STAT 420 Final Project

Analyzing factors that contribute to prices of houses in Saratoga county, NY.

Group 30

Michael Garbus mgarbus2
Navyaa Sanan navyaas2
Chloe Streit cstreit2
Junseok Yang jyang247

Index

- Introduction
- Methods
- Results
- Discussion
- Appendix

Introduction

The data is information on houses in Saratoga County, NY USA, from 2006. It comes from DASL , and the variables are the following:

Description of dataset:

- price : price (US dollars)
- lotSize : size of lot (acres)
- age : age of house (years)
- landValue : value of land (US dollars)
- livingArea : living area (square feet)
- pctCollege : percent of neighborhood that graduated college
- bedrooms : number of bedrooms
- firplaces : number of fireplaces
- bathrooms : number of bathrooms (half bathrooms have no shower or tub)
- rooms : number of rooms
- heating : type of heating system
- fuel : fuel used for heating
- sewer : type of sewer system
- waterfront : whether property includes waterfront
- newConstruction : whether the property is a new construction
- centralAir : whether the house has central air

Reasons for interest in this data set:

This analysis would be helpful in that it provides the reader an idea of the living environment in Saratoga, NY. The analysis will provide potential home buyers with information pertinent to the process of choosing a home. Oftentimes, potential buyers have a specific budget and want to ensure that they can find all of the home features within that budget in the area where they are seeking; consequently, an analysis quickly provides all of this information. Our main goal is to help people predict a house price in Saratoga county, NY with a model we make based on the dataset. It may help people to have an idea (see the trend) whether the type of a house they are looking for is higher or lower than the average price we predict based on our model and make a decision.

Dataset

```
library(readxl)
house = read_xlsx("data.xlsx")
house
```

```
## # A tibble: 1,728 x 16
##     price lotSize   age landValue livingArea pctCollege bedrooms fireplaces
##     <dbl>   <dbl> <dbl>     <dbl>      <dbl>      <dbl>    <dbl>      <dbl>
##  1 132500    0.09    42     50000        906         35        2          1
##  2 181115    0.92     0     22300       1953         51        3          0
##  3 109000    0.19   133      7300       1944         51        4          1
##  4 155000    0.41    13     18700       1944         51        3          1
##  5  86060    0.11     0     15000        840         51        2          0
##  6 120000    0.68    31     14000       1152         22        4          1
##  7 153000    0.4     33     23300       2752         51        4          1
##  8 170000    1.21    23     14600       1662         35        4          1
##  9  90000    0.83    36     22200       1632         51        3          0
## 10 122900    1.94     4     21200       1416         44        3          0
## # ... with 1,718 more rows, and 8 more variables: bathrooms <dbl>, rooms <dbl>,
## #   heating <chr>, fuel <chr>, sewer <chr>, waterfront <chr>,
## #   newConstruction <chr>, centralAir <chr>
```

Methodology

Modeling

Splitting our data into training and testing subsets

In order to later verify the model, we split the dataset so we can calculate the predictive power of the model we come up with.

```
trn_idx  = sample(nrow(house), size = trunc(0.6 * nrow(house)))
trn_data = house[trn_idx, ]
tst_data = house[-trn_idx, ]
```

- Training data uses 1036 data points and testing data uses 692 data points. This is a 60/40 split.
- We first decided to start off with a model based off of only the significant predictors found in the full model, to focus on modeling the signal, not the noise.

```r
full_house_mod = lm(price ~ ., data = trn_data)
summary(full_house_mod)
```

```
##
## Call:
## lm(formula = price ~ ., data = trn_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -227570  -36430   -4303   27693  457112
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -1.205e+04  2.560e+04  -0.471   0.6380
## lotSize                 7.398e+03  3.043e+03   2.432   0.0152 *
## age                    -4.378e+00  8.711e+01  -0.050   0.9599
## landValue               8.619e-01  6.552e-02  13.156  < 2e-16 ***
## livingArea              7.239e+01  6.544e+00  11.063  < 2e-16 ***
## pctCollege             -6.685e+01  2.020e+02  -0.331   0.7407
## bedrooms               -8.749e+03  3.510e+03  -2.492   0.0129 *
## fireplaces              2.664e+03  3.987e+03   0.668   0.5042
## bathrooms               2.328e+04  4.779e+03   4.872 1.28e-06 ***
## rooms                   2.637e+03  1.312e+03   2.009   0.0448 *
## heatinghot air          6.993e+03  1.587e+04   0.441   0.6595
## heatinghot water/steam -5.260e+03  1.663e+04  -0.316   0.7519
## fuelgas                 4.672e+03  1.565e+04   0.299   0.7653
## fueloil                 1.368e+03  1.659e+04   0.083   0.9343
## sewerpublic/commercial  2.277e+04  2.192e+04   1.039   0.2993
## sewerseptic             2.041e+04  2.203e+04   0.927   0.3543
## waterfrontYes           1.300e+05  1.901e+04   6.839 1.38e-11 ***
## newConstructionYes     -3.938e+04  9.574e+03  -4.113 4.22e-05 ***
## centralAirYes           7.911e+03  4.696e+03   1.685   0.0924 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60800 on 1017 degrees of freedom
## Multiple R-squared:  0.6199, Adjusted R-squared:  0.6132
## F-statistic: 92.15 on 18 and 1017 DF,  p-value: < 2.2e-16
```

```r
faraway::vif(full_house_mod)
```

```
##                lotSize                    age             landValue
##               1.262489               1.590650               1.462846
##             livingArea             pctCollege               bedrooms
##               4.363178               1.249026               2.269915
##             fireplaces              bathrooms                  rooms
##               1.402615               2.647205               2.552661
##         heatinghot air heatinghot water/steam               fuelgas
##              16.075497              10.532534              15.081028
##                fueloil sewerpublic/commercial            sewerseptic
##               8.680584              28.182478              28.034707
##          waterfrontYes     newConstructionYes          centralAirYes
##               1.063351               1.268869               1.415259
```

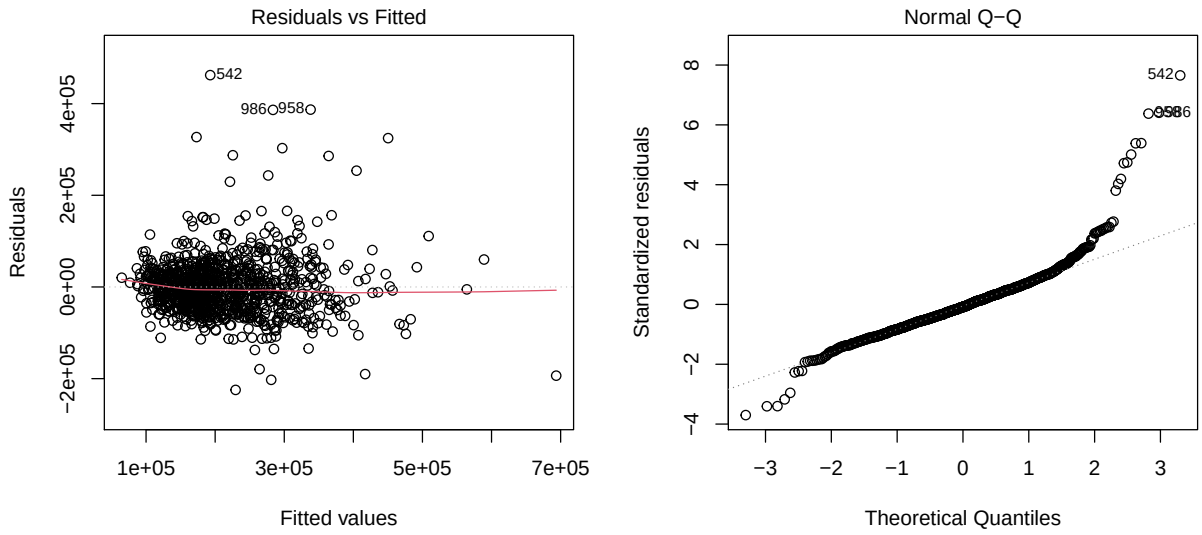
- It looks like `sewer`, `fuel` and `heating` have a high VIF. This makes sense, as they are dummy variables. The other variance inflation factors are all below 5, so we do not need to clean the data before using a model. Our first model is as follows:

```
first_model = lm(price ~ age + landValue + lotSize + livingArea + pctCollege + bedrooms + bathrooms + r
summary(first_model)
```

```
##
## Call:
## lm(formula = price ~ age + landValue + lotSize + livingArea +
##     pctCollege + bedrooms + bathrooms + rooms + waterfront +
##     newConstruction + centralAir, data = trn_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -224601  -35112   -5220   28666  462129
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.254e+04  1.272e+04   0.986 0.324501
## age                -1.750e+01  8.128e+01  -0.215 0.829543
## landValue           8.730e-01  6.449e-02  13.538  < 2e-16 ***
## lotSize             6.049e+03  2.778e+03   2.178 0.029639 *
## livingArea          7.324e+01  6.394e+00  11.454  < 2e-16 ***
## pctCollege         -6.916e+00  1.969e+02  -0.035 0.971984
## bedrooms           -9.134e+03  3.475e+03  -2.628 0.008715 **
## bathrooms           2.340e+04  4.661e+03   5.020 6.09e-07 ***
## rooms               2.782e+03  1.307e+03   2.129 0.033507 *
## waterfrontYes       1.296e+05  1.887e+04   6.867 1.14e-11 ***
## newConstructionYes -3.630e+04  9.460e+03  -3.837 0.000132 ***
## centralAirYes       1.226e+04  4.430e+03   2.767 0.005757 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60860 on 1024 degrees of freedom
## Multiple R-squared:  0.6166, Adjusted R-squared:  0.6125
## F-statistic: 149.7 on 11 and 1024 DF,  p-value: < 2.2e-16
```

- We received a pretty good Adjusted R-Squared, of 0.6125091. Maybe we could get a higher Adjusted R-Squared if we performed a transformation.

```
par(mfrow = c(1,2))
plot(first_model, which = 1)
plot(first_model, which = 2)
```

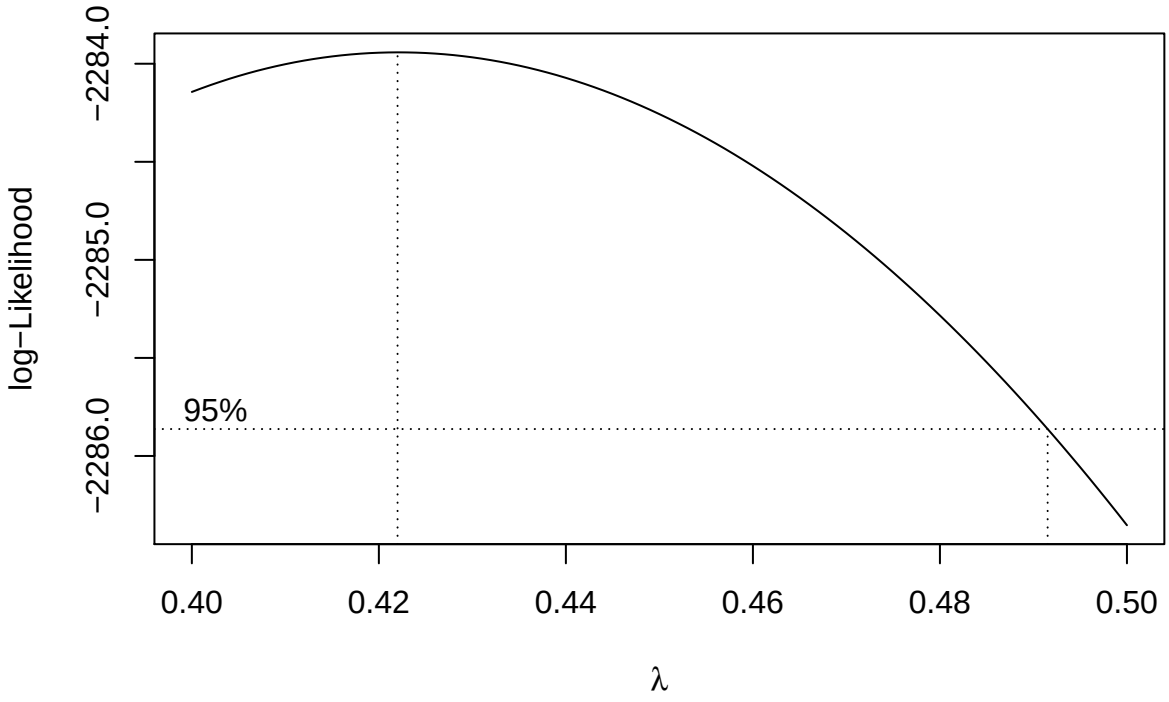



```
shapiro.test(resid(first_model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(first_model)
## W = 0.88846, p-value < 2.2e-16
```

- Observing the Q-Q Plot, we can see that this represents over-dispersed data indicating that there should be adjustments made to this model. As we can see from the Shapiro-Wilk test, the model is not normal. We used the Box-Cox method to decide how to transform the response, price:

```
library(MASS)
library(faraway)
boxcox(first_model, plotit = T, lambda = seq(0.4,0.5,0.001))
```

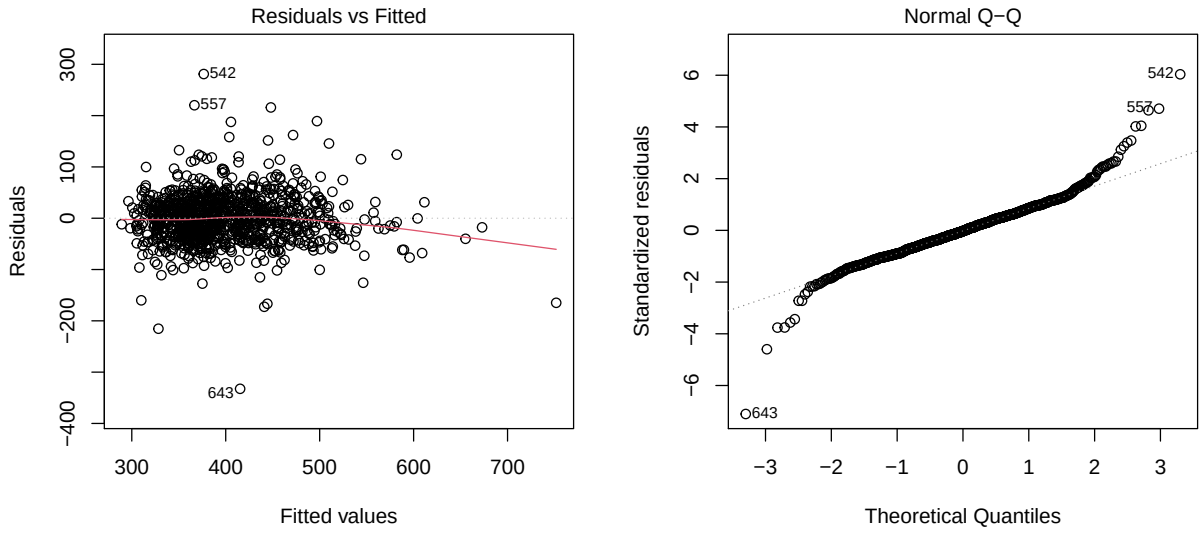



```
log_model = lm(((price^0.42) - 1)/0.42 ~ age + landValue + lotSize + livingArea + pctCollege
               + bedrooms + bathrooms + rooms + waterfront
               + newConstruction + centralAir, data = trn_data)
summary(log_model)
```

```
##
## Call:
## lm(formula = ((price^0.42) - 1)/0.42 ~ age + landValue + lotSize +
##     livingArea + pctCollege + bedrooms + bathrooms + rooms +
##     waterfront + newConstruction + centralAir, data = trn_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -332.52  -28.25   -1.31   26.34  281.07
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         2.333e+02  9.812e+00  23.771  < 2e-16 ***
## age                -7.592e-02  6.269e-02  -1.211 0.226147
## landValue           6.149e-04  4.974e-05  12.364  < 2e-16 ***
## lotSize             5.512e+00  2.142e+00   2.573 0.010220 *
## livingArea          5.340e-02  4.932e-03  10.829  < 2e-16 ***
## pctCollege          1.253e-01  1.518e-01   0.825 0.409405
## bedrooms           -2.116e+00  2.680e+00  -0.789 0.430111
## bathrooms           1.911e+01  3.595e+00   5.317 1.30e-07 ***
```

```
## rooms               1.544e+00  1.008e+00   1.532 0.125805
## waterfrontYes       9.875e+01  1.455e+01   6.786 1.95e-11 ***
## newConstructionYes -2.468e+01  7.296e+00  -3.383 0.000744 ***
## centralAirYes       9.370e+00  3.417e+00   2.742 0.006210 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 46.94 on 1024 degrees of freedom
## Multiple R-squared:  0.6129, Adjusted R-squared:  0.6088
## F-statistic: 147.4 on 11 and 1024 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(1,2))
plot(log_model, which = 1)
plot(log_model, which = 2)
```




```
shapiro.test(resid(log_model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(log_model)
## W = 0.94819, p-value < 2.2e-16
```

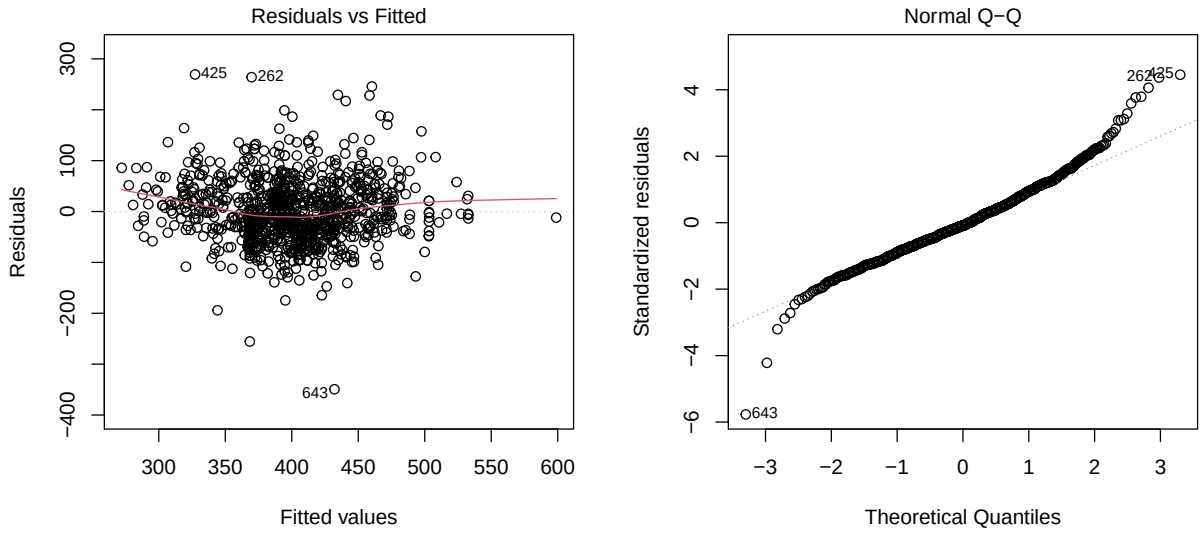
- There has been an improvement from the first_model to the log_model; however, it is still indicating that the data is over-dispersed and has positive excess kurtosis, just in a lesser extent from the previous model.
- We still have a very low Shapiro-Wilk test P-value. We decided to try logarithmizing the predictors too.

Added new variables for logarithmization

```
trn_data["age_log"] = log(trn_data$age)
trn_data["age_log"][trn_data["age_log"] < 0] = 0
trn_data["loglotSize"] = log(trn_data$lotSize)
trn_data["loglotSize"][trn_data["loglotSize"] < 0] = 0
```

```
log_log_model = lm(((price^0.42) - 1)/0.42 ~ log(age + landValue + lotSize + livingArea
                                                 + (pctCollege) + bedrooms + bathrooms
                                                 + rooms) + waterfront + newConstruction
                   + centralAir, data = trn_data)

par(mfrow = c(1,2))
plot(log_log_model, which = 1)
plot(log_log_model, which = 2)
```

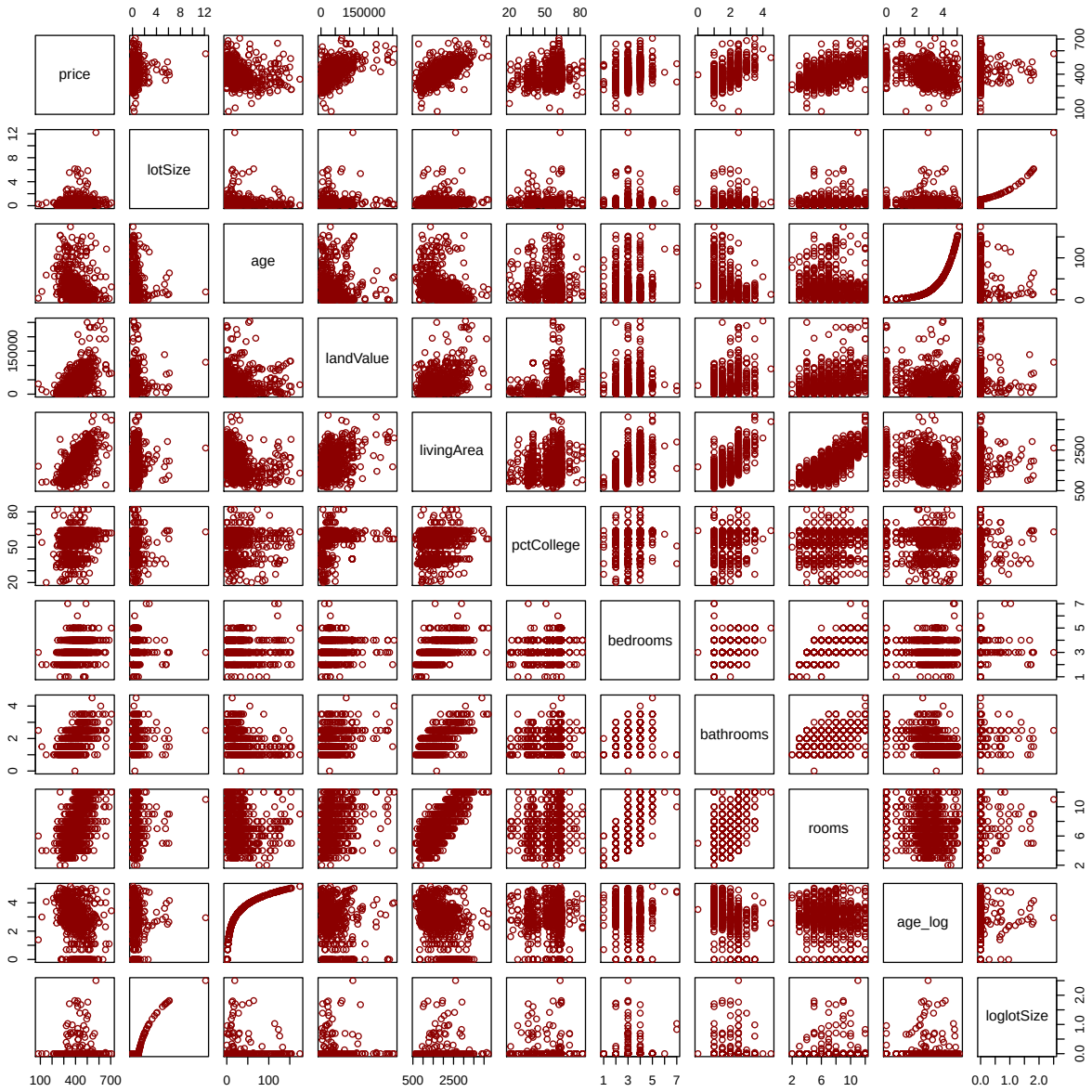



```
shapiro.test(resid(log_log_model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(log_log_model)
## W = 0.97346, p-value = 7.857e-13
```

- The `log_log_model` residual plot indicates that this plot has started to get closer to being normally distributed, however improvement is needed to better fit more expensive homes. It looks like it helped a bit, but not as much as we were expecting. Let's try looking at the pairs of data.

```
house_df = data.frame(trn_data)
house_df$price = ((house_df$price^0.42) - 1)/0.42
house_df = subset(house_df, select = -c(fireplaces, heating,fuel,sewer,waterfront,newConstruction,centra
pairs(house_df, col = "darkred")
```



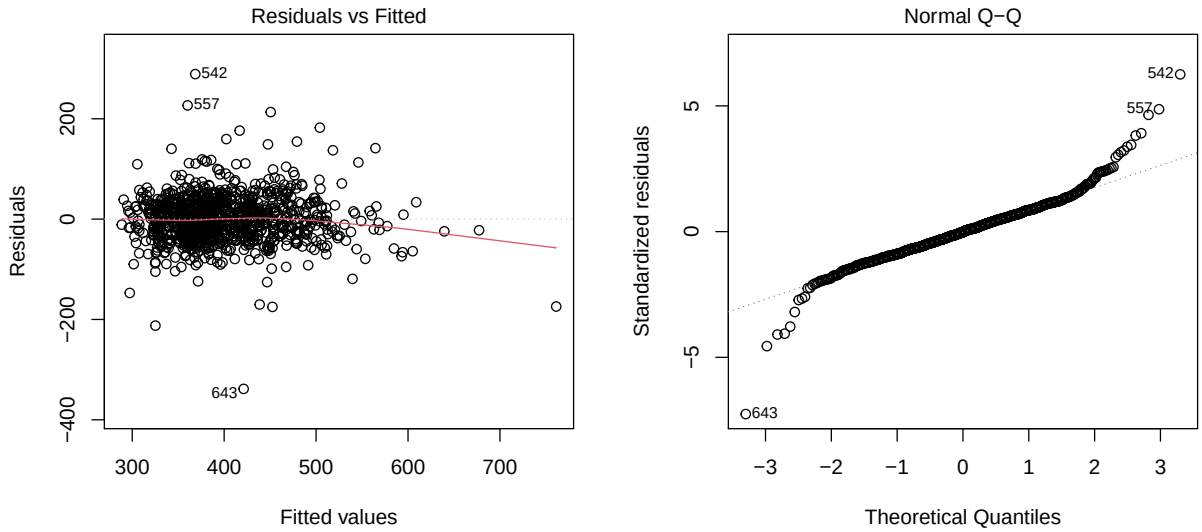
It looks like `lotSize` and `pctCollege` could benefit from transforming. We can also include a quadratic interaction term for `age`.

```
log_int_model = lm(((price^0.42) - 1)/0.42 ~ (age*livingArea + landValue + loglotSize
                                              + age_log + log(pctCollege)
                                              + bedrooms*livingArea + bathrooms*livingArea
                                              + (rooms*livingArea) + I(age^2)) + waterfront
                   + newConstruction + centralAir, data = trn_data)
shapiro.test(resid(log_int_model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(log_int_model)
```

```
## W = 0.94618, p-value < 2.2e-16
```

```
par(mfrow = c(1,2))
plot(log_int_model, which = 1)
plot(log_int_model, which = 2)
```



- The above Residuals vs Fitted plot illustrates that the errors are normal and centered around 0 which is a favorable result. The Q-Q plot looks nice as well.
- We then had to consider if any variables need to be transformed. We exclude the categorical variables, as they would not benefit from transformation.

We can use ANOVA to see if these terms add predictive power compared to standard model.

```
anova(log_model, log_int_model)
```

```
## Analysis of Variance Table
##
## Model 1: ((price^0.42) - 1)/0.42 ~ age + landValue + lotSize + livingArea +
##     pctCollege + bedrooms + bathrooms + rooms + waterfront +
##     newConstruction + centralAir
## Model 2: ((price^0.42) - 1)/0.42 ~ (age * livingArea + landValue + loglotSize +
##     age_log + log(pctCollege) + bedrooms * livingArea + bathrooms *
##     livingArea + (rooms * livingArea) + I(age^2)) + waterfront +
##     newConstruction + centralAir
##   Res.Df     RSS Df Sum of Sq     F  Pr(>F)
## 1   1024 2255988
## 2   1018 2229613  6     26375 2.007 0.06208 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(log_model)$"adj"
```

```
## [1] 0.6087916
```

```
summary(log_int_model)$"adj"
```

```
## [1] 0.6110864
```

- When running this ANOVA test, we can find out that the p-value is fairly low. This means that we have very strong evidence against the null, indicating that there is at least some linear relationship between the response variable `((price^0.42) - 1)/0.42)` and additional predictors that the larger model `log_int_model` have. Also, the adjusted r^2 values went 0.2294782% up, although it is not significantly high, we might say that some of the larger model's additional predictors have a higher chance of contributing more predictive power to this model.

We now update the test dataset and prepare for further comparison.

Updating the test dataset

```
tst_data["age_log"] = log(tst_data$age)
tst_data["age_log"][tst_data["age_log"] < 0] = 0
tst_data["loglotSize"] = log(tst_data$lotSize)
tst_data["loglotSize"][tst_data["loglotSize"] < 0] = 0
```

```
rmse = function(actual, predicted) {
  sqrt(mean((actual - predicted) ^ 2))
}
```

Testing the model

```
final = log_int_model

predictions = predict((final), tst_data)
predictions = (predictions[1:length(predictions)] * 0.42 + 1)^(1/0.42)

train_error = c(rmse(trn_data$price, predict(final, trn_data)))
test_error = c(rmse(tst_data$price, predictions))


p_e = sum(abs(predictions - tst_data$price)/predictions * 100)/nrow(tst_data)
```

```
final = log_log_model

predictions = predict((final), tst_data)
predictions = (predictions[1:length(predictions)] * 0.42 + 1)^(1/0.42)

ltrain_error = c(rmse(trn_data$price, predict(final, trn_data)))
```

```
ltest_error = c(rmse(tst_data$price, predictions))



lp_e = sum(abs(predictions - tst_data$price)/predictions * 100)/nrow(tst_data)
```

```
final = log_model

predictions = predict((final), tst_data)
predictions = (predictions[1:length(predictions)] * 0.42 + 1)^(1/0.42)

lmtrain_error = c(rmse(trn_data$price, predict(final, trn_data)))
lmtest_error = c(rmse(tst_data$price, predictions))



lmp_e = sum(abs(predictions - tst_data$price)/predictions * 100)/nrow(tst_data)
```

- The `log_int_model` got a training RMSE of 2.3322914×10^5 and a testing RMSE of 5.6243484×10^4. It also got a percent error of 19.1181015%. This is better than our `log_log_model`'s results of 2.332377×10^5, 7.8255582×10^4, and 27.5915887%. We found that when comparing our `log_int_model` to our original `log_model`, that the `log_model` had a percent error of 19.12375% (which is sometimes better than the `log_int_model` percent error due to random chance), a similar training RMSE of 2.3322927×10^5, and a larger testing RMSE of 5.6324063×10^4. Although there is a occasionally a lower percent error, because of the higher testing RMSE and poorer diagnostics, we decided to choose `log_int_model` to represent our data.

Results

- The data has indicated that the `log_int_model` model is most favorable. After comparing diagnostics of previous models, as well as comparing RMSEs and percent errors, we decided to use `log_int_model`. While the model has a preferable percent error and RMSE output, as well as a preferable Q-Q plot output, our Shapiro-Wilk test continues to have a low P-value. This means that it is unlikely that our data is being sampled from a normal distribution. When this assumption fails to be met, our coefficient estimates could be a little less reliable than our SE values suggest, and we might have some bias in predicting individual values.
- We also compared this model to a few different models found through AIC and BIC. The `log_int_model` model proved to have the best overall results regarding the percentage error and RMSE scores. The RMSE, LOOCV and percentage error for models found using AIC or BIC (`backstep`, `astep` and `fstep`), as well as informative graphs regarding the models are included in the appendix.

Summary Information of Model

Since the model has more than 15 predictors we included a summary of the coefficents of our preferred model below:

```
summary(log_int_model)$coefficients
```

```
##                          Estimate   Std. Error    t value     Pr(>|t|)
```

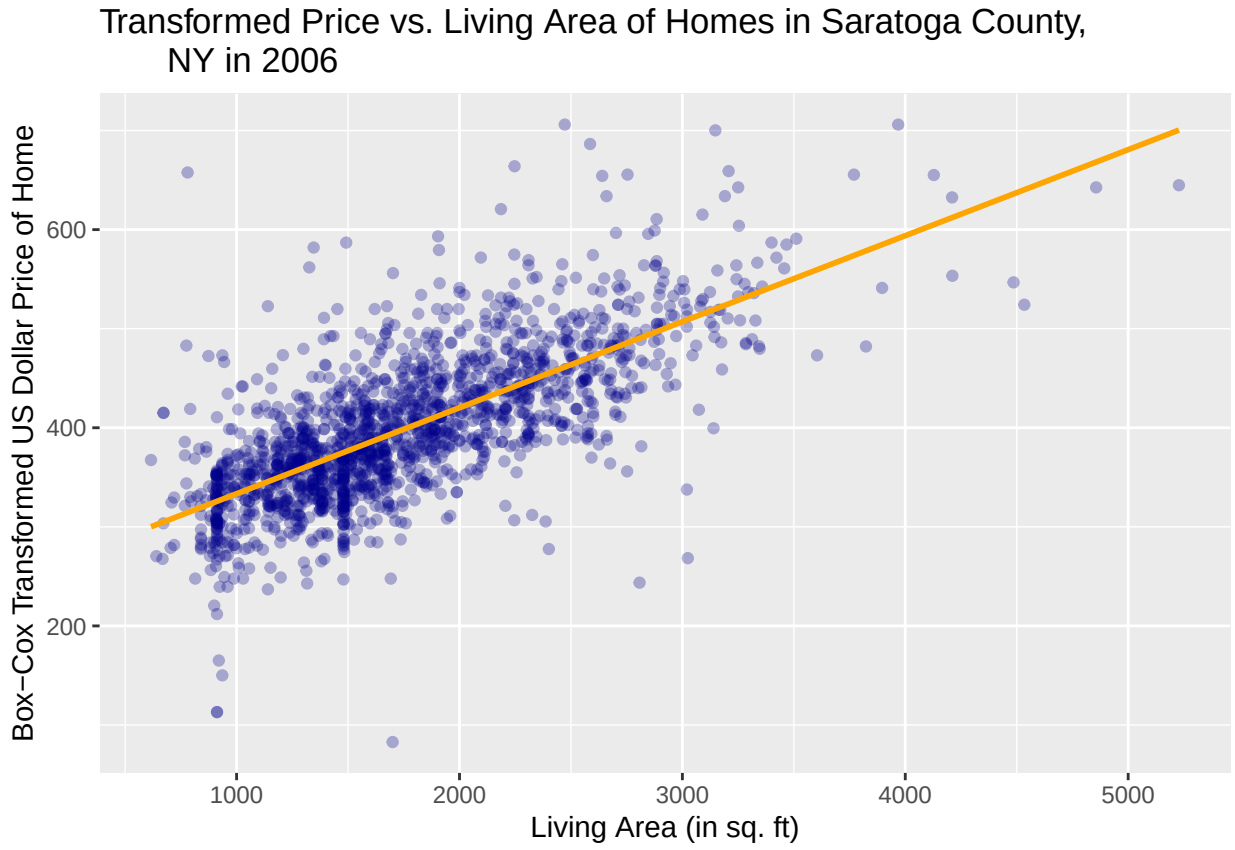
```
## (Intercept)           1.914076e+02 3.398774e+01  5.6316659 2.308147e-08
## age                  -3.922685e-01 3.448006e-01 -1.1376677 2.555270e-01
## livingArea            6.725587e-02 1.240372e-02  5.4222322 7.351365e-08
## landValue             6.031040e-04 5.190827e-05 11.6186483 2.159780e-29
## loglotSize            1.736075e+01 7.088128e+00  2.4492718 1.448221e-02
## age_log              -7.168562e-01 3.793138e+00 -0.1889876 8.501402e-01
## log(pctCollege)       9.924729e+00 7.307008e+00  1.3582480 1.746859e-01
## bedrooms              5.190015e+00 7.655840e+00  0.6779158 4.979791e-01
## bathrooms             1.154808e+01 9.780745e+00  1.1806954 2.379995e-01
## rooms                 2.692914e+00 2.838037e+00  0.9488649 3.429145e-01
## I(age^2)              3.725471e-03 1.951073e-03  1.9094472 5.648531e-02
## waterfrontYes         1.016657e+02 1.460803e+01  6.9595789 6.104859e-12
## newConstructionYes   -3.206353e+01 8.759654e+00 -3.6603644 2.647932e-04
## centralAirYes         7.915748e+00 3.423501e+00  2.3121792 2.096660e-02
## age:livingArea       -8.255584e-05 1.068539e-04 -0.7726046 4.399358e-01
## livingArea:bedrooms  -3.628501e-03 4.148551e-03 -0.8746430 3.819745e-01
## livingArea:bathrooms  3.171840e-03 5.048884e-03  0.6282259 5.299969e-01
## livingArea:rooms     -7.546913e-04 1.536119e-03 -0.4912975 6.233219e-01
```

The model has a R-squared value of 0.6174743, saying that around 61.7474342% of the variance in the model is explained by the predictors. The adjusted R-squared is 0.6110864, meaning that around 61.1086389% of the correlation is explained by more than random chance.

Graphing of Strongest Predictors

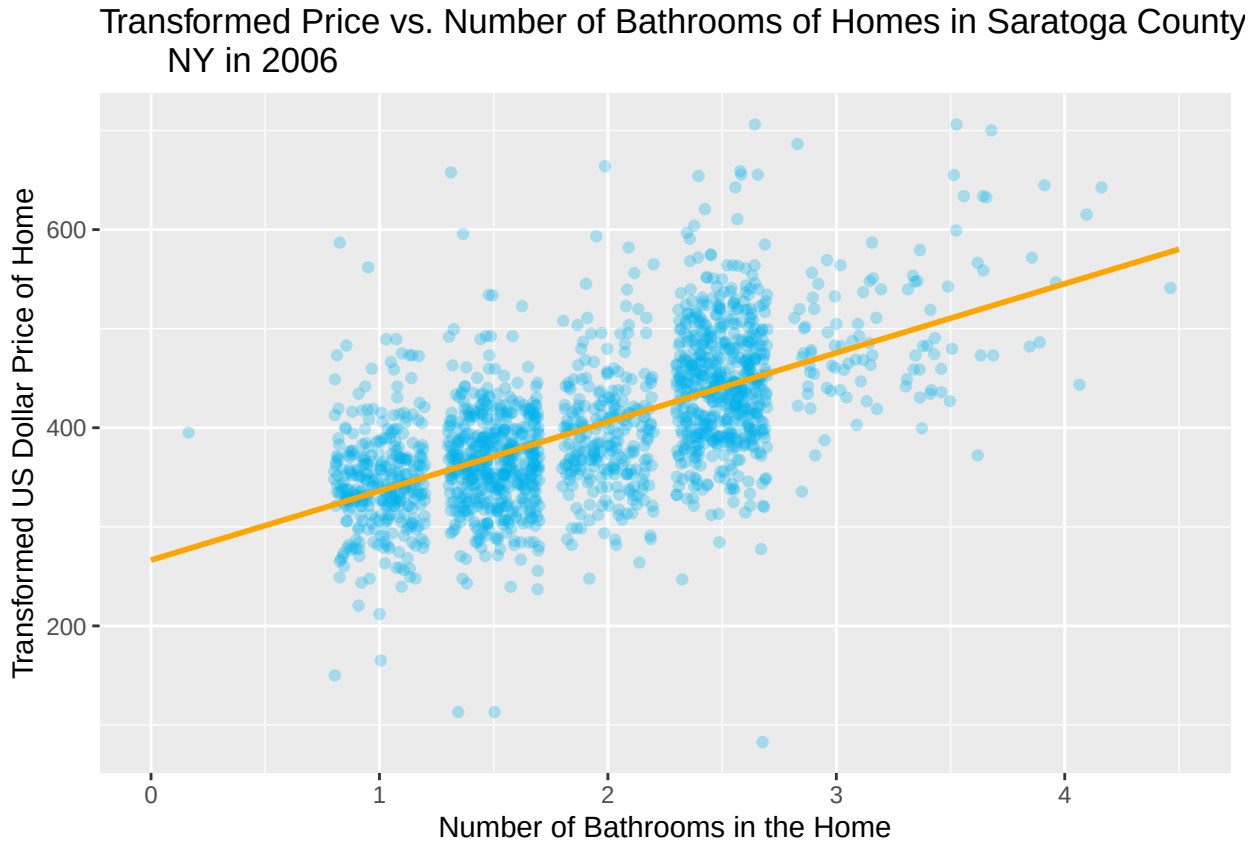
```
library(ggplot2)
ggplot(house, aes(y=((price^0.42) - 1)/0.42, x=livingArea))+
  geom_jitter(color = 'darkblue', alpha = 0.3)+
  labs(y="Box-Cox Transformed US Dollar Price of Home",x="Living Area (in sq. ft)",
       title = "Transformed Price vs. Living Area of Homes in Saratoga County,
       NY in 2006") +
  geom_smooth(method = "lm", se = F, col = "orange")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



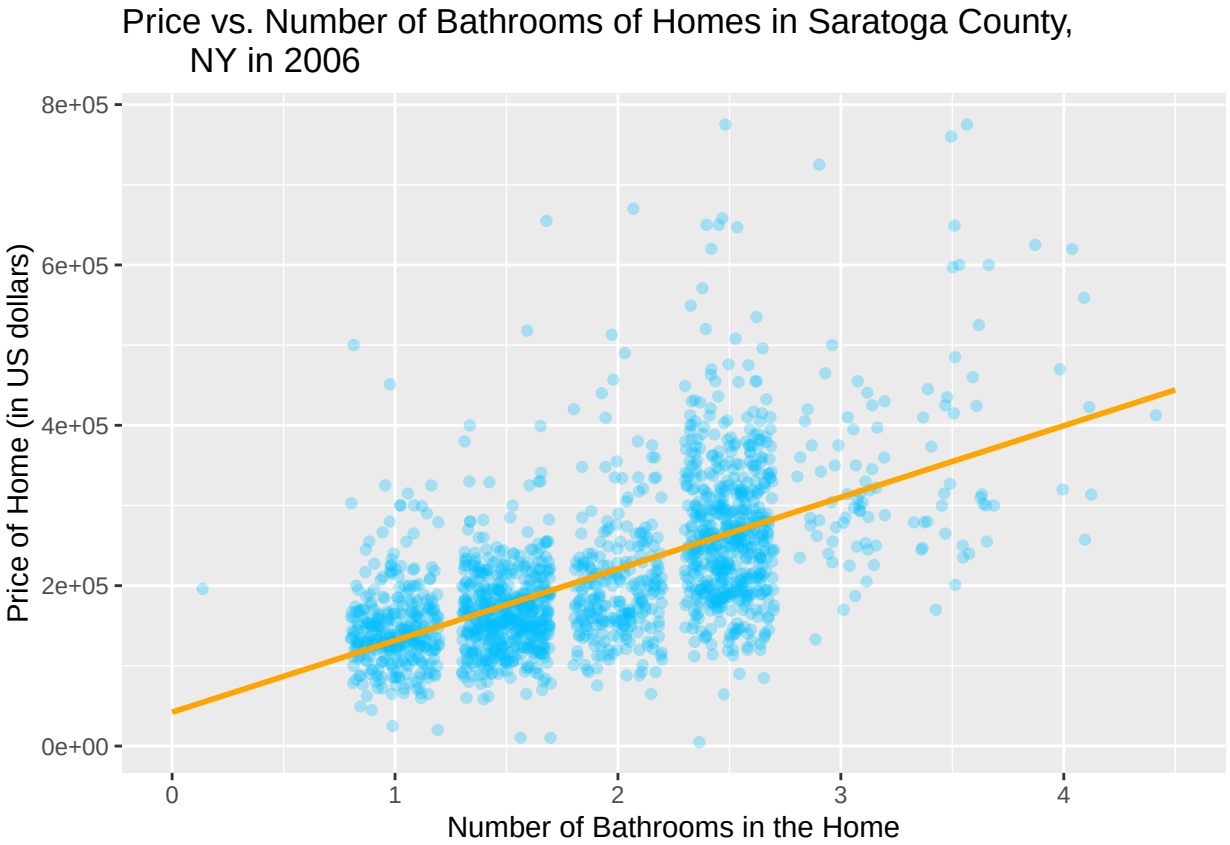

```
ggplot(house, aes(y=((price^0.42) - 1)/0.42,
                  x=bathrooms))+geom_jitter(color='deepskyblue2',alpha=0.3)+
  labs(y="Transformed US Dollar Price of Home",
       x="Number of Bathrooms in the Home",
       title = "Transformed Price vs. Number of Bathrooms of Homes in Saratoga County,
       NY in 2006") +
  geom_smooth(method = "lm", se = F, col = "orange")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
ggplot(house, aes(y=price,
                  x=bathrooms))+geom_jitter(color='deepskyblue1',alpha=0.3)+
  labs(y="Price of Home (in US dollars)",
       x="Number of Bathrooms in the Home",
       title = "Price vs. Number of Bathrooms of Homes in Saratoga County,
       NY in 2006") +
  geom_smooth(method = "lm", se = F, col = "orange")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



In the first plot above, we can see that most homes have a living area between 1000 and 3000 acres. There are a few properties that are outliers in this situation; however, it appears that a majority of the observations are clustered and consistent in the homes that have a transformed price between 300 and 500.

The scatterplot comparing home prices and the number of bathrooms in a home shows a strong positive linear relationship. The more expensive homes do not closely follow this relationship, but given context in how some homes have valuation that is based on subjective principles, such as a Frank Lloyd Wright designed property - it could have 1 and a half baths but still be priced above 600,000 dollars because of its historical and cultural significance. The transformation of price seems to handle these outliers better, however, some outliers are still present.

Discussion

- Our results indicate that there were several models that were adequate in representing the data; nonetheless, the `log_int model` presented as the model to best fit our dataset. The final model is useful in predicting pricing of homes in the Saratoga County, NY. Predicting pricing for these homes is relevant to real estate brokers, potential home buyers, home sellers trying to decide on a marketable price, and many more individuals with interest in this market. One of the areas that has limited the success of the model is the difficulty in pricing expensive houses. There also may be some bias present in the model due to

Appendix

Here are some models we considered but did not go through with.

As we are trying to predict the price of a house rather than explain the price of a house, we first decided to use AIC before BIC, as AIC prefers more complex models.

```
step(full_house_mod, direction = "backward",k = 2, trace = 0)
```

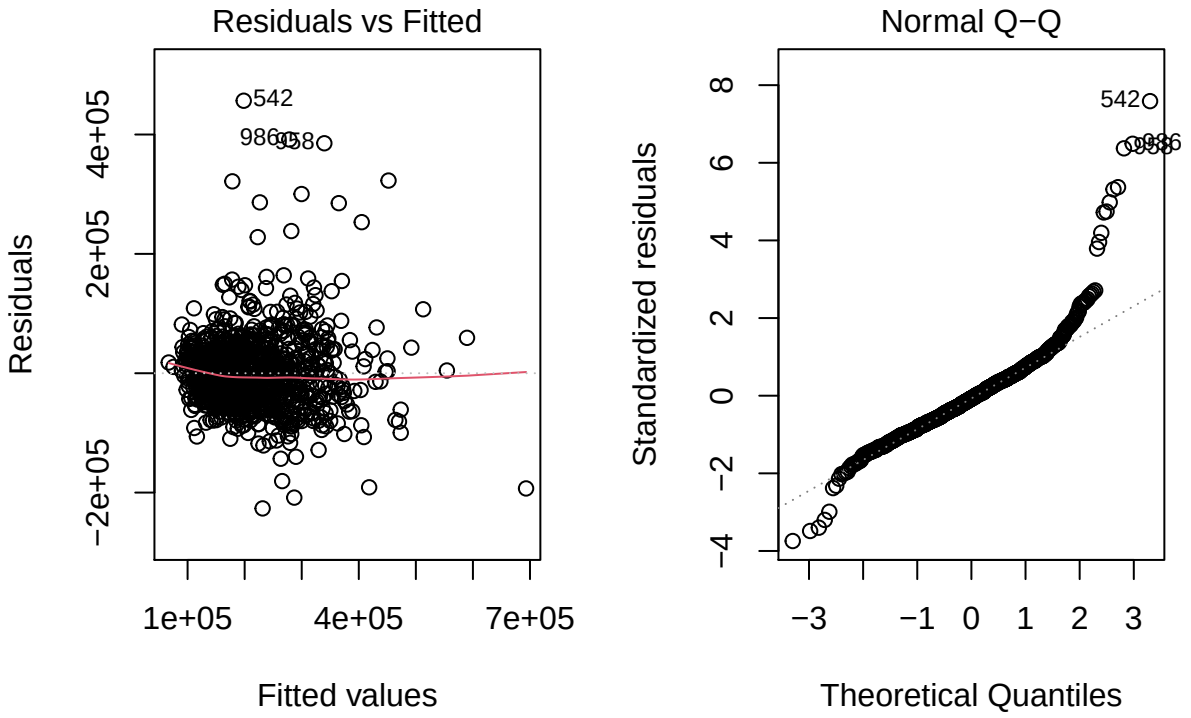
```
##
## Call:
## lm(formula = price ~ lotSize + landValue + livingArea + bedrooms +
##     bathrooms + rooms + heating + waterfront + newConstruction +
##     centralAir, data = trn_data)
##
## Coefficients:
##            (Intercept)                 lotSize               landValue
##              6.066e+03               6.232e+03               8.674e-01
##             livingArea                bedrooms               bathrooms
##              7.271e+01              -9.053e+03               2.437e+04
##                  rooms          heatinghot air  heatinghot water/steam
##              2.719e+03               1.031e+04              -1.523e+03
##          waterfrontYes      newConstructionYes           centralAirYes
##              1.291e+05              -3.872e+04               8.595e+03
```

```
astep = step(full_house_mod, direction = "backward",k = 2, trace = 0)
summary(full_house_mod)
```

```
##
## Call:
## lm(formula = price ~ ., data = trn_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -227570  -36430   -4303   27693  457112
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -1.205e+04  2.560e+04  -0.471   0.6380
## lotSize                   7.398e+03  3.043e+03   2.432   0.0152 *
## age                      -4.378e+00  8.711e+01  -0.050   0.9599
## landValue                 8.619e-01  6.552e-02  13.156  < 2e-16 ***
## livingArea                7.239e+01  6.544e+00  11.063  < 2e-16 ***
## pctCollege               -6.685e+01  2.020e+02  -0.331   0.7407
## bedrooms                 -8.749e+03  3.510e+03  -2.492   0.0129 *
## fireplaces                2.664e+03  3.987e+03   0.668   0.5042
## bathrooms                 2.328e+04  4.779e+03   4.872 1.28e-06 ***
## rooms                     2.637e+03  1.312e+03   2.009   0.0448 *
## heatinghot air            6.993e+03  1.587e+04   0.441   0.6595
## heatinghot water/steam   -5.260e+03  1.663e+04  -0.316   0.7519
## fuelgas                   4.672e+03  1.565e+04   0.299   0.7653
## fueloil                   1.368e+03  1.659e+04   0.083   0.9343
## sewerpublic/commercial    2.277e+04  2.192e+04   1.039   0.2993
## sewerseptic               2.041e+04  2.203e+04   0.927   0.3543
## waterfrontYes             1.300e+05  1.901e+04   6.839 1.38e-11 ***
## newConstructionYes       -3.938e+04  9.574e+03  -4.113 4.22e-05 ***
## centralAirYes             7.911e+03  4.696e+03   1.685   0.0924 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 60800 on 1017 degrees of freedom
## Multiple R-squared:  0.6199, Adjusted R-squared:  0.6132
## F-statistic: 92.15 on 18 and 1017 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(1,2))
plot(astep, which = 1)
plot(astep, which = 2)
```



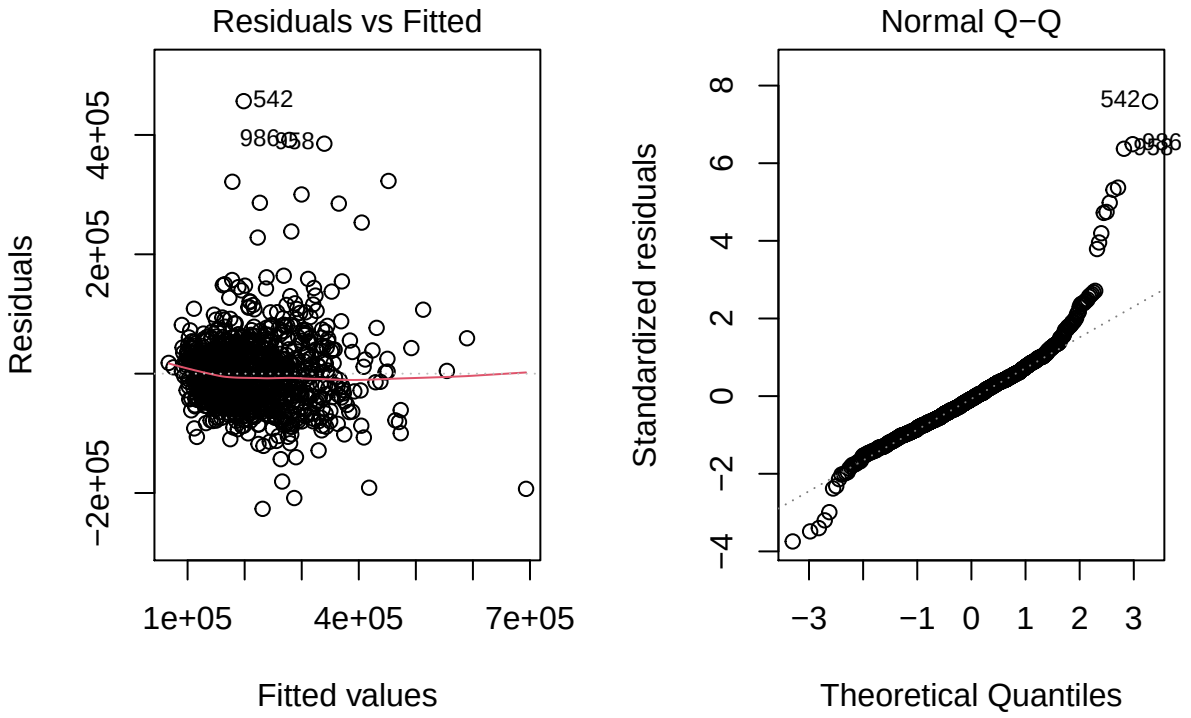
- Our first `AIC` result produced a model that predicted price from lotSize, age, landValue, livingArea, bedrooms, bathrooms, rooms, fuel, waterfront, newConstruction, and centralAir.
- However, we want to be sure if we are producing a model with the best predictive power. We will use a forward search and a stepwise search.

```
empty_house_mod = lm(price ~ 1, data = trn_data)
#consider fullest model
step(empty_house_mod, direction = "forward",scope = price ~ lotSize + age + landValue + livingArea + pc
    bedrooms + bathrooms + rooms + heating + fuel + sewer +  waterfront + newConstruction +
    centralAir, k = 2, trace = 0)
```

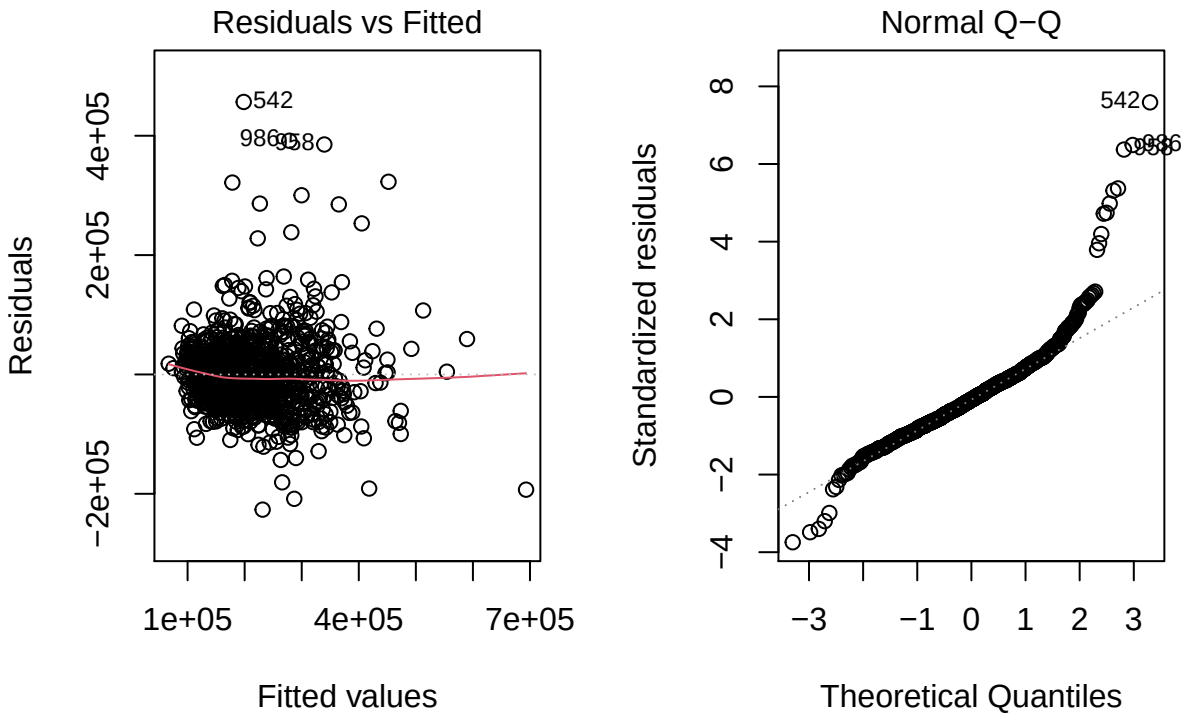
```
##
## Call:
## lm(formula = price ~ livingArea + landValue + waterfront + bathrooms +
##     newConstruction + heating + lotSize + bedrooms + rooms +
```

```
##     centralAir, data = trn_data)
##
## Coefficients:
##             (Intercept)               livingArea                landValue
##               6.066e+03                7.271e+01                8.674e-01
##          waterfrontYes                bathrooms      newConstructionYes
##               1.291e+05                2.437e+04               -3.872e+04
##         heatinghot air  heatinghot water/steam                  lotSize
##               1.031e+04               -1.523e+03                6.232e+03
##                bedrooms                    rooms           centralAirYes
##              -9.053e+03                2.719e+03                8.595e+03
```

```
fstep = step(empty_house_mod, direction = "forward",scope = price ~ lotSize + age + landValue + livingAr
    bedrooms + bathrooms + rooms + heating + fuel + sewer +  waterfront + newConstruction +
    centralAir, k = 2, trace = 0)
par(mfrow = c(1,2))
plot(fstep, which = 1)
plot(fstep, which = 2)
```




```
backstep = step(empty_house_mod, direction = "both",scope =price ~ lotSize + age + landValue + livingAre
    bedrooms + bathrooms + rooms + heating + fuel + sewer +  waterfront + newConstruction +
    centralAir, k = 2, trace = 0)
par(mfrow = c(1,2))
plot(backstep, which = 1)
plot(backstep, which = 2)
```



- The above Q-Q Plot possesses a positive skew but it does an essentially perfect fit of the values from the -2 to approximately 1.5 quantile.This plot indicates that the model is less accurate in pricing the more expensive houses in Saratoga County.
- Compare with LOOCV.

```
loocv = function(model) {
  sqrt(mean((resid(model) / (1 - hatvalues(model))) ^ 2))
}
loocv(astep)
```

```
## [1] 61212.95
```

```
loocv(fstep)
```

```
## [1] 61212.95
```

```
loocv(backstep)
```

```
## [1] 61212.95
```

- We got the same LOOCV value for each model. Therefore, each model is equally as good.

```
final = astep

predictions = predict(final, tst_data)

train_error = c(rmse(trn_data$price, predict(final, trn_data)))
test_error = c(rmse(tst_data$price, predictions))

train_error
```

```
## [1] 60309.04
```

```
test_error
```

```
## [1] 54694.31
```

```
sum(abs(predictions - tst_data$price)/predictions * 100)/nrow(tst_data)
```

```
## [1] 19.39511
```

```
final = backstep

predictions = predict(final, tst_data)

train_error = c(rmse(trn_data$price, predict(final, trn_data)))
test_error = c(rmse(tst_data$price, predictions))

train_error
```

```
## [1] 60309.04
```

```
test_error
```

```
## [1] 54694.31
```

```
sum(abs(predictions - tst_data$price)/predictions * 100)/nrow(tst_data)
```

```
## [1] 19.39511
```

```
final = fstep

predictions = predict(final, tst_data)

train_error = c(rmse(trn_data$price, predict(final, trn_data)))
test_error = c(rmse(tst_data$price, predictions))

train_error
```

```
## [1] 60309.04
```

```
test_error
```

```
## [1] 54694.31
```

```
sum(abs(predictions - tst_data$price)/predictions * 100)/nrow(tst_data)
```

```
## [1] 19.39511
```

```
final = log_int_model

predictions = predict((final), tst_data)
predictions = (predictions[1:length(predictions)] * 0.42 + 1)^(1/0.42)

train_error = c(rmse(trn_data$price, predict(final, trn_data)))
test_error = c(rmse(tst_data$price, predictions))

train_error
```

```
## [1] 233229.1
```

```
test_error
```

```
## [1] 56243.48
```

```
sum(abs(predictions - tst_data$price)/predictions * 100)/nrow(tst_data)
```

```
## [1] 19.1181
```

Although `log_int_model` did not have the lowest percent error, it did have better diagnostics compared to the others, so we chose `log_int_model`.