Lab 6: Shrinkage Methods

Erin Canada

November 27, 2018

In the computational section of this Lab you will return to the baseball dataset found in the file hitters.csv. Recall that this dataset records the salary of n= 263 Major League Baseball players during the 1987 season as well as q= 19 statistics associated with the performance of each player during the previous season. Specifically, the dataset contains observations from the following variables: • AtBat: Number of times at bat in 1986 • Hits: Number of hits in 1986 • HmRun: Number of home runs in 1986 • Runs: Number of runs in 1986 • RBI: Number of runs batted in in 1986 • Walks: Number of walks in 1986 • Years: Number of years in the major leagues • CAtBat: Number of times at bat during his career • CHits: Number of hits during his career • CHmRun: Number of home runs during his career • CRuns: Number of runs during his career • CRBI: Number of runs batted in during his career • CWalks: Number of walks during his career • League: A categorical variable with levels A (for American) and N (for National) indicating the player's league at the end of 1986 • Division: A factor with levels E (for East) and W (for West) indicating the player's division at the end of 1986 • PutOuts: Number of put outs in 1986 • Assists: Number of assists in 1986 • Errors: Number of errors in 1986 • Salary: 1987 annual salary on opening day in thousands of dollars • NewLeague: A factor with levels A and N indicating the player's league at the beginning of 1987

Interest lies in developing a model that relates a player's annual salary to their previous performance. Your job in this Lab is to investigate several such models. Where computation is required, you must perform the calculations in R.

##(a) Randomly split the observed data into a training set with 210 observations and a held-out test set containing 53 observations. For purposes of reproducibility, please set the seed to be 1 using the command set.seed(1).

```
setwd("C:/Users/Erin Canada/Documents/USF/Fall 2018/Regression/Lab")
hitter <- read.csv("hitters.csv")

## Consider the full model:
m <- lm(Salary ~ ., data = hitter)
summary(m)
```

```
##
## Call:
## lm(formula = Salary ~ ., data = hitter)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -907.62 -178.35  -31.11  139.09 1877.04
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  163.10359   90.77854   1.797 0.073622 .
## AtBat         -1.97987    0.63398  -3.123 0.002008 **
## Hits           7.50077    2.37753   3.155 0.001808 **
## HmRun          4.33088    6.20145   0.698 0.485616
## Runs          -2.37621    2.98076  -0.797 0.426122
## RBI           -1.04496    2.60088  -0.402 0.688204
## Walks          6.23129    1.82850   3.408 0.000766 ***
## Years         -3.48905   12.41219  -0.281 0.778874
## CAtBat        -0.17134    0.13524  -1.267 0.206380
```

```
## CHits          0.13399    0.67455   0.199 0.842713
## CHmRun        -0.17286    1.61724  -0.107 0.914967
## CRuns          1.45430    0.75046   1.938 0.053795 .
## CRBI           0.80771    0.69262   1.166 0.244691
## CWalks        -0.81157    0.32808  -2.474 0.014057 *
## LeagueN       62.59942   79.26140   0.790 0.430424
## DivisionW   -116.84925   40.36695  -2.895 0.004141 **
## PutOuts        0.28189    0.07744   3.640 0.000333 ***
## Assists        0.37107    0.22120   1.678 0.094723 .
## Errors        -3.36076    4.39163  -0.765 0.444857
## NewLeagueN   -24.76233   79.00263  -0.313 0.754218
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 315.6 on 243 degrees of freedom
## Multiple R-squared:  0.5461, Adjusted R-squared:  0.5106
## F-statistic: 15.39 on 19 and 243 DF,  p-value: < 2.2e-16
```

```
## Split data into training and test
set.seed(1)
train <- sample(1:dim(hitter)[1], dim(hitter)[1]/4)
test <- (-train)

## Now let's investigate the optimal ridge regression model. First we need some pre-processing
X <- model.matrix(Salary ~ ., hitter)[,-1]
y <- hitter$Salary
```

##(b) Fit ridge and LASSO regression models for 1000 values of lambda in the range 0.001 to 1010 to the training data. For each type of model construct a plot of the parameter estimates versus lambda with each individual parameter represented as a line of adifferent color on these plots.

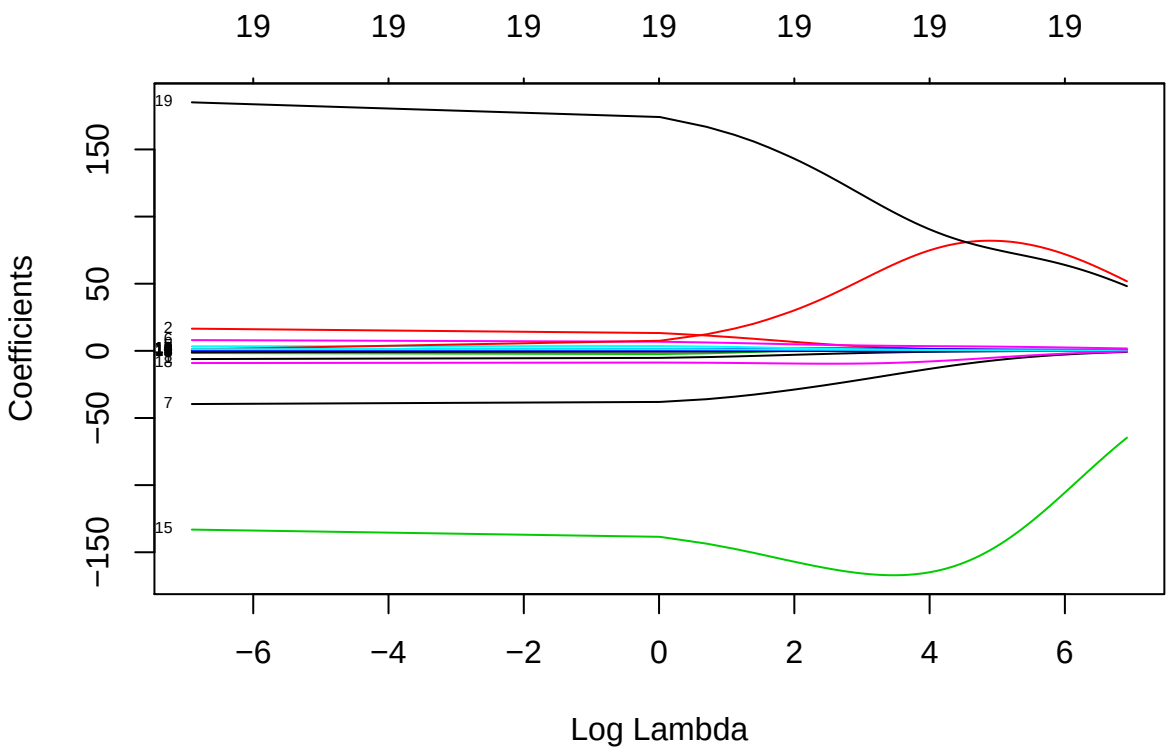
```
library(glmnet)
```

```
## Loading required package: Matrix
```

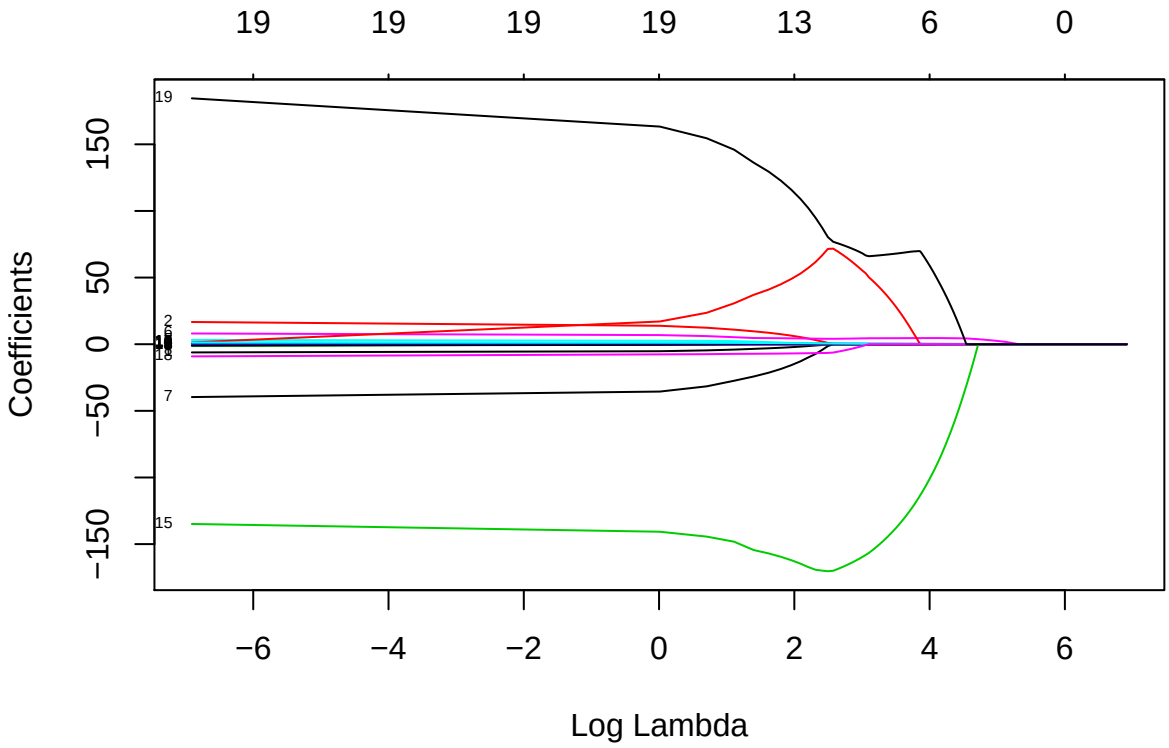
```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-16
```

```
#Ridge
grid <- seq(1010,0.001,length = 1000)
ridge.mod <- glmnet(X[train,], y[train], alpha=0, lambda=grid)
plot(ridge.mod, xvar = "lambda", label = TRUE)
```

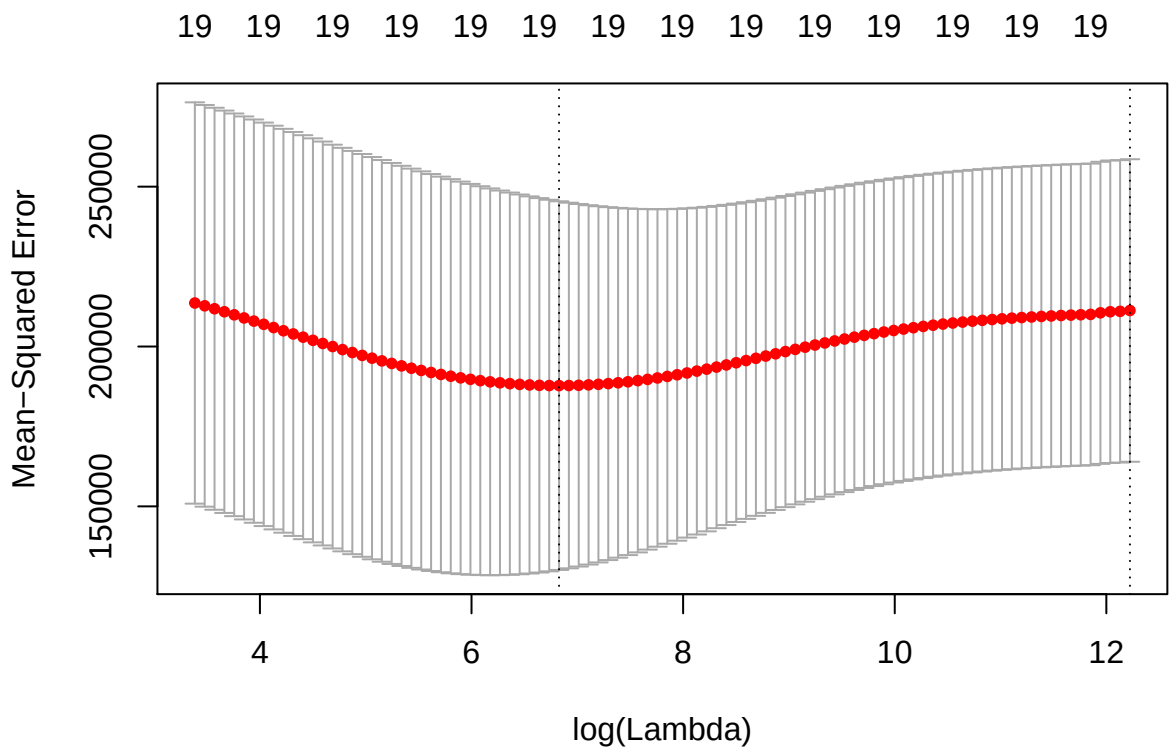



```
#Lasso
lasso.mod <- glmnet(X[train,], y[train], alpha=1, lambda=grid)
plot(lasso.mod, xvar = "lambda", label = TRUE)
```



(c) Using 10-fold cross validation on the training data, find the best ridge regression model. That is, find the optimal value of lambda and the beta estimates that this corresponds to.

```
#Ridge
cvr.out <- cv.glmnet(X[train,], y[train], alpha=0)
plot(cvr.out)
```




```
bestlam.r <- cvr.out$lambda.min
bestlam.r
```

```
## [1] 922.5995
```

```
predict(ridge.mod, s=bestlam.r, type = "coefficients") #Best ridge regression model
```

```
## 20 x 1 sparse Matrix of class "dgCMatrix"
##                            1
## (Intercept) 189.309959086
## AtBat         0.045698695
## Hits          0.390606585
## HmRun         0.695742607
## Runs          0.743340651
## RBI           0.509786198
## Walks         1.833829291
## Years        -0.947506270
## CAtBat        0.005208949
## CHits         0.022968627
## CHmRun        0.021426508
## CRuns         0.052529856
## CRBI          0.035949253
## CWalks        0.063535949
## LeagueN      54.025182505
## DivisionW   -68.404050998
## PutOuts       0.157710383
## Assists       0.124734975
```

```
## Errors       -0.608270741
## NewLeagueN   49.995572955
```

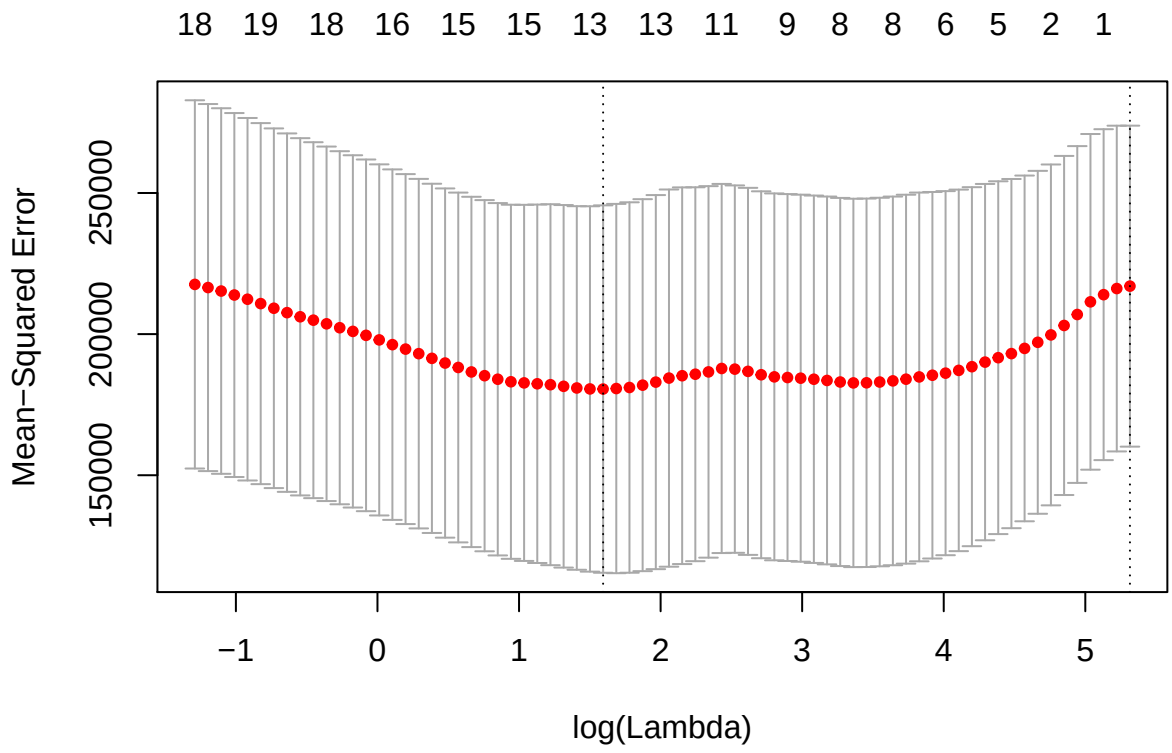
```
paste(cat("Optimal Ridge Lambda:"), bestlam.r)
```

```
## Optimal Ridge Lambda:
```

```
## [1] " 922.599466265185"
```

##(d) Using 10-fold cross validation on the training data, find the best LASSO regression model. That is, find the optimal value of lambda and the beta estimates that this corresponds to.

```
#Lasso
cvl.out <- cv.glmnet(X[train,], y[train], alpha=1)
plot(cvl.out)
```




```
bestlam.l <- cvl.out$lambda.min
bestlam.l
```

```
## [1] 4.923636
```

```
predict(lasso.mod, s=bestlam.l, type = "coefficients")[1:12,] #Best LASSO regression model
```

```
##   (Intercept)         AtBat          Hits         HmRun          Runs
##  3.332698e+02 -3.123453e+00  8.733806e+00  1.472058e-01  0.000000e+00
##           RBI         Walks         Years        CAtBat         CHits
##  1.650806e+00  4.572284e+00 -2.176977e+01  0.000000e+00  5.576005e-06
##        CHmRun         CRuns
##  0.000000e+00  5.851180e-01
```

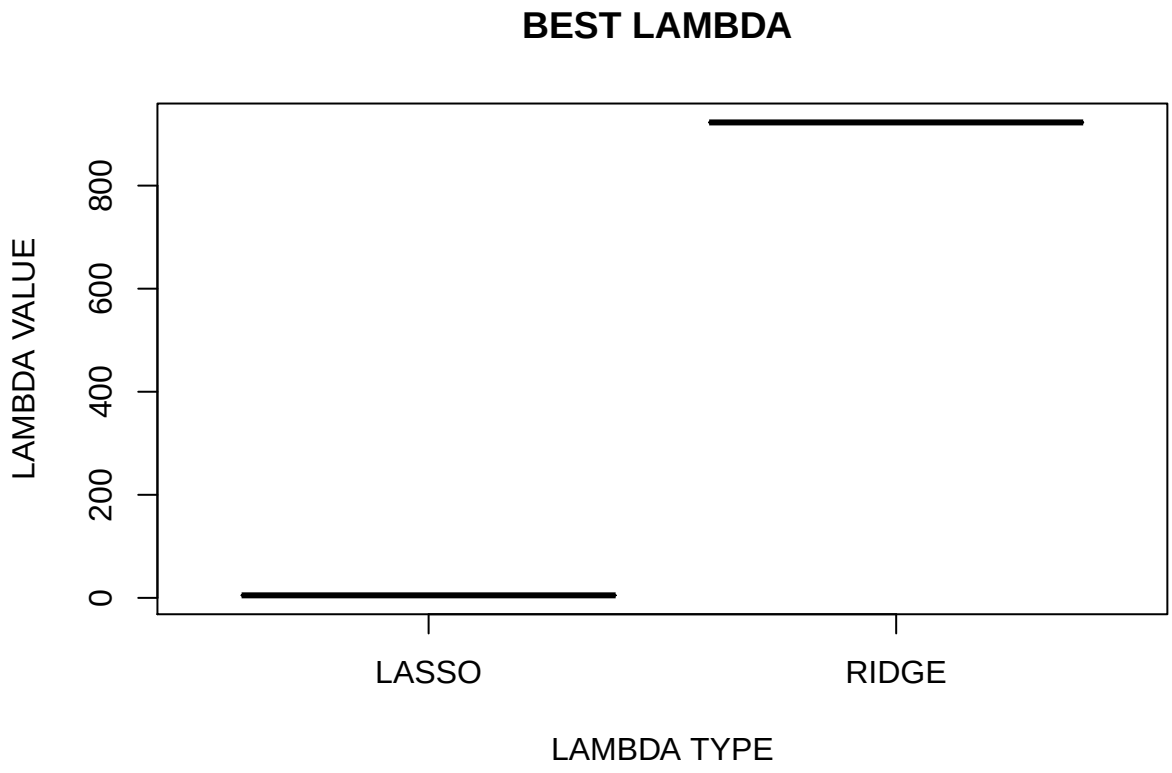
```
paste(cat("Optimal Lasso Lambda:"), bestlam.l)
```

```
## Optimal Lasso Lambda:
```

```
## [1] " 4.92363585464986"
```

##(e) Compare and contrast the models from parts (c) and (d).

```
compare.lam <- data.frame(c("LASSO","RIDGE"),c(bestlam.l,bestlam.r))
plot(compare.lam,xlab = "LAMBDA TYPE",ylab = "LAMBDA VALUE", main = "BEST LAMBDA ")
```



As we can see, the Ridge regression has a much larger lambda which can shrink the betas toward zero. Partially due to the fact in Ridge regression can step closely to lambda equaling 0 but can never be exactly zero. Lasso, however can be particularily small because can shrink towards zero and actually become zero, leaving larger betas. From here, Ridge seems to be on a better track.

##(f) Compare the predictive accuracy of the best ridge and LASSO regression models from parts (c) and (d), and the best stepwise selection model from Lab 5 (which included the predictors AtBat, Hits, Walks, CAtBat, CRuns, CRBI, CWalks,DivisionW, PutOuts and Assists). In particular, use these models to predict the observations from the held-out test set and calculate the corresponding root mean squared error (RMSE) in each case. Based on this criterion, which model is the best?

```
##Best Stepwise
step.mod <- lm(Salary ~ AtBat + Hits +Walks + CAtBat + CRuns + CRBI +
               CWalks + Division + PutOuts +Assists, data = hitter[train,])

step.pred <- predict(step.mod, newdata = hitter[test,])
rmse.step <-sqrt(mean((step.pred - y[test])^2))
```

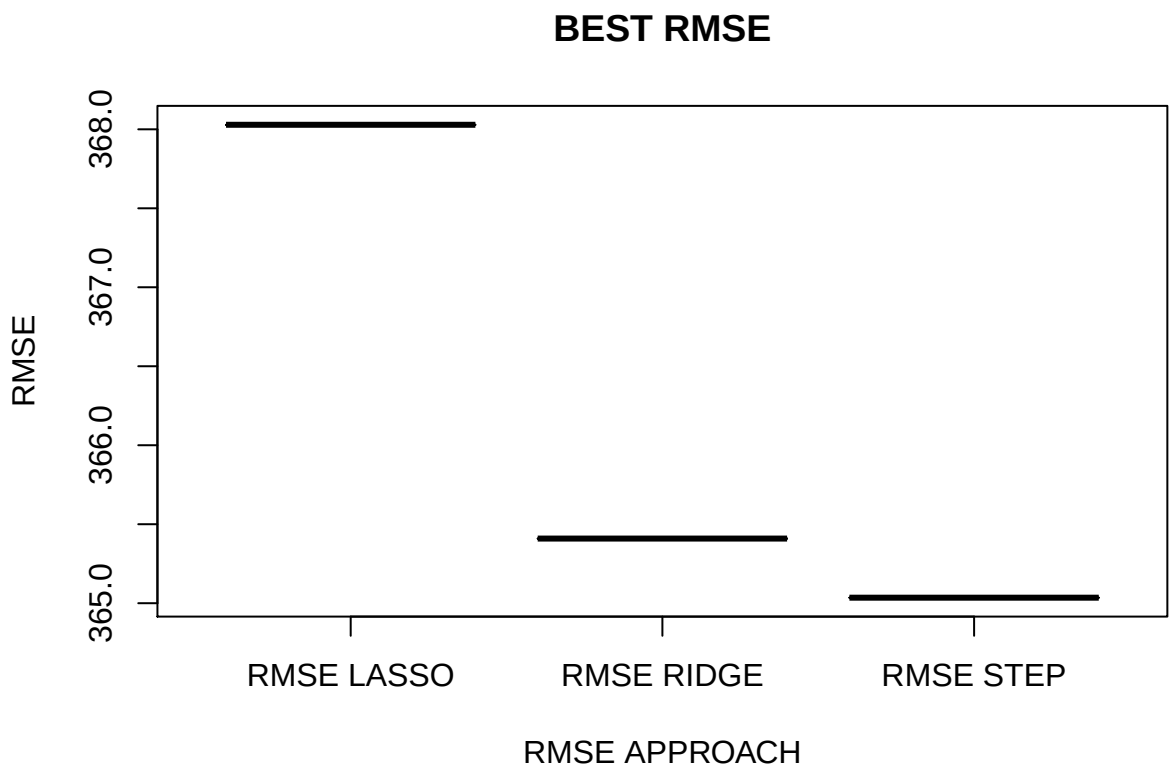
```r
##Ridge
## Next we use the ridge regression model with the best lambda
#value to predict observations in the test set

ridge.pred <- predict(ridge.mod, s=bestlam.r, newx=X[test,])

#The is the predictive RMSE of the best ridge regression model
rmse.r <- sqrt(mean((ridge.pred - y[test])^2))


##Lasso
## Next we use the lasso regression model with the best lambda
#value to predict observations in the test set
lasso.pred <- predict(lasso.mod, s=bestlam.l, newx=X[test,])
#The is the predictive RMSE of the best LASSO regression model
rmse.l <- sqrt(mean((lasso.pred - y[test])^2))


compare <- data.frame(c('rmse.step'= "RMSE STEP",'rmse.r'="RMSE RIDGE",
                        'rmse.l'= "RMSE LASSO"),c(rmse.step,rmse.r,rmse.l))
plot(compare,xlab = "RMSE APPROACH",ylab = "RMSE", main = "BEST RMSE")
```



Based on the Criterion, RMSE RIDGE has the lowest RMSE, highest lambda and would be considered the best model to predict future salaries due to low predicted error.