Modeling Median Home Values in Boston
Stat 151A

Michael Knopf

April 23, 2015

1 Introduction

In attempt to model the median value of owner-occupied homes in the Boston area, we will assume this quantity varies linearly with some or all of the variables listed in the following section, up to the presence of random, uncorrelated errors that normally distributed with mean zero and constant variance.

In our initial analysis, we will identify and attempt to remedy the challenges presented by the data. Mainly, these challenges consist of contradictions to these assumptions, but another issue that arises is skew. Next, we will apply various variable selection methods in order to choose the best model. The criteria we will use combine measures of accuracy and overfitting in order to produce the model that will best predict median home value for new obervations. Next, we will attempt to eliminate outlier observations from the model. Finally, we will perform diagnostics on the chosen model, again testing for contradictions of our initial assumptions and evaluating its overall performance.

Once the model has been produced and tested, we will attempt to interpret its results. Note that the model we seek throughout the majority of this process is the one that predicts best, though this will not necessarily be the one that explains best. The first model we will produce is meant to answer the simple question, "Given a new region with values x_i in each of some p categories, what is our best prediction for the median value of owner-occupied homes in that region?"

Other questions we have are about the factors that *cause* home values to be high in one area, but low in another. Which independent variables have the highest influence on home values? If city council is trying to raise the median home value in their town, what should be their target areas for improvement? Should they hire more teachers, or more police officers? Should they raise or lower taxes? Should they create incentives to build larger homes, or enact laws to improve the air quality? There is no doubt that most of these actions would have a positive effect on home values, but which produce the greatest outcome in proportion to their cost?

The model that does best at prediction will actually prove to be relatively ineffective in answering these sorts of questions. To find answers to these sorts of questions, we will need to investigate smaller models. In order to keep the length of this report to a minimum, I will mainly discuss the steps I took to produce the first model, the one tailored for prediction value. However, I will also share one smaller model I have produced, in order to show how it is more applicable to these sorts of inquiries.

2 Data

We will be working with the BostonHousing2 dataset in R. The data consists of 506 observations on 19 variables. *medv* is the the target variable. A dictionary of the variables is given below.

Variable	Description
town	name of town
tract	census tract
lon	longitude of census tract
lat	latitude of census tract
medv	median value of owner-occupied homes in USD 1000's
cmedv	corrected median value of owner-occupied homes in USD 1000's
crim	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	nitric oxides concentration (parts per 10 million)
rm	average number of rooms per dwelling
age	proportion of owner-occupied units built prior to 1940
dis	weighted distances to five Boston employment centres
rad	index of accessibility to radial highways
tax	full-value property-tax rate per USD 10,000
ptratio	pupil-teacher ratio by town
b	$1000(B - 0.63)^2$ where B is the proportion of blacks by town
lsat	percentage of lower status of the population

An initial glance at the data reveals that $medv = cmedv$ for all but 8 tracts, thus only one of these variables should be used (although the documentation seems to imply that *cmedv* is more current, we will use *medv* because the assignment specifically says to model this variable). Some of the most highly correlated variable pairs are (rad, tax), $(nox, indus)$, and (nox, dis), which have correlations of 0.910, .764, and $-.769$, respectively. Beyond these, many high correlations are present. This will be a constant challenge, as we run the risk of multicollinearity in our model.

3 Analysis

3.1 Preliminary Analysis of the Data

The dataset includes the variables *lon* and *lat*, which give the geographic position of each observed tract. The map in Figure 1 contains markers at the northernmost, westernmost, and easternmost tracts in the dataset (the southernmost point lies directly next to the easternmost point, Marshfield). The observations are more or less evenly spread throughout a radius around central Boston. It is intuitive that position *relative to the central city*, rather than absolute vertical and horizontal position, is the better indicator of home value, and further analysis has confirmed this fact.

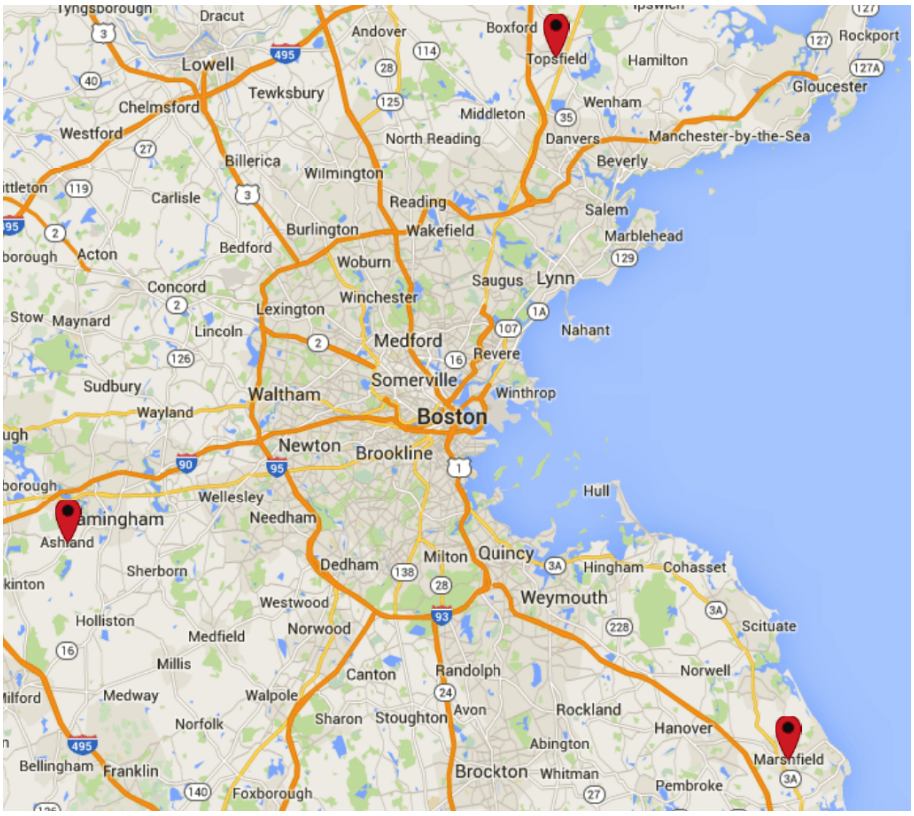

Figure 1: The three most extreme regions for which data were collected are plotted here: Topsfield, Marshfield, and Ashland.

Therefore, I have substited for these variables with two new ones, defined as

$$londist = |lat + 71.0589|$$
$$latdist = |lon - 42.3601|$$

since the heart of the city is located at $(-71.0589, 42.3601)$ (I have taken them in absolute value simply because they have proven to perform better this way).

Another aspect of the data deserving attention is skewed variables. Most variables have an acceptable degree of symmetry, though some do show a notable skew. In particular, the variable *crim* (crime rate) is heavily skew left. 65.6% of the values fall below 1%, and the frequency thins rapidly as values rise. I have considered variable transformations to fix this issue; in particular I have tested models that use $\log(crim)$ as a substitue. Figure 2 shows the effects of this transformation on the relationship between the variable and *medv*.

However, once this transformation has been performed, none of my variable selection methods prefer to keep the variable in the model. I have decided to leave this data as is. One justification for this is the argument that crime rates below 1% are effectively indistinguishable from each other. Areas with crime rates this low are seen as essentially safe; as we move into more dangerous areas, crime increases more than linearly, yet the value of owner-occupied homes *does* decrease proportionately with this non-linear rise.

I have fit *medv* against each explanatory variable separately. These plots are given in Figure 3. Most variables exhibit signs of a linear relationship, though some do not. Three variables I have looked at intensively are *dis*, *tax*, and *rad*.

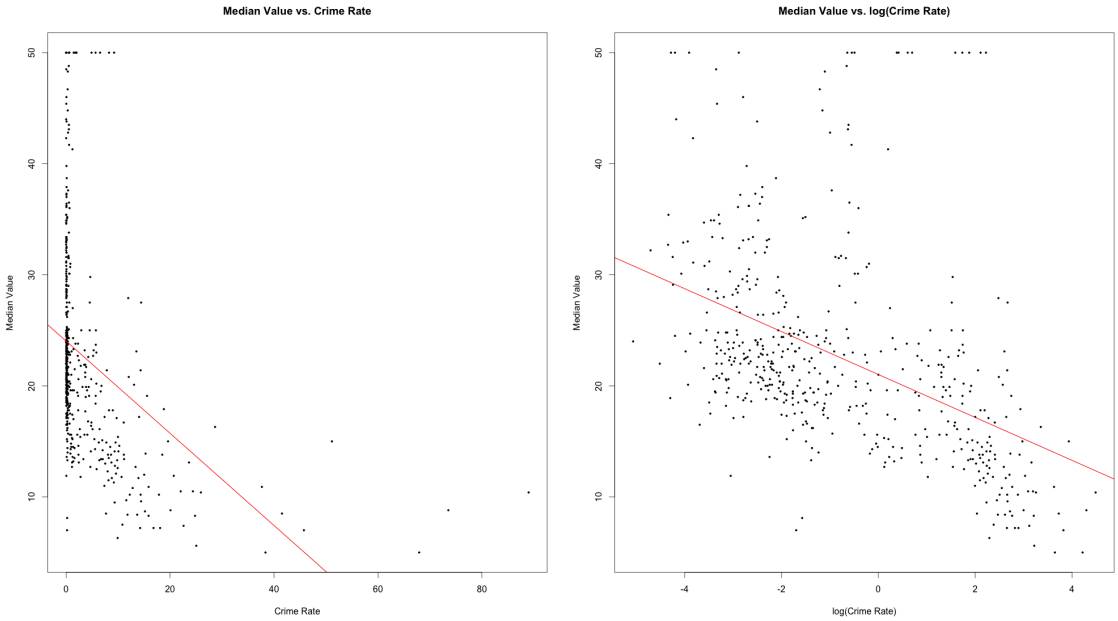

Figure 2: The effects of a logarithmic transformation on the left-skewed variable *crim*.

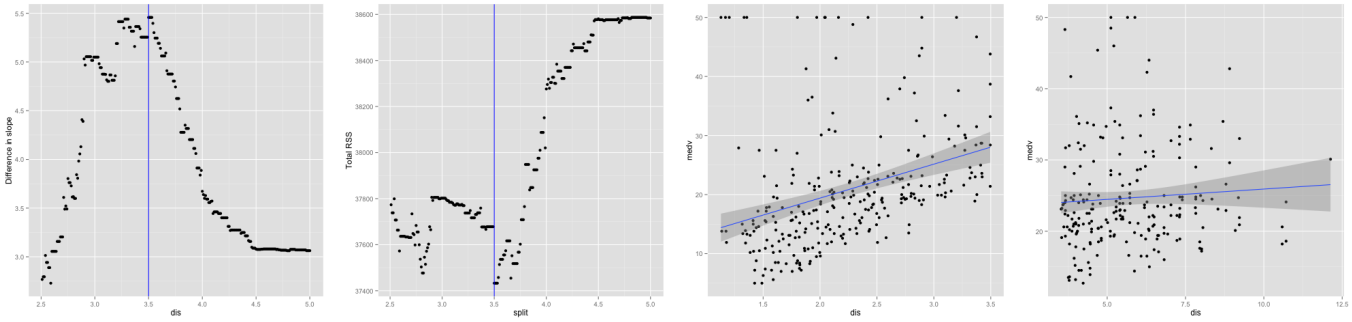

Figure 3: The left two plots show the effects of splitting the data at different values of *dis*. The right two plots are the corresponding simple linear fits after dividing the data based on whether $dis < 3.5$ (left) or $dis \geq 3.5$ (right).

It appears that *medv* relates differently to *dis* depending on whether or not $dis < 3.5$. For observations with $dis < 3.5$, the coefficient of *dis* in the simple linear regression model is significantly higher than those for which $dis > 3.5$. Figure 4 provides plots related to this issue. The left two plots were useful in identifying 3.5 as the natural value to split the data at. I tested 250 possible split values ranging between 2.5 and 5. For each of these values, I fit the data separately for observations above and below the split. The leftmost plot is the resulting difference in slope between the two models. The next plot is the sum of the squared residuals from both of the models. 3.5 was the value that resulted both in the greatest difference in slope and the lowest total RSS. The two right plots show the simple linear fits for data above and below this value. To reflect this issue, I added the interaction term $(dis < 3.5) \cdot dis$ to the model. Although it was significant in some smaller models, overall it did not seem to be useful, so I have eliminated it from the full model. I have done a similar analysis for the variables *tax* and *rad* (see Figure 5), however the results were similar and I have ultimately left them in the full model as they were.

Finally, I have made plots of the explanatory variables against one another, given in Figure

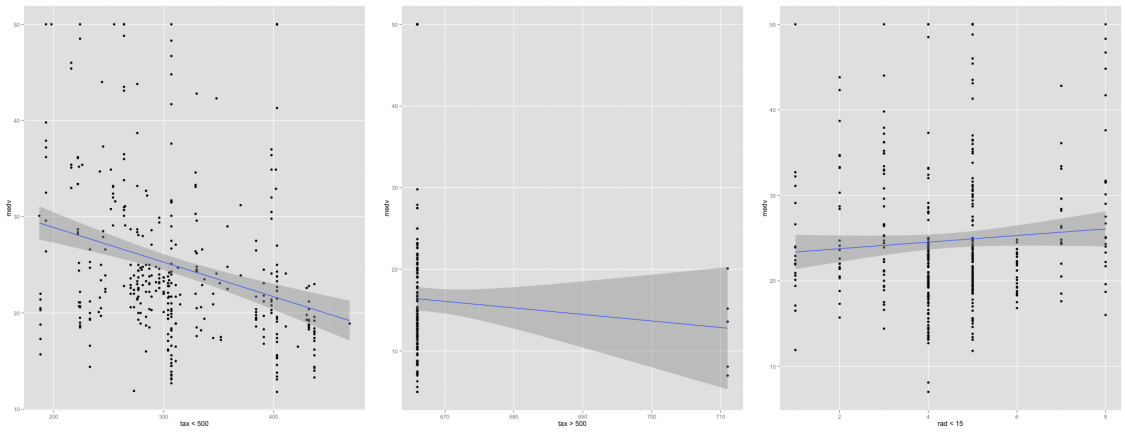

Figure 4: Simple linear regression against tax and rad using subsets of the data based on natural splits according to these variables.

6. Unfortunately, some of these plots show significant relationships among the chosen explanatory variables. This issue is discussed further in Section 4.2.

3.2 Variable Selection

The full model of study is

$$\begin{aligned}\text{MEDV} = &\text{LATDIST} + \text{LONDIST} + \text{CRIM} + \text{ZN} + \text{CHAS} + \text{NOX} + \text{RM} \\ &+ \text{DIS} + \text{RAD} + \text{TAX} + \text{PTRATIO} + \text{B} + \text{LSTAT}.\end{aligned}$$

I have run several variable selection procedures in order to select the best submodel. These procedures were

1. Backward selection using pValue as criteria.
2. Forward selection using pValue as criteria.
3. Backward selection using Mallow's C_p as criteria.
4. Backward selection using Adjusted R^2 as criteria.
5. The step function using AIC as criteria.
6. The step function using BIC as criteria.

The 4th method, using Adjusted R^2, produced the model

$$\begin{aligned}\text{MEDV} = &\text{LATDIST} + \text{LONDIST} + \text{CRIM} + \text{ZN} + \text{NOX} + \text{RM} \\ &+ \text{DIS} + \text{RAD} + \text{TAX} + \text{PTRATIO} + \text{B} + \text{LSTAT}\end{aligned}$$

Every other method produced the model

$$\begin{aligned}\text{MEDV} = &\text{LONDIST} + \text{CRIM} + \text{ZN} + \text{NOX} + \text{RM} + \text{DIS} \\ &+ \text{RAD} + \text{TAX} + \text{PTRATIO} + \text{B} + \text{LSTAT}.\end{aligned}$$

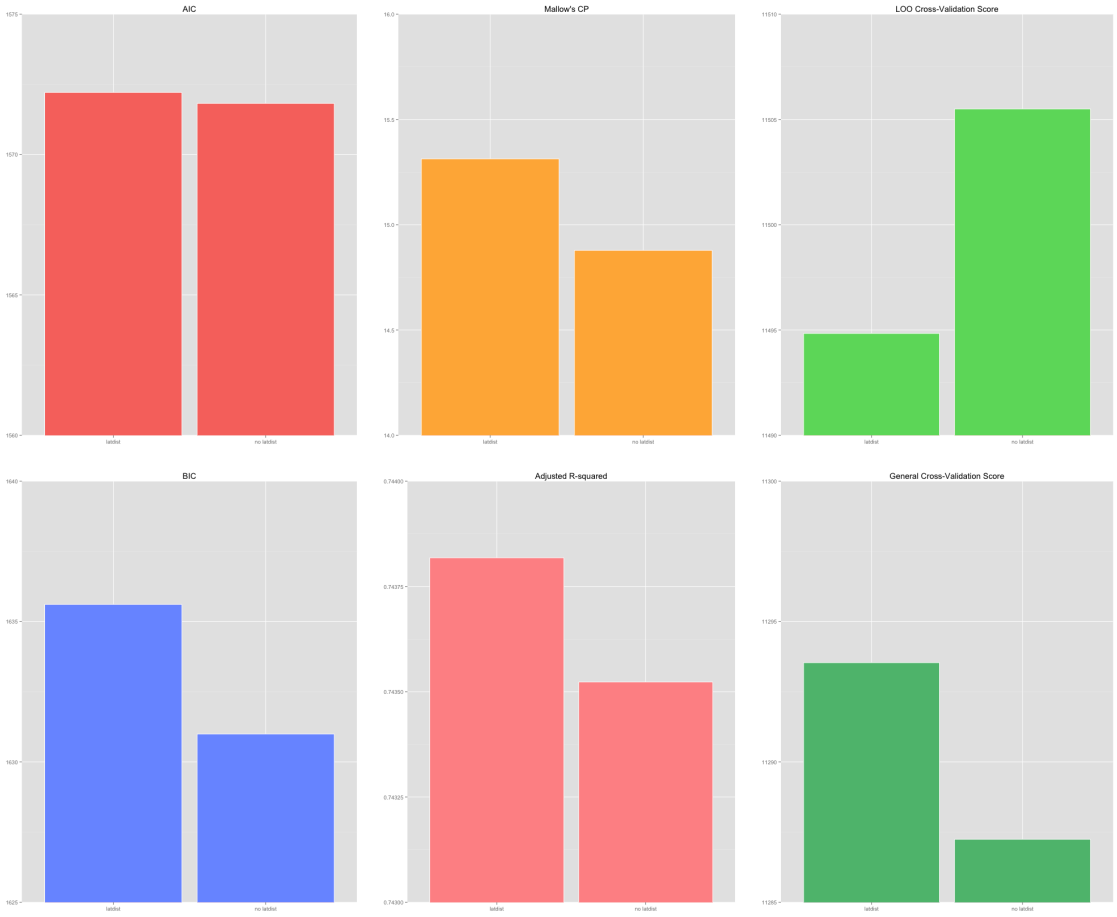

Figure 5: Model comparison based on selection criteria and leave-one-out cross-validation. The left model includes *latdist* while the right model does not.

which differs only by its exclusion of *latdist*.

Figure 3 displays the comparison of these two models based on the criteria used for selection, as well as their leave-one-out and general cross-validation scores. All criteria except for LOO cross-validation prefer the smaller model. Futhermore, *latdist* does not show any significant relationship with *medv* when considered in simple linear regression (see Figure 3). Overall, it does not appear that the predictive value of the smaller model is significantly less than that of the larger one, especially in light of the risk of overfitting. I have chosen the smaller model as my final fit.

3.3 Regression Diagnostics

I have used the diagnostic plots in Figure 9 to evaluate the assumptions of the linear model. The plot of the residuals against the fitted values shows hints of nonlinearity, since the scatter has a concave upward shape. However, the qq-plot of the residuals against the normal quantiles does show that the residuals are approximately normal, though somewhat right skewed. The plots of the residuals against each explanatory variables (as well as against the fitted values) does support the assumption of constant variance. Some of these plots appear to have a larger range on one side than the other, but this is due to skewed data rather than nonconstant variance (compare these plots with those in Figure 8 to see that these instances line up with those of uneven distributions of expanatory variables).

I have performed t-tests on each observation to determine whether its standardized predicted residual is significant. The resulting p-values are plotted in Figure 6, and I have identified and removed those that fall below the critical line of .05 (after applying the Bonferroni correction).

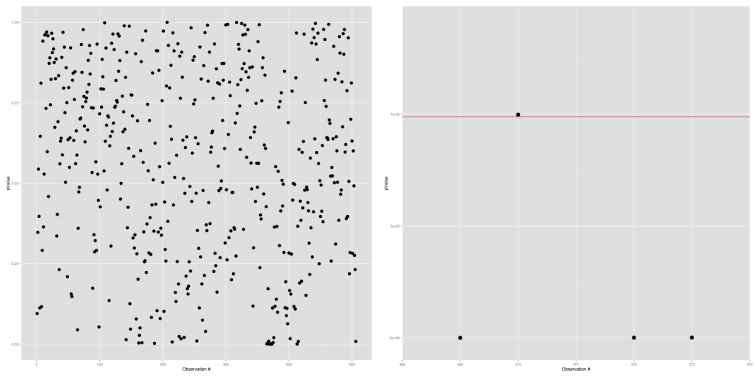

Figure 6: All p-values are on the left. Influential observations with the Bonferroni-corrected critical value are on the right.

4 Results and Conclusions

4.1 The Fit

```
Call:
lm(formula = medv ~ londist + crim + zn + chas + nox + rm + dis +
    rad + tax + ptratio + b + lstat, data = bost[-which(pValues <
    bonferroni), ])

Residuals:
     Min       1Q   Median       3Q      Max
-14.0883  -2.4031  -0.4924   1.9151  19.5268

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  32.696599   4.792329   6.823 2.64e-11 ***
londist      26.030654   6.866851   3.791 0.000169 ***
crim         -0.109809   0.029404  -3.734 0.000210 ***
zn            0.040032   0.012079   3.314 0.000988 ***
chas          1.891099   0.775444   2.439 0.015093 *
nox         -18.028122   3.194950  -5.643 2.84e-08 ***
rm            4.285432   0.373691  11.468  < 2e-16 ***
dis          -1.904177   0.231625  -8.221 1.82e-15 ***
rad           0.255340   0.056781   4.497 8.62e-06 ***
tax          -0.011343   0.003019  -3.757 0.000193 ***
ptratio      -0.988666   0.115556  -8.556  < 2e-16 ***
b             0.007434   0.002409   3.086 0.002146 **
lstat        -0.454664   0.043963 -10.342  < 2e-16 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

Residual standard error: 4.226 on 490 degrees of freedom
Multiple R-squared:  0.7836,  Adjusted R-squared:  0.7783
F-statistic: 147.8 on 12 and 490 DF,  p-value: < 2.2e-16
```

Every variable included in the model is significant, though *chas* is on the borderline. I believe this model is one of the best we could use for prediction purposes, up to performing advanced techniques that I am not to familiar with.

4.2 Interpretation

Now, we will try to apply the model's results to our questions. Suppose an area wishes to raise the median value of owner-occupied homes. It would want to perform better in those categories which would have the greatest impact on this quantity. The answer to this question lies in the coefficients. However, the coefficients alone give us no information, since they can only be interpreted with information about the spread of the corresponding explanatory variables.

A valid question to ask is, "If a tract is currently around the first quartile in some category, what effect would improving to the third quartile have on its median home value?" Let Q_i and q_i be the third and first quartiles of the ith explanatory variable, respectively, and define a statistic I_i, called "improvement" by

$$I_i = \beta_i(Q_i - q_i)$$

This statistic accounts for the spread of the data. One unit of a given quantity is completely arbitrary, but using quartiles gives us a way to compare the categories with each other. Thus, we might attempt to answer the question of which categories have the highest influence on median home value by ranking them according to the absolute value of this statistic, which is listed in the right column of the below table.

	coef	min	firstQuartile	median	thirdQuartile	max	improvement
dis	-1.904176556	1.13700	2.10525	3.21570	5.213250	12.1265	-5.9181807
rad	0.255340236	1.00000	4.00000	5.00000	24.000000	24.0000	5.1068047
lstat	-0.454663717	1.73000	6.97000	11.41000	17.025000	37.9700	-4.5716437
tax	-0.011342949	187.00000	279.00000	330.00000	666.000000	711.0000	-4.3897212
rm	4.285432181	3.56100	5.88750	6.20900	6.627000	8.7800	3.1690771
nox	-18.028122291	0.38500	0.44900	0.53800	0.624000	0.8710	-3.1549214
ptratio	-0.988665826	12.60000	17.35000	19.00000	20.200000	22.0000	-2.8176976
londist	26.030653716	0.00010	0.01610	0.03860	0.080900	0.2489	1.6867864
zn	0.040031692	0.00000	0.00000	0.00000	12.500000	100.0000	0.5003962
crim	-0.109809254	0.00632	0.08193	0.25356	3.551845	88.9762	-0.3810288
b	0.007434401	0.32000	375.95500	391.50000	396.235000	396.9000	0.1507696
chas	1.891098661	1.00000	1.00000	1.00000	1.000000	2.0000	0.0000000
(Intercept)	32.696599020	NA	NA	NA	NA	NA	NA
medv	NA	5.00000	16.90000	21.20000	25.000000	50.0000	NA

The difficulty with this interpretation is rooted in multicollinearity; many of the explanatory variables exhibit significant relationships with each other. The best example of this is the variable *dis*, which has a negative coefficient in the full model but a positive coefficient in the simple linear model $MEDV \sim DIS$. In the full model, *dis* is being interpreted in the presence of other variables which it has a linear relationship with; it has a positive relationship with *londist* and *zn*, but a negative relationship with *nox* and *age*. We can no longer interpret the effects of increasing *dis* in this way, since it assumes we hold all other variables constant. The truth is that any reasonable increase in *dis* would necessitate a response in these four other variables that we are not taking into account.

It is possible to account for these relationships and interpret this model, though the results would be too complicated to comprehend. For instance, we might look at the the effect on *medv* of moving from the third to first quartile in *dis* while *also* moving the corresponding amount in each category that *dis* has a relationship with. This would amount to looking at the quantity

$$\sum_{x \in \text{vars}} \beta_{x \sim dis} \beta_{dis}(Q_{dis} - q_{dis}) = I_{dis} \sum_{x \in \text{vars}} \beta_{x \sim dis}$$

where β_{dis} is the coefficient of *dis* in our model and $\beta_{x \sim dis}$ is the coefficient of x in a simple linear regression of x against *dis*. This expression returns a positive value, while earlier the percieved impact of *dis* was negative. So even the signs of the coefficients in this model cannot be trusted.

Of course, this formula does not account for other relationships between explanatory variables, and so is also not valid. To take them all into account simultaneously would be mathematically complicated, but even more problematic is that the resulting value would be impossible to decide which variable should be viewed as the independent cause of the increase.

This model is built for prediction. It is too large and inclusive to be used effectively for interpretation. If we are interested in understanding the relationship between each variable and the target variable, we would want to build a smaller model. Otherwise, we are forced to interpret the variables in the presence of each other, which becomes increasingly complicated as the model grows and multicollinearity arises.

Furthermore, this model includes variables that could not even be interpreted in a simple linear regression. For instance, what is b? A measure of the population's deviation from 63% black? Does this have any meaning at all? Possibly all it does is add a degree of freedom to help the model fit a plane to the data, while in truth it has no reasonable relation to *medv* whatsoever.

I have used an alternative variable selection method to produce a smaller model. First, I eliminated variables that are not useful in explaining *medv*. Specifically, these were b and $lstat$. Clearly, having many lower status people in the population correlates with low property values, however a lower status population is likely caused by the fact that those people can afford to live there, i.e. the median home value is low.

Next, I worked backwards from the full model, eliminating variables that did not satisfy a criteria that measures a variable's desirability based on its significance in the model and its correlation with the other explanatory variables present. After doing this several times, using various cutoff values, the model eventually produced was

$$MEDV = RM + PTRATIO + CRIM$$

A summary is given below.

```
> summary(ftDropOutliers)

Call:
lm(formula = medv ~ rm + ptratio + crim, data = bost[-which(pValues <
    bonferroni), ])

Residuals:
     Min       1Q   Median       3Q      Max
-18.2124  -2.7641   0.1068   2.5272  14.9498

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -6.22829    3.29065  -1.893    0.059 .
rm           8.00451    0.33196  24.113  < 2e-16 ***
ptratio     -1.14914    0.10668 -10.772  < 2e-16 ***
crim        -0.20903    0.02566  -8.147 3.06e-15 ***
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1         1

Residual standard error: 4.7 on 495 degrees of freedom
Multiple R-squared:  0.7187,  Adjusted R-squared:  0.717
F-statistic: 421.5 on 3 and 495 DF,  p-value: < 2.2e-16


> sink()
```

	coef	min	firstQuartile	median	thirdQuartile	max	improvement
rm	8.0045079	4.13800	5.888000	6.2090	6.622000	8.7800	5.8753088
ptratio	-1.1491436	12.60000	17.150000	19.0000	20.200000	22.0000	-3.5048880
crim	-0.2090281	0.00632	0.081005	0.2498	3.242325	88.9762	-0.6608047
(Intercept)	-6.2282944	NA	NA	NA	NA	NA	NA
medv	NA	5.00000	16.800000	21.1000	25.000000	50.0000	NA

Using this smaller model, interpretation becomes realistic. Notice that the signs of the coefficients actually align with how our intuition says those variables should affect property value. This is because the variables present are relatively uncorrelated ($\text{cor}(crim, rm) = -0.2176540$, $\text{cor}(crim, ptratio) = 0.2867482$, $\text{cor}(rm, ptratio) = -0.3552455$). Therefore, we can actually talk about the effects of improvement in one of these categories on median home value, without getting lost in the corresponding effects on the other categories.

It appears that the most important factor is the number of rooms per dwelling, followed by the ratio of pupils to teachers. Crime rate is relatively less important. Having big houses with many rooms is probably something that only the home-owners themselves can control, but at least we could tell city council to spend its money on more teachers, not on a bigger police force.

5 Appendix

5.1 Plots

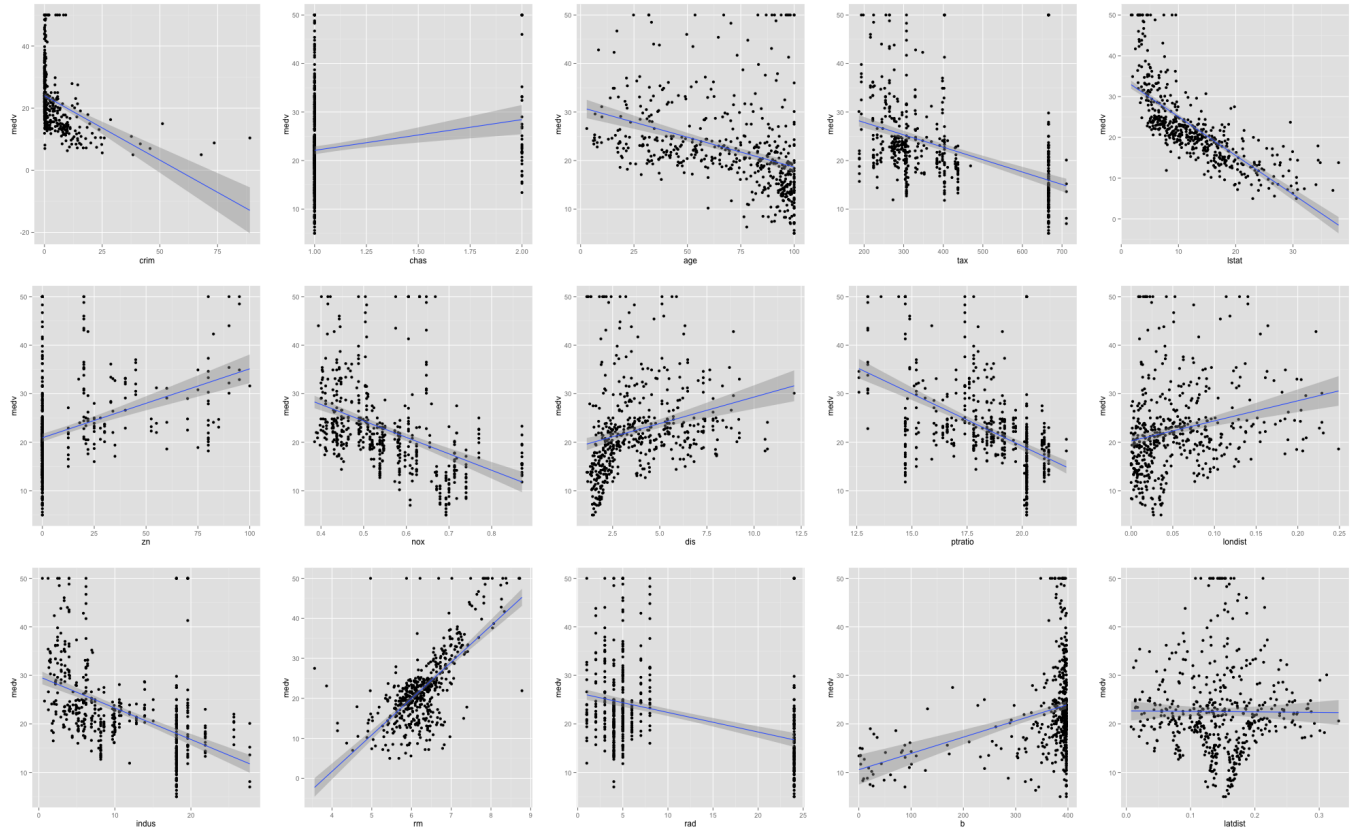

Figure 7: Median value of owner occupied homes plotted against and fit to each individual explanatory variable.

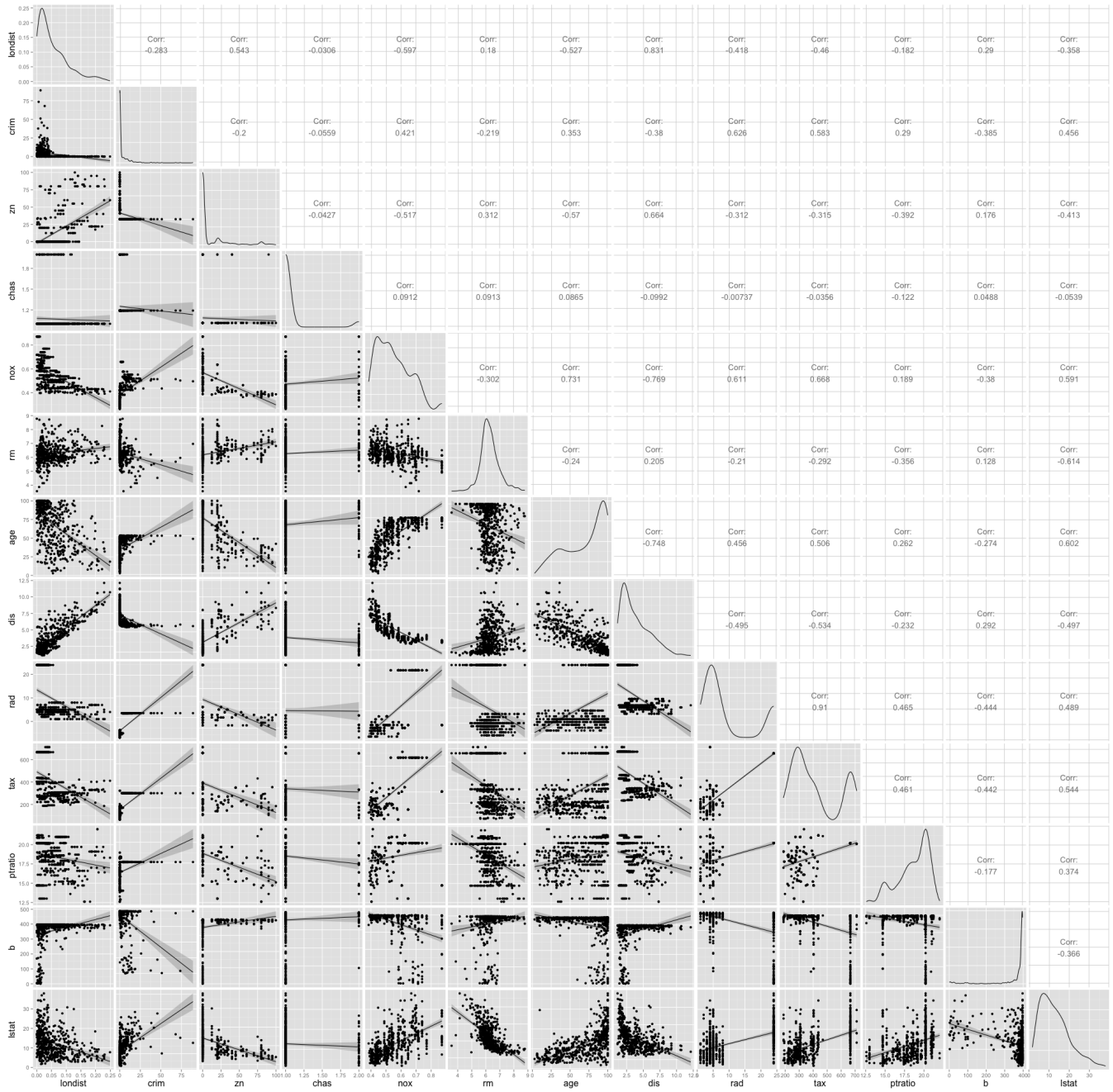

Figure 8: Pairwise plots of explanatory variables against one another (only those used in the final model) along with lines of best fit are below the diagonal. Distribution of explanatory variabales are on diagonal. Correlations are above diagonal.

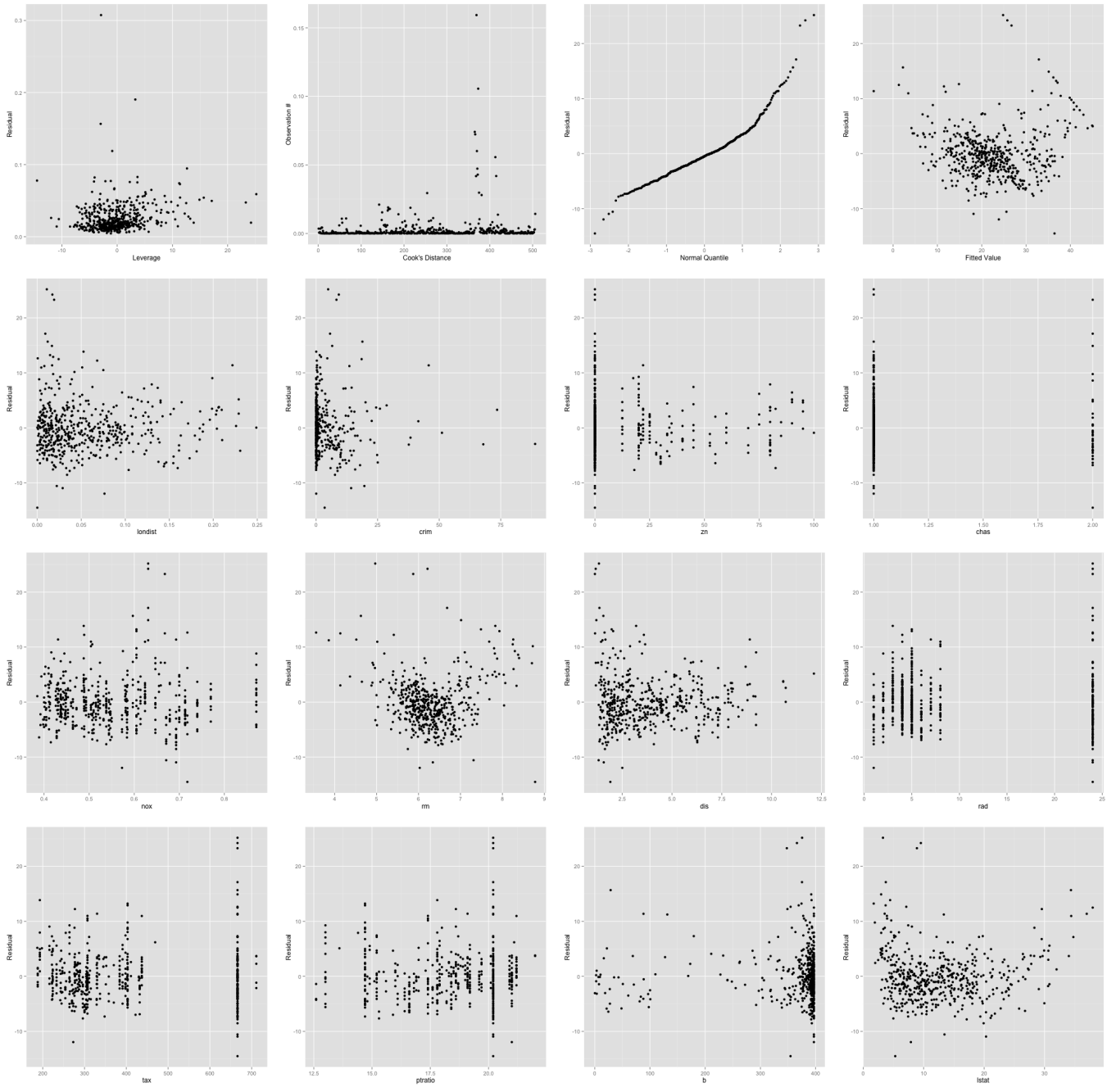

Figure 9: Diagnostic plots for testing the assumptions of the linear model.

5.2 Code

5.2.1 Data Analysis

```
library(mlbench)
library(ggplot2)
library(plyr)
library(reshape2)
library(grid)
library(lattice)
library(GGally)
data(BostonHousing2)

# Initialize dataframe and matrix
bost = BostonHousing2
rm(BostonHousing2)
for(i in 2:ncol(bost)) {
  bost[,i] = as.numeric(bost[,i])
}
bost$chas = as.numeric(bost$chas)
bostmat = as.matrix(bost[,-(1:2)])


# How many tracts differ in these categories?  Just 8.
sum(bost$medv - bost$cmedv != 0)
bost$town[bost$medv - bost$cmedv != 0]

# Check for highly correlated variables
cormat = cor(bostmat)
which(abs(cormat) > .75 & cormat < 1, arr.ind = TRUE)

# If lon and lat are modified to measure proximity from Boston's center,
# is a better model produced?
bost$lon = bost$lon - (-71.0589)
bost$lat = bost$lat - 42.3601
bost$londist = abs(bost$lon)
bost$latdist = abs(bost$lat)


# Look for skewed data and non-linear relationships by fitting medv to each individual explanatory
    variable
plots.simple = list()
for (i in 1:length(names(bost)[-(1:6)])) {
  var = (names(bost)[-(1:6)])[i]
  plots.simple[[i]] = qplot(x=bost[,var], y=medv, data = bost, geom = c("point", "smooth"), method
      = "lm", xlab = var)
}
plots.simple = list()
plots.simple[[1]] = qplot(x=crim, y=medv, data = bost, geom = c("point", "smooth"), method = "lm",
    xlab = 'crim')
plots.simple[[2]] = qplot(x=zn, y=medv, data = bost, geom = c("point", "smooth"), method = "lm",
    xlab = 'zn')
plots.simple[[3]] = qplot(x=indus, y=medv, data = bost, geom = c("point", "smooth"), method = "lm",
     xlab = 'indus')
plots.simple[[4]] = qplot(x=chas, y=medv, data = bost, geom = c("point", "smooth"), method = "lm",
    xlab = 'chas')
plots.simple[[5]] = qplot(x=nox, y=medv, data = bost, geom = c("point", "smooth"), method = "lm",
    xlab = 'nox')
plots.simple[[6]] = qplot(x=rm, y=medv, data = bost, geom = c("point", "smooth"), method = "lm",
    xlab = 'rm')
plots.simple[[7]] = qplot(x=age, y=medv, data = bost, geom = c("point", "smooth"), method = "lm",
    xlab = 'age')
plots.simple[[8]] = qplot(x=dis, y=medv, data = bost, geom = c("point", "smooth"), method = "lm",
    xlab = 'dis')
plots.simple[[9]] = qplot(x=rad, y=medv, data = bost, geom = c("point", "smooth"), method = "lm",
    xlab = 'rad')
plots.simple[[10]] = qplot(x=tax, y=medv, data = bost, geom = c("point", "smooth"), method = "lm",
    xlab = 'tax')
plots.simple[[11]] = qplot(x=ptratio, y=medv, data = bost, geom = c("point", "smooth"), method = "
    lm", xlab = 'ptratio')
```

```
plots.simple[[12]] = qplot(x=b, y=medv, data = bost, geom = c("point", "smooth"), method = "lm",
    xlab = 'b')
plots.simple[[13]] = qplot(x=lstat, y=medv, data = bost, geom = c("point", "smooth"), method = "lm"
    , xlab = 'lstat')
plots.simple[[14]] = qplot(x=londist, y=medv, data = bost, geom = c("point", "smooth"), method = "
    lm", xlab = 'londist')
plots.simple[[15]] = qplot(x=latdist, y=medv, data = bost, geom = c("point", "smooth"), method = "
    lm", xlab = 'latdist')

multiplot(plotlist = plots.simple, cols = 5)


# Are tax and rad categorical?  Perform simple linear regression on subsets where
# tax is above/below cutoff line.
bost.taxlow = bost[which(bost$tax < 500),c('medv','tax')]
taxlow = qplot(x=tax, y=medv, data = bost.taxlow, geom = c("point", "smooth"), method = "lm", xlab
    = 'tax < 500')
bost.taxhigh = bost[which(bost$tax >= 500),c('medv','tax')]
taxhigh = qplot(x=tax, y=medv, data = bost.taxhigh, geom = c("point", "smooth"), method = "lm",
    xlab = 'tax > 500')

bost.radlow = bost[which(bost$rad < 15),c('medv','rad')]
radlow = qplot(x=rad, y=medv, data = bost.radlow, geom = c("point", "smooth"), method = "lm", xlab
    = 'rad < 15')
bost.radhigh = bost[which(bost$rad >= 15),c('medv','rad')]
qplot(x=rad, y=medv, data = bost.radhigh, geom = c("point", "smooth"), method = "lm", xlab = 'rad >
     15')
mean(bost.radhigh$medv)

multiplot(plotlist = list(taxlow, taxhigh, radlow), cols = 3)

bost$rad.high = bost$rad > 15

ft1 = lm(medv ~ latdist + londist + crim + zn + indus + chas + nox +
           rm + age+ dis+ rad + tax + ptratio + b + lstat,
         data = bost)

ft2 = lm(medv ~ latdist + londist + crim + zn + indus + chas + nox +
           rm + age+ dis+ rad.high + tax + ptratio + b + lstat,
         data = bost)



# Is dis categorical?
qplot(x=dis, y=medv, data = bost, geom = c("point", "smooth"), method = "lm", xlab = 'dis')

# Test different split values.  Obtain total RSS and difference of slopes for each split.
diff = sapply(1:250, function(i) {
  split = 2.5 + i/100
  ft.dislow = lm(medv ~ dis, data = bost[which(bost$dis < split),])
  ft.dishigh = lm(medv ~ dis, data = bost[which(bost$dis > split),])
  return(ft.dislow$coefficients[2] - ft.dishigh$coefficients[2])
})

rss = sapply(1:250, function(i) {
  split = 2.5 + i/100
  ft.dislow = lm(medv ~ dis, data = bost[which(bost$dis < split),])
  ft.dishigh = lm(medv ~ dis, data = bost[which(bost$dis > split),])
  return(deviance(ft.dislow) + deviance(ft.dishigh))
})

# Set split value and plot resulting fits
splitvalue = min(2.5 + which(rss == min(rss))/100)
plots = list()
plots[[1]] = qplot(2.5 + (1:250)/100, diff, xlab = "dis", ylab = "Difference in slope") + geom_
    vline(x = 3.5, col = 'blue')
plots[[2]] = qplot(2.5 + (1:250)/100, rss, xlab = 'split', ylab = "Total RSS") + geom_vline(x =
    3.5, col = 'blue')
plots[[3]] = qplot(x=dis, y=medv, data = bost[which(bost$dis < splitvalue),], geom = c("point", "
    smooth"), method = "lm", xlab = 'dis')
plots[[4]] = qplot(x=dis, y=medv, data = bost[which(bost$dis >= splitvalue),], geom = c("point", "
    smooth"), method = "lm", xlab = 'dis')
```

```
multiplot(plotlist = plots, cols = 4)

# Check summaries of split models
ft.dislow = lm(medv ~ dis, data = bost[which(bost$dis < splitvalue),])
ft.dishigh = lm(medv ~ dis, data = bost[which(bost$dis >= splitvalue),])

# Create interaction term and check summary in new full model
bost$dislow = bost$dis * (bost$dis < splitvalue)

summary(lm(medv ~ latdist + londist + crim + zn + indus + chas + nox +
          rm + age + dis + dislow + rad + tax + ptratio + b + lstat,
        data = bost))
summary(lm(medv ~ dis + dislow, data = bost))

# Create pairwise plots of explanatory variables
ggpairs(bost[,vars], lower=list(continuous="smooth"))
```

5.2.2 Variable Selection

```
allvars = c('medv', 'latdist', 'londist', 'crim', 'zn', 'indus', 'chas', 'nox',
            'rm', 'age', 'dis', 'rad', 'tax', 'ptratio', 'b', 'lstat')

# Initialize full model
fullmodel = lm(medv ~ latdist + londist + crim + zn + indus + chas + nox +
                 rm + age + dis + rad + tax + ptratio + b + lstat,
               data = bost)

# Compute parameters used in variable selection
sigma = summary(fullmodel)$sigma
n = nrow(bost)
p = length(fullmodel$coefficients) - 1
summary(fullmodel)

getMallows = function(model) {
  p = length(model$coefficients)
  (deviance(model) / sigma^2) - (n-2-2*p)
}

getAIC = function(model) {
  n*log(deviance(model) / n) + 2*(1+length(model$coefficients))
}

getBIC = function(model) {
  n*log(deviance(model) / n) + log(n)*(1+length(model$coefficients))
}

getCVscore = function(model) {
  sum(model$residuals^2/ (1-hatvalues(model))^2)
}

getGCVscore = function(model) {
  p = length(model$coefficients)
  deviance(model)*(1 + 2*(1+p)/n)
}

# Backward selection based on p-value
backward = function(vars = allvars, critical = .15) {
  model = lm(medv ~ ., bost[,vars])
  pvalues = summary(model)$coefficients[-1,4]
  var = which(pvalues == max(pvalues))
  if(pvalues[var] > critical) {
    vars = vars[-(var+1)]
    return(backward(vars, critical))
  } else {
    return(model)
  }
}
```

```
# Forward selection based on p-value
forward = function(vars = 1, critical = .15) {
  if (length(vars) == length(allvars))
  {
    return(lm(medv ~ ., bost[,allvars]))
  }

  pvalues = sapply(1:length(allvars), function(var) {
    if(var %in% vars) {
      return(1)
    }
    model = lm(medv ~ ., bost[,allvars[c(vars,var)]])
    p = rev(summary(model)$coefficients[,4])[1]
    return(p[length(p)])
  })
  var = which(pvalues == min(pvalues))
  if(pvalues[var] < critical) {
    vars = c(vars,var)
    return(forward(vars = vars, critical = critical))
  } else {
    return(lm(medv ~ ., bost[, allvars[vars]]))
  }
}

# Backward selection based on adjusted R squared
adjRsqr = function(vars = allvars) {
  ft = lm(medv ~., bost[,vars])
  currentAdjR = summary(ft)$adj.r.squared
  adjR = sapply(vars[-1], function(var) {
    model = lm(medv ~., bost[,vars[vars != var]])
    adjR = summary(model)$adj.r.squared
    return(adjR)
  })
  var = which(adjR == max(adjR))
  if(adjR[var] > currentAdjR)
  {
    return(adjRsqr(vars[-(var+1)]))
  } else {
    return(ft)
  }
}

# AIC selection using step function
AIC = function() {
  model = lm(medv ~ ., bost[,allvars])
  return(step(model, direction = "both", trace = 0))
}

# BIC selection using step function
BIC = function() {
  model = lm(medv ~ ., bost[,allvars])
  return(step(model, direction="both", k = log(nrow(bost)), trace = 0))
}

# Backward selection using Mallow's Cp
Mallow = function(vars = allvars) {
  ft = lm(medv ~ ., bost[,vars])
  currentMallows = getMallows(ft)

  p = p-1

  mallows = sapply(vars[-1], function(var) {
    model = lm(medv ~., bost[,vars[vars != var]])
    return(getMallows(model))
  })
  var = which(mallows == min(mallows))
  if(mallows[var] < currentMallows)
  {
    return(Mallow(vars[-(var+1)]))
  } else {
    return(ft)
  }
```

```
}

# Get model selected by each method
models = list()
models[[1]] = backward()
models[[2]] = forward()
models[[3]] = adjRsqr()
models[[4]] = AIC()
models[[5]] = BIC()
models[[6]] = Mallow()

# Inspect summaries
for (ft in models) print(summary(ft))

# Alphabetize coefficients to compare models
coefs = lapply(models, function(ft) names(ft$coefficients)[-1])
coefs = sapply(coefs, function(x) x = x[order(x)])
print(coefs)

# Only two unique models were produced
ft = list()
ft[[1]] = models[[3]]  # This model also includes latdist
ft[[2]] = models[[1]]  # All but one of the methods produced this model; no latdist

# Create data frame of criteria
crit = sapply(ft, function(model) c(getAIC(model), getBIC(model),
                                    getMallows(model), getCVscore(model),
                                    summary(model)$adj.r.squared,
                                    getGCVscore(model)))
criteria = data.frame(model = c("latdist", "no latdist"),
                      AIC = crit[1,], BIC = crit[2,], Mallow = crit[3,],
                      CV = crit[4,], adj.r.squared = crit[5,],
                      GCV = crit[6,])

# Force R to respect order of factor levels (for plotting purposes)
criteria$model = factor(criteria$model, levels = c("latdist", "no latdist"))

# Barplot comparison of criteria
crit.barplot.AIC = qplot(x = model, y = AIC, fill = 'red',
                         main = "AIC", xlab = '', ylab = "",
                         data = criteria, geom = 'bar', stat = 'identity',
                         position = 'dodge') + theme(legend.position="none") +
  geom_bar(stat="identity", colour="white") + coord_cartesian(ylim=c(1560,1575))

crit.barplot.BIC = qplot(x = model, y = BIC, fill = 'blue',
                         main = "BIC", xlab = '', ylab = "",
                         data = criteria, geom = 'bar', stat = 'identity',
                         position = 'dodge') + theme(legend.position="none") +
  geom_bar(stat="identity", fill="#799AFF", colour="white") + coord_cartesian(ylim=c(1625,1640))

crit.barplot.Mallow = qplot(x = model, y = Mallow, fill = 'red',
                            main = "Mallow's CP", xlab = '', ylab = "",
                            data = criteria, geom = 'bar', stat = 'identity',
                            position = 'dodge') + theme(legend.position="none") +
  geom_bar(stat = "identity", fill="#FFB445", colour="white") + coord_cartesian(ylim=c(14,16))

crit.barplot.adjr = qplot(x = model, y = adj.r.squared, fill = 'red',
                          main = "Adjusted R-squared", xlab = '', ylab = "",
                          data = criteria, geom = 'bar', stat = 'identity',
                          position = 'dodge') + theme(legend.position="none") +
  geom_bar(stat="identity", fill = "#FF9494", colour="white") + coord_cartesian(ylim=c(.743,.744))

crit.barplot.CV = qplot(x = model, y = CV, fill = 'red',
                        main = "LOO Cross-Validation Score", xlab = '', ylab = "",
                        data = criteria, geom = 'bar', stat = 'identity',
                        position = 'dodge') + theme(legend.position="none") +
  geom_bar(stat = "identity", fill="#6ADA6A", colour="white") + coord_cartesian(ylim=c(11490,11510)
      )

crit.barplot.GCV = qplot(x = model, y = GCV,
                         main = "General Cross-Validation Score", xlab = '', ylab = "",
                         data = criteria, geom = 'bar', stat = 'identity',
```

```
                        position = 'dodge') + theme(legend.position="none") +
  geom_bar(stat = "identity", fill="#5CBD7D", colour="white") + coord_cartesian(ylim=c(11285,11300)
      )

# Make barplots
multiplot(crit.barplot.AIC, crit.barplot.BIC, crit.barplot.Mallow,
          crit.barplot.adjr, crit.barplot.CV, crit.barplot.GCV, cols = 3)
```

5.2.3 Regression Diagnostics and Outlier Elimination

```
# Initialize selected model
ft = lm(formula = medv ~ londist + crim + zn + chas + nox + rm + dis +
     rad + tax + ptratio + b + lstat, data = bost)

# Compute statistics used in outlier detection
coefs = names(ft$coefficients)[-1]
n = length(ft$fitted.values)
p = length(coefs)
x = matrix(c(rep(1,n), as.matrix(bost[,coefs])), nrow = n)
h = x %*% solve(t(x)%*% x) %*% t(x)

fit = unname(ft$fitted.values)
res = unname(ft$residuals)
sigma = summary(ft)$sigma
stdRes = unname(res / (sigma*(1-diag(h))))
predRes = unname(res / (1 - diag(h)))
stdPredRes = unname(stdRes*sqrt((n-p-1) / (n-p-stdRes^2)))
cook = stdRes^2 * diag(h) / ((1-diag(h)) * (p+1))

# Make regression diagnostic plots
plots = list()
plots[[1]] = qplot(x = res, y = diag(h), xlab = "Leverage", ylab = "Residual",
                   geom = 'point') + theme(legend.position="none")
plots[[2]] = qplot(x = 1:n, y = cook, xlab = "Cook's Distance", ylab = "Observation #",
                   geom = 'point') + theme(legend.position="none")
plots[[3]] = qplot(x = qnorm(p = (1:n)/(n+1)), y = sort(res), geom = 'point',
                    xlab = "Normal Quantile", ylab = "Residual") + theme(legend.position="none")
plots[[4]] = qplot(x = fit, y = res, xlab = "Fitted Value", ylab = "Residual",
                   geom = 'point') + theme(legend.position="none")
plots[[5]] = qplot(x = bost$londist, y = res, xlab = "londist", ylab = "Residual",
                   geom = 'point') + theme(legend.position="none")
plots[[6]] = qplot(x = bost$crim, y = res, xlab = "crim", ylab = "Residual",
                   geom = 'point') + theme(legend.position="none")
plots[[7]] = qplot(x = bost$zn, y = res, xlab = "zn", ylab = "Residual",
                   geom = 'point') + theme(legend.position="none")
plots[[8]] = qplot(x = bost$chas, y = res, xlab = "chas", ylab = "Residual",
                   geom = 'point') + theme(legend.position="none")
plots[[9]] = qplot(x = bost$nox, y = res, xlab = "nox", ylab = "Residual",
                   geom = 'point') + theme(legend.position="none")
plots[[10]] = qplot(x = bost$rm, y = res, xlab = "rm", ylab = "Residual",
                   geom = 'point') + theme(legend.position="none")
plots[[11]] = qplot(x = bost$dis, y = res, xlab = "dis", ylab = "Residual",
                   geom = 'point') + theme(legend.position="none")
plots[[12]] = qplot(x = bost$rad, y = res, xlab = "rad", ylab = "Residual",
                   geom = 'point') + theme(legend.position="none")
plots[[13]] = qplot(x = bost$tax, y = res, xlab = "tax", ylab = "Residual",
                   geom = 'point') + theme(legend.position="none")
plots[[14]] = qplot(x = bost$ptratio, y = res, xlab = "ptratio", ylab = "Residual",
                   geom = 'point') + theme(legend.position="none")
plots[[15]] = qplot(x = bost$b, y = res, xlab = "b", ylab = "Residual",
                   geom = 'point') + theme(legend.position="none")
plots[[16]] = qplot(x = bost$lstat, y = res, xlab = "lstat", ylab = "Residual",
                   geom = 'point') + theme(legend.position="none")

multiplot(plotlist = plots, layout = matrix(1:16, nrow = 4, byrow = TRUE))

# Identify significant observations (outliers)
bonferroni = .05 / n
pValues = sapply(stdPredRes, function(t) {
```

```
  2*(1-pt(abs(t), n-p-2))
})

# Plot pValues vs. observation number and compare to Bonferroni line
plot.pValues = qplot(x = 1:n, y = pValues, xlab = "Observation #", ylab = "pValue",
      geom = 'point', size = I(4)) + theme(legend.position="none")

# Plot influential observations with Bonferroni line
plot.bon = qplot(x = 1:n, y = pValues, xlab = "Observation #", ylab = "pValue",
      geom = 'point', size = I(5)) + theme(legend.position="none") + geom_hline(y=bonferroni, col =
           "red") +
      coord_cartesian(xlim=c(368,374), ylim = c(-.00001, bonferroni*1.5))

multiplot(plot.pValues, plot.bon, cols = 2)

# Create model with outliers dropped
ftDropOutliers = lm(formula = medv ~ londist + crim + zn + chas + nox + rm + dis +
                      rad + tax + ptratio + b + lstat, data = bost[-which(pValues < bonferroni),])
```

5.2.4 Interpretation

```
data = bost[-which(pValues < bonferroni),]
ft = lm(formula = medv ~ rm + crim + ptratio, data = data)
variables = names(ft$coefficients)[-1]
summ = data.frame(coef = c(ft$coefficients),
                  min = c(0, apply(data[,variables], 2, min)),
                  firstQuartile = c(0, apply(data[,variables],2, function(x) quantile(x, probs =
                      .25))),
                  median = c(0, apply(data[,variables], 2, median)),
                  thirdQuartile = c(0, apply(data[,variables],2, function(x) quantile(x, probs =
                      .75))),
                  max = c(0, apply(data[,variables], 2, max)))
summ = rbind(c(0,min(data$medv),
               quantile(data$medv, probs = .25),
               median(data$medv),
               quantile(data$medv, probs = .75),
               max(data$medv)),
             summ)
rownames(summ)[1] = "medv"

summ$improvement = summ$coef*(summ$thirdQuartile - summ$firstQuartile)
summ = summ[rev(order(abs(summ$improvement))),]
summ[4,2:7] = rep(NA,6)
summ[5,c(1,7)] = c(NA,NA)

setwd("~/Desktop/Stat 151/Midterm 2/report")
sink(file = "summary2.R")
summary(ftDropOutliers)
sink()

setwd("~/Desktop/Stat 151/Midterm 2/report")
sink(file = "interpretation2.R")
summ
sink()

dat2 = as.matrix(bost[-which(pValues < bonferroni),c('crim','rm','ptratio')])
cormat2 = cor(dat2)
diag(cormat2) = 0
```