Final Problem 1

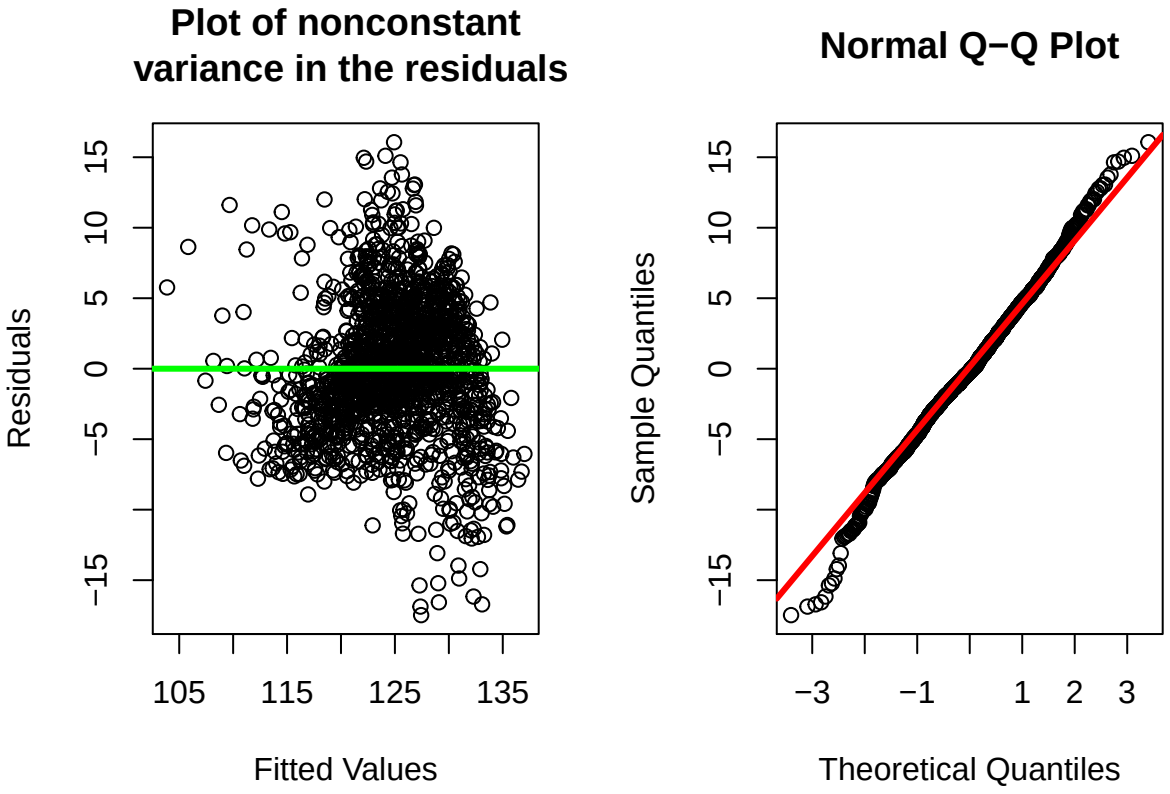
Michael Frasco

December 3, 2014

Fit a linear model of pressure on all other variables. Diagnose it to decide whether you see problems other than nonlinearity with it. Decide whether the output and the variables need transformation. Run the transformed model.

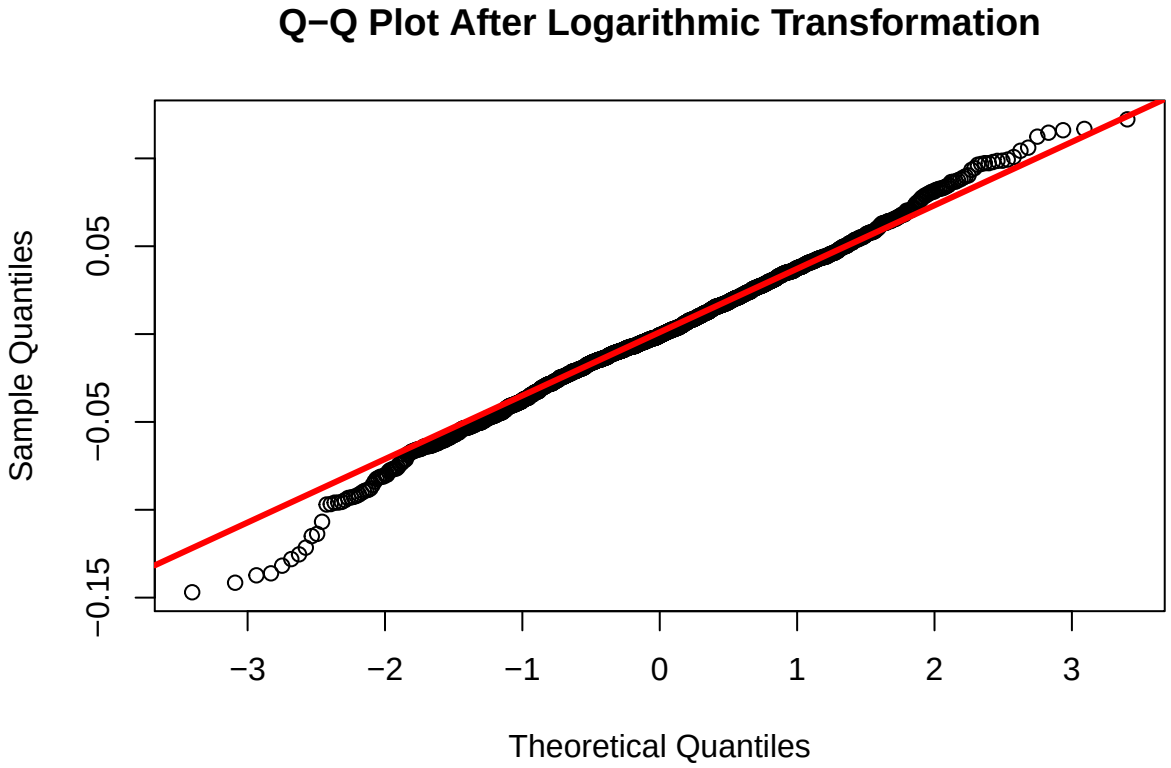
SUMMARY FOR a) After running an ordinary least squares regression and examining diagnostics for my fit, I noticed a couple problems with my model other than nonlinearity: the residuals do not appear to be normally distributed, there is nonconstant variance in the plot of the residuals, and there is strong correlation between successive residuals (the data is ordered by the predictor variables). After experimenting with many transformations, I found that a logarithmic transformation of the response variable improved the normality of the residuals the best.

Below is a plot of the residuals against the fitted values, which reveals a sudden increase in the variance of the residuals around 125. On the right is a qqplot of the residuals, which deviates from the normal line in a pattern indicative of a heavy tailed distribution.



To improve the normality of the residuals, I tried using a Box-Cox transformation on the response variable. However, the suggested power transformations of 1.5 or 2 did not improve the normality. Since histograms of the predictor variables are heavily right-skewed, I tried various logarithmic and square root transformations on the both the predictor and the response variables. However, none of these models fixed the nonconstant variance of the residuals.

In the end, I found that a logarithmic transformation of the response variable fixed the normality of the residuals the best. This transformation also had an improvement on the model's R-squared. Below is the impact of that transformation on the normality of the residuals.

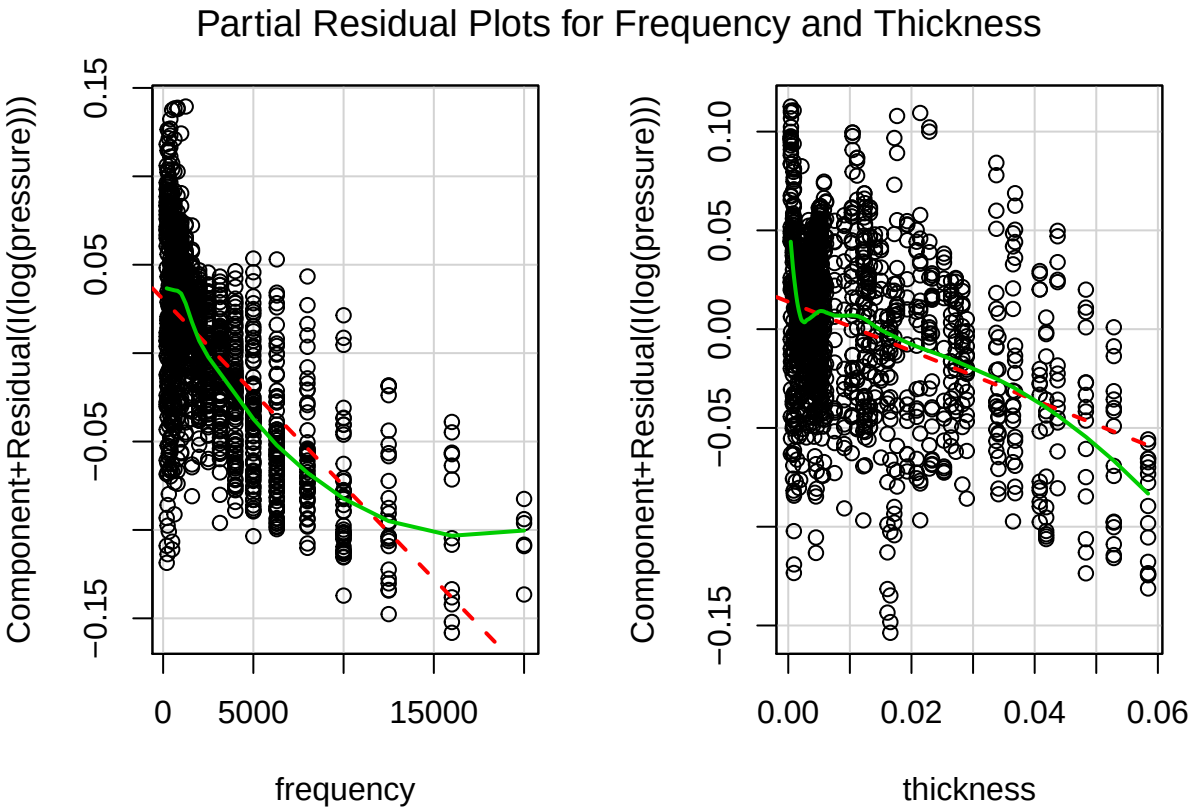


While the lower tail still deviates from the normal line, the residuals on the upper tail are pulled closer to the normal line. With that being said, this transormation is not great, and I might prefer to not implement this transformation to fix normality because I maintain the interpretability of my model.

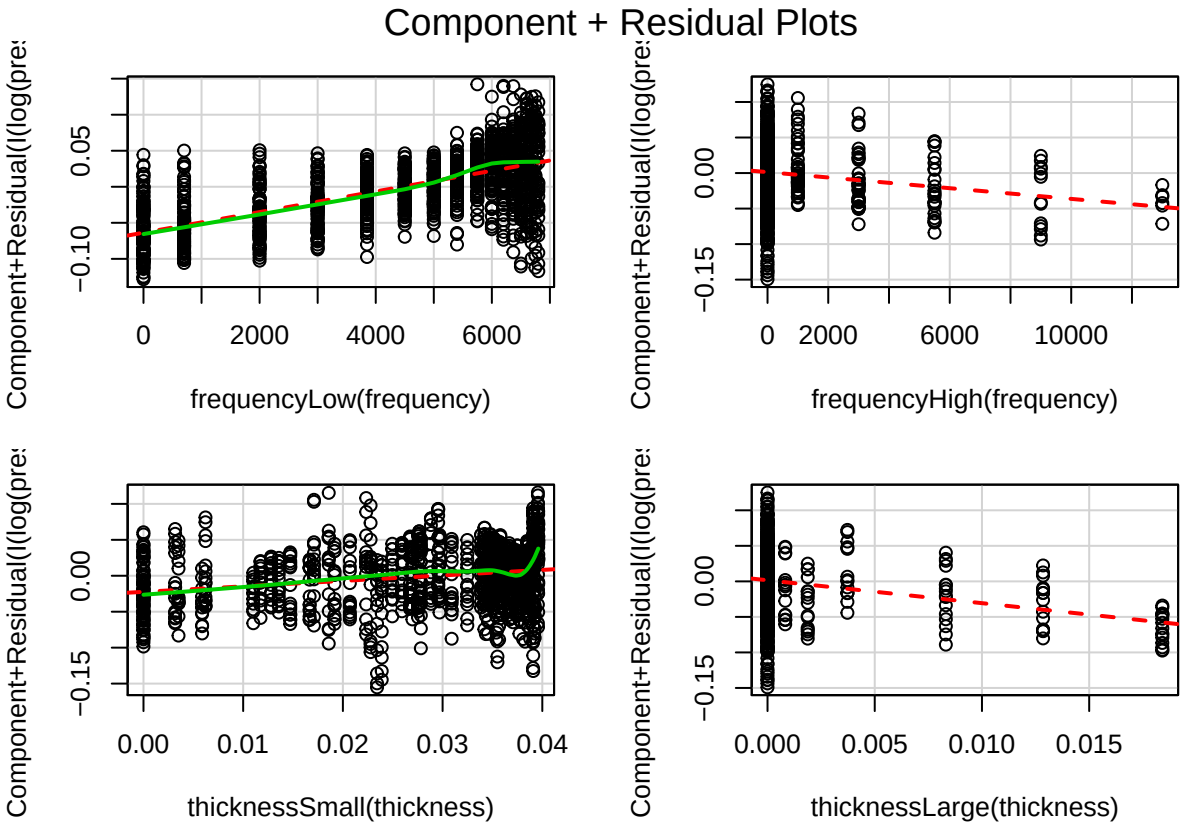
With your best model from above, decide whether there is nonlinearity in the fit. Decide how to fix this nonlinearity and if the changes are significant

SUMMARY FOR b) I use broken stick regression on two predictor variables to improve nonlinearity. This transformation fixes the problem very significantly.

In order to check the structure of my model, which assumes a linear relationship, we can use partial residual plots. In a multivariate setting like this, partial residual plots allow us to examine the relationship between one predictor and the response. After examining all five predictors, I noticed the most evidence for nonlinearity in the plots for frequency and thickness, which are shown below. In the graph below, the red line is an ordinary least squares regression and the green line is a loess smoother, used to reveal nonlinearity.



I tried several methods to fix these nonlinear relationship, such as combining functional transformations and including polynomial terms in the regression. However, I found that using broken stick regression fixed nonlinearity the best while still allowing for a reasonable interpretation of the model. I will explain the reasoning behind this model in part c). I chose the breakpoints for the sticks at the values in the partial residual plot where I saw the trend of the graph change. For frequency this is at 7000 and for thickness this is at 0.04.



The changes to the partial residual plots are significant. For each of the four plots, there no longer appears to be a nonlinear relationship between the variables and the response. The green loess smoother follows a path very similar to the red ordinary least squares regression line. Based on R-squared, the predictions for this fit are much better than the predictions for the log model.

Discuss the pluses and minuses of the models in a) and b). Which is the best and how would you explain or deal with its imperfections?

SUMMARY FOR C) I believe that the broken stick model is better than the logarithmic model. Even though the broken stick model adds potentially confusing extra parameters, I believe there is a reasonable way to think about these extra parameters, and it does a better job at prediction.

The first advantages of the stick model is that it does a better job at predicting the response. Taking into account the two extra predictors, the adjusted R-squared for the stick model is 0.5544, while for the log model it is 0.5219. A disadvantage for both models is that they include a logarithmic transformation on the response variable. This effects the interpretation of the model since the regression coefficients and the error must be considered multiplicatively instead of additively. However, the logarithmic transformation is not neccessary for the stick model. (Furthermore, it isn't really necessary in the log model since it doesn't do a great job of improving normality.)

The big advantage of the log model (or the base model) is that it maintains the predictor variables as the experimenters designed them, which helps improve interpretability. The stick model adds an extra parameter for the frequency and the thickness variables, preventing someone from clearly understanding what relationship these variables have on the response. Furthermore, the extra parameters for these variables have coefficients with different signs, meaning that a certain point the effect of these variables on the response changes direction. This could be confusing.

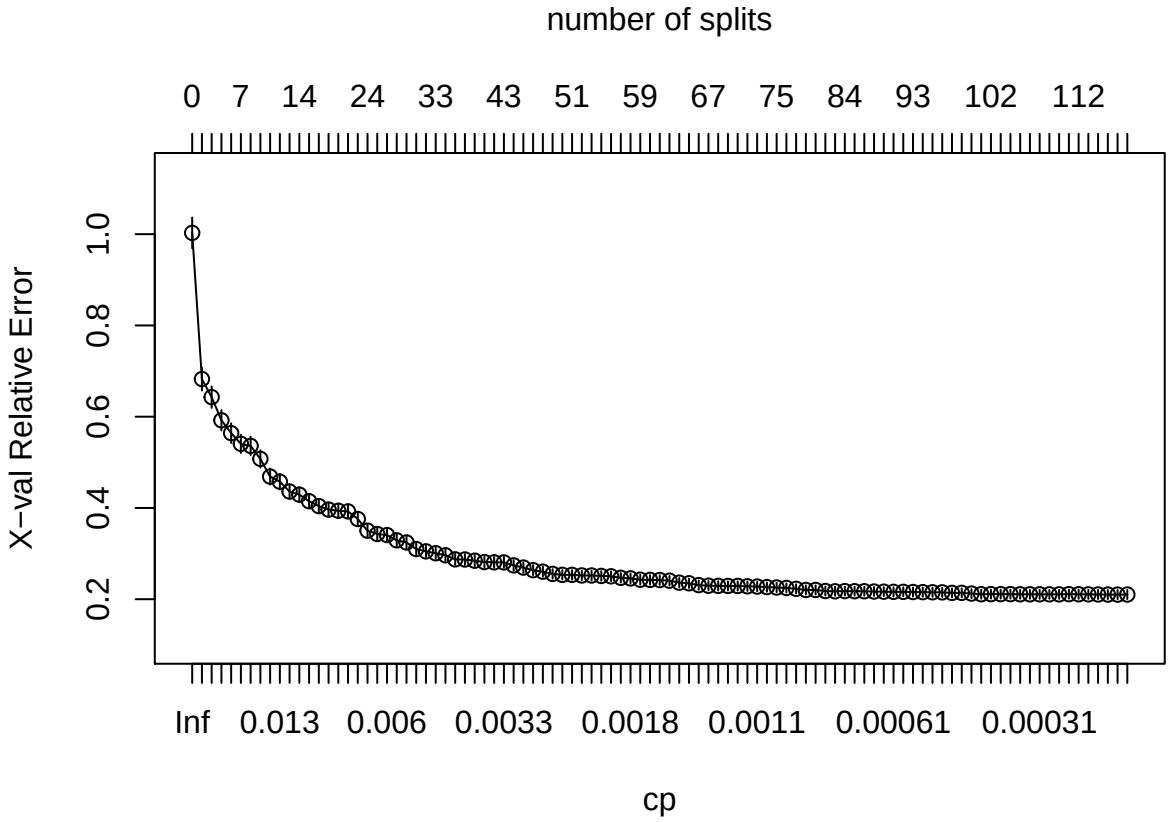
However, I would attempt to explain the extra parameters in the stick model by pointing to the categorical nature of the variables. In this designed experiment, even though there are 1503 observations, the number of unique observations in each predictor is small (21 for frequency and 105 for thickness). Since the predictor

variables are not continuous, using a broken stick model does not mean that at some arbitrary point the relationship of the variable changes. I chose the split points for each parameter so that we can interpret frequency as split into low and high frequency variables and we can interpret thickness as split into small and large thickness variables. While not a perfect explanation, I believe it is good enough to support my choice of the stick model as the preferred model.

Attempt now to fit a tree model. Decide what seems to be a good choice for the complexity parameter. With this complexity parameter, fit the best model and diagnose its residuals.

The complexity parameter (cp) represents the improvement in a tree's R-squared / degrees of freedom that must be exist for a tree to increse the number of splits. In order to choose the appropriate cp (i.e. the appropriate sized tree), we choose the value of cp that minimizes the crossvalidated error provided by rpart. Theoretically, a tree that is too large will overfit the data and result in poor cross validated predictions. Furthermore, since there is some error in the cross validated error across the K folds and since we prefer smaller trees, we choose the value of cp that has a cross validated error within one standard deviation of the minimum cross validated error.

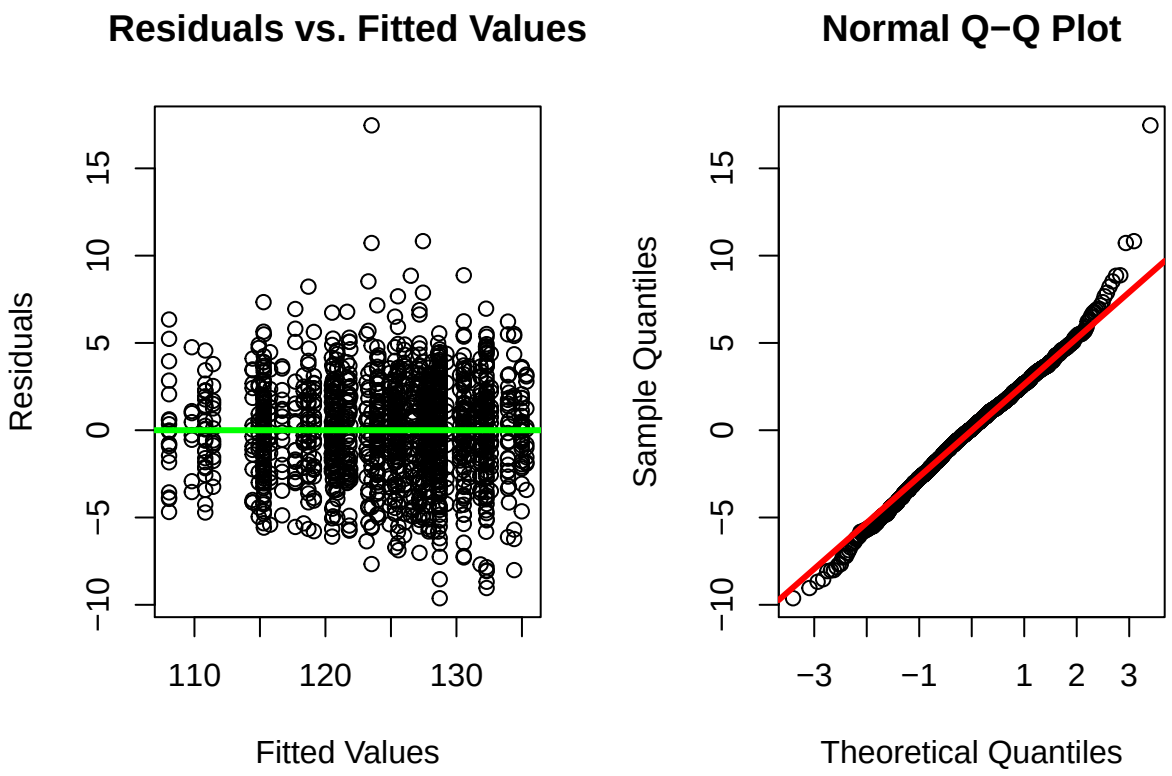
However, as seen in the graphs below, it appears like the cross validated error never increases. I will choose a very small complexity parameter of 0.0001 to start, so we can examine how the tree changes with cp. Later I will prune back the tree with a better value of cp.



As you can see, the cross validated error stops decreasing at a significant rate and the graph above flattens out around a cp value of 0.0011. This represents a pretty small increase in the R-squared of our model, so we choose to stop growing our tree here. Since the cross validated standard error is 0.0107 at this level, we use the one standard error rule and choose the largest cp with a cross validated error less than one standard deviation from our cross validated error. As a result we choose a cp of 0.00135.

Now we can prune back this tree to remove any splits that do not meet the threshold set by our complexity parameter of 0.00135.

Now that we have our best tree with chosen complexity parameter, we can diagnose its residuals.



The nonconstant variance that existed in the plot of the residuals for the least squares model has completely disappeared. The residuals for the tree model are symmetric and centered around zero. The residuals are also very close to normally distributed. Although there is one point that has an extremely high value, the rest of the points follow the normal distribution better than the least squares model.

Furthermore, the sum of the residuals for the tree model is about 11,000, while the sum of the residuals for the stick model is around 33,000. So the tree model does a much better job at prediction. (However, it does use significantly more degrees of freedom.)

Discuss dfferences between tree models and linear models in general. Do you see evidence of that here?

SUMMARY TO E) Generally, in terms of the bias-variance tradeoff, I can say that tree models have low bias but high variance, while linear models have high bias but low variance. However, we do not see evidence of high variance in this tree model. Linear models are generally more interpretible than tree models, especially when the trees get large, as in this case.

One of the big drawbacks to tree models is that they are prone to overfitting (i.e. high variance) as the size of the tree increases. In class we addressed the overfitting of trees by creating many trees and averaging them together (i.e. a random forest). However, I do not see evidence of overfitting in this tree model. As shown by the cross validation plot, the cross validated error is consistently decreases as the tree gets larger.

One of the drawbacks to linear models is that they are global models, with a single predictive formula applying over the entire data space. When the data has complicated, nonlinear interactions, assembling a global model can be difficult. This results in bias. I saw evidence of this in the nonlinearity of the two predictors that I fixed with a broken stick model. The simple linear model was unable to capture the nonlinear relationship. However, this is relatively easy for a tree model to do, since it can easily sub-divide features into different regions. For this problem, since the predictor variables are very categorical in nature, the tree was able to perform very well.

One of the reasons that I find tree models very appealing is that we can throw lots of predictors into the model without fear that they will ruin the model. The tree model is computed in a similar amount of time as the linear model, so we do not need to worry about the extra complexity. Some of the extra parameters may

explain the response well, and if not, they will be ignored by the tree. On the other hand in linear models, extra parameters will harm our model. Furthermore, trees are very robust to outliers, unlike linear models.

Finally, one drawback to the tree model is that they are unstable. A small change in the data being used can give very different trees. This is because the tree aims to optimize at each split, not over all splits (i.e. it is not globally optimal.) Because of this, if there is an error in one of the first splits, that error will propagate throughout the entire tree.