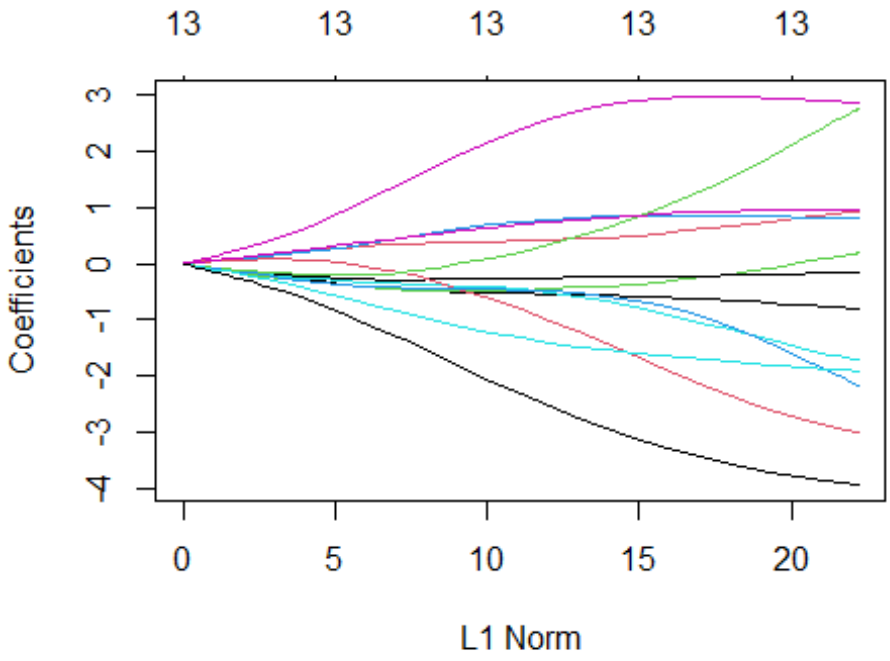
5052 Statistical Machine Learning Project

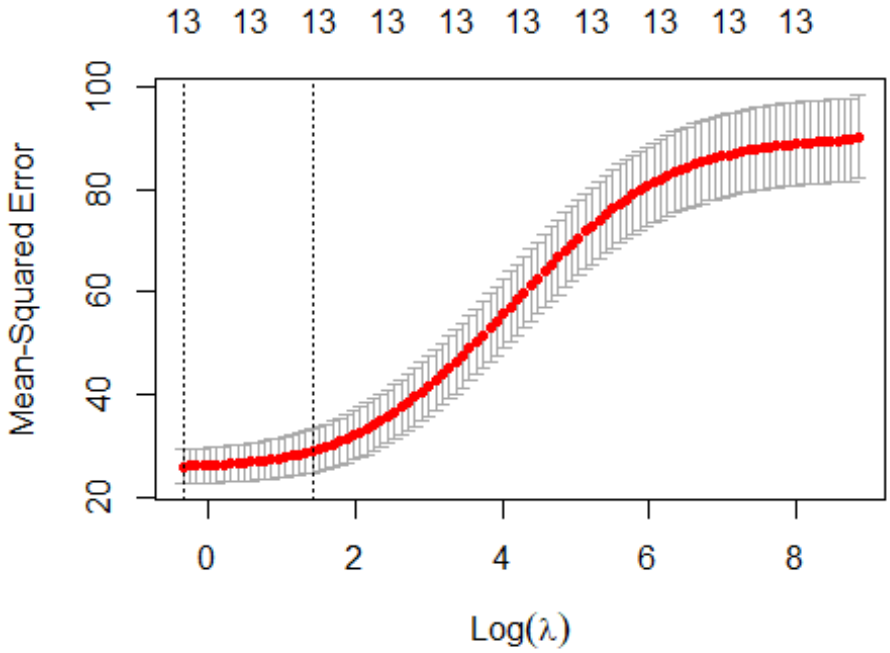
Kah Meng Soh

2022-04-24

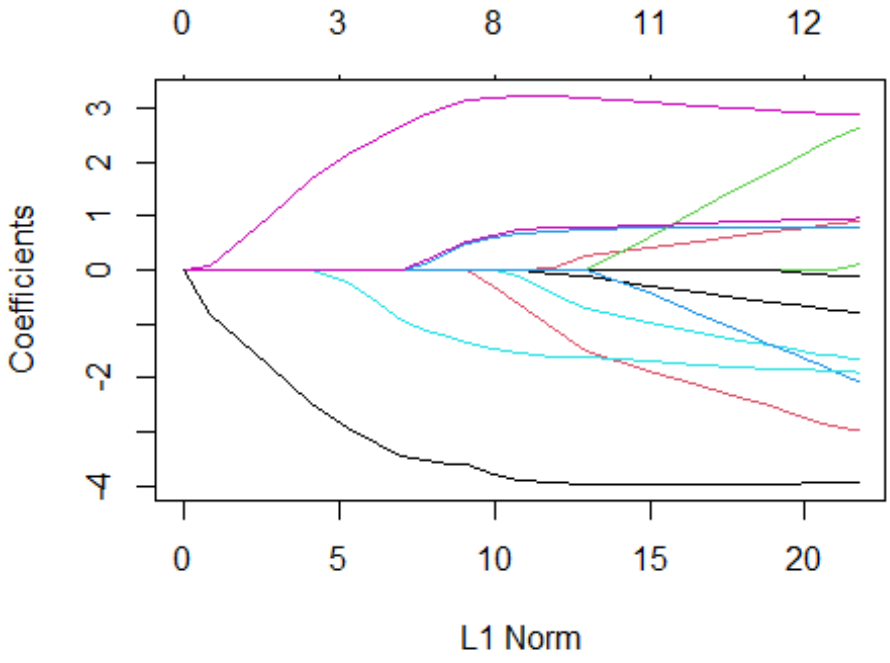
```
library(glmnet)

## 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)
```



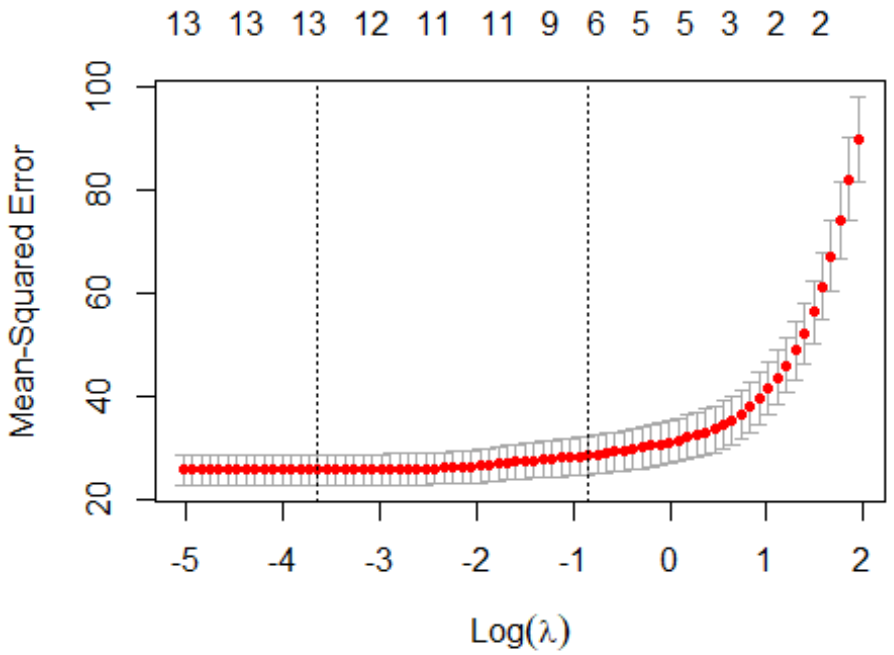

```
set.seed (1)
cv.out <- cv.glmnet(xtrain, ytrain, alpha = 0)
plot(cv.out)
```



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

## [1] 0.7014097

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

## [1] 17.13438

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

##   (Intercept)          CRIM            ZN         INDUS          CHAS
##  27.832905126  -0.087189969   0.032417614  -0.038828587   2.902056950
##           NOX            RM           AGE           DIS           RAD
## -11.787030343   4.012856001  -0.003811209  -1.109846262   0.151202898
##           TAX       PTRATIO             B         LSTAT
##  -0.005662260  -0.852615617   0.009063426  -0.470965108

#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] 0.02640763

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

## [1] 17.30783

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

##   (Intercept)          CRIM            ZN         INDUS          CHAS
##  34.534498651  -0.098911809   0.041691045   0.000000000   2.687154509
##           NOX            RM           AGE           DIS           RAD
## -16.367317555   3.863272653   0.000000000  -1.401512451   0.255320467
##           TAX       PTRATIO             B         LSTAT
##  -0.009936431  -0.930918959   0.009040888  -0.522503775

#OLS Regression
train=data.frame(train)
test=data.frame(test)
ols.mod = lm(V14~.,train)
```

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

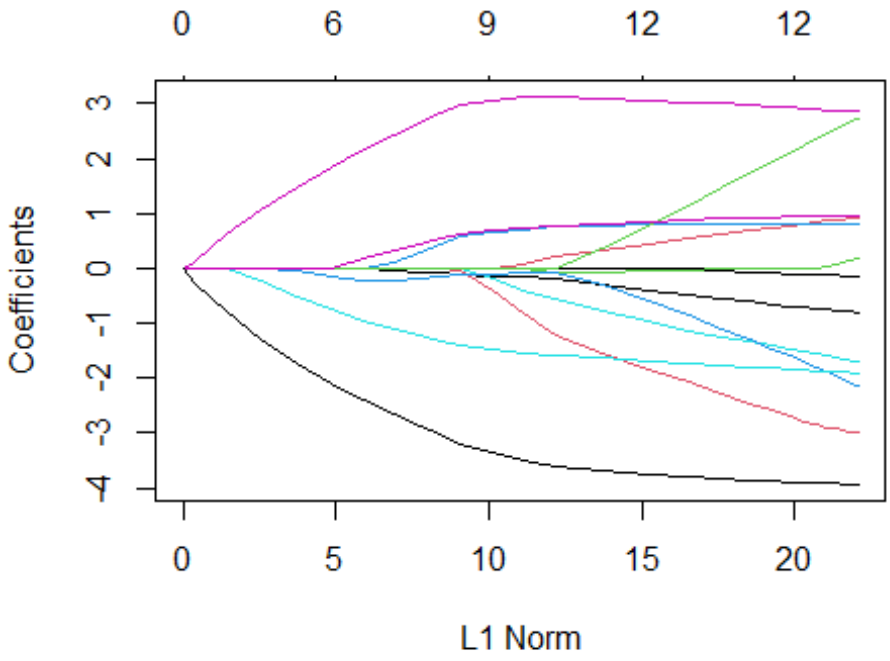
```
## [1] 17.33601
```

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

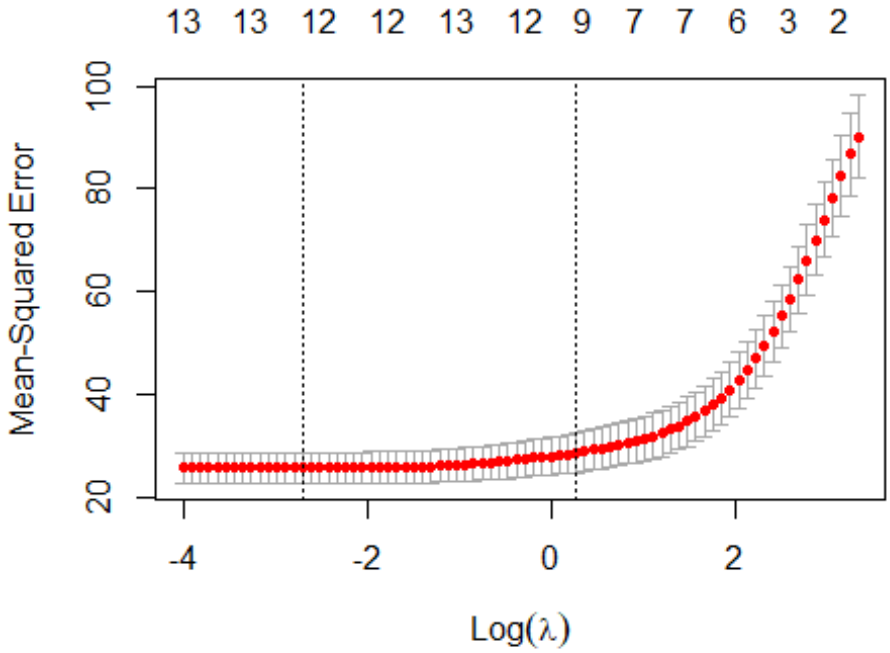
```
##
## Call:
## lm(formula = MEDV ~ ., data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.595  -2.730  -0.518   1.777  26.199
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
## CRIM        -1.080e-01  3.286e-02  -3.287 0.001087 **
## ZN           4.642e-02  1.373e-02   3.382 0.000778 ***
## INDUS        2.056e-02  6.150e-02   0.334 0.738288
## CHAS         2.687e+00  8.616e-01   3.118 0.001925 **
## NOX         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
## RM           3.810e+00  4.179e-01   9.116  < 2e-16 ***
## AGE          6.922e-04  1.321e-02   0.052 0.958229
## DIS         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
## RAD          3.060e-01  6.635e-02   4.613 5.07e-06 ***
## TAX         -1.233e-02  3.760e-03  -3.280 0.001112 **
## PTRATIO     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
## B            9.312e-03  2.686e-03   3.467 0.000573 ***
## LSTAT       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.745 on 492 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7338
## F-statistic: 108.1 on 13 and 492 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)
```

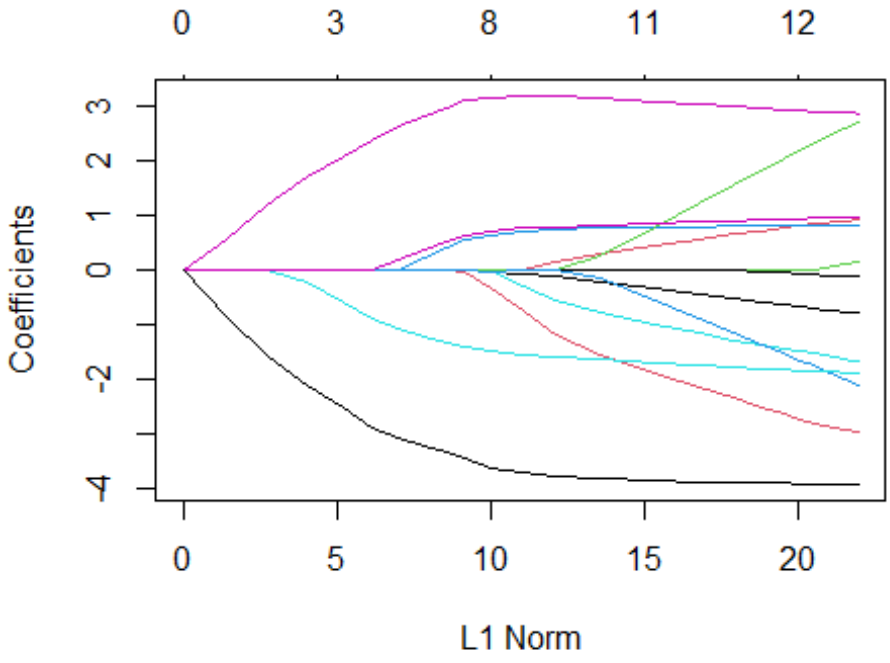
```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```



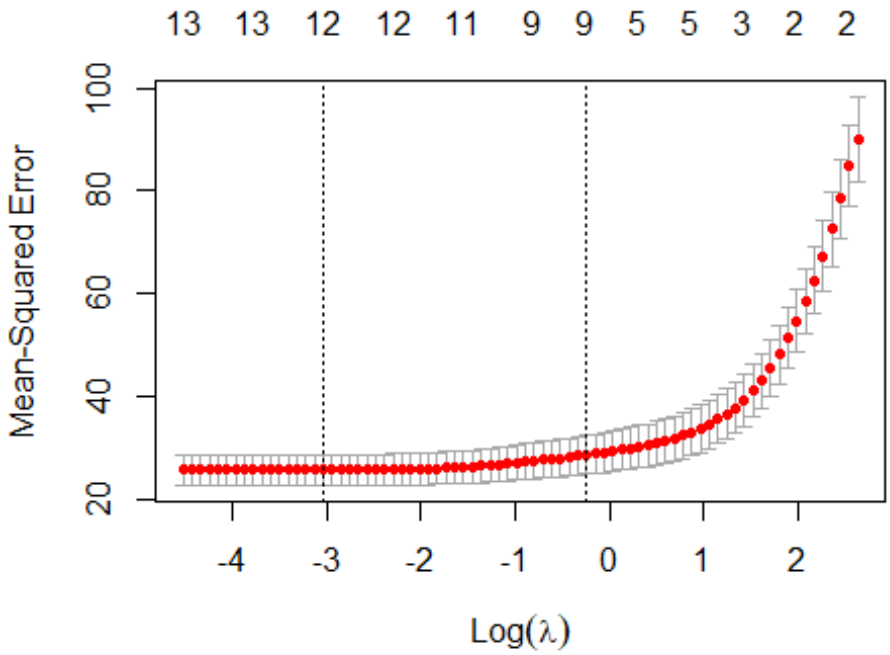

```
set.seed (1)
cv.out <- cv.glmnet(xtrain, ytrain, alpha = 0.25)
plot(cv.out)
```



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

## [1] 0.06633903

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

## [1] 17.28993

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

##   (Intercept)          CRIM            ZN         INDUS          CHAS
##  34.344806370  -0.100031837   0.041942716   0.000000000   2.714860720
##           NOX            RM           AGE           DIS           RAD
## -16.296787282   3.867940286   0.000000000  -1.394731303   0.255435991
##           TAX       PTRATIO             B         LSTAT
##  -0.009938417  -0.929608035   0.009123188  -0.519268916

#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
```

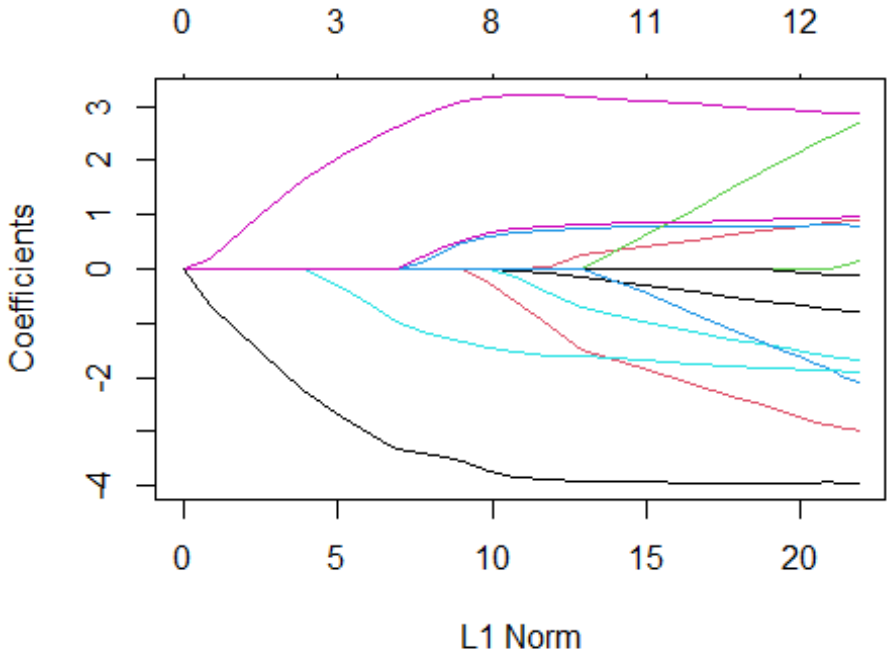


```
set.seed (1)
cv.out <- cv.glmnet(xtrain, ytrain, alpha = 0.5)
plot(cv.out)
```

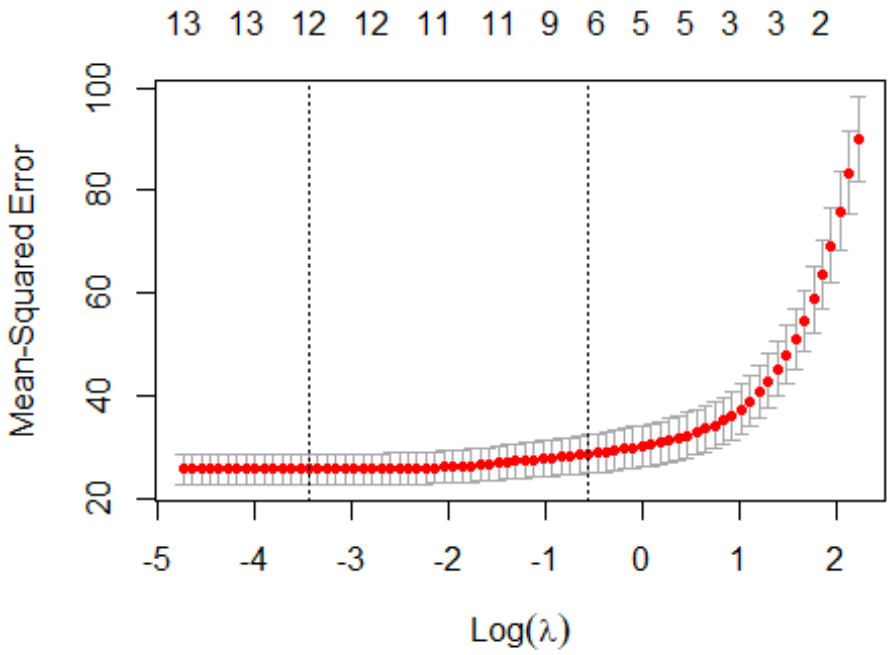



```
bestlam <- cv.out$lambda.min
bestlam

## [1] 0.04812329

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

## [1] 17.30388

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

##   (Intercept)          CRIM            ZN         INDUS          CHAS
##  34.297687865  -0.098686899   0.041418273   0.000000000   2.697137494
##           NOX            RM           AGE           DIS           RAD
## -16.251186383   3.869843374   0.000000000  -1.390058404   0.251589960
##           TAX       PTRATIO             B         LSTAT
##  -0.009769944  -0.929183357   0.009057632  -0.521011571

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

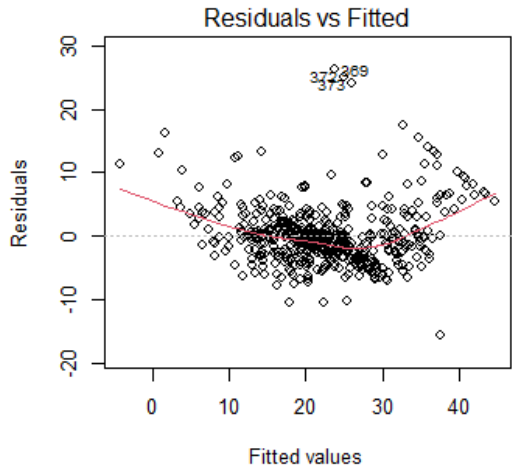
```
## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):
## collapsing to unique 'x' values
```

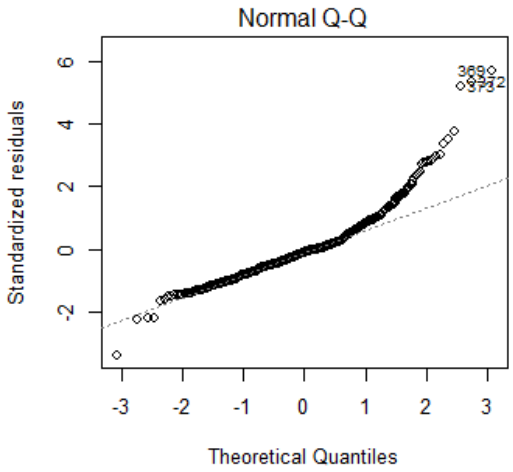


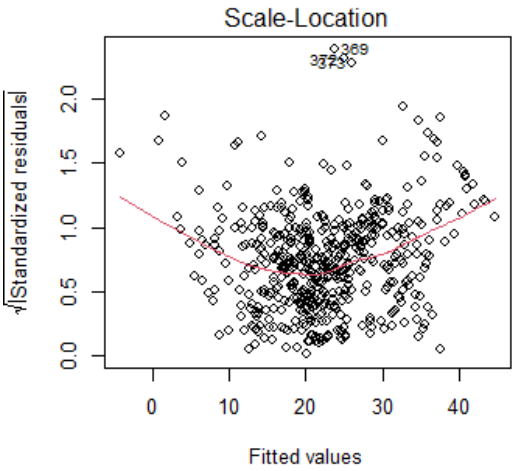

```
set.seed (1)
cv.out <- cv.glmnet(xtrain, ytrain, alpha = 0.75)
plot(cv.out)
```

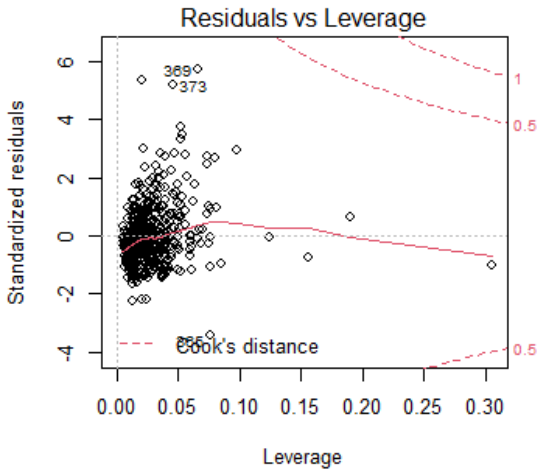



```
bestlam <- cv.out$lambda.min
bestlam

## [1] 0.03208219

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

## [1] 17.29769

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

##   (Intercept)          CRIM            ZN         INDUS          CHAS
##  34.585155585  -0.099604188   0.041754881   0.000000000   2.693163356
##           NOX            RM           AGE           DIS           RAD
## -16.424465605   3.862139544   0.000000000  -1.402668918   0.256969564
##           TAX       PTRATIO             B         LSTAT
##  -0.009975095  -0.932139576   0.009063270  -0.521832502

# 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't fit for linear regression, let's see if some
underlying assumption of OLS violated.
par(mfrow = c(2, 2))
plot(ols.modfull)
```

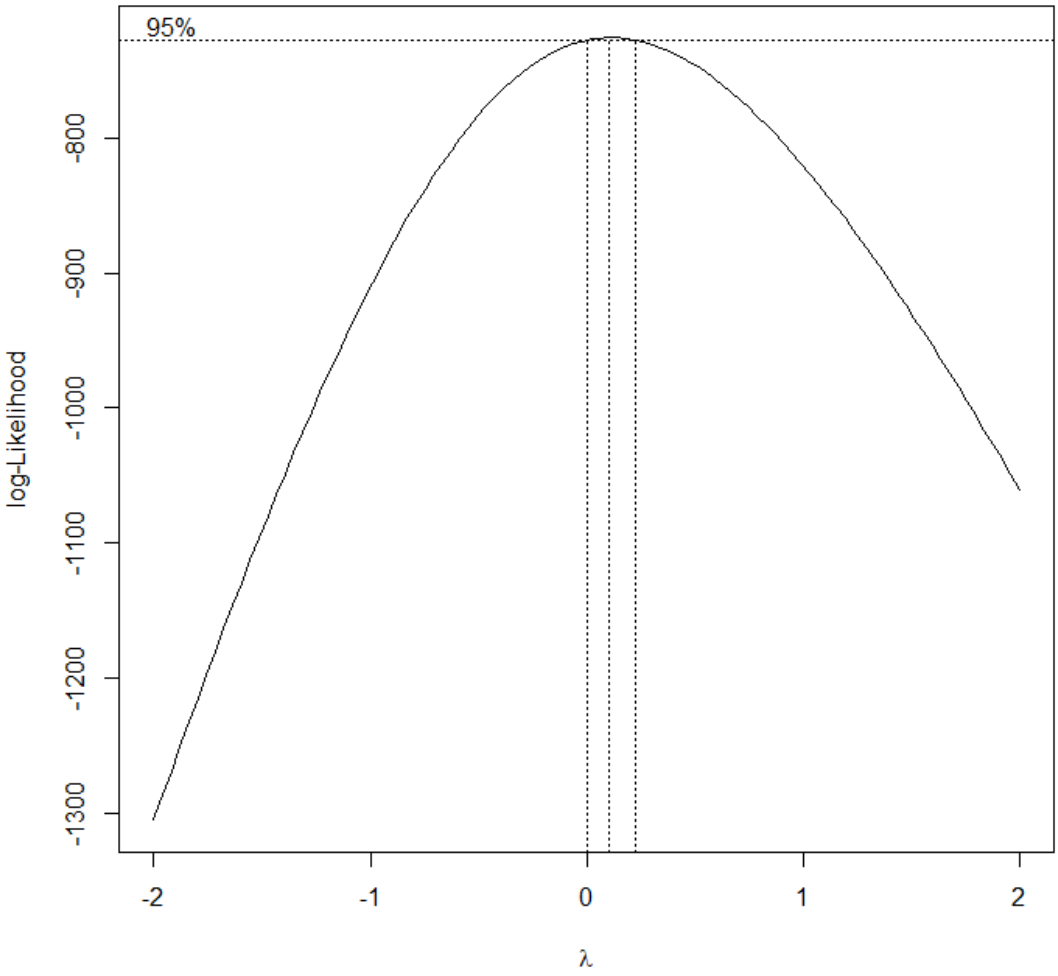




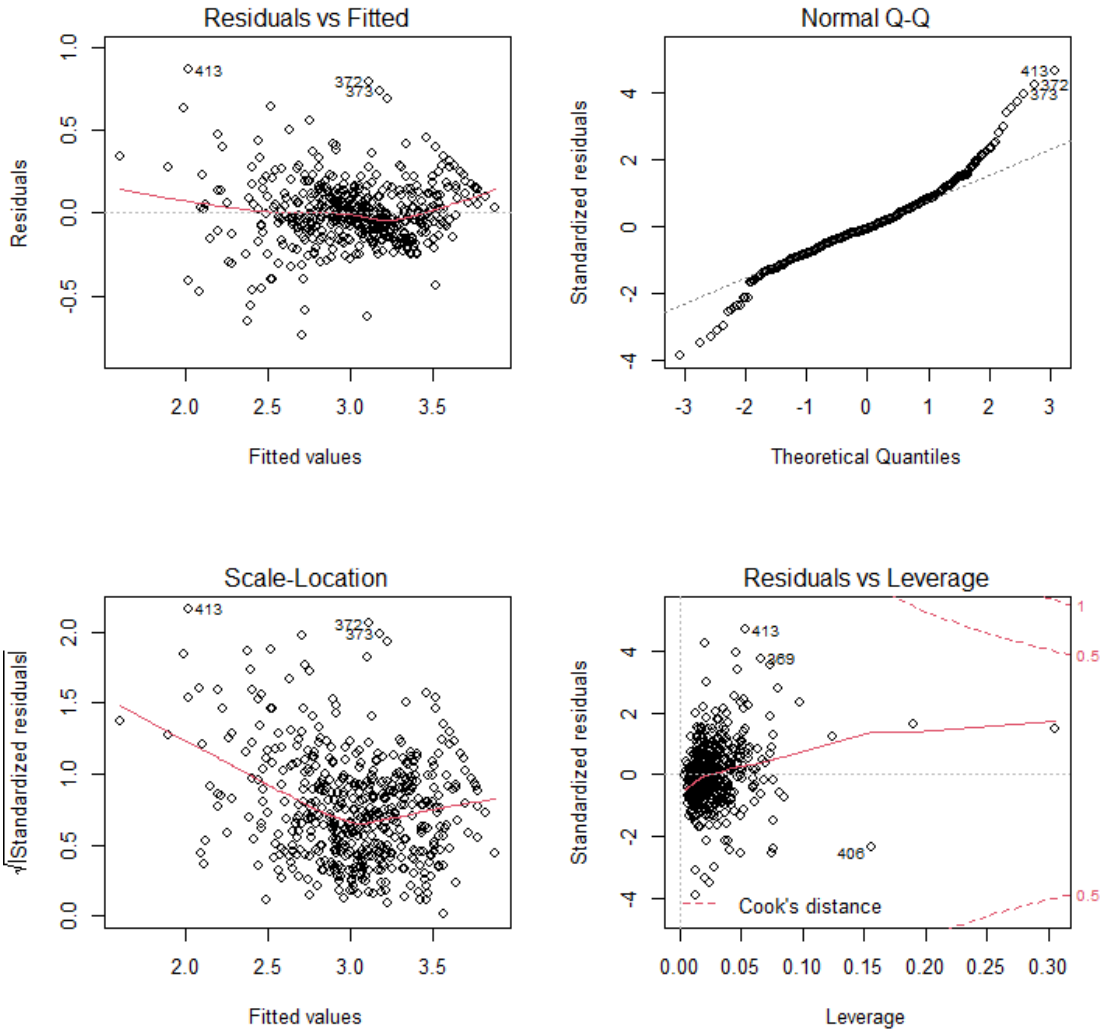





```
#There is banana shape in the residual vs fitted plot suggesting non-
linearity of data, let's try boxcox transformation just to be sure.
par(mfrow = c(1, 1))
library(MASS)
bc=boxcox(ols.modfull)
```




```
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 underlying model is non-linear.
```