Final Project

Emily Briggs

Goals of the Project

I will be using statistical functions in R to explore and analyze data concerning Boston Housing. Although there are 13 total predictors in the dataset, my main goal is to look at a select five variables from the set and analyze which of those may be significant in influencing the median values of homes in the Boston area. The variables I will be looking at are crim, nox, ptratio, b, and lstat. These variables are of particular interest to me because they are social factors rather than economic, which I suppose may have an interesting effect on median house values. I am aiming to develop a functional and useful multiple linear regression model for the data which only includes significant predictors and can effectively predict the median values given inputs of predictor values.

Description of Data

The data I will be looking into is the Boston Housing Data from the mlbench package, which can be found here. It contains housing data for 506 census tracts of Boston from the 1970 census. The original data are 506 observations on 14 variables, with 'medv' being the target variable.

The dataset contains the following variables:

- 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
- lstat: percentage of lower status of the population
- medv: median value of owner-occupied homes in USD 1000's

```
#Downloading data and checking the structure
library(mlbench)
library(car)
```

```
Loading required package: carData
```

```
data(BostonHousing)
head(BostonHousing)
```

```
     crim zn indus chas   nox    rm  age    dis rad tax ptratio      b lstat
1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98
2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14
3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03
4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63  2.94
5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90  5.33
6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12  5.21
  medv
1 24.0
2 21.6
3 34.7
4 33.4
5 36.2
6 28.7
```

```
str(BostonHousing)
```

```
’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   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ 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    : num  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 ...
 $ b      : 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 ...
```

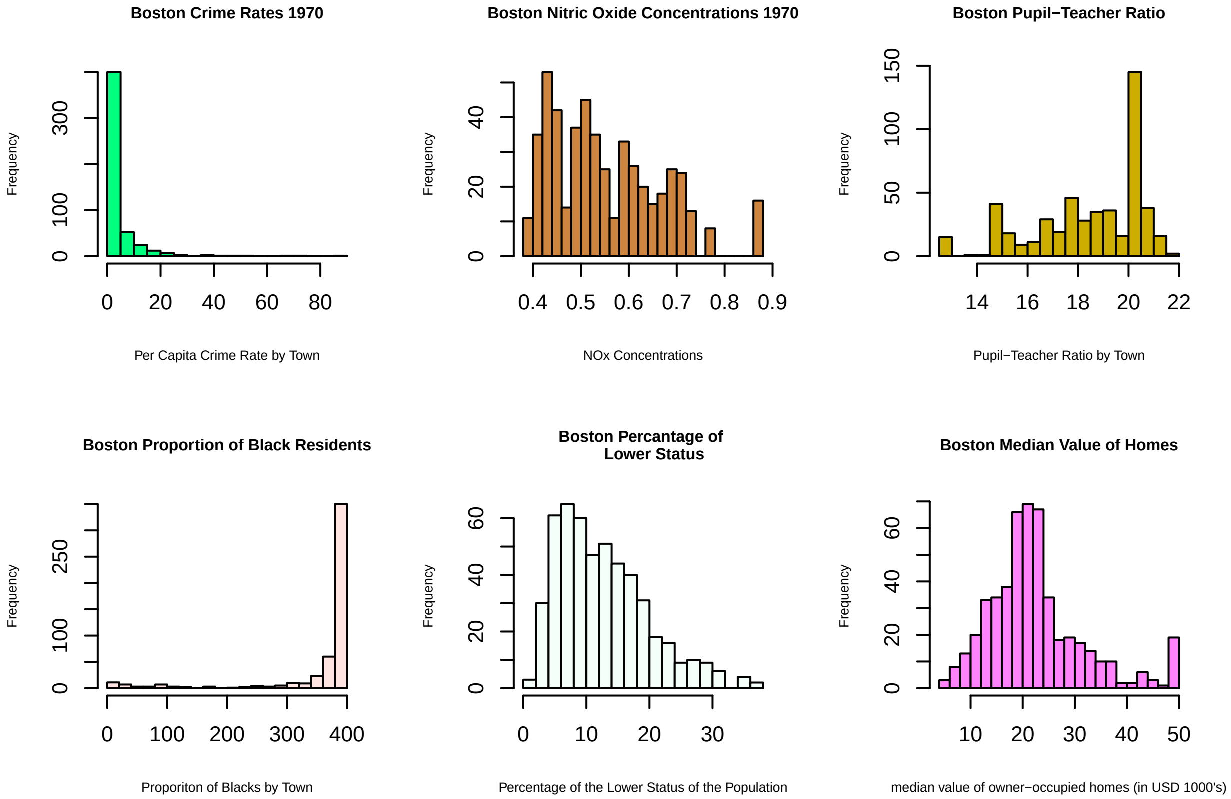
```
#For this analysis, I only want to look at crim, nox, ptratio, b, and lstat
bos <- as.data.frame(BostonHousing[, c('crim', 'nox', 'ptratio', 'b', 'lstat',
                                       'medv')])
str(bos)
```

```
'data.frame':   506 obs. of  6 variables:
 $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
 $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
 $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
 $ b      : 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 ...
```

```
#Getting a five number summary for each variable in the dataset
summary(bos)
```

```
      crim                nox            ptratio            b
 Min.   : 0.00632   Min.   :0.3850   Min.   :12.60   Min.   :  0.32
 1st Qu.: 0.08205   1st Qu.:0.4490   1st Qu.:17.40   1st Qu.:375.38
 Median : 0.25651   Median :0.5380   Median :19.05   Median :391.44
 Mean   : 3.61352   Mean   :0.5547   Mean   :18.46   Mean   :356.67
 3rd Qu.: 3.67708   3rd Qu.:0.6240   3rd Qu.:20.20   3rd Qu.:396.23
 Max.   :88.97620   Max.   :0.8710   Max.   :22.00   Max.   :396.90
     lstat            medv
 Min.   : 1.73   Min.   : 5.00
 1st Qu.: 6.95   1st Qu.:17.02
 Median :11.36   Median :21.20
 Mean   :12.65   Mean   :22.53
 3rd Qu.:16.95   3rd Qu.:25.00
 Max.   :37.97   Max.   :50.00
```

```
#Looking at the frequency distributions of each variable
par(mfrow = c(2, 3))
hist(bos$crim, col = "springgreen", main = "Boston Crime Rates 1970", xlab =
       "Per Capita Crime Rate by Town", cex.main = 0.7, cex.lab = 0.6, breaks =
       20)
hist(bos$nox, col = "peru", main = "Boston Nitric Oxide Concentrations 1970",
     xlab = "NOx Concentrations", cex.main = 0.7, cex.lab = 0.6, breaks = 20)
hist(bos$ptratio, col = "gold3", main = "Boston Pupil-Teacher Ratio", xlab =
       "Pupil-Teacher Ratio by Town", cex.main = 0.7, cex.lab = 0.6, breaks =
       20)
hist(bos$b, col = "mistyrose", main = "Boston Proportion of Black Residents",
     xlab = "Proporiton of Blacks by Town", cex.main = 0.7, cex.lab = 0.6,
     breaks = 20)
hist(bos$lstat, col = "mintcream", main = "Boston Percantage of
     Lower Status", xlab = "Percentage of the Lower Status of the Population",
     cex.main = 0.7, cex.lab = 0.6, breaks = 20)
hist(bos$medv, col = "orchid1", main = "Boston Median Value of Homes",
     xlab = "median value of owner-occupied homes (in USD 1000's)",
     cex.main = 0.7, cex.lab = 0.6, breaks = 20)
```



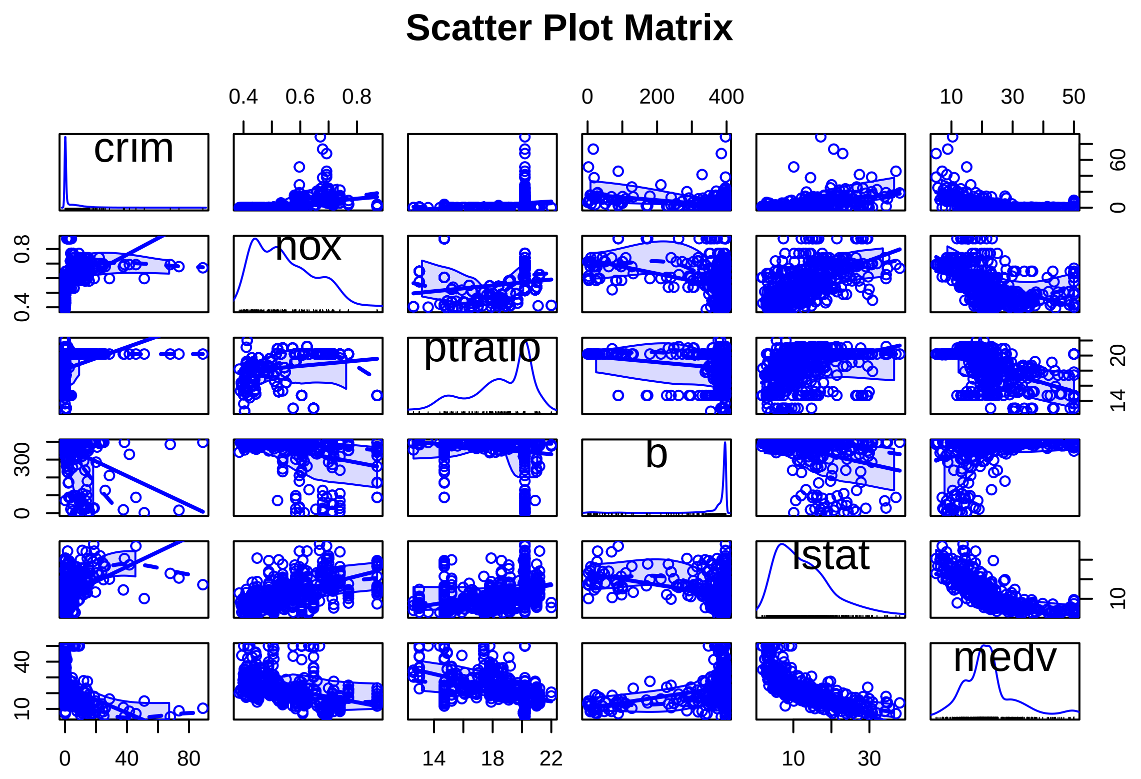
Looking at the histograms for each variable, we can see that most of them are not approximately normal. The distributions for nox, lstat, and the dependent variable medv are fairly right skewed, the distribution for ptratio is left skewed. The distributions for crim and b are highly skewed. This might explain some of the abnormalites we see when checking assumptions later in the analysis, specifically when checking the normality of the residuals.

```
#Examine the relationships two at a time (bivariate correlations)
cor(bos)
```

```
             crim       nox   ptratio          b     lstat       medv
crim    1.0000000 0.4209717 0.2899456 -0.3850639 0.4556215 -0.3883046
nox     0.4209717 1.0000000 0.1889327 -0.3800506 0.5908789 -0.4273208
ptratio 0.2899456 0.1889327 1.0000000 -0.1773833 0.3740443 -0.5077867
```

```
b       -0.3850639 -0.3800506 -0.1773833  1.0000000 -0.3660869  0.3334608
lstat    0.4556215  0.5908789  0.3740443 -0.3660869  1.0000000 -0.7376627
medv    -0.3883046 -0.4273208 -0.5077867  0.3334608 -0.7376627  1.0000000
```

```
library(car)
scatterplotMatrix(bos, main = "Scatter Plot Matrix")
```



Note, that lstat and medv have relatively strong (negative) correlation.

```
#Now, I will fit a multiple linear regression model with all five predictors
bos_fit <- lm(medv ~ crim + nox + ptratio + b + lstat, data = bos)
(summ_fit1 <- summary(bos_fit))
```

```
Call:
lm(formula = medv ~ crim + nox + ptratio + b + lstat, data = bos)

Residuals:
     Min       1Q   Median       3Q      Max 
-12.3062  -3.6537  -0.8722   1.9247  26.8317 

Coefficients:
```

```
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 50.418163   3.104971  16.238   <2e-16 ***
crim        -0.013596   0.035639  -0.381   0.7030    
nox          1.465277   2.862189   0.512   0.6089    
ptratio     -1.122598   0.129682  -8.657   <2e-16 ***
b            0.006162   0.003182   1.937   0.0534 .  
lstat       -0.800496   0.048772 -16.413   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.771 on 500 degrees of freedom
Multiple R-squared:  0.6101,    Adjusted R-squared:  0.6062 
F-statistic: 156.5 on 5 and 500 DF,  p-value: < 2.2e-16
```

From the summary, we see that the value of the F statistic is 156.5 and the p- value is less than 2e-16. We have strong evidence to support that not all coefficients are 0. In particular, we can see from the summary that ptratio, b, and lstat may be of significance to the model (from significance codes). We can see from the p values that ptratio and lstat are significant at the .05 significance level (and lower), and b is significant at the 0.1 significance level.

```
#Exploring R squared values
summ_fit1$r.squared
```

```
[1] 0.6101187
```

```
summ_fit1$adj.r.squared
```

```
[1] 0.6062199
```

R squared = .6101187, which suggests that 61.01187% of the variability in the data can be explained by our model. Adjusted r squared = .606, which suggests that 60.6% of the variability in the data can be explained by our model after adjusting with a penalty for more complex models.

```
#Analyzing coefficients
summ_fit1$coefficients
```

```
               Estimate  Std. Error      t value     Pr(>|t|)
(Intercept) 50.41816279 3.104970979  16.2378854 6.277079e-48
crim        -0.01359600 0.035639238  -0.3814895 7.030021e-01
nox          1.46527684 2.862188833   0.5119428 6.089171e-01
ptratio     -1.12259829 0.129682464  -8.6565157 6.712364e-17
b            0.00616212 0.003182061   1.9365184 5.336735e-02
lstat       -0.80049600 0.048772268 -16.4129337 9.664736e-49
```

As mentioned above, only ptratio and lstat are significant at the 5% level. The regression coefficients indicate the expected increase in the dependent variable (median value) for a unit change in a predictor variable, holding all other predictor variables constant. For example, the regression coefficient for ptratio is -1.123, so an increase of 1% in pupil-teacher ratio is associated with a 1.123% decrease in the murder rate on average, controlling for crime rate, nitric oxides concentration, proportion of blacks, and lower status. The coefficient is significantly different from zero, with p-value < .0001.

```
#Obtaining confidence intervals for each coefficient
confint(bos_fit)
```

```
                    2.5 %      97.5 %
(Intercept)  4.431776e+01 56.51856087
crim        -8.361711e-02  0.05642512
nox         -4.158122e+00  7.08867602
ptratio     -1.377388e+00 -0.86780858
b           -8.973859e-05  0.01241398
lstat       -8.963198e-01 -0.70467216
```

Interpretation: For example, [-1.377, -0.868] is a 95% confidence interval for the true change in median house value for a 1% change in pupil-teacher ratio.

As we have seen above, only three predictors (ptratio, lstat, and b) are significant to the regression model, and b is only significant at the 0.1 level. In hopes of finding the best model, I will test to see if models containing less predictors are as adequate at predicting as the full model.

```
#Creating new model with three predictors
bos_fit3 <- lm(medv ~ ptratio + b + lstat, data = bos)
#Using anova()
anova(bos_fit3, bos_fit)
```

```
Analysis of Variance Table

Model 1: medv ~ ptratio + b + lstat
Model 2: medv ~ crim + nox + ptratio + b + lstat
  Res.Df   RSS Df Sum of Sq      F Pr(>F)
1    502 16666
2    500 16654  2    11.678 0.1753 0.8393
```

```
#Using Akaike Information Criterion (AIC)
AIC(bos_fit, bos_fit3)
```

```
         df      AIC
bos_fit   7 3217.872
bos_fit3  5 3214.227
```

```r
#Creating new model with two predictors (now excluding b)
bos_fit2 <- lm(medv ~ ptratio, lstat, data = bos)
#Using anova()
anova(bos_fit2, bos_fit3)
```

```
Analysis of Variance Table

Model 1: medv ~ ptratio
Model 2: medv ~ ptratio + b + lstat
  Res.Df   RSS Df Sum of Sq      F    Pr(>F)
1    504 22342
2    502 16666  2    5676.3 85.488 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#Using AIC
AIC(bos_fit3, bos_fit2)
```

```
         df      AIC
bos_fit3  5 3214.227
bos_fit2  3 3358.540
```

```r
#Verifying with adjusted r-squared
(summ_fit3 <- summary(bos_fit3))
```

```
Call:
lm(formula = medv ~ ptratio + b + lstat, data = bos)

Residuals:
     Min       1Q   Median       3Q      Max
-12.3063  -3.6707  -0.8439   1.9123  26.9096

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 51.294202   2.615734  19.610   <2e-16 ***
ptratio     -1.133111   0.127843  -8.863   <2e-16 ***
b            0.006122   0.003021   2.026   0.0433 *
lstat       -0.792905   0.040989 -19.344   <2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.762 on 502 degrees of freedom
Multiple R-squared:  0.6098,    Adjusted R-squared:  0.6075
F-statistic: 261.6 on 3 and 502 DF,  p-value: < 2.2e-16
```

After running anova and AIC tests for the model with three predictors versus the full model, we can see that the reduced model predicts just as well as the full model, and it is justified to drop crim and nox. The anova test yields a high p value (.8393), which tells us the test is insignificant and crim and nox do not add to linear prediction above and beyond the other three variables. Since the reduced model has a smaller AIC, this result is further verified because models with smaller AIC values-indicating adequate fit with fewer parameters-are preferred. However, dropping b from the set of predictors is not advisable. After creating a model containing only ptratio and lstat and running the same tests, we see that it is useful to keep b in the model. The anova test is significant and the AIC for the model with three predictors is lower than that of the model with only two. In conclusion, the best model for our data is the one containing three predictors - ptratio, b, and lstat. This is verified because the adjusted r squared value for the model with three predictors is slightly higher than the value for the full model. We will use this model from now on.

Now that we have a good model, we can check assumptions.

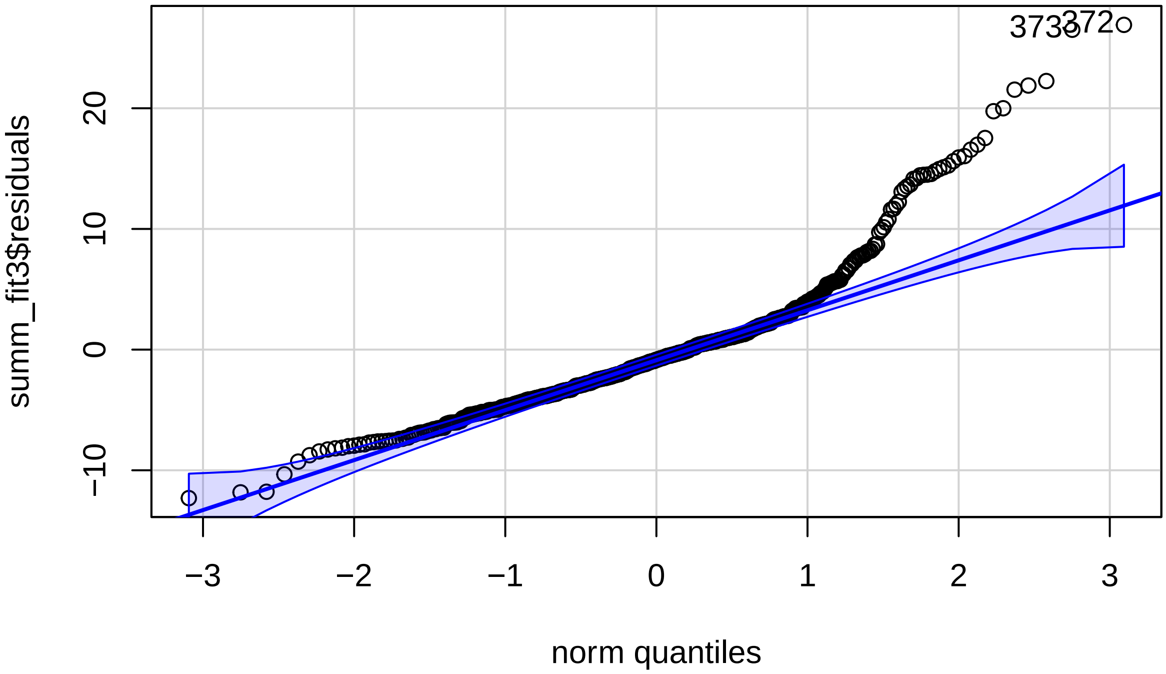
```
#Normality of residuals (Shapiro-Wilk and qqPlot)
shapiro.test(summ_fit3$residuals)
```

```

    Shapiro-Wilk normality test

data:  summ_fit3$residuals
W = 0.88932, p-value < 2.2e-16
```

```
qqPlot(summ_fit3$residuals)
```



```
[1] 372 373
```

This output shows us that the normality assumption is not validated. The shapiro wilks test yields a very small p value, telling us we can reject the null hypothesis that the residuals have a normal distribution. The qqplot shows a highly skewed distribution. We can try to perform some corrective measures.

```
summary(powerTransform(bos$medv))
```

```
bcPower Transformation to Normality
         Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
bos$medv    0.2166        0.33       0.0582        0.375

Likelihood ratio test that transformation parameter is equal to 0
 (log transformation)
                           LRT df      pval
LR test, lambda = (0) 7.311122  1 0.0068529

Likelihood ratio test that no transformation is needed
                           LRT df       pval
LR test, lambda = (1) 87.26983  1 < 2.22e-16
```

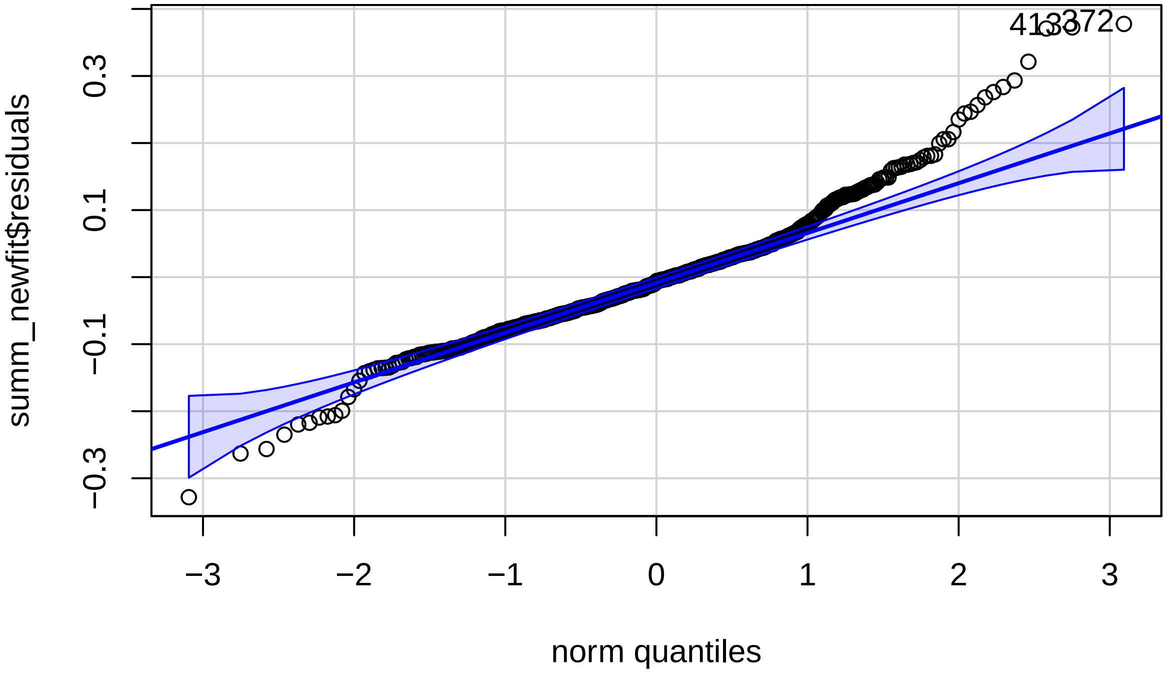
This summary tells us the hypothesis that lambda = 1 can be rejected (very small p value), so there is sufficient evidence that a transformation on the response variable could be useful. We can try replacing medv with medv$^{0.2166}$.

```
bos$mod_medv <- (bos$medv ^ .2166)
new_fit1 <- lm((mod_medv) ~ ptratio + b + lstat, data = bos)
summ_newfit <-  summary(new_fit1)
shapiro.test(summ_newfit$residuals)
```

```
    Shapiro-Wilk normality test

data:  summ_newfit$residuals
W = 0.96112, p-value = 2.661e-10
```

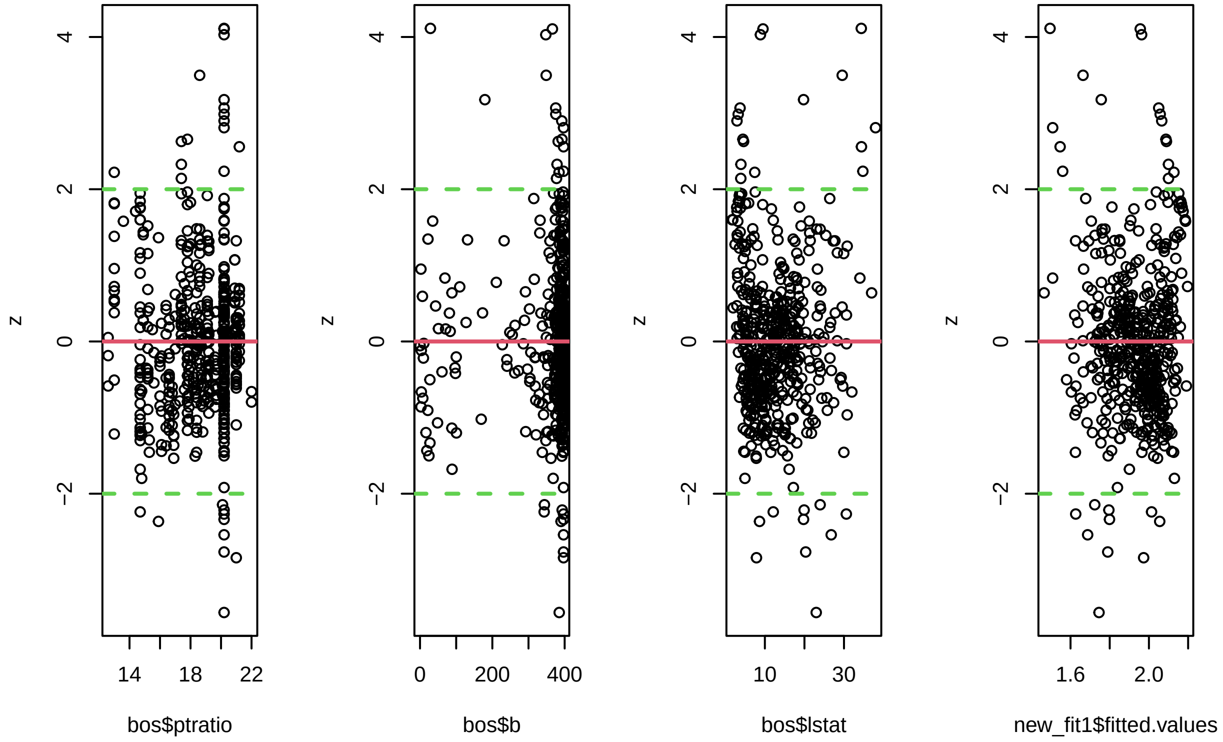
```
qqPlot(summ_newfit$residuals)
```



```
[1] 372 413
```

This transformation increases the p-value slightly, but we still cannot validate the assumption of normality. We will move on for now, but keep this in mind.

```
#Equal variances assumption
z <- rstudent(new_fit1)
par(mfrow = c(1,4))
plot(bos$ptratio, z)
abline(h=0,col=2,lwd=2)
abline(h=2, col=3,lwd=2, lty=2)
abline(h=-2,col=3,lwd=2, lty=2)
plot(bos$b, z)
abline(h=0,col=2,lwd=2)
abline(h=2, col=3,lwd=2, lty=2)
abline(h=-2,col=3,lwd=2, lty=2)
plot(bos$lstat, z)
abline(h=0,col=2,lwd=2)
abline(h=2, col=3,lwd=2, lty=2)
abline(h=-2,col=3,lwd=2, lty=2)
plot(new_fit1$fitted.values, z)
abline(h=0,col=2,lwd=2)
abline(h=2, col=3,lwd=2, lty=2)
abline(h=-2,col=3,lwd=2, lty=2)
```

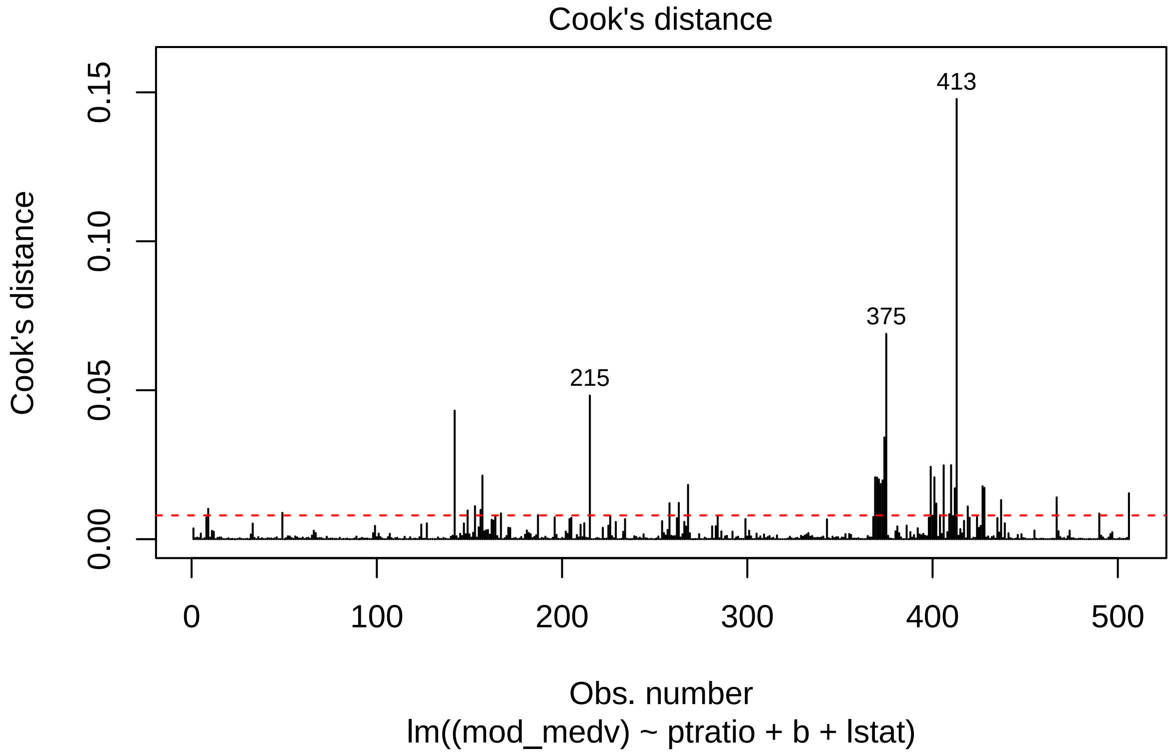


None of the plots really show any trend in the studentized residuals (although the residuals for b are skewed). They are for the most part centered around 0., and most points fall between -2 and 2.

```
#Running a test for outliers
outlierTest(new_fit1)
```

```
    rstudent unadjusted p-value Bonferroni p
413 4.112815         4.5681e-05     0.023115
372 4.105571         4.7088e-05     0.023827
373 4.029816         6.4495e-05     0.032634
```

```
#Observing influential points
cutoff <- 4/(nrow(bos)-length(new_fit1$coefficients))
plot(new_fit1, which=4, cook.levels=cutoff)
abline(h=cutoff, lty=2, col="red")
```

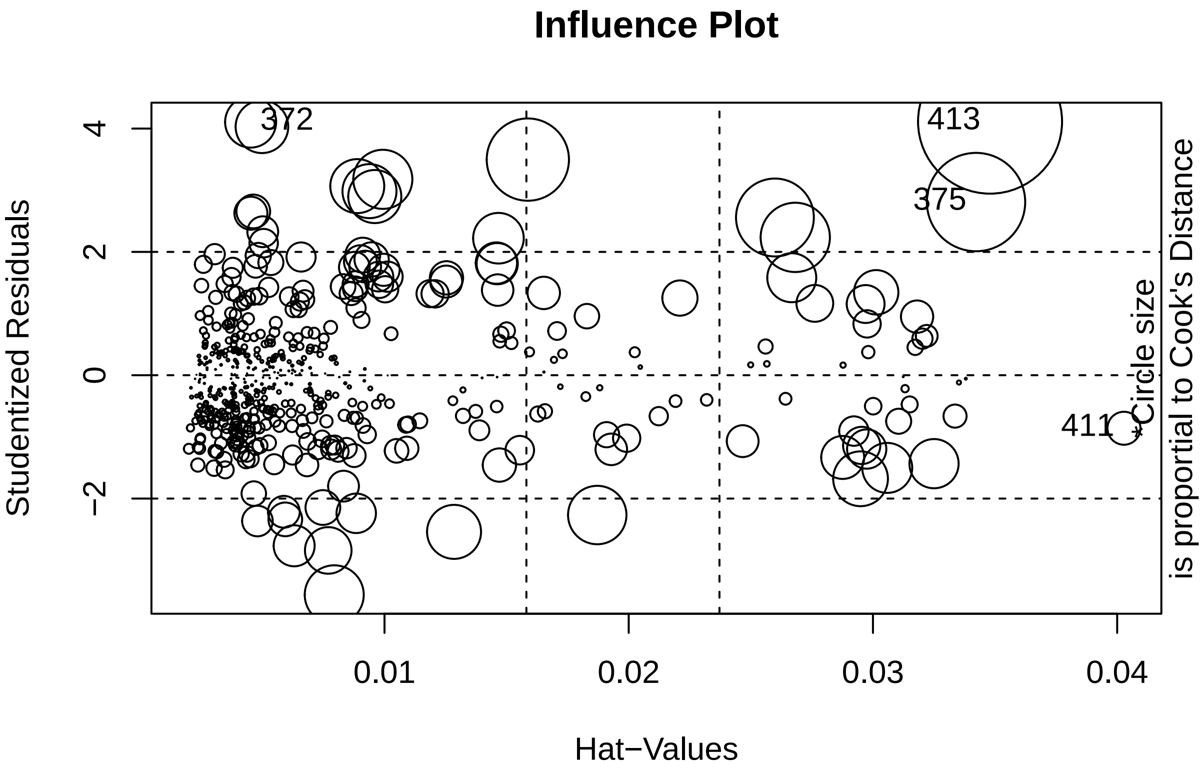


```
influencePlot(new_fit1, main="Influence Plot")
```

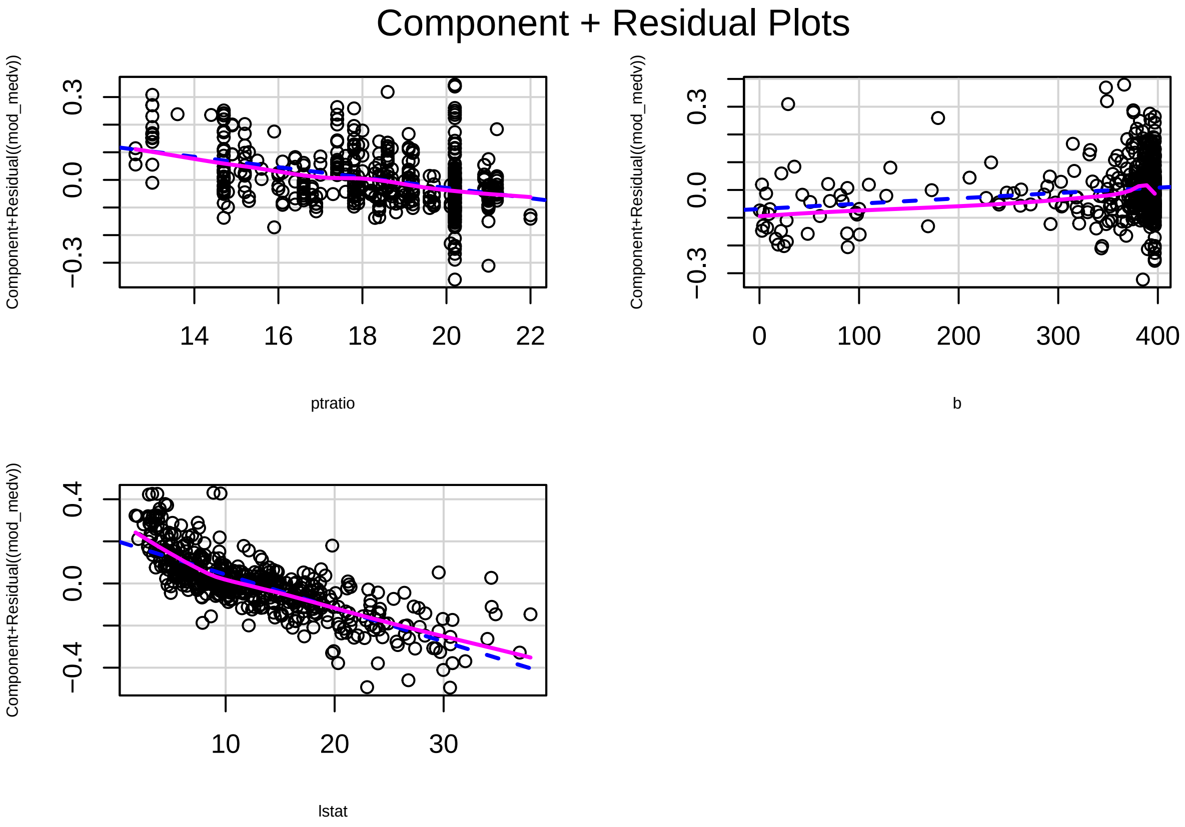
```
       StudRes         Hat       CookD
372  4.1055712 0.004518691 0.018542215
```

```
375  2.8080634 0.034221116 0.068905544
411 -0.8595718 0.040277606 0.007756188
413  4.1128153 0.034795245 0.147762385
```

```
mtext("*Circle size
is proportial to Cook's Distance", side =4)
```



```
#Checking linearity assumption
crPlots(new_fit1, cex.lab =0.6)
```



```
boxTidwell(mod_medv ~ ptratio + b + lstat, data = bos)
```

```
        MLE of lambda Score Statistic (z)  Pr(>|z|)
ptratio      -4.32613              1.7276   0.08406 .
b             0.45975             -1.4818   0.13839
lstat         0.13661              7.8504 4.146e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

iterations =  14
```

Although the crPlots appear fairly linear, the boxTidwell test shows us that a transformation on the ptratio and lstat predictors could be useful (p-values = .08406 and 4.146e-15). We can try using ptratio$^{-4.32613}$ and lstat$^{0.13661}$.

```
#Transforming predictor variables
bos$mod_pt <- (bos$ptratio ^ -4.32613)
bos$mod_lstat <- (bos$lstat ^ .13661)
new_fit2 <- lm(mod_medv ~ mod_pt + b + mod_lstat, data = bos)
(summ_newfit2 <- summary(new_fit2))
```

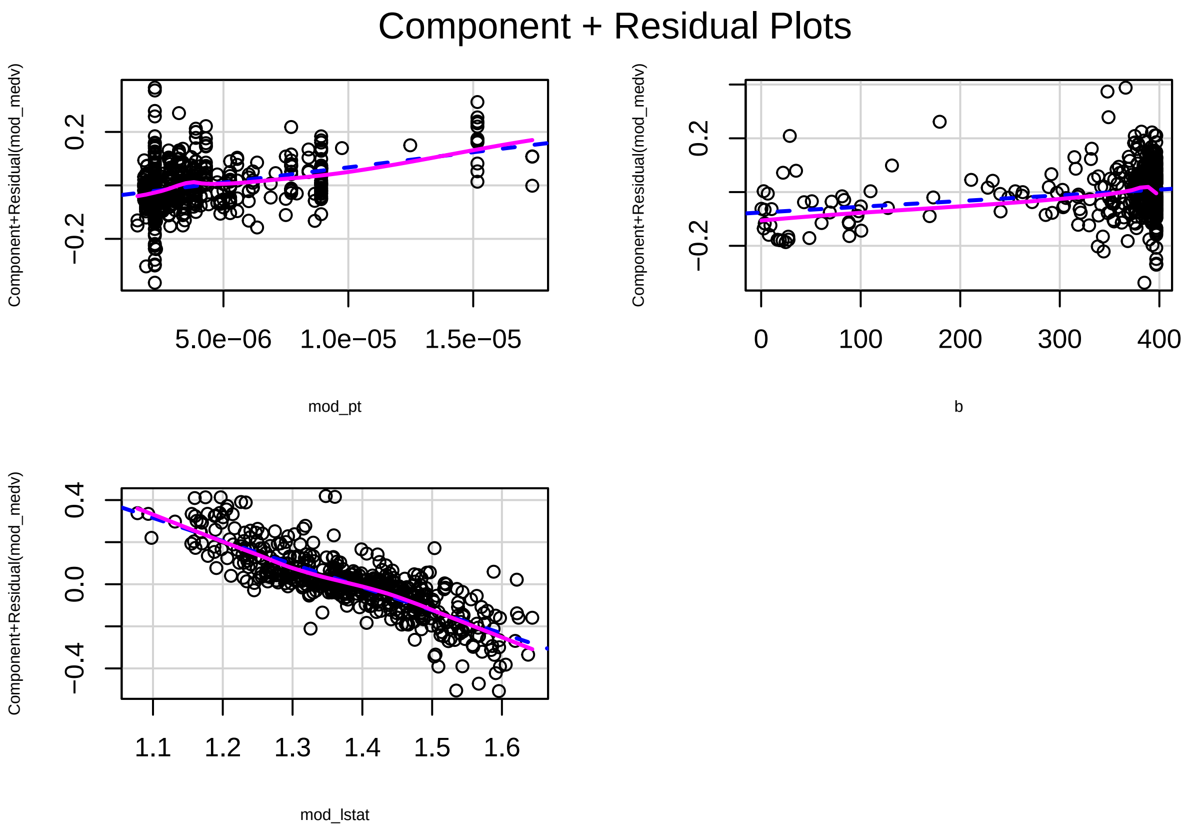
```
Call:
lm(formula = mod_medv ~ mod_pt + b + mod_lstat, data = bos)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.34334 -0.05526 -0.00226  0.05002  0.38617 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.337e+00  6.283e-02  53.113  < 2e-16 ***
mod_pt       1.137e+04  1.411e+03   8.059 5.66e-15 ***
b            2.132e-04  4.542e-05   4.694 3.47e-06 ***
mod_lstat   -1.098e+00  3.868e-02 -28.381  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.08736 on 502 degrees of freedom
Multiple R-squared:  0.7363,    Adjusted R-squared:  0.7347 
F-statistic: 467.2 on 3 and 502 DF,  p-value: < 2.2e-16
```

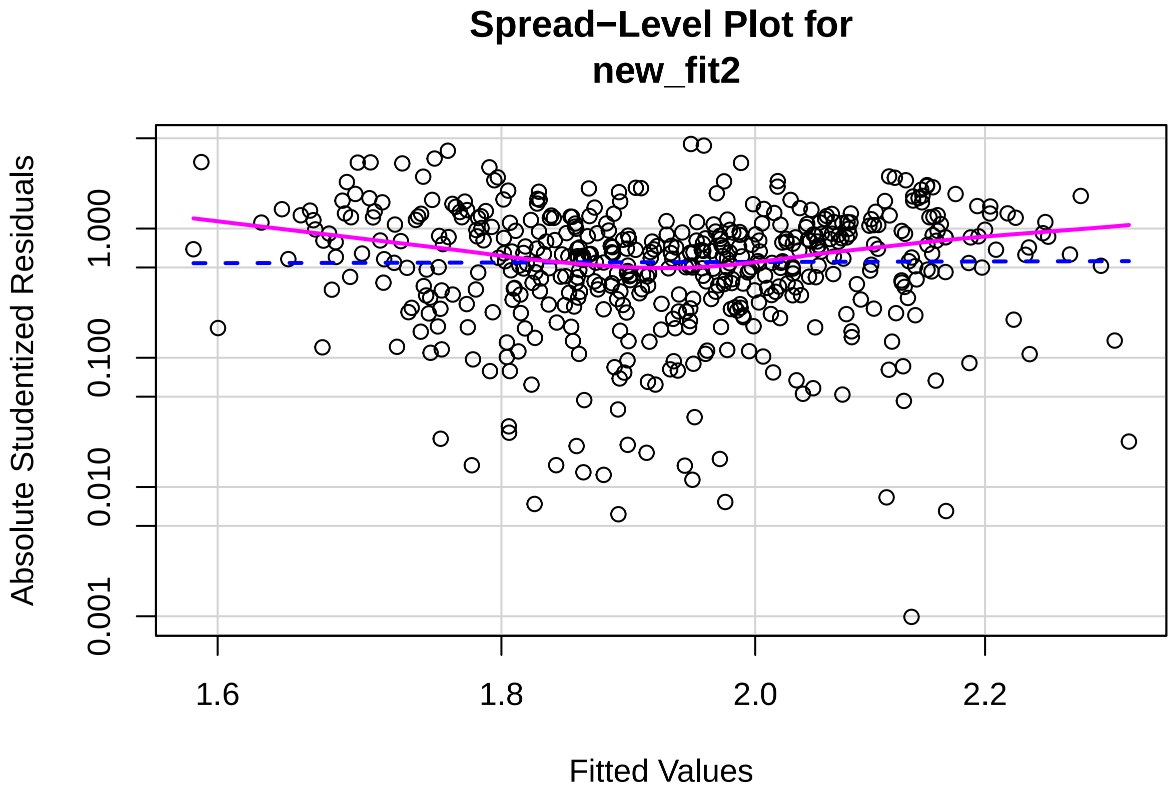
```r
crPlots(new_fit2, cex.lab = 0.6)
```



```
#Checking homoscedasticity
ncvTest(new_fit2)
```

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 6.526573, Df = 1, p = 0.010627
```

```
spreadLevelPlot(new_fit2)
```



```
Suggested power transformation:  0.9055718
```

The ncv Test yields a low p value (.010627), which indicates heteroscedasticity may be present. This is not ideal. The spread level plot is somewhat parabolic, which is also not ideal as it indicates the homoscedasticity assumption may not be validated. The suggested power transformation is close to 1, and does not have any real effect on improving the model, so we will ignore it. For now, we will acknowledge that many of our assumptions are not validated, and keep this in mind when using the linear model.

```
#Checking for multicollinearity
library(car)
vif(new_fit2)
```

```
   mod_pt         b mod_lstat 
 1.116236  1.137992  1.252222 
```

In this case, we don't see any evidence for multicollinearity, which is good. We know this because none of the VIF's are higher than 5. The square root of the VIF indicates the degree to which the confidence interval for that variable's regression parameter is expanded relative to a model with uncorrelated predictors.

Lastly, we can check if adding an interaction term will improve our model. I will check interactions between all combinations of the three predictors and see if adding the term significantly increases the adjusted R-squared value from the model without interaction terms.

```
#Obtaining adjusted R-squared for model with no interaction
(summ_newfit2$adj.r.squared)
```

```
[1] 0.7347084
```

```
#Adding interaction between ptratio and b
int1 <- lm(mod_medv~ mod_pt + b + mod_lstat + mod_pt:b, data = bos)
summ_int1 <- summary(int1)
(summ_int1$adj.r.squared)
```

```
[1] 0.7352688
```

```
#Adding interaction between ptratio and lstat
int2 <- lm(mod_medv~ mod_pt + b + mod_lstat + mod_pt:mod_lstat, data = bos)
summ_int2 <- summary(int2)
(summ_int2$adj.r.squared)
```

```
[1] 0.7342275
```

```
#Adding interaction between b and lstat
int3 <- lm(mod_medv~ mod_pt + b + mod_lstat + b:mod_lstat, data = bos)
summ_int3 <- summary(int3)
(summ_int3$adj.r.squared)
```

```
[1] 0.7341902
```

It appears no interaction is present, and it is unnecessary to include an interaction term in our model. Adjusted r-squared does not significantly increase for the addition of any interaction term, and the summaries show that the interaction term in all the new models is never significant.

Conclusion

After running multiple tests to observe, analyze, and try to polish our model, we are left with the multiple linear regression model containing three predictors- ptratio, b, and lstat- with power transformations on ptratio and lstat, as well as the response variable medv. The final model reads:
$medv^{.2166} = 3.337 + 11370 * ptratio^{-4.32613} + .0002132 * b - 1.098 * lstat^{.13661} + error$
This is the most predictive model developed with the tools I used, however it is important to acknowledge that the assumptions for normality, linearity, and homoscedasticity were not validated by the tests I ran. Overall, it is still an adequate predictive model for median Boston house values given pupil-teacher ratio, proportion of blacks, and lower status as independent variables.

We can test the model's predictive ability:

```r
#Dividing data into a training sample (70%) and a validation sample (30%)
set.seed(1234)
train <- sample(nrow(bos), 0.7*nrow(bos))
bos.train <- bos[train,]
bos.validate <- bos[-train,]

#Using the training set data to fit a multiple linear regression model
fit_train <- lm(mod_medv ~ mod_pt + b + mod_lstat, data = bos.train)
summary(fit_train)
```

```
Call:
lm(formula = mod_medv ~ mod_pt + b + mod_lstat, data = bos.train)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.33725 -0.05674 -0.00487  0.05411  0.38653 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.390e+00  8.102e-02  41.843  < 2e-16 ***
mod_pt       1.177e+04  1.795e+03   6.557 1.97e-10 ***
b            1.718e-04  5.853e-05   2.936  0.00355 ** 
mod_lstat   -1.126e+00  4.945e-02 -22.775  < 2e-16 ***
---
Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Residual standard error: 0.09232 on 350 degrees of freedom
Multiple R-squared:  0.7184,    Adjusted R-squared:  0.716 
F-statistic: 297.6 on 3 and 350 DF,  p-value: < 2.2e-16
```

```
#predicting the target variable
predictions <- predict(fit_train, bos.validate)


# computing model performance metrics
library('caret')
```

```
Loading required package: ggplot2

Loading required package: lattice
```

```
data.frame(R2 = R2(predictions, bos.validate$mod_medv),
           RMSE = RMSE(predictions, bos.validate$mod_medv),
           MAE = MAE(predictions, bos.validate$mod_medv))
```

```
         R2       RMSE        MAE
1 0.7829823 0.07509149 0.05802189
```

The model seems to be effective in making predictions of median house values. RMSE, or root mean mean-squared error, explains on an average how much of the predicted value will be from the actual value. Based on RMSE = .0751, we can conclude that on an average predicted value will be off by .0751 from the actual value, which is very low. MAE, or mean absolute error, measures the accuracy of the predicted values, and is also very low. We can conclude that our model is remarkably accurate.

(*Aside:* Because the notes did not have too much information on supervised learning for linear regression, I followed an article from Rishu Mishra on GeeksforGeeks which can be found here. I also used a few data science functions from the caret package, which can be found here.)