Modeling Boston Housing Prices Using Regularization Methods

Pascal Berlage

Tasks

The study of housing markets has been an extensive topic in economics and public policy research. Acquiring a knowledge of the housing prices is essential for policymakers, urban planners and real estate agents trying to understand issues about affordability. Accurate house price models help us in identifying poor neighborhoods, measuring environmental impact on property values and improving taxation rules. The Boston Housing data set (Harrison & Rubinfeld, 1978) is one of the most commonly used data sets , which include socioeconomic, environmental and structural variables that affect housing in the Boston metropolitan area . It has 506 observations to estimate the median value of owner-occupied homes in suburban Boston from the 1970s. There are 13 predictor variables in the data set listed below:

- crim: Town-specific per capita crime rate
- zn: The percentage of residential land that is zoned for lots larger than 25,000 square feet
- indus: The percentage of each town's non-retail business acres
- chas: Charles River dummy variable If the tract borders the river, it is 1; if not, it is 0.
- nox: The concentration of nitrogen oxides (parts per 10 million)
- rm: The average number of rooms in a house
- age: The percentage of owner-occupied homes constructed before 1940
- dis: Weighted travel times to five job hubs in Boston
- rad: Radial highway accessibility index
- tax: The full-value property tax rate per $10,000
- ptratio: the ratio of students to teachers by town
- black - 1000 (Bk - 0.63).where Bk is the percentage of black residents in each town.
- lstat: The population's lower status (percent)

The data set is a typical regression problem that encapsulates the complexity of variable selection and over-fitting in models. When having some intercorrelation between predictors and a smaller sample size, it's an valuable test to find differences between traditional and regularized regression approaches.

We will be investigating whether LASSO or ridge regression methods are better performers than traditional subset selection in predicting house prices. Best subset is widely known for its easy interpretations and variable selection, LASSO uses automatic variable selection to keep efficient accuracy, and ridge deals with correlated predictors using coefficient shrinkage without completely removing them. The data set is well known online, and past analyses have used multiple methods to interpret the data. Their methods included step-wise selection, ridge regression for dealing with intercorrelation, and the LASSO method.

There were 3 different methods we used for interpreting the data. (1) best subset selection with BIC criterion for easy interpretation and explicit variable selection. (2) LASSO method for automated variable selection with the L1 penalty; and (3) ridge regression with an L2 penalty to deal with multicollinearity and keeping the predictors. The analysis done reveals that regularization methods did better than best subset selection, with ridge regression yielding the best prediction accuracy and all methods consistently finding room count, neighborhood status and school quality as the most impactful price variables.

Results

We found that both regularization methods did better than best subset selection, with ridge regression getting the most optimal prediction accuracy (MSE = 27.223, $R^2 = 0.596$). Furthermore, the LASSO method was close to ridge, with a prediction accuracy of (MSE = 27.320, $R^2 = 0.595$). Best subset selection was followed last (MSE = 28.801, $R^2 = 0.573$), with blatant signs of over-fitting. The results indicated that home prices were most affected by room count, neighborhood wealth and school quality in all 3 models.

Discussion

We are fairly confident in the findings, as they've shown to be consistent in each method. The reality that LASSO and ridge HDI tests both had 9 statistically significant predictors improves the consistency. Ridge regression yielded the most optimal prediction accuracy (MSE: 27.223, R^2: 0.596), next one being LASSO (MSE: 27.320, R^2: 0.595). The two regularization methods perform significantly better than best subset selection (MSE: 28.801, R^2: 0.573). The improvement of 5.1% using ridge instead of the best subset selection method shows a slight difference, and the small 0.35% difference between ridge and LASSO concludes that they yield similar outputs. Additionally, the strength of important predictors such as the room count (rm), neighborhood status (lstat), and school quality (ptratio) with all three models, combined with a high statistical significance suggests that these are major influences of housing prices in the Boston area.

However, our confidence is moderate with the R-squared values (0.57-0.60). It means that the models identify the primary relationships, and a portion of the variance in the Boston housing prices is unknown. Though this is a common obstacle in real world interpretations, and can be explained for many reasons such as architecture, property style, or local market trends.

The ridge regression and LASSO's improved performance makes it plausible to believe it is more effective to keep all potential predictors and shrink them towards zero doing hard variable selection instead of completely removing them. This also signifies that the housing prices in 1970s Boston were not influenced just by a small quantity of factors, but by a larger quantity of smaller predictors. An example of this is the shrinkage of the nox coefficient from -15.58 in the best subset selection model to -9.97 in the ridge regression model. This signifies that the impact of pollution is mixed with other negative traits, such as older housing stocks or amount of crime. Ridge regression's skill to distribute predictive responsibility among the correlated variables provides a more stable and credible estimate of its true, solitary effect.

There are a couple methods that could have been used to further improve the analysis without changing methods, such as k-fold cross validation, which provides a more stable performance estimate instead of a unique training set-validation split. Another improvement is bootstrapping, which would give us a confidence interval for performance metrics, providing enhanced uncertainty quantification for MSE and R^2 estimates. There are other solutions which could also have provided advantages. Elastic net could be used to get an L1 and L2 regularization, which deals with different predictors more effectively than ridge or LASSO. Bayesian regression is another, as it would give complete uncertainty quantification for coefficients and predictions.

While best subset selection gives us the clearest interpretation, its likely to get over fitted which lowers its credibility for prediction. The LASSO and ridge methods yielded better results, with ridge being slightly more effective. For functional housing prediction, regularization methods provide a better balance for easy interpreting and accuracy, but there are still areas of improvement for diverse modeling approaches.

Code with Analysis

```
library(MASS)
library(leaps)
library(glmnet)
str(Boston)
```

```
## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ black  : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

```
names(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

```
library(leaps)
set.seed(1)
```

```
#creating a training and validation set
d <- nrow(Boston)
value_train <- sample(1:d, size = round(0.7 * d), replace = FALSE)
training_set <- Boston[value_train, ]
validation_set <- Boston[-value_train, ]

#output
print(paste("training set:", nrow(training_set)))
```

```
## [1] "training set: 354"
```

```
print(paste("validation set:", nrow(validation_set)))
```

```
## [1] "validation set: 152"
```

```
library(MASS)
library(leaps)
library(car)
```

```r
set.seed(1)

#using the best subset selection
subset_selection <- regsubsets(medv ~ ., data = training_set, nvmax = 13)
optimal_summary <- summary(subset_selection)
model_best <- which.min(optimal_summary$bic)
optimal_coef <- coef(subset_selection, model_best)

#printing output for best model size
print(paste("best model size (BIC):", model_best))
```

```
## [1] "best model size (BIC): 9"
```

```r
print("coefficients:")
```

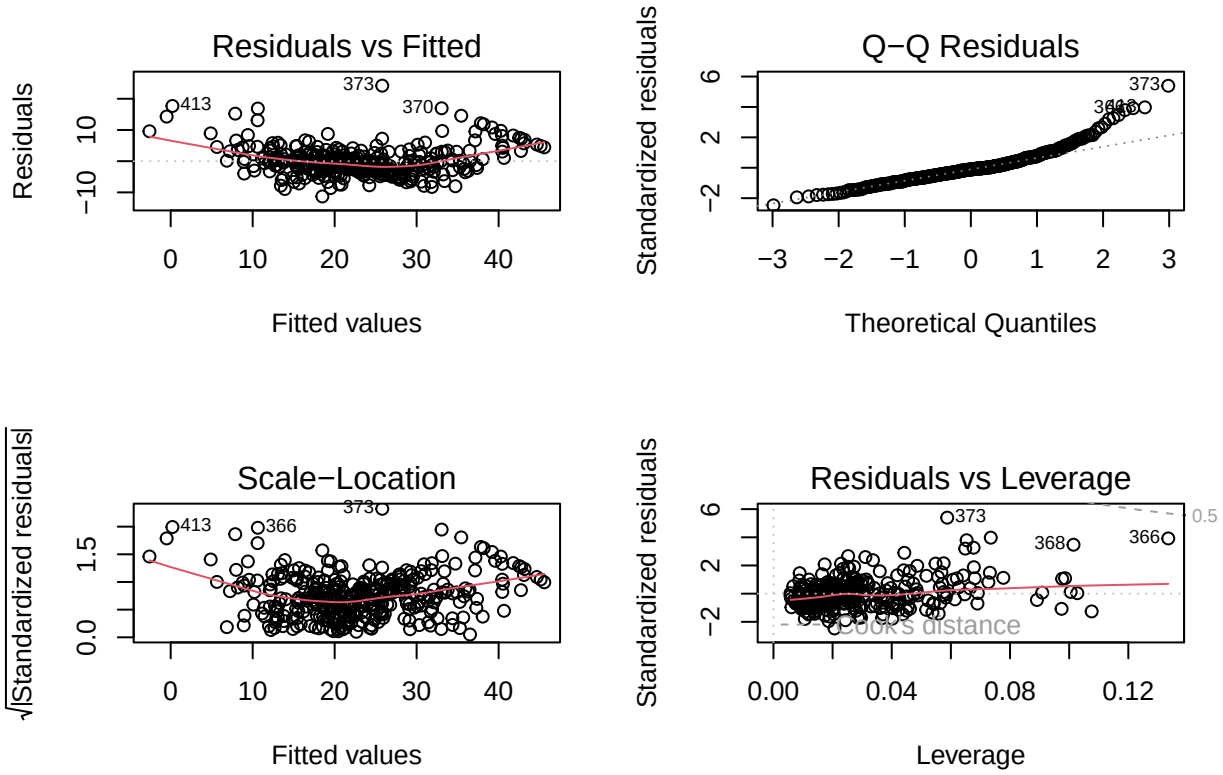
```
## [1] "coefficients:"
```

```r
print(optimal_coef)
```

```
##  (Intercept)         chas          nox           rm          dis          rad
##  28.46336881   3.44791468 -15.57582410   4.92201724  -1.10120852   0.26472039
##          tax      ptratio        black        lstat
##  -0.01179543  -1.05450009   0.01138604  -0.51043785
```

Our linear regression equation in this case is: medv = 28.463 + 3.448*chas* - *15.576*nox + 4.922*rm* - *1.101*dis + 0.265*rad* - *0.012*tax - 1.055*ptratio* + *0.011*black - 0.510*lstat

```r
#refit best model
optimal_vars <- names(optimal_coef)[-1]
lm <- lm(as.formula(paste("medv ~", paste(optimal_vars, collapse = " + "))), data = training_set)


#creating visual plot
par(mfrow = c(2,2))
plot(lm)
```




```
par(mfrow = c(1,1))
```

For the graphs:

- Residuals vs Fitted graph: has a U shaped pattern, indicating its non linearity.
- Q-Q Plot: There is a stronger deviation from the line at the upper tail, meaning there is a right tail.
- Scale Location: red trend slopes down, then goes upwards. The variability increases at higher fitted values.
- Residuals vs Leverage: some points beyond Cook's distance lines, but most points have low leverage.

```
# hypothesis testing
print("model summary:")
```

```
## [1] "model summary:"
```

```
print(summary(lm))
```

```
##
## Call:
## lm(formula = as.formula(paste("medv ~", paste(optimal_vars, collapse = " + "))),
```

```
##     data = training_set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.2999  -2.7938  -0.5164   1.8201  24.1955
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  28.463369   6.051980   4.703 3.71e-06 ***
## chas          3.447915   0.928371   3.714 0.000238 ***
## nox         -15.575824   4.039656  -3.856 0.000138 ***
## rm            4.922017   0.474971  10.363  < 2e-16 ***
## dis          -1.101209   0.188115  -5.854 1.12e-08 ***
## rad           0.264720   0.069194   3.826 0.000155 ***
## tax          -0.011795   0.003789  -3.113 0.002006 **
## ptratio      -1.054500   0.143614  -7.343 1.53e-12 ***
## black         0.011386   0.003047   3.736 0.000219 ***
## lstat        -0.510438   0.056552  -9.026  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.624 on 344 degrees of freedom
## Multiple R-squared:  0.7721, Adjusted R-squared:  0.7661
## F-statistic: 129.5 on 9 and 344 DF,  p-value: < 2.2e-16
```

```
# finding validation amount
optimal_prediction <- predict(lm, newdata = validation_set)
optimal_mse <- mean((validation_set$medv - optimal_prediction)^2)
print(paste("validation MSE:", round(optimal_mse, 3)))
```

```
## [1] "validation MSE: 28.801"
```

```
# finding validation R^2
ss_resid <- sum((validation_set$medv - optimal_prediction)^2)
ss_tot <- sum((validation_set$medv - mean(validation_set$medv))^2)
rsq_validation <- 1 - (ss_resid/ss_tot)


print(paste("validation R^2:", round(rsq_validation, 3)))
```

```
## [1] "validation R^2: 0.573"
```

When analyzing the best subset selection with BIC, the best model for finding median home value contains 9 predictors: chas, nox, rm, dis, rad, tax, ptratio, black, and lstat. We excluded crim, zn, indus and age. In the linear regression equation shown above, all coefficients are statistically significant ($p<0.05$). The validation MSE is 28.801, meaning that there is satisfactory prediction accuracy on unseen data. The R^2 value is 0.573, meaning it explains about 57.3% of the variance in housing prices on unseen data.

```
library(glmnet)
library(hdi)
set.seed(1)
```

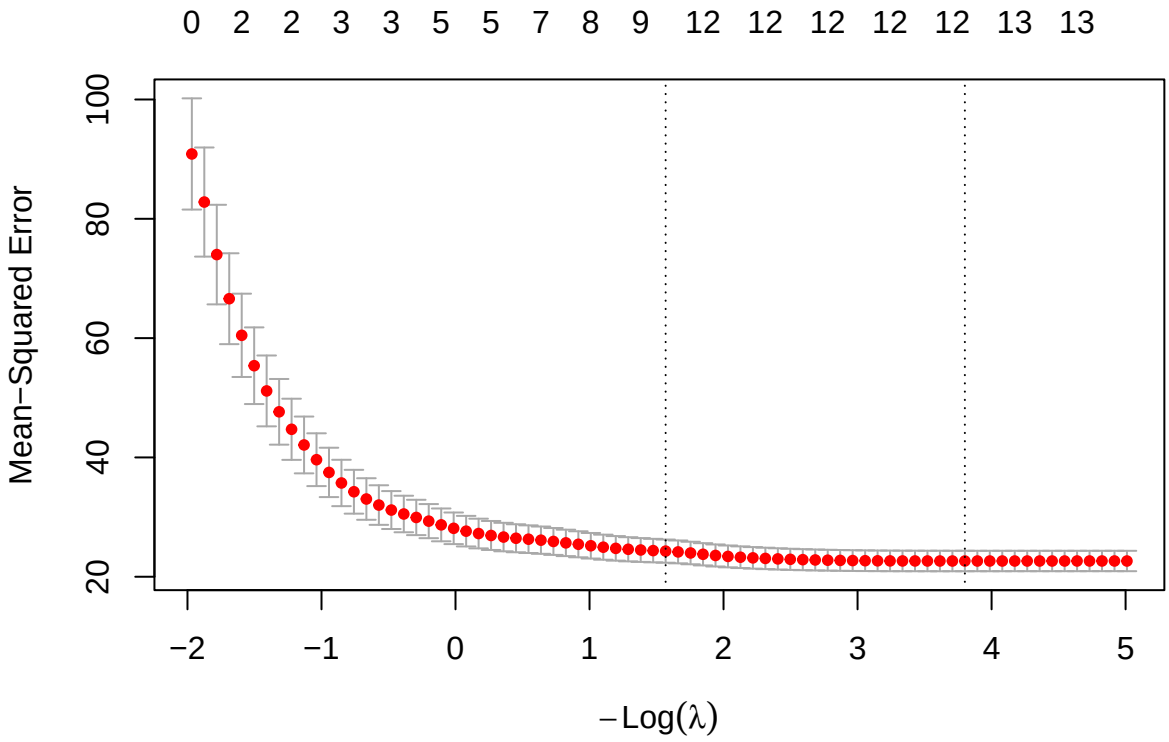
```r
train_model_1 <- model.matrix(medv ~ ., training_set)[,-1]
train_model_2 <- training_set$medv
valid_model_1 <- model.matrix(medv ~ ., validation_set)[,-1]
valid_model_2 <- validation_set$medv

# using the cross validation method
cross_la <- cv.glmnet(train_model_1, train_model_2, alpha = 1)
optimal_la_lasso <- cross_la$lambda.min
fitting_lasso <- glmnet(train_model_1, train_model_2, alpha = 1, lambda = optimal_la_lasso)

# coefficients
print("LASSO coefficients:")
```

```
## [1] "LASSO coefficients:"
```

```r
print(coef(fitting_lasso))
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept)  2.660783e+01
## crim        -6.375115e-02
## zn           3.220586e-02
## indus        3.500995e-04
## chas         3.450251e+00
## nox         -1.339019e+01
## rm           4.888803e+00
## age         -1.333644e-02
## dis         -1.350846e+00
## rad          2.623138e-01
## tax         -1.162061e-02
## ptratio     -9.227724e-01
## black        1.046836e-02
## lstat       -4.775709e-01
```

```r
# graph for cross-validation
plot(cross_la)
```




```
print(paste("Optimal lambda:", round(optimal_la_lasso, 5)))
```

```
## [1] "Optimal lambda: 0.02236"
```

With a lambda value of 0.02236, the equation for the LASSO model is: medv = 26.61 - 0.064*crim* + *0.032*zn + 0.00035*indus* + *3.45*chas - 13.39*nox* + *4.89*rm - 0.013*age* - *1.35*dis + 0.262*rad* - *0.012*tax - 0.923*ptratio* + *0.010*black - 0.478*lstat

The lambda value maintains the effectiveness of regularization, with coefficients like indus and age almost removed from the model with values close to 0.

```
# refitting with hdi
print("LASSO inference with hdi:")
```

```
## [1] "LASSO inference with hdi:"
```

```
lasso_inference <- lasso.proj(x = train_model_1, y = train_model_2)

# p values/confidence intervals
print("LASSO p-values from hdi:")
```

```
## [1] "LASSO p-values from hdi:"
```

```r
print(lasso_inference$pval)
```

```
##         crim           zn        indus         chas          nox           rm
## 1.790747e-01 5.274771e-02 8.561583e-01 2.363464e-04 1.978866e-03 4.799993e-23
##          age          dis          rad          tax      ptratio        black
## 2.541985e-01 3.672789e-09 1.145570e-04 9.836935e-04 3.739768e-09 6.915002e-04
##        lstat
## 1.452213e-13
```

```r
print("95% CI:")
```

```
## [1] "95% CI:"
```

```r
print(confint(lasso_inference))
```

```
##                 lower        upper
## crim    -1.468901e-01  0.027405589
## zn      -3.823378e-04  0.064827378
## indus   -1.269011e-01  0.152766040
## chas     1.691133e+00  5.552785639
## nox     -2.262591e+01 -5.074793797
## rm       4.043779e+00  6.043748325
## age     -4.879238e-02  0.012901761
## dis     -1.874585e+00 -0.939487374
## rad      1.410843e-01  0.432537673
## tax     -2.204190e-02 -0.005600264
## ptratio -1.258934e+00 -0.630702757
## black    4.668841e-03  0.017440642
## lstat   -5.928678e-01 -0.344349695
```

```r
# finding the prediction and validation MSE
prediction_l <- predict(fitting_lasso, s = optimal_la_lasso, newx = valid_model_1)
lasso_mse <- mean((valid_model_2 - prediction_l)^2)
print(paste("validation MSE (LASSO):", round(lasso_mse, 3)))
```

```
## [1] "validation MSE (LASSO): 27.32"
```

```r
# finding the validation R-squared
resid_lasso <- sum((valid_model_2 - prediction_l)^2)
sum_sq_lasso <- sum((valid_model_2 - mean(valid_model_2))^2)
sq_lasso_r <- 1 - (resid_lasso/sum_sq_lasso)

print(paste("validation R-squared (LASSO):", round(sq_lasso_r, 3)))
```

```
## [1] "validation R-squared (LASSO): 0.595"
```

The LASSO method using cross validation gave an optimal tuning parameter of lambda = 0.02236. For this method, it kept all the coefficients but shrunk the coefficients, some nearly to zero (indus and age are 0.00035 and -0.0133). We also see shrunken coefficients in the fitted model when comparing it to the least ordinary squares. This was most prevalent in the nox coefficient (-13.39 vs -14.58 in the refitted model).

When the hypothesis test results were found using the hdi package, it showed that 9 out of 13 coefficients were statistically significant ($p < 0.05$). The 4 non-statistically significant variables were crim ($p = 0.179$), indus ($p = 0.856$), age ($p = 0.254$), and zn ($p = 0.053$), demonstrating how LASSO identifies significant predictors by reducing them.

Overall, the LASSO method yielded a validation MSE of 27.32 and R^2 of 0.595, showing a slight improvement in MSE compared to the best subset selection. The gap between training R^2 and validation R^2 compared to best subset selection means there was some overfitting.

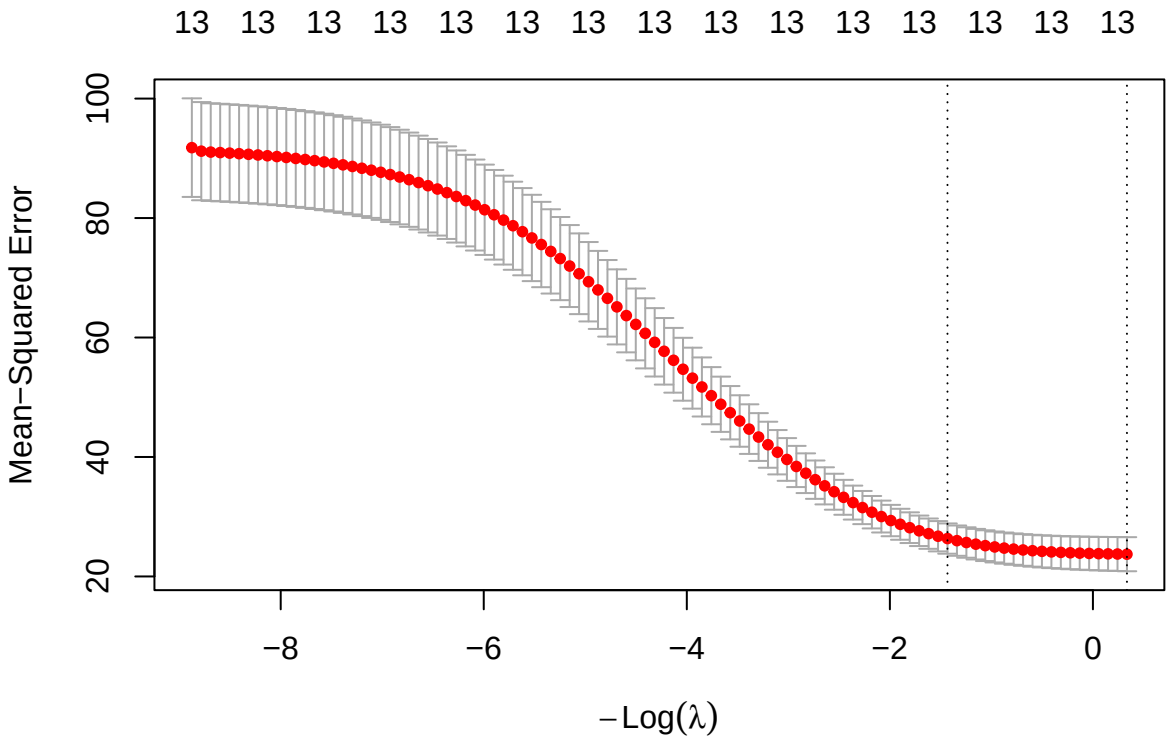
```
# cross validation ridge regression
cross_val_ridge <- cv.glmnet(train_model_1, train_model_2, alpha = 0)
opti_lambda_r <- cross_val_ridge$lambda.min
fitting_ridge <- glmnet(train_model_1, train_model_2, alpha = 0, lambda = opti_lambda_r)


print("ridge coefficients:")
```

```
## [1] "ridge coefficients:"
```

```
print(coef(fitting_ridge))
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) 21.657096405
## crim        -0.058210466
## zn           0.025117633
## indus       -0.031424837
## chas         3.650759583
## nox         -9.968530480
## rm           4.896687345
## age         -0.014689551
## dis         -1.080639450
## rad          0.156374790
## tax         -0.006898286
## ptratio     -0.860180419
## black        0.010100238
## lstat       -0.435877428
```

```
# ridge graph
plot(cross_val_ridge)
```




```
print(paste("optimal lambda (ridge):", round(opti_lambda_r, 5)))
```

```
## [1] "optimal lambda (ridge): 0.71521"
```

With a lambda value of 0.71521, the equation for the ridge model is:

medv = 21.66 - 0.058*crim* + *0.025*zn - 0.031*indus* + *3.65*chas - 9.97*nox* + *4.90*rm - 0.015*age* - *1.08*dis + 0.156*rad* - *0.007*tax - 0.860*ptratio* + *0.010*black - 0.436*lstat

```
print("ridge inference with hdi:")
```

```
## [1] "ridge inference with hdi:"
```

```
ridge_inferen <- ridge.proj(x = train_model_1, y = train_model_2)


# p-values and confidence intervals from hdi
print("ridge p-values from hdi:")
```

```
## [1] "ridge p-values from hdi:"
```

```
print(ridge_inferen$pval)
```

```
##         crim           zn        indus         chas          nox           rm
## 1.273404e-01 4.016266e-02 7.347423e-01 6.493942e-04 2.203148e-03 4.602384e-20
##          age          dis          rad          tax      ptratio        black
## 3.720621e-01 1.811070e-08 2.000335e-04 2.772490e-03 1.701734e-08 1.360076e-03
##        lstat
## 3.203512e-13
```

```
print("95% confidence intervals:")
```

```
## [1] "95% confidence intervals:"
```

```
print(confint(ridge_inferen))
```

```
##                 lower        upper
## crim     -0.159128909  0.019877881
## zn        0.001584946  0.069036947
## indus    -0.120985289  0.171557999
## chas      1.463232691  5.418566946
## nox     -23.472868488 -5.148722335
## rm        3.826918982  5.906673755
## age      -0.046634406  0.017449134
## dis      -1.880326752 -0.909099806
## rad       0.138541595  0.447279467
## tax      -0.021848844 -0.004553148
## ptratio  -1.262865627 -0.611491022
## black     0.004143477  0.017210058
## lstat    -0.606577487 -0.349399894
```

```
# finding prediction and validation MSE
prediction_ridge <- predict(fitting_ridge, s = opti_lambda_r, newx = valid_model_1)
ridge_mse <- mean((valid_model_2 - prediction_ridge)^2)


print(paste("validation MSE (ridge):", round(ridge_mse, 3)))
```

```
## [1] "validation MSE (ridge): 27.223"
```

```
# finding validation R^2 for ridge
ridge_sum_of_squares <- sum((valid_model_2 - prediction_ridge)^2)
r_total_sum <- sum((valid_model_2 - mean(valid_model_2))^2)
sqr_ridge <- 1 - (ridge_sum_of_squares/r_total_sum)


print(paste("validation R-squared (ridge):", round(sqr_ridge, 3)))
```

```
## [1] "validation R-squared (ridge): 0.596"
```

The ridge regression method with cross validation yielded an optimal tuning parameter of lambda = 0.71521, which is a lot larger than LASSO's lambda value. This is because there is more regularization. Ridge kept all its predictors but shrunk its coefficients for all variables. We can see that the nox variable dropped the most in the OLS model dropped from -14.58 to -9.97 in the ridge model.

When using the hdi package for refitting, we see that 9 out of 13 predictors are statistically significant ($p < 0.05$), which is consistent with the LASSO method.There are three variables that are not statistically significant and are the same as LASSO: crim ($p = 0.127$), indus ($p = 0.735$), and age ($p = 0.372$). .It's safe to conclude that ridge yielded the best validation MSE of 27.223 and R^2 of 0.596.The minimal gap between the values of R^2 (0.778) and validation R^2 (0.596) show good generalization.

Comparing the 3 models, the one with best prediction accuracy is ridge regression with a validation MSE of 27.223, second best is LASSO with a validation MSE of 27.320, and third place is the best subset selection (MSE = 28.801). This matches not only with MSE values, but also with R-squared as ridge has the highest R^2 of 0.596. Therefore, ridge regression is most optimal for prediction accuracy.

When looking at trading off, it is important to judge each model carefully:

- Best subset selection (MSE = 28.801, $R^2 = 0.573$): 9 predictors, most interpretable model but has worse prediction accuracy, higher MSE than both ridge and LASSO
- LASSO (MSE = 27.320, $R^2 = 0.595$): 9 predictors, optimal balance and almost same accuracy as ridge, with a very small increase in MSE, though has automatic variable selection
- Ridge regression (MSE = 27.223, $R^2 = 0.596$): 13 predictors, best accuracy though uses all predictors so it may be harder to interpret

Judging the 3 models, I would choose the LASSO method, even if the MSE is lower, as they are almost the same (27.320 vs 27.223). It also selects the most significant variables, which gives a model easier to interpret than ridge's containing 13 predictors.In summary, the cost of a tiny increase in MSE is worth the benefits of a more optimal and interpretable model.