# Ridge Regression and Lasso

*Chris Kalra*

*4/11/2019*

```
# Set up for a Ridge Regression and a Lasso
#install.packages("glmnet")
library(glmnet) ; library(ISLR)
```

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

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

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

```
Hitters2 = na.omit(Hitters)
x = model.matrix(Salary~., data=Hitters2)[,-1] # the [-1] removes 'Salary' from 'x'
y = Hitters2$Salary
class(x) # ensuring 'x' is a matrix and not a data frame, as the glmnet function
```

```
## [1] "matrix"
```

```
# requires 'x' to be a matrix
```

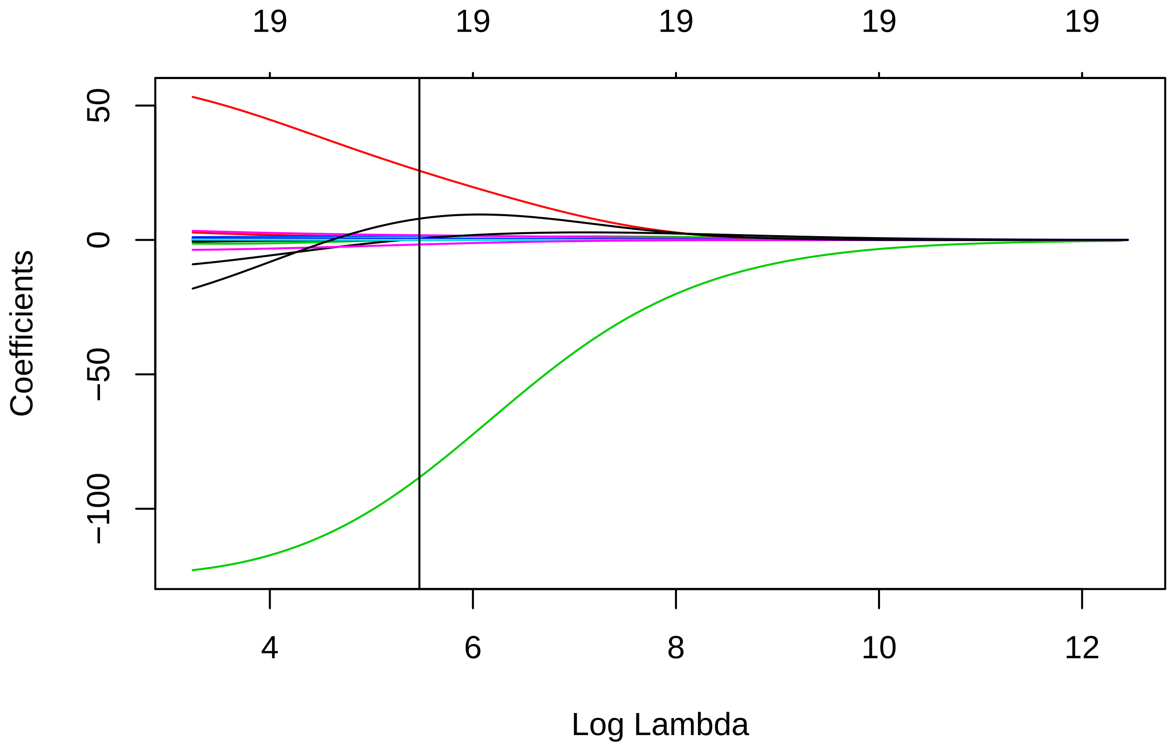
## Ridge Regression

```
ridge_mod = glmnet(x, y, alpha=0)
#alpha = 0 performs ridge regression, and alpha = 1 performs lasso
plot(ridge_mod, xvar = "lambda")

set.seed(1)
ridge_cvfit = cv.glmnet(x, y, alpha=0)
ridge_cvfit$lambda.min #selected lambda value ; $lambda.min is that value that
```

```
## [1] 238.0769
```

```
# minimizes cross-validation error
plot(ridge_mod, xvar = "lambda") ; abline(v=log(ridge_cvfit$lambda.min))
```



```r
coef(ridge_cvfit, s="lambda.min") #notice that the shrunken coefficients approach 0
```

```
## 20 x 1 sparse Matrix of class "dgCMatrix"
##                        1
## (Intercept)  10.35569021
## AtBat         0.04633830
## Hits          0.96376522
## HmRun         0.27163150
## Runs          1.10118079
## RBI           0.87606196
## Walks         1.75331031
## Years         0.50454902
## CAtBat        0.01124891
## CHits         0.06274116
## CHmRun        0.43896753
## CRuns         0.12471202
## CRBI          0.13253839
## CWalks        0.03672947
## LeagueN      25.75710221
## DivisionW   -88.36043501
## PutOuts       0.18483877
## Assists       0.03847012
## Errors       -1.68470903
## NewLeagueN    7.91725605
```

## Model Creation based on Ridge Regression Variable Selection

```r
newfitlm=lm(Salary ~ Runs + Walks + Years + League + Division + Errors + NewLeague, data=Hitters2)
summary(newfitlm) #reduced model
```

```
##
## Call:
## lm(formula = Salary ~ Runs + Walks + Years + League + Division +
##     Errors + NewLeague, data = Hitters2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -733.23 -205.03  -50.05  126.57 2124.43
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -122.185     79.524  -1.536  0.12567
## Runs           5.436      1.289   4.218 3.43e-05 ***
## Walks          3.662      1.468   2.495  0.01322 *
## Years         35.385      4.777   7.407 1.89e-12 ***
## LeagueN       81.920     89.211   0.918  0.35935
## DivisionW   -124.950     44.577  -2.803  0.00545 **
## Errors        -1.881      3.505  -0.537  0.59204
## NewLeagueN   -16.321     87.969  -0.186  0.85296
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 359 on 255 degrees of freedom
## Multiple R-squared:  0.3837, Adjusted R-squared:  0.3668
## F-statistic: 22.68 on 7 and 255 DF,  p-value: < 2.2e-16
```

```r
newfitlm2=lm(Salary ~ Runs + Walks + Years + League + Division + Errors, data=Hitters2)
summary(newfitlm2) #further reduction
```

```
##
## Call:
## lm(formula = Salary ~ Runs + Walks + Years + League + Division +
##     Errors, data = Hitters2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -731.72 -205.11  -49.08  126.17 2122.66
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -122.728     79.320  -1.547  0.12304
## Runs           5.432      1.286   4.223 3.35e-05 ***
## Walks          3.654      1.464   2.496  0.01320 *
## Years         35.387      4.768   7.421 1.71e-12 ***
## LeagueN       67.755     46.062   1.471  0.14253
## DivisionW   -124.998     44.492  -2.809  0.00535 **
## Errors        -1.855      3.496  -0.531  0.59619
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 358.3 on 256 degrees of freedom
## Multiple R-squared:  0.3836, Adjusted R-squared:  0.3692
## F-statistic: 26.56 on 6 and 256 DF,  p-value: < 2.2e-16
```

```
newfitlm3=lm(Salary ~ Runs + Walks + Years + League + Division, data=Hitters2)
summary(newfitlm3) #further reduction
```

```
##
## Call:
## lm(formula = Salary ~ Runs + Walks + Years + League + Division,
##     data = Hitters2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -724.35 -207.07  -56.97  123.35 2125.45
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -133.927     76.353  -1.754  0.08062 .
## Runs           5.297      1.259   4.207 3.58e-05 ***
## Walks          3.702      1.459   2.537  0.01178 *
## Years         35.737      4.716   7.578 6.39e-13 ***
## LeagueN       64.333     45.545   1.413  0.15901
## DivisionW   -125.515     44.419  -2.826  0.00509 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 357.8 on 257 degrees of freedom
## Multiple R-squared:  0.383,  Adjusted R-squared:  0.371
## F-statistic:  31.9 on 5 and 257 DF,  p-value: < 2.2e-16
```

```
notdoneyetlm=lm(Salary ~ Runs + Hits + Walks + Years + Division, data=Hitters2)
summary(notdoneyetlm) #final model ; all significant factors, but
```

```
##
## Call:
## lm(formula = Salary ~ Runs + Hits + Walks + Years + Division,
##     data = Hitters2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -731.62 -209.29  -56.61  103.52 2203.93
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -147.802     71.254  -2.074  0.03905 *
## Runs          -1.093      2.407  -0.454  0.65025
## Hits           3.431      1.194   2.873  0.00441 **
## Walks          4.698      1.455   3.229  0.00140 **
## Years         33.885      4.679   7.242 5.13e-12 ***
## DivisionW   -132.246     43.910  -3.012  0.00286 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 353.5 on 257 degrees of freedom
## Multiple R-squared:  0.3975, Adjusted R-squared:  0.3858
## F-statistic: 33.91 on 5 and 257 DF,  p-value: < 2.2e-16
```

```
# shouldn't "Hits" and "HmRun" be significant, by intuition?
cor(Hitters2$Hits, Hitters2$Runs) # Multicolinearity
```

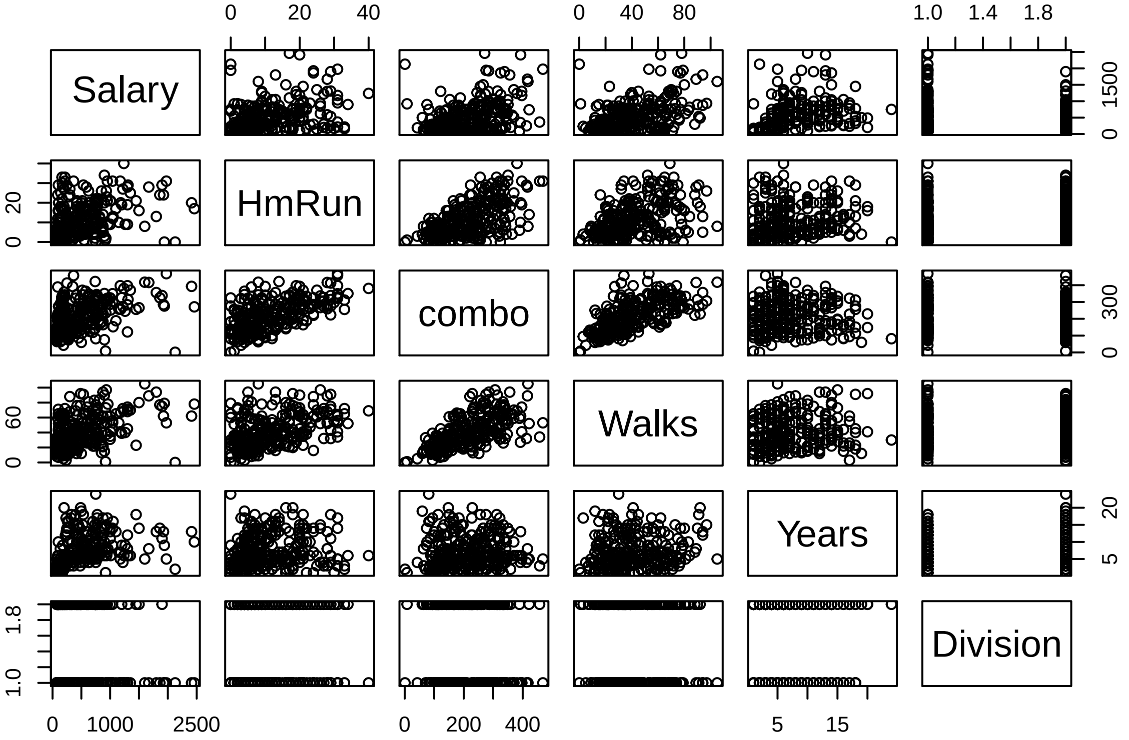
```
## [1] 0.9106301
```

```
cor(Hitters2$Runs, Hitters2$HmRun) ; cor(Hitters2$Hits, Hitters2$HmRun)
```

```
## [1] 0.6310759
```

```
## [1] 0.5306274
```

```
div=mean(Hitters2$Hits)/mean(Hitters2$Runs)
Hitters2$combo=(Hitters2$Hits + div*Hitters2$Runs)
pairs(Salary ~ HmRun + combo + Walks + Years + Division, data=Hitters2)
```



```
# No extreme correlations
combolm=lm(Salary ~ HmRun + combo + Walks + Years + Division, data=Hitters2)
summary(combolm)
```

```
##
## Call:
## lm(formula = Salary ~ HmRun + combo + Walks + Years + Division,
##     data = Hitters2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -750.20 -208.47  -44.86  126.50 2177.96
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -120.924     70.932  -1.705 0.089441 .
## HmRun          2.813      3.159   0.891 0.373984
## combo          1.315      0.357   3.682 0.000282 ***
## Walks          3.751      1.368   2.742 0.006532 **
## Years         34.492      4.689   7.356 2.54e-12 ***
## DivisionW   -129.066     44.036  -2.931 0.003684 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 355 on 257 degrees of freedom
## Multiple R-squared:  0.3926, Adjusted R-squared:  0.3807
## F-statistic: 33.22 on 5 and 257 DF,  p-value: < 2.2e-16
```

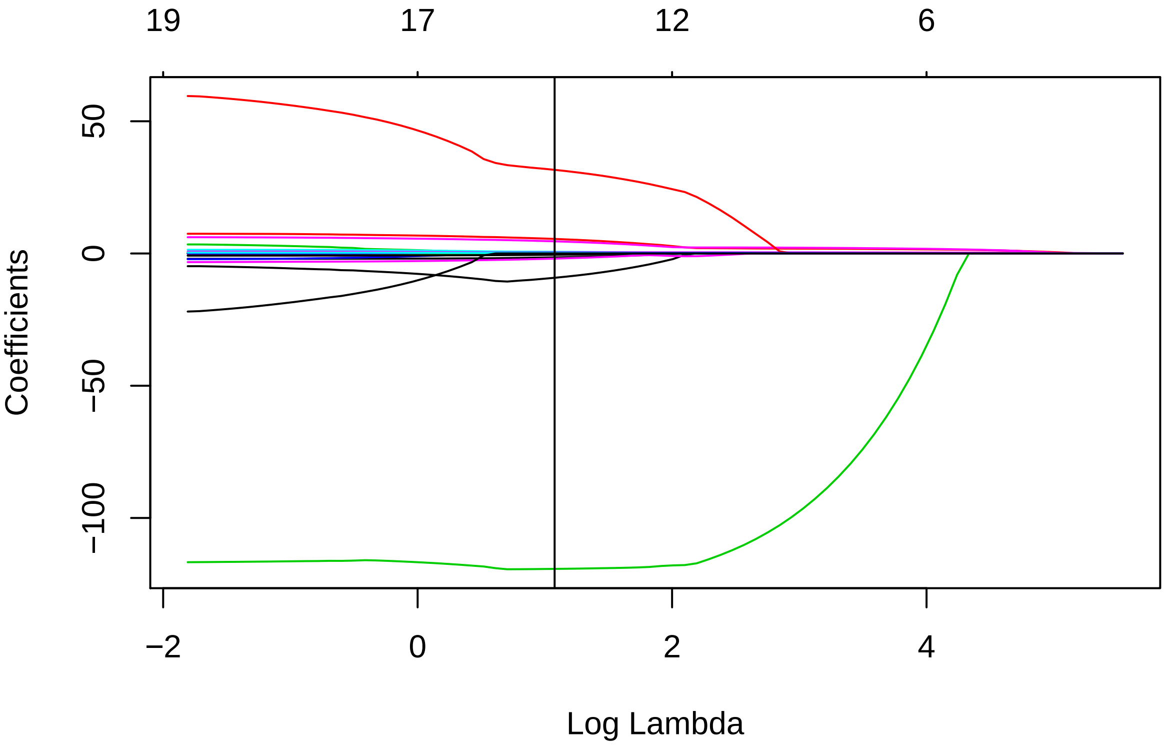
```
# HmRun is not statistically significant, but the researcher opted to keep it in due to
# the intuitive relationship between salary and number of homeruns.
# Therefore, 'combolm' is our final model
```

## Lasso

```
lasso_mod = glmnet(x, y, alpha=1)
plot(lasso_mod, xvar = "lambda")
set.seed(1)
lasso_cvfit = cv.glmnet(x, y, alpha=1)
lasso_cvfit$lambda.min #selected lambda value ; $lambda.min is that value that
```

```
## [1] 2.935124
```

```
# minimizes cross-validation error
plot(lasso_mod, xvar = "lambda") ; abline(v=log(lasso_cvfit$lambda.min))
```



```
coef(lasso_cvfit, s="lambda.min")
```

```
## 20 x 1 sparse Matrix of class "dgCMatrix"
##                        1
## (Intercept)  117.5258436
## AtBat         -1.4742901
## Hits           5.4994256
## HmRun          .
## Runs           .
## RBI            .
## Walks          4.5991651
## Years         -9.1918308
## CAtBat         .
## CHits          .
## CHmRun         0.4806743
## CRuns          0.6354799
## CRBI           0.3956153
## CWalks        -0.4993240
## LeagueN       31.6238173
## DivisionW   -119.2516409
## PutOuts        0.2704287
## Assists        0.1594997
## Errors        -1.9426357
## NewLeagueN     .
```

```
# 6 variables have been shrunken towards 0 by the Lasso method

lasso_coefs = as.numeric(coef(lasso_cvfit, s="lambda.min"))
sum(abs(lasso_coefs)==0)
```

```
## [1] 6
```

```
# 6 additional variables have been set = 0 by the Lasso method
```