Ridge Regression and the Lasso

Mats Hansson

15 oktober 2016

Data

We using the Hitters data to set up a glm model, the data is in the packages `ISLR`.

```
library(glmnet)
```

```
## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-5
```

```
library(ISLR)
summary(Hitters)
```

```
##      AtBat            Hits         HmRun            Runs
##  Min.   : 16.0   Min.   :  1   Min.   : 0.00   Min.   :  0.00
##  1st Qu.:255.2   1st Qu.: 64   1st Qu.: 4.00   1st Qu.: 30.25
##  Median :379.5   Median : 96   Median : 8.00   Median : 48.00
##  Mean   :380.9   Mean   :101   Mean   :10.77   Mean   : 50.91
##  3rd Qu.:512.0   3rd Qu.:137   3rd Qu.:16.00   3rd Qu.: 69.00
##  Max.   :687.0   Max.   :238   Max.   :40.00   Max.   :130.00
##
##       RBI              Walks            Years           CAtBat
##  Min.   :  0.00   Min.   :  0.00   Min.   : 1.000   Min.   :   19.0
##  1st Qu.: 28.00   1st Qu.: 22.00   1st Qu.: 4.000   1st Qu.:  816.8
##  Median : 44.00   Median : 35.00   Median : 6.000   Median : 1928.0
##  Mean   : 48.03   Mean   : 38.74   Mean   : 7.444   Mean   : 2648.7
##  3rd Qu.: 64.75   3rd Qu.: 53.00   3rd Qu.:11.000   3rd Qu.: 3924.2
##  Max.   :121.00   Max.   :105.00   Max.   :24.000   Max.   :14053.0
##
##      CHits            CHmRun           CRuns             CRBI
##  Min.   :   4.0   Min.   :  0.00   Min.   :   1.0   Min.   :   0.00
##  1st Qu.: 209.0   1st Qu.: 14.00   1st Qu.: 100.2   1st Qu.:  88.75
##  Median : 508.0   Median : 37.50   Median : 247.0   Median : 220.50
##  Mean   : 717.6   Mean   : 69.49   Mean   : 358.8   Mean   : 330.12
##  3rd Qu.:1059.2   3rd Qu.: 90.00   3rd Qu.: 526.2   3rd Qu.: 426.25
##  Max.   :4256.0   Max.   :548.00   Max.   :2165.0   Max.   :1659.00
##
##      CWalks          League  Division    PutOuts          Assists
##  Min.   :   0.00   A:175   E:157     Min.   :   0.0   Min.   :  0.0
##  1st Qu.:  67.25   N:147   W:165     1st Qu.: 109.2   1st Qu.:  7.0
##  Median : 170.50                     Median : 212.0   Median : 39.5
##  Mean   : 260.24                     Mean   : 288.9   Mean   :106.9
##  3rd Qu.: 339.25                     3rd Qu.: 325.0   3rd Qu.:166.0
```

```
##  Max.   :1566.00                    Max.   :1378.0   Max.   :492.0
##
##      Errors          Salary       NewLeague
##  Min.   : 0.00   Min.   :  67.5   A:176
##  1st Qu.: 3.00   1st Qu.: 190.0   N:146
##  Median : 6.00   Median : 425.0
##  Mean   : 8.04   Mean   : 535.9
##  3rd Qu.:11.00   3rd Qu.: 750.0
##  Max.   :32.00   Max.   :2460.0
##                  NA's   :59
```

There are some missing values here, so before we proceed we will remove them:

```
Hitters=na.omit(Hitters)
```

Model selection using a validation set

Lets make a training and validation set, so that we can choose a good glm model.

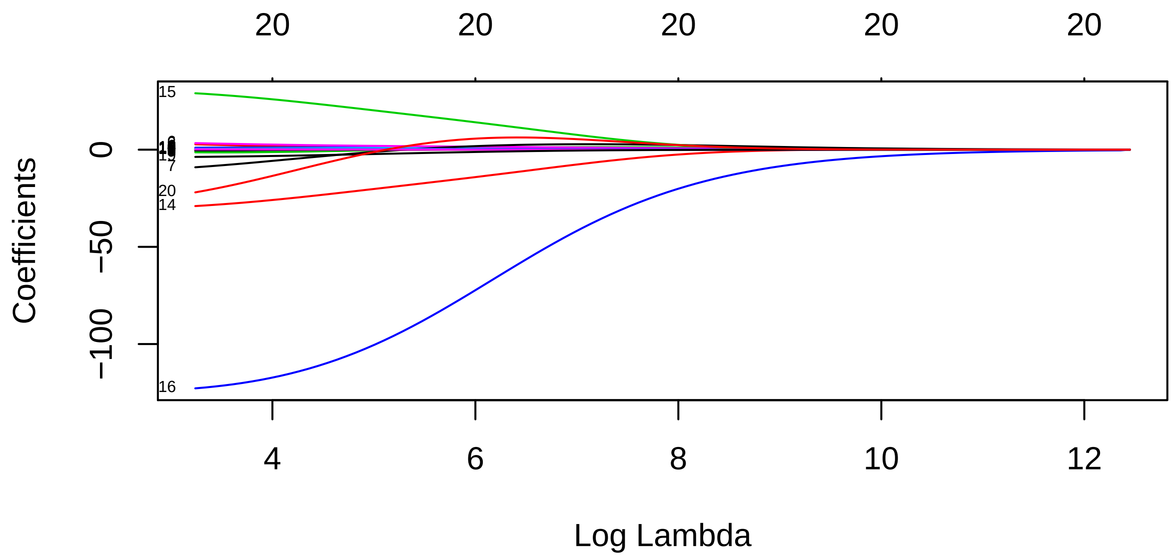
```
set.seed(1)
train=sample(seq(263), 180, replace = FALSE)
```

We will use the packages `glmnet` which does not use the model formula language, so we will set up and `x` and `y`.

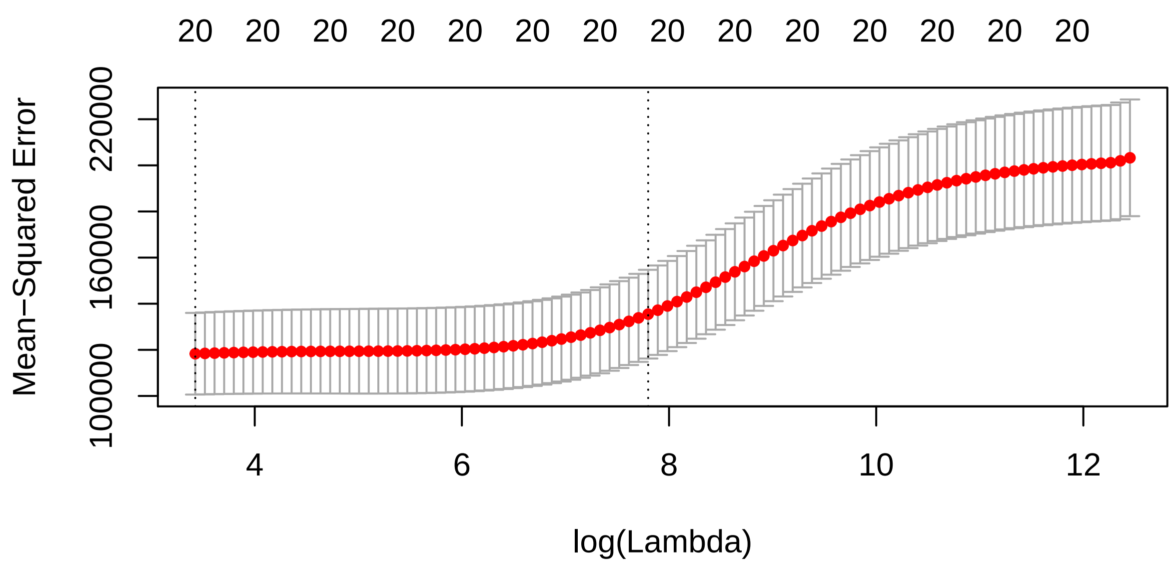
```
x=model.matrix(Salary~.-1, data=Hitters)
y=Hitters$Salary
```

First we will fit a ridge regression model. This achivied by calling `glmnet` with `alpha=0`. There is also `cv.glmnet`function which will do the cross-validation for us.

```
fit.ridge=glmnet(x,y, alpha = 0)
plot(fit.ridge, xvar="lambda", label=TRUE)
```

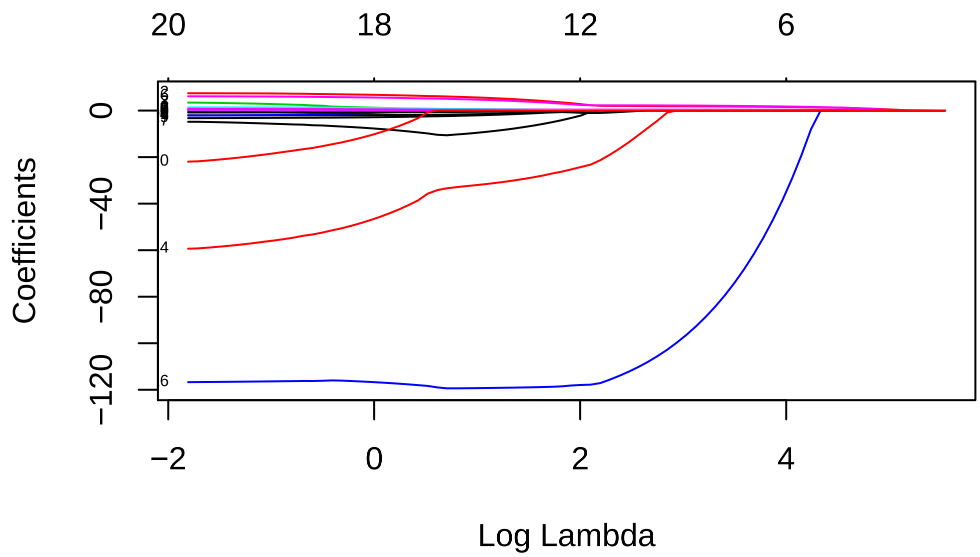



```
cv.ridge=cv.glmnet(x,y, alpha=0)
plot(cv.ridge)
```



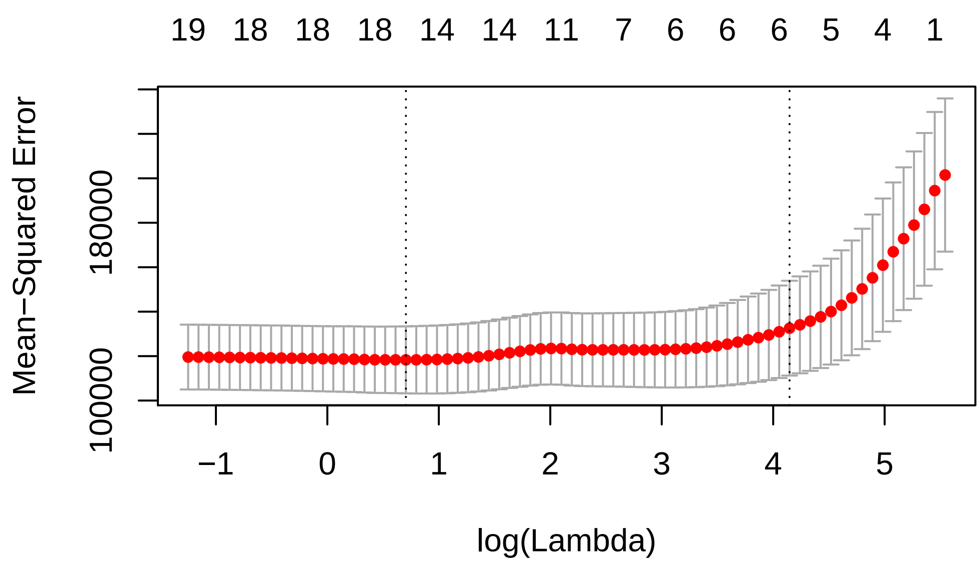
Now we fit a lasso model; for this we use the deafult `alpha=1`.

```
fit.lasso=glmnet(x,y, alpha = 1)
plot(fit.lasso, xvar="lambda", label=TRUE)
```




```
### an alternativ way to plot the model
### plot(fit.lasso, xvar="dev", label=TRUE)

cv.lasso=cv.glmnet(x,y)
plot(cv.lasso)
```




```
coef(cv.lasso)
```

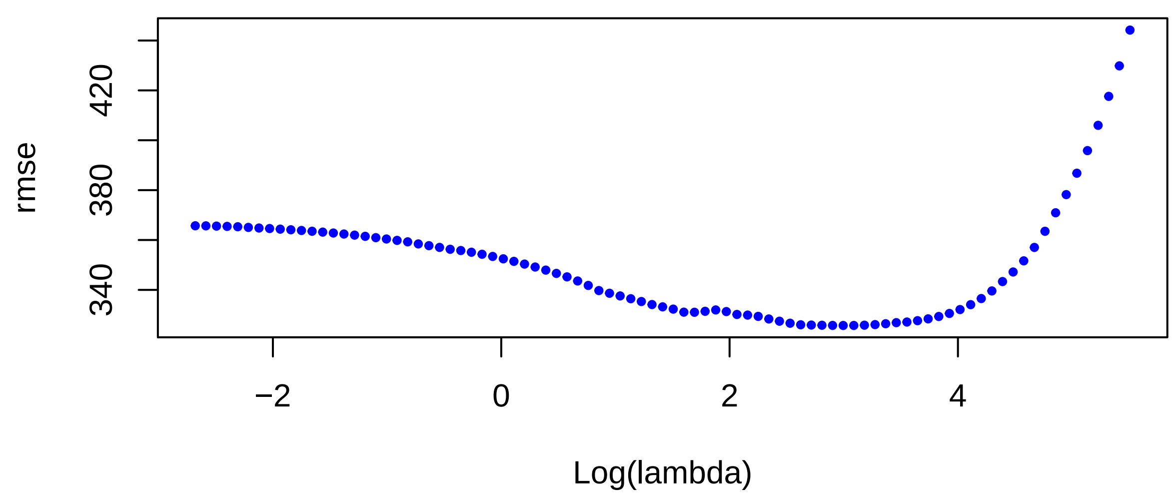
```
## 21 x 1 sparse Matrix of class "dgCMatrix"
```

```
##                        1
## (Intercept) 115.3773590
## AtBat         .
## Hits          1.4753071
## HmRun         .
## Runs          .
## RBI           .
## Walks         1.6566947
## Years         .
## CAtBat        .
## CHits         .
## CHmRun        .
## CRuns         0.1660465
## CRBI          0.3453397
## CWalks        .
## LeagueA       .
## LeagueN       .
## DivisionW   -19.2435216
## PutOuts       0.1000068
## Assists       .
## Errors        .
## NewLeagueN    .
```

Suppose we want to use out earlier train/validation division to select the `lambda` for the lasso.

```r
lasso.tr=glmnet(x[train,], y[train], alpha = 1)
pred=predict(lasso.tr, x[-train,])

rmse=sqrt(apply((y[-train]-pred)^2, 2, mean ))
plot(log(lasso.tr$lambda), rmse, type="b", xlab="Log(lambda)", col="blue", cex=0.5, pch=19)
```



```r
lam.best=lasso.tr$lambda[order(rmse)[1]]

coef(lasso.tr, s=lam.best)
```

```
## 21 x 1 sparse Matrix of class "dgCMatrix"
##                        1
## (Intercept)  107.9416686
## AtBat          .
## Hits           0.1591252
## HmRun          .
## Runs           .
## RBI            1.7340039
## Walks          3.4657091
## Years          .
## CAtBat         .
## CHits          .
## CHmRun         .
## CRuns          0.5386855
## CRBI           .
## CWalks         .
## LeagueA      -30.0493021
## LeagueN        .
## DivisionW   -113.8317016
## PutOuts        0.2915409
## Assists        .
## Errors         .
## NewLeagueN     2.0367518
```