ISLR Chapter 6 Exercises

2023-07-01

Contents


```
library(ISLR)
library(tools)
library(ggplot2)
```

Conceptual

Question 1

a Best subset will have the smallest training RSS, as for a given number of variables in the model, the models considered by forward / backward stepwise selection will be a subset of the models chosen by best subset selection. Best subset therefore looks at every model that forward / backward stepwise selection looks at, in addition to other models not examined by the other two procedures. Since it is possible that some of these other models have lower training RSS, the model chosen by best subset will have RSS equal to or lower than the models chosen by forward/backward stepwise selection.

b This is impossible to answer; best subset might have picked a model that is overfit to the data, since it has more candidate models to choose from, in which case the other two methods would have better test RSS, or it might have picked a model that actually describes the response-predictor relationship better than the other two methods, in which case it would have better test RSS than the other methods.

c

i True

ii True

iii False

iv False

v False

Question 2

a `iii` is correct because lasso gives biased estimates of the regression coefficients, but they have lower variance than the OLS estimates. Depending on the balances of these two competing factors, the prediction accuracy could be worse or better.

b `iii`, for the same reasons as a.

c `ii` as non-linear methods are more flexible and hence have lower bias but higher variance.

Question 3

a `iv`. The training error will always decrease because as we increase `s`, we decrease the effect of the inequality constraint so that the minimization of the sum of squared errors dominates the objective function. As `s` approaches `infinity`, there is no inequality constraint, so the objective purely minimizes the sum of squared errors, which is the RSS.

b `ii`. The test error will initially decrease, as the decrease in bias will outweigh the increase in variance. As we continue to increase `s`, however, the increase in variance will outweigh the decrease in bias and the test error will start to increase.

c `iii`. Variance always increases as model flexibility increases, so increasing `s` will always increase variance.

d `iv`. Bias always decreases as model flexibility increases, so increasing `s` will always decrease bias.

e `v`. The irreducible error is constant, as this is a property of the test point rather than the trained model.

Question 4

a `iii`. The training error will always increase because as we increase λ, we increase the effect of the inequality constraint so that the minimization of the sum of squared errors plays less of a role in the objective function. As λ approaches `infinity`, all the coefficients except the intercept approach 0, and assuming the predictors have been centered to have mean 0 the train RSS will simply be the scaled variance of the target variable, $(\bar{y} - y_i)^2$

b `ii`. The test error will initially decrease, as the decrease in variance will outweigh the increase in bias. As we continue to increase λ, however, the increase in bias will outweigh the decrease in variance and the test error will start to increase.

c `iv`. Variance always decreases as model flexibility decreases, so increasing λ will always decrease variance.

d `iii`. Bias always increases as model flexibility decreases, so increasing λ will always increase bias.

e `v`. The irreducible error is constant, as this is a property of the test point rather than the trained model.

Question 5

a $\min_{\beta_1,\beta_2} \sum_{i=1}^{2} (\beta_1 x_{i1} + \beta_2 x_{i2} - y_i)^2 + \sum_{j=1}^{2} \beta_j^2$

b Differentiate with respect to β_1 and set equal to zero to solve for beta:

$$2\sum_{i=1}^{2} x_{i1}(\beta_1 x_{i1} + \beta_2 x_{i2} - y_i) + 2\lambda\beta_1 = 0$$

$$\sum_{i=1}^{2}(x_{i1}^2\beta_1 + x_{i1}^2\beta_2 - y_i) + \lambda\beta_1 = 0$$

$$\sum_{i=1}^{2} x_{i1}^2\beta_1 + \lambda\beta_1 = \sum_{i=1}^{2} -x_{i1}^2\beta_2 + y_i x_i$$

$$\beta_1 = \frac{\sum_{i=1}^{2} -x_{i1}^2\beta_2 + y_i x_i}{\lambda + \sum_{i=1}^{2} x_{i1}^2}$$

Similarly we find that

$$\beta_2 = \frac{\sum_{i=1}^{2} -x_{i1}^2\beta_1 + y_i x_i}{\lambda + \sum_{i=1}^{2} x_{i1}^2}$$

The symmetry in these expressions leads us to conclude that $\beta_1 = \beta_2$

c $\min_{\beta_1,\beta_2} \sum_{i=1}^{2}(\beta_1 x_{i1} + \beta_2 x_{i2} - y_i)^2 + \lambda\sum_{j=1}^{2}|\beta_j|$

d This part is easier if we rewrite the above in its alternative formulation,

$$\min_{\beta_1,\beta_2} \sum_{i=1}^{2}(\beta_1 x_{i1} + \beta_2 x_{i2} - y_i)^2 \ s.t. \sum_{j=1}^{2}|\beta_j| \leq t$$

The level curves of the function we are minimizing are given by

$$\beta_1 x_{11} + \beta_2 x_{11} - y_1 + \beta_1 x_{21} + \beta_2 x_{21} - y_2 = C$$

$$\beta_1 + \beta_2 = C$$

Which implies that the level curves are lines in the β_1, β_2 space. If λ is non-zero, this implies that the inequality constraint is binding, or that the inequality is in fact an equality. This means that the solution must lie on one of the edges of the diamond represented by $\sum_{j=1}^{2}|\beta_j| \leq t$. Since these edges are also straight lines, they touch the level curve at infinitely many points.

Question 6

First I will derive a more general form of the ridge regression estimates when the design matrix is orthogonal. I assume that the predictors have been centered to have mean 0 as well, so that the estimate for β_0 is equal to $\hat{y}$ for the OLS estimates, ridge estimates, and lass estimates. The vector β that follows is then the vector of coefficients with the first element removed.

Least squares loss function:

$$\min_{\beta}(y - X\beta)^T(y - X\beta)$$

$\min_{\beta}(y^T - \beta^T X^T)(y - X\beta)$

$\min_{\beta} y^T y - y^T X\beta - \beta^T X^T y + \beta^T X^T X\beta$

$\min_{\beta} y^T y - 2\beta^T X^T y + \beta^T X^T X\beta$

Differentiate with respect to β and set equal to zero:

$-2X^T y + 2X^T X\beta = 0$

$\beta = (X^T X)^{-1} X^T y$

Since X is an orthogonal matrix, $X^T X = I$ and $(X^T X)^{-1} = I$

Then $\beta = X^T y$

Ridge regression loss function:

$\min_{\beta}(y - X\beta)^T(y - X\beta) + \lambda\beta^T\beta$

$\min_{\beta}(y^T - \beta^T X^T)(y - X\beta) + \lambda\beta^T\beta$

$\min_{\beta} y^T y - y^T X\beta - \beta^T X^T y + \beta^T X^T X\beta + \lambda\beta^T\beta$

$\min_{\beta} y^T y - 2\beta^T X^T y + \beta^T X^T X\beta + \lambda\beta^T\beta$

Differentiate with respect to β and set equal to zero:

$-2X^T y + 2X^T X\beta + 2\lambda\beta = 0$

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

Since X is an orthogonal matrix, $X^T X + \lambda I = (1 + \lambda)I$ and $(X^T X + \lambda I)^{-1} = \frac{1}{1+\lambda} I$

Then $\beta = \frac{X^T y}{1+\lambda} = \frac{\beta_{OLS}}{1+\lambda}$

Lasso loss function :

$\min_{\beta}(y - X\beta)^T(y - X\beta) + \lambda||\beta||_1$

$\min_{\beta}(y^T - \beta^T X^T)(y - X\beta) + \lambda||\beta||_1$

$\min_{\beta} y^T y - y^T X\beta - \beta^T X^T y + \beta^T X^T X\beta + \lambda||\beta||_1$

$\min_{\beta} y^T y - 2\beta^T X^T y + \beta^T X^T X\beta + \lambda||\beta||_1$

$\min_{\beta} y^T y - 2\beta^T X^T y + \beta^T\beta + \lambda||\beta||_1$

$\min_{\beta} y^T y - 2\beta^T \beta_{ols} + \beta^T\beta + \lambda||\beta||_1$

Take an element `i` of the gradient and set equal to 0. We refer to the coefficient corresponding to this as β now, which is a scalar rather than a vector.

There are two cases:

1. $\beta_{ols} \geq 0$, in which case $\beta \geq 0$ so as to minimize the loss function.

Then $-2\beta_{ols} + 2\beta + \lambda = 0$

$\beta = \beta_{ols} - \frac{\lambda}{2}$

Since we assumed that $\beta \geq 0$, we take only the positive part of this equation:

$\beta = (\beta_{ols} - \frac{\lambda}{2})^+$

$\beta = (sign(\beta_{ols})|\beta_{ols}| - \frac{\lambda}{2})^+$

2. $\beta_{ols} \leq 0$, in which case $\beta \leq 0$ so as to minimize the loss function.

Then $-2\beta_{ols} + 2\beta - \lambda = 0$

$\beta = \beta_{ols} + \frac{\lambda}{2}$

Since we assumed that $\beta \leq 0$, we take only the negative part of this equation:

$\beta = (\beta_{ols} + \frac{\lambda}{2})^-$

$\beta = sign(|\beta_{ols}|)(|\beta_{ols}| - \frac{\lambda}{2})^+$

We get the same equation for β in both cases.

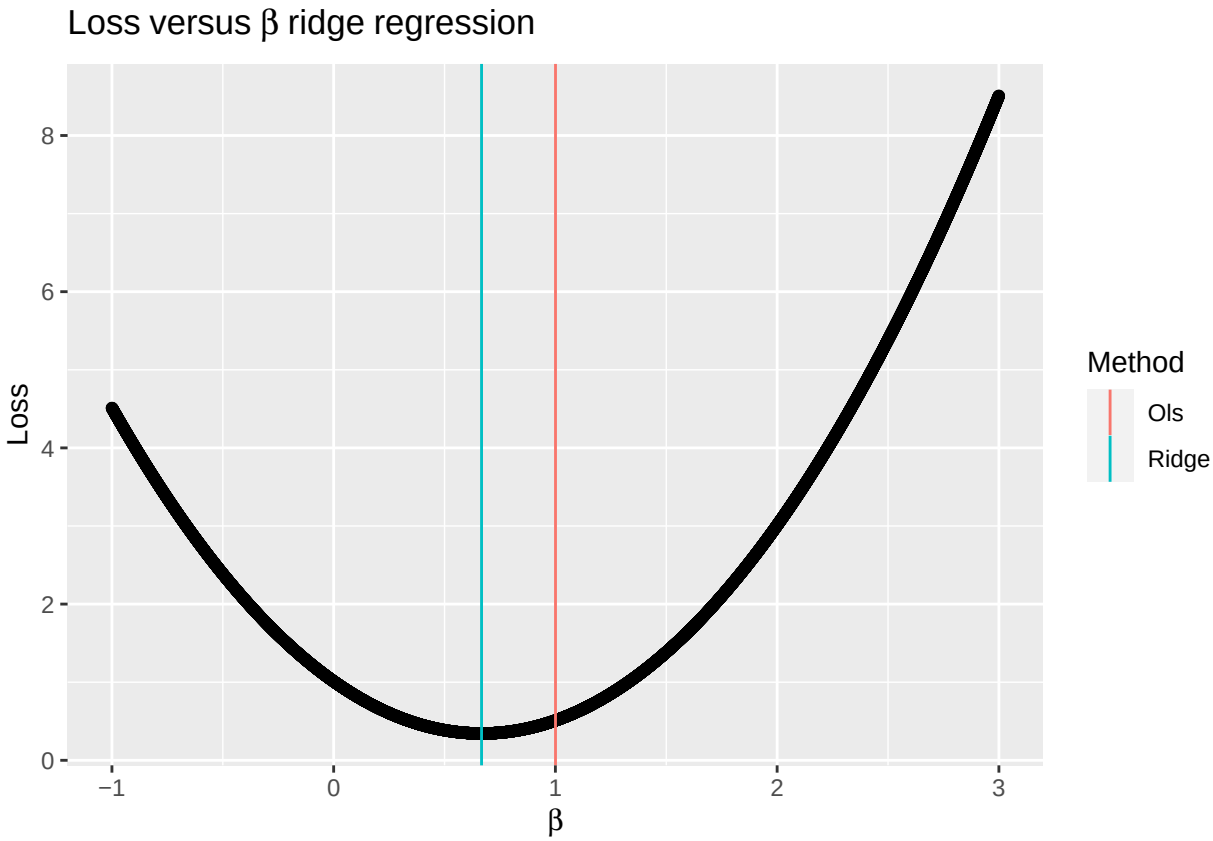
Question 6

I answer this using the general form of the ridge regression / lasso estimates in the case of a centered variable with magnitude 1.

a

```
generate_response <- function(df, population_coefficient) {
  df$y <- 10 + population_coefficient * df$x + df$eps
  df
}

compare_theoretical_to_simulation <- function(df, betas, method = c("ridge", "lasso")) {

  lambda <- 0.5

  ols_model <- lm(y ~ x, data = df)
  ols_coef <- coef(ols_model)[[2]]

  method <- match.arg(method)
  regularizer <- switch(
    method,
    ridge = function(beta) beta ^ 2,
    lasso = function(beta) abs(beta)
  )

  scaling_factor <- switch(
    method,
    ridge = function(lambda, beta_ols) beta_ols / (1 + lambda),
    lasso = function(lambda, beta_ols) {
      if (abs(beta_ols) >= lambda / 2) {
        sign(beta_ols) * (abs(beta_ols) - lambda / 2)
      } else {
        0
      }
    }
  )

  loss <- sapply(
```

```
    betas,
    function(beta, df, lambda, regularizer) {
      sum((df$y - mean(df$y) - beta * df$x) ^ 2) + lambda * regularizer(beta)
    },
    df = df,
    lambda = lambda,
    regularizer = regularizer
  )

  df_loss <- data.frame(beta = betas, loss = loss)
  df_vertical_lines <- data.frame(
    estimates = c(ols_coef, scaling_factor(lambda, ols_coef)),
    method = c("ols", method)
  )

  plot_results(df_loss, df_vertical_lines)

}

plot_results <- function(df_loss, df_vertical_lines) {
  ggplot2::ggplot(data = df_loss) +
    ggplot2::geom_point(ggplot2::aes(x = beta, y = loss)) +
    ggplot2::geom_vline(data = df_vertical_lines, ggplot2::aes(xintercept = estimates, color = tools::t
    ggplot2::labs(x = expression(beta), y = "Loss", color = "Method")
}

nrows <- 10000

set.seed(1)
df <- data.frame(x = rnorm(nrows), eps = rnorm(nrows, sd = 0.001))
df$x <- df$x / sqrt(sum(df$x^2))
df <- generate_response(df, population_coefficient = 1)

betas <- seq(-1, 3, length = 10000)

plot_ridge <- compare_theoretical_to_simulation(df, betas, method = "ridge")
plot_ridge <- plot_ridge +
  ggplot2::ggtitle(expression(Loss~versus~beta~ridge~regression))

plot_ridge
```



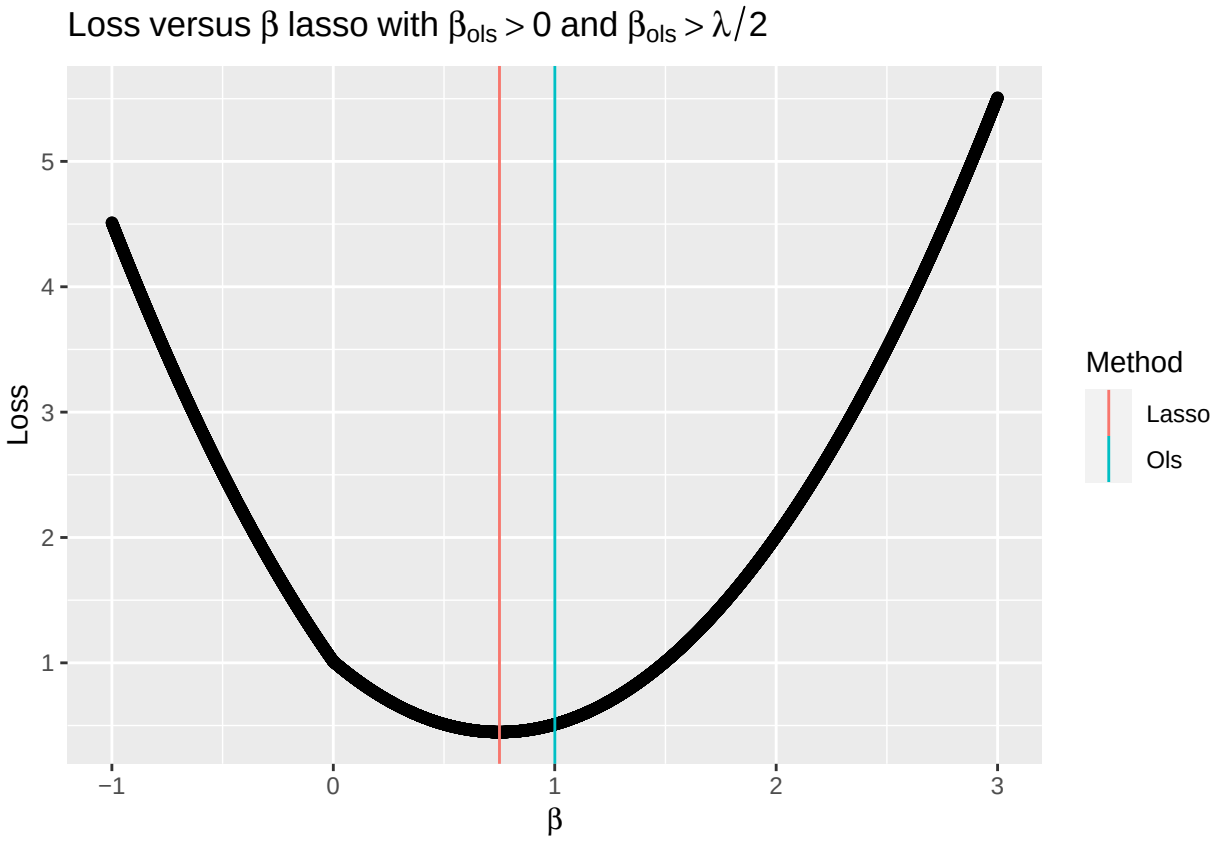
We see that the theoretical value of β_{ridge} aligns well with the minimum of the loss in the simulation.

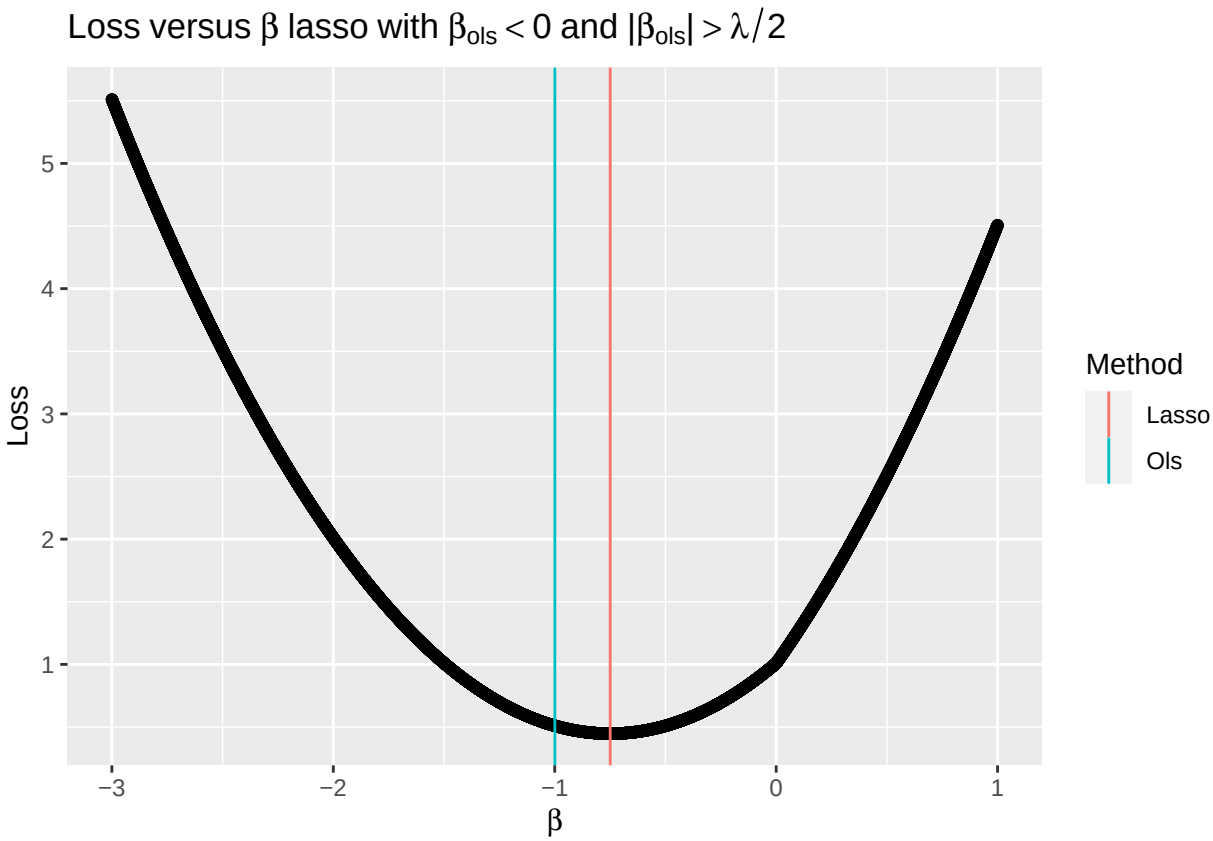
b We will consider three cases here:

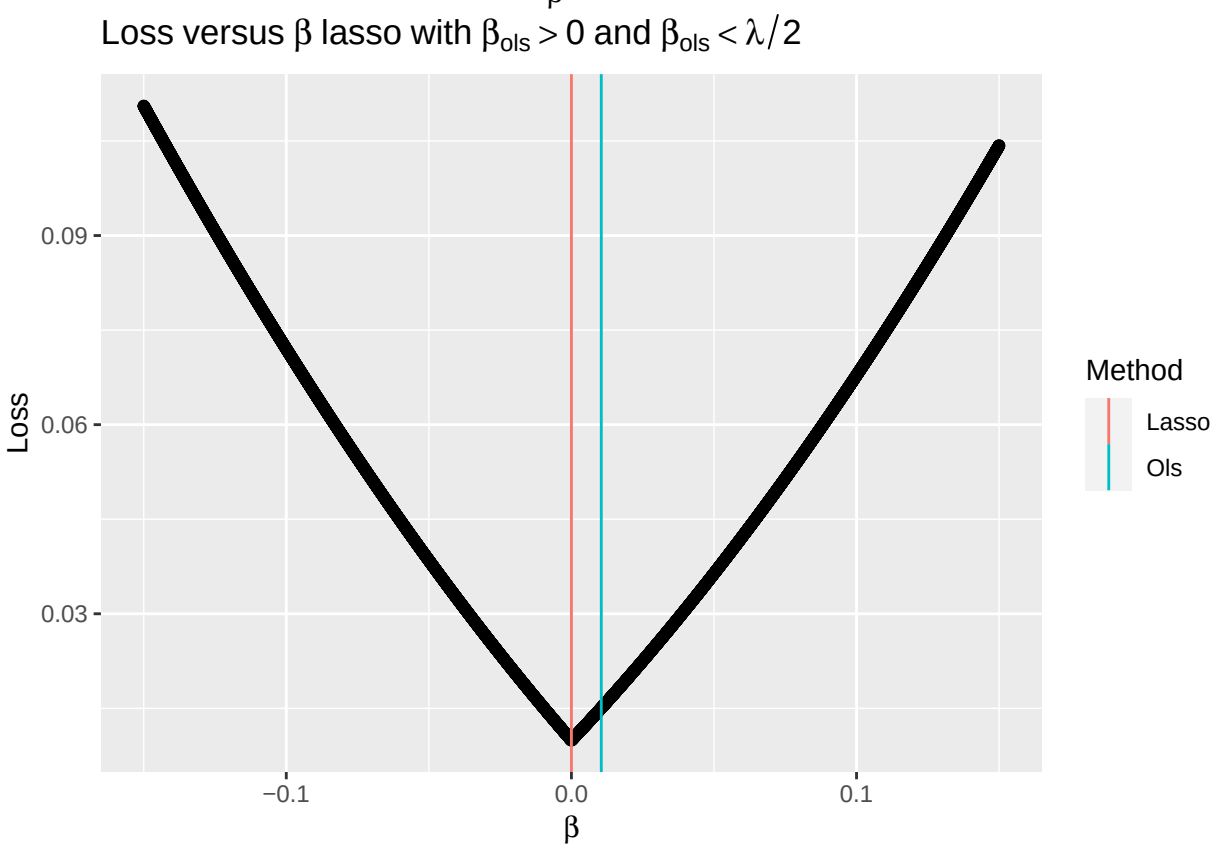
1. $\beta_{ols} < 0$ and $\beta_{ols} < -\frac{\lambda}{2}0$
2. $\beta_{ols} > 0$ and $\beta_{ols} > \frac{\lambda}{2}0$
3. $\beta_{ols} > 0$ and $\beta_{ols} < \frac{\lambda}{2}0$

```
param_list <- list(
  lasso_1 = list(
    betas = seq(-1, 3, length = 10000),
    population_coefficient = 1,
    title = expression(Loss~versus~beta~lasso~with~beta[ols]>0~and~beta[ols]>lambda/2)
  ),
  lasso_2 = list(
    betas = seq(-3, 1, length = 10000),
    population_coefficient = -1,
    title = expression(Loss~versus~beta~lasso~with~beta[ols]<0~and~paste("|", beta[ols], "|")>lambda/2)
  ),
  lasso_3 = list(
    betas = seq(-0.15, 0.15, length = 10000),
    population_coefficient = 0.01,
    title = expression(Loss~versus~beta~lasso~with~beta[ols]>0~and~beta[ols]<lambda/2)
  )
)

for (params in param_list) {
  df_lasso <- generate_response(df, params$population_coefficient)
```

```
  plot <- compare_theoretical_to_simulation(df_lasso, params$beta, method = "lasso")
  plot <- plot + ggplot2::labs(title = params$title)
  print(plot)
}
```




Loss versus β lasso with $\beta_{ols} < 0$ and $|\beta_{ols}| > \lambda/2$
Loss
5
4
3
2
1
−3
−2
−1
0
1
β
Method
Lasso
Ols


Loss versus β lasso with $\beta_{ols} > 0$ and $\beta_{ols} < \lambda/2$
Loss
0.09
0.06
0.03
−0.1
0.0
0.1
β
Method
Lasso
Ols

Applied

“