Statistical Learning Project

Ammar Hasan

11 November 2018

Contents

1 Introduction

This report summaries the steps undertaken to produce and evaluate linear regression models of the value of housing in Boston Standard Metropolitan to predict the value of logarithmic crime rate (lcrim) using other variables. The model would be built after exploratory and unsupervised statistical analysis of the data which is carried first to gain an understanding on the data characteristics and structure before hand. The models would be build using both sub-setting (best fit and step wise) and regularisation methods (LASSO and Ridge Fit) methods, these methods will be compared using k fold cross validation.

Any output (tables and plots) is placed in the Appendix at the end of the report with the R code that generated it, the description and analysis of the methods and their output is found in the document body which cross references the appendix content. Values in tables and other numerical results are corrected to 3 decimal places unless stated otherwise.

2 Exploratory Data Analysis

Before building a linear model, it is wise to first understand the overall structure of the data itself to get a feeling for the data characteristics and how the variables relate to one another - especially how they relate to the response variable (lcrime).

2.1 Data Spread and Location

To analyse the distribution of the data, the quantiles and means will be examined. In particular, this part of the report will examine outliers, scale, consistency and certainty.

2.1.1 Mean Vector

Using the col means functions a vector of mean averages can be produced for all the predictors and the response variables in the Boston data-set.

The returned table (transposed) is in table 1 in Appendix A.2, and shows that the means are well spread out from one another, which suggests a difference in the nature of the measurements.

Examining the structure of the variables in the Boston data set using the ?Boston confirms that the nature of the measurements vary from Full-value property-tax rate per $10,000 for tax to Nitrogen oxides parts per 10 million for nox for instance, which obviously suggests that the measurements cannot be directly compared scale to scale (standarisation might be required).

2.1.2 Box Plot and Quantiles

As previously stated in the previous section, the variables are of different natures and scales, hence any scale to scale comparison needs standarisation. To standardise the data a scale transform was applied using the base R scale() function, and using the boxplot base function the plot in figure 1 is generated.

The following stands out of the plot:

- black, rm, zn and medv predictor variables have significantly more outliers than the other variables.And hence more uncertain and also their averages can get skewed.
- chas predictor variable seems to have a very tight ranges that are practically identical. This is because this is a binary variable.

- zn, rad and tax predictor variables have a short Q1 to Q2 range compared to the Q2 to Q3 range, suggesting that the lower values of the data are very tightly clustered. black has the opposite problem.
- rm, lstat, mdev and ptratio have long minimum and maximum ranges in comparison to their IQR ranges, and hence more extreme values. This means that their averages can get skewed.

2.2 Data Relationships

This section of the report will look into how the variables (predictors and response) in bivariate data relate and interact using numerical correlation matrices and graphic pairs plot

2.2.1 Pairs Plot

The following relationships stick out when observing the pairs plot for the response against other variables in figure 2:

- Relationships/correlation with lcrime:
  - age and medv has a moderate negative relationship with lcrime
  - nox and lstat has a strong/moderate positive relationship with lcrim
  - rm and zn both have weak/moderate negative relationships with lcrime
  - tax, rad, pratio, indus and black appear to have unclear correlation
- Predictor variables relationships:
  - Some predictors have strong relations with one another: rm has a strong negative relationship with lstat but a strong positive one with medv. medv and lstat also have a strong negative relationship
- Chas (river dummy variable) and disf (distance to Boston employment centered) appear to have values in levels and are also difficult to analyse in a pairs plot

2.2.2 Correlation Matricies

The correlation matrix in table 2 confirms the findings of the previous section but also helps clarify some of relationships that were unclear before:

- Chas has a very weak relationship with lcrime, and a weak or very weak relationship with most other variables.
- disf has a moderate negative relationship with lcrime, a strong/moderate negative relationship with indus and a positive moderate relation with zn
- Tax and rad have strong relationships with lcrime that were difficult to spot before due to irregularities in their plots. Moreover, indus has a moderate positive relationship with lcrime and ptartio a weak positive one.
- Tax and rad have a very strong positive relationship

2.2.3 Data Relationships Summary

To summarise, it seems that the relationships suggest that there are a couple of variables that might have a strong impact to model (e.g. rad or tax). Moreover, the relationships also suggest that many will be subsetted due to relationships that can be represented with other variables(e.g. rad or tax), or very poor relationships with all variables (e.g. chas).

3 Unsupervised Learning

This section looks into apply unsupervised learning techniques to help understanding patterns and structures in the data to help understand their effects on the model. In this report, Principle Component Analysis will be used to help find which set of predictors cause the most variation on the data to help with model coefficient interpretation and gaining an understanding of which predictors contribute to the model.

3.1 Principle Component Analysis

3.1.1 Variation Proportions

Table 3 shows the summary of PCA run on the data, showing the variance each PC to and the accumulation of it. The summary shows that the first component contributed to 50% of the variance, and the first 4 contribute to 70%. Since the first 3 contribute to 70% and also where the variation curve in the scree plot in 3 diminishes for a second time.

3.1.2 Component Interpretation (According to Table 4)

3.1.2.1 Component 1

Dominated by positive lcrime, tax, indus and nox. This means that it represents areas with large non-retail business but high crime and nitrogen oxide pollution.

3.1.2.2 Component 2

Dominated by negative rm and medv, meaning that it represents less median house values and number of rooms.

3.1.2.3 Component 3

Dominated high accessibility highways, tax rate and residual areas.

3.1.2.4 Plot

A plot of the observations scores for component 1 vs component2 is shown in figure 4 shows most observations score high for component 2 but are more varied for component 1. Meaning that most observations have relative low median house value and number of rooms, but a varied crime, pollution and non-retail business.

4 Supervised Learning (Linear Modelling)

In this section of the report linear models are fitting and tested against each other using K Fold Cross Validation. By the end of this section a summary will conclude which models are best and why.

4.1 Model Fitting

4.1.1 Subset Selection using Best Fit

Best Fit Subset Selection Tries all possible predictors p combinations to find the "best" model using SSE (optimisation problem). The method's results are shown in table 5 as the selected best predictors from 13 variables (everything except lcrime) to 1 variable.

4.1.1.1 Subsetting Results analysis

When we look at the results we can notice that the first variable to be dropped is tax. tax being the first dropped predictor seems to lines up with the discussion in the Data Relationships Summary, as it was stated that either tax/rad can be represented by the other (and hence can be dropped).

Moreover it was also stated in the Relationships summary that chas correlated little to lcrime and other variables (little to no prediction can be made with it), so it's no surprise that it was dropped second.

Also, it can be noticed that usually the highest impacting predictors in the PCs stick for longer before dropping with a few exceptions. This particularly true for PC1, as with the exception of tax (which was subsetted for rad) some of the high scoring variables stuck around for long (e.g. nox), which is not surprising since it would be sensible for the most variation causing variables to cause variation to lcrim (and hence become important for prediction).

4.1.1.2 Which Subset to Select

Nonetheless, to decide which number of predictors would be the best choice, there are various techniques that can be followed to evaluate whether removing variables is worthwhile (SSE based or MSE based).

The results using based SSE measurements (Adjusted R^2, BIC, Cp) are shown in figure 5, and they show that generally the gain from dropping out predictors is optimal somewhere around 5-10 before becoming worse.

For the K Fold Cross Validation (10 fold picked here) also shown in figure 5, the best result occurs with 9 predictors which is similar with other results (and is identical to the adjusted Cp choice), and since this measurement is based on experimentation with tests errors it is preferred. The 10 Fold Cross Validation choice of 9 predictors would result in chas, rm, mdev and tax being removed from the model.

4.1.1.3 Selected Subset Coefficient Discussion

4.1.2 Ridge Regression

Ridge Regression uses a modified loss function to add a penalety to large coefficients to improve predictive solution (constraints variance).

4.1.2.1 Lambda Choice

4.1.2.2 Selected Lambda Coefficent Discussion

4.1.3 LASSO

LASSO is quite similar to Ridge Regression, however unlike ridge regression which always gets all p variables LASSO can subset (coefficients can become zero). This is because of the modified SS_E which takes the absolute value instead of the square of each coefficient.

4.1.3.1 Lambda Choice

4.1.3.2 Selected Lambda Coefficent Discussion

4.2 Model Evaluation Using k-fold Cross Validation

4.3 Best Model Discussion and Conclusions

5 Appendix

This section contains all supplementary material and is divided into three sections (Tables, Plots and Abbreviations). The code required to generate the supplementary material is also included

5.1 A.1 Abbrevations and Shorthands

5.1.1 Abbrevations

(X)DP: X Decimal Points

PCA: Principle Component Analysis

PC: Principle Component

SS_E: Residual Sum of Squares

MSE: Mean Square Error

R^2 or Rsq: Coefficient of Determination

C_p: Mallow's C_p

BIC: Bayesian Information Criterion

LASSO: Least Absolute Shrinkage and Selection Operator

5.1.2 Variable Shorthands

lcrim: Natural logarithm of the per capita crime rate by town.

zn: Proportion of residential land zoned for lots over 25,000 sq.ft.

indus: Proportion of non-retail business acres per town.

chas: Charles River dummy variable (=1 if tract bounds river; =0 otherwise).

nox: Nitrogen oxides concentration (parts per 10 million).

rm: Average number of rooms per dwelling.

age: Proportion of owner-occupied units built prior to 1940.

disf: A numerical vector representing an ordered categorical variable with four levels depending on the weighted mean of the distances to five Boston employment centres (=1 if distance < 2.5, =2 if 2.5 <= distance < 5, =3 if 5 <= distance < 7.5, =4 if distance >= 7.5).

rad: Index of accessibility to radial highways.

tax: Full-value property-tax rate per $10,000.

pratio: Pupil-teacher ratio by town.

black: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town.

lstat: Lower status of the population (percent).

medv: Median value of owner-occupied homes in $1000s.

Table 1:

	x
lcrim	-0.780
zn	11.364
indus	11.137
chas	0.069
nox	0.555
rm	6.285
age	31.425
disf	1.960
rad	9.549
tax	408.237
ptratio	18.456
black	356.674
lstat	12.653
medv	22.533

5.2 A.2 Tables

5.2.1 Code to Generate Table 1 (Transposed and Correct to 3DP)

```
table(colMeans(Boston), '')
```

Table 2: Correlation Matrix (3DP)

	lcrim	zn	indus	chas	nox	rm	age	disf	rad	tax	ptratio	black	lstat	medv
lcrim	1.000	-0.517	0.731	0.028	0.789	-0.307	-0.658	-0.683	0.853	0.828	0.390	-0.479	0.627	-0.454
zn	-0.517	1.000	-0.534	-0.043	-0.517	0.312	0.570	0.612	-0.312	-0.315	-0.392	0.176	-0.413	0.360
indus	0.731	-0.534	1.000	0.063	0.764	-0.392	-0.645	-0.727	0.595	0.721	0.383	-0.357	0.604	-0.484
chas	0.028	-0.043	0.063	1.000	0.091	0.091	-0.087	-0.082	-0.007	-0.036	-0.122	0.049	-0.054	0.175
nox	0.789	-0.517	0.764	0.091	1.000	-0.302	-0.731	-0.776	0.611	0.668	0.189	-0.380	0.591	-0.427
rm	-0.307	0.312	-0.392	0.091	-0.302	1.000	0.240	0.213	-0.210	-0.292	-0.356	0.128	-0.614	0.695
age	-0.658	0.570	-0.645	-0.087	-0.731	0.240	1.000	0.758	-0.456	-0.506	-0.262	0.274	-0.602	0.377
disf	-0.683	0.612	-0.727	-0.082	-0.776	0.213	0.758	1.000	-0.477	-0.537	-0.234	0.322	-0.511	0.291
rad	0.853	-0.312	0.595	-0.007	0.611	-0.210	-0.456	-0.477	1.000	0.910	0.465	-0.444	0.489	-0.382
tax	0.828	-0.315	0.721	-0.036	0.668	-0.292	-0.506	-0.537	0.910	1.000	0.461	-0.442	0.544	-0.469
ptratio	0.390	-0.392	0.383	-0.122	0.189	-0.356	-0.262	-0.234	0.465	0.461	1.000	-0.177	0.374	-0.508
black	-0.479	0.176	-0.357	0.049	-0.380	0.128	0.274	0.322	-0.444	-0.442	-0.177	1.000	-0.366	0.333
lstat	0.627	-0.413	0.604	-0.054	0.591	-0.614	-0.602	-0.511	0.489	0.544	0.374	-0.366	1.000	-0.738
medv	-0.454	0.360	-0.484	0.175	-0.427	0.695	0.377	0.291	-0.382	-0.469	-0.508	0.333	-0.738	1.000

5.2.2 Code to Generate Table 2 (Correct to 3DP)

```
table(cor(Boston), 'Correlation Matrix (3DP)')
```

Table 3: PCA Summary (Contribution to Variation)

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11	PC12	PC13	PC14
Standard deviation	2.645	1.286	1.118	0.934	0.927	0.809	0.638	0.588	0.500	0.460	0.434	0.389	0.322	0.233
Proportion of Variance	0.500	0.118	0.089	0.062	0.061	0.047	0.029	0.025	0.018	0.015	0.013	0.011	0.007	0.004
Cumulative Proportion	0.500	0.618	0.707	0.769	0.831	0.878	0.907	0.931	0.949	0.964	0.978	0.989	0.996	1.000

5.2.3 Code to Generate Table 3 (Correct to 3DP)

```
# Perform PCA based on the standardised data (means and data nature vary)
pca = prcomp(Boston, scale=TRUE)
table(summary(pca)$importance, 'PCA Summary (Contribution to Variation)')
```

5.2.4 Code to Generate Table 4 (Correct to 3DP)

```
# List PC 1, 2 and 3
table(pca$rotation[,1:3], 'PCA Components')
```

Table 4: PCA Components

	PC1	PC2	PC3
lcrim	0.341	-0.136	0.181
zn	-0.239	0.058	0.394
indus	0.324	-0.095	-0.070
chas	-0.001	-0.387	-0.255
nox	0.320	-0.227	-0.087
rm	-0.190	-0.492	0.285
age	-0.291	0.208	0.264
disf	-0.294	0.287	0.220
rad	0.295	-0.078	0.450
tax	0.315	-0.043	0.381
ptratio	0.196	0.331	0.116
black	-0.190	0.019	-0.378
lstat	0.297	0.238	-0.150
medv	-0.251	-0.475	0.095

5.2.5 Code to Generate Table 5

```
# fit model
bss = regsubsets(lcrim ~ ., data=Boston, method="exhaustive", nvmax= 13)

# Summarise
bssSummary = summary(bss)

tableTxt(bssSummary$outmat, "Best Subset Selection")
```

Table 5: Best Subset Selection

	zn	indus	chas	nox	rm	age	disf	rad	tax	ptratio	black	lstat	medv
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)	*	*	*	*	*	*	*	*	*	*	*	*	*

Table 6: Best Subset Coefficient Selection (9 Predictors)

	x
(Intercept)	-2.586
zn	-0.011
indus	0.015
nox	3.409
age	-0.005
disf	-0.145
rad	0.142
ptratio	-0.050
black	-0.001
lstat	0.028

```
# use coef function to find the coefficient
# id represents the choice for predictor number
table(coef(bss, id = 9), "Best Subset Coefficient Selection (9 Predictors)")
```

```
## Fit a ridge regression model with idea lambda (0.001)
ridgeFit = glmnet(as.matrix(Boston[-1]),
                  as.vector(Boston$lcrim), alpha=0, standardize=FALSE, lambda=0.001)
table(as.data.frame(as.matrix(coef(ridgeFit, s = 0.001))),
      "Ridge Regression Coefficients with Lambda = 0.001")
```

```
## Fit a LASSO regression model with idea lambda (0.001)
lassoFit = glmnet(as.matrix(Boston[-1]),
                  as.vector(Boston$lcrim), alpha=1, standardize=FALSE, lambda=0.001)
table(as.data.frame(as.matrix(coef(lassoFit, s = 0.001))),
      "LASSO Coefficients with Lambda = 0.001")
```

Table 7: Ridge Regression Coefficients with Lambda = 0.001

	1
(Intercept)	-2.227
zn	-0.011
indus	0.018
chas	-0.031
nox	3.064
rm	-0.056
age	-0.005
disf	-0.146
rad	0.146
tax	0.000
ptratio	-0.048
black	-0.001
lstat	0.031
medv	0.007

Table 8: LASSO Coefficients with Lambda = 0.001

	1
(Intercept)	-2.411
zn	-0.011
indus	0.017
chas	-0.016
nox	3.222
rm	-0.050
age	-0.005
disf	-0.136
rad	0.145
tax	0.000
ptratio	-0.046
black	-0.001
lstat	0.031
medv	0.007

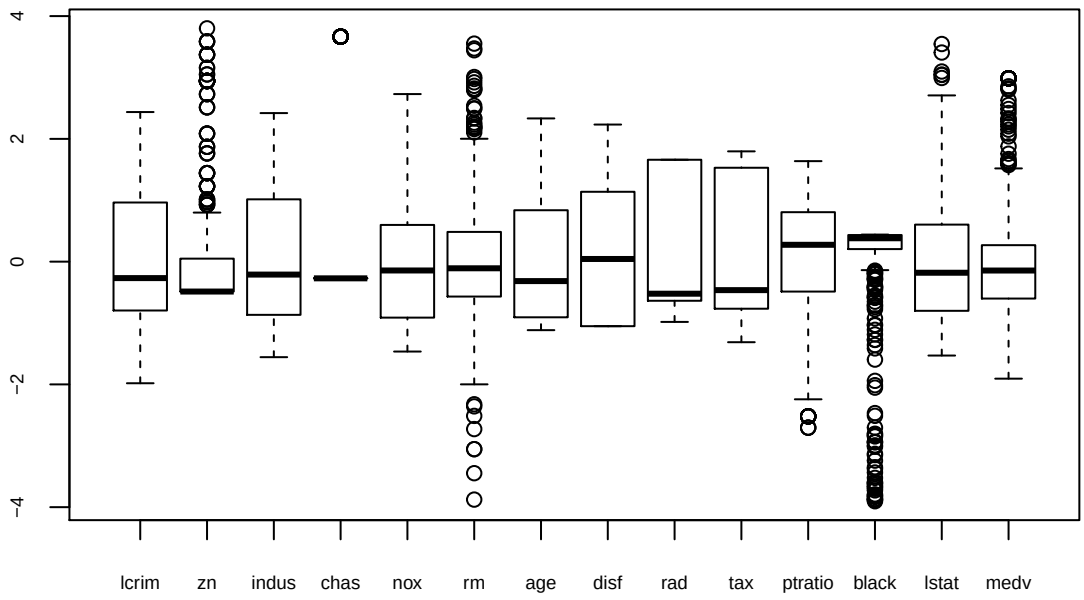

Figure 1: Box Plot

5.3 A.3 Plots

5.3.1 Code to Generate Figure 1

```
# scale transforms to deal with the variation in the nature of the measurements
boxplot(scale(Boston), cex.axis=0.6)
```

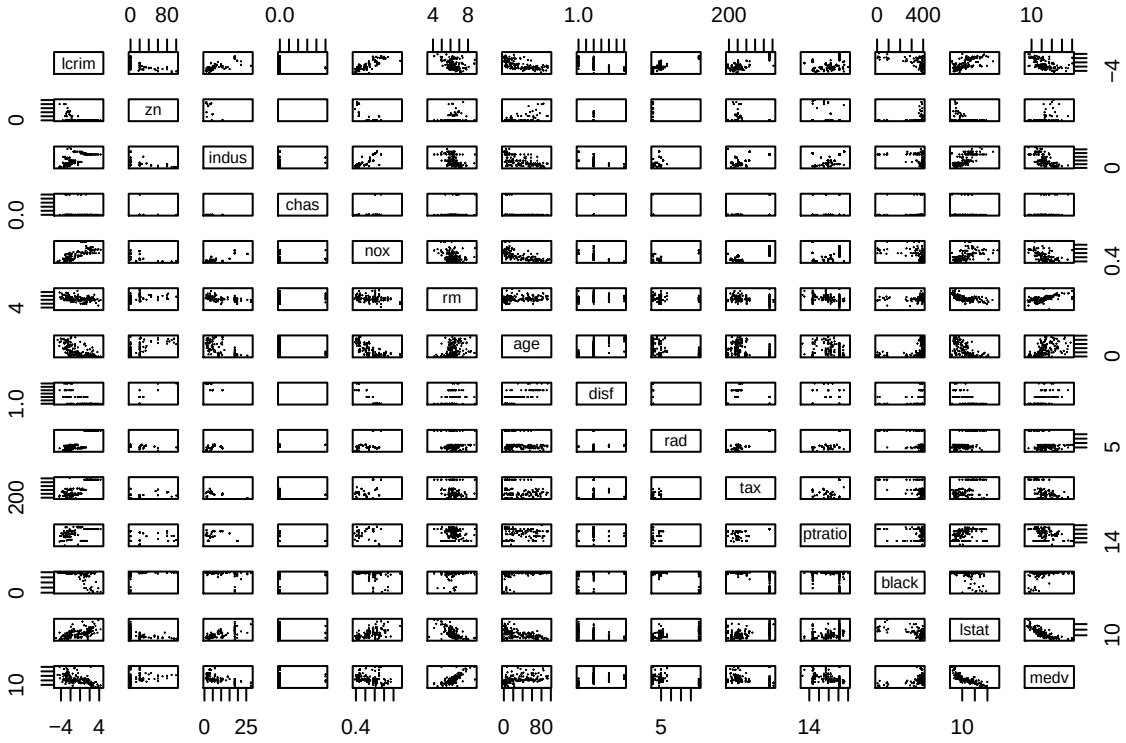

Figure 2: Pairs Plot

5.3.2 Code to Generate Figure 2

```
pairs(Boston, cex=0.0005)
```

5.3.3 Code to Generate Figure 3

```
plot(pca, type='l', main='Scree Plot for Boston Housing Values')
title(xlab='Principle Component number')
```

5.3.4 Code to Generate Figure 4

```
# Plot PCA 1 against PCA 2
plot(pca$x[,1], pca$x[,2], main ="Principle Component 1 vs 2 for Boston Housing Values",
     xlab="Component 1", ylab="Component 2")
```

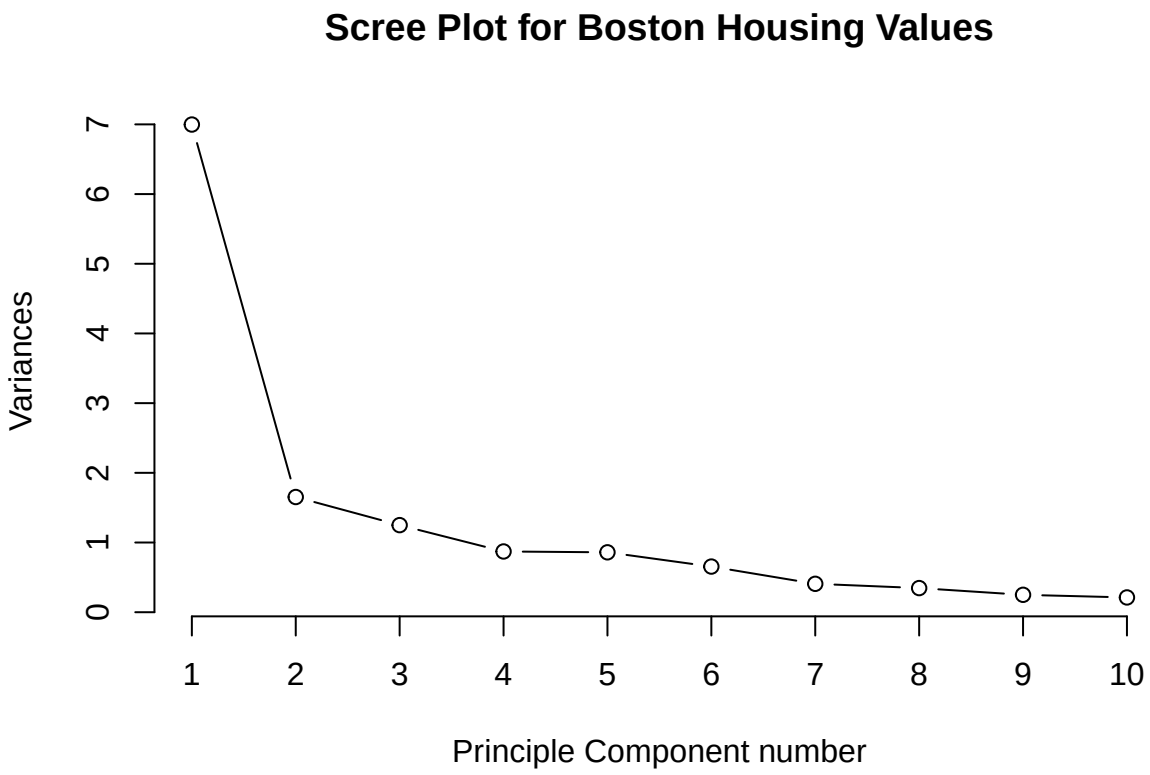

Figure 3: Scree Plot

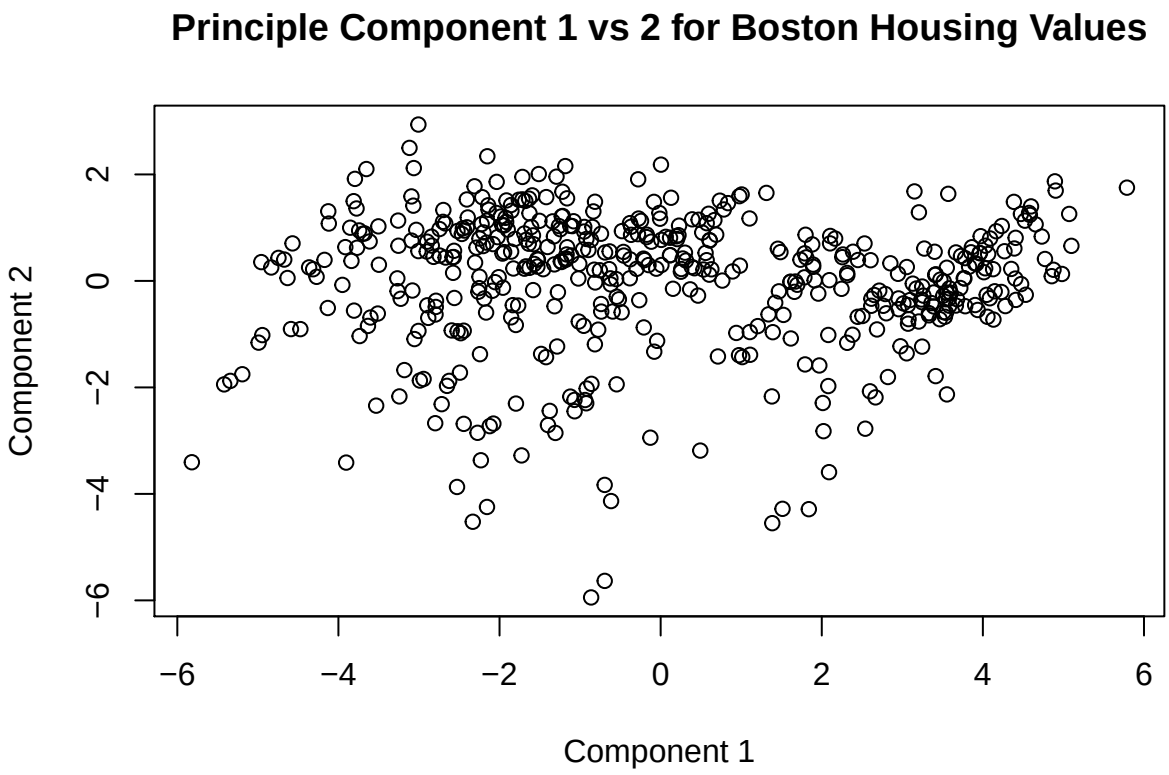

Figure 4: PCA Plot

5.3.5 Code to Generate Figure 5

```
# *** SSE Based Measurements ***

# Find best score for SSE based measurements (already done by bss summary)
bestAdjr2 = which.max(bssSummary$adjr2)
bestCp = which.min(bssSummary$cp)
bestBic = which.min(bssSummary$bic)

# *** 10 fold cross validation ***

# 10 fold cv

# Set the seed to make the analysis reproducible
set.seed(1)

# 10-fold cross validation
nFolds = 10

# Find n and p
p = ncol(Boston) - 1 # number of predictors (no lcrim)
n = nrow(Boston)

# Sample fold-assignment index
foldIndex = sample(nFolds, n, replace=TRUE)

# Hold fold sizes
foldSizes = numeric(nFolds)

# Compute fold sizes
for(k in 1:nFolds) foldSizes[k] = length(which(foldIndex==k))

# create the matrix to store the folds
cvBssErrors = matrix(NA, p, nFolds)

# Find MSEs for all fold combination
for(k in 1:nFolds) {
  # Fit models by best-subset selection (no k-th fold)
  bssTmpFit =
    regsubsets(lcrim ~ ., data=Boston[foldIndex!=k,], method="exhaustive", nvmax=p)

  # For each model M_m where m=1,...,p:
  for(m in 1:p) {
    # Compute fitted values for the k-th fold
    bssTmpPredict = predict(bssTmpFit, Boston[foldIndex==k,], m)
    # Work out MSE for the k-th fold
    cvBssErrors[m, k] = mean((Boston[foldIndex==k,]$lcrim - bssTmpPredict)^2)
  }
}

# Compute a weighted average MSE
bssMse = numeric(p)
# For models M_1,...,M_p:
```

```r
for(m in 1:p) {
  bssMse[m] = weighted.mean(cvBssErrors[m,], w=foldSizes)
}

# Identify model with the lowest MSE
bestCv = which.min(bssMse)

# *** Plotting ***

# Create multi-panel plotting device:
par(mfrow=c(2,2))

# Produce plots, highlighting optimal value of preductors:
plot(1:13, bssSummary$adjr2, xlab="Number of predictors", ylab="Adjusted Rsq",
type="b")
points(bestAdjr2, bssSummary$adjr2[bestAdjr2], col="red", pch=16)

plot(1:13, bssSummary$cp, xlab="Number of predictors", ylab="Cp", type="b")
points(bestCp, bssSummary$cp[bestCp], col="red", pch=16)

plot(1:13, bssSummary$bic, xlab="Number of predictors", ylab="BIC", type="b")
points(bestBic, bssSummary$bic[bestBic], col="red", pch=16)

plot(1:p, bssMse, xlab="Number of predictors", ylab="10-fold CV Error", type="b")
points(bestCv, bssMse[bestCv], col="red", pch=16)
```

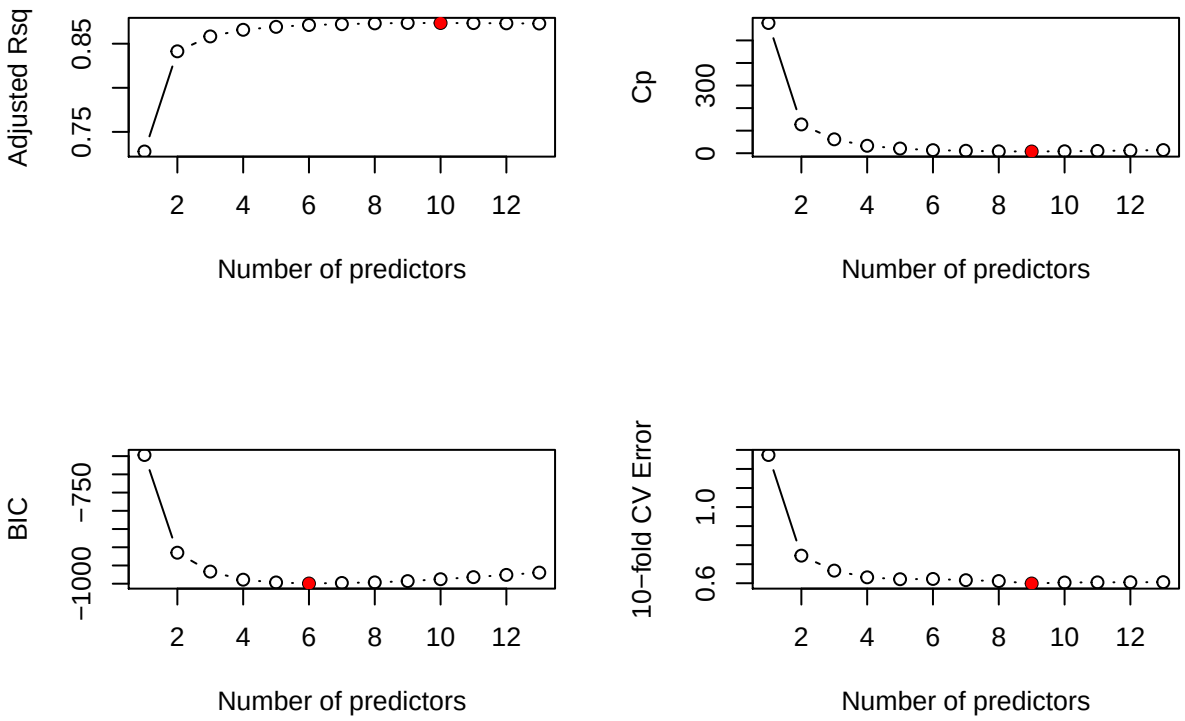

Figure 5: Best Subset Predictor Selection

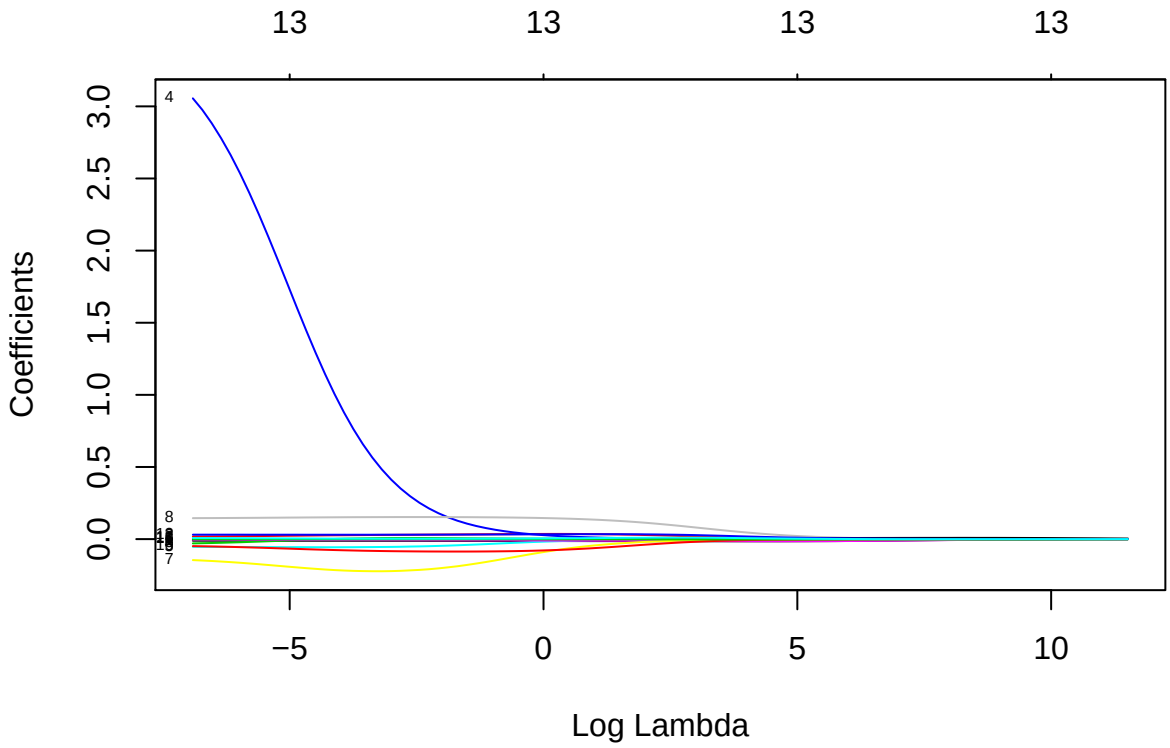

Figure 6: Ridge Fit Coefficents versus Lambdas

5.3.6 Code to Generate Figure 6

```
# grid of values for lambda (10^5 to 10^-3)
grid = 10^seq(5, -3, length=100)

## Fit a ridge regression model for each value of the tuning parameter
ridgeFit = glmnet(as.matrix(Boston[-1]),
                  as.vector(Boston$lcrim), alpha=0, standardize=FALSE, lambda=grid)

plot(ridgeFit, xvar="lambda", col=1:13, label=TRUE)
```

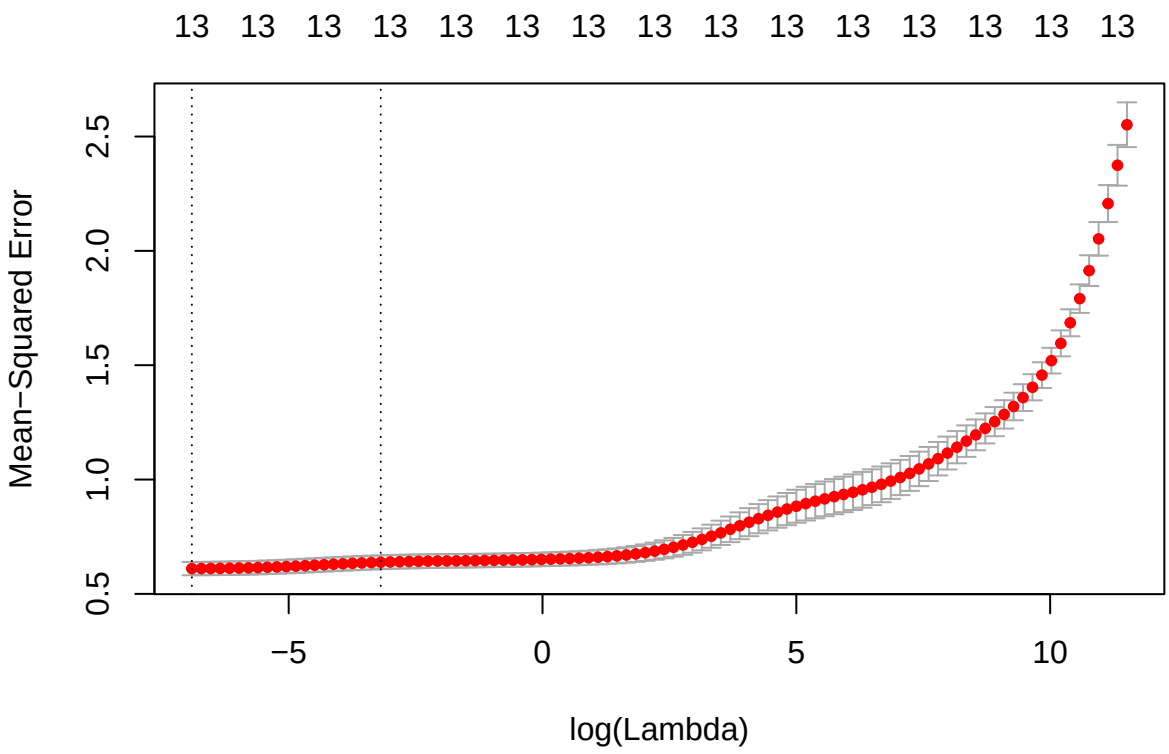

Figure 7: Ridge Fit 10 Fold Cross Validation

5.3.7 Code to Generate Figure 7 with Minimum Result

```
# Fit a ridge regression model using all lambdas with cv validation
# 0 -> Ridge regression not LASSO
ridgeFitCv = cv.glmnet(as.matrix(Boston[-1]),
                       as.vector(Boston$lcrim), alpha=0, standardize=FALSE, lambda=grid)

plot(ridgeFitCv) # plot cross validated error
```

```
# Find exact minimum
(lambdaMin = ridgeFitCv$lambda.min)
```

```
## [1] 0.001
```

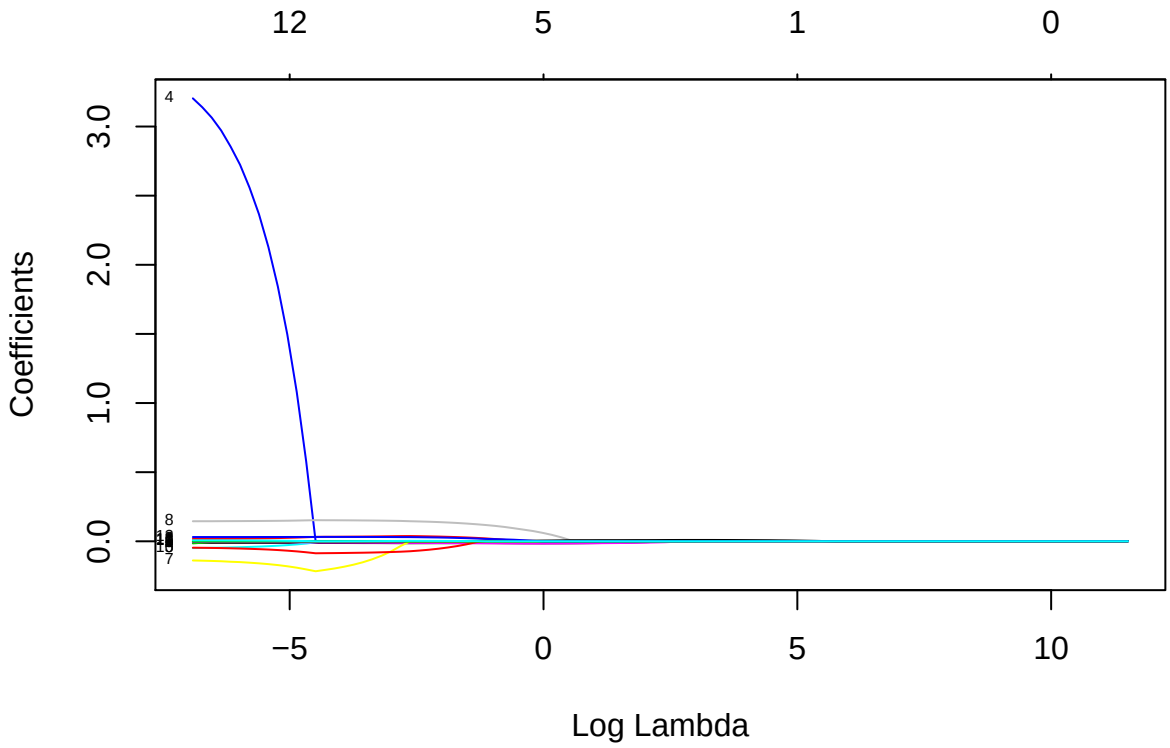

Figure 8: LASSO Coefficents versus Lambdas

5.3.8 Code to Generate Figure 8

```
# Fit a ridge regression model for each value of the tuning parameter
lasso = glmnet(as.matrix(Boston[-1]),
               as.vector(Boston$lcrim), alpha=1, standardize=FALSE, lambda=grid)

plot(lasso, xvar="lambda", col=1:13, label=TRUE)
```

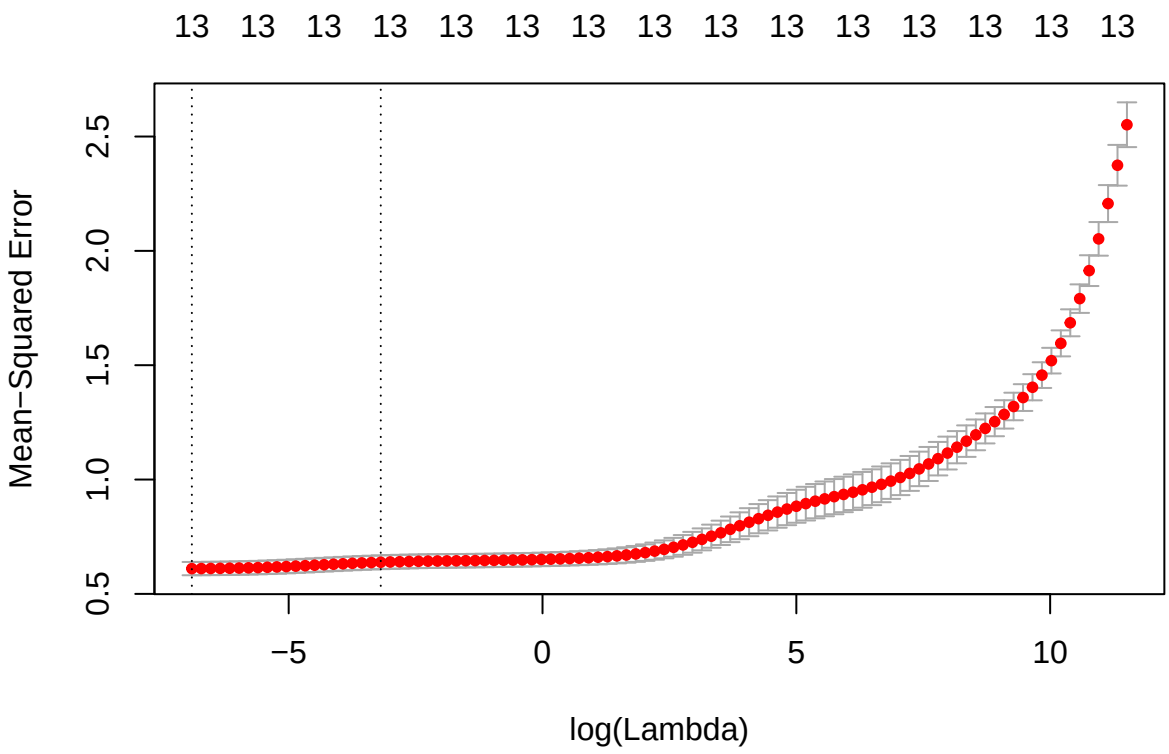

Figure 9: LASSO 10 Fold Cross Validation

5.3.9 Code to Generate Figure 9 with Minimum Result

```
# Fit a ridge regression model using all lambdas with cv validation
# 0 -> Ridge regression not LASSO
lassoCv = cv.glmnet(as.matrix(Boston[-1]),
                    as.vector(Boston$lcrim), alpha=1, standardize=FALSE, lambda=grid)

plot(ridgeFitCv) # plot cross validated error
```

```
# Find exact minimum
(lambdaMin = lassoCv$lambda.min)
```

```
## [1] 0.001
```