5052 Statistical Machine Learning Project

Kah Meng Soh

2022-04-24

```
library(glmnet)
```

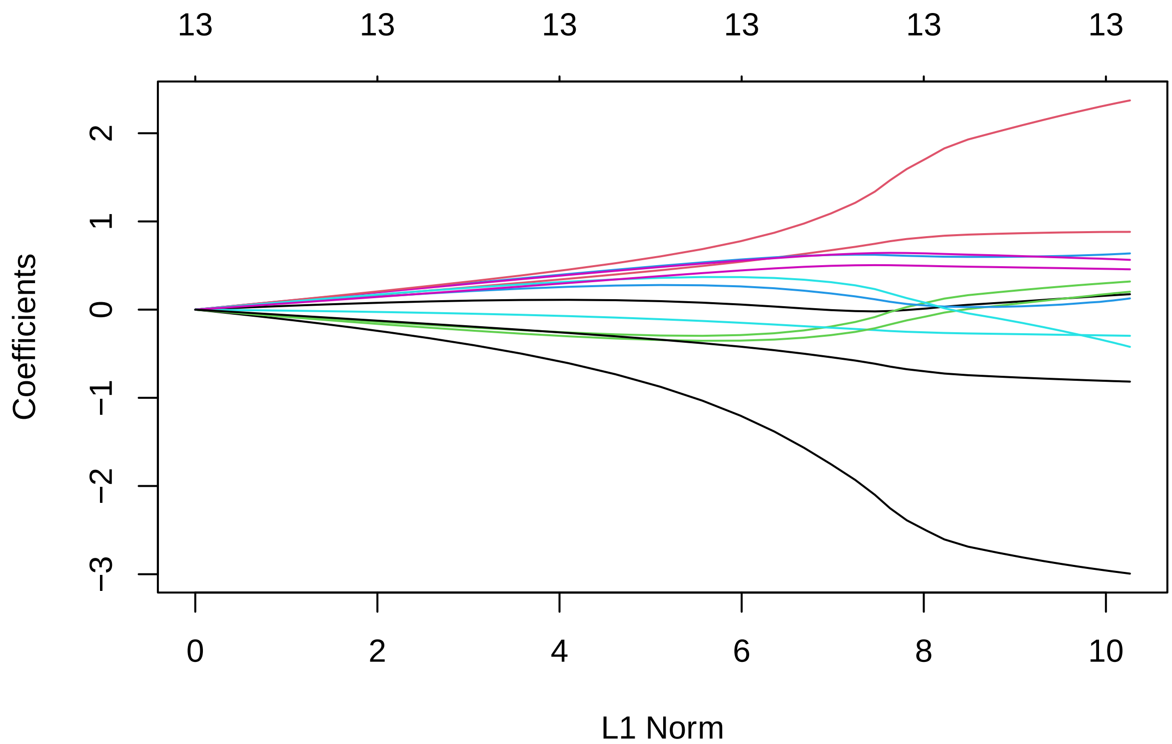
```
## Warning: package 'glmnet' was built under R version 4.1.3
```

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

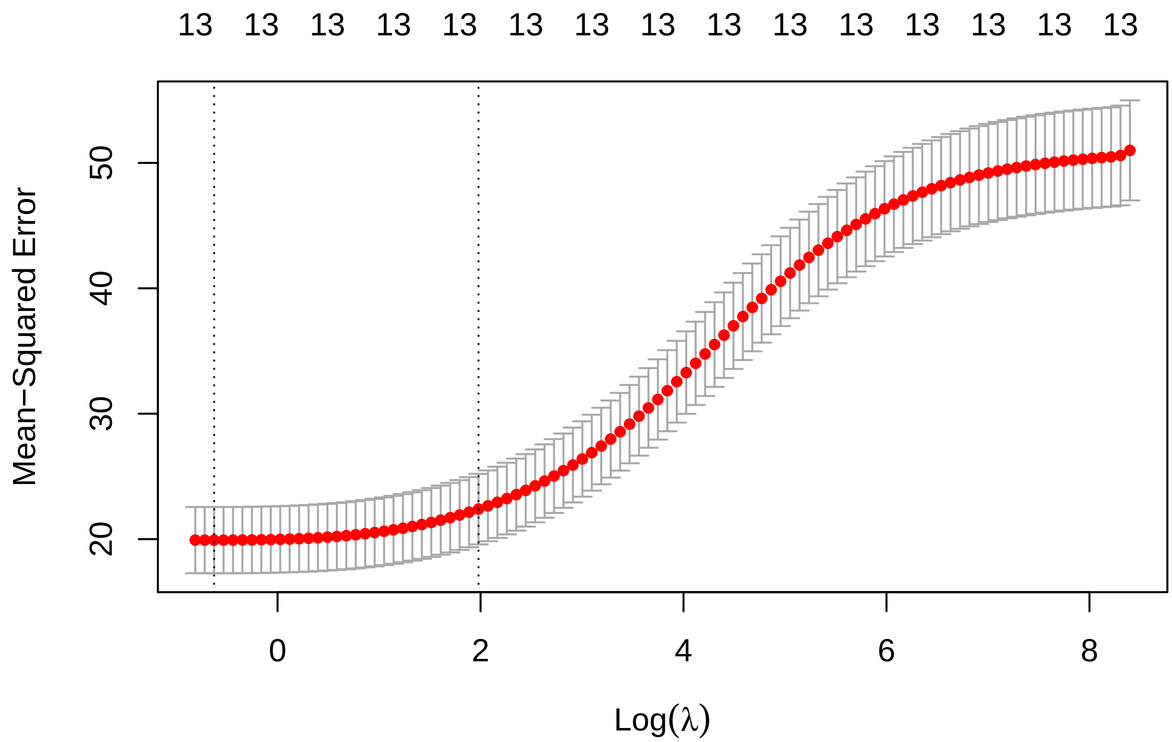
```
## Loaded glmnet 4.1-4
```

```
data=read.csv('C:/Users/micke/Desktop/Boston.csv')
data_scaled <- cbind(scale(data[,1:13]),data[,14])
xfull <- data[,1:13]
yfull <- data[,14]
set.seed(1)
size <- floor(0.8 *  nrow(data_scaled))
trainset <- sample(seq_len(nrow(data_scaled)), size = size)
train <- data_scaled[trainset, ]
xtrain <- train[,1:13]
ytrain <- train[,14]
test <- data_scaled[-trainset,]
xtest <- test[,1:13]
ytest <- test[,14]
```

```
#Ridge Regression (alpha=0)
grid <- 10^seq(10, -2, length = 100)
ridge.mod <- glmnet(xtrain, ytrain, alpha = 0, lambda = grid)
plot(ridge.mod)
```




```
set.seed (1)
cv.out <- cv.glmnet(xtrain, ytrain, alpha = 0)
plot(cv.out)
```




```
bestlam <- cv.out$lambda.min
bestlam
```

```
## [1] 0.5351856
```

```
ridge.pred <- predict(ridge.mod , s = bestlam,newx = xtest)
mean (( ridge.pred - ytest)^2)
```

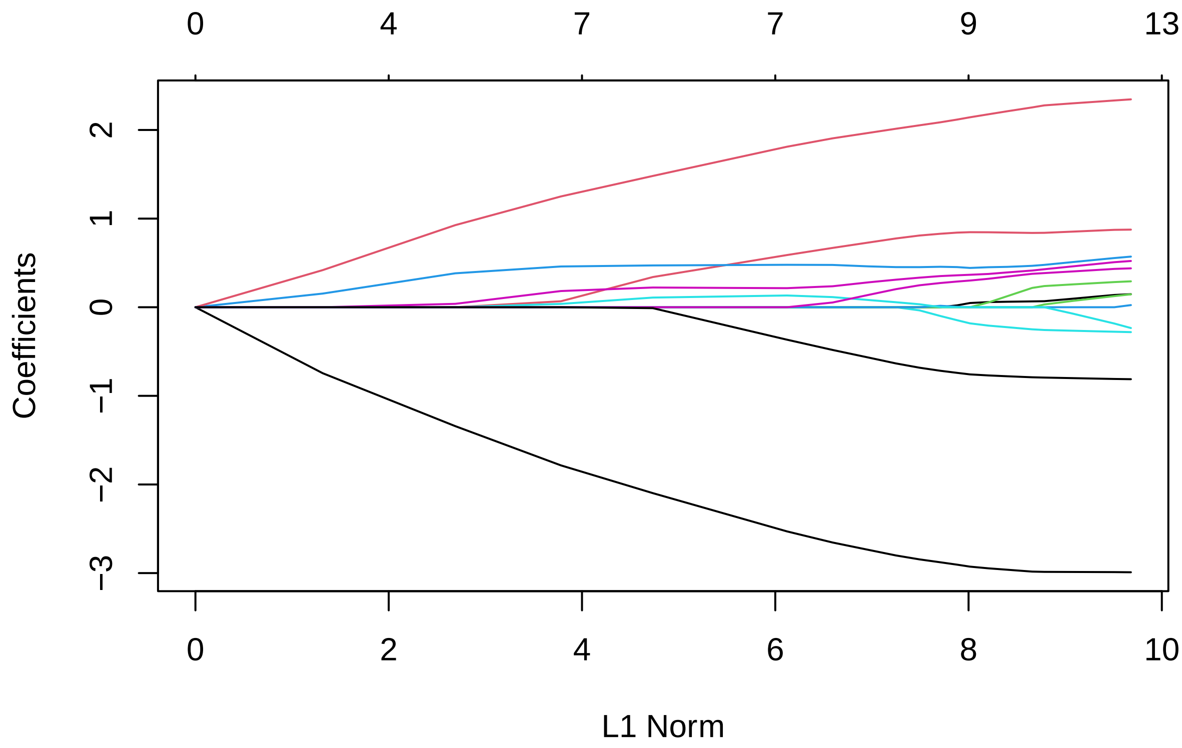
```
## [1] 14.77947
```

```
coef <- glmnet(xfull, yfull, alpha = 0)
predict(coef , type = "coefficients", s = bestlam)[1:14, ]
```

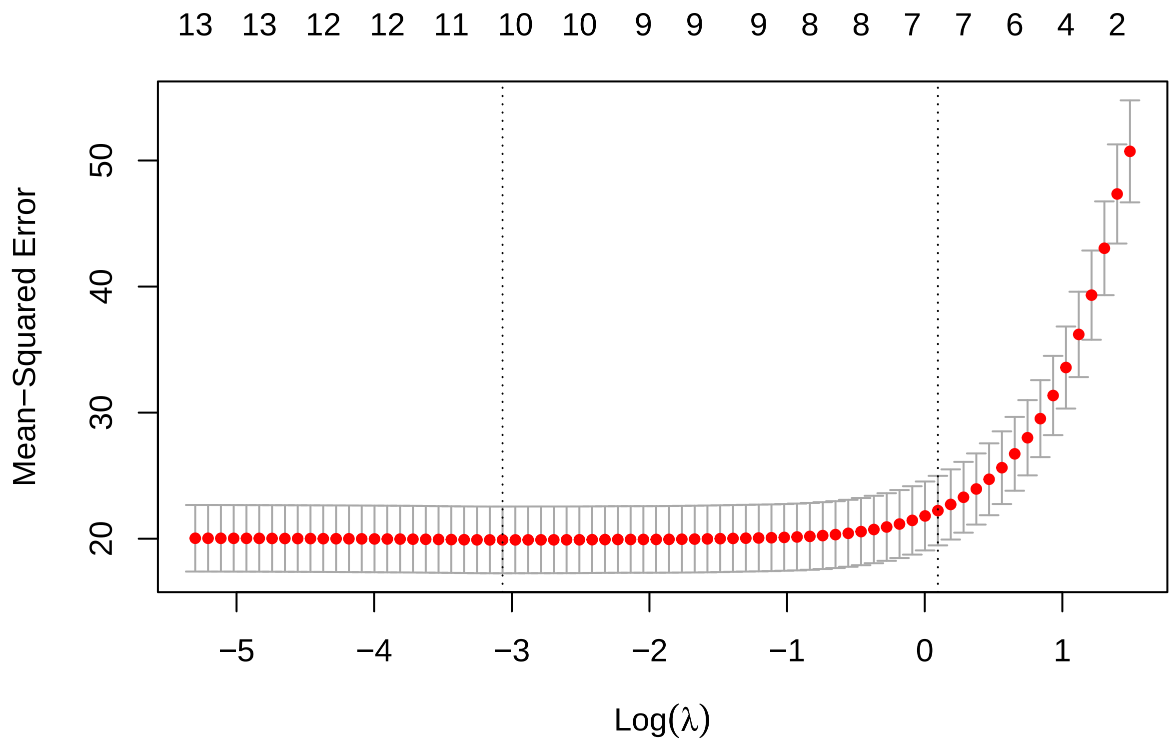
```
##  (Intercept)            X         CRIM           ZN        INDUS         CHAS
## 27.826955292 -0.001203380  0.096158020  0.008949904  0.086844314 -0.955692138
##          NOX           RM          AGE          DIS          RAD          TAX
##  5.901173898 -4.027397442  0.073770929  0.057393868  0.032810947  0.000917734
##      PTRATIO            B
##  0.140751760 -0.007696599
```

```
#Lasso Regression (alpha=1)
lasso.mod <- glmnet(xtrain, ytrain, alpha = 1, lambda = grid)
plot(lasso.mod)
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```




```
set.seed (1)
cv.out <- cv.glmnet(xtrain, ytrain, alpha = 1)
plot(cv.out)
```




```
bestlam <- cv.out$lambda.min
bestlam
```

```
## [1] 0.04654771
```

```
lasso.pred <- predict(lasso.mod , s = bestlam,newx = xtest)
mean (( lasso.pred - ytest)^2)
```

```
## [1] 14.43777
```

```
coef <- glmnet(xfull, yfull, alpha = 1)
predict(coef , type = "coefficients", s = bestlam)[1:14, ]
```

```
##   (Intercept)             X          CRIM            ZN         INDUS          CHAS
## 31.690947761   0.000000000   0.096199990   0.010378337   0.075099239 -0.830553410
##           NOX            RM           AGE           DIS           RAD           TAX
##   4.032377483 -4.393469001   0.084259847   0.035239579   0.034167165   0.000000000
##       PTRATIO             B
##   0.092351600 -0.007991245
```

```
#OLS Regression
train=data.frame(train)
test=data.frame(test)
```

```r
ols.mod = lm(V14~.,train)
ols.pred= predict(ols.mod, newdata=test)
mean (( ols.pred - ytest)^2)
```

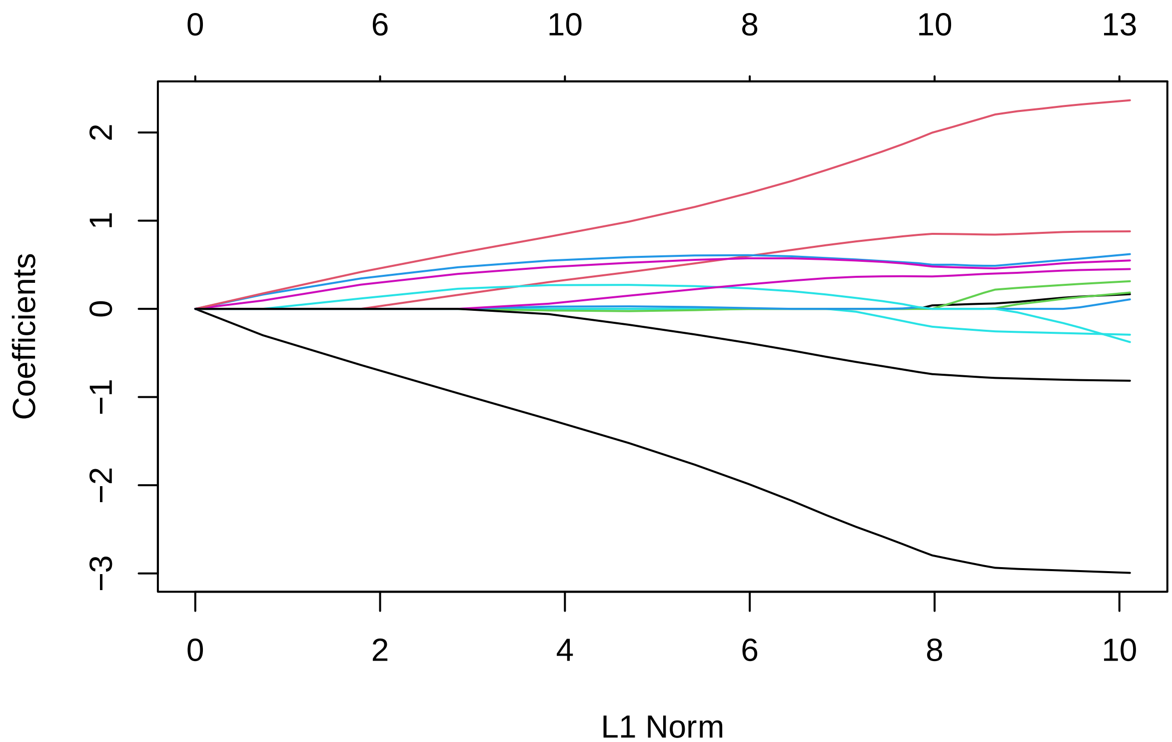
```
## [1] 14.50773
```

```r
ols.modfull = lm(MEDV~.,data)
summary(ols.modfull)
```

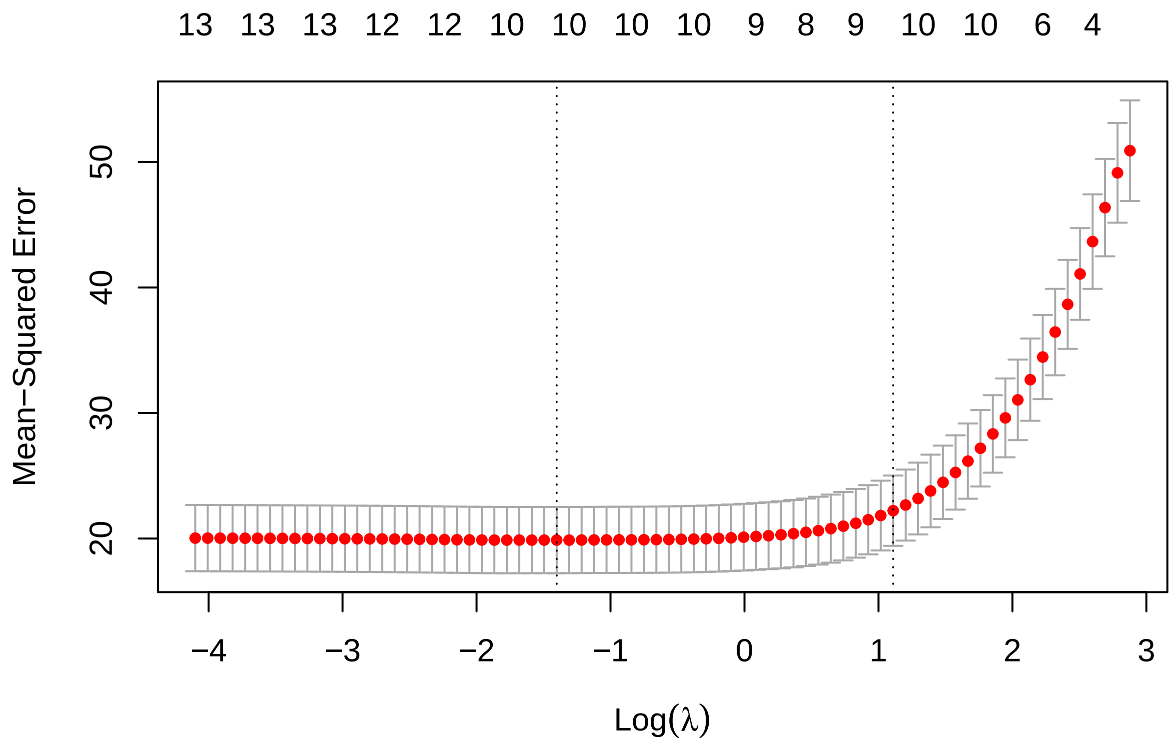
```
##
## Call:
## lm(formula = MEDV ~ ., data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.8948  -2.7585  -0.4663   1.7963  26.0911
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.458826   5.100994   7.147 3.22e-12 ***
## X            -0.002526   0.002080  -1.215 0.225046
## CRIM         -0.108762   0.032855  -3.310 0.001000 **
## ZN            0.048031   0.013785   3.484 0.000538 ***
## INDUS         0.019932   0.061468   0.324 0.745871
## CHAS          2.705245   0.861298   3.141 0.001786 **
## NOX         -17.541602   3.822390  -4.589 5.66e-06 ***
## RM            3.839225   0.418422   9.175  < 2e-16 ***
## AGE          -0.001938   0.013380  -0.145 0.884866
## DIS          -1.493304   0.199892  -7.471 3.68e-13 ***
## RAD           0.324925   0.068111   4.771 2.43e-06 ***
## TAX          -0.011598   0.003807  -3.046 0.002443 **
## PTRATIO      -0.947985   0.130822  -7.246 1.67e-12 ***
## B             0.009357   0.002685   3.485 0.000536 ***
## LSTAT        -0.526184   0.050704 -10.377  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.743 on 491 degrees of freedom
## Multiple R-squared:  0.7414, Adjusted R-squared:  0.734
## F-statistic: 100.6 on 14 and 491 DF,  p-value: < 2.2e-16
```

```r
#Elastic-Net Regression (alpha=0.25)
en.mod <- glmnet(xtrain, ytrain, alpha = 0.25, lambda = grid)
plot(en.mod)
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```




```
set.seed (1)
cv.out <- cv.glmnet(xtrain, ytrain, alpha = 0.25)
plot(cv.out)
```




```
bestlam <- cv.out$lambda.min
bestlam
```

```
## [1] 0.2461333
```

```
en.pred <- predict(en.mod , s = bestlam,newx = xtest)
mean (( en.pred - ytest)^2)
```

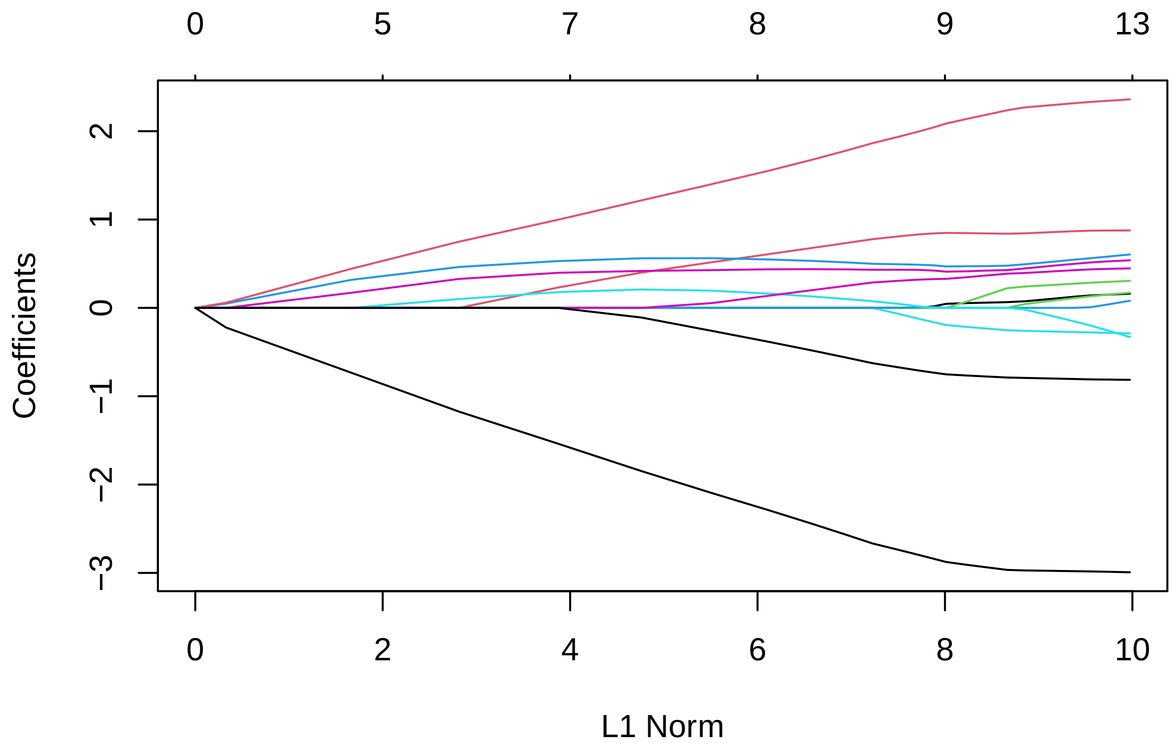
```
## [1] 14.66154
```

```
coef <- glmnet(xfull, yfull, alpha = 0.25)
predict(coef , type = "coefficients", s = bestlam)[1:14, ]
```

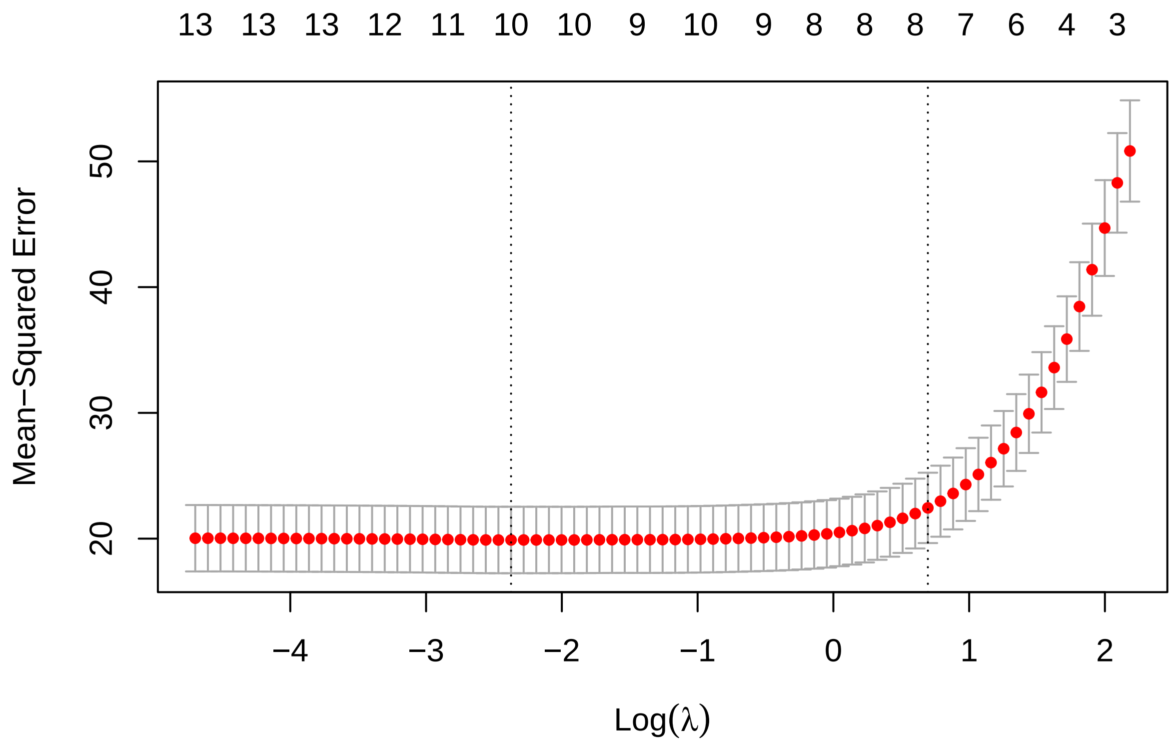
```
##   (Intercept)             X          CRIM            ZN         INDUS
## 30.9339420692  0.0000000000  0.0951289398  0.0065256277  0.0726368669
##          CHAS           NOX            RM           AGE           DIS
## -0.7742990650  4.3186370290 -4.2444735517  0.0779824509  0.0000000000
##           RAD           TAX       PTRATIO             B
##  0.0277256363  0.0004871178  0.0959397144 -0.0077286520
```

```
#Elastic-Net Regression (alpha=0.5)
en.mod <- glmnet(xtrain, ytrain, alpha = 0.5, lambda = grid)
plot(en.mod)
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```




```r
set.seed (1)
cv.out <- cv.glmnet(xtrain, ytrain, alpha = 0.5)
plot(cv.out)
```




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

```
## [1] 0.09309542
```

```r
en.pred <- predict(en.mod , s = bestlam,newx = xtest)
mean (( en.pred - ytest)^2)
```

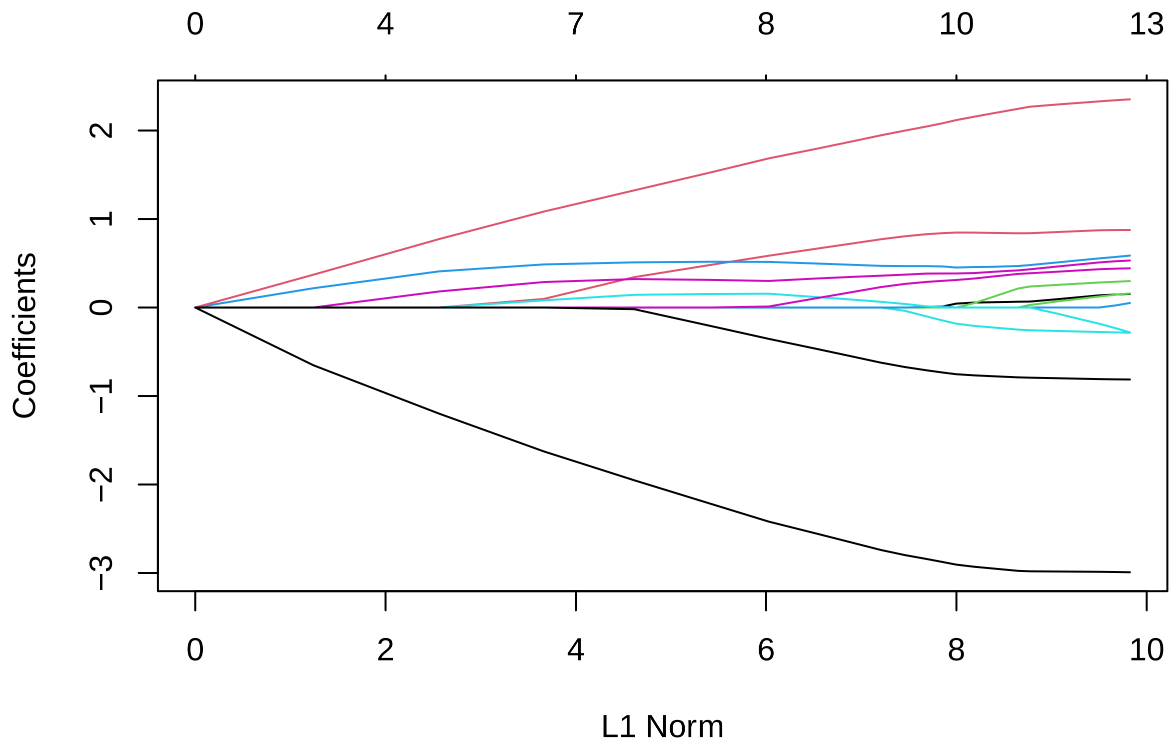
```
## [1] 14.47749
```

```r
coef <- glmnet(xfull, yfull, alpha = 0.5)
predict(coef , type = "coefficients", s = bestlam)[1:14, ]
```

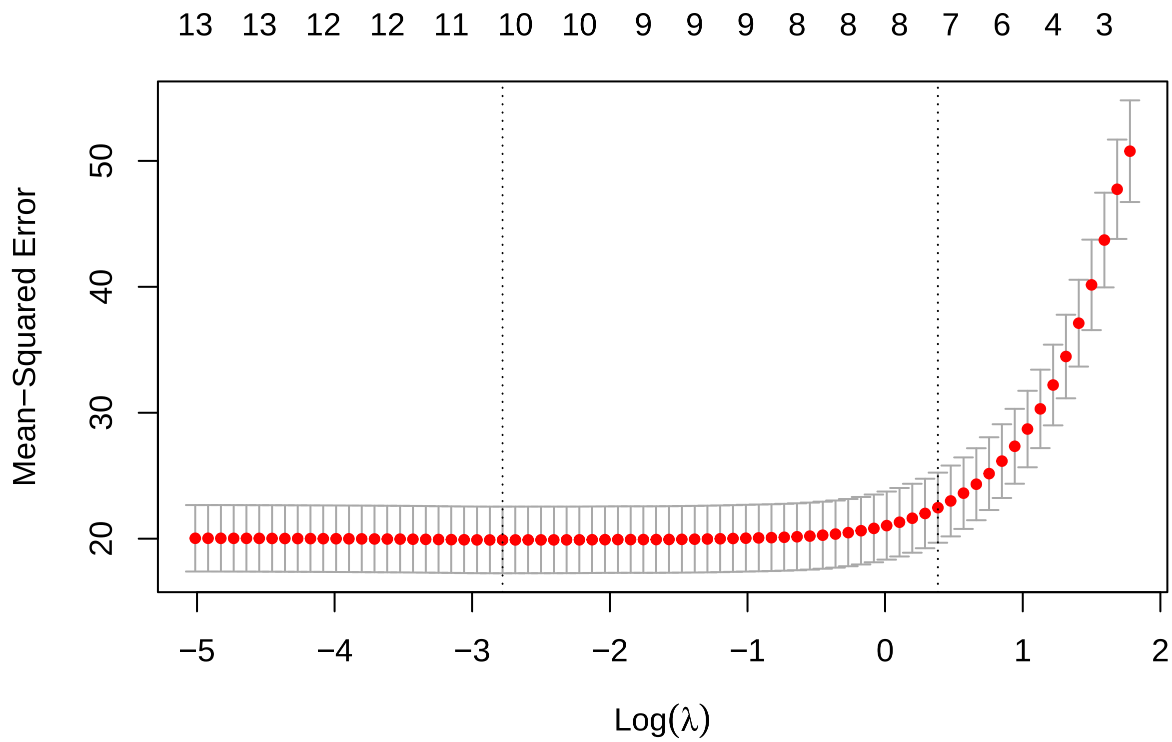
```
##   (Intercept)             X          CRIM            ZN         INDUS
##  3.139752e+01 -6.895468e-06  9.618278e-02  9.885703e-03  7.564503e-02
##          CHAS           NOX            RM           AGE           DIS
## -8.319786e-01  4.218923e+00 -4.357463e+00  8.277538e-02  2.974948e-02
##           RAD           TAX       PTRATIO             B
##  3.372231e-02  0.000000e+00  9.638423e-02 -7.942063e-03
```

```r
#Elastic-Net Regression (alpha=0.75)
en.mod <- glmnet(xtrain, ytrain, alpha = 0.75, lambda = grid)
plot(en.mod)
```

```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```




```
set.seed (1)
cv.out <- cv.glmnet(xtrain, ytrain, alpha = 0.75)
plot(cv.out)
```




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

```
## [1] 0.06206361
```

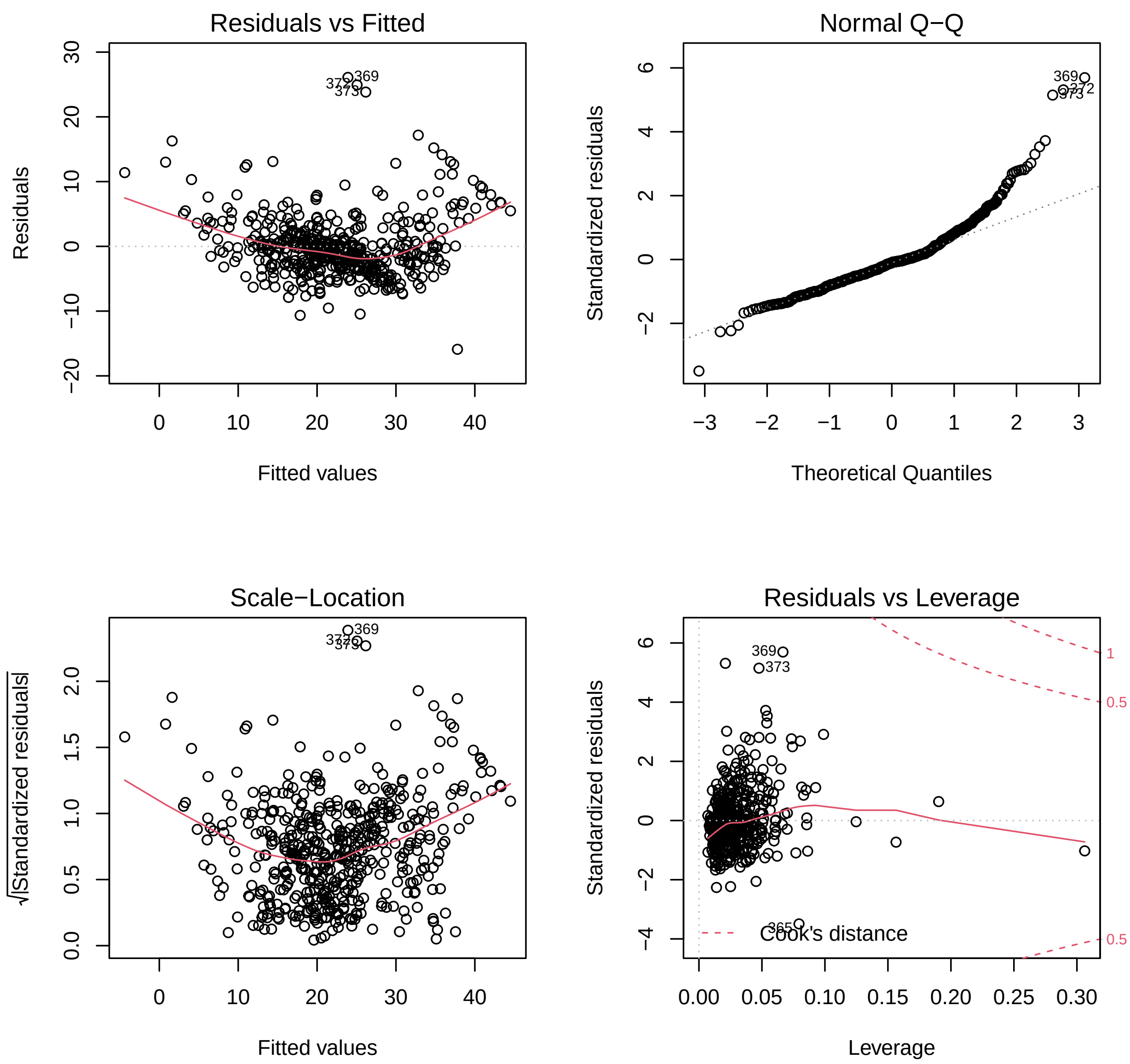
```r
en.pred <- predict(en.mod , s = bestlam,newx = xtest)
mean (( en.pred - ytest)^2)
```

```
## [1] 14.45078
```

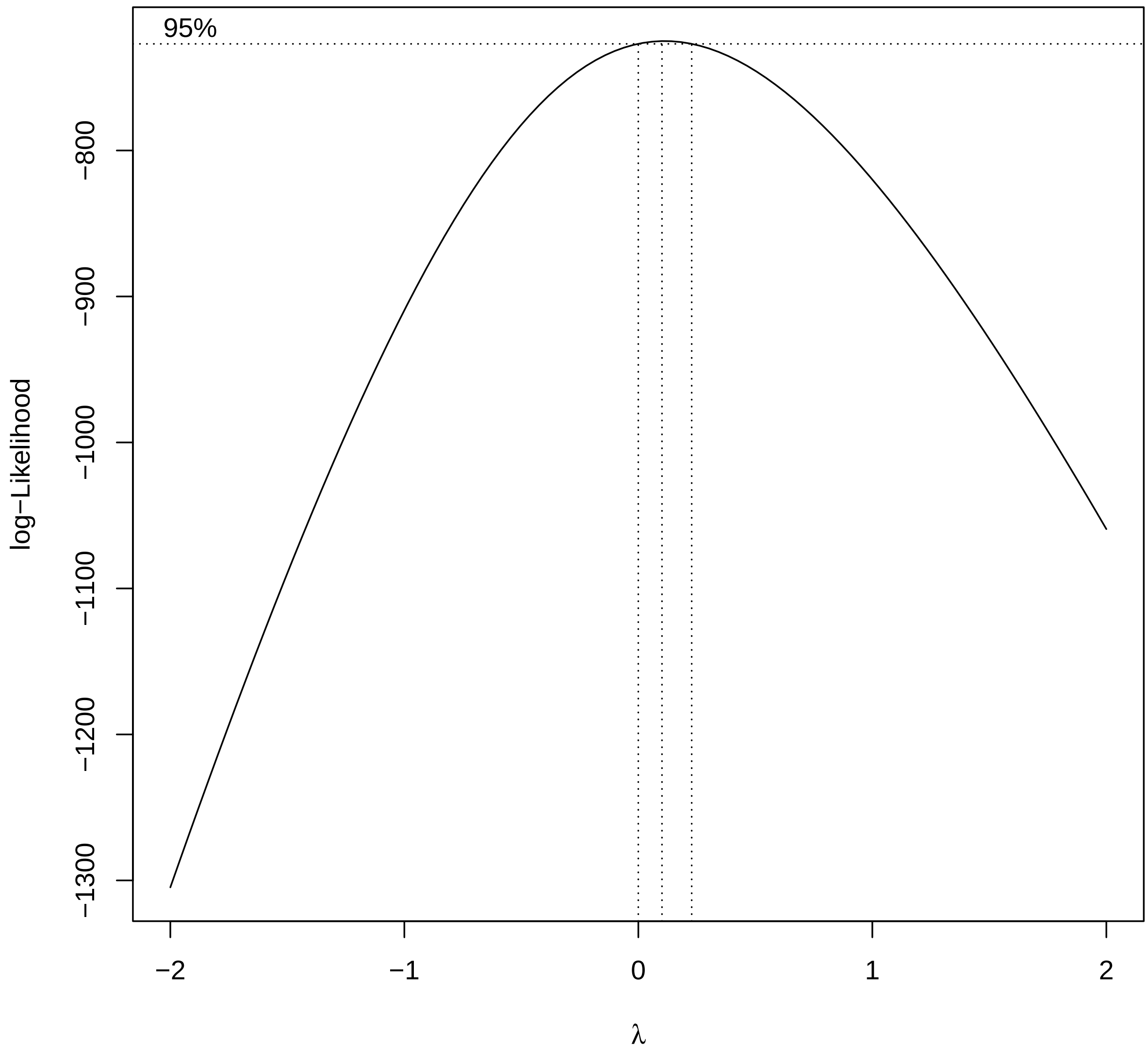
```r
coef <- glmnet(xfull, yfull, alpha = 0.75)
predict(coef , type = "coefficients", s = bestlam)[1:14, ]
```

```
## (Intercept)           X        CRIM          ZN       INDUS        CHAS
## 31.56994197  0.00000000  0.09622312  0.01020119  0.07515211 -0.83108151
##         NOX          RM         AGE         DIS         RAD         TAX
##  4.12893649 -4.38104567  0.08369280  0.03375657  0.03384228  0.00000000
##     PTRATIO           B
##  0.09408938 -0.00797361
```

```r
# To study the linearity assumption of OLS we will look at OLS although lasso regression is better.
#MSE of OLS 14.50773 is too high compared to the sample mean of response 22.532806, maybe the data isn'
par(mfrow = c(2, 2))
plot(ols.modfull)
```



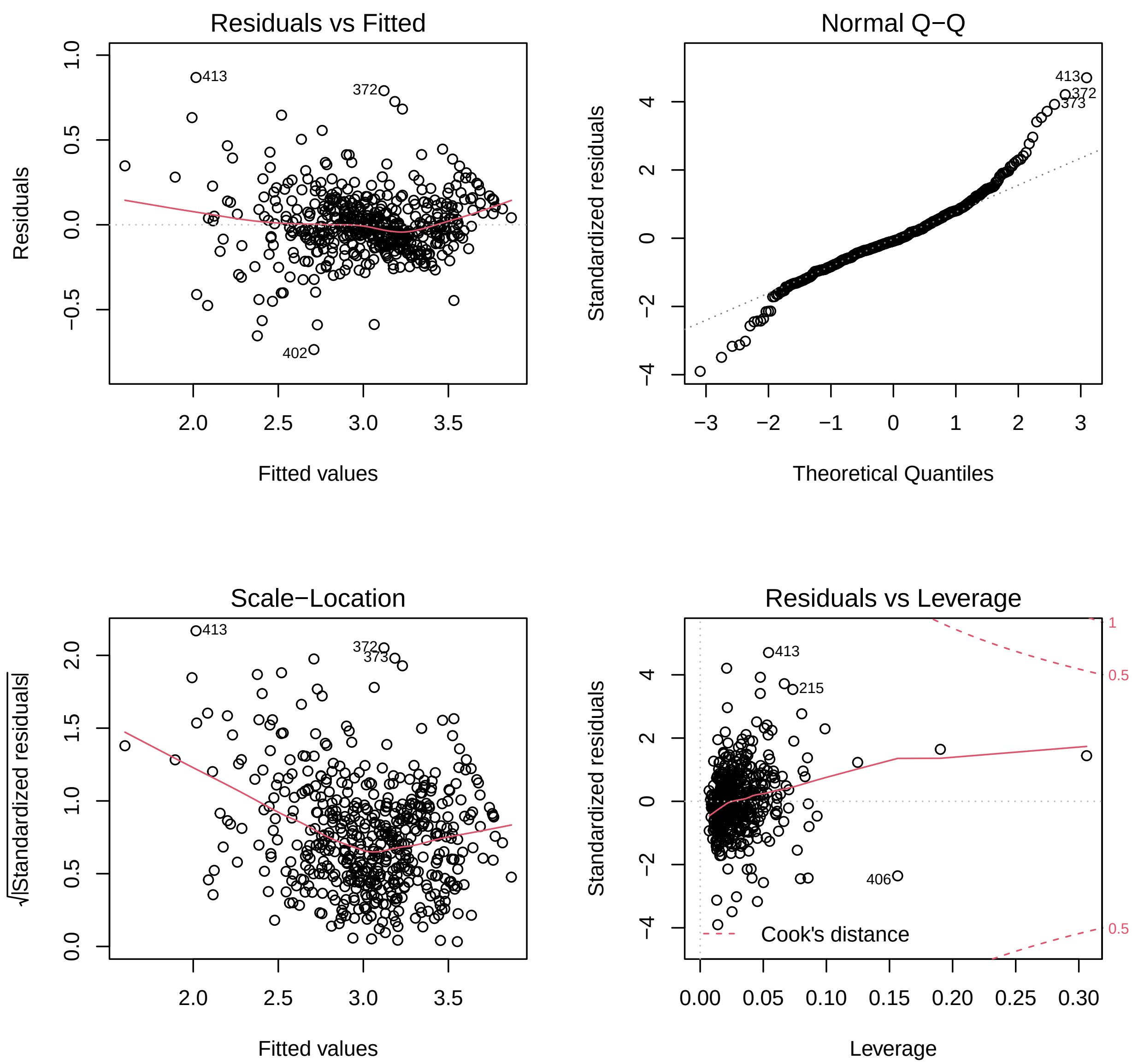

```
#There is banana shape in the residual vs fitted plot suggesting non-linearity of data, let's try boxco
par(mfrow = c(1, 1))
library(MASS)
bc=boxcox(ols.modfull)
```




```
i=which.max(bc$y)
bc$x[i]
```

```
## [1] 0.1010101
```

```
#Suggested transformation is power of close to 0, which is log transformation.
ols.modfull = lm(log(MEDV)~.,data)
par(mfrow = c(2, 2))
plot(ols.modfull)
```




```
#The transformed model still failed the non-linearity assumption, further suggesting that the true unde
```