ADEC 7430 Individual Project #1 Solution:

Linear Regression, Variable Selection, Ridge Regression and Lasso

Dr. Nathaniel D. Bastian
Woods College of Advancing Studies, Boston College

1 Introduction

In this analysis, we use the diabetes data from Efron et al.[1] to examine the effects of ten baseline predictor variables [age, sex, body mass index (bmi), average blood pressure (map), and six blood serum measurements (tc, ldl, hdl, tch, ltg, glu)] on a quantitative measure of disease progression one year after baseline. We employ several statistical learning techniques using the diabetes data to fit linear regression, ridge regression, and lasso models. We also incorporate best subset selection and cross-validation techniques.

The data set, which comes from the R package "lars", consists of 442 diabetes patients. Originally, this diabetes data was used by Efron et al.[2] in their development of the least angle regression (LARS) method for model selection - a useful and less greedy version of the classical forward selection methods.

2 Analysis and Results

Upon receipt of the data, we partitioned the 442 diabetes patients into a training set (75%) used to fit our models and into a test set (25%) used for prediction. We also conducted exploratory data analysis to look for correlations between predictors (using scatterplot matrices) and to confirm our assumptions of normality (using histograms). Following this, we fit several models to the training set, extracted the estimated model coefficients, predicted the responses for the test set, and calculated the mean prediction error and its standard error in the test set.

2.1 Least Squares Regression Model

For the first model, we used ordinary least squares regression to fit a model using **all** ten predictor variables. Only four out of the ten predictors (not including the intercept) were statistically significant (p-value < 0.05). The residual standard error (RSE) of this model was 54.05 (321 degrees of freedom) with an R^2 value of 0.5213. **Table 1** shows the linear regression model coefficient estimates from the training set (those with * are statistically significant).

Table 1: Linear Regression Model Coefficient Estimates

(Intercept)*	age	sex*	bmi*	map*	tc	ldl	hdl	tch	ltg	glu
149.92	-66.76	-304.65	518.66	388.11	-815.27	387.60	162.90	323.83	673.62	94.21

After fitting the linear regression model, we also obtained a 95% confidence interval (CI) for the model coefficient estimates and plotted the model diagnostics (see **Figure 1**) which confirmed linearity (i.e. there were not any non-linear patterns). Additionally, we calculated the variance inflation factors (VIFs) to check for collinearity between predictor variables, and we found that the following predictors had VIF values above 10: tc, ldl, hdl and ltg. To resolve the collinearity issue, we could easily remove these co-variates from the regression model (especially since they were not statistically significant).

We next used the fitted linear regression model (containing all ten predictor variables) to predict the responses for the test data. Further, we computed prediction intervals (PIs) and CIs for the predicted responses. Last, we estimated the test mean squared error (MSE) by calculating the test data mean prediction error (3111.27) and its standard error (361.09).

2.2 Best Subset Selection Using BIC

For the second model, we applied best subset selection using Bayesian information criterion (BIC) to select the optimal number of predictors. We then extracted the model coefficient estimates, predicted the responses for the test set, and calculated the estimated test MSE and its standard error.

Upon applying best subset selection to the training data using BIC, we found that the "best" subset had **six** predictor variables (sex, bmi, map, tc, tch, ltg). **Figure 2** shows two plots of the BIC values. When looking at the left plot, we see that the lowest BIC value corresponds to the six "best" predictors (look at the top row of black boxes). Also, the right plot in

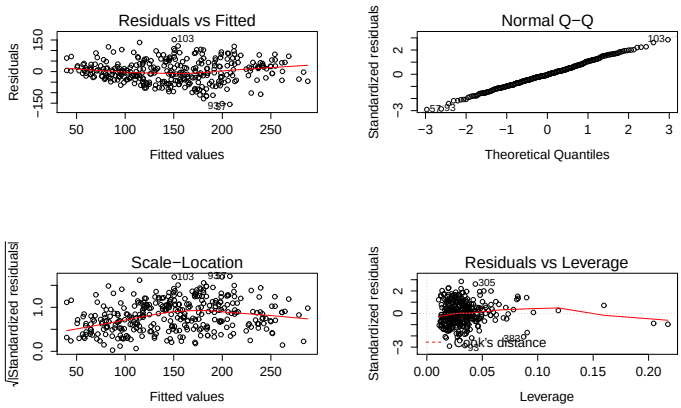

Figure 1: Linear Regression Model Diagnostic Plots

Figure 2 shows a curve of the BIC values as the number of predictors increases. We can clearly see that the optimal number of predictors corresponds to the lowest point (red dot) on the curve, which has the smallest BIC value of -201.13.

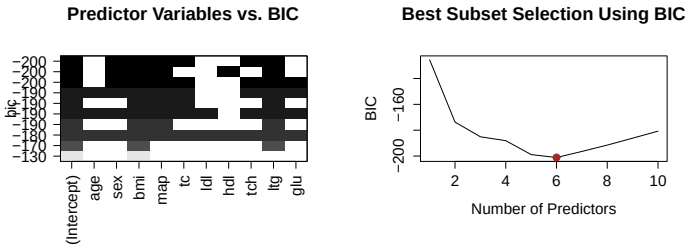

Figure 2: Using BIC to Select the Best Number of Predictors

Next, we extracted the model coefficients as displayed in **Table 2** below. As validation, we also fit a linear regression model (from the training data) using only these six best predictor variables to confirm their statistical significance (p-value < 0.001, RSE $= 53.94$, $R^2 = 0.5172$).

Table 2: Best Subset Selection (BIC) Model Coefficient Estimates

(Intercept)	sex	bmi	map	tc	tch	ltg
150.12	-306.04	538.83	389.07	-379.04	332.67	527.57

We next used the best subset selection model containing the six best predictor variables to predict the responses for the test data. Moreover, we used the six-variable fitted linear regression model to verify these predicted responses, as well as compute PIs and CIs. We also plotted the model diagnostics (which verified the linearity) and confirmed that there was no collinearity between these six predictors (all VIF values < 5). For both the best subset selection model and associated linear regression model, we estimated the test MSE by calculating the test data mean prediction error (3095.48) and it standard error (369.75).

When comparing the best subset selection model (with six predictors) to the previous linear regression model (with all ten predictors), the estimated test MSE is slightly lower (but its standard error is slightly higher). Note, however, that the R^2 value was greater and the RSE was lower for the ten-variable linear regression model compared to the six-variable model.

2.3 Best Subset Selection Using 10-fold Cross-Validation

For the third model, we applied best subset selection using 10-fold cross-validation to select the optimal number of predictors. We then extracted the model coefficient estimates, predicted the responses for the test set, and calculated the estimated test MSE and its standard error.

Upon applying best subset selection to the training data using 10-fold cross-validation (CV), we found that the best subset again had **six** predictor variables. **Figure 3** provides two plots showing the curve of CV errors as the number of predictors increases. When looking at the left plot, we see that the lowest mean CV error value (2978.91) corresponds to the six best

predictors (brown dot). This is confirmed in the right plot by the lowest root mean squared error (RMSE) value (54.58) on the curve (blue dot).

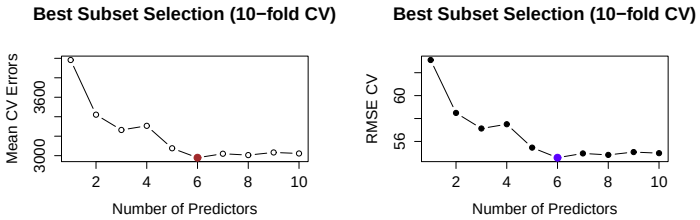

Figure 3: Using 10-fold CV to Select the Best Number of Predictors

Since the 10-fold CV method selected a six-variable model, we performed best subset selection again on the full training data set to get the "best" six-variable model. We then extracted the model coefficients as displayed in **Table 3** below. As validation, we again fit a linear regression model (from the training data) using only these six best predictor variables confirming their statistical significance (p-value < 0.001, RSE $= 53.94$, $R^2 = 0.5172$).

Table 3: Best Subset Selection (10-fold CV) Model Coefficient Estimates

(Intercept)	sex	bmi	map	tc	tch	ltg
150.12	-306.04	538.83	389.07	-379.04	332.67	527.57

As you can see, these model coefficient estimates are identical to those found by applying the BIC method. Therefore, the predicted responses for the test data, the PIs and CIs, the model diagnostics, the estimated test MSE (3095.48) and its standard error (369.75) were also identical to those found previously using BIC.

2.4 Ridge Regression Model Using 10-fold Cross-Validation

For the fourth model, we used ridge regression to fit a model using 10-fold cross-validation to select the largest value of λ such that the CV error is within one standard error of the minimum. Before conducting cross-validation to determine the appropriate λ value, we first fit the ridge regression model using the training set. **Figure 4** provides two plots of the estimated ridge regression coefficient values and CV error for varying values of λ.

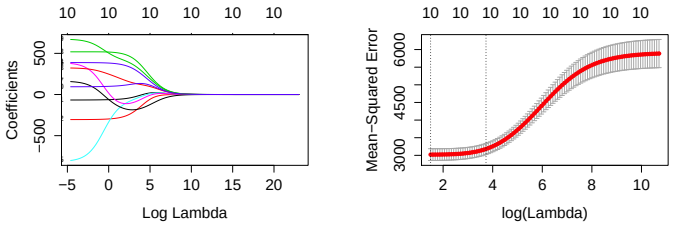

Figure 4: Estimated Coefficients and 10-fold CV Error for Ridge Regression Model

When looking at the left plot, we see that the estimated ridge regression coefficients decrease as the value of λ increases. In the right plot, we see that the lowest CV error is when $\ln(\lambda)$ equals roughly 1.59, which corresponds to $\lambda = 4.90$. When using this value of λ, the ridge regression model has a test data mean prediction error of 3074.38 with a standard error of 357.96. However, the largest value of λ such that the CV error is within one standard error of the minimum is $\lambda = 41.67$. Using this value of λ, the estimated test MSE is 3070.87 with a standard error of 350.55 (which is clearly a more accurate model compared to using the "minimal" value). We next extracted the estimated regression coefficients using $\lambda = 41.67$. You will notice in **Table 4** that the model has "shrunk" the coefficients towards zero (but none are equal to zero).

2.5 Lasso Model Using 10-fold Cross-Validation

For the final model, we used the lasso to fit a model using 10-fold cross-validation to select the largest value of λ such that the CV error is within one standard error of the minimum. Before conducting cross-validation to determine the appropriate λ

Table 4: Ridge Regression Model Coefficient Estimates

(Intercept)	age	sex	bmi	map	tc	ldl	hdl	tch	ltg	glu
149.99	-11.33	-156.91	374.45	264.90	-31.97	-66.90	-174.01	123.97	307.69	134.48

value, we first fit the lasso model using the training set. **Figure 5** provides two plots of the estimated lasso model coefficient values and CV error for varying values of λ.

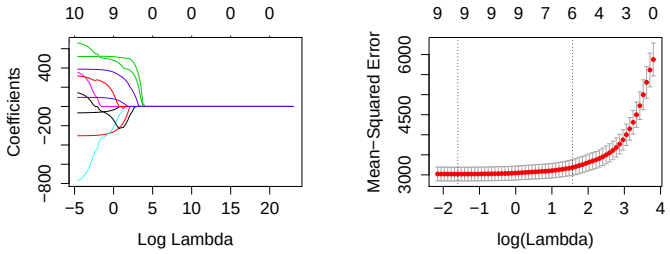

Figure 5: Estimated Coefficients and 10-fold CV Error for Lasso Model

When looking at the left plot, we see that the estimated lasso model coefficients decrease (some to zero) as the value of λ increases. In the right plot, we see that the lowest CV error is when $\ln(\lambda)$ equals roughly -1.6, which corresponds to $\lambda = 0.203$. When using this value of λ, the lasso model has a test data mean prediction error of 3103.65 with a standard error of 363.00. Note that both of these values are higher than the ridge regression model using the minimal value for λ. Nonetheless, the largest value of λ such that the CV error is within one standard error of the minimum is $\lambda = 4.79$. Using this value of λ, the estimated test MSE is 2920.04 with a standard error of 346.2, making the lasso model more accurate for prediction (i.e. lowest estimated test MSE and SE) compared to both the ridge regression model and the previous lasso model (using the "minimal" value).

In **Table 5** are the extracted estimated lasso model coefficients using $\lambda = 4.79$, where you will notice that the model eliminated (i.e., shrunk to zero) four out of the ten predictor variables.

Table 5: Lasso Model Coefficient Estimates

(Intercept)	sex	bmi	map	hdl	ltg	glu
149.95	-119.62	501.56	270.93	-180.29	390.55	16.59

3 Concluding Remarks

In this analysis, we employed several statistical learning methods to fit models for predicting disease progression (one year after baseline) using a subset (or all) of the ten baseline predictor variables. The model with the lowest test data mean prediction error (estimated test MSE) was the lasso model using 10-fold cross-validation (2920.04, SE = 346.2) followed by the ridge regression model using 10-fold cross-validation (3070.9, SE = 350.5). Nonetheless, the other fitted models provided relatively accurate predictions as well. **Table 6** below summarizes the five models:

Table 6: Summary of Statistical Learning Models

Model	(Intercept)	age	sex	bmi	map	tc	ldl	hdl	tch	ltg	glu	Test MSE	SE
OLS	149.92	-66.76	-304.65	518.66	388.11	-815.27	387.60	162.90	323.83	673.62	94.21	3111.27	361.09
BSS - BIC	150.12		-306.04	538.83	389.07	-379.04			332.67	527.57		3095.48	369.75
BSS - CV	150.12		-306.04	538.83	389.07	-379.04			332.67	527.57		3095.48	369.75
RR	149.99	-11.33	-156.91	374.45	264.90	-31.97	-66.90	-174.01	123.97	307.69	134.48	3070.87	350.55
Lasso	149.95		-119.62	501.56	270.93			-180.29		390.55	16.59	2920.04	346.22

4 Appendix

In this section, we provide the R code used to conduct all of the data analysis.

```
# Individual Project #1
# ADEC 7430: Big Data Econometrics

# Load the relevant libraries
library(lars)
library(car)
library(leaps)
library(glmnet)

# Load the diabetes data
data(diabetes)
data.all <- data.frame(cbind(diabetes$x, y = diabetes$y))

# There are 10 x baseline variables (age, sex, bmi, map, tc, ldl, hdl, tch, ltg, glu)
# The response (quantitative) variable is a measure of disease progression one year after baseline
fix(data.all)
names(data.all)
dim(data.all)

# Partition the patients into two groups: training (75%) and test (25%)
n <- dim(data.all)[1]                      # sample size = 442
set.seed(1306)                             # set random number generator seed to enable
# repeatability of results
test <- sample(n, round(n/4))              # randomly sample 25% test
data.train <- data.all[-test,]
data.test <- data.all[test,]
x <- model.matrix(y ~ ., data = data.all)[,-1] # define predictor matrix
# excl intercept col of 1s
x.train <- x[-test,]                           # define training predictor matrix
x.test <- x[test,]                             # define test predictor matrix
y <- data.all$y                                # define response variable
y.train <- y[-test]                            # define training response variable
y.test <- y[test]                              # define test response variable
n.train <- dim(data.train)[1]                  # training sample size = 332
n.test <- dim(data.test)[1]                    # test sample size = 110

# Fit the following models to the TRAINING set. For each model, extract the model
# coefficient estimates, predict the responses for the TEST set, and calculate
# the "mean prediction error" (and its standard error) in the TEST set.

# Plot a Scatterplot Matrix of the TRAINING data set and TEST data set
pairs(data.train)
pairs(data.test)

# Plot Histograms to check for normality of predictor variables
par(mfrow = c(3, 3))
hist(data.train$age); hist(data.train$bmi); hist(data.train$map); hist(data.train$tc);
hist(data.train$ldl); hist(data.train$hdl); hist(data.train$tch); hist(data.train$ltg);
hist(data.train$glu)

## Question 1: Fit a Least Squares Regression Model using all ten predictors
fix(data.train)                            # View the TRAINING data set
lm.fit <- lm(y ~ ., data = data.train)     # Fit the model using the TRAINING data set
summary(lm.fit)                            # Summary of the linear regression model
coef(lm.fit)                               # Extract the estimated regression model coefficients
confint(lm.fit)                            # Obtain a 95% CI for the coefficient estimates
par(mfrow = c(2, 2)); plot(lm.fit)         # Plot the model diagnostics
```

```
vif(lm.fit); cor(data.train)             # Check for collinearity (VIF > 10; tc, ldl, hdl, ltg)
fix(data.test)                                  # View the TEST data set
predict(lm.fit, data.test)                      # Predict the responses for the TEST data set
predict(lm.fit, data.test, interval = "prediction") # Prediction Interval of Predicted Responses
predict(lm.fit, data.test, interval = "confidence") # Confidence Interval of Predicted Responses
mean((data.test$y - predict(lm.fit, data.test))^2)  # Mean Predictor Error (test MSE) = 3111.27
sd((data.test$y - predict(lm.fit, data.test))^2)/sqrt(n.test) # Standard Error = 361.09


## Question 2: Apply Best Subset Selection using BIC to select the number of predictors and then
# fit a least squares regression model using the "best" subset of predictor variables
regfit.full <- regsubsets(y ~ ., data = data.train, nvmax = 10)
summary(regfit.full)
par(mfrow = c(1, 2))
plot(regfit.full, scale = "bic", main = "Predictor Variables vs. BIC")
reg.summary <- summary(regfit.full)
reg.summary$bic
reg.summary$bic[6]
plot(reg.summary$bic, xlab = "Number of Predictors", ylab = "BIC", type = "l",
main = "Best Subset Selection Using BIC")
which.min(reg.summary$bic)
points(6, reg.summary$bic[6], col = "brown", cex = 2, pch = 20)
coef(regfit.full, 6)

# Predict "function" for regsubsets()
predict.regsubsets <- function(object, newdata, id,...){
form <- as.formula(object$call[[2]])
mat <- model.matrix(form, newdata)
coefi <- coef(object, id = id)
xvars <- names(coefi)
mat[, xvars]%*%coefi
}
mean((data.test$y - predict(regfit.full, data.test, id = 6))^2)  # Mean Predictor Error = 3095.483
sd((data.test$y - predict(regfit.full, data.test, id = 6))^2)/sqrt(n.test) # Standard Error = 369.75

lm.bic <- lm(y ~ sex + bmi + map + tc + tch + ltg, data = data.train)
summary(lm.bic)                          # Summary of the linear regression model
coef(lm.bic)                             # Extract the estimated regression model coefficients
confint(lm.bic)                          # Obtain a 95% CI for the coefficient estimates
par(mfrow = c(2, 2)); plot(lm.bic)       # Plot the model diagnostics
vif(lm.bic);                             # Check for collinearity (all VIF <= 10)
predict(lm.bic, data.test)                      # Predict the responses for the TEST data set
predict(lm.bic, data.test, interval = "prediction") # Prediction Interval of Predicted Responses
predict(lm.bic, data.test, interval = "confidence") # Confidence Interval of Predicted Responses
mean((data.test$y - predict(lm.bic, data.test))^2)  # Mean Predictor Error = 3095.483
sd((data.test$y - predict(lm.bic, data.test))^2)/sqrt(n.test) # Standard Error = 369.75


## Question 3: Apply Best Subset Selection using 10-fold Cross-Validation to select the number
# of predictors and then fit the least squares regression model using the "best" subset.
k <- 10
set.seed(1306)
folds <- sample(1:k, nrow(data.train), replace = TRUE)
cv.errors <- matrix(NA, k, 10, dimnames = list(NULL, paste(1:10)))

# Let's write our own predict method
predict.regsubsets <- function(object, newdata, id,...){
form <- as.formula(object$call[[2]])
mat <- model.matrix(form, newdata)
coefi <- coef(object, id = id)
```

```
xvars <- names(coefi)
mat[, xvars]%*%coefi
}

for (j in 1:k) {
best.fit <- regsubsets(y ~ ., data = data.train[folds != j, ], nvmax = 10)
for (i in 1:10) {
pred <- predict(best.fit, data.train[folds == j, ], id = i)
cv.errors[j, i] = mean((data.train$y[folds == j] - pred)^2)
}
}

# This gives us a 10x10 matrix, of which the (i, j)th element corresponds
# to the test MSE for the ith cross-validation fold for the best j-variable model
cv.errors
mean.cv.errors <- apply(cv.errors, 2, mean)
mean.cv.errors
which.min(mean.cv.errors)
mean.cv.errors[6]

par(mfrow = c(1,2))
plot(mean.cv.errors, type = 'b', xlab = "Number of Predictors", ylab = "Mean CV Errors",
main = "Best Subset Selection (10-fold CV)")
points(6, mean.cv.errors[6], col = "brown", cex = 2, pch = 20)

rmse.cv = sqrt(apply(cv.errors, 2, mean))
rmse.cv[6]
plot(rmse.cv, pch = 19, type = "b", xlab = "Number of Predictors", ylab = "RMSE CV",
main = "Best Subset Selection (10-fold CV)")
points(6, rmse.cv[6], col = "blue", cex = 2, pch = 20)

# The cross-validation selects a 6-variable model, so we perform best subset
# selection on the training data set to get the best 6-variable model
reg.best <- regsubsets(y ~ ., data = data.train, nvmax = 10)
coef(reg.best, 6)

mean((data.test$y - predict(reg.best, data.test, id = 6))^2)  # Mean Predictor Error = 3095.483
sd((data.test$y - predict(reg.best, data.test, id = 6))^2)/sqrt(n.test) # Standard Error = 369.75

lm.cv.best <- lm(y ~ sex + bmi + map + tc + tch + ltg, data = data.train)
summary(lm.cv.best)                          # Summary of the linear regression model
coef(lm.cv.best)                             # Extract the estimated regression model coefficients
confint(lm.cv.best)                          # Obtain a 95% CI for the coefficient estimates
par(mfrow = c(2, 2)); plot(lm.cv.best)       # Plot the model diagnostics
vif(lm.cv.best)                              # Check for collinearity (all VIF <= 10)
predict(lm.cv.best, data.test)                          # Predict the responses for the TEST data
predict(lm.cv.best, data.test, interval = "prediction") # PI of Predicted Responses
predict(lm.cv.best, data.test, interval = "confidence") # CI of Predicted Responses
mean((data.test$y - predict(lm.cv.best, data.test))^2)  # Mean Predictor Error = 3095.483
sd((data.test$y - predict(lm.cv.best, data.test))^2)/sqrt(n.test) # Standard Error = 369.75


## Question 4: Ridge regression model using 10-fold cross-validation to select that largest
# value of lambda s.t. the CV error is within 1 s.e. of the minimum
par(mfrow = c(1,2))
grid <- 10^seq(10, -2, length = 100)
ridge.mod <- glmnet(x.train, y.train, alpha = 0, lambda = grid, thresh = 1e-12)
plot(ridge.mod, xvar = "lambda", label = TRUE)

set.seed(1306)
```

```
cv.out <- cv.glmnet(x.train, y.train, alpha = 0)
plot(cv.out)
bestlam <- cv.out$lambda.min
bestlam                                     # Lambda = 4.904021 (leads to smallest CV error)
log(bestlam)
ridge.mod <- glmnet(x.train, y.train, alpha = 0, lambda = bestlam)
ridge.pred <- predict(ridge.mod, s = bestlam, newx = x.test)
mean((ridge.pred - y.test)^2)               # Mean Prediction Error = 3074.378
sd((ridge.pred - y.test)^2)/sqrt(n.test)    # Standard Error = 357.9628

largelam <- cv.out$lambda.1se
largelam                                    # Lambda = 41.67209 (largest lambda w/in 1 SE)
ridge.mod <- glmnet(x.train, y.train, alpha = 0, lambda = largelam)
ridge.pred <- predict(ridge.mod, s = largelam, newx = x.test)
mean((ridge.pred - y.test)^2)               # Mean Prediction Error = 3070.87
sd((ridge.pred - y.test)^2)/sqrt(n.test)    # Standard Error = 350.5467

# Here are the estimated coefficients
predict(ridge.mod, type = "coefficients", s = largelam)[1:11,]


## Question 5: Lasso model using 10-fold cross-validation to select that largest
# value of lambda s.t. the CV error is within 1 s.e. of the minimum
par(mfrow = c(1,2))
grid <- 10^seq(10, -2, length = 100)
lasso.mod <- glmnet(x.train, y.train, alpha = 1, lambda = grid, thresh = 1e-12)
plot(lasso.mod, xvar = "lambda", label = TRUE)

set.seed(1306)
cv.out <- cv.glmnet(x.train, y.train, alpha = 1)
plot(cv.out)
bestlam <- cv.out$lambda.min
bestlam                                     # Lambda = 0.2026 (leads to smallest CV error)
log(bestlam)
lasso.mod <- glmnet(x.train, y.train, alpha = 1, lambda = bestlam)
lasso.pred <- predict(lasso.mod, s = bestlam, newx = x.test)
mean((lasso.pred - y.test)^2)               # Mean Prediction Error = 3103.65
sd((lasso.pred - y.test)^2)/sqrt(n.test)    # Standard Error = 363.0016

largelam <- cv.out$lambda.1se
largelam                                    # Lambda = 4.791278 (largest lambda w/in 1 SE)
lasso.mod <- glmnet(x.train, y.train, alpha = 1, lambda = largelam)
lasso.pred <- predict(lasso.mod, s = largelam, newx = x.test)
mean((lasso.pred - y.test)^2)               # Mean Prediction Error = 2920.041
sd((lasso.pred - y.test)^2)/sqrt(n.test)    # Standard Error = 346.2248

# Here are the estimated coefficients
lasso.coef <- predict(lasso.mod, type = "coefficients", s = largelam)[1:11,]
lasso.coef[lasso.coef != 0]
```

References

[1] Efron, B. and Hastie, T. (2003). *LARS software for R and Splus*. http://www-stat.stanford.edu/ hastie/Papers/LARS.

[2] Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004) Least angle regression. *Annals of Statistics*, **32**, 407451.