Exercise 3

Tobias Raidl, 11717659

2023-11-02

Setup

```
load("building.RData")
sample <- sample(c(TRUE, FALSE), nrow(df), replace = TRUE, prob = c(0.7, 0.3))
train <- df[sample, ]
test <- df[!sample, ]
```

1. Ridge Regression

(a)

How can you interpret the plot? This plot shows the ridge regression coefficients for varying values of the tuning parameter λ. On top we can see the number of variables in the model. With increasing λ the coefficients shrink towards 0.

Which default parameters are used for lambda? When automatically generated, the λ sequence is determined by lambda.max and lambda.min.ratio. The latter is the ratio of smallest value of the generated λ sequence (say lambda.min) to lambda.max. The program generates nlambda values linear on the log scale from lambda.max down to lambda.min. lambda.max is not user-specified but is computed from the input x and y: it is the smallest value for λ such that all the coefficients are zero. For alpha = 0 (ridge) lambda.max would be infinity: in this case we pick a value corresponding to a small value for alpha close to zero.)

What is the meaning of the parameter alpha? Setting $\alpha = 1$ yields Lasso regression (default), and $\alpha = 0$ is doing Ridge regression

```
library(dplyr)
```

```
## Warning: Paket 'dplyr' wurde unter R Version 4.1.3 erstellt
```

```
##
## Attache Paket: 'dplyr'
```

```
## Die folgenden Objekte sind maskiert von 'package:stats':
##
##     filter, lag
```

```
## Die folgenden Objekte sind maskiert von 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(glmnet)
```

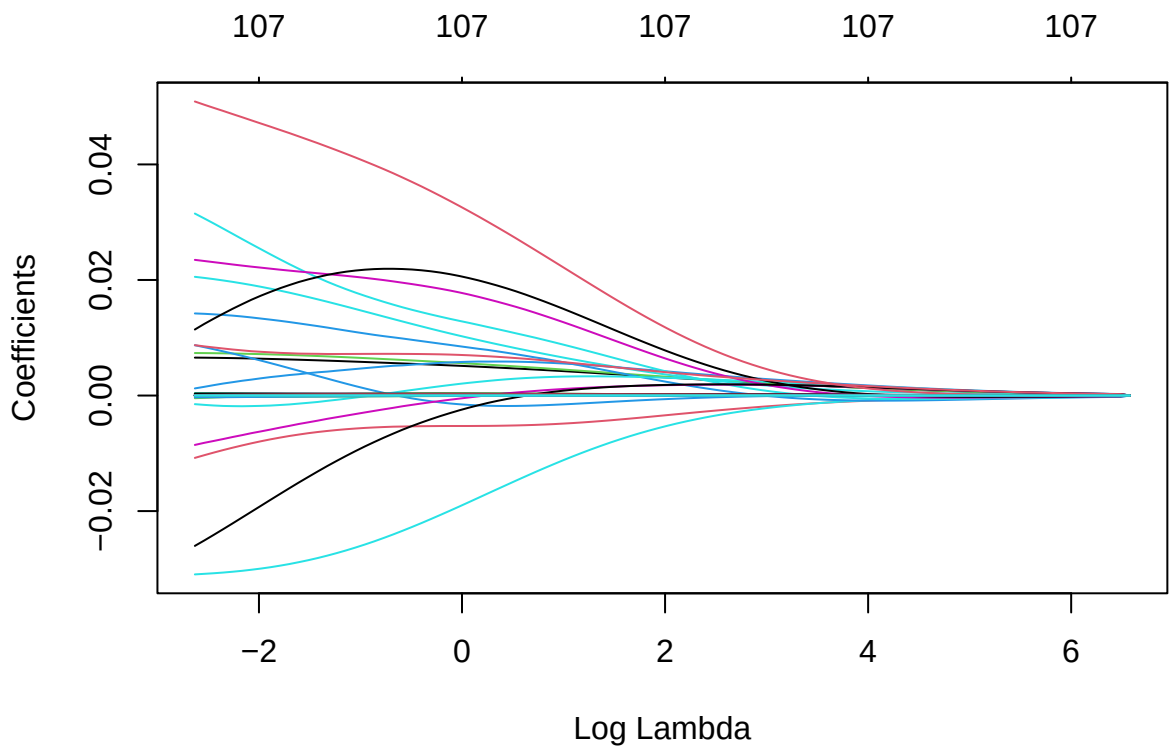
```
## Warning: Paket 'glmnet' wurde unter R Version 4.1.3 erstellt
```

```
## Lade nötiges Paket: Matrix
```

```
## Warning: Paket 'Matrix' wurde unter R Version 4.1.3 erstellt
```

```
## Loaded glmnet 4.1-7
```

```
ridge <- glmnet(as.matrix(df[, -1]), df[, 1], alpha = 0)
# print(ridge)
plot(ridge, xvar = "lambda")
```

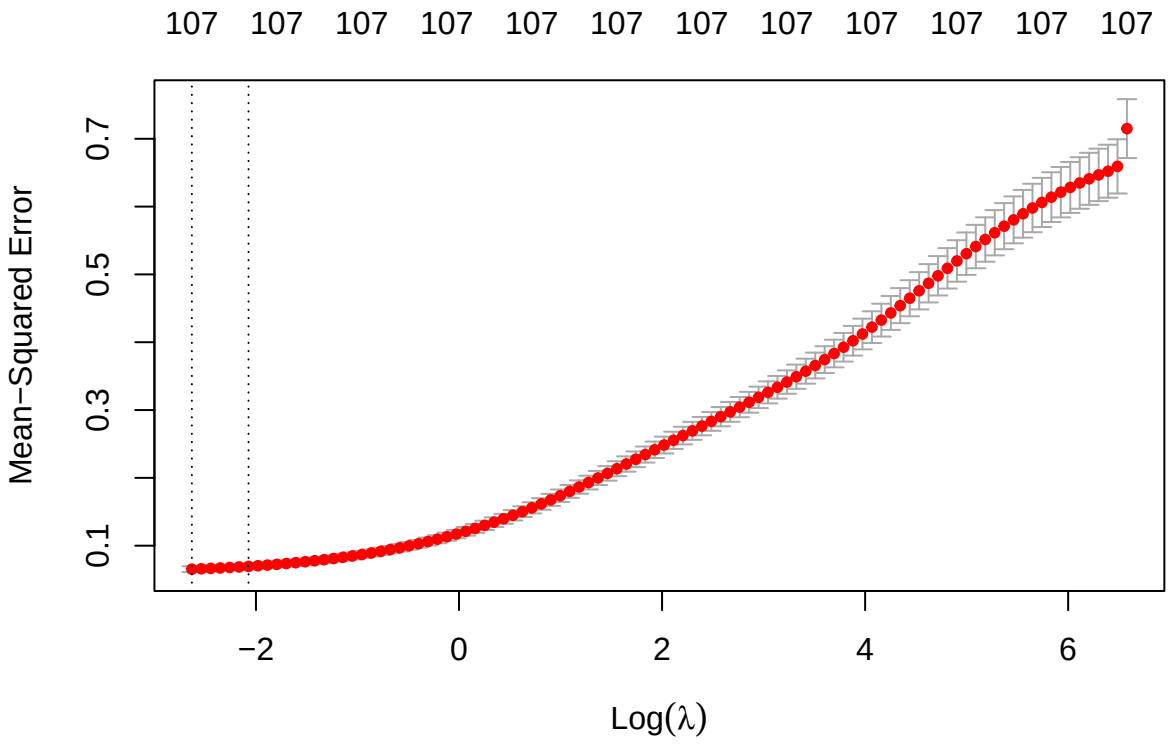


(b)

This plot shows the cross-validated MSE for different values of λ in Ridge regression.

How do you obtain the optimal tuning parameter and the regression coefficients? shows the MSE together with their standard errors. The left dashed line indicates the smallest MSE, and the right dashed line points at the optimal λ for with the MSE is still below the bound defined by the smallest MSE plus its standard error. This λ is selected if we go for the "one-standard error rule". Thereofre we choose "lambda.1se" as lambda.

```
ridge.cv <- cv.glmnet(as.matrix(df[, -1]), df[, 1], alpha = 0)
plot(ridge.cv)
```



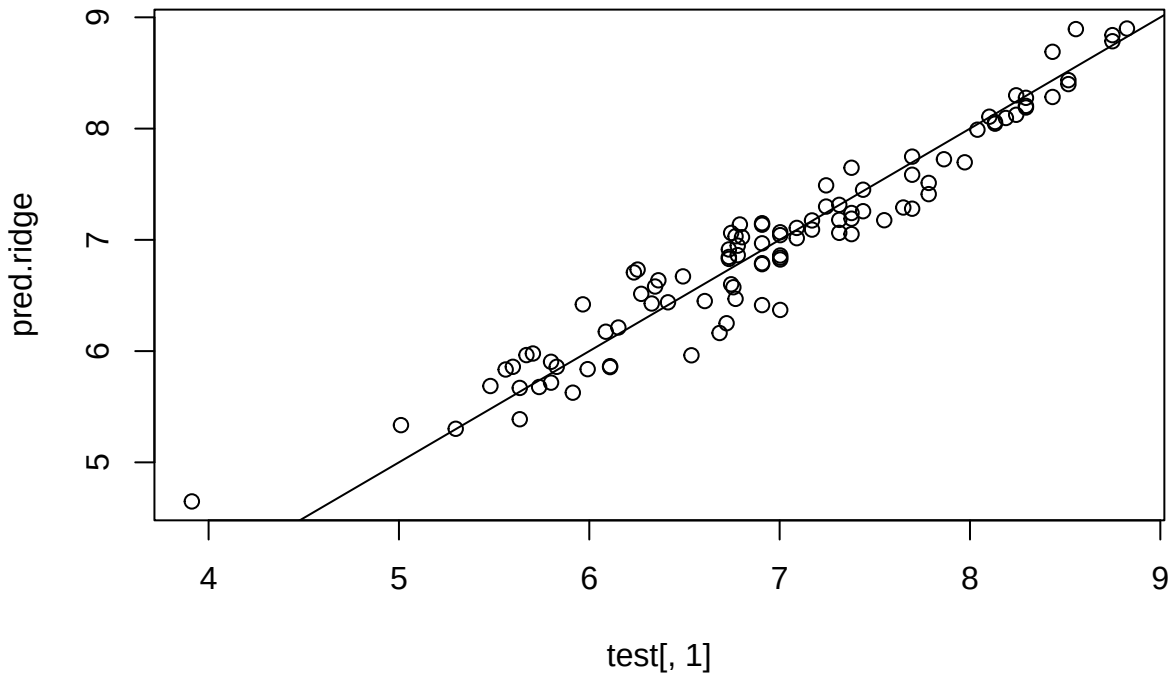

```
# coef(ridge.cv, s='lambda.1se')
```

(c)

```
pred.ridge <- predict(ridge.cv, newx = as.matrix(test[, -1]), s = "lambda.1se")
cat(paste("R square:", cor(test[, 1], pred.ridge)^2, "\n", "RMSE:", sqrt(mean((test[,
    1] - pred.ridge)^2)), "\n"))
```

```
## R square: 0.932581769354399
##  RMSE: 0.245998984717992
```

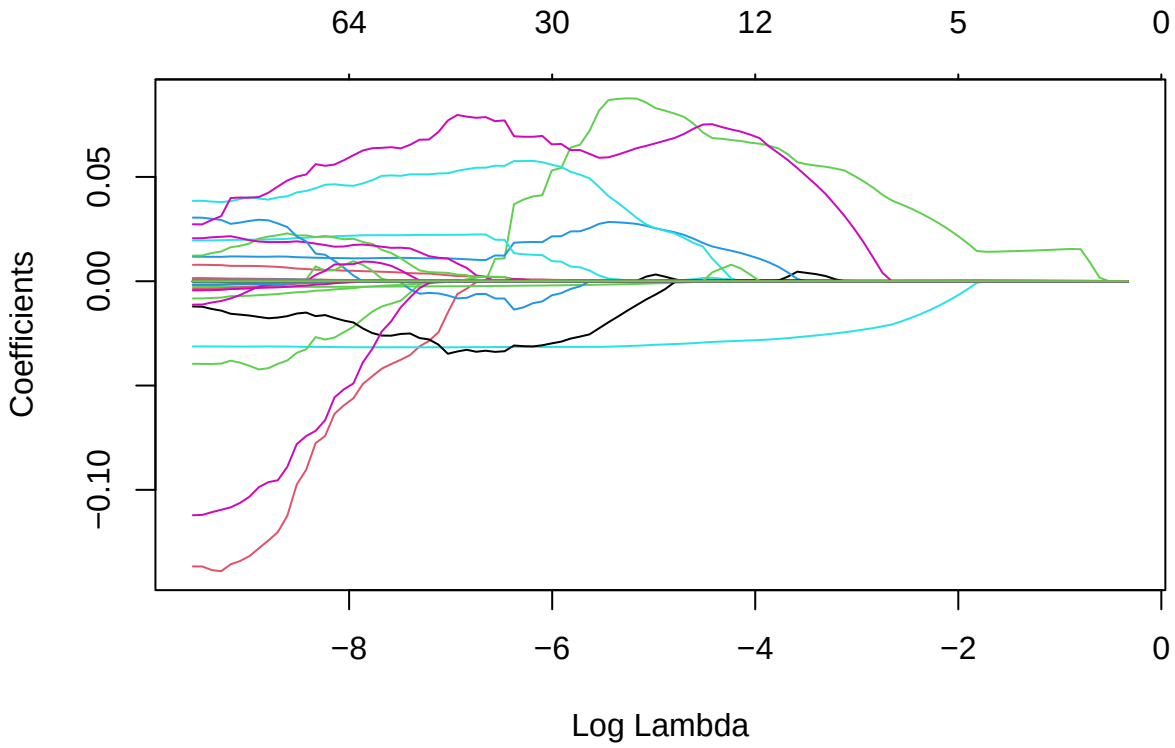
```r
plot(test[, 1], pred.ridge)
abline(c(0, 1))
```



2. Lasso Regression

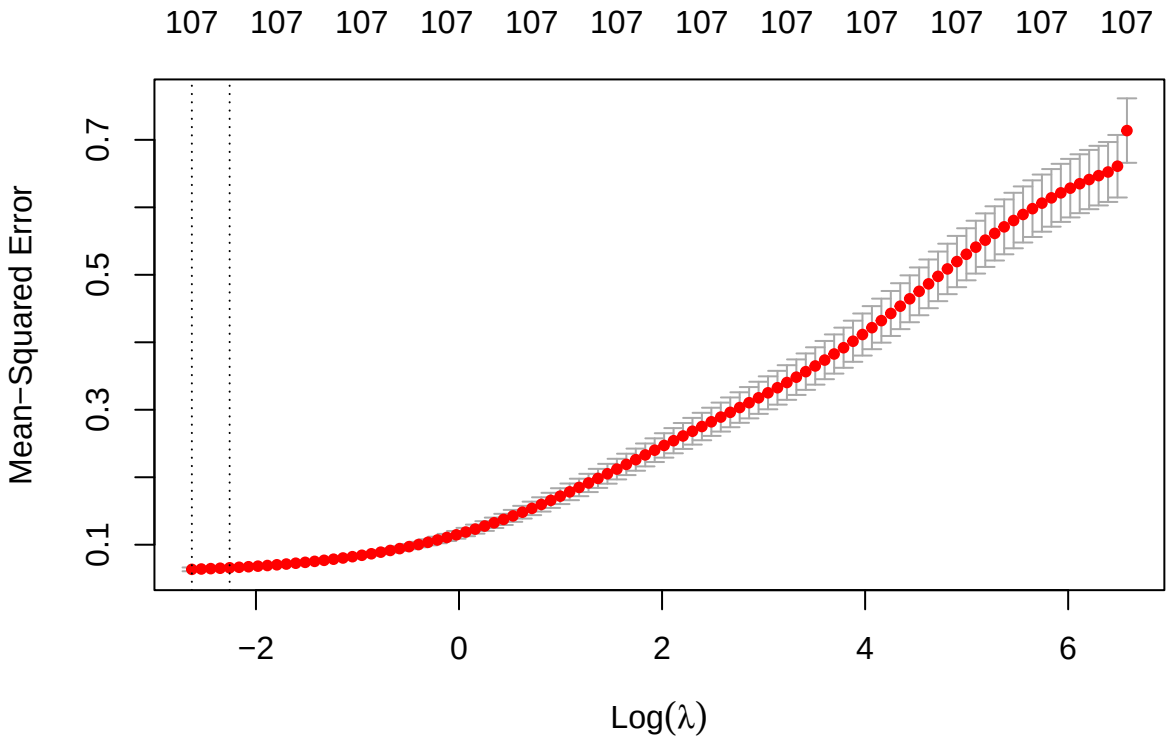
(a)

```r
library(dplyr)
library(glmnet)
lasso <- glmnet(as.matrix(df[, -1]), df[, 1], alpha = 1)
# print(lasso)
plot(lasso, xvar = "lambda")
```



(b)

```
lasso.cv <- cv.glmnet(as.matrix(df[, -1]), df[, 1], alpha = 0)
plot(lasso.cv)
```



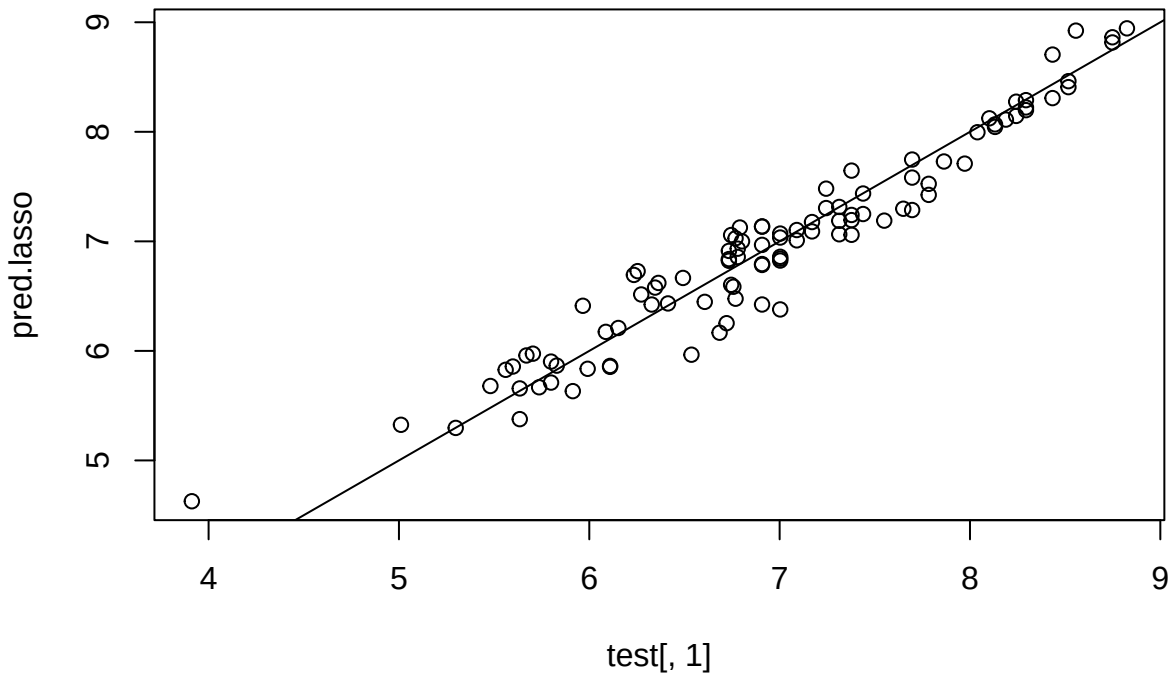

```
# coef(lasso.cv, s='lambda.1se')
```

(c)

```
pred.lasso <- predict(lasso.cv, newx = as.matrix(test[, -1]), s = "lambda.1se")
cat(paste("R square:", cor(test[, 1], pred.lasso)^2, "\n", "RMSE:", sqrt(mean((test[,
    1] - pred.lasso)^2)), "\n"))
```

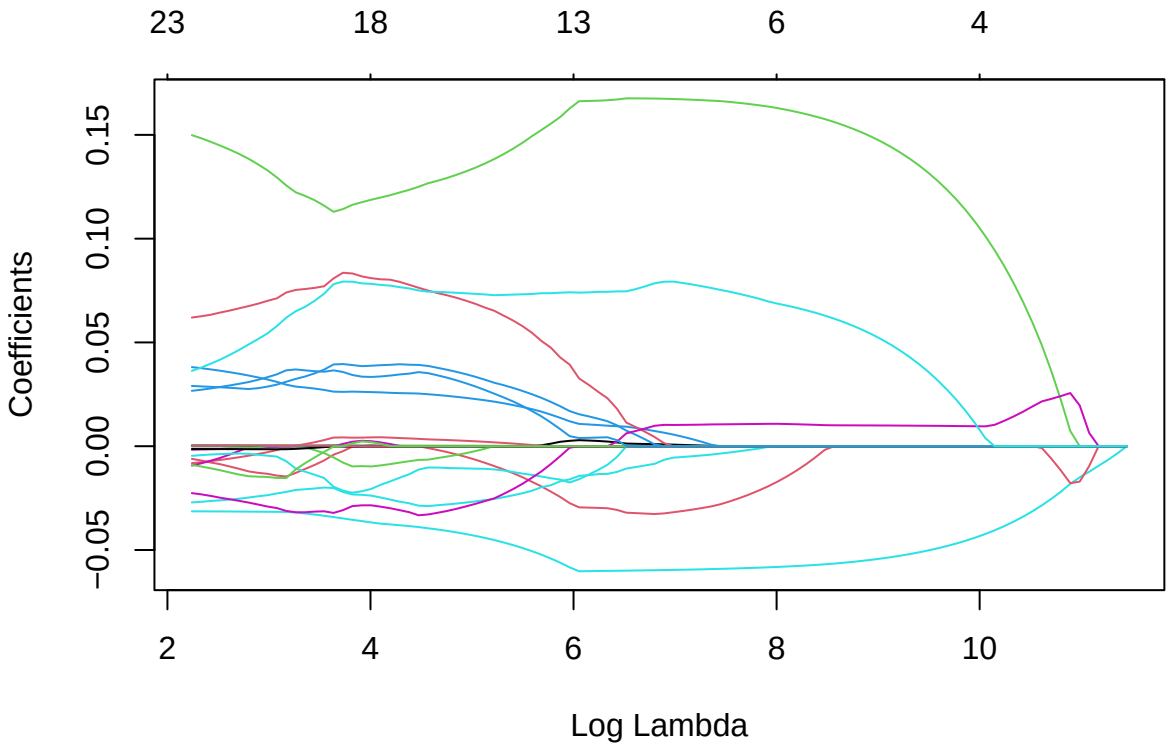
```
## R square: 0.93467211448134
##  RMSE: 0.241889450693133
```

```
plot(test[, 1], pred.lasso)
abline(c(0, 1))
```



3. Adaptive Lasso Regression

```
coef.ridge <- coef(ridge.cv, s = "lambda.1se")
alasso <- glmnet(as.matrix(train[, -1]), train[, 1], penalty.factor =
↪  1/abs(coef.ridge[-1]))
plot(alasso, xvar = "lambda")
```



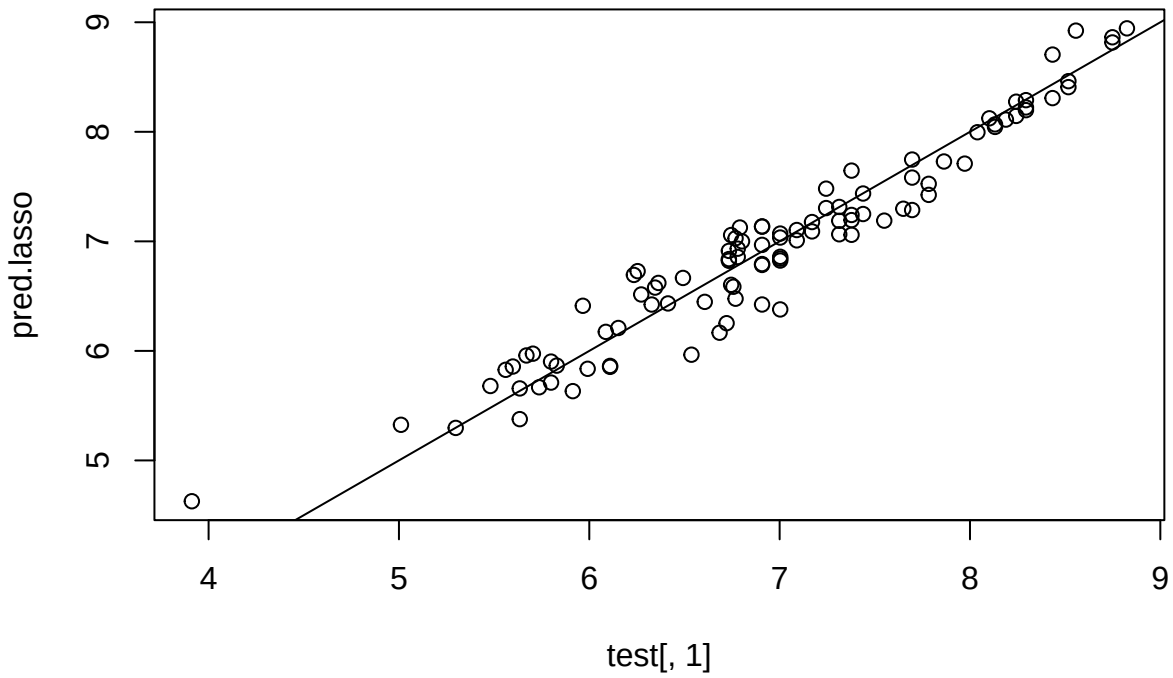

```
alasso.cv <- cv.glmnet(as.matrix(train[, -1]), train[, 1], penalty.factor =
↪  1/abs(coef.ridge[-1]))
# coef(alasso.cv,s='lambda.1se')

pred.alasso <- predict(alasso.cv, newx = as.matrix(test[, -1]), s = "lambda.1se")
cor(test[, 1], pred.alasso)^2  # R^2 for test data
```

```
##      lambda.1se
## [1,]  0.9273591
```

```
sqrt(mean((test[, 1] - pred.alasso)^2))  # RMSE_test
```

```
## [1] 0.2594676
```

```
plot(test[, 1], pred.lasso)
abline(c(0, 1))
```


pred.lasso
test[, 1]
5
6
7
8
9
4
5
6
7
8
9