eNote 8

Ridge and Lasso Regression

Indhold

8.1 Reading material

The similar sections from eNote4 on PCR and eNote6 on PLS, where the general reading for these methods are given, are repeated here:

PCR and the other biased regression methods presented in this course (PLS, Ridge and Lasso) are all together with even more methods (as e.g. MLR=OLS) introduced in each of the three books

- The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second Edition, February 2009, Trevor Hastie, Robert Tibshirani, Jerome Friedman

- Ron Wehrens (2012). Chemometrics With R: Multivariate Data Analysis in the Natural Sciences and Life Sciences. Springer, Heidelberg.(Chapter 8 and 9)
- K. Varmuza and P. Filzmoser (2009). Introduction to Multivariate Statistical Analysis in Chemometrics, CRC Press. (Chapter 4)

The latter two ones are directly linked with R-packages, and here we will most directly use the latter. These things are NOT covered by the Lattin book, so we give here a reading list for the most relevant parts of chapter 4 of the Varmuza and Filzmoser book, when it comes to syllabus content for course 27411:

- Section 4.1 (4p) (Concepts - ALL models)
- Section 4.2.1-4.2.3 (6p)(Errors in (ALL) models)
- Section 4.2.5-4.2.6 (3.5p)(CV+bootstrap - ALL models)
- [Section 4.3.1-4.3.2.1 (9.5p)(Simple regression and MLR (=OLS))]
- [Section 4.5.3 (1p) (Stepwise Variable Selection in MLR)]
- Section 4.6 (2p) (PCR)
- Section 4.7.1 (3.5p) (PLS)
- Section 4.8.2 (2.5p) (Ridge and Lasso)
- Section 4.9-4.9.1.2 (5p) (An example using PCR and PLS)
- Section 4.9.1.4-4.9.1.5 (5p) (An example using Ridge and Lasso)
- Section 4.10 (2p) (Summary)

8.2 Presentation material

8.2.1 Multivariate Calibration

Multivariate calibration

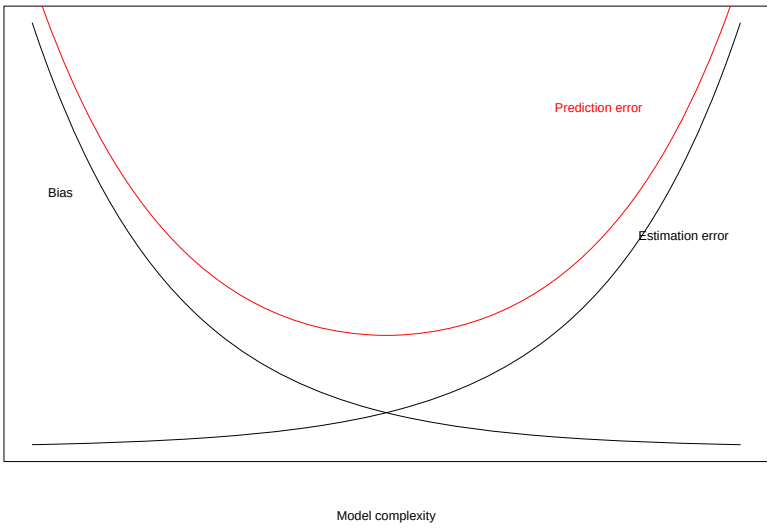
- Classical (chemical) calibration: Model x as function of y
- Predict y fro x by "going backwards".
- We are doing Inverse calibration: $y = f(x)$

- What is a good model: In theory: $f(x) = \mathrm{E}(Y|x = x)$.
- If $Y = \mathrm{E}(Y|x = x) + \epsilon$ and $\hat{f}(x)$ is some model estimator, expected prediction error:

$$\mathrm{E}_{Y|X=x}((y - \hat{f}(x))^2) =$$
$$\mathrm{E}((y - \mathrm{E}(Y|X = x))^2) + \mathrm{E}((f(x) - \hat{f}(x))^2) + (f(x) - \mathrm{E}(Y|X = x))^2$$
$$= \sigma^2 + \mathrm{Var}(\hat{f}(x)) + \mathrm{Bias}^2$$

Optimal predictor

Find optimal predictor by CV



Basic model: MLR $\mathrm{E}(y_i|x_{i1}, x_{i2}, \ldots, x_{ip}) = b_0 + b_1 x_{i1} + b_2 x_{i2} + \cdots + b_p x_{ip}$

$y = xb + \epsilon$

- Estimation: $\hat{b} = (x'x)^{-1}x'y$
- Problem: The parameters are badly estimated!
- HIGH Uncertainty!

- HIGH value of one parameter can go together with a parameter of opposite sign - NOT changing the model

Basic model: MLR

- Uncertainty:

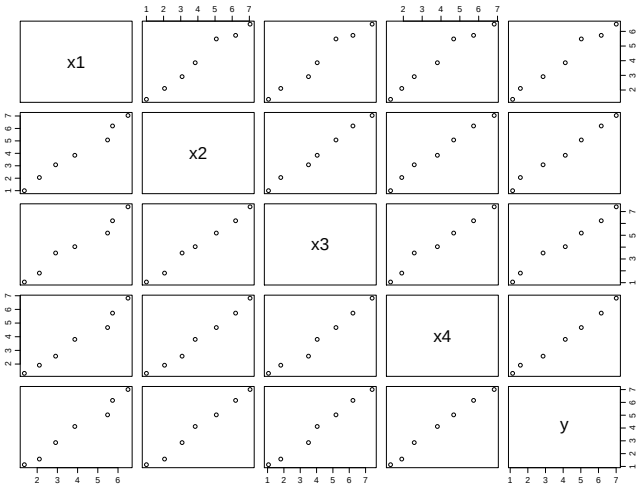
$$\begin{aligned} \text{Var}(\hat{b}) &= \hat{\sigma}^2 (x'x)^{-1} \\ &= \hat{\sigma}^2 \sum_{j=1}^{p} (v_j v_j') / d_j^2 \end{aligned}$$

  vv_j is the jth principal loading vector
  d_j^2 is the jth principal eigenvalue of $x'x$
  $(\text{Var}(xv_j) = d_j^2/n)$

- SMALL principal directions will induce HIGH uncertainty!

8.2.2 Example data

Example Data



MLR results:

	Estimate	Std. Error	t	P-val
Intercept	-0.1668	0.4309	-0.387	0.736
x1	-0.8141	1.6888	-0.482	0.677
x2	-0.1027	0.8635	-0.119	0.916
x3	2.0695	1.4133	1.464	0.281
x4	-0.1354	0.6183	-0.219	0.847

b_3 becomes too large AND the others too small.

8.2.3 Ridge regression

Ridge regression

- Penalizes (numerically) large regression coefficients
- $\hat{b}^{Ridge}$ minimizes (for a given s)

$$\sum_{i=1}^{n}\left(y_i - b_0 - \sum_{j=1}^{p} b_j x_{ij}\right)^2 \text{ subject to } \sum_{j=1}^{p} b_j^2 \leq s$$

- $\hat{b}^{Ridge}$ minimizes (for a given λ)

$$\sum_{i=1}^{n}\left(y_i - b_0 - \sum_{j=1}^{p} b_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{p} b_j^2$$

8.2.4 Lasso regression

Lasso regression

- Penalizes (numerically) large regression coefficients
- $\hat{b}^{Lasso}$ minimizes (for a given s)

$$\sum_{i=1}^{n}\left(y_i - b_0 - \sum_{j=1}^{p} b_j x_{ij}\right)^2 \text{ subject to } \sum_{j=1}^{p} |b|_j \leq s$$

- $\hat{b}^{Lasso}$ minimizes (for a given λ)

$$\sum_{i=1}^{n}\left(y_i - b_0 - \sum_{j=1}^{p} b_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{p} |b|_j$$

8.2.5 Ridge regression: Solution

Ridge regression: Solution

- Add λ to the diagonal of $x^{'}x$:

$$b^{Ridge} = (x^{'}x + \lambda I)^{-1}x^{'}y$$

- RR-predictor: $\hat{y}^{Ridge} = xb^{Ridge}$
- Operational: Try different λs - choose the optimal by Cross Validation.

8.2.6 The L_2-methods

MLR(LS), PCR and RR

$$\hat{y}^{LS} = xb^{LS} = \sum_{j=1}^{p} u_j u_j^{'} y$$

$$\hat{y}^{Ridge} = xb^{Ridge} = \sum_{j=1}^{p} u_j \frac{d_j^2}{d_j^2 + \lambda} u_j^{'} y$$

$$\hat{y}^{PCR} = U\theta^{PCR} = xb^{PCR} = \sum_{j=1}^{A} u_j u_j^{'} y$$

u_j is the jth principal score values

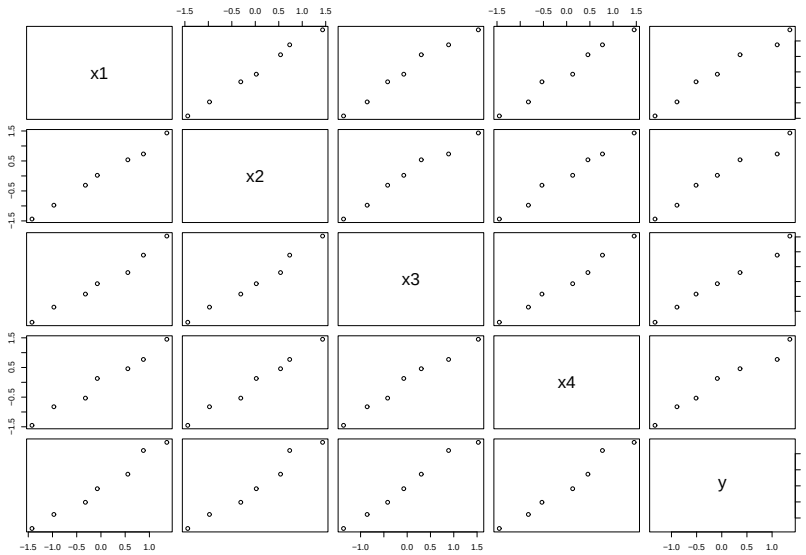
8.2.7 All methods

MLR(LS), PCR and RR/Lasso

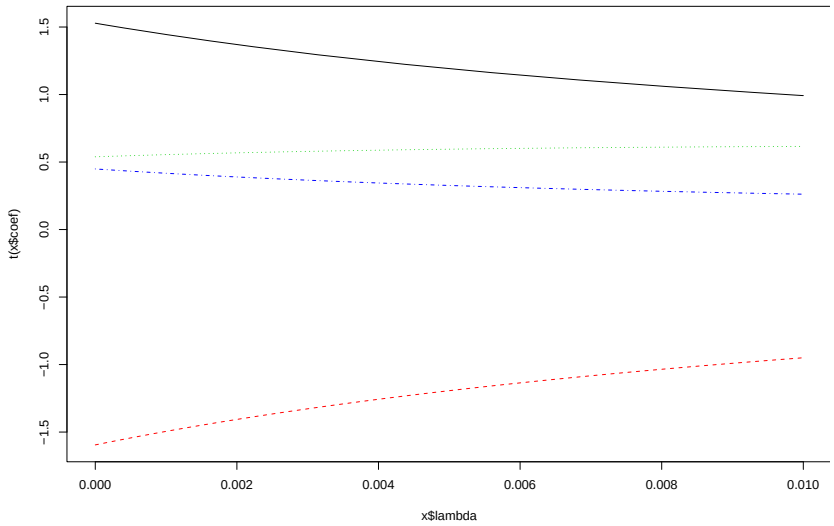
- The LS-solution is a vector in space.
- Ridge regression *shrinks* the LS-solution
- Highest shrinkage in low-variance directions
- PCR also shrinks: No shrinkage in main directions, 100% shrinkage in low variance directions.
- (PLS is ALSO a shrinkage method - allthough ”expansion”is also possible)
- Lasso also shrinks - some coefficients down to EAXCTLY zero
- Lasso is hence also an automated variable selection method
- Lasso is an L_1 method - VERY different from:
- The others are L_2 methods

8.2.8 Little Ridge example

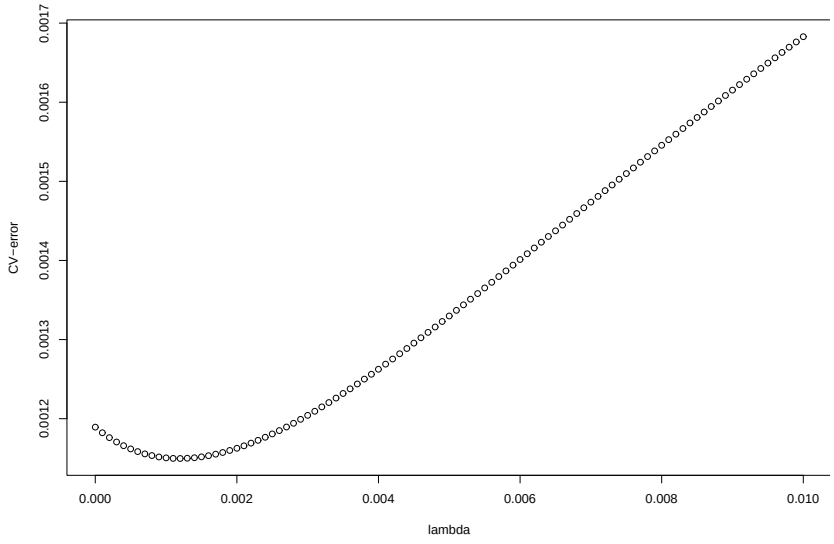
Example Data



Regression coefficients



Cross Validation



MLR and RR results:

	MLR Estimate	RR estimate ($\lambda = 0.0012$)
x1	1.651	1.543
x2	-1.723	-1.595
x3	0.5819	0.6023
x4	0.4848	0.4440

8.2.9 Ridge/Lasso - How to do it! (same as for PCR/PLS)

How to do it?(same as for PCR/PLS)

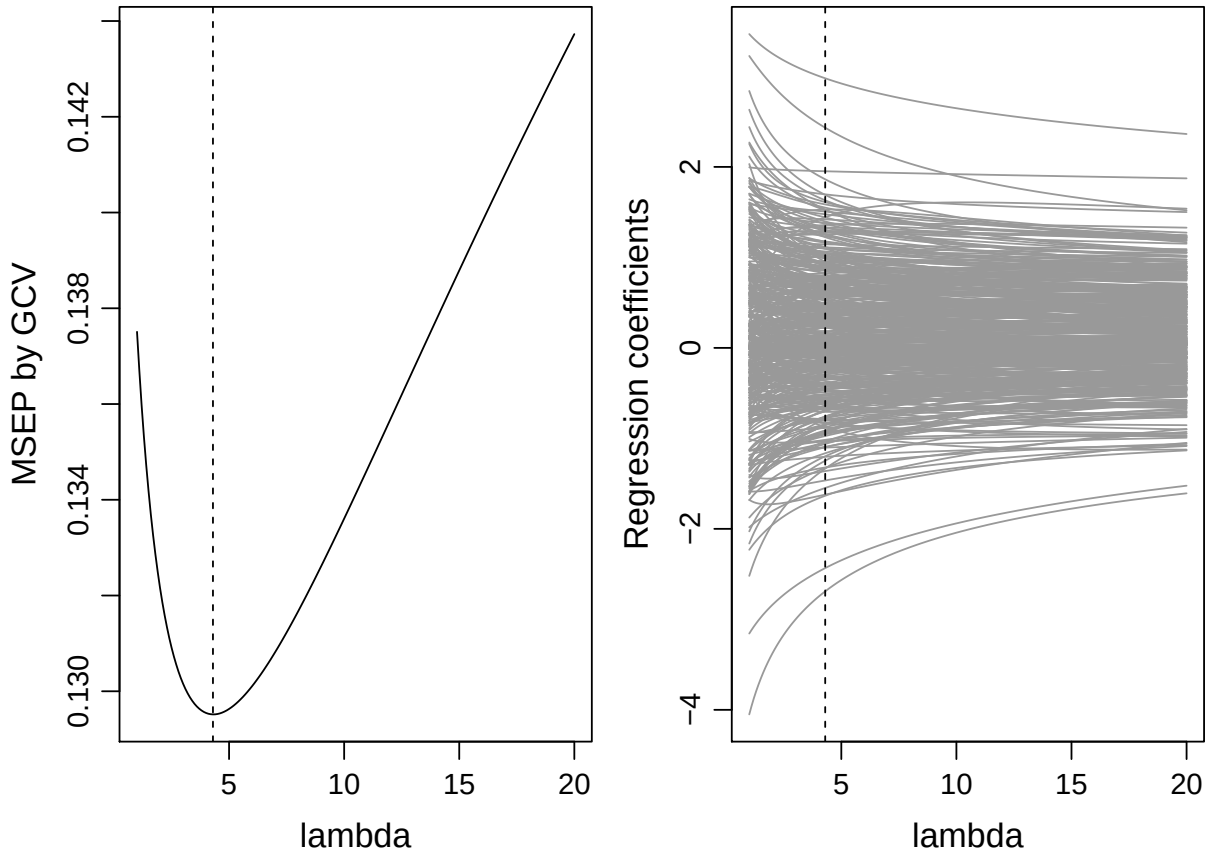
1. Explore data
2. Do modelling (choose number of components, consider variable selection)
3. Validate (residuals, outliers, influence etc) (not so straightforward)
4. Iterate e.g. on 2. and 3.
5. Interpret, conclude, report.
6. If relevant: predict future values.

8.3 Example: PAC data from package `chemometrics`

8.3.1 Ridge Regression

```
library(chemometrics, warn.conflicts = FALSE)
data(PAC)
#?PAC
```

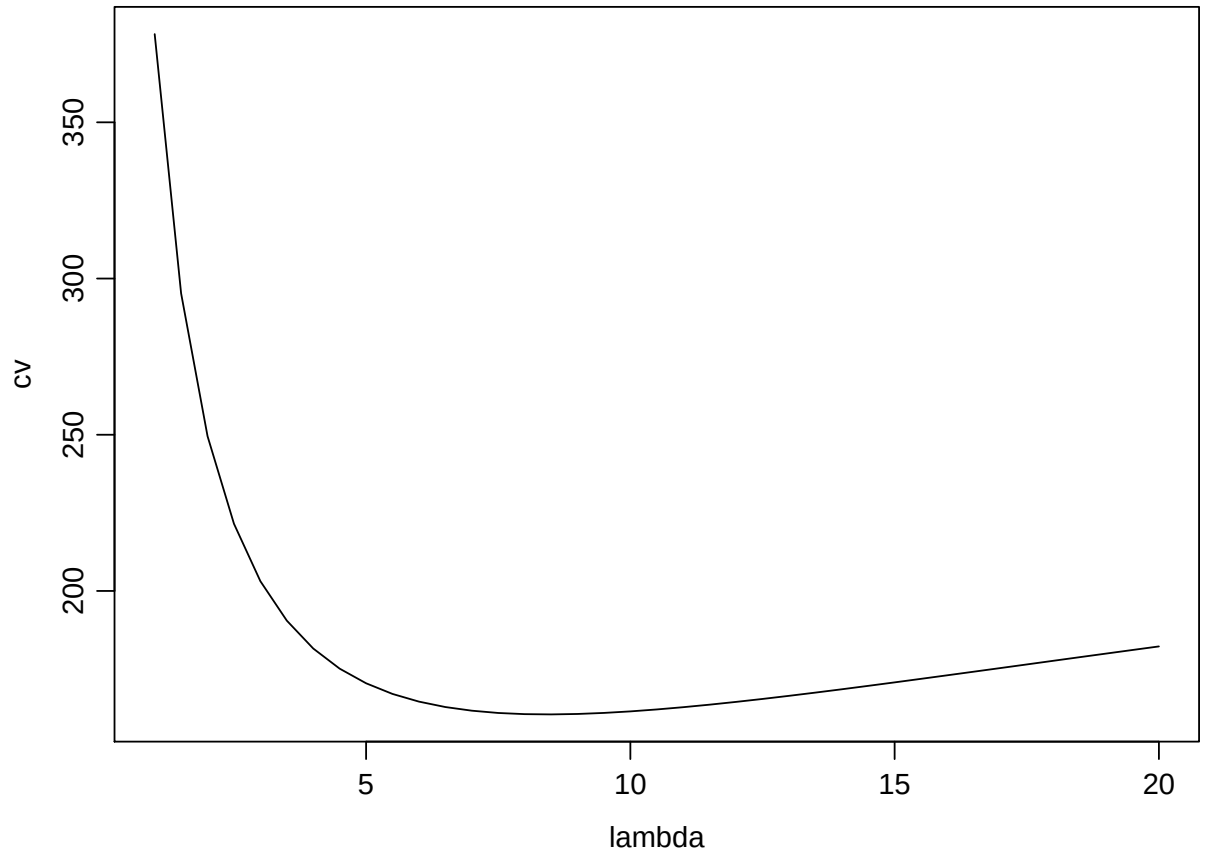
```
# First find optimal lambda:

# By GCV: (formula approximation of real CV), sec 4.3.2.2 in Varmuza-book
ridge_res <- plotRidge(y ~ X, data = PAC, lambda = seq(1, 20, by = 0.1))
```




```
ridge_res$lambdaopt
```

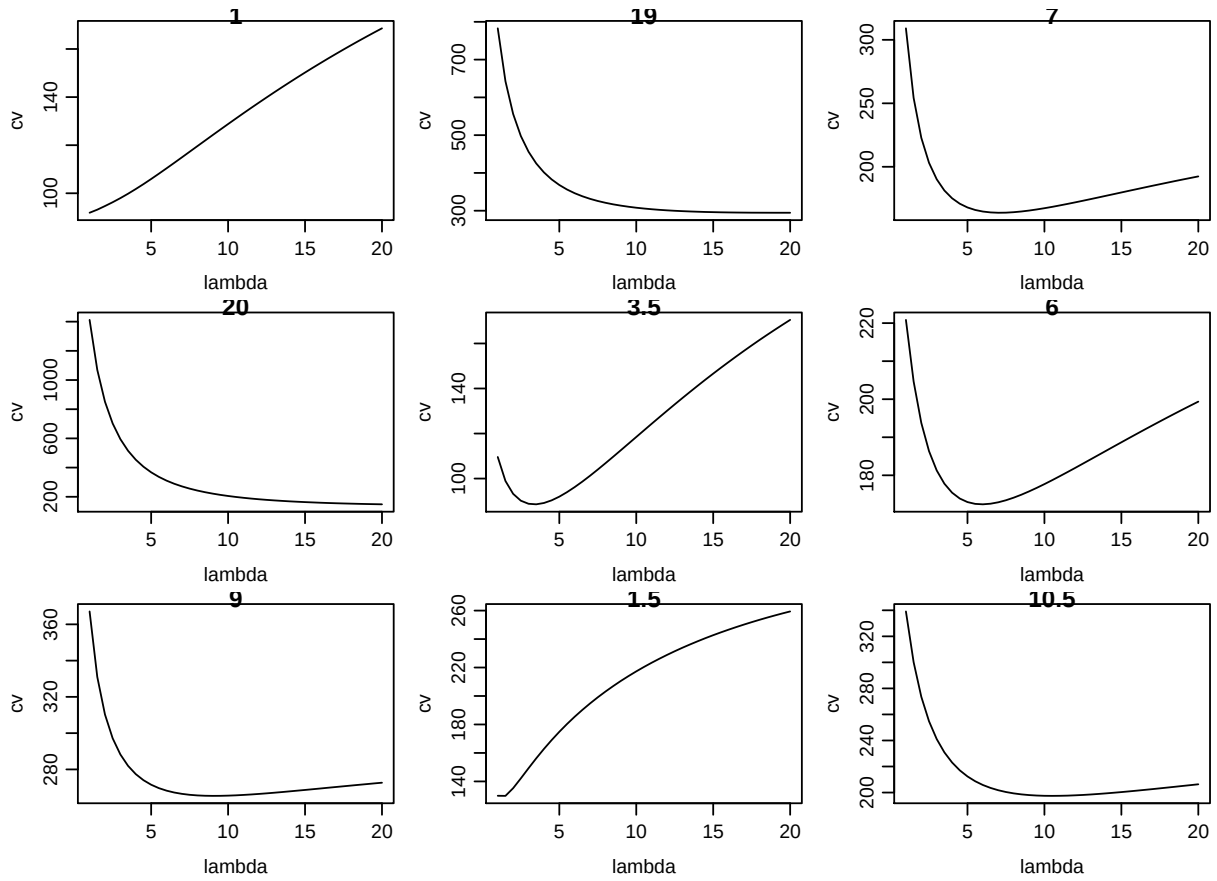
```
[1] 4.3
```

```
# By real CV, e.g. LOO:
# Note that the ridgeCV function shown in the book does NOT do this for us
# It only looks into an allready chosen optimal lambda
# So we use the ridge.CV function from the parcor package:

library(parcor, warn.conflicts = FALSE)
par(mfrow=c(1, 1))
# First Full LOO
res_LOO <- ridge.cv(scale(PAC$X), PAC$y, lambda = seq(1, 20, by = 0.5),
                    k = 209, plot.it = TRUE)
```



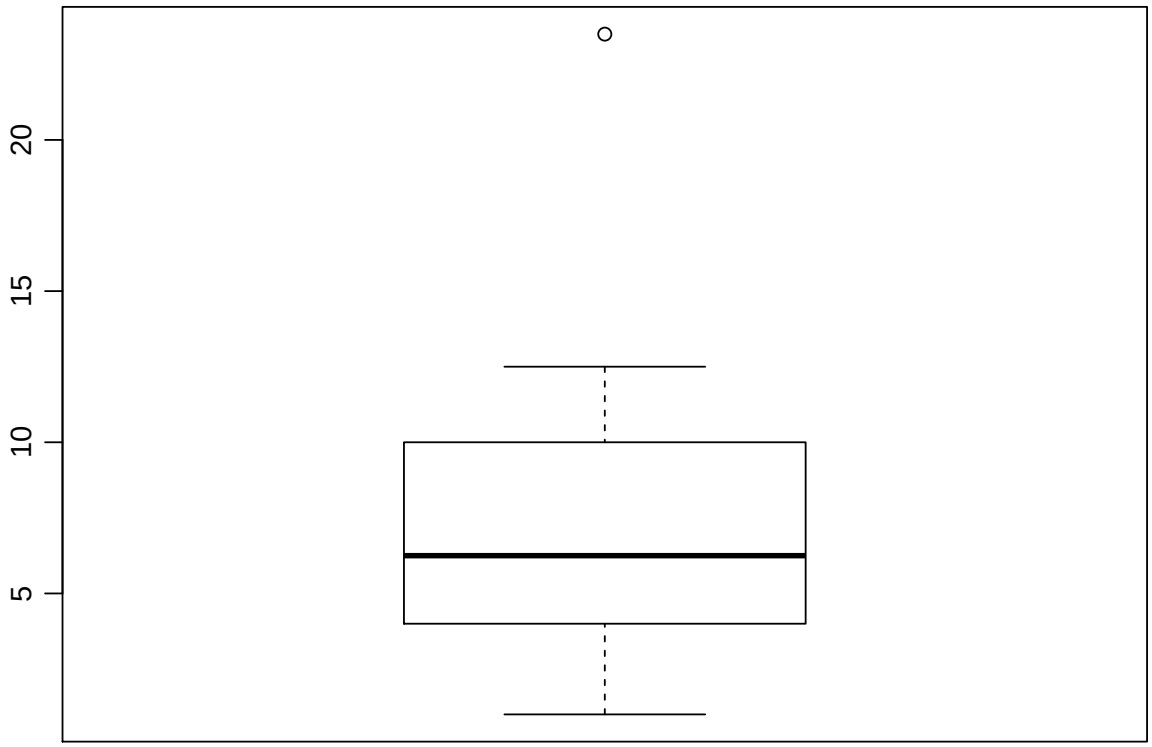

```
res_LOO$lambda.opt


[1] 8.5
```

```
# Then 9 (random) versions of 10-fold CV:
par(mfrow=c(3,3))
for ( i in 1:9){
res_CV10=ridge.cv(scale(PAC$X),PAC$y,lambda=seq(1,20,by=0.5),k=10,plot.it=TRUE)
title(main=res_CV10$lambda.opt)
}
```



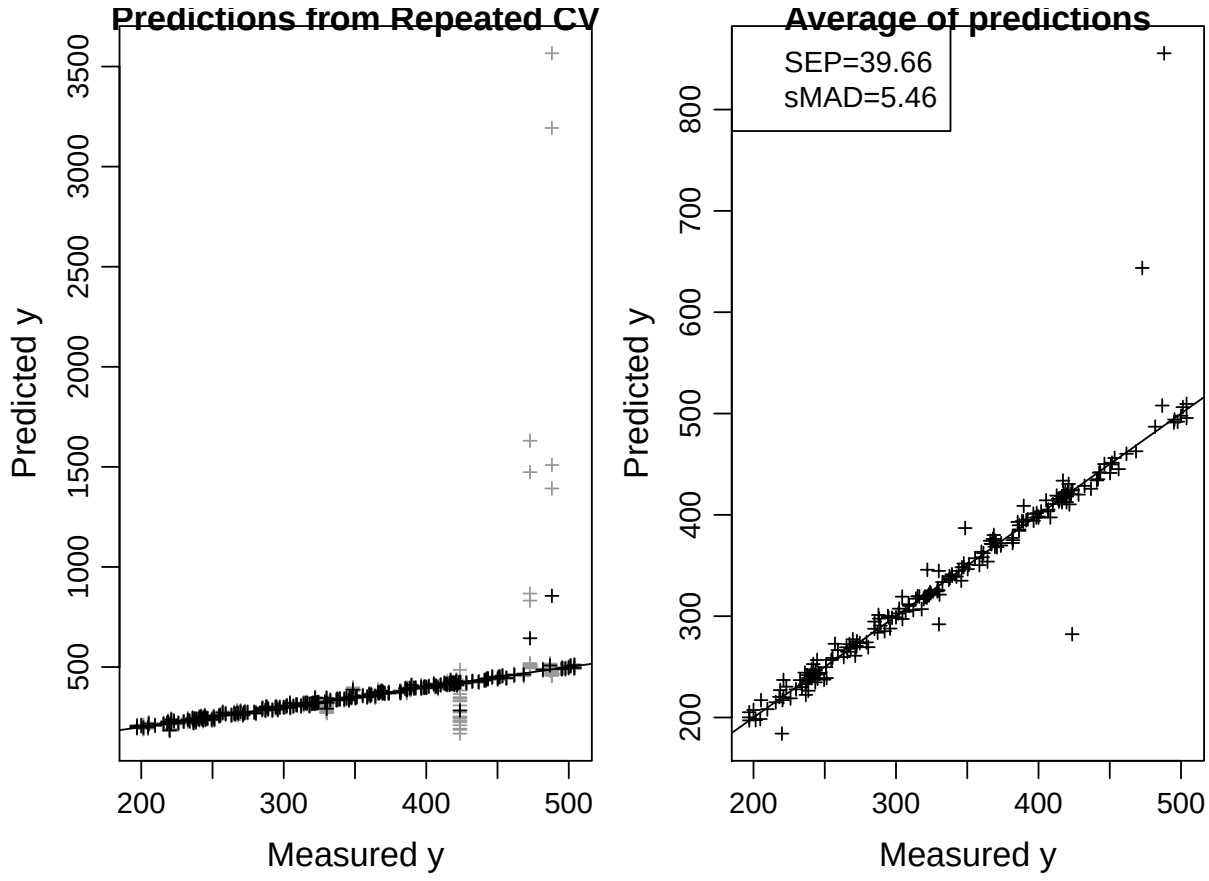

```
# Let's do our own "repeated double 10-fold CV":
K=10
lambda.opts <- rep(0, K)
for ( i in 1:K ){
  res_CV10 <- ridge.cv(scale(PAC$X), PAC$y,
              lambda = seq(1, 30, by = 0.5),
              k = 10, plot.it = FALSE)
  lambda.opts[i] <- res_CV10$lambda.opt
}

median(lambda.opts)


[1] 6.25
```

```
par(mfrow = c(1, 1))
boxplot(lambda.opts)
```

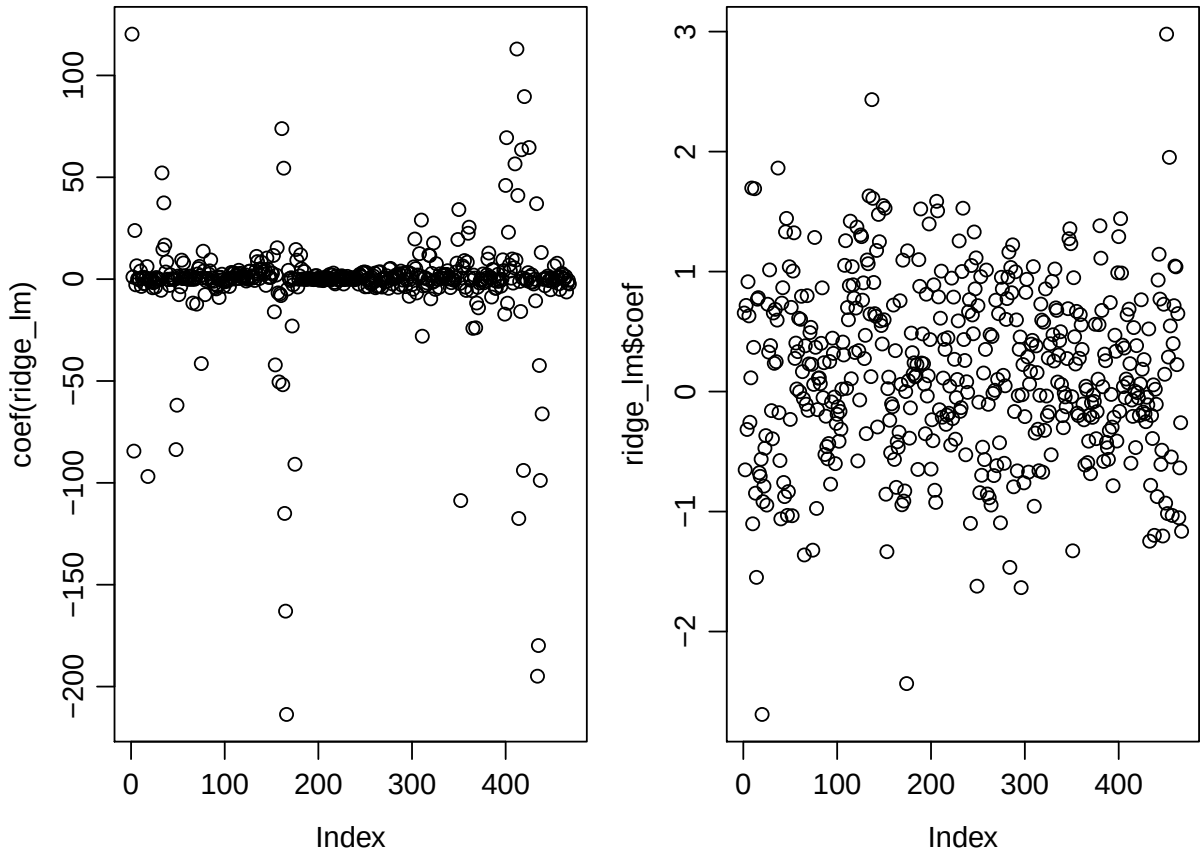



```
# Now let's try the ridgeCV to investigate the choice(s) more carefully:
# As suggested by the book
res <- ridgeCV(y ~ X, data = PAC, lambdaopt = 4.3,
               repl = 20)


[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
[1] 11
[1] 12
```

```
[1] 13
[1] 14
[1] 15
[1] 16
[1] 17
[1] 18
[1] 19
[1] 20
```

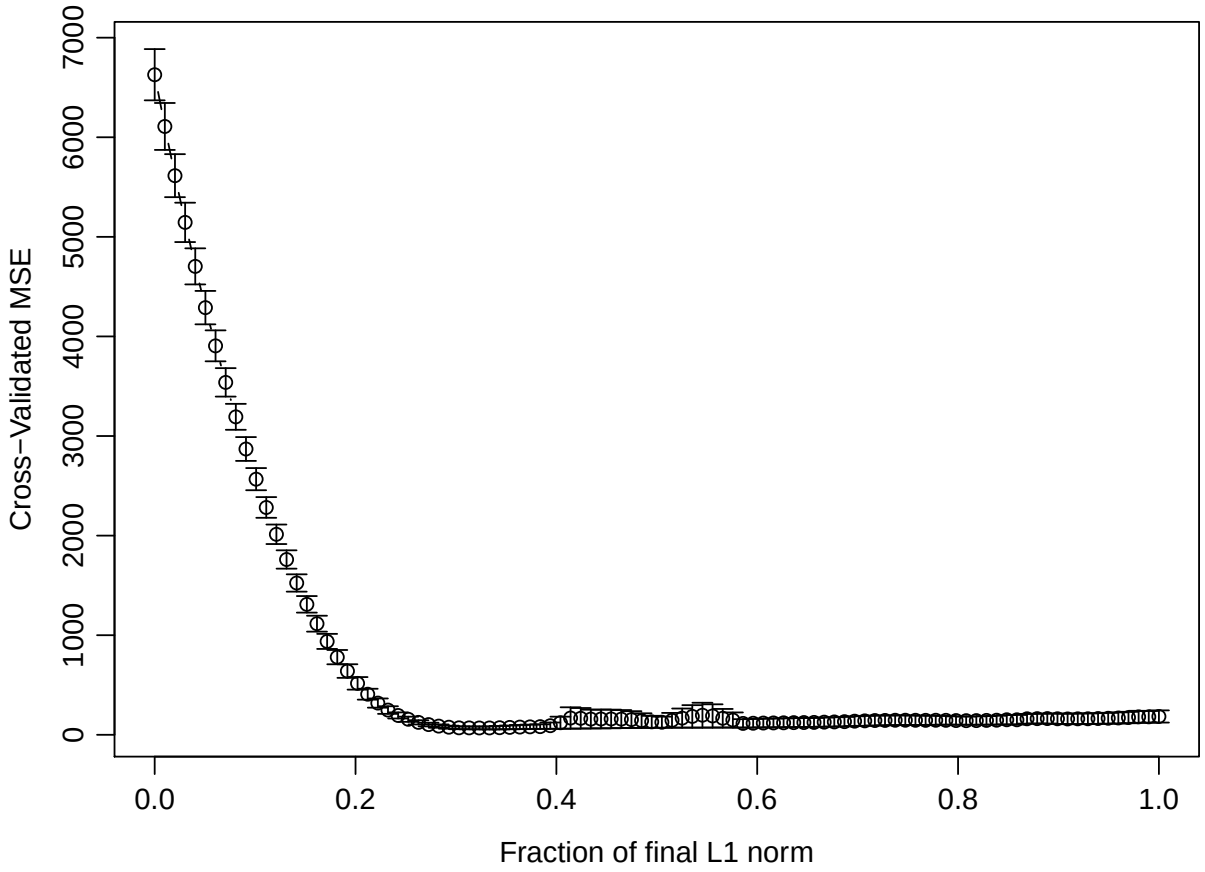



```
# Look at the coefficients:
ridge_lm <- lm.ridge(y ~ X, data = PAC, lambda = 4.3)

# Coefficients on raw X-scale:
plot(coef(ridge_lm))

# Coefficients on standardized  X-scale:
plot(ridge_lm$coef)
```

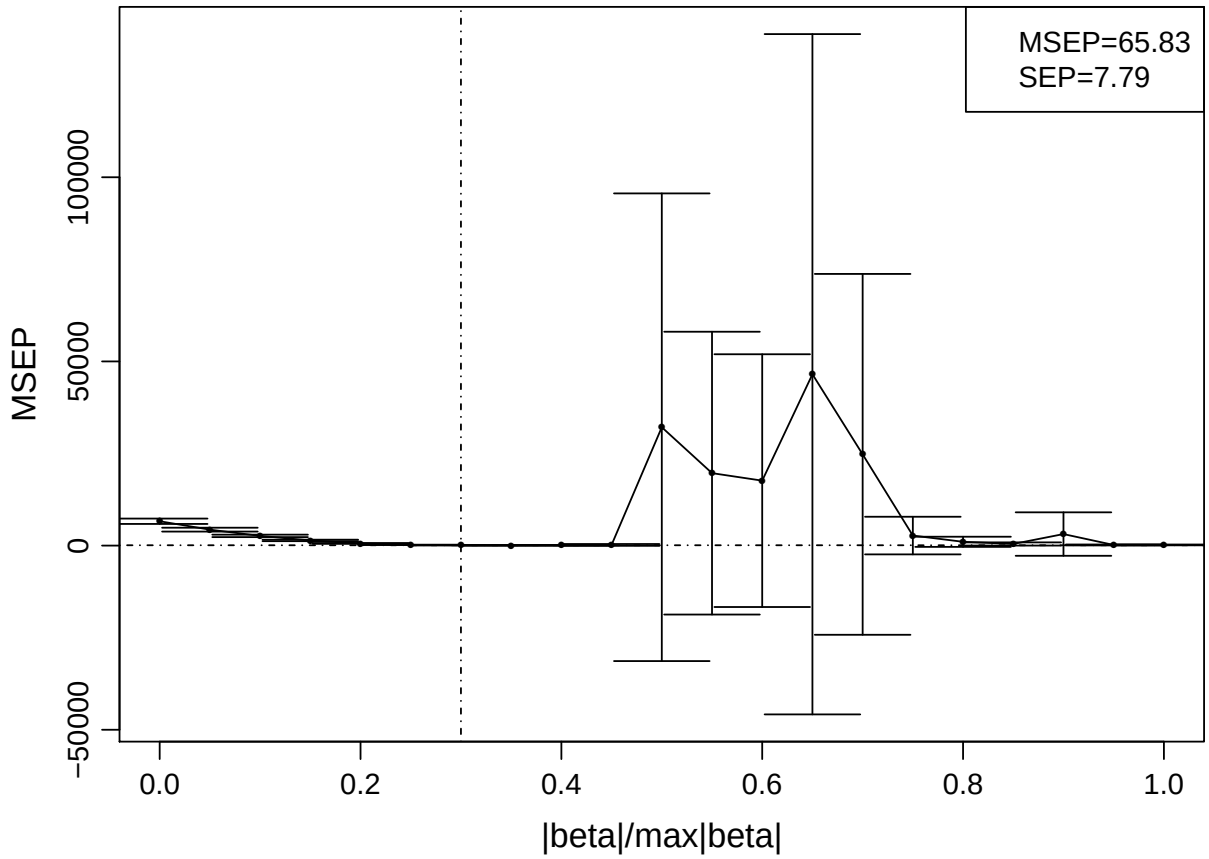


8.3.2 Lasso Regression

```
# Now let's try to fit the lasso:
library(lars)
# We could use the lars package:
lasso_result <- lars(PAC$X, PAC$y)
cv.lars(PAC$X, PAC$y)
```



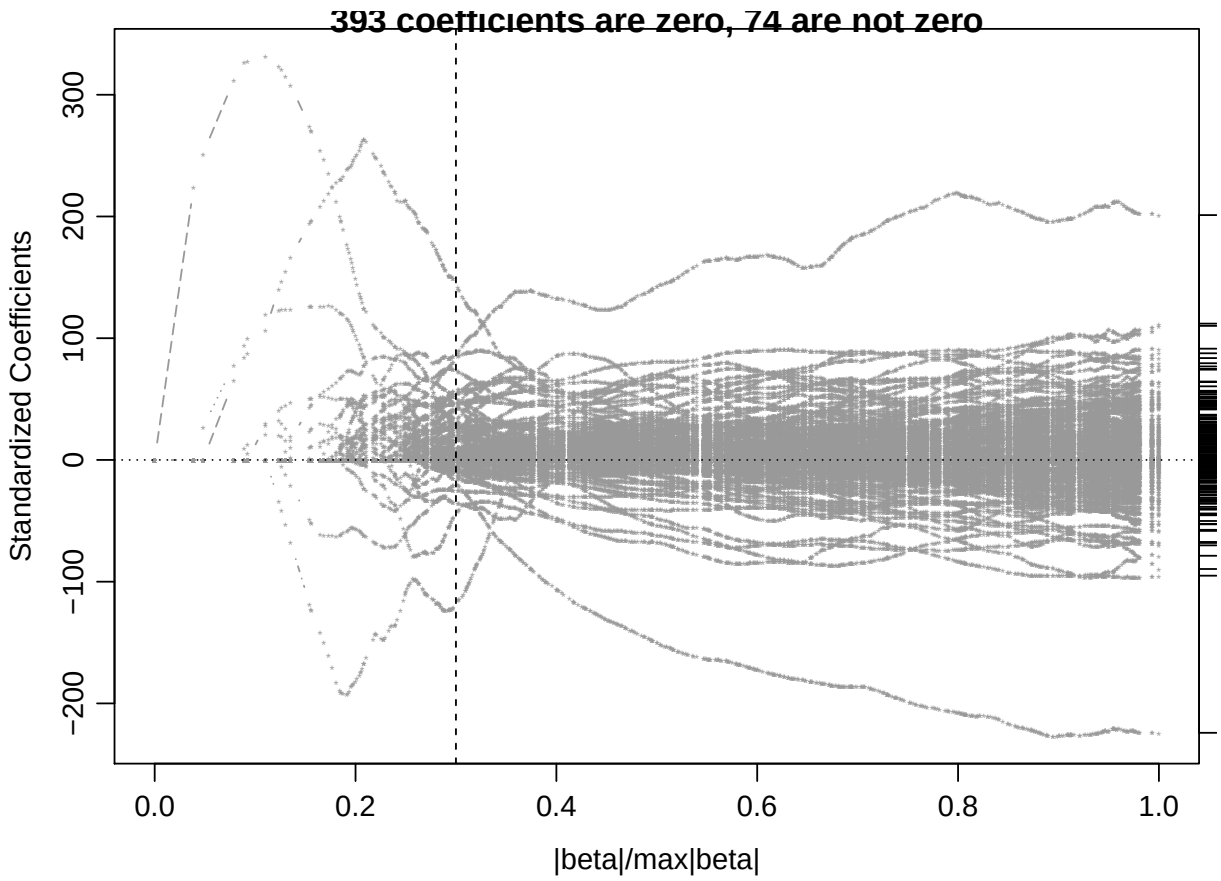

```
# Or from the chemometrics package:
res_lasso <- lassoCV(y ~ X, data = PAC,
                     fraction = seq(0, 1, by = 0.05), K = 10)
```



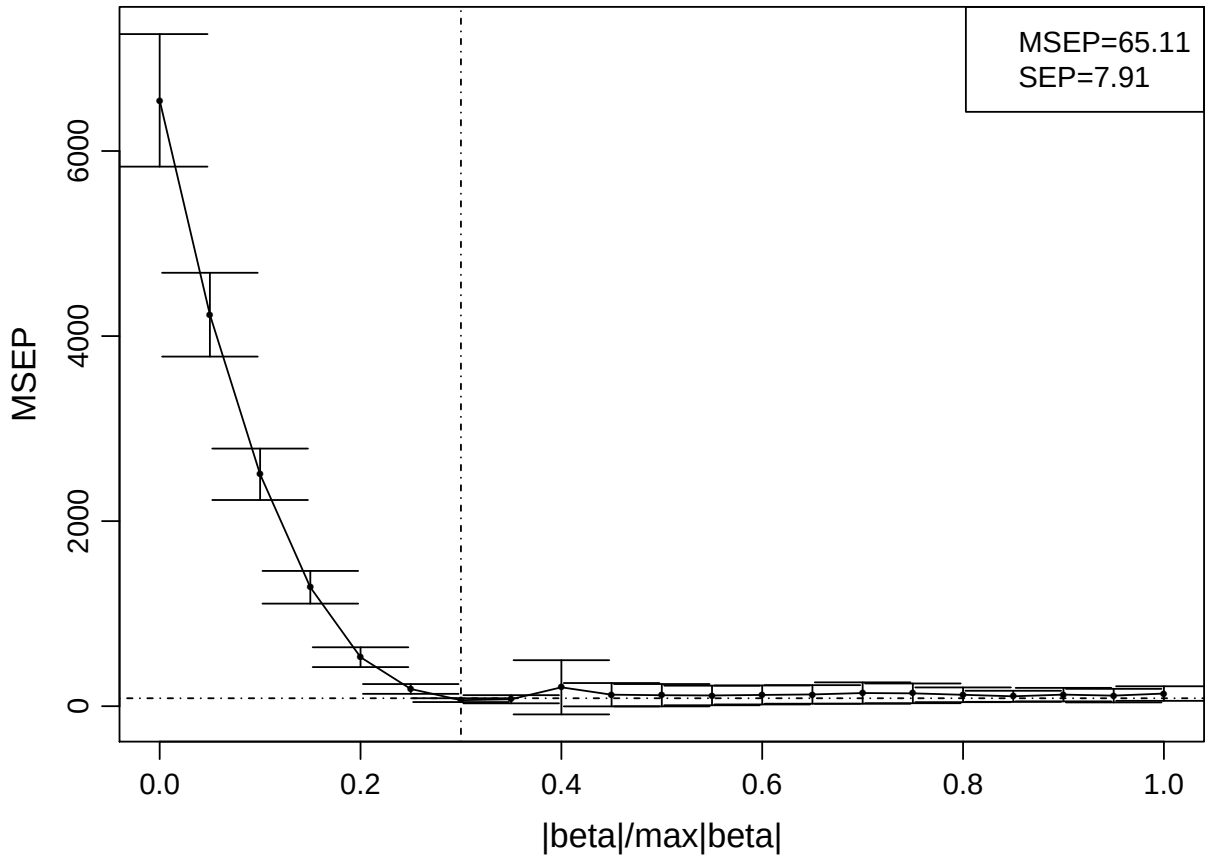

```
res_lasso$sopt
```

```
[1] 0.3
```

```
res_coef <- lassocoef(y ~ X, data = PAC, sopt = res_lasso$sopt)
```



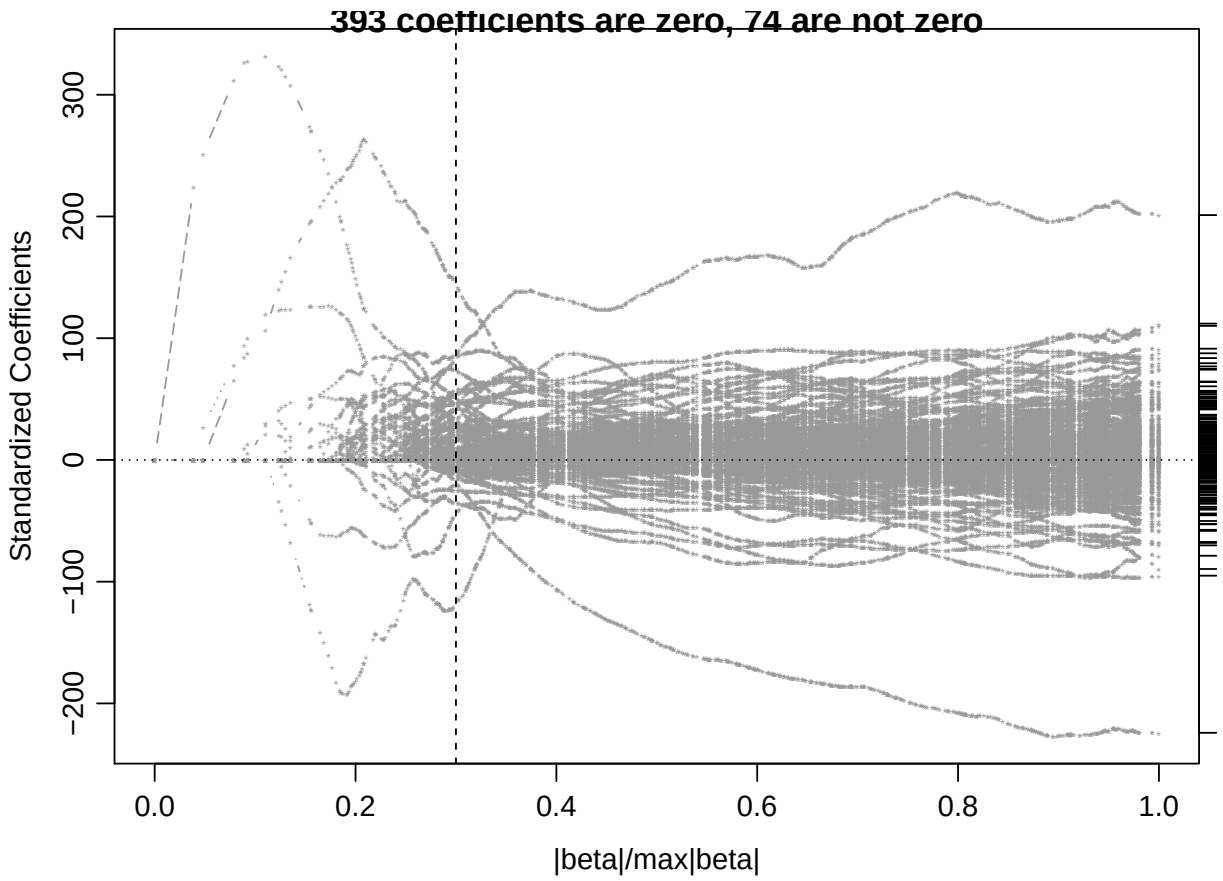

```r
# In an auto-scaled version:
res_lasso_scaled <- lassoCV(y ~ scale(X), data = PAC,
                            fraction = seq(0, 1, by = 0.05), K = 10)
```



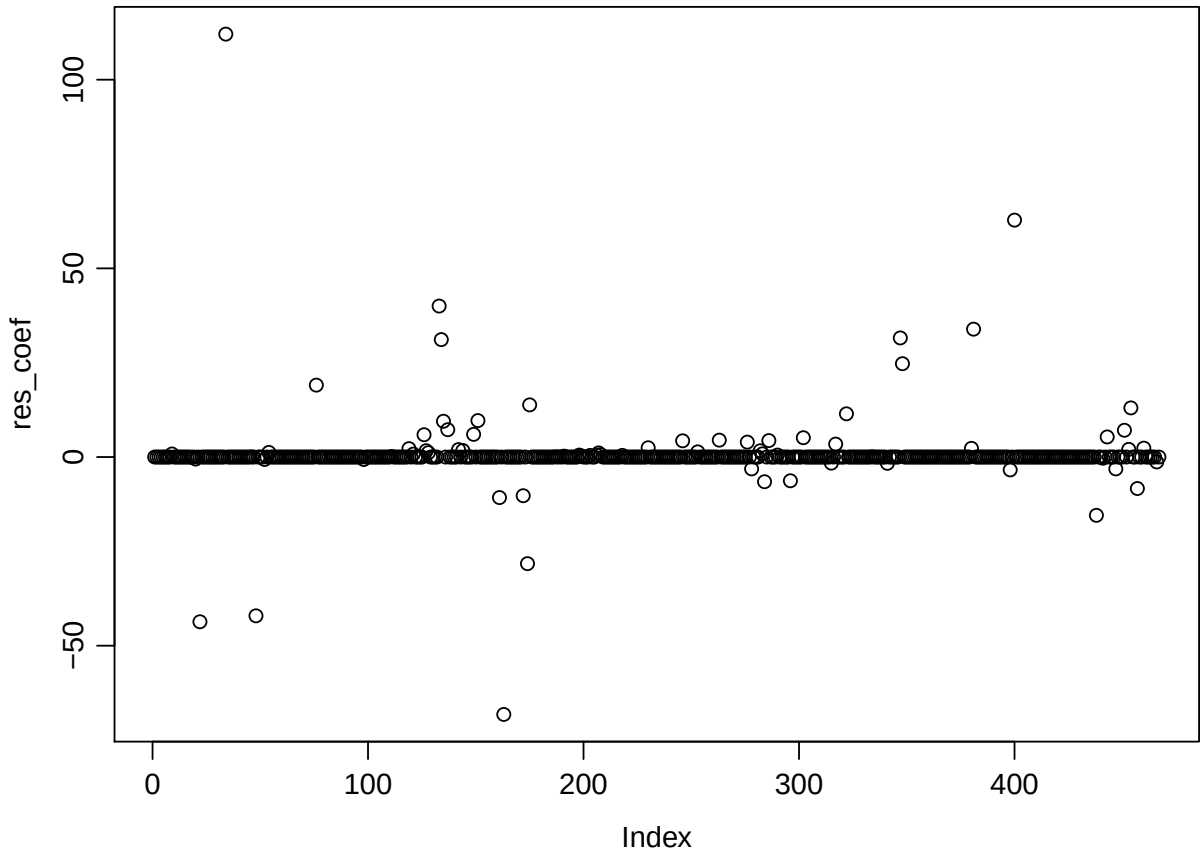

```
res_lasso_scaled$sopt
```

```
[1] 0.3
```

```
res_coef_scaled <- lassocoef(y ~ scale(X), data = PAC,
                             sopt = res_lasso_scaled$sopt)
```




```
# The actual coefficients can be looked at by: (on raw X-scale - I believe.:-)...)
res_coef <- res_coef$coefficients
plot(res_coef)
```




```
beta_lasso <- res_coef
```

8.3.3 Prediction by ridge and lasso

```
# Let's try to predict a NEW data set
# here we take  the original data - just for illustration:
PAC_test=PAC

# First by ridge regression:
ridge_lm <- lm.ridge(y ~ X, data = PAC, lambda = 4.3)

# Coefficients on raw X-scale: (WITH intercept)
beta_ridge <- coef(ridge_lm)

yhat_ridge <- beta_ridge[1] + PAC_test$X %*% beta_ridge[-1]
```

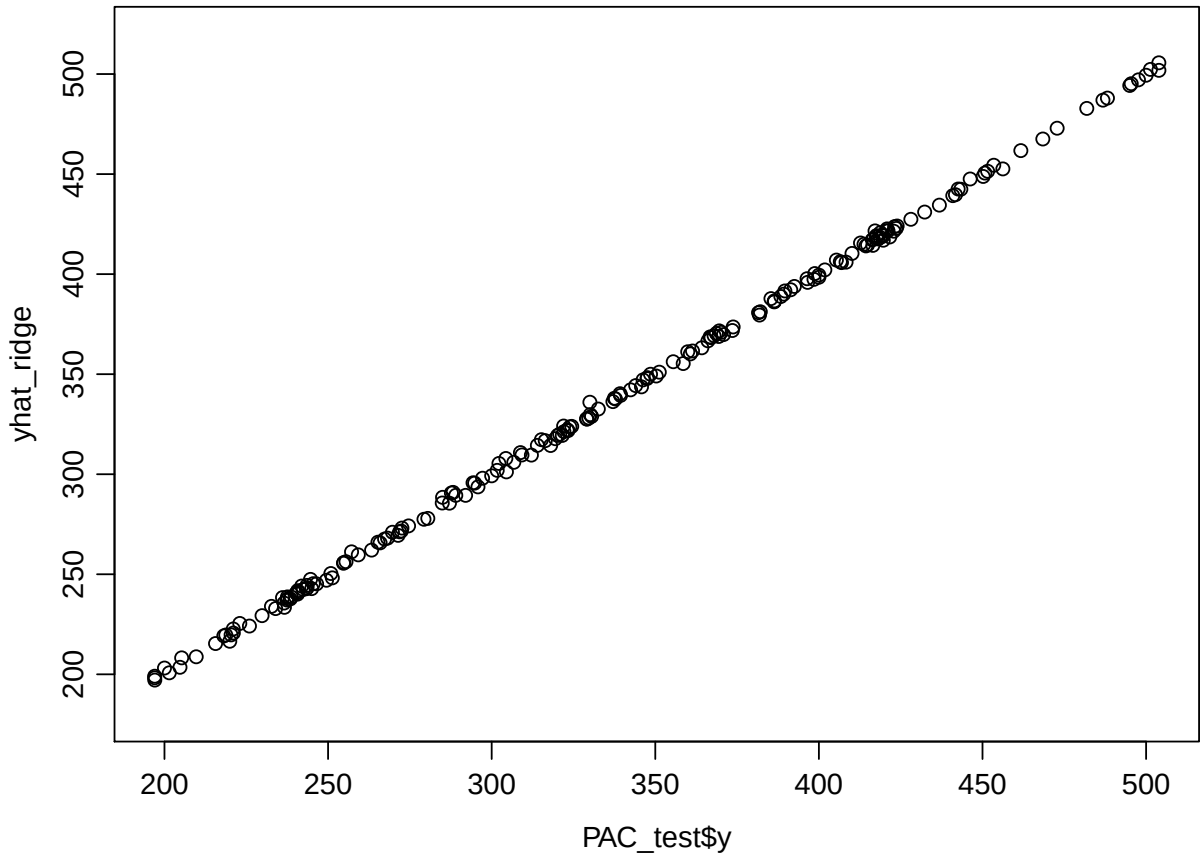
```
residuals <- PAC$y - yhat_ridge
MSEP <- mean(residuals^2)
MSEP
```

```
[1] 2.4868
```

```
sqrt(MSEP)
```

```
[1] 1.577
```

```
plot(PAC_test$y, yhat_ridge, ylim = c(180, 520))
```




```
# Then by lasso regression:
```

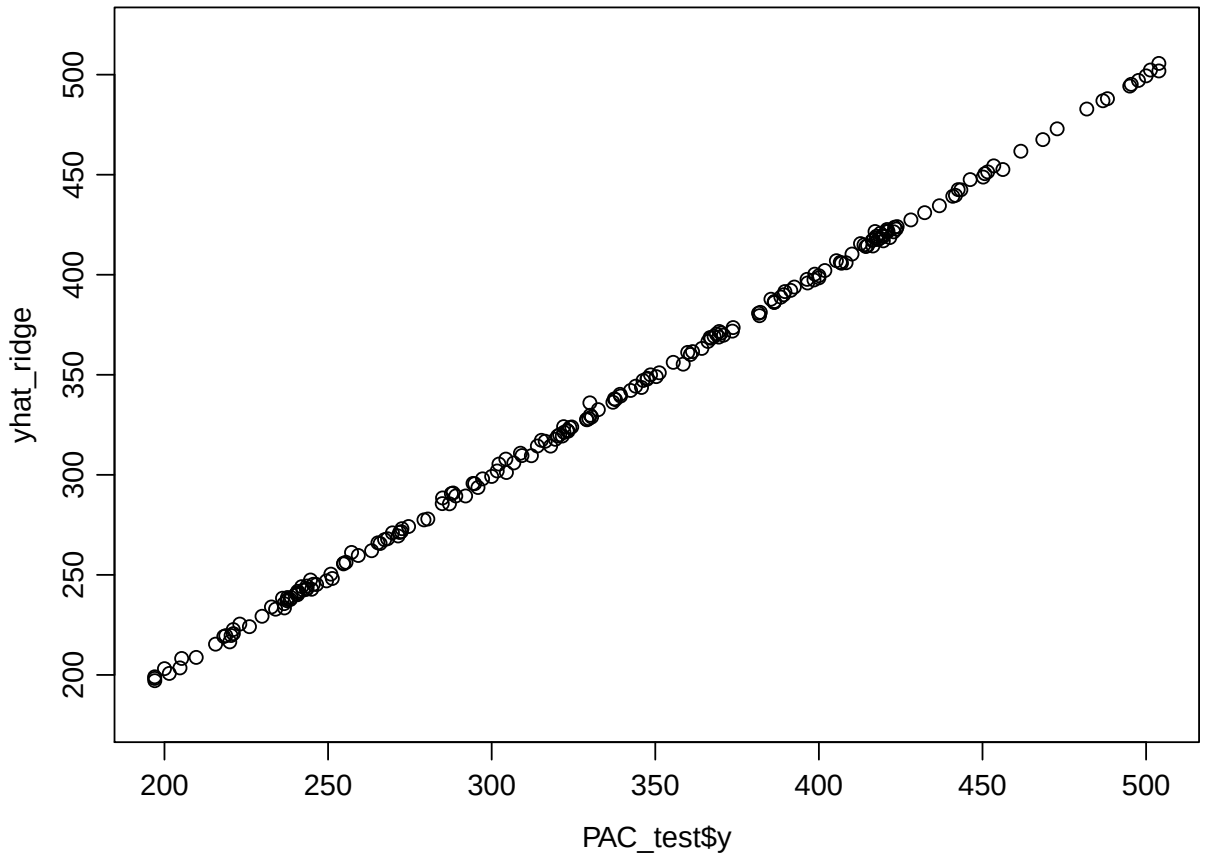
```
yhat_lasso <- predict(lasso_result, newx = PAC_test$X, s = 0.3,
                      mode = "fraction")$fit
residuals <- PAC_test$y - yhat_lasso
MSEP <- mean(residuals^2)
MSEP
```

```
[1] 15.767
```

```
sqrt(MSEP)
```

```
[1] 3.9708
```

```
plot(PAC_test$y, yhat_ridge, ylim = c(180, 520))
```



8.4 Exercises

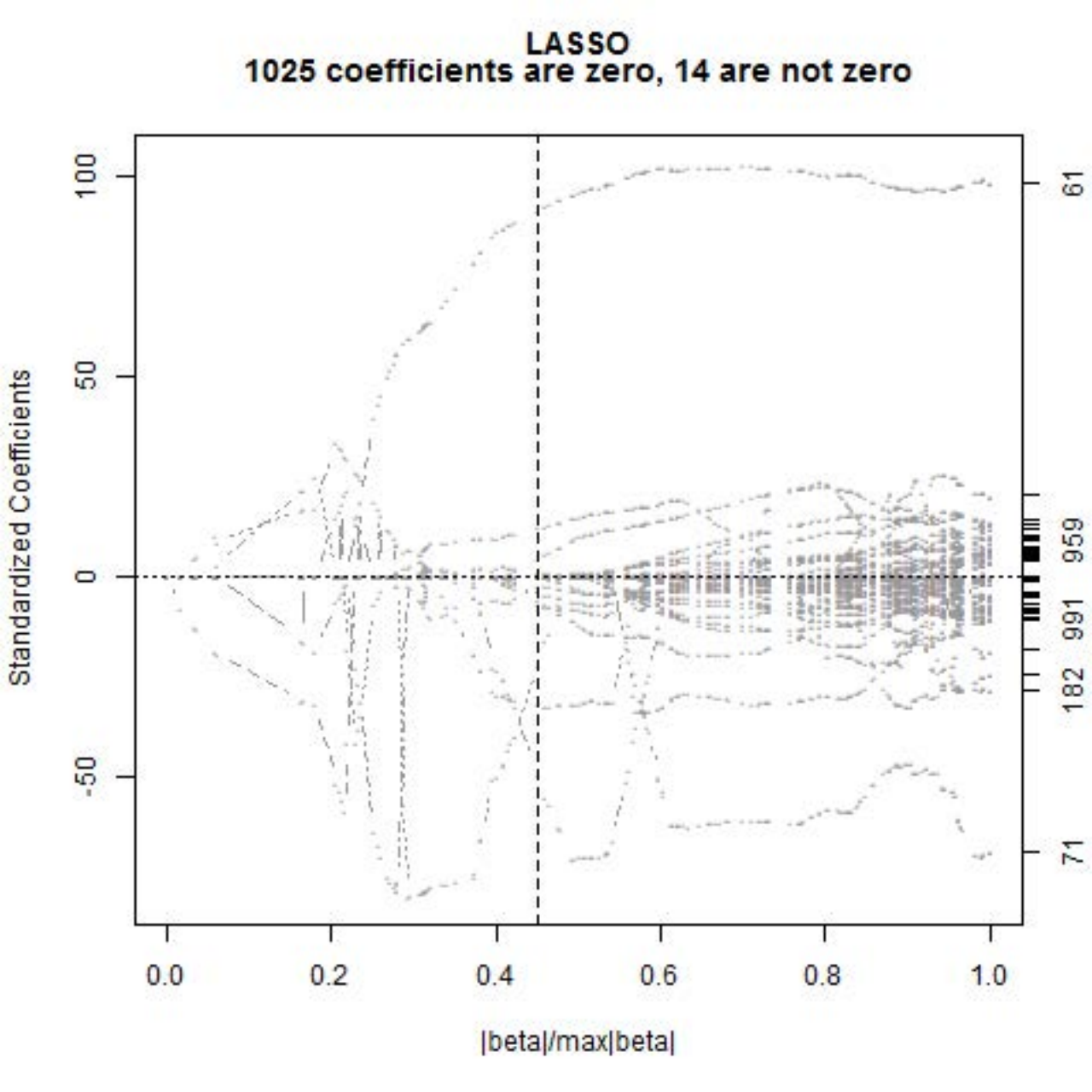
|||| Exercise 1 Exercise: Prostate Cancer data and maybe pectin data

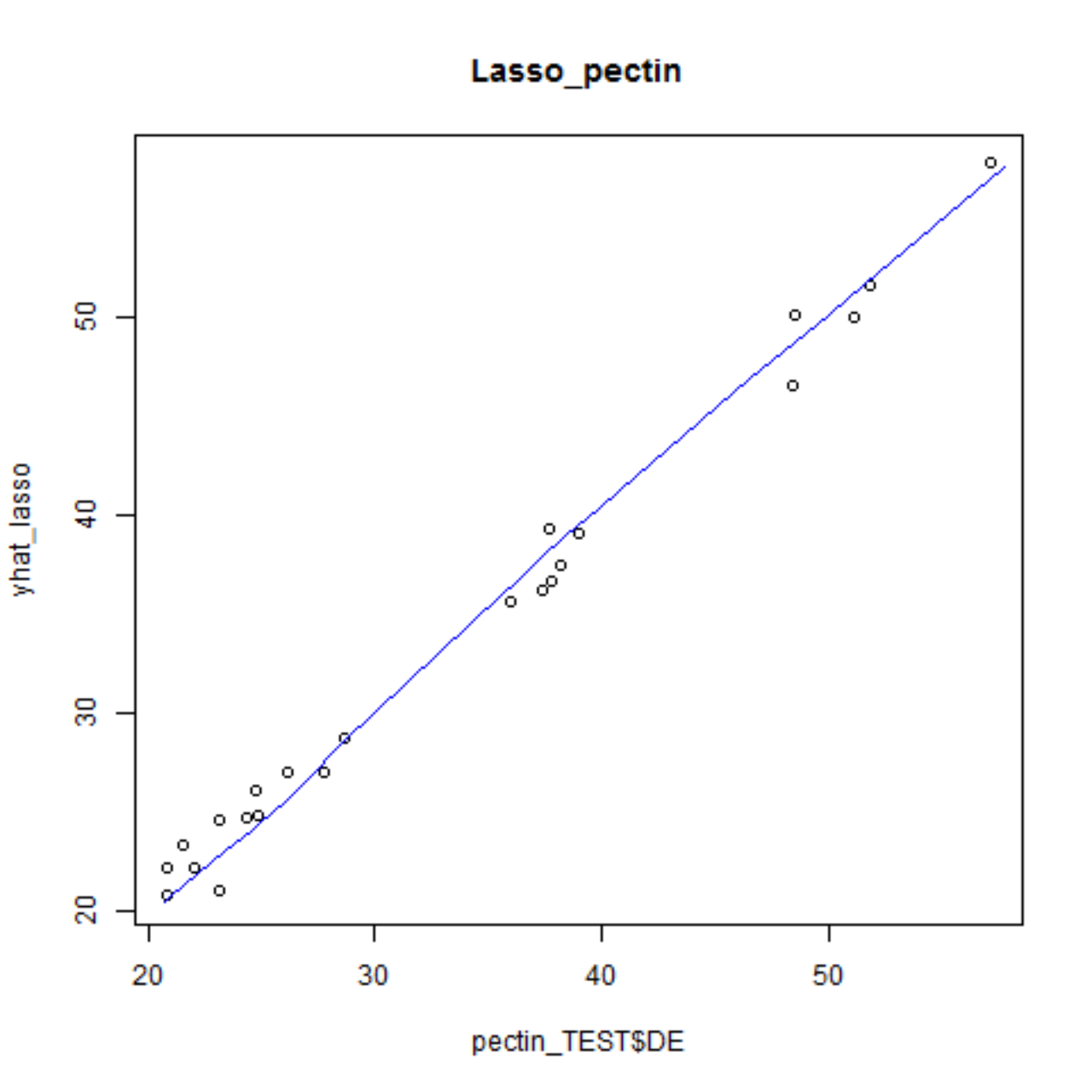
a) Try both ridge and lasso regression on the Prostate cancer data - a scaled version.

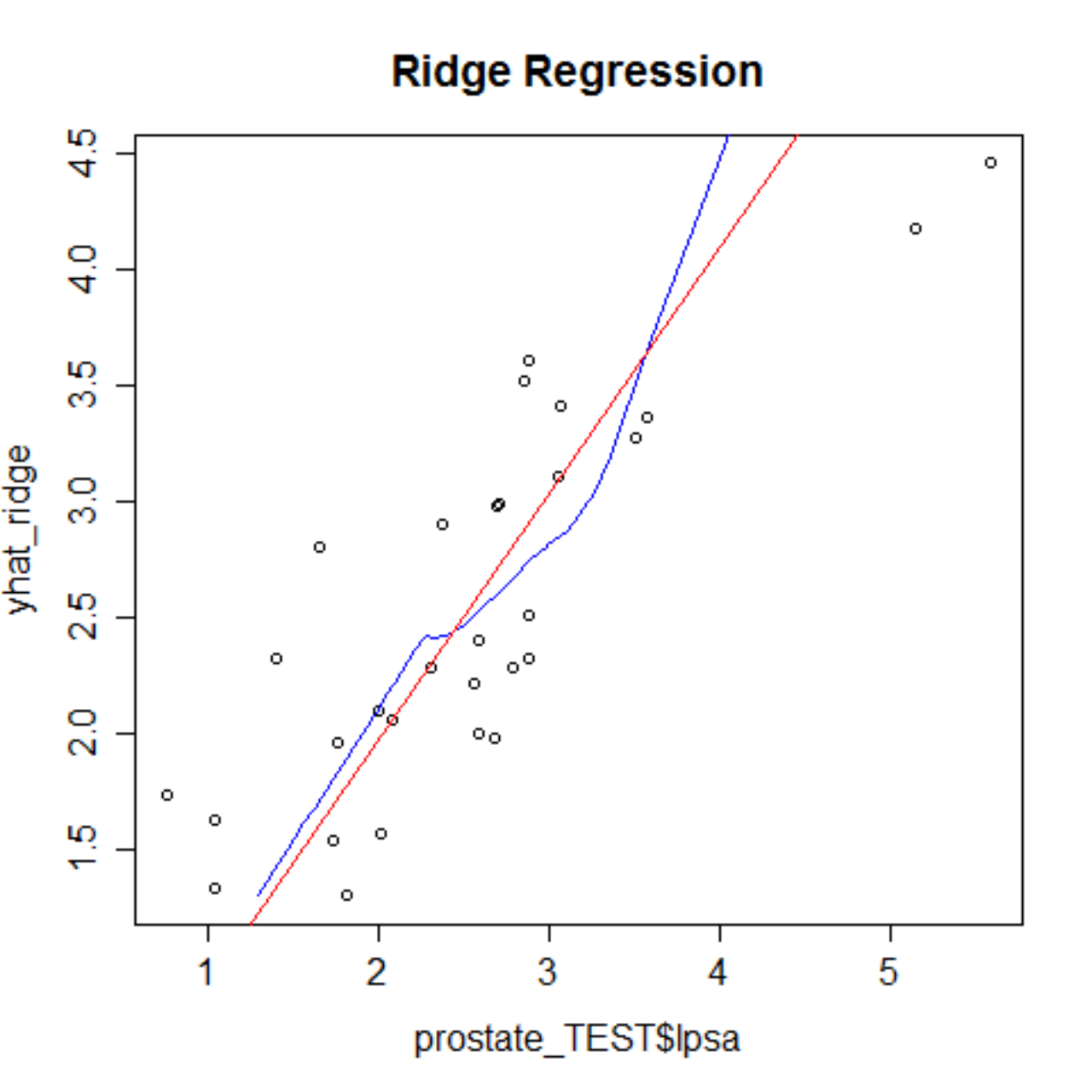
1. Work on the training data
2. Find the optimal penalty
3. Will lasso completely leave out some of the variables?
4. Interpret/Look at the coefficients
5. Predict the test data - compare results with other methods, e.g. MLR/PCR/(PLS)
6. For the ridge-model: Compare the ”size”of variable coefficients with coefficients from MLR and PCR - what can you conclude?

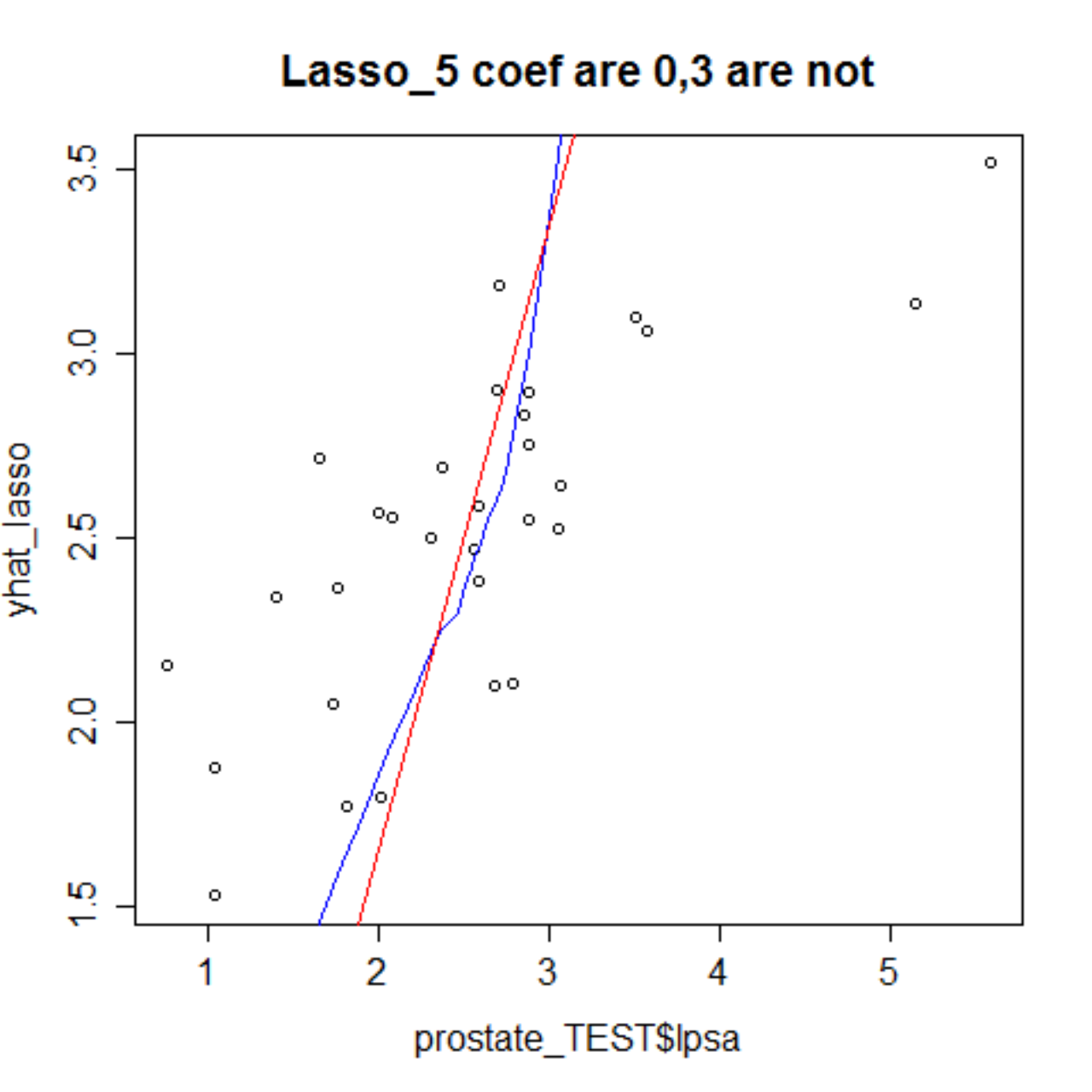
b) Try lasso on the pectin NIR data - how many wavelengths will be completely left out?

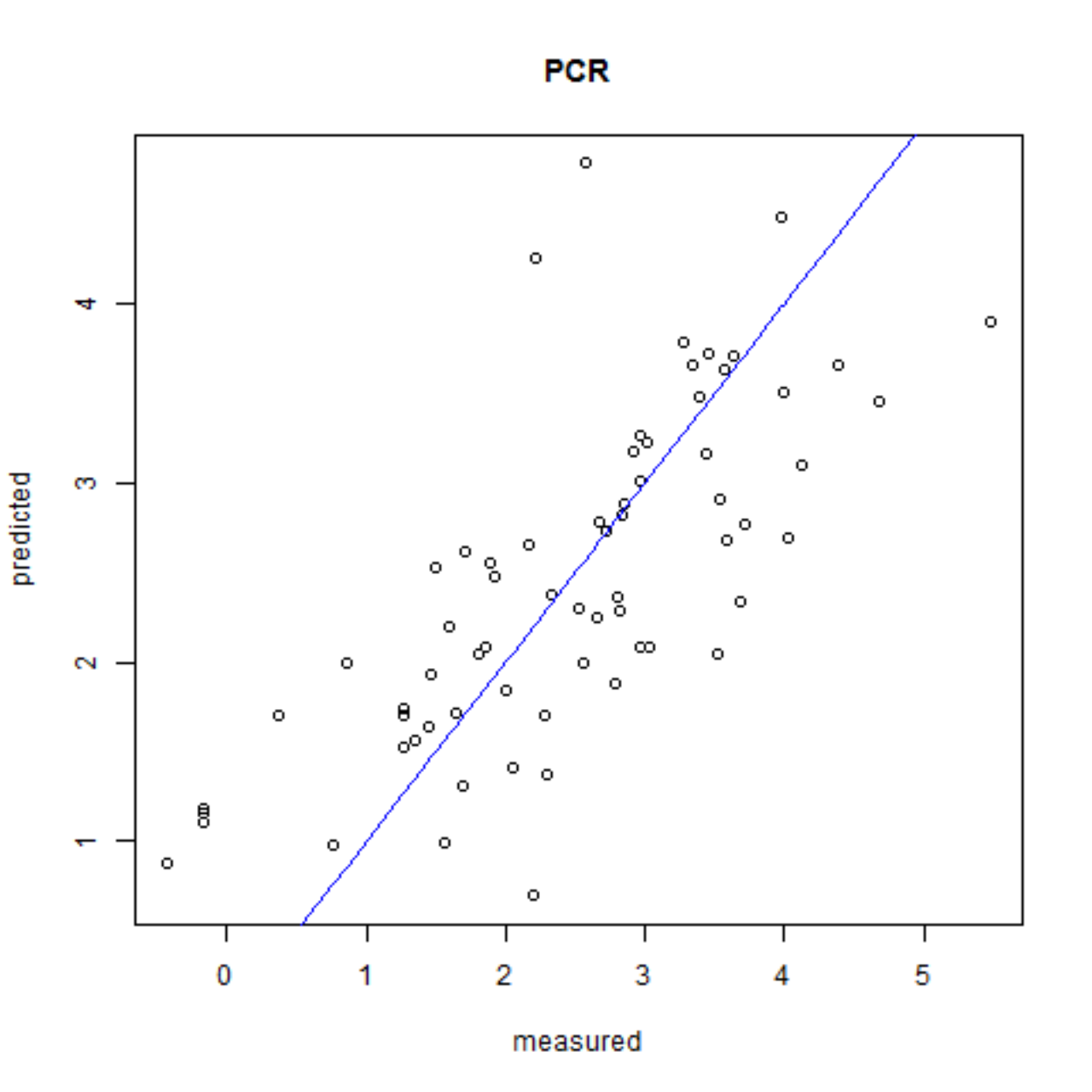
```
PCR gives more linear fitting than MLR,Lasso,RR.
```

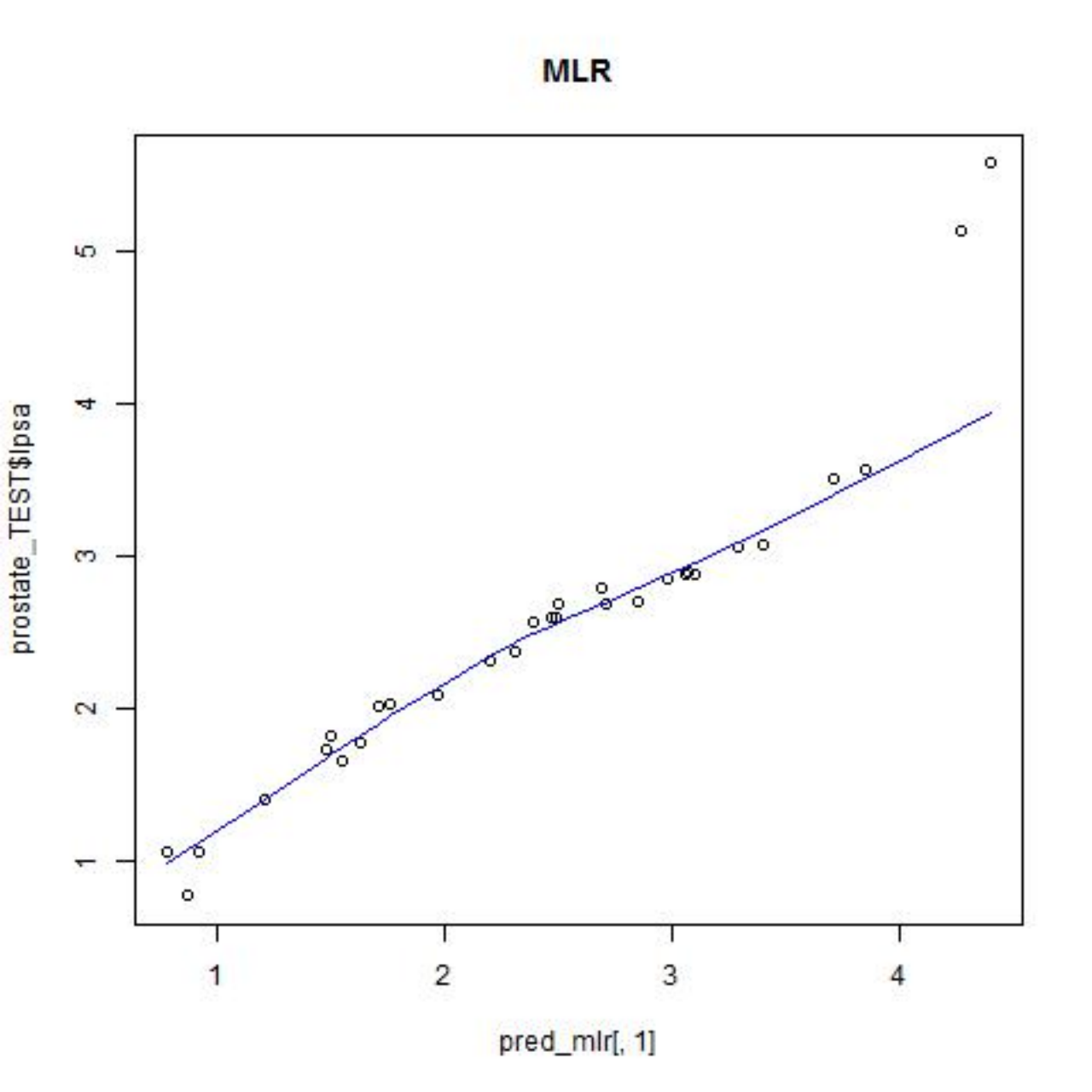

LASSO
1025 coefficients are zero, 14 are not zero
Standardized Coefficients
100
50
0
-50
0.0
0.2
0.4
0.6
0.8
1.0
|beta|/max|beta|
61
959
991
182
71


Lasso_pectin
yhat_lasso
pectin_TEST$DE
20
30
40
50
20
30
40
50


Ridge Regression
yhat_ridge
prostate_TEST$lpsa
1.5
2.0
2.5
3.0
3.5
4.0
4.5
1
2
3
4
5


Lasso_5 coef are 0,3 are not
yhat_lasso
1.5
2.0
2.5
3.0
3.5
1
2
3
4
5
prostate_TEST$lpsa


PCR
predicted
measured
0
1
2
3
4
5
1
2
3
4


MLR
prostate_TEST$lpsa
pred_mlr[, 1]
1
2
3
4
5
1
2
3
4