Ridge Regression and LASSO

UBCO MDS — DATA 571

Motivation

- We've now seen a couple of alternatives to the linear model for regression.
- BUT, the linear model still reigns supreme in most realms of science and industry due to its simplicity (which helps for inference).
- Let's see how we can improve upon the linear model.

Variable Selection

- You have seen (570) forward, backward, and mixed stepwise selection for getting rid of useless predictors.
- Those methods were largely based on the *p*-values resulting from the *t*-tests on the coefficients...certainly not ideal, especially for large predictor sets.
- The model measures discussed in 570 (adjusted R^2, AIC, BIC, etc) can help choose between models, but not suggest which models to compare.
- Let's start looking at some more modern methods to intertwine these goals...

Ridge Regression

- Recall: Least squares estimation minimizes the RSS

$$RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2$$

- Ridge regression includes a penalty in the estimation process, instead minimizing

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}\beta_j^2 = RSS + \lambda\sum_{j=1}^{p}\beta_j^2$$

where λ is a tuning parameter.

- This penalty shrinks the β_j estimates towards 0.

λ

- If $\lambda = 0$ then the estimation process is simply least squares once again.
- As $\lambda \to \infty$ then the penalty grows and the coefficients approach (but never equal) 0.
- Clearly, some care is required in determining λ. Before diving into that further, let's introduce another (more popular) method of penalizing least squares...

Lasso

- The 'least absolute shrinkage and selection operator' or lasso is arguably the most popular modern method applied to linear models. It is quite similar to ridge regression...

- The lasso minimizes

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}|\beta_j| = RSS + \lambda\sum_{j=1}^{p}|\beta_j|$$

  where λ is a tuning parameter.

- It turns out that this penalty has an important advantage: it will force some coefficients to 0, whereas RR will only force them 'close' to 0.

λ again

- Both the lasso and ridge regression require specification of λ
- In fact, for each value of λ we will see different coefficients. So how do we find the best one?
- The answer, as usual, is cross-validation.

Example: Proof of Concept

- This approach (especially lasso) is popular with large data sets to simultaneously model in a linear fashion and perform variable selection.
- BUT, even if variable selection is not your goal, shrinkage methods are still useful!
- Let's see why...

Example: Proof of Concept

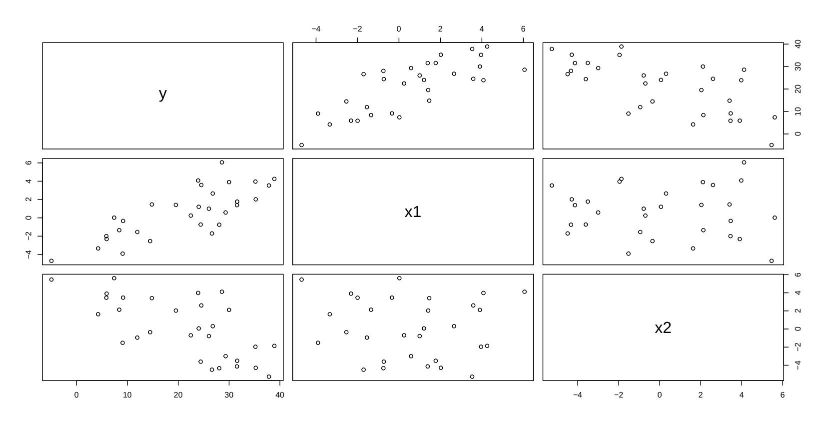
- We generate $n = 30$ data under the following model

$$Y = 20 + 3X_1 - 2X_2 + \epsilon$$

- Both X_1 and X_2 are sampled independently as normal. ϵ is sampled as N(0, 4).

- ...

Example: Proof of Concept



Example: Proof of Concept

```
> summary(lm(y~x1+x2))

Call:
lm(formula = y ~ x1 + x2)

Residuals:
    Min      1Q  Median      3Q     Max
-4.7524 -0.9086 -0.1260  1.7005  3.0130

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  19.4425     0.3636   53.47   <2e-16 ***
x1            3.0808     0.1335   23.07   <2e-16 ***
x2           -2.1267     0.1097  -19.39   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 1.941 on 27 degrees of freedom
Multiple R-squared:  0.9734,Adjusted R-squared:  0.9714
F-statistic: 494.4 on 2 and 27 DF,  p-value: < 2.2e-16
```

Example: Proof of Concept

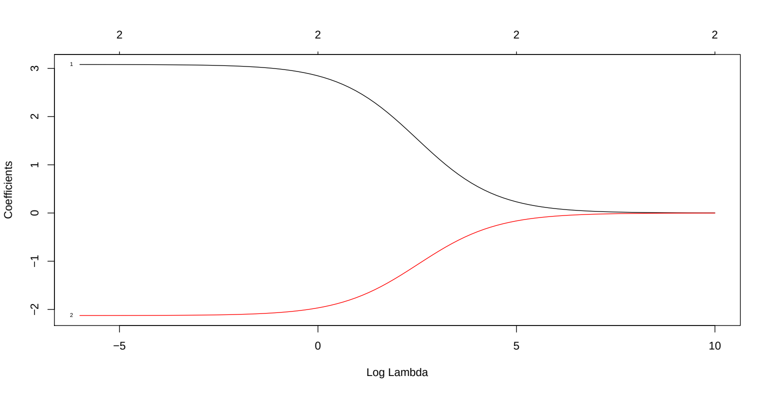
- So, the standard approach gives us the estimated model

$$\hat{Y} = 19.44 + 3.08X_1 - 2.12X_2$$

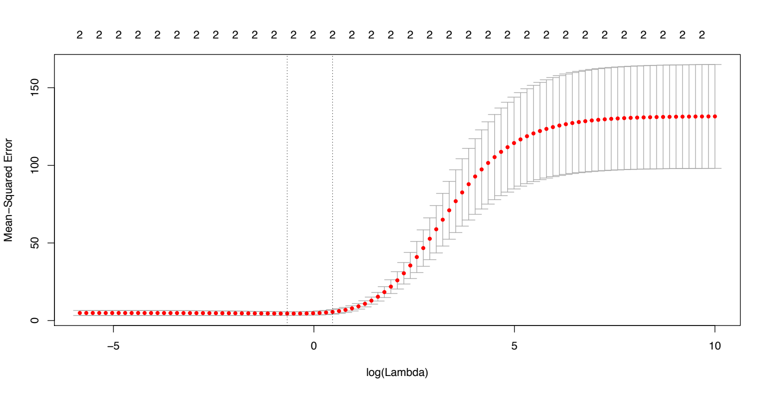
- As compared to the true model, we are underestimating the intercept, and overestimating (in absolute value sense) the coefficients

- So, let's look at ridge regression. Firstly, we need to determine the tuning parameter. Some built-in functions give us the following plots

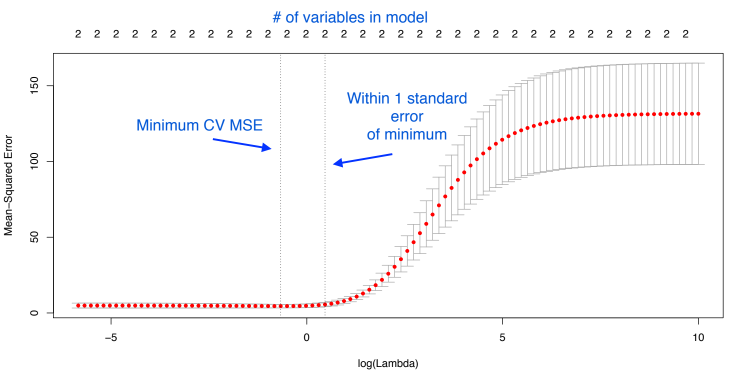
Example: Proof of Concept



Example: Proof of Concept



Example: Proof of Concept



Example: Proof of Concept

```
> coef(rrsimmin)
3 x 1 sparse Matrix of class "dgCMatrix"
                   s0
(Intercept) 19.460996
x1           3.045163
x2          -2.102821
> coef(rrsim1se)
3 x 1 sparse Matrix of class "dgCMatrix"
                   s0
(Intercept) 19.581891
x1           2.811993
x2          -1.946284
```

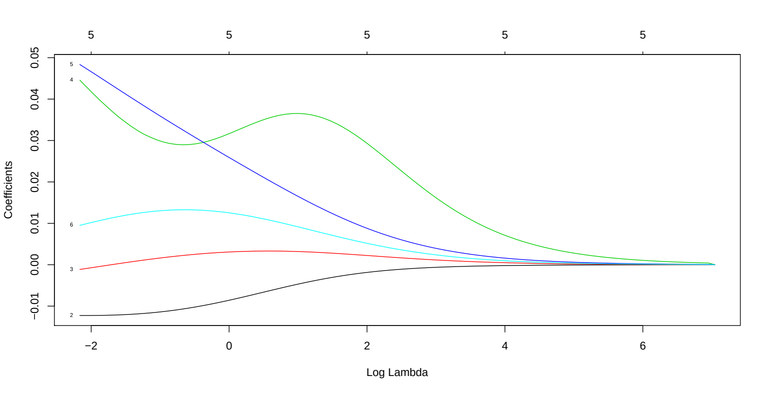
Example: Proof of Concept

- $$\lambda = 0.000 \qquad \hat{Y} = 19.44 + 3.08X_1 - 2.12X_2$$
  $$\lambda = 0.141 \qquad \hat{Y} = 19.46 + 3.04X_1 - 2.10X_2$$
  $$\lambda = 1.152 \qquad \hat{Y} = 19.58 + 2.81X_1 - 1.94X_2$$
- As compared to the true model,
  $$Y = 20 + 3X_1 - 2X_2 + \epsilon$$
  the closest coefficient estimations are (arguably) at $\lambda = 0.141$.
- This is 'proof' that even in simple, low dimensional cases, shrinkage estimators can be useful!

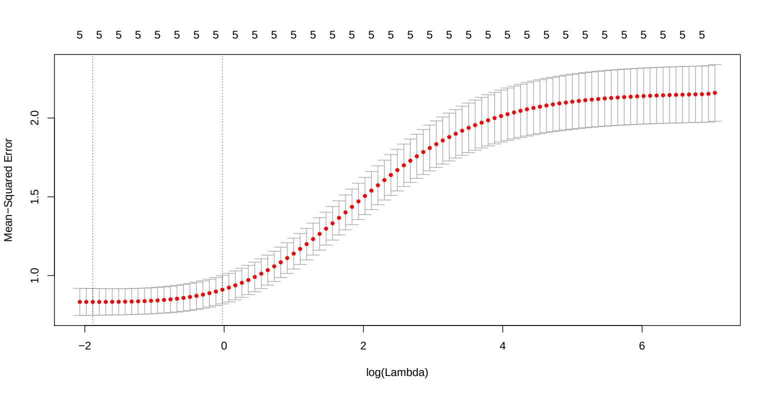
Example: Beer Data

```
> summary(lm(price~., data=beer[,-1]))

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.7174521  0.6807382   5.461 8.56e-07 ***
qlty        -0.0111435  0.0064446  -1.729   0.0887 .
cal         -0.0055034  0.0089957  -0.612   0.5429
alc          0.0764520  0.1752318   0.436   0.6641
bitter       0.0677117  0.0153219   4.419 3.98e-05 ***
malty        0.0002832  0.0099639   0.028   0.9774
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.8661 on 63 degrees of freedom
Multiple R-squared:  0.6678,Adjusted R-squared:  0.6415
F-statistic: 25.33 on 5 and 63 DF,  p-value: 6.588e-14
```

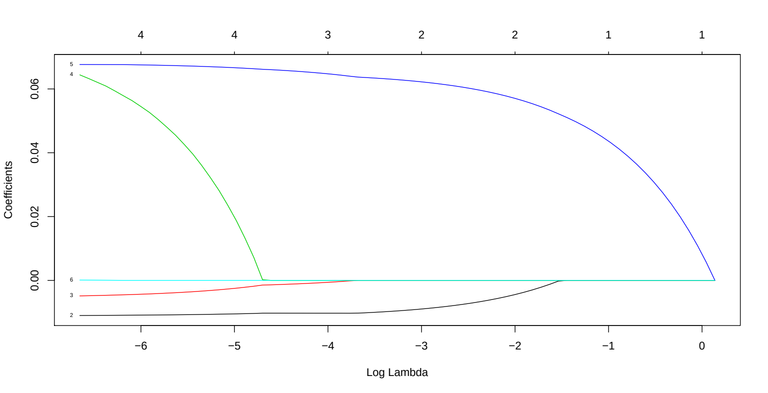
Example: Beer Data - RR



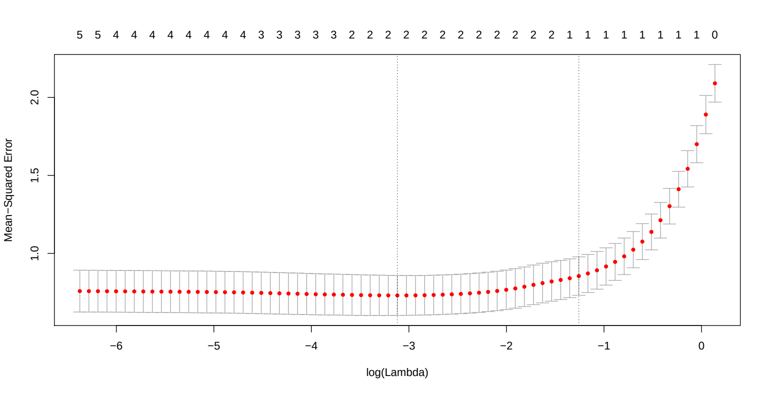
Example: Beer Data - RR



Example: Beer Data - lasso



Example: Beer Data - lasso



Example: Beer Data Summary

	LM	RR	LASSO
(Intercept)	3.7174521392	3.6868302681	3.327790179
qlty	-0.0111434572	-0.0122442317	-0.008507677
cal	-0.0055034319	-0.0009067841	.
alc	0.0764519623	0.0426635115	.
bitter	0.0677117274	0.0474364378	0.061693640
malty	0.0002831604	0.0098597370	.

- While the true model is unknown, we do know a reduced model is more useful for this data (from previous analyses).

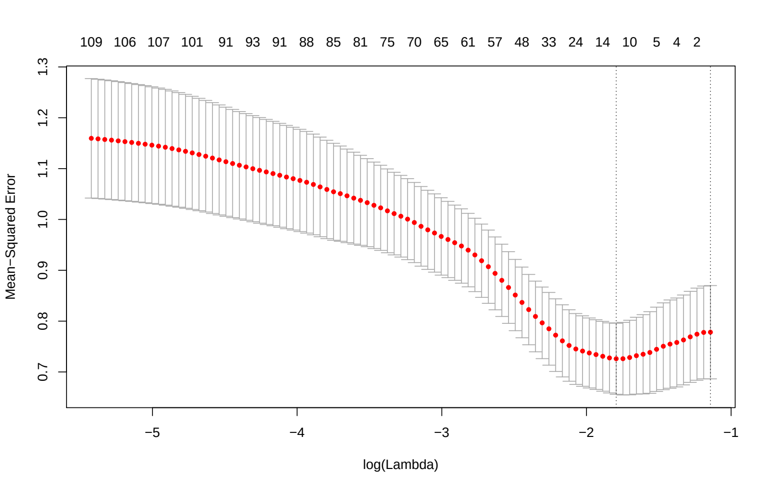
Benefits

- Examples shown thus far are relatively uncommon examples for RR/LASSO because they do not showcase the core benefits of regularization
- Recall (perhaps) that least squares cannot be fit when $p > n$
- No longer the case with regularization in effect! So LASSO is a common technique for high-dimensional data sets.
- Relatedly, the removal of variables takes care of multicollinearity problems.

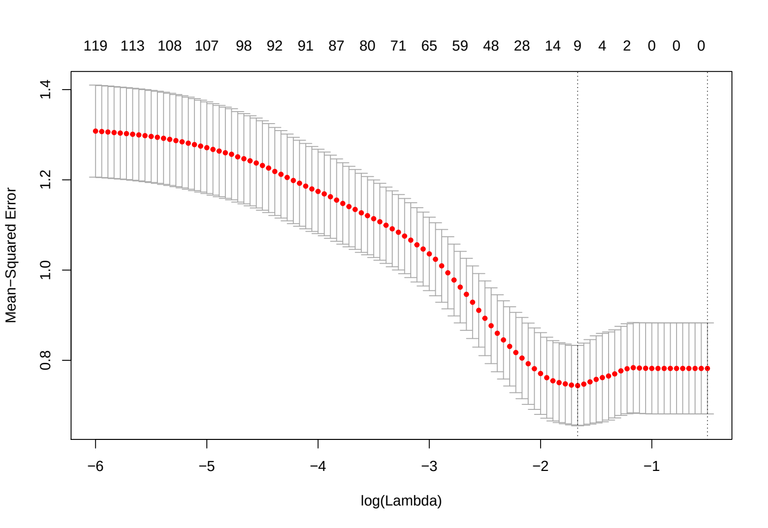
Example: High-Dimensional

```
> bmat <- matrix(rnorm(50000), nrow=100)
> dim(bmat)
[1] 100 500
> y <- rnorm(100)
> bsimcv <- cv.glmnet(bmat, y, alpha=1)
> plot(bsimcv)
```

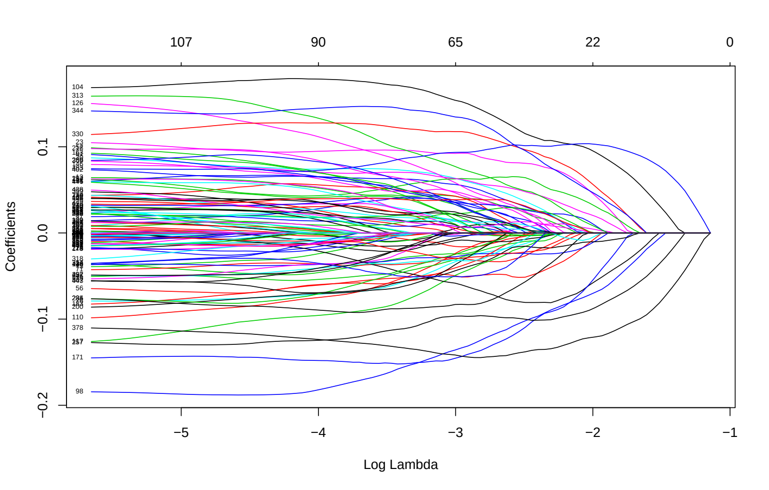
Example: High-Dimensional



Example: High-Dimensional



Example: High-Dimensional



More Thoughts

- Some questions have arisen over the years...let's take a deeper dive on:
- Why does RR lead to no variables removed, while LASSO does remove variables?
- What is the effect of scaling on the variables?

Variable Removal

- Firstly, note that the RR optimization, minimizing

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

could be equivalently written as the standard RSS optimization, minimizing RSS

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2$$

subject to $\sum_{j=1}^{p} \beta_j^2 \leq t$ where t is some positive tuning parameter.

- For LASSO, the constraint would be adjusted to $\sum_{j=1}^{p} |\beta_j| \leq t$
- This formulation will aid visualizations, but may also aid your understanding of regularization in general.

Variable Removal

- Now, why does this change in constraint for LASSO lead to variable removal?
- Hopefully the interactive visualizations in the app will shed some light on this!!

Variable Removal

From ESL...

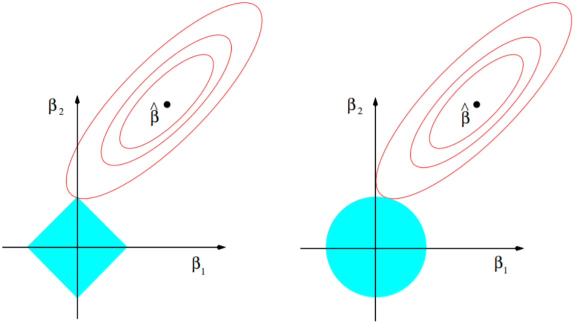


FIGURE 3.11. *Estimation picture for the lasso (left) and ridge regression (right). Shown are contours of the error and constraint functions. The solid blue areas are the constraint regions $|\beta_1| + |\beta_2| \leq t$ and $\beta_1^2 + \beta_2^2 \leq t^2$, respectively, while the red ellipses are the contours of the least squares error function.*

Scaling

- While it is true that you don't need to worry about scaling when using glmnet in R (it auto-standardizes and retransforms back), it's worth exploring why that automation is important.
- It is relatively easy to see why scaling DOES have an effect on LASSO and RR, and why implementations will generally auto-scale.
- The penalty terms, say $\sum_{j=1}^{p} \beta_j^2 \leq t$, describe a 'fair' relationship between coefficients, assuming the coefficients are on similar scales...
- Let's walk through this a bit

Scaling

```
> set.seed(35521)
> x1 <- rnorm(30)
> x2 <- rnorm(30)
> y <- x1 + x2 + rnorm(length(x1), sd=0.025)
```

Scaling

- Note that the RSS term is scale invariant on the predictors.

```
> summary(lm(y~x1+x2))

Call:
lm(formula = y ~ x1 + x2)

Residuals:
      Min        1Q    Median        3Q       Max
-0.059405 -0.011357 -0.001575  0.021256  0.037663

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.008552   0.004677  -1.829   0.0785 .
x1           1.003029   0.005007 200.334   <2e-16 ***
x2           0.993667   0.005484 181.198   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.02426 on 27 degrees of freedom
Multiple R-squared:  0.9996,Adjusted R-squared:  0.9996
F-statistic: 3.372e+04 on 2 and 27 DF,  p-value: < 2.2e-16
```

Scaling

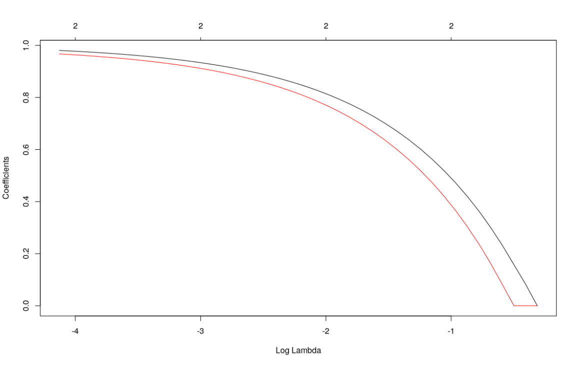
```
> x22 <- x2/10
> summary(lm(y~x1+x22))

Call:
lm(formula = y ~ x1 + x22)

Residuals:
      Min        1Q    Median        3Q       Max
-0.059405 -0.011357 -0.001575  0.021256  0.037663

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.008552   0.004677  -1.829   0.0785 .
x1           1.003029   0.005007 200.334   <2e-16 ***
x22          9.936672   0.054839 181.198   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.02426 on 27 degrees of freedom
Multiple R-squared:  0.9996,Adjusted R-squared:  0.9996
F-statistic: 3.372e+04 on 2 and 27 DF,  p-value: < 2.2e-16
```

Scaling

```
> x12 <- x1/10
> summary(lm(y~x12+x2))

Call:
lm(formula = y ~ x12 + x2)

Residuals:
      Min        1Q    Median        3Q       Max
-0.059405 -0.011357 -0.001575  0.021256  0.037663

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.008552   0.004677  -1.829   0.0785 .
x12         10.030291   0.050068 200.334   <2e-16 ***
x2           0.993667   0.005484 181.198   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.02426 on 27 degrees of freedom
Multiple R-squared:  0.9996,Adjusted R-squared:  0.9996
F-statistic: 3.372e+04 on 2 and 27 DF,  p-value: < 2.2e-16
```

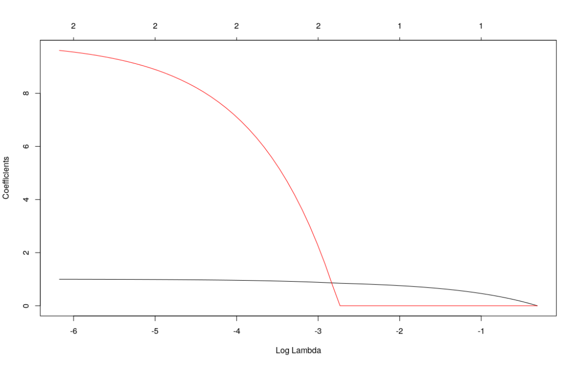
Scaling

```
> run1 <- cv.glmnet(cbind(x1,x2), y, standardize=FALSE)
> plot(run1$glmnet.fit, "lambda")
```



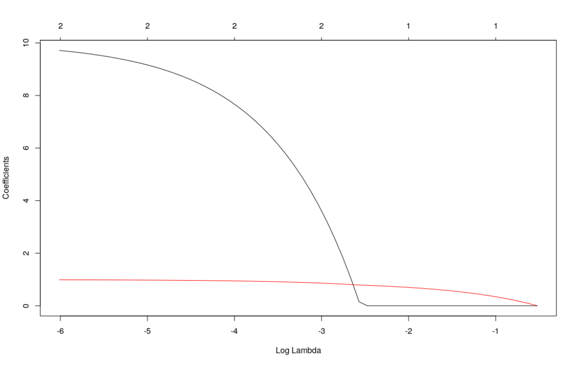
Scaling

```
> run2 <- cv.glmnet(cbind(x1,x22), y, standardize=FALSE)
> plot(run2$glmnet.fit, "lambda")
```



Scaling

```
> run3 <- cv.glmnet(cbind(x12,x2), y, standardize=FALSE)
> plot(run3$glmnet.fit, "lambda")
```



THE UNIVERSITY OF BRITISH COLUMBIA