Ridge and Lasso regression

GLMNET is used for ridge and lasso regression

```
## Loaded lars 1.2
```

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

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

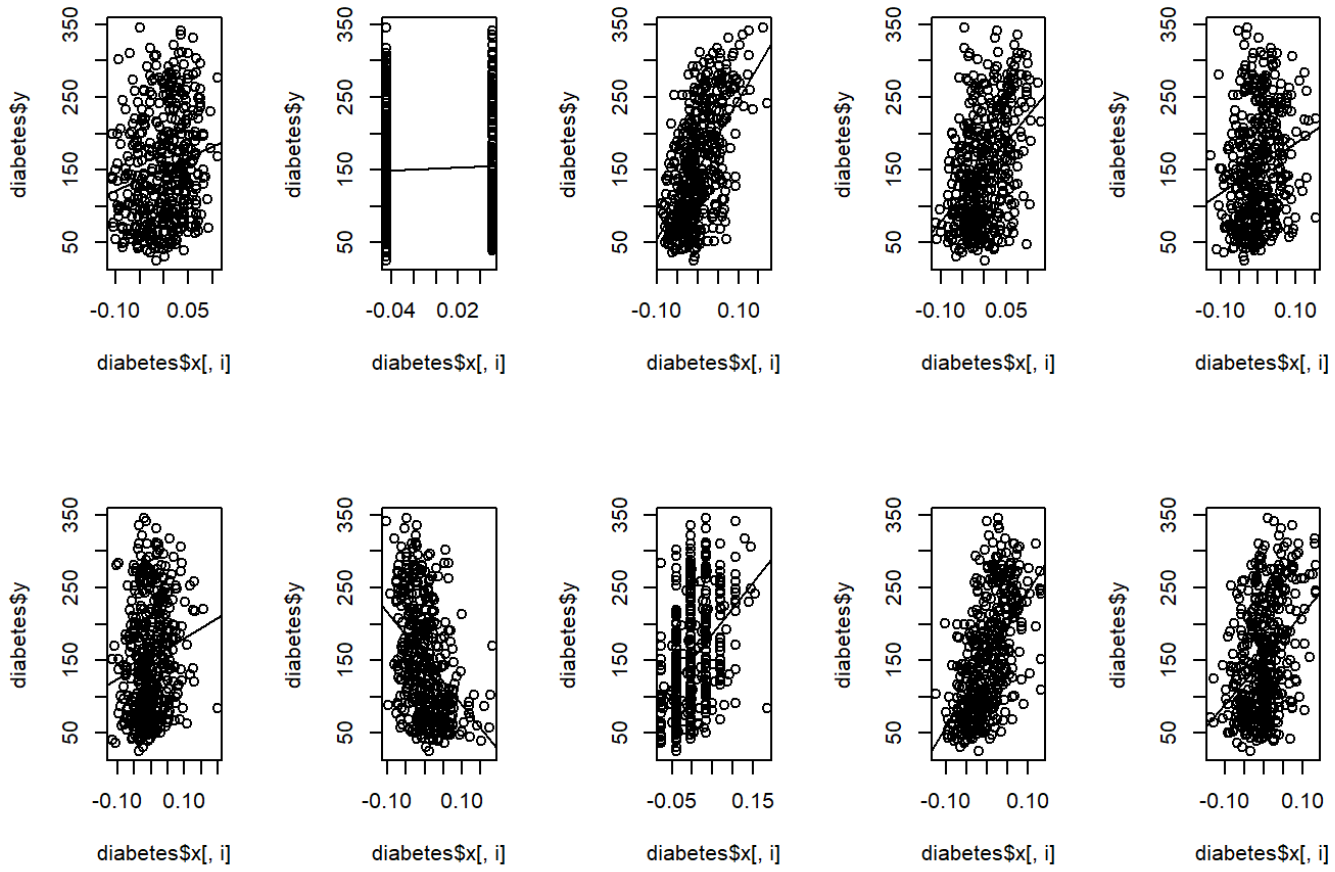
```
## Warning: package 'foreach' was built under R version 3.4.4
```

```
## Loaded glmnet 2.0-13
```

Diabetes is a dataset contain variables related to diabetes

Ploting all the graph between X and y.(As X contains 10 variables so loop goes for 1 to 10 for all the variables)

```
par(mfrow = c(2,5))
for(i in 1:10){
  plot(diabetes$x[,i],diabetes$y, type = 'p' , xlab = names(diabetes$x[i]))
abline(lm(y ~ x[,i], data = diabetes))
}
```



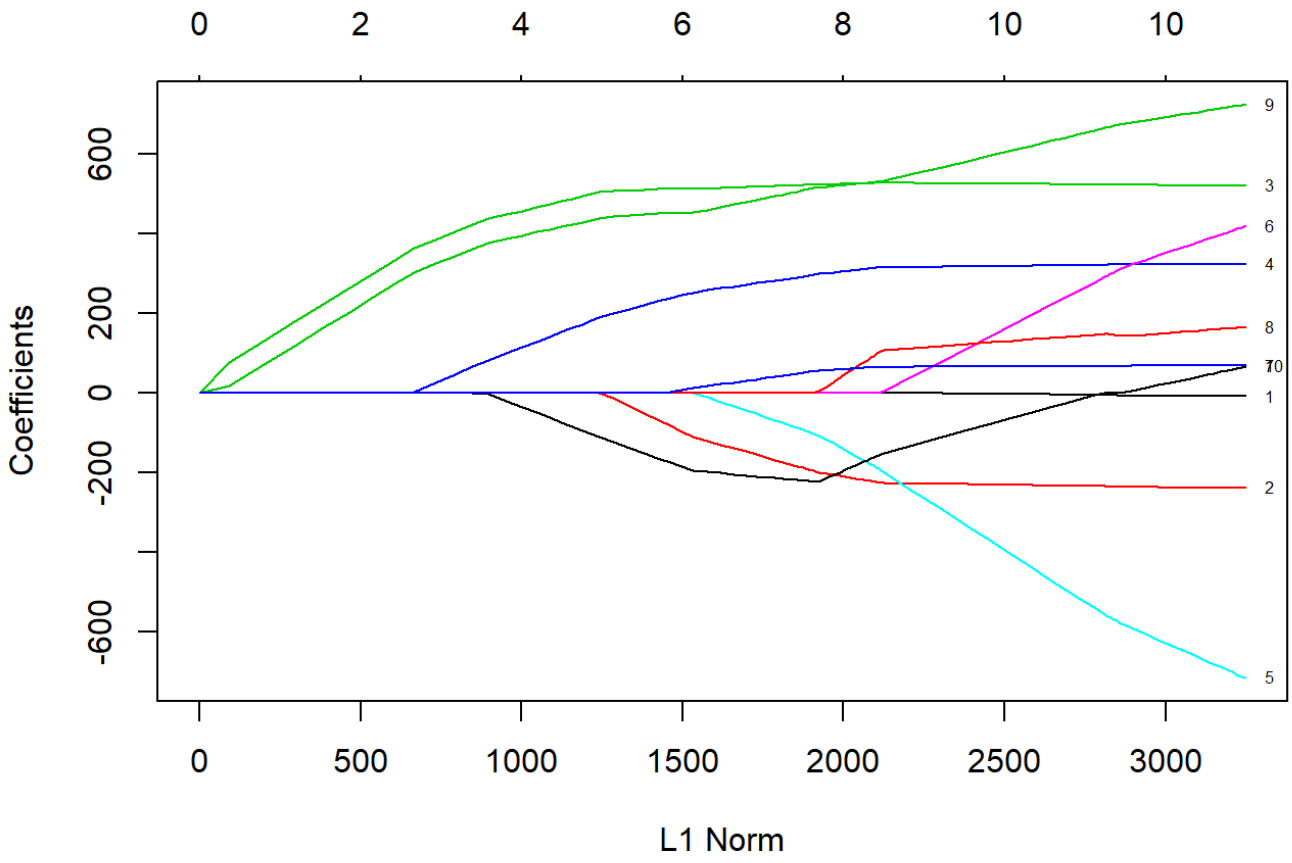
Linear model by lm function.Y as a funtion of x

```
lin <- lm(y ~ x, data = diabetes)
summary(lin)
```

```
##
## Call:
## lm(formula = y ~ x, data = diabetes)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -155.829  -38.534   -0.227   37.806  151.355
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  152.133      2.576  59.061  < 2e-16 ***
## xage         -10.012     59.749  -0.168 0.867000
## xsex        -239.819     61.222  -3.917 0.000104 ***
## xbmi         519.840     66.534   7.813 4.30e-14 ***
## xmap         324.390     65.422   4.958 1.02e-06 ***
## xtc         -792.184    416.684  -1.901 0.057947 .
## xldl         476.746    339.035   1.406 0.160389
## xhdl         101.045    212.533   0.475 0.634721
## xtch         177.064    161.476   1.097 0.273456
## xltg         751.279    171.902   4.370 1.56e-05 ***
## xglu          67.625     65.984   1.025 0.305998
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54.15 on 431 degrees of freedom
## Multiple R-squared:  0.5177, Adjusted R-squared:  0.5066
## F-statistic: 46.27 on 10 and 431 DF,  p-value: < 2.2e-16
```

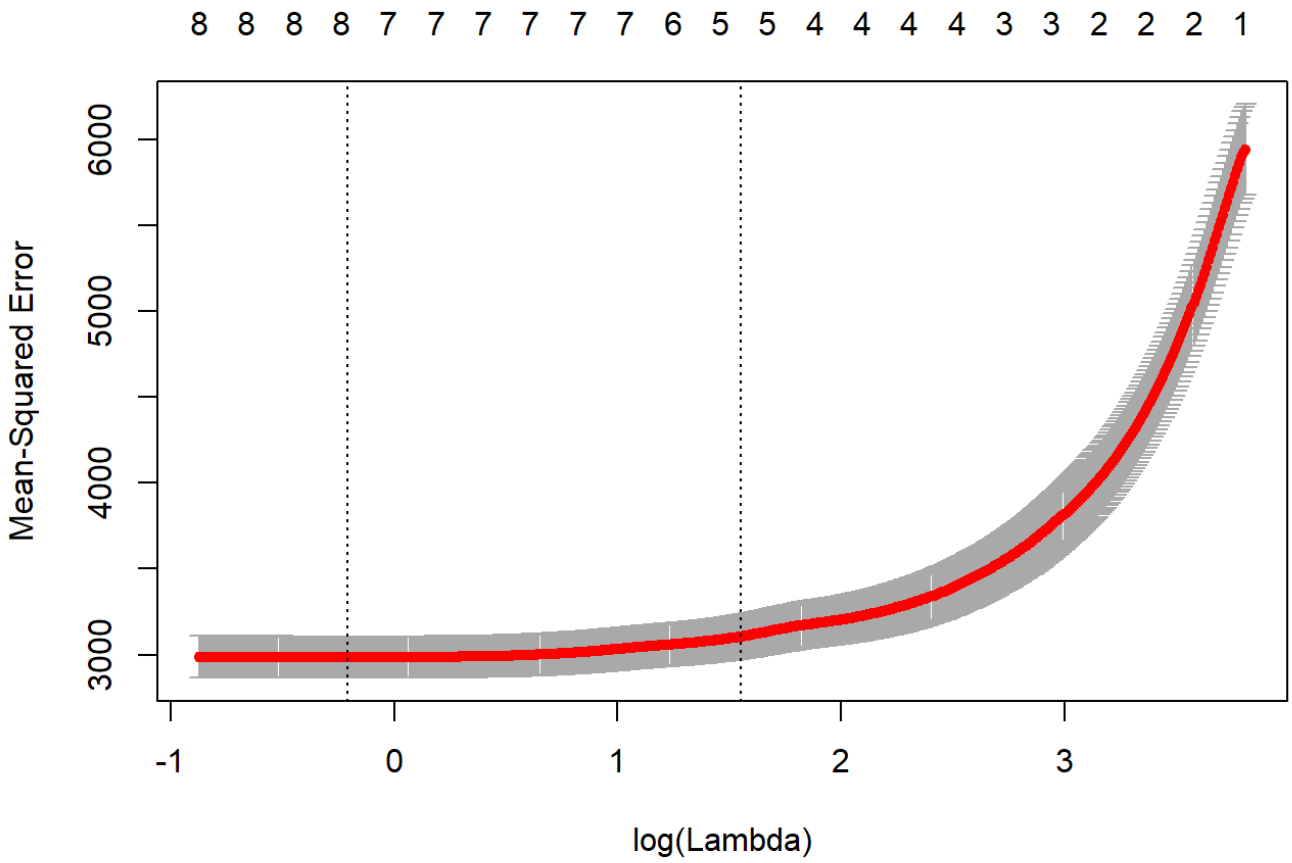
This will show all residuals and Coefficients.
The values in Coefficients shows that they are highly correlated with the Y. More the number of stars more significant the values is.
. shows the significant values but lesser than * values.
Value here of R squared is near to 0.5 that means they are not highly correlated.As predictor jointly explain 51% of observed varience.

```
par(mfrow= c(1,1))
glm <- glmnet(diabetes$x,diabetes$y, alpha = 1)
plot.glmnet(glm, xvar = 'norm', label = T)
```



glmnet is used for LASSO and Ridge. Alpha 1 means LASSO and alpha 0 means Ridge.

```
cvg <- cv.glmnet(diabetes$x,diabetes$y, alpha = 1, nlambda = 1000)
plot.cv.glmnet(cvg)
```



Cross validation of LASSO with the help of cv.glmnet function.
Used to get better value of lamdda for better fitting of equation

It will show all the values of mean squred error as lambda increases.

```
cvg$lambda.min
```

```
## [1] 0.8109769
```

```
fit <- glmnet(diabetes$x,diabetes$y, alpha = 1, lambda = cvg$lambda.min)
fit$beta
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##             s0
## age    .
## sex -203.61798
## bmi  523.01887
## map  300.64589
## tc  -119.77932
## ldl    .
## hdl -212.78093
## tch   18.08458
## ltg  517.38458
## glu   56.70284
```

LASSO regression using lambda value as the minimun value of lambda

Now try with new lambda with in one standard error

```
cvg$lambda.1se
```

```
## [1] 4.718166
```

```
fit <- glmnet(diabetes$x,diabetes$y, alpha = 1, lambda =  cvg$lambda.1se)
fit$beta
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##             s0
## age    .
## sex  -56.08781
## bmi  509.89411
## map  223.38374
## tc     .
## ldl    .
## hdl -155.74813
## tch    .
## ltg  447.88560
## glu    .
```

We can observe that with this lambda value few beta values has been changed and few are compresed to zero.But we can observe the differnce
Now only few most significant values are left.this will reduce complexity but meanwhile increases mean sqaure error if we use this model on train data and predict on test data

Linear model by lm function.Y as a funtion of X2

```
model_ols2 <- lm(diabetes$y ~ diabetes$x2)

summary(model_ols2)
```

```
##
## Call:
## lm(formula = diabetes$y ~ diabetes$x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -158.216  -30.809   -3.857   31.348  153.946
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          152.133      2.532  60.086  < 2e-16 ***
## diabetes$x2age        50.721     65.513   0.774   0.4393
## diabetes$x2sex      -267.344     65.270  -4.096 5.15e-05 ***
## diabetes$x2bmi       460.721     84.601   5.446 9.32e-08 ***
## diabetes$x2map       342.933     72.447   4.734 3.13e-06 ***
## diabetes$x2tc      -3599.542  60575.187  -0.059   0.9526
## diabetes$x2ldl      3028.281  53238.699   0.057   0.9547
## diabetes$x2hdl      1103.047  22636.179   0.049   0.9612
```

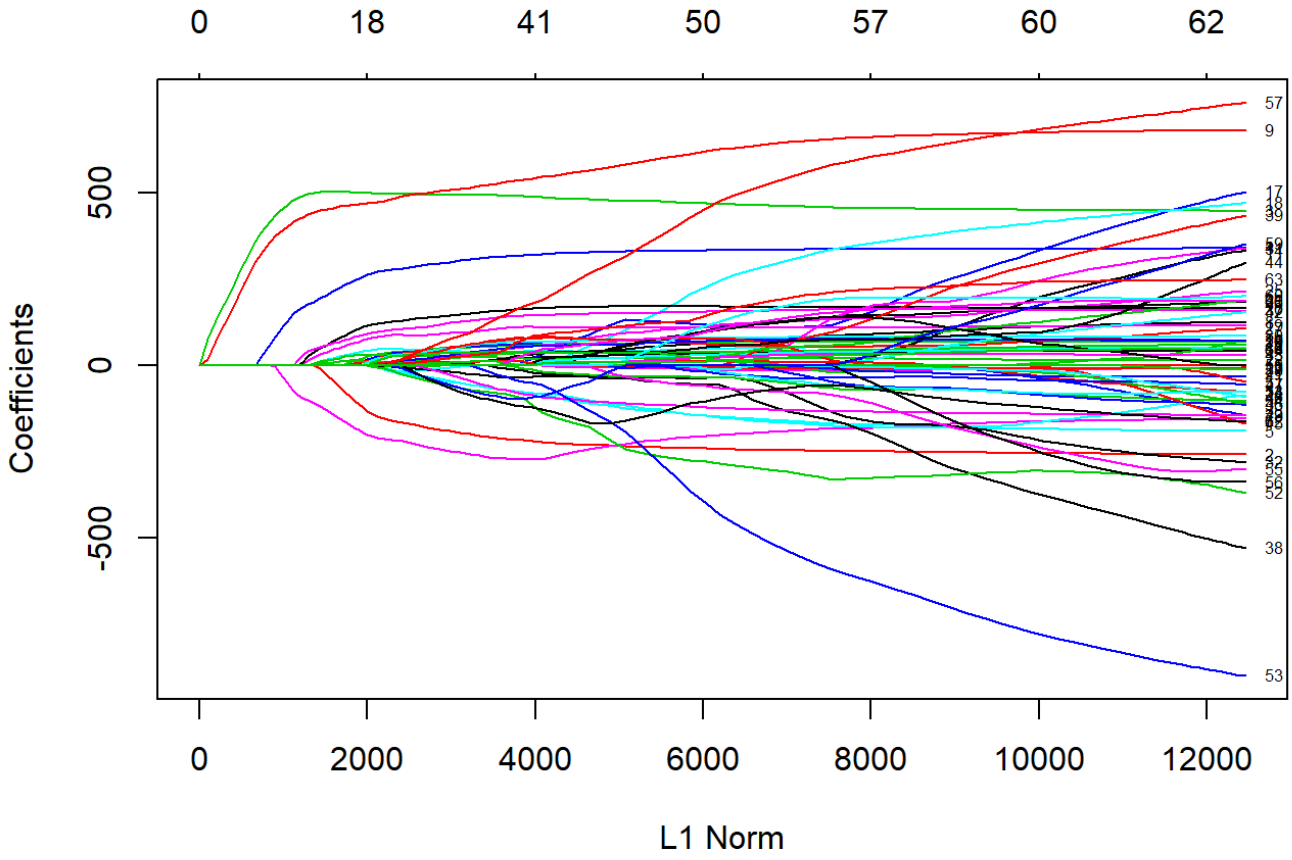
```
## diabetes$x2tch        74.937    275.807   0.272   0.7860
## diabetes$x2ltg      1828.210  19914.504   0.092   0.9269
## diabetes$x2glu        62.754     70.398   0.891   0.3733
## diabetes$x2age^2      67.691     69.470   0.974   0.3305
## diabetes$x2bmi^2      45.849     83.288   0.550   0.5823
## diabetes$x2map^2      -8.460     71.652  -0.118   0.9061
## diabetes$x2tc^2     6668.449   7059.159   0.945   0.3454
## diabetes$x2ldl^2    3583.174   5326.148   0.673   0.5015
## diabetes$x2hdl^2    1731.821   1590.574   1.089   0.2769
## diabetes$x2tch^2     773.374    606.967   1.274   0.2034
## diabetes$x2ltg^2    1451.581   1730.103   0.839   0.4020
## diabetes$x2glu^2     114.149     94.122   1.213   0.2260
## diabetes$x2age:sex   148.678     73.407   2.025   0.0435 *
## diabetes$x2age:bmi   -18.052     79.620  -0.227   0.8208
## diabetes$x2age:map    18.534     76.303   0.243   0.8082
## diabetes$x2age:tc   -158.891    617.109  -0.257   0.7970
## diabetes$x2age:ldl   -67.285    494.527  -0.136   0.8918
## diabetes$x2age:hdl   209.245    280.614   0.746   0.4563
## diabetes$x2age:tch   184.960    210.330   0.879   0.3798
## diabetes$x2age:ltg   124.667    223.765   0.557   0.5778
## diabetes$x2age:glu    62.575     80.377   0.779   0.4367
## diabetes$x2sex:bmi    64.612     77.902   0.829   0.4074
## diabetes$x2sex:map    88.472     74.744   1.184   0.2373
## diabetes$x2sex:tc    433.598    590.709   0.734   0.4634
## diabetes$x2sex:ldl  -352.823    468.951  -0.752   0.4523
## diabetes$x2sex:hdl  -124.731    273.870  -0.455   0.6491
## diabetes$x2sex:tch  -131.223    199.714  -0.657   0.5115
## diabetes$x2sex:ltg  -118.995    226.493  -0.525   0.5996
## diabetes$x2sex:glu    45.758     73.650   0.621   0.5348
## diabetes$x2bmi:map   154.720     86.340   1.792   0.0739 .
## diabetes$x2bmi:tc   -302.045    667.930  -0.452   0.6514
## diabetes$x2bmi:ldl   241.540    561.026   0.431   0.6671
## diabetes$x2bmi:hdl   121.942    329.884   0.370   0.7118
## diabetes$x2bmi:tch   -33.445    230.836  -0.145   0.8849
## diabetes$x2bmi:ltg   114.673    255.987   0.448   0.6544
## diabetes$x2bmi:glu    23.377     91.037   0.257   0.7975
## diabetes$x2map:tc    478.303    682.264   0.701   0.4837
## diabetes$x2map:ldl  -326.740    574.317  -0.569   0.5697
## diabetes$x2map:hdl  -187.305    309.589  -0.605   0.5455
## diabetes$x2map:tch   -58.294    198.601  -0.294   0.7693
## diabetes$x2map:ltg  -154.795    271.966  -0.569   0.5696
## diabetes$x2map:glu  -133.476     91.314  -1.462   0.1447
## diabetes$x2tc:ldl  -9313.775  11771.220  -0.791   0.4293
## diabetes$x2tc:hdl  -3932.025   3816.572  -1.030   0.3036
## diabetes$x2tc:tch  -2205.910   1761.843  -1.252   0.2113
## diabetes$x2tc:ltg  -3801.442  13166.091  -0.289   0.7729
## diabetes$x2tc:glu   -176.295    595.459  -0.296   0.7673
## diabetes$x2ldl:hdl  2642.645   3165.926   0.835   0.4044
## diabetes$x2ldl:tch  1206.822   1470.512   0.821   0.4123
## diabetes$x2ldl:ltg  2773.697  10960.214   0.253   0.8004
## diabetes$x2ldl:glu    85.626    505.102   0.170   0.8655
## diabetes$x2hdl:tch  1188.406   1002.242   1.186   0.2365
## diabetes$x2hdl:ltg  1467.845   4609.793   0.318   0.7503
## diabetes$x2hdl:glu   217.541    296.749   0.733   0.4640
## diabetes$x2tch:ltg   389.805    624.671   0.624   0.5330
## diabetes$x2tch:glu   235.693    235.064   1.003   0.3167
## diabetes$x2ltg:glu    83.525    264.726   0.316   0.7525
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 53.23 on 377 degrees of freedom
## Multiple R-squared:  0.5924, Adjusted R-squared:  0.5233
## F-statistic: 8.563 on 64 and 377 DF,  p-value: < 2.2e-16
```

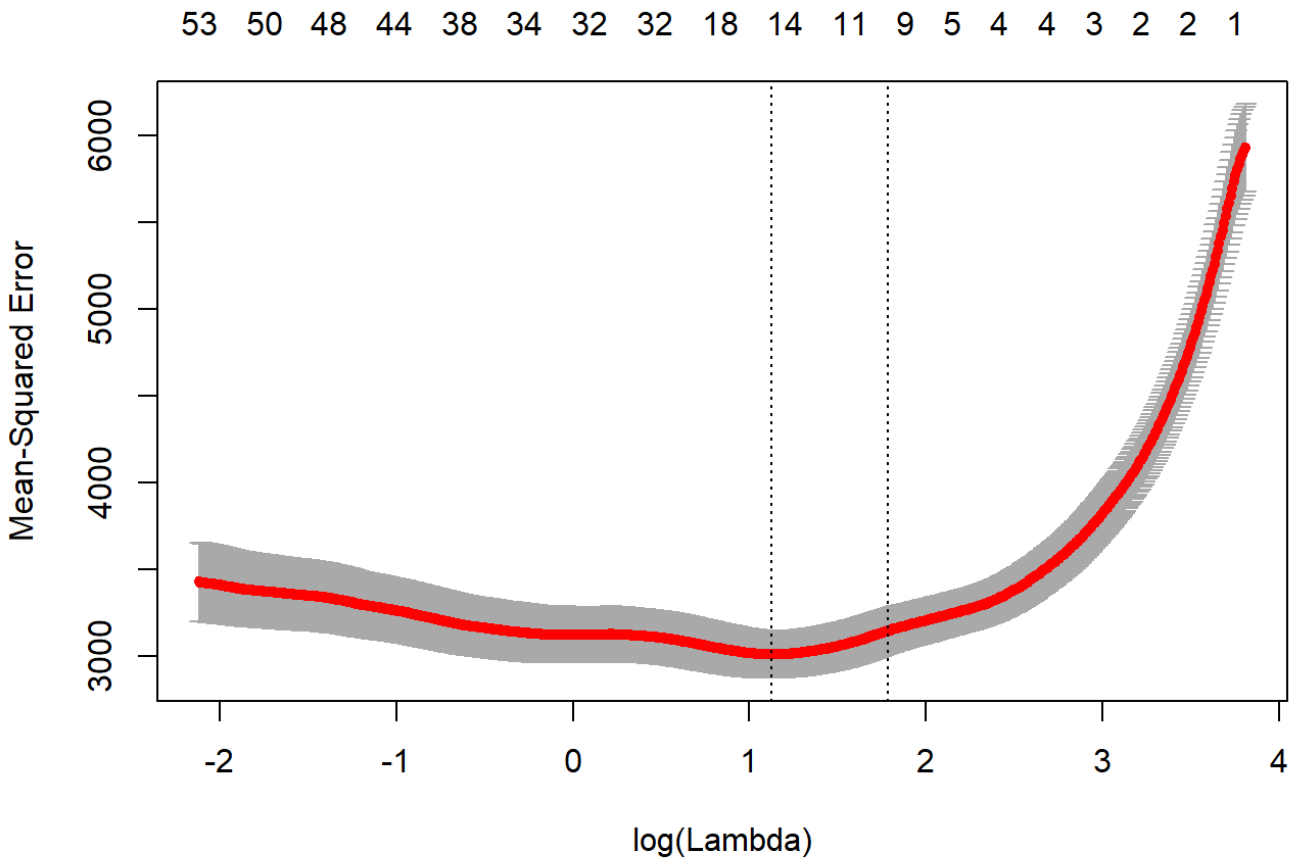
```
model_lasso1 <- glmnet(diabetes$x2, diabetes$y)
```

glmnet is used for LASSO and Ridge.Alpha 1 means LASSO and alpha 0 means Ridge.

```
plot.glmnet(model_lasso1, xvar = "norm", label = T)
```




```
cv_fit1 <- cv.glmnet(diabetes$x2,diabetes$y, alpha = 1, nlambda = 1000)
plot.cv.glmnet(cv_fit1)
```




```
fit1 <- glmnet(diabetes$x2,diabetes$y, alpha = 1, lambda = cv_fit1$lambda.min)
```

Cross validation of LASSO with the help of cv.glmnet function.
Used to get better value of lamdda for better fitting of equation

```
fit1$beta
```

```
## 64 x 1 sparse Matrix of class "dgCMatrix"
##                   s0
## age         .
## sex      -110.066452
## bmi       501.757722
## map       250.568913
## tc          .
## ldl         .
## hdl      -186.344533
## tch         .
## ltg       467.492705
## glu        17.264497
## age^2       6.435989
## bmi^2      38.121823
## map^2       .
## tc^2        .
## ldl^2       .
## hdl^2       .
## tch^2       .
## ltg^2       .
## glu^2      68.875707
```

```
## age:sex  106.521069
## age:bmi    .
## age:map   29.925456
## age:tc     .
## age:ldl    .
## age:hdl    .
## age:tch    .
## age:ltg    8.159214
## age:glu   11.957597
## sex:bmi    .
## sex:map    .
## sex:tc     .
## sex:ldl    .
## sex:hdl    .
## sex:tch    .
## sex:ltg    .
## sex:glu    .
## bmi:map   85.023660
## bmi:tc     .
## bmi:ldl    .
## bmi:hdl    .
## bmi:tch    .
## bmi:ltg    .
## bmi:glu    .
## map:tc     .
## map:ldl    .
## map:hdl    .
## map:tch    .
## map:ltg    .
## map:glu    .
## tc:ldl     .
## tc:hdl     .
## tc:tch     .
## tc:ltg     .
## tc:glu     .
## ldl:hdl    .
## ldl:tch    .
## ldl:ltg    .
## ldl:glu    .
## hdl:tch    .
## hdl:ltg    .
## hdl:glu    .
## tch:ltg    .
## tch:glu    .
## ltg:glu    .
```

We can observe that with this lambda value few beta values has been changed and few are compresed to zero

```
cv_fit1$lambda.1se
```

```
## [1] 5.941189
```

```
fit <- glmnet(diabetes$x, diabetes$y, alpha = 1, lambda=cv_fit1$lambda.1se)
fit$beta
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##              s0
## age    .
## sex   -9.457099
## bmi  506.422970
## map  196.706950
## tc     .
## ldl    .
## hdl -121.136065
## tch    .
## ltg  441.023248
## glu    .
```

We can observe that with this lambda value few beta values has been changed and few are compresed to zero.But we can observe the difference
Now only few most significant values are left.this will reduce complexity but meanwhile increases mean sqaure error if we use this model on train data and predict on test data

Biglasso

```
library(biglasso)
```

```
## Loading required package: bigmemory
```

```
## Loading required package: ncvreg
```

```
data(colon)

X <- colon$X
Y <- colon$y

dim(X)
```

```
## [1]   62 2000
```

```
X[1:5, 1:5]
```

```
##   Hsa.3004 Hsa.13491 Hsa.13491.1 Hsa.37254 Hsa.541
## t  8589.42   5468.24     4263.41   4064.94 1997.89
## n  9164.25   6719.53     4883.45   3718.16 2015.22
## t  3825.71   6970.36     5369.97   4705.65 1166.55
## n  6246.45   7823.53     5955.84   3975.56 2002.61
## t  3230.33   3694.45     3400.74   3463.59 2181.42
```

```
X.bm <- as.big.matrix(X)

str(X.bm)
```

```
## Formal class 'big.matrix' [package "bigmemory"] with 1 slot
##   ..@ address:<externalptr>
```

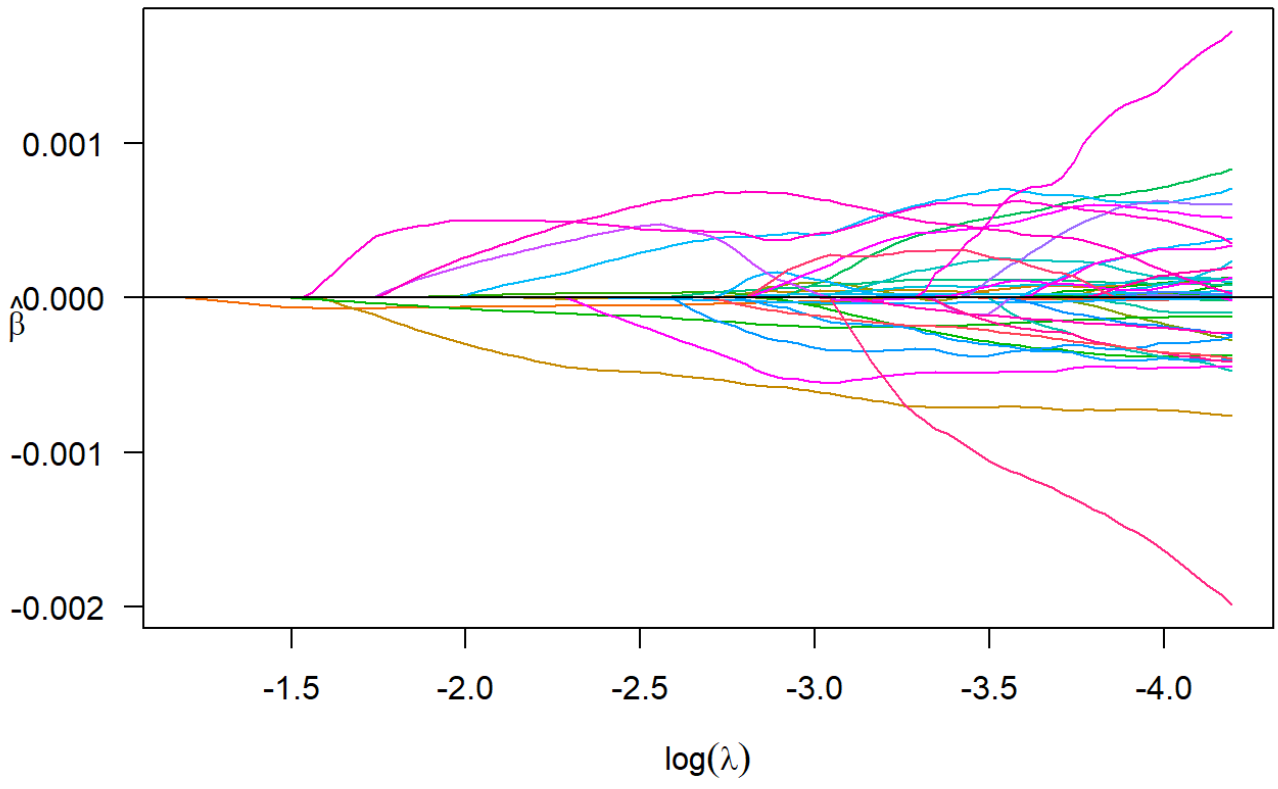
```
dim(X.bm)
```

```
## [1]   62 2000
```

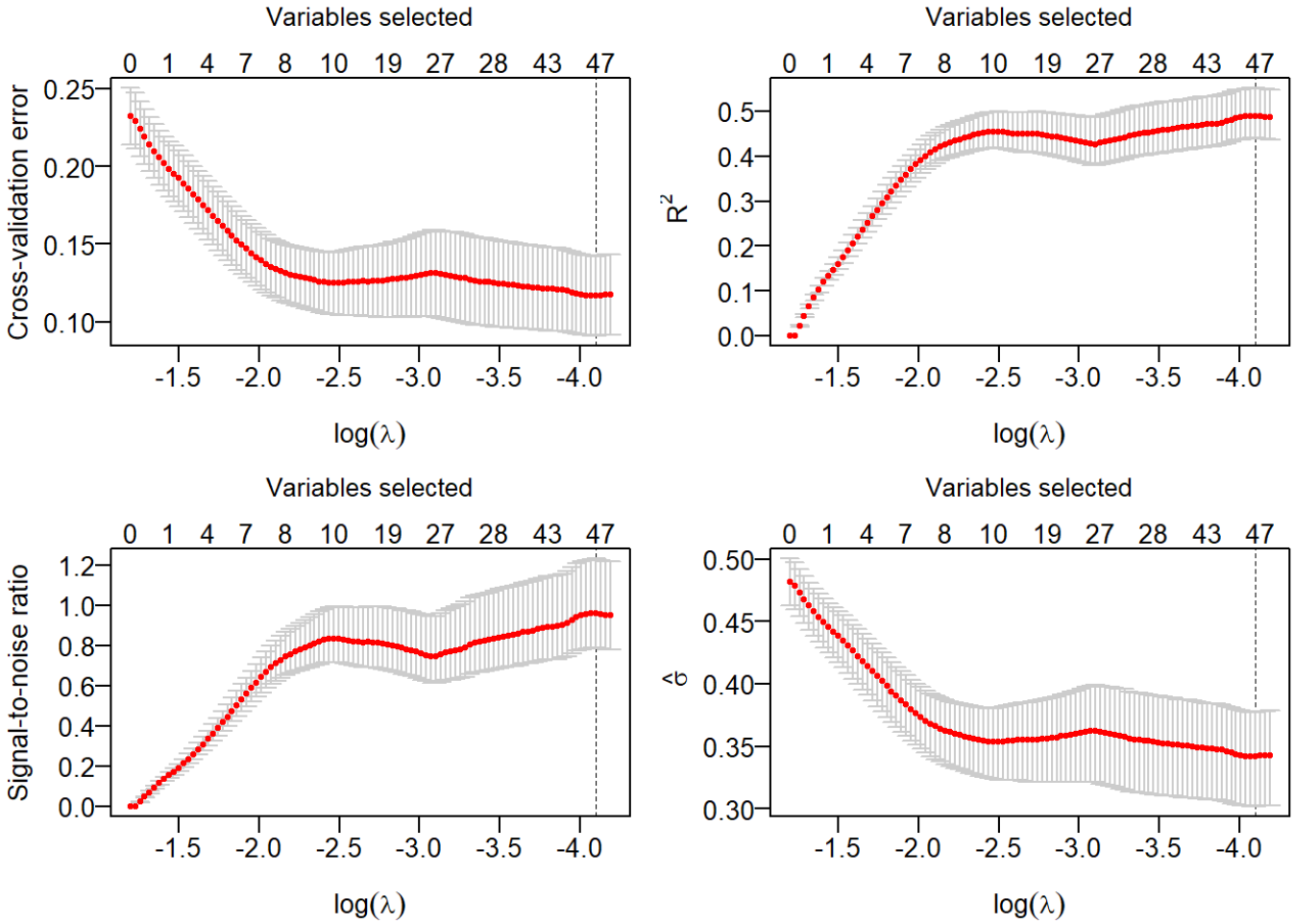
```
X.bm[1:5, 1:5]
```

```
##   Hsa.3004 Hsa.13491 Hsa.13491.1 Hsa.37254 Hsa.541
## t  8589.42   5468.24     4263.41   4064.94 1997.89
## n  9164.25   6719.53     4883.45   3718.16 2015.22
## t  3825.71   6970.36     5369.97   4705.65 1166.55
## n  6246.45   7823.53     5955.84   3975.56 2002.61
## t  3230.33   3694.45     3400.74   3463.59 2181.42
```

```
fit <- biglasso(X.bm, Y, screen = "SSR-BEDPP")

plot(fit)
```



```
cvfit <- cv.biglasso(X.bm , Y, seed = 1234, nfolds = 10, ncores = 4)
par(mfrow  = c(2,2), mar = c(3.5, 3.5, 3, 1) ,mgp = c(2.5, 0.5, 0))

plot(cvfit, type = "all")
```




```
summary(cvfit)
```

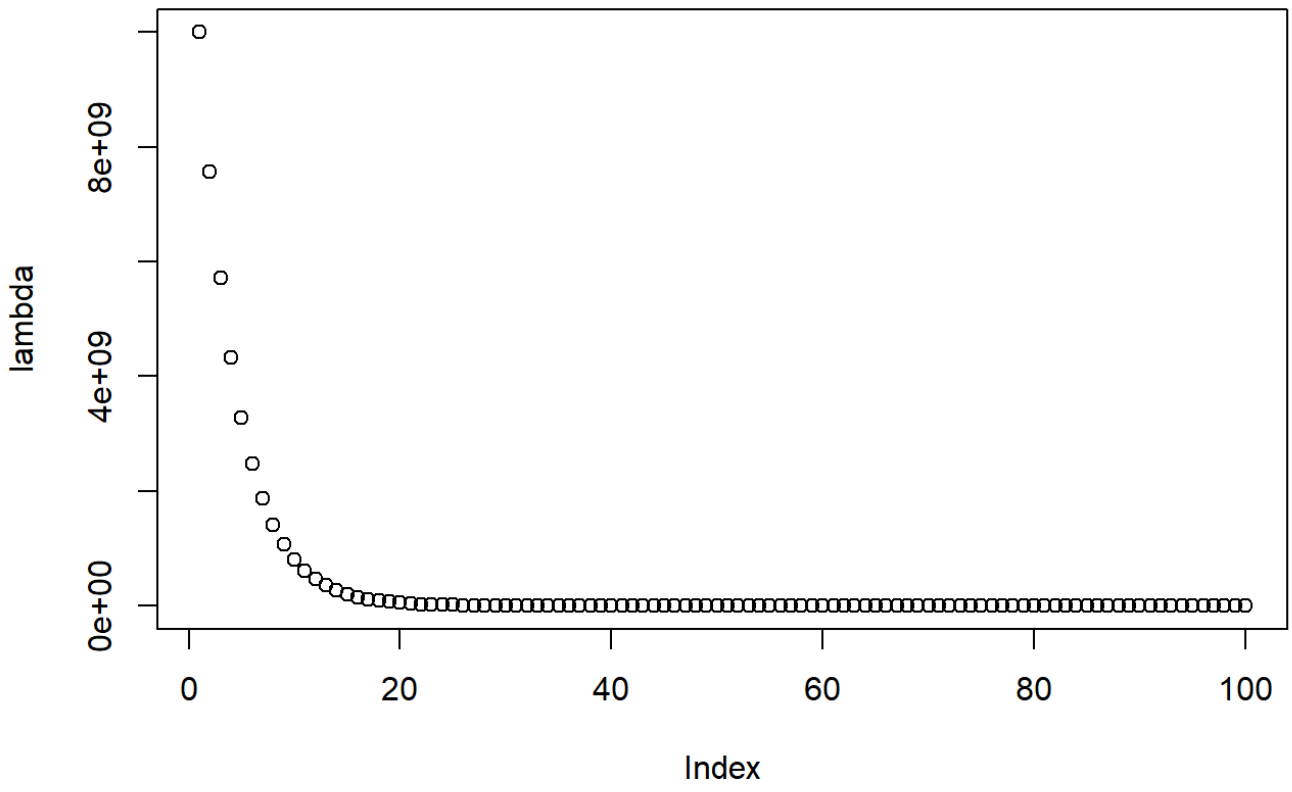
```
## lasso-penalized linear regression with n=62, p=2000
## At minimum cross-validation error (lambda=0.0165):
## -------------------------------------------------
##   Nonzero coefficients: 46
##   Cross-validation error (deviance): 0.12
##   R-squared: 0.49
##   Signal-to-noise ratio: 0.96
##   Scale estimate (sigma): 0.342
```

```
coef(cvfit)[which(coef(cvfit) != 0)]
```

```
##  [1]  7.000690e-01 -1.068388e-05 -8.774821e-06  1.469851e-05 -1.279683e-05
##  [6] -1.237284e-05  1.143071e-04 -7.435492e-04  1.141762e-04 -2.244924e-04
## [11]  1.905808e-05  5.447871e-05  1.168477e-04 -1.243593e-04  1.583372e-05
## [16] -3.744562e-04  7.753436e-04 -1.255786e-05  8.553825e-05 -9.841401e-05
## [21] -4.167749e-04  1.247990e-04  6.261196e-06  1.151005e-04  1.458936e-04
## [26]  3.773071e-05  6.565699e-04  3.536769e-04 -7.666380e-06 -4.027543e-04
## [31] -2.844216e-04 -2.123102e-04  2.624890e-05  6.996317e-06  6.055567e-04
## [36]  1.041546e-04  5.319337e-04 -4.489856e-04  3.107549e-04  1.578863e-03
## [41]  4.404346e-04  9.230497e-05 -2.166536e-04  1.749590e-04 -3.892690e-04
## [46] -1.822198e-03 -3.690431e-04
```

Custom example

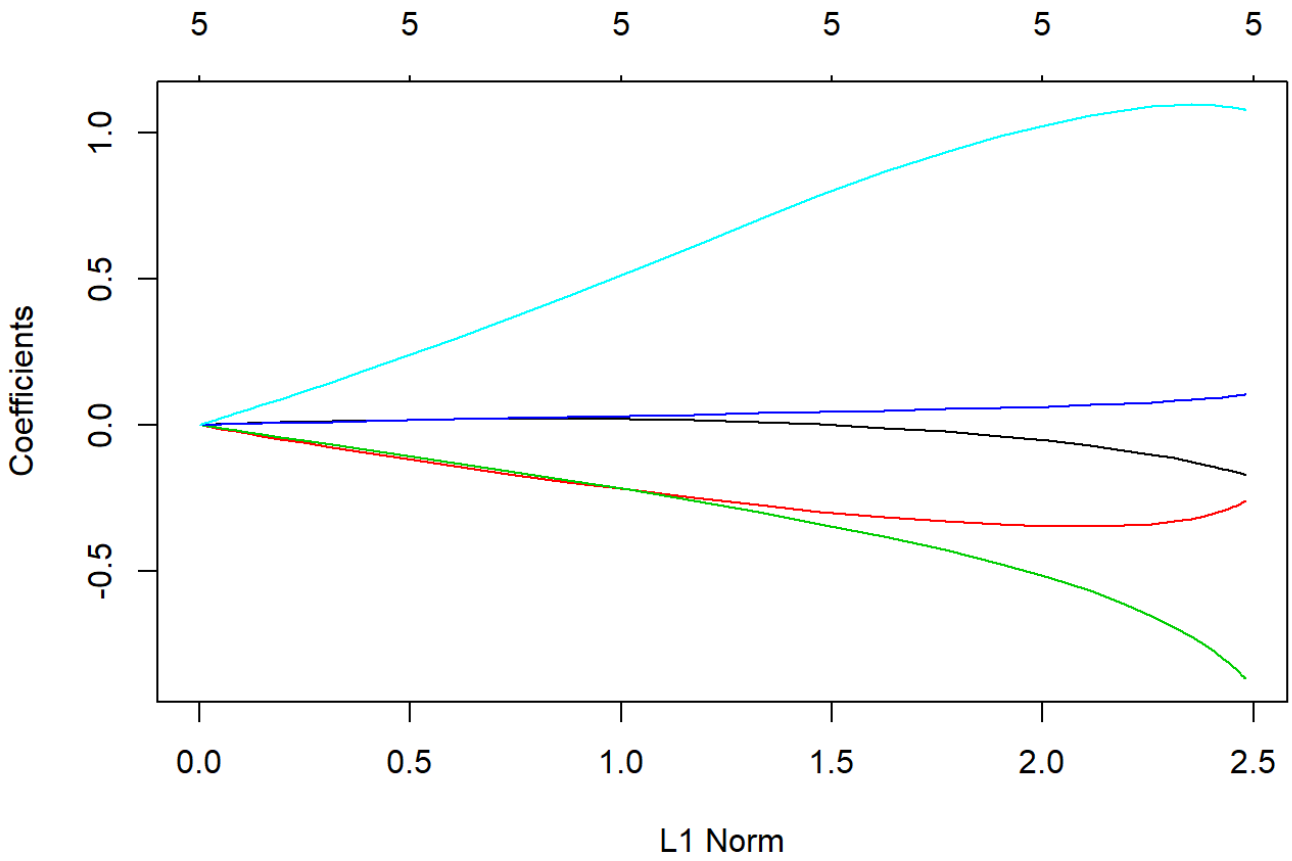
```
swiss <- datasets::swiss


x <- model.matrix(Fertility ~ . , swiss)[,-1]
y <- swiss$Fertility

lambda <- 10^seq(10, -2, length = 100)
par(mfrow = c(1,1))
plot(lambda)
```



```
set.seed(489)

train = sample(1:nrow(x),nrow(x)/2)
test = (-train)
ytest = y[test]

swisslm <- lm(Fertility~., data = swiss)


a <- coef(swisslm)
a
```

```
##      (Intercept)      Agriculture      Examination        Education
##       66.9151817       -0.1721140       -0.2580082       -0.8709401
##         Catholic Infant.Mortality
##        0.1041153        1.0770481
```

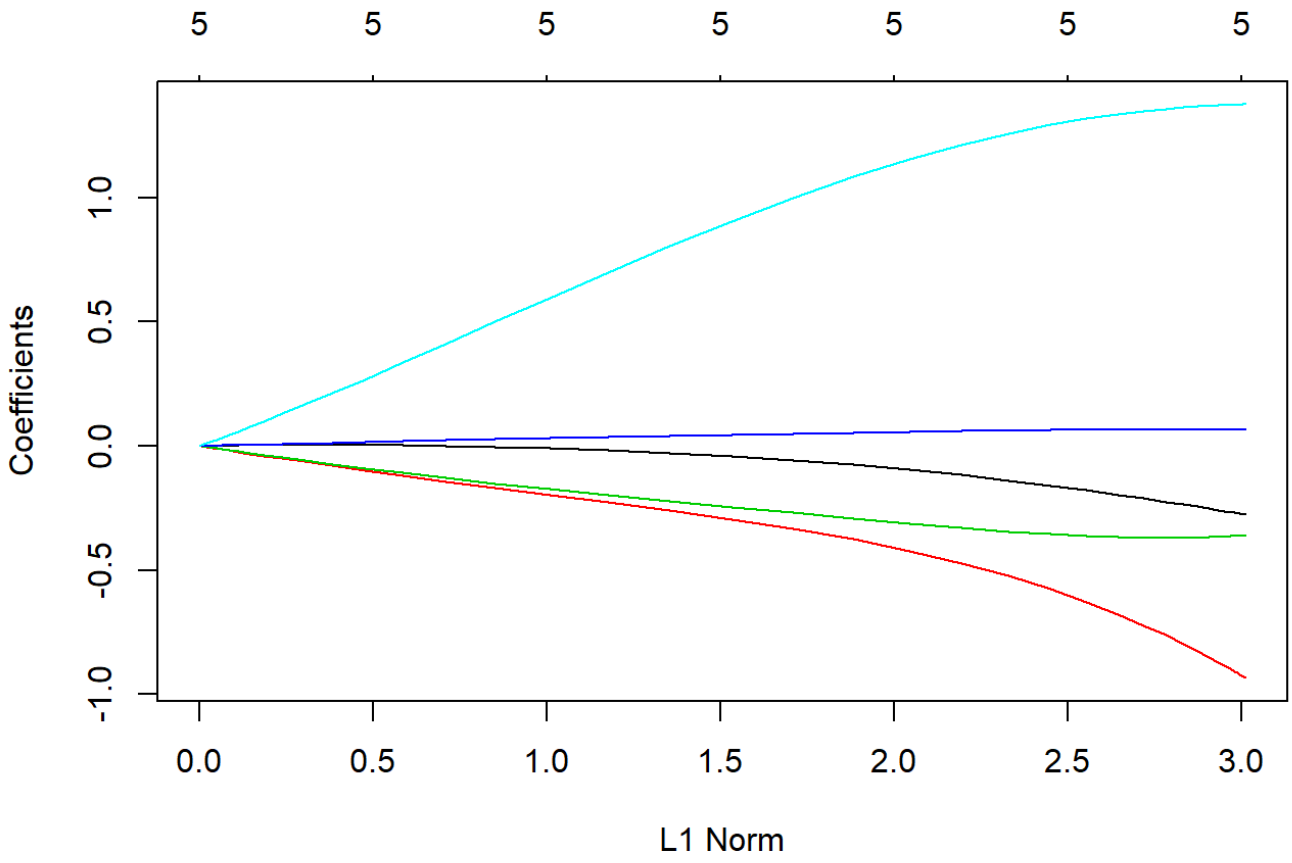
```
ridge.mod <- glmnet(x, y, alpha = 0, lambda = lambda)
plot(ridge.mod)
```




```
predict(ridge.mod, s = 0, type = 'coefficients')[1:6,]
```

```
##      (Intercept)      Agriculture      Examination        Education
##       66.8911177       -0.1714307       -0.2603091       -0.8681376
##         Catholic Infant.Mortality
##        0.1037196        1.0776950
```

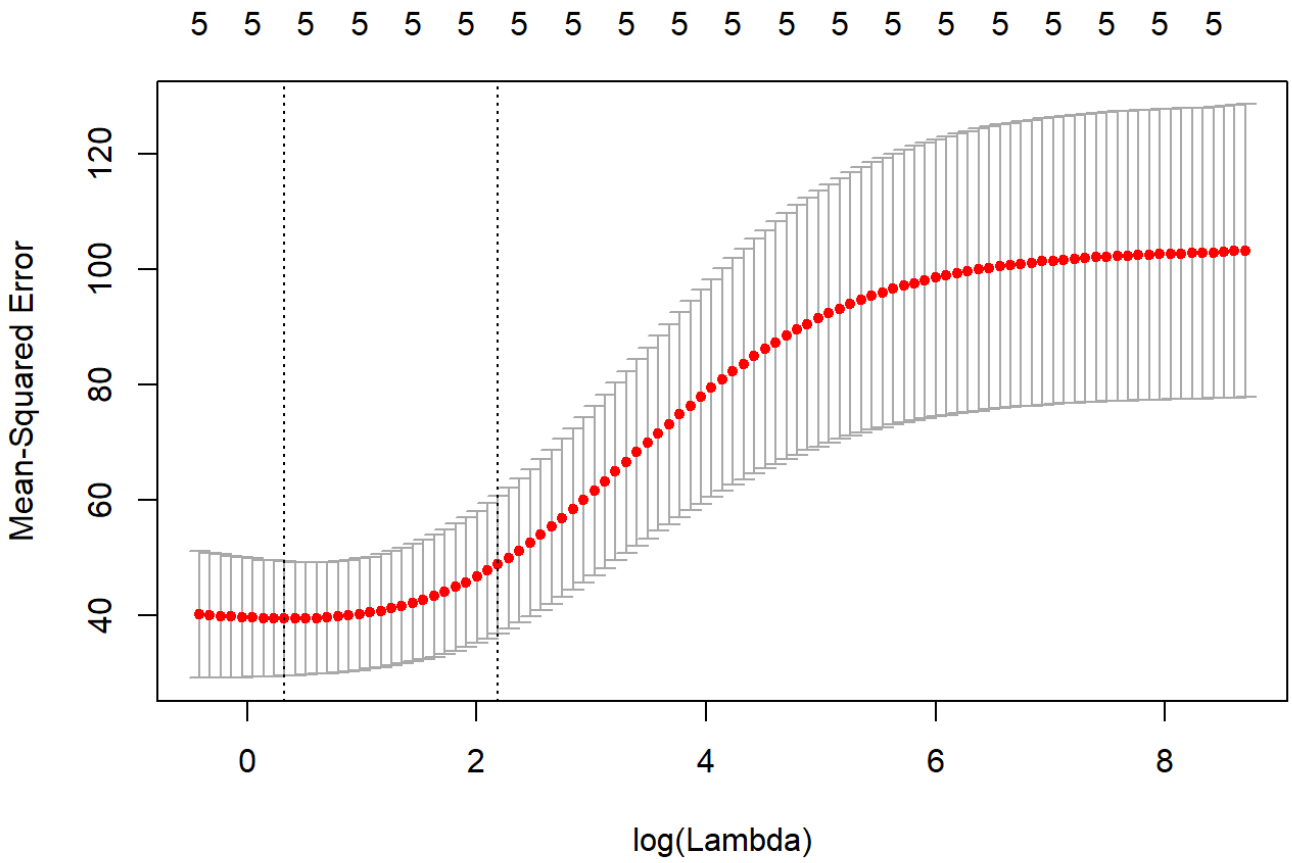
```
swisslm <- lm(Fertility~., data = swiss, subset = train)

ridge.mod <- glmnet(x[train,], y[train], alpha = 0, lambda = lambda)
plot(ridge.mod)
```




```
cv.out <- cv.glmnet(x[train,], y[train], alpha = 0)
```

```
## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations
## per fold
```

```
plot(cv.out)
```




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

```
## [1] 1.386317
```

```
ridge.pred <- predict(ridge.mod, s = bestlam, newx = x[test,])

s.pred <- predict(swisslm, newdata = swiss[test,])

mean((s.pred-ytest)^2)
```

```
## [1] 106.0087
```

```
mean((ridge.pred - ytest)^2)
```

```
## [1] 93.02157
```

```
out = glmnet(x[train,],y[train], alpha = 0)
predict(ridge.mod, type = "coefficients", s = bestlam)[1:6,]
```

```
##      (Intercept)      Agriculture      Examination        Education
##      64.90631178      -0.16557837      -0.59425090      -0.35814759
##         Catholic Infant.Mortality
##       0.06545382       1.30375306
```

```
lasso.mod <- glmnet(x[train,], y[train], alpha = 1, lambda = lambda)
lasso.pred <- predict(lasso.mod, s = bestlam, newx = x[test,])
mean((lasso.pred-ytest)^2)
```

```
## [1] 124.1039
```

```
lasso.coef  <- predict(lasso.mod, type = 'coefficients', s = bestlam)[1:6,]
lasso.coef
```

```
##      (Intercept)      Agriculture      Examination        Education
##      54.72576032      -0.01493362      -0.40726342      -0.05839363
##         Catholic Infant.Mortality
##       0.03829186       1.19563533
```