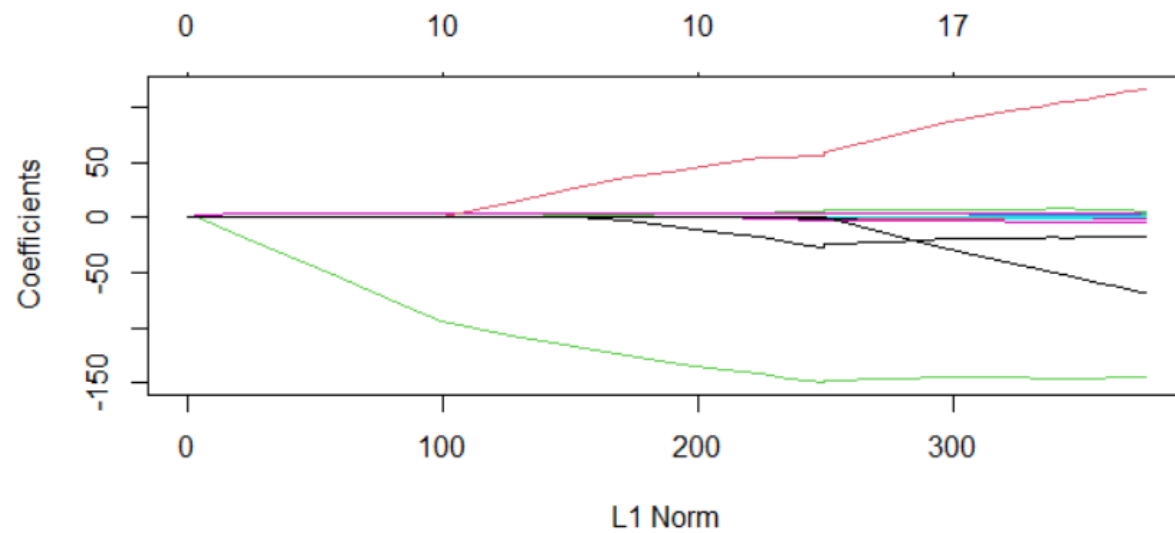
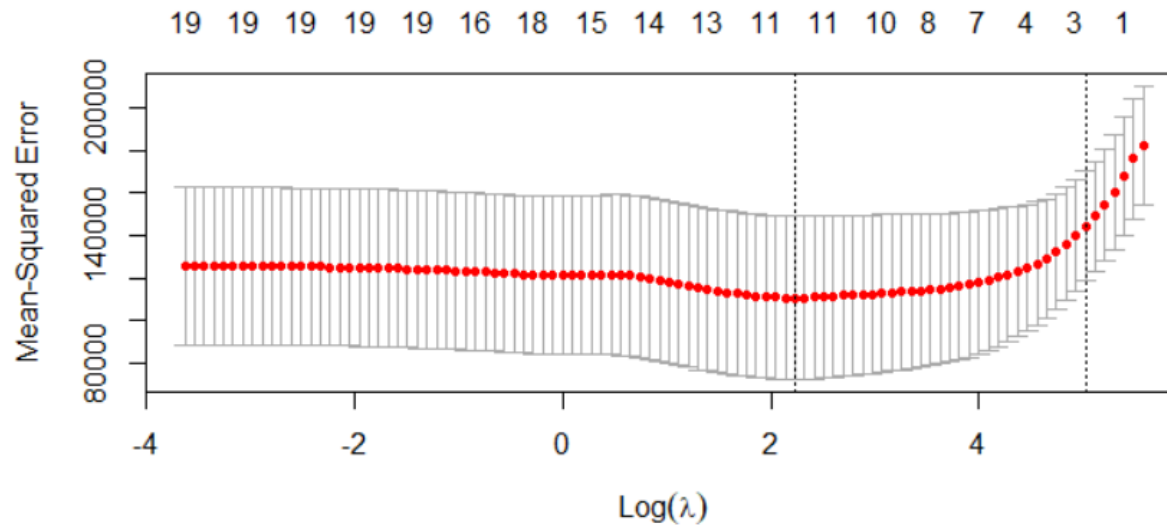


```
#####
> # Lasso Example from 6.5.2 #
> #####
> library(ISLR2)
> names(Hitters)
[1] "AtBat"      "Hits"      "HmRun"     "Runs"      "RBI"      "Walks"     "Years"
[8] "CAAtBat"    "CHits"     "CHmRun"    "CRuns"     "CRBI"     "CWalks"    "League"
[15] "Division"   "PutOuts"   "Assists"   "Errors"    "Salary"   "NewLeague"
> Hitters <- na.omit(Hitters)
>
> # The data have to be separated into a matrix of features and a vector
> # of responses.
>
> x <- model.matrix(Salary~ ., Hitters)[, -1]
> y <- Hitters$Salary
> library(glmnet)
> grid <- 10^seq(10, -2, length = 100) # divides the sequence interval (10,-2) into 100 pieces
> set.seed(1)
> train <- sample(1:nrow(x), nrow(x)/2, replace=FALSE) # split the data in half
> test <- (-train)
> y.test <- y[test]
> lasso.mod <- glmnet(x[train, ], y[train], alpha = 1, lambda = grid) #alpha=0 -> ridge, alpha=1 -> lasso
> plot(lasso.mod)
```



```
> set.seed(1)
> cv.out <- cv.glmnet(x[train, ], y[train], alpha = 1, nfolds=10) # Normally we would use all the data here
> plot(cv.out)
```



The first vertical line at $\log(\lambda)=2.2$ is the minimum CV error. The second
 ### vertical line is at $+1$ standard deviation at about $\log(\lambda)=5$. This larger
 ### λ will have a more sparse set of features.

```
> bestlam <- cv.out$lambda.min
> cv.out$index # This gives the minimum and +1 sd index for lambda
  Lambda
min      37
lse       7
> cv.out$lambda[cv.out$index[2]] # the +1sd lambda
[1] 151.3542

> # Now generate predictions on the test sets and compute the MSE
> lasso.pred <- predict(lasso.mod, s = bestlam ,newx = x[test , ])
> mean ((lasso.pred - y.test)^2)
[1] 143673.6
> lasso.pred2 <- predict(lasso.mod,s = cv.out$lambda[cv.out$index[2]] ,newx = x[test , ])
> mean ((lasso.pred2 - y.test)^2)
```

```

[1] 163594.8
>
> # Now examine the coefficients for the estimates of the full model
> out <- glmnet(x, y, alpha = 1, lambda = grid)
> lasso.coef <- predict(out , type = "coefficients",s = bestlam)[1:20, ]
> lasso.coef
(Intercept)      AtBat      Hits      HmRun      Runs      RBI
1.27479059 -0.05497143  2.18034583  0.00000000  0.00000000  0.00000000
Walks      Years      CAtBat      CHits      CHmRun      CRuns
2.29192406 -0.33806109  0.00000000  0.00000000  0.02825013  0.21628385
CRBI      CWalks      LeagueN      DivisionW      PutOuts      Assists
0.41712537  0.00000000  20.28615023 -116.16755870  0.23752385  0.00000000
Errors      NewLeagueN
-0.85629148  0.00000000
> lasso.coef2 <- predict(out , type = "coefficients",s =cv.out$lambda[cv.out$index[2]] ) [1:20, ]
> lasso.coef2
(Intercept)      AtBat      Hits      HmRun      Runs      RBI      Walks
372.14347751  0.00000000  0.44275715  0.00000000  0.00000000  0.00000000  0.31335072
Years      CAtBat      CHits      CHmRun      CRuns      CRBI      CWalks
0.00000000  0.00000000  0.00000000  0.00000000  0.08289762  0.22157658  0.00000000
LeagueN      DivisionW      PutOuts      Assists      Errors      NewLeagueN
0.00000000  0.00000000  0.00000000  0.00000000  0.00000000  0.00000000

```