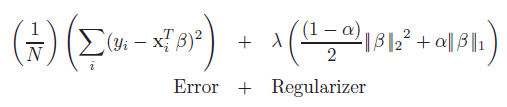


Table 1 shows results for the unregulrized linear model fitted for Taiwanese credit card user defaults database.

Table 1. Results for unregulrized linear model fitted on Taiwanese credit card user defaults database.

|  |  |
| --- | --- |
|  | Unregulrized Linear Model |
| Alpha | 0 |
| Lambda | 0 |
| Misclassification Error | 0.1895333 |
| Accuracy | 0.8104667 |

To study performance of the regulrized linear models, cv.glmnet() function is used with various values of lambda and alpha. For ridge regression alpha of zero, for lasso regression an alpha of 1 and for elastic net regression alpha values between 0 and 1 are used. In these cases, the lambda values are not detremined fo the function and sc.glmnet searches for the best lambda value itslef. It’s worth mentioning that for each case of specific lambda and alpha values, a 10-fold cross validation is poerfomred by the function. The results of the regulrized linear models are provided in Table 2. A selection of the cross validated misscalssification erros result figures are presented in Figures 1 to 6.

According to the obtained results, the accuracies does not flactiate significantly ranging from 78% to 81%. The maximum accuracy (resp. least misclassification error) is 81.07 % obtained from 4th scenario of the elastic net regulrized reggresion in which the an alpha of 0.3 is used. Also it can be seen thet the regression accuracy is slightly deacrased by using regularization and unreglrized regression models stands in the second place with a misclassification error of 81.05%.

Table 2. Results for regulrized linear model fitted on Taiwanese credit card user defaults database.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Scenario 1  (ridge reg) | Scenario 2  (Elastic net) | Scenario 3  (Elastic net) | Scenario 4  (Elastic net) | Scenario 5  (Elastic net) | Scenario 6  (Elastic net) | Scenario 7  (Elastic net) | Scenario 8  (Elastic net) | Scenario 9  (Elastic net) | Scenario 10  (Elastic net) | Scenario 11  (lasso reg) |
| Alpha | 0 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1 |
| Lambda for Minimum MCE | 0.0147951 | 0.0013798 | 0.0013232 | 0.0010625 | 0.0008746 | 0.0005293 | 0.0006399 | 0.0011545 | 0.0006963 | 0.00061892 | 0.00055703 |
| Misclassification Error | 0.1934333 | 0.1900333 | 0.1897333 | 0.1893333 | 0.1898667 | 0.1898 | 0.1898667 | 0.1898667 | 0.1897333 | 0.18963333 | 0.1898 |
| Accuracy | 0.8065667 | 0.8099667 | 0.8102667 | 0.8106667 | 0.8101333 | 0.8102 | 0.8101333 | 0.8101333 | 0.8102667 | 0.81036667 | 0.8102 |

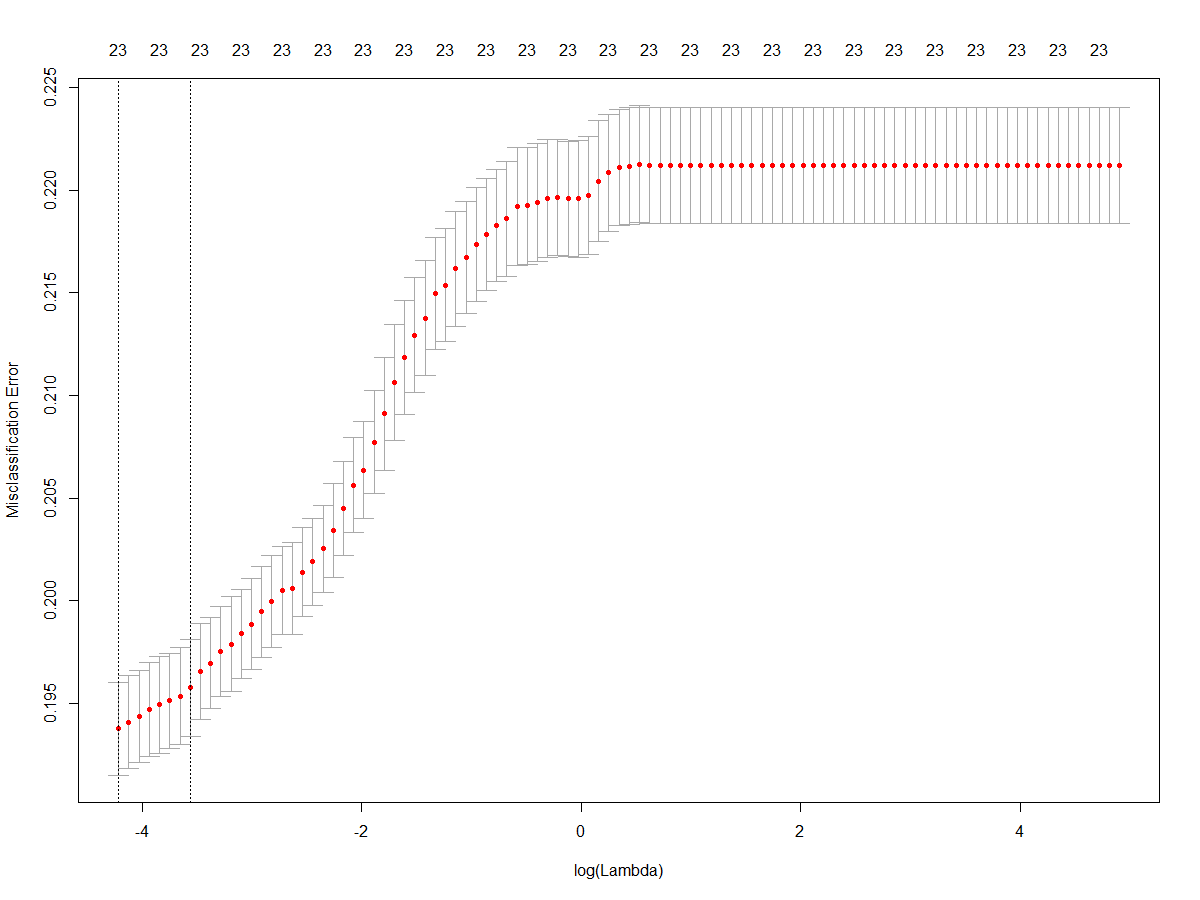


Figure 1. Cross validated misscalssification erros for various Lamndas and Alpha of 0 (ridge regression)

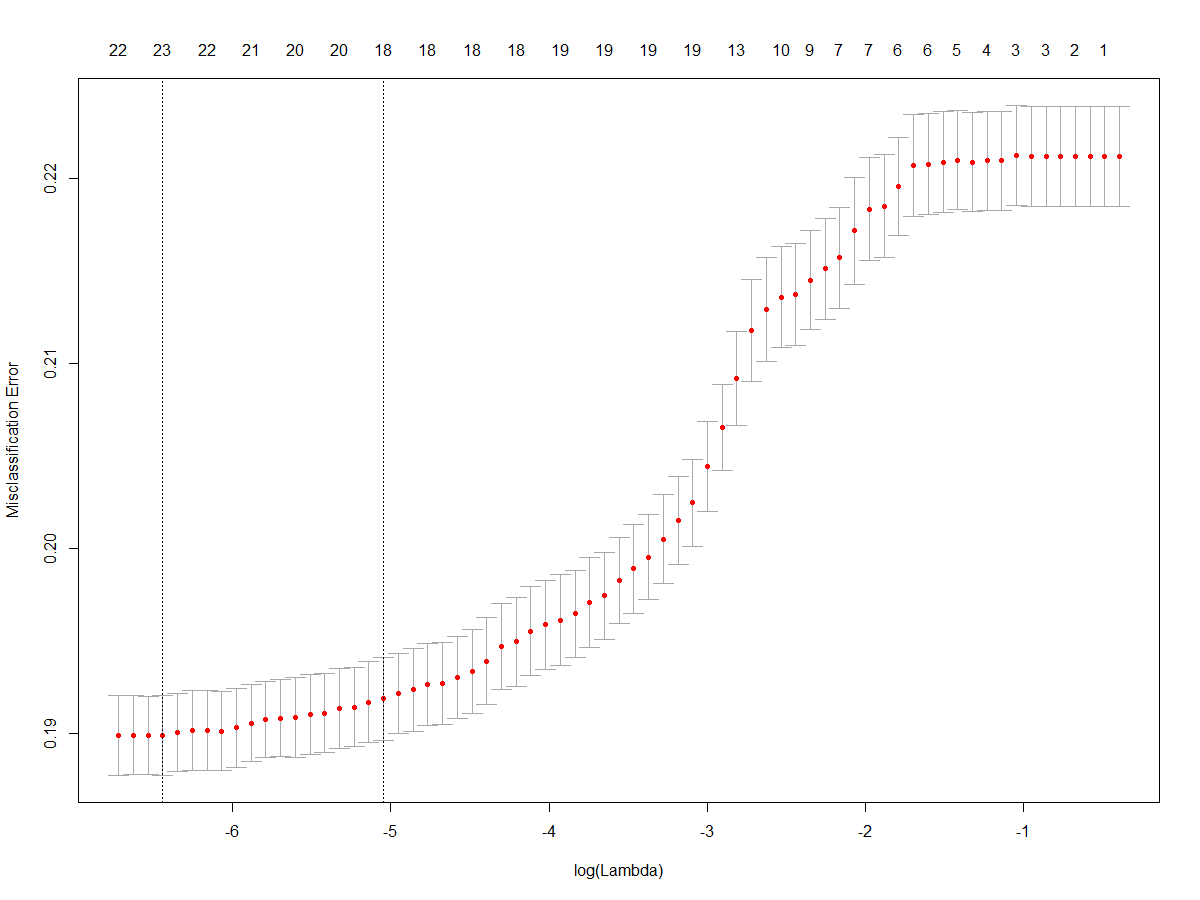


Figure 2. Cross validated misscalssification erros for various Lamndas and Alpha of 0.2 (elastic net regression)

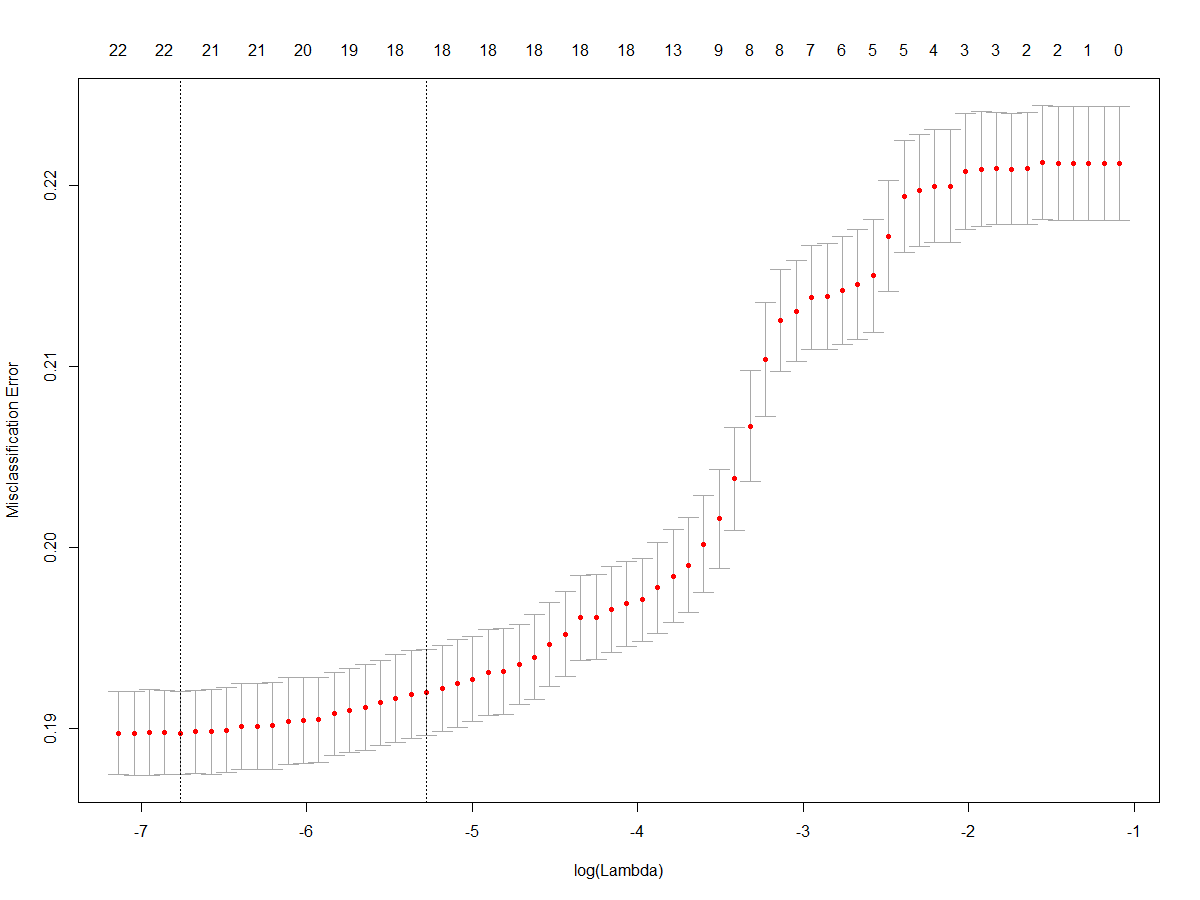


Figure 3. Cross validated misscalssification erros for various Lamndas and Alpha of 0.4 (elastic net regression)

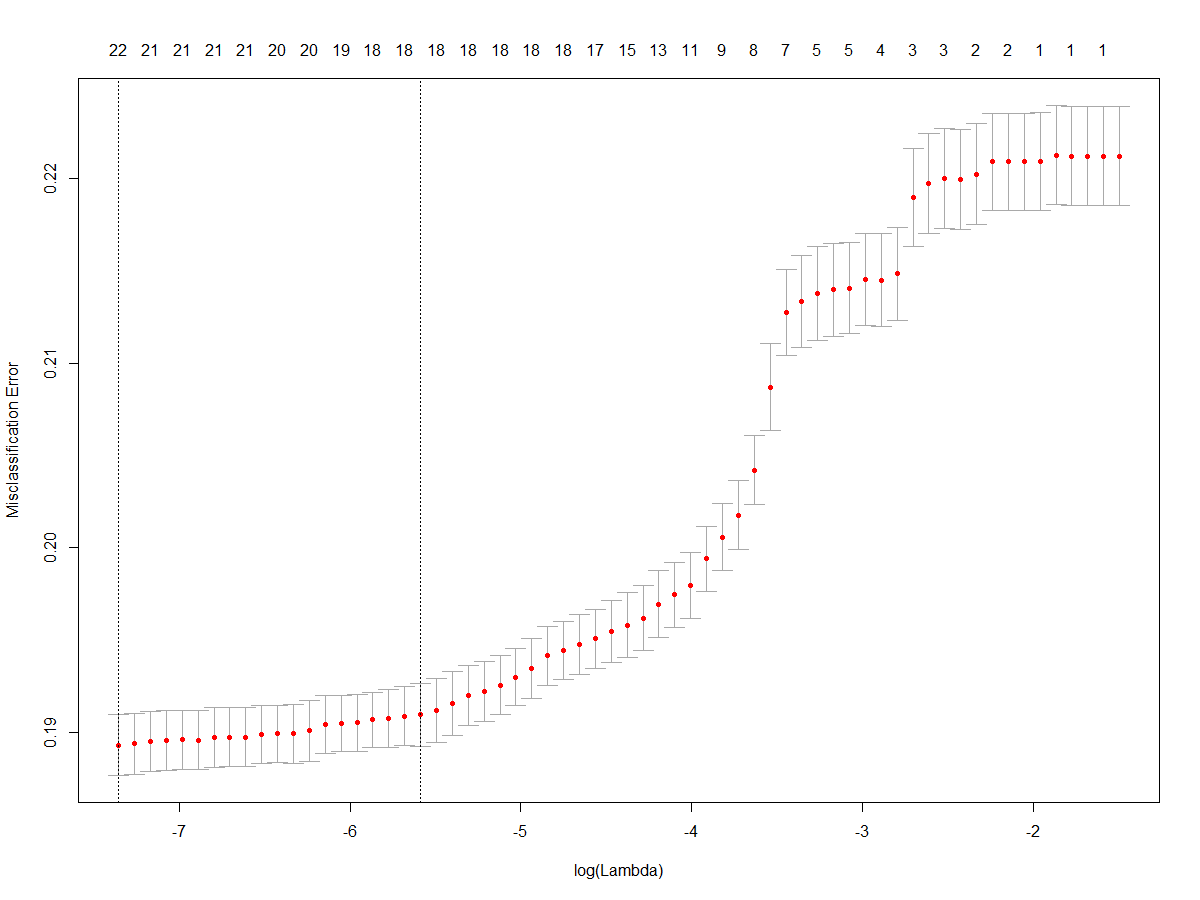


Figure 4. Cross validated misscalssification erros for various Lamndas and Alpha of 0.6 (elastic net regression)

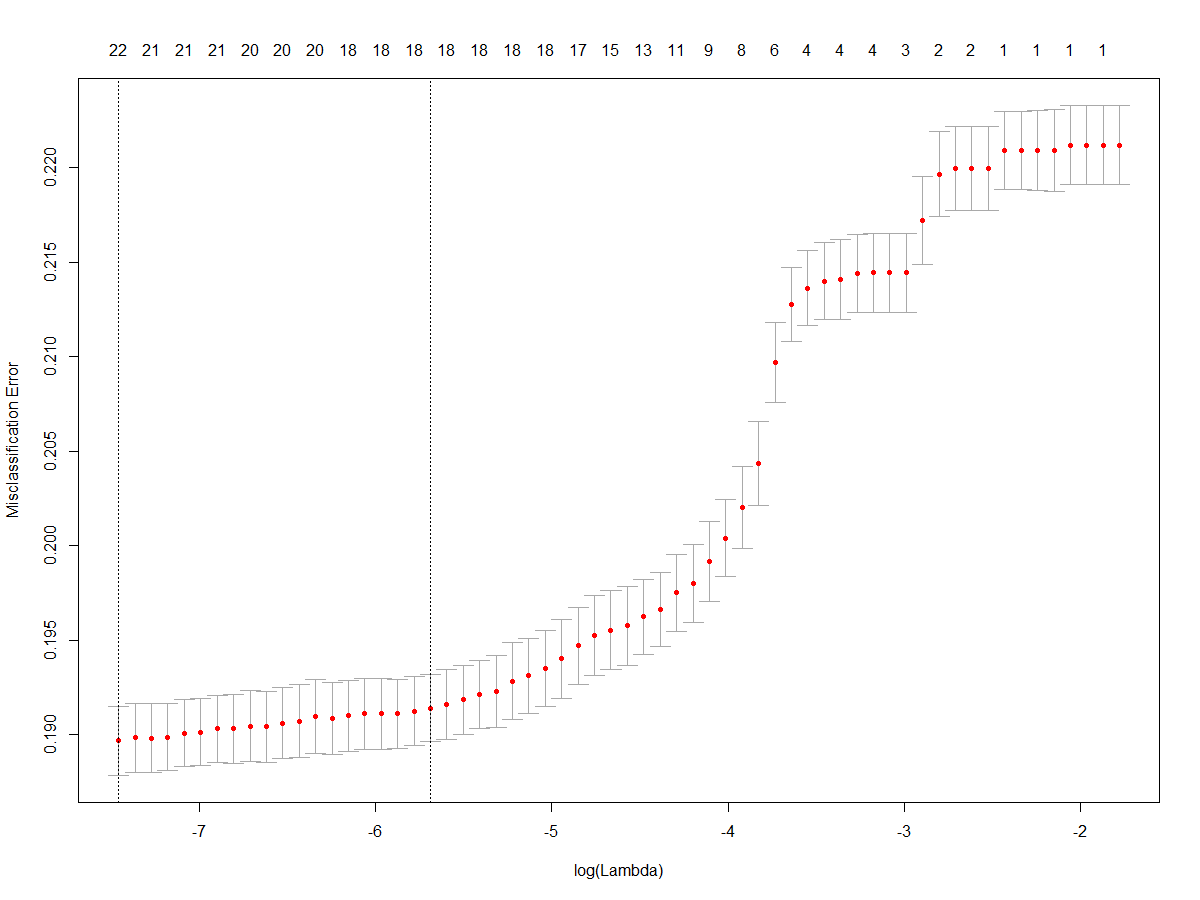


Figure 5. Cross validated misscalssification erros for various Lamndas and Alpha of 0.8 (elastic net regression)

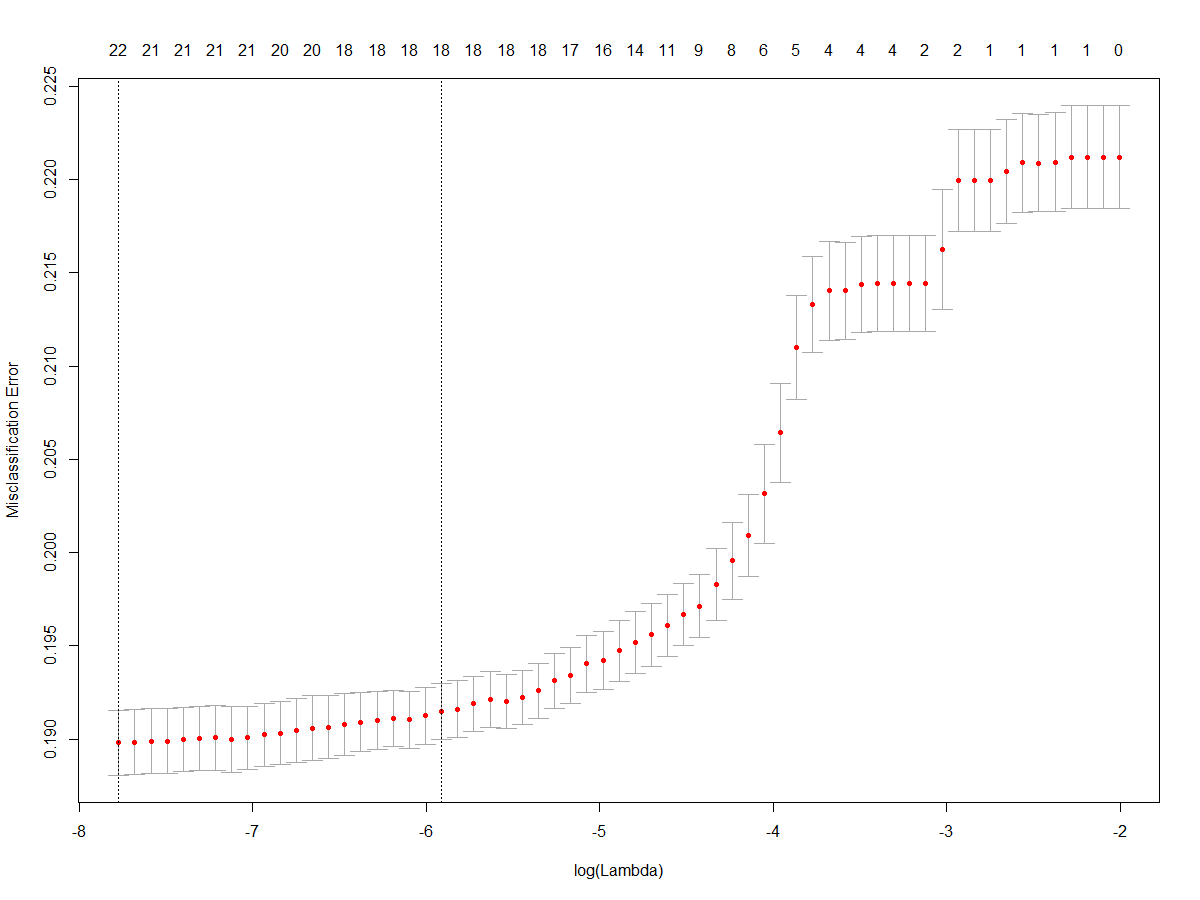


Figure 6. Cross validated misscalssification erros for various Lamndas and Alpha of 1.0 (lasso regression)