PSTAT 126

Regression Analysis

Lecture 11

Shrinkage Methods

High Dimensional Problems - Large p

Consider the MLR model with n observations and p predictors:

$$y_i = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi} + \epsilon_i \qquad i = 1, \ldots, n$$

1. If $p > n$ the LS estimate of $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^T$ is not unique (there are more unknowns than constraints) $\Rightarrow$ imprecise estimates and inaccurate prediction.
2. If $p < n$ we may face collinearity issues, as some predictors can be expressed as linear combinations of others $\Rightarrow \boldsymbol{X}^T\boldsymbol{X}$ non-invertible.
3. Even if there are no collinearity problems when $p < n$, prediction performance may decrease by using too many predictors.

For all cases, we may need to remove or "shrink" irrelevant predictors so that our model only incorporates useful information.

Collinearity Issues

- *Exact Collinearity*: $\boldsymbol{X}^T\boldsymbol{X}$ is singular $\rightarrow$ LS estimate of $\boldsymbol{\beta}$ is not unique.
- *Multicollinearity*: $\boldsymbol{X}^T\boldsymbol{X}$ close to singular due to strong correlation among predictors $\rightarrow$ imprecise estimates of $\boldsymbol{\beta}$.

Collinearity can be detected in several ways:

1. Examination of correlation matrix: High correlation (either close to -1 or $+1$) may indicate large pairwise collinearity.
2. Running a regression of x_i on all other predictors: If R_i^2 is close to 1, it means there is a linear dependency among predictors.

Car drivers Example

Researchers at the HuMoSim laboratory at the University of Michigan collected data on 38 drivers to investigate where different drivers will position the seat depending on their size and age. y: *hipcenter* (Horizontal distance from the midpoint of the hips to a fixed location in the car).

```
data(seatpos)
lmod<- lm( hipcenter~., seatpos)
summary(lmod)$coefficients
```

```
##                 Estimate  Std. Error     t value   Pr(>|t|)
## (Intercept) 436.43212823 166.5716187  2.62008697 0.01384361
## Age           0.77571620   0.5703288  1.36012113 0.18427175
## Weight        0.02631308   0.3309704  0.07950283 0.93717877
## HtShoes      -2.69240774   9.7530351 -0.27605845 0.78446097
## Ht            0.60134458  10.1298739  0.05936348 0.95306980
## Seated        0.53375170   3.7618942  0.14188376 0.88815293
## Arm          -1.32806864   3.9001969 -0.34051323 0.73592450
## Thigh        -1.14311888   2.6600237 -0.42974011 0.67056106
## Leg          -6.43904627   4.7138601 -1.36598163 0.18244531
```

```
summary(lmod)$r.squared ## The R2 is moderately large, why?
```

```
## [1] 0.6865535
```

Car drivers Example

We check pairwise correlations:

```
round(cor(seatpos[,-9]),2)
```

```
##              Age Weight HtShoes    Ht Seated  Arm Thigh   Leg
## Age         1.00   0.08   -0.08 -0.09  -0.17 0.36  0.09 -0.04
## Weight      0.08   1.00    0.83  0.83   0.78 0.70  0.57  0.78
## HtShoes    -0.08   0.83    1.00  1.00   0.93 0.75  0.72  0.91
## Ht         -0.09   0.83    1.00  1.00   0.93 0.75  0.73  0.91
## Seated     -0.17   0.78    0.93  0.93   1.00 0.63  0.61  0.81
## Arm         0.36   0.70    0.75  0.75   0.63 1.00  0.67  0.75
## Thigh       0.09   0.57    0.72  0.73   0.61 0.67  1.00  0.65
## Leg        -0.04   0.78    0.91  0.91   0.81 0.75  0.65  1.00
```

Regularization or Shrinkage Methods

We present alternative estimation methods that allow us to *shrink* extra information, obtained from a large number of predictors p, into a more useful form. Generally speaking we fit a model with p predictors by constraining or regularizing the size of the coefficient estimates $\hat{\beta}_j$. As a result, we will obtain estimates with smaller variance (more precision).

- Ridge Regression
- Lasso Regression

Ridge Regression

- **Assumption**: Regression coefficients (after normalization) should not be very large, so that we should bound/restrict the size of the coefficients (shrinkage).
- **Scenario**: Applications where we have large p and we believe many of them should have some effect on the response. Effective in the presence of collinearity.

Ridge Regression

After normalizing the predictors, and centering the response; the Ridge Regression estimate chooses $\boldsymbol{\beta}$ that minimizes:

$$SSR + \lambda \sum_{j=1}^{p} \beta_j^2 = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_2^2$$

For some choice of the shrinkage penalty $\lambda \geq 0$. Ridge regression is an example of *penalized regression* with penalty term $\sum_{j=1}^{p} \beta_j^2$, which is restricted to be small.

Ridge Regression

Equivalently the optimization problem boils down to finding $\boldsymbol{\beta}$ that minimizes:

$$(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \quad subject\ to \quad \sum_{j=1}^{p} \beta_j^2 \leq t^2$$

We find the LS solution subject to an upper bound on the size of the coefficients.

The use of ridge regression can also be justified from a Bayesian perspective where a prior distribution on the coefficients puts more weight on smaller values.

Ridge Regression

The solution for the estimates of β can be written as:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

Note:

- $(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})$ is always invertible even if $\boldsymbol{X}^T\boldsymbol{X}$ is not.
- Ridge regression estimate is biased (Less accurate than LS estimate).
- Ridge regression estimate could have smaller variance (more precise than LS estimate).

Ridge Regression

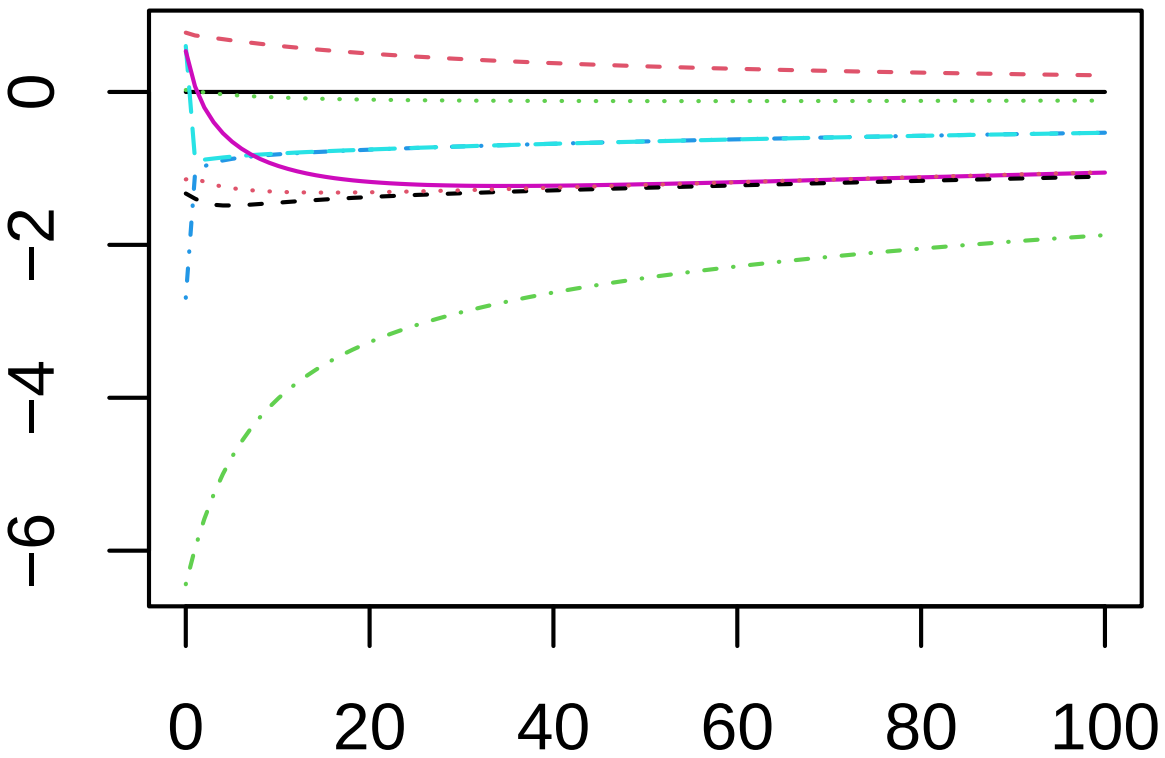
The solution for the estimates of β can be written as:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

Shrinkage penalty $\boldsymbol{\lambda}$: Notice that if $\lambda = 0$ this solution will produce the LS estimate. On the contrary if $\lambda \to \infty$ the ridge regression will produce estimates all approaching zero.
The ridge regression estimates highly depend on $\boldsymbol{\lambda}$!!

Car drivers Example

```
require(MASS)
par(mar = c(2, 2, 0.5, 0.5))
seatpos<- scale((seatpos), center = TRUE, scale = FALSE)
seatpos<- as.data.frame(seatpos)
rgmod<- lm.ridge(hipcenter~., seatpos, lambda = seq(0,100, len=100))
matplot(rgmod$lambda, coef(rgmod), type="l", xlab = "lambda", ylab = "Beta hat", cex=0.8)
```



Car drivers Example

```
require(MASS)
a<-which.min(rgmod$GCV);a ##Generalized crossvalidation
```

```
##  22.222222
##         23
```

```
coef(rgmod)[a,]
```

```
##                         Age         Weight       HtShoes             Ht
## -4.617067e-16  4.839312e-01 -1.043315e-01 -7.445594e-01 -7.428084e-01
##        Seated            Arm          Thigh            Leg
## -1.194625e+00 -1.361721e+00 -1.307340e+00 -3.167155e+00
```

Lasso Regression

Consider the MLR model with n observations and p predictors:

$$y_i = \beta_0 + \beta_1 x_{1i} + \ldots + \beta_p x_{pi} + \epsilon_i \qquad i = 1, \ldots, n$$

For the Lasso Regression we choose $\boldsymbol{\beta}$ that minimizes:

$$SSR + \lambda \sum_{i=1}^{p} |\beta_j| = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_1$$

Which is equivalent to finding $\boldsymbol{\beta}$ that minimizes:

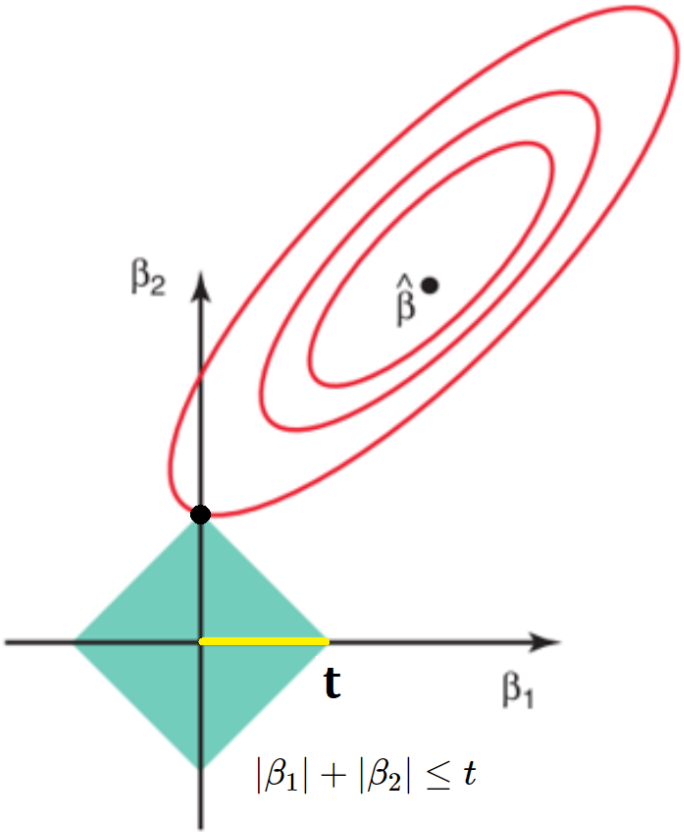
$$(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) \quad subject\ to \quad \sum_{i=1}^{p} |\beta_j| \leq t$$

Lasso Regression vs Ridge Regression

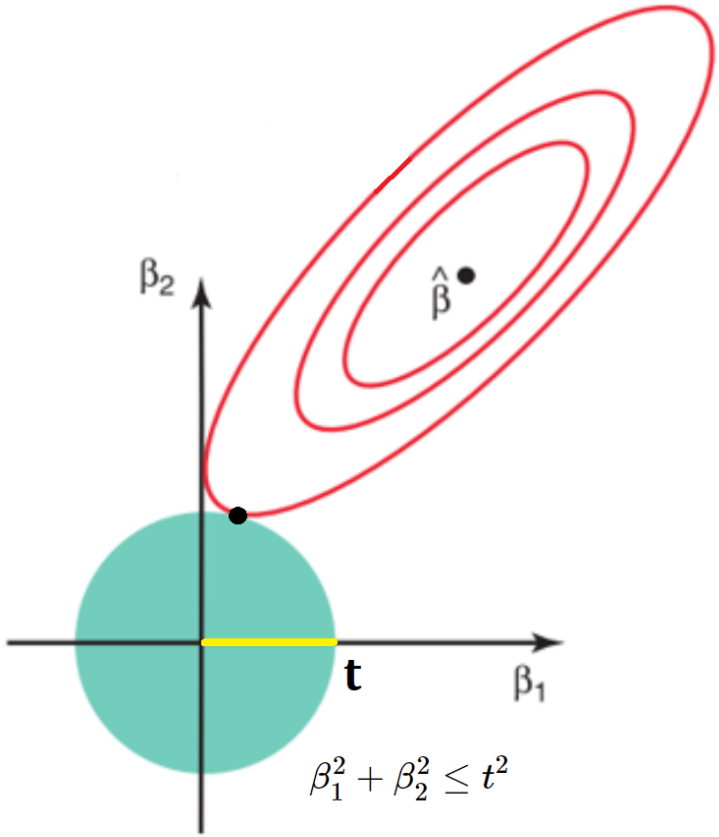
The important difference between lasso and ridge regression in in the nature of the solutions:

- For Lasso: When $p = 2$, the L_1 constraint $\sum_{j=1}^{p} |\beta_j| \leq t$ defines a square in two dimensions with vertices on the coordinates axes.
- For Lasso, some of the β's are shrunk to exactly zero. Thus, its usage is most appropriate when we believe the effects are *sparse*: That the response can be explained by a small number of predictors.
- Lasso can be regarded as a type of variable selection method. When $\hat{\beta}_j = 0$ it means we can remove the corresponding predictor x_j. In contrast Ridge does not eliminate any variables; it only shrinks the $\hat{\beta}_j$ to smaller values.

Lasso Regression vs Ridge Regression



Lasso Regression



Ridge Regression

Tuning parameter λ

λ controls the strength of the L_1 penalty. λ can be defined as the amount of shrinkage:

- When $\lambda = 0$, no parameters are eliminated. The estimate is equal to the one found with LS.
- As λ increases, more and more coefficients are set to zero and eliminated (theoretically, when $\lambda = \infty$, all coefficients are eliminated).
- As λ increases, bias increases but variance decreases. (More precision but less accuracy).
- As λ decreases, variance increases but bias decreases. (More accuracy but less precision).

Lasso Regression: Standardization

It is best to apply ridge regression / lasso after standardizing the predictors:

$$\tilde{x}_j = \frac{x_{ij} - \bar{x}_j}{SD_{x_j}}$$

SD_{x_j} corresponds to the standard deviation of the jth predictor and $\bar{x}_j$ is the average of the jth predictor.

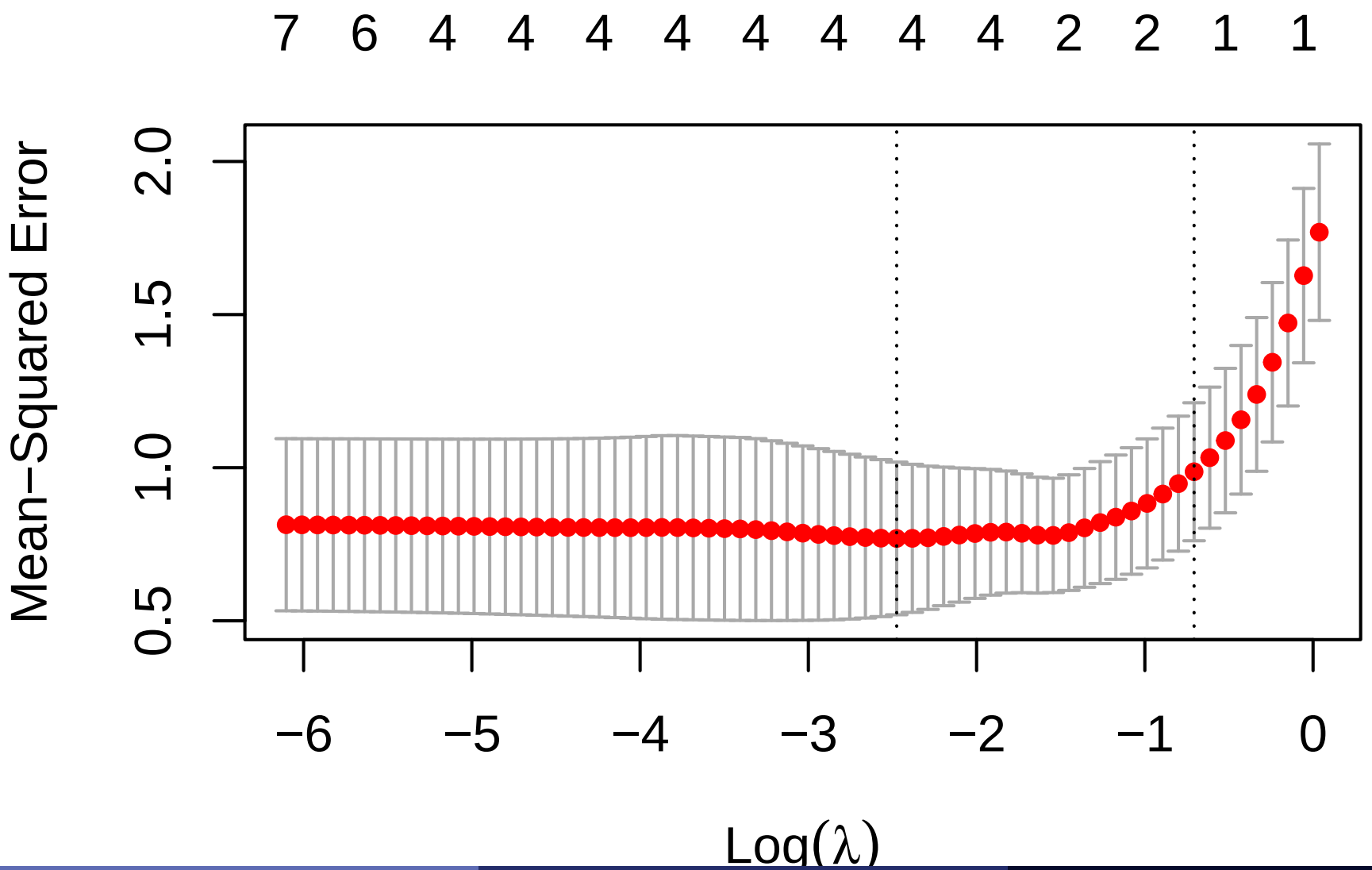
States Example

```
require(glmnet)
data(state)
statedata<- data.frame(state.x77, row.names =state.abb)
y <- statedata$Life
x <- scale(data.matrix(statedata[,-4]))
cv_model <- cv.glmnet(x, y, alpha = 1)
#find optimal lambda value that minimizes test MSE
best_lambda <- cv_model$lambda.min;best_lambda
```

```
## [1] 0.0841684
```

States Example

```
par(mar = c(7, 4, 2.2, 0.5));plot(cv_model, cex=0.8)
```



States Example

```
#find coefficients of best model
best_model <- glmnet(x, y, alpha = 1, lambda = best_lambda)
coef(best_model)
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) 70.8786000
## Population   0.1291629
## Income        .
## Illiteracy    .
## Murder       -0.9311791
## HS.Grad       0.3038190
## Frost        -0.1335086
## Area          .
```

```
new = matrix(c(0.12,0.3, 2, -0.7, 0.74,1.5, 2.7),
             nrow=1, ncol=7) ##Scaled new observation
predict(best_model, s = best_lambda, newx = new)
```

```
##            s1
## [1,] 71.57049
```

Spectroscopy example

A Tecator Infratec Food and Feed Analyzer working in the wavelength range 850- 1050 nm by the Near Infrared Transmission (NIT) principle was used to collect data on samples of finely chopped pure meat. For each sample, the fat content was measured along with a 100 channel spectrum of absorbances. Determination of fat content via analytical chemistry is time consuming we would like to build a model to predict the fat content of new samples using the 100 absorbances which can be measured more easily. $n = 215$, $p = 100$ (different absorbance spectrum), y:Fat content.

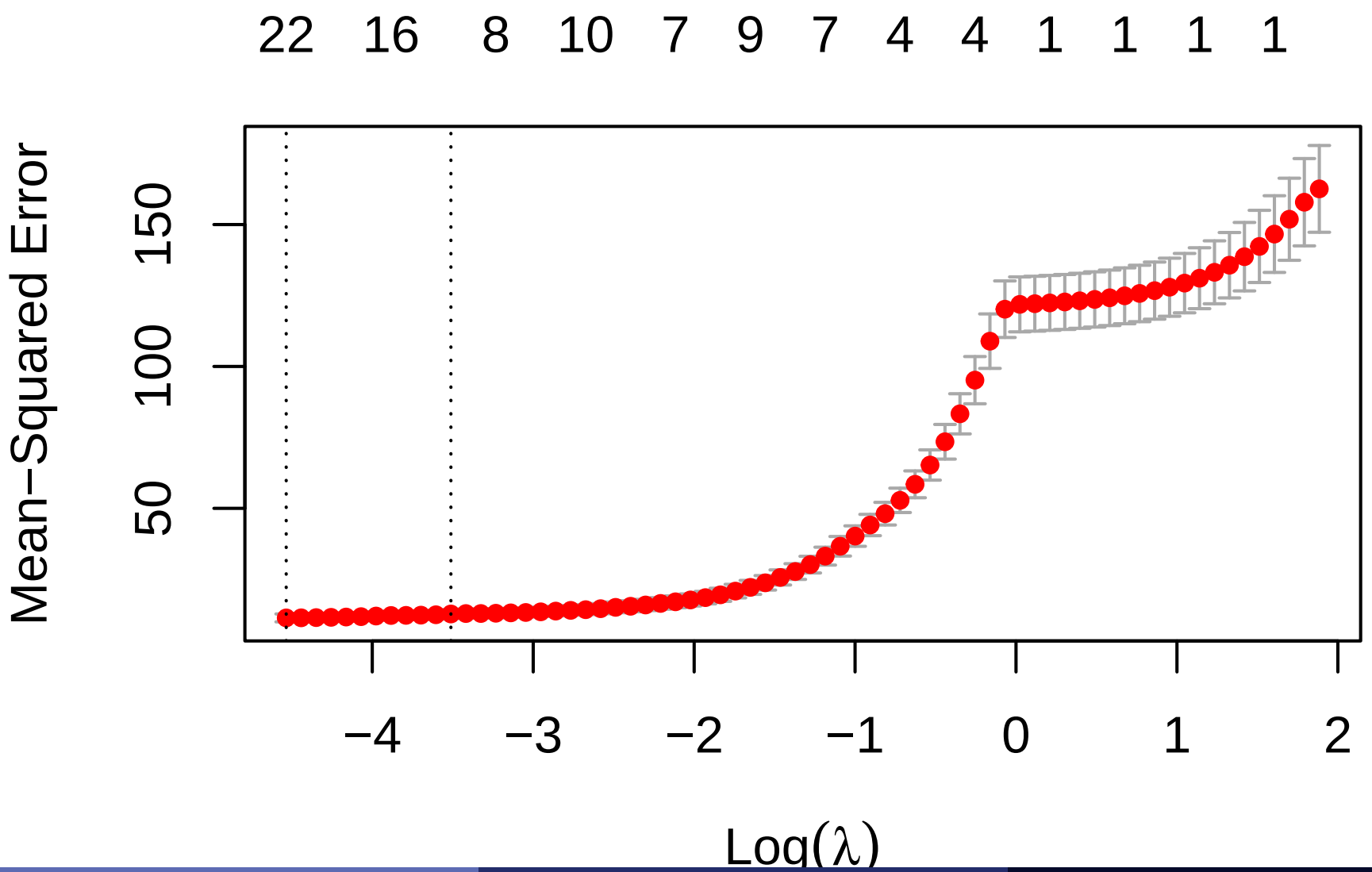
Spectroscopy example

```
require(faraway)
data(meatspec)
y <- meatspec$fat
x <- scale(data.matrix(meatspec[,-101]))
cv_model <- cv.glmnet(x, y, alpha = 1)
#find optimal lambda value that minimizes test MSE
best_lambda <- cv_model$lambda.min;best_lambda
```

```
## [1] 0.01072971
```

Spectroscopy Example

```
par(mar = c(7, 4, 2.2, 0.5));plot(cv_model, cex=0.8)
```



Spectroscopy Example

```r
#find coefficients of best model
best_model <- glmnet(x, y, alpha = 1, lambda = best_lambda)
par(mar = c(7, 4, 0.5, 0.5));plot(coef(best_model), type="h", xlab="index", ylab="Coefficient", col="blue", cex
```