Modeling Sales Price of Homes Using Physical Characteristics

Nicholas Lewis and Jacob Meyer

12/7/2018

Abstract

Can the sales price of a home be predicted using physcial characteristics of the home? We use square footage, number of bedrooms and bathrooms, garage size, year built, quality, style, lot size, and indicators for the presence of air conditioning, a pool, and an adjacent highway to determine the best possible model to make such a prediction. In doing this, we address the interests of the city tax assessor in a new and unique way. The final model contains only log(Square Footage), Year Built, Quality, and the interaction between log(square footage) and Number of Bathrooms to predict log(Sales Price). This can be easily transformed back to Sales Price to form a prediction that is highly accurate according to the data.

Introduction

The real estate market is unique when compared to other markets because of the diverse range of influences such as location and other physical characteristics on individual homes [1]. The 11 potential predictor variables investigated here encompass a large subset of these physical characteristics. We assume that the effect of location on Sales Price is the same throughout the data because all homes here were collected from the same city.

A common way of modeling homes is by using the repeat sales method. This method looks at homes sold more than once, and the difference between the two sales prices for a home is used to create an index through linear modeling [2]. Unfortunately, that model assumes no changes have been made to the house in between sales. Common sense tells us that this is often not the case. After contemplating this inaccurate assumption, we hypothesize that a model can be created using only single measurements, not comparisons. In other words, we ask the questions:

Can we effectively use physical characteristics to predict the sales price of a single home? If so, to what extent can we do so accurately?

Upon request from the city's tax assessor, we use the selling prices of local homes in his city and variables describing their characteristics to build a model predicting the sales price of homes. Through exploratory data analysis and then formal analysis, we are able to produce a model that answers these questions.

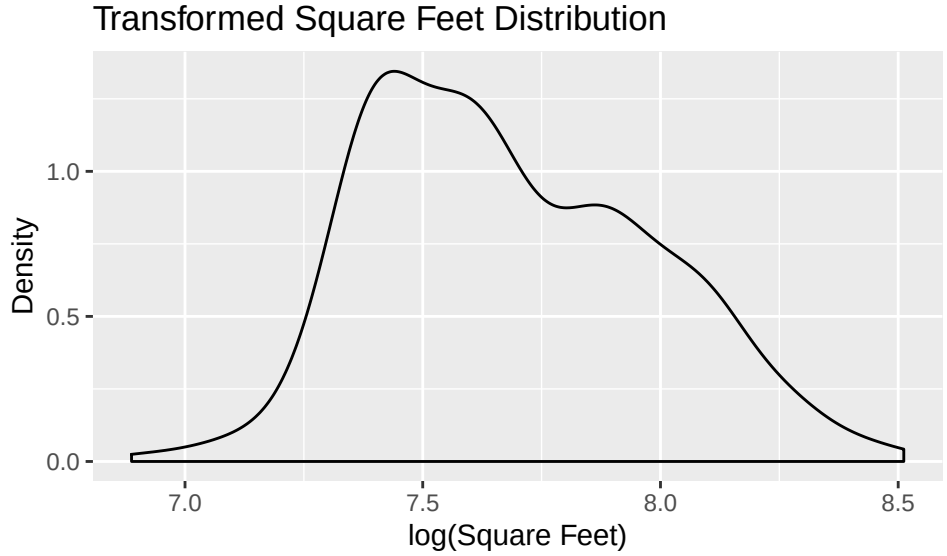
The Data Set

The data set includes 522 entries of data from arms-length home sales in an undisclosed city in the Midwest during 2002. The data was collected by the city tax assessor with the intention of predicting the future sales prices of homes. It includes sales price, finished area (in square feet), number of bedrooms and bathrooms, presence/absence of air conditioning, garage size (in number of cars), presence/absence of a pool, the year it was built, level of quality, style, the lot size, and the presence/absence of an adjacent highway for each home.

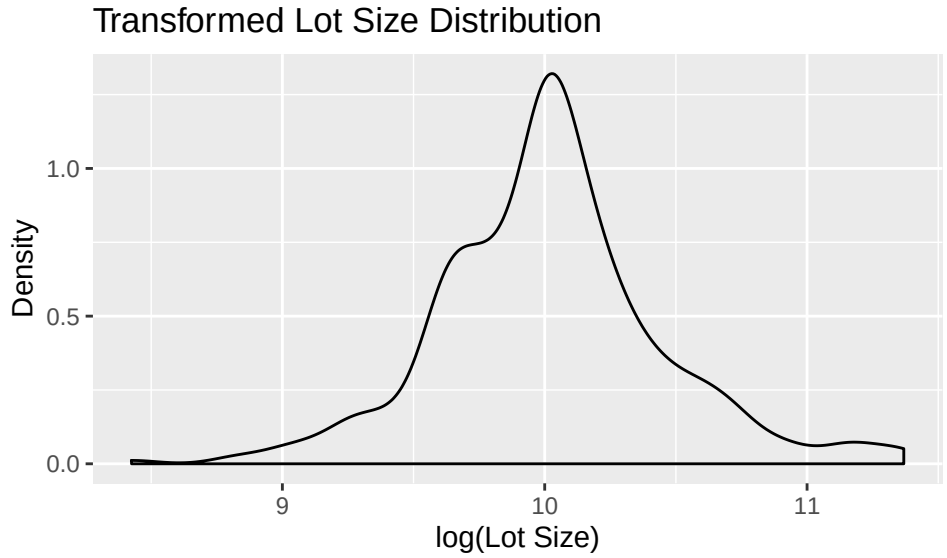
Exploratory Data Analysis (EDA)

For the exploratory analysis, we use a training set that is a randomly selected subset of the original data set.

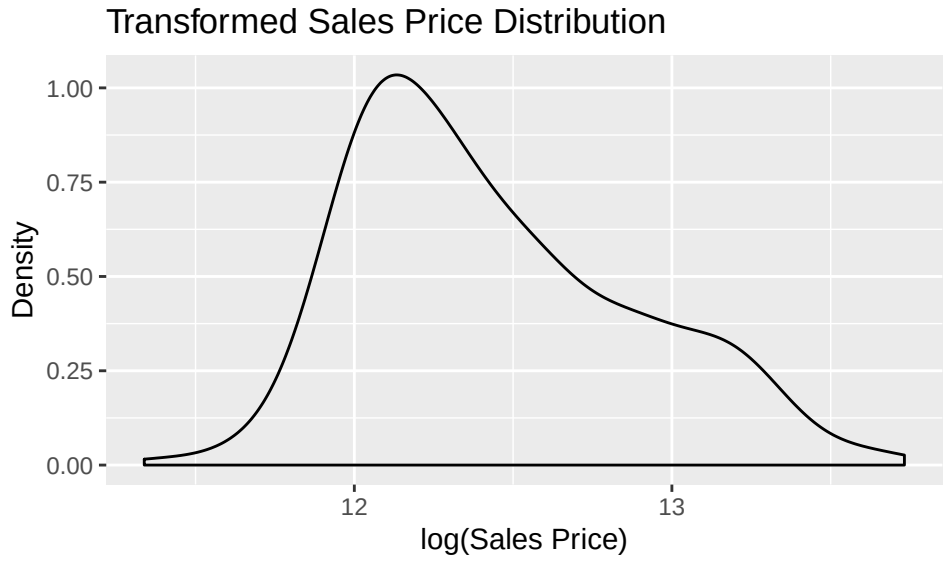
In order to observe the trends in the data, we looked first at a series of plot matrices (Appendix A). From these visuals, several transformations appear to be in order. The density curve for square feet appears to deviate from the normal distribution. After a logarithmic transformation, the data is clearly more normal. This also improves the consistency of variance when looking at the scatterplot of Sales Price vs log(Square Feet) (as opposed to Sales Price vs Square Feet).



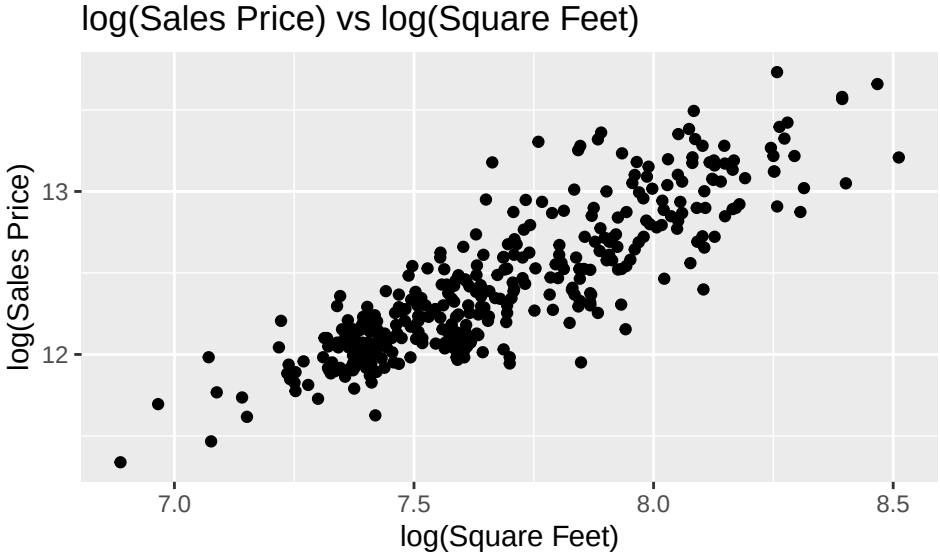
Similarly, the density curve of Lot Size is skewed, and a logarithmic transformation helps to normalize the data.



Interestingly, the distribution of the response variable also indicates that a transformation may be in order. Although it is not enough by itself to justify a transformation in a multiple regression situation, the non-constant variation in the scatterplot of Sales Price vs. Square Footage in the plot matrix also supports this decision. After a logarithmic transformation, the data appears to be more normal.



For this reason, we will actually be creating a model to predict the $log(Sales\ Price)$. The following graph shows improvement between log(Sales Price) and log(Square Feet) in the consistency of the variance after transformations.



It is also true that the variable representing the year in which the home was built may not be normal; however, a transformation does change the distribution (Appendix B). It will suffice untransformed; still, it is important to note that the left-skewed nature of the distribution indicates that recently built homes are represented more here.

Additionally, the plot matrix highlights a possible interaction term. The distribution of style seems to change drastically based on quality. Because this can be hard to see in the matrix, a larger version of the graph is included in Appendix C. We also consider that two interaction terms, the number of beds with square footage and number of baths with square footage, may be significant. It is logical that a small house with lots of bedrooms would sell differently than a large house with the same number of bedrooms.

Based off of this information, we next investigate graphs of the residuals versus fitted values from a tentative model incoporating all variables, interaction terms, and transformations. The graphs are available in Appendix D, and show no obvious, influential outliers. Diagnostic plots of Cook's distance for outliers supports this conclusion.

Analysis

To select a model, we consider all subsets of a model containing each individual term (or its transformation) and interactions established during EDA. They are then ranked using Bozdogen's Information Complexity (BIC) values, and this system is used to select the best model. A visualization of this process is included in Appendix E.

However, a summary of the best model (Appendix F) also shows that several of the variables are not significant. Because of this, an extra-sum-of-squares test is performed. The results show that there is not sufficient evidence to suggest that there is a difference in the models including or excluding these terms (Appendix G). Additionally, the adjusted R^2 values and mean squared errors for the two models are essentially equivalent. For this reason, we choose the reduced model to avoid over-complication.

The following equation represents the model.

$$\begin{aligned} log(SalesPrice) = 2.229957 &+ .674833 * (log(SquareFeet)) \\ &+ .002601 * (YearBuilt) - .324327 * (Quality2) \\ &- 0.411337 * (Quality3) + .010708 * (log(SquareFeet) * (NumberofBaths)) \end{aligned}$$

The model accounts for 84.99% of the variability and has a mean squared error of only 0.0319.

Conclusion

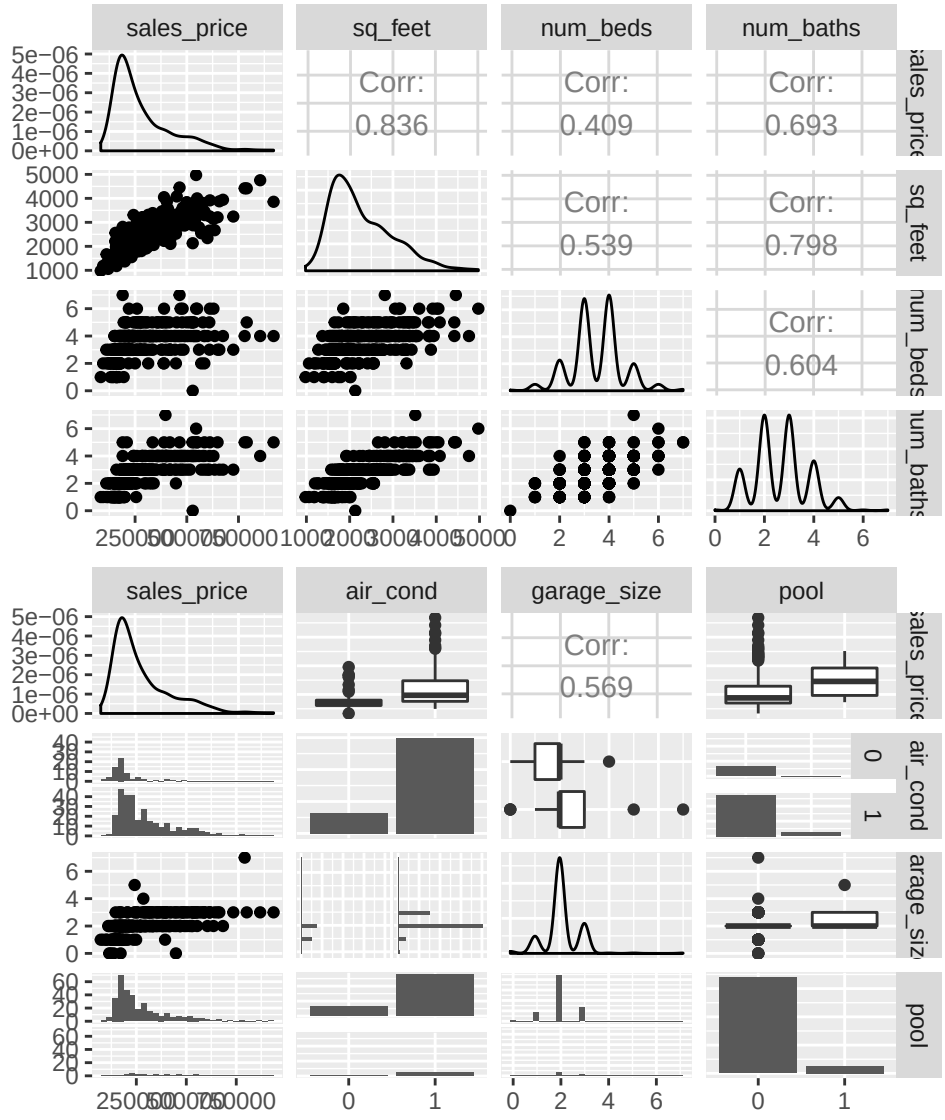
This final model's accuracy will be truly tested when applied to homes currently on the market in this area and compared to the true sales price. Still, the accuracy statistics that we have at this time prove that the final model is capable of predicting extremely well for the data set supplied by the city tax assessor. Within the limitations of our error checking, the model successfully uses individual sales of homes to eliminate the need for assumptions that were present in the repeat sales method.

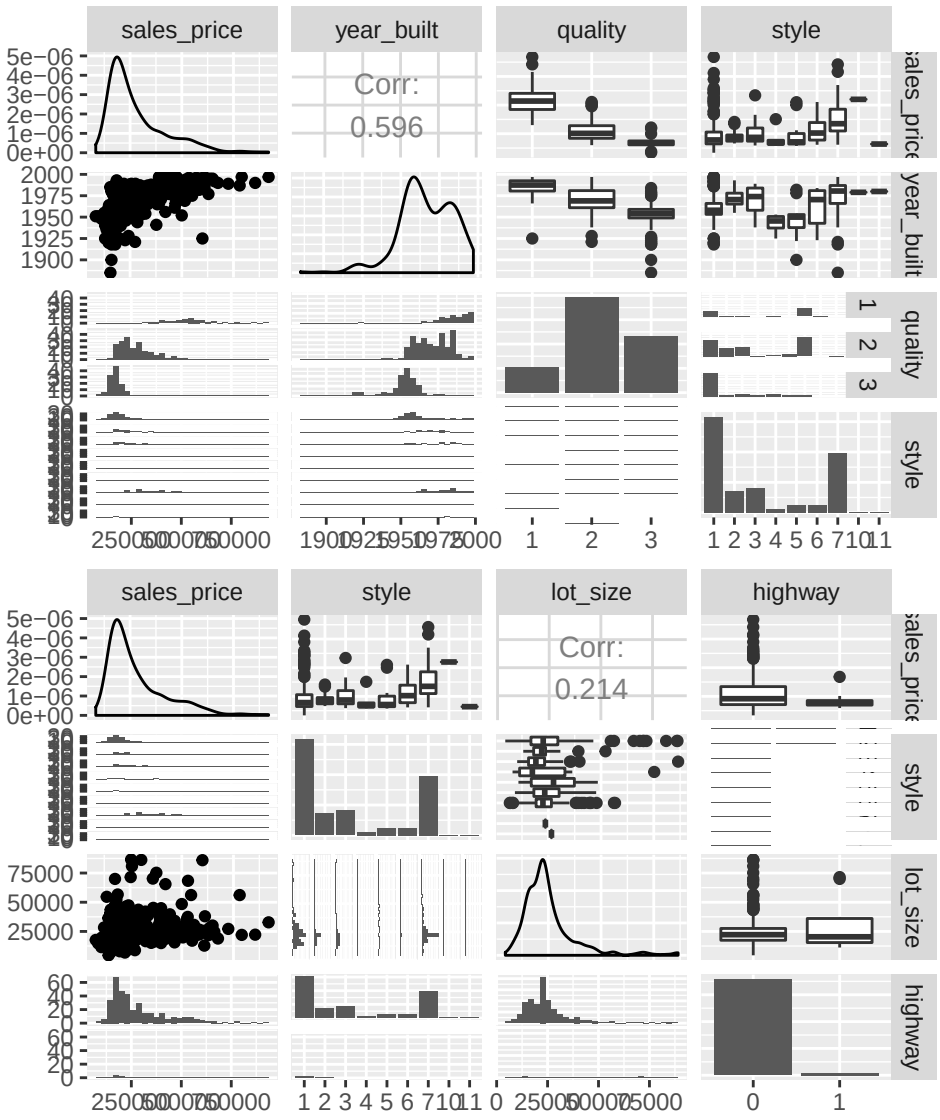
The final model can be useful in predicting sales prices of individual homes as well as acquiring general information about the sales prices of homes in the city. For instance, the selling price of the average home in the city is $261467.

It is important to recognize the limitations of the model imposed by the data. The data was collected only from one city, and thus the results of this analysis should only be applied to that city. While the *process* may be applied to sales price data in other locations, the *specific model and coefficients* are not applicable due to the vast variation in the price of housing across the United States and the world. Additionally, this model is only reasonably capable of predicting sales price of homes that were built between 1885 and 1998. We cannot extrapolate to cases that would fall outside of the years for which the data supporting the model exists. Therefore, newly built houses fall outside of the scope of our model.

Appendix A: Scatterplot Matrices

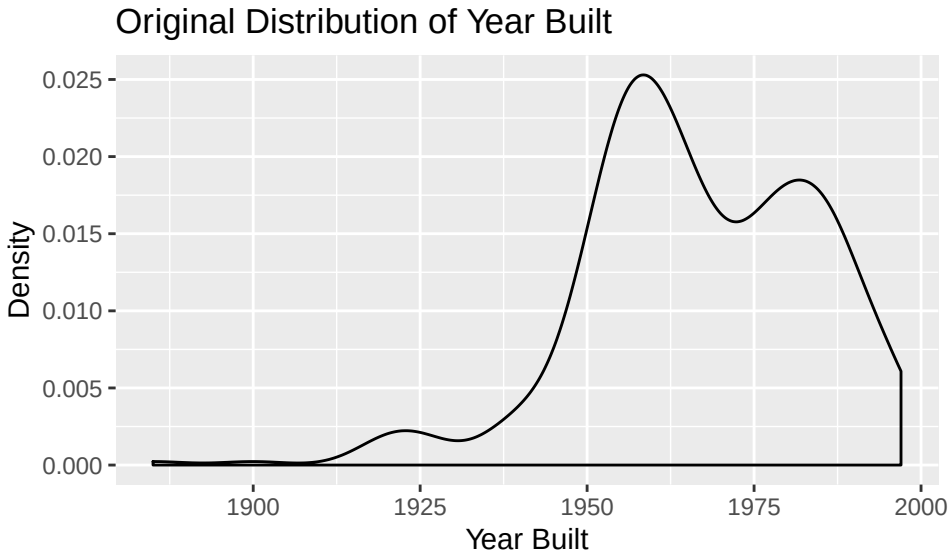
The following are four matrices of scatterplots and relevant graphs detailing distributions and interactions between variables. Unfortunately, not all variables can be compared at once because of limitations in the readability of the graphs. Instead, each matrix includes the untransformed response variable and three possible predictor variables.

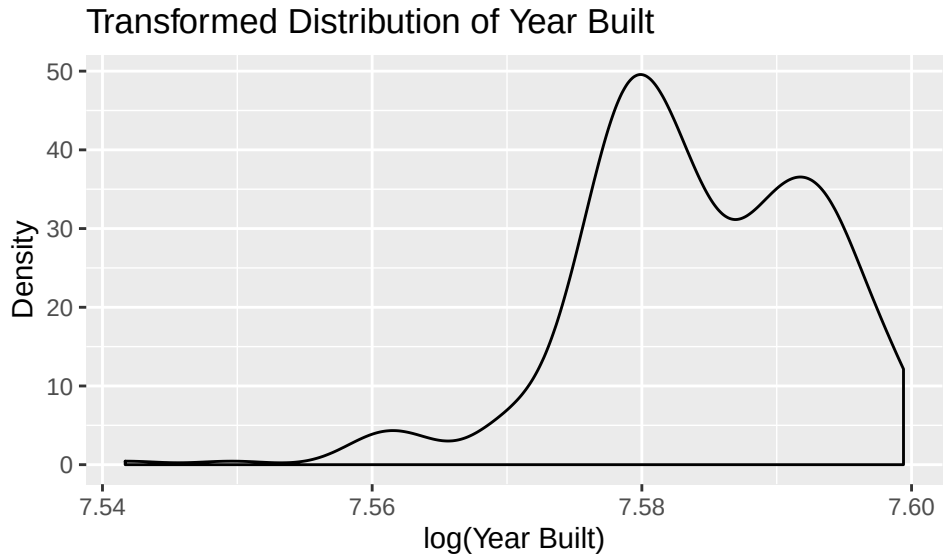




Appendix B: Year Built Transformations

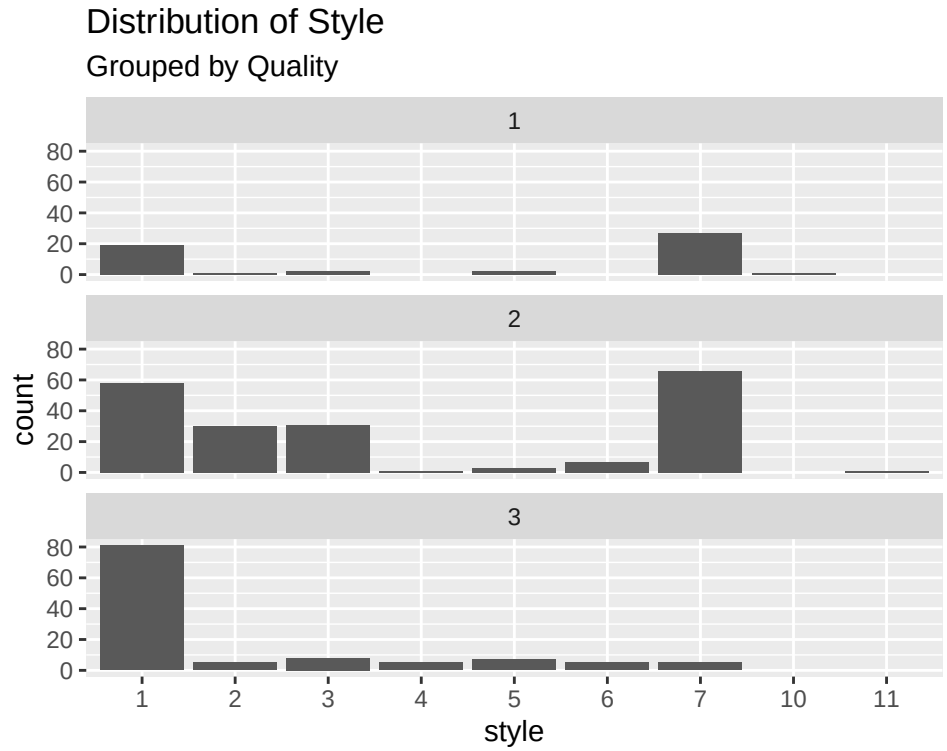
Several possible transformations were attempted including $1/year\_built$, $year\_built^2$, and $log(year\_built)$. All transformations resulted in the same shape distribution (with the exception that $1/year\_built$ was inverted). Only the logarithmic transformation is shown here.





Appendix C: Style and Quality Interaction

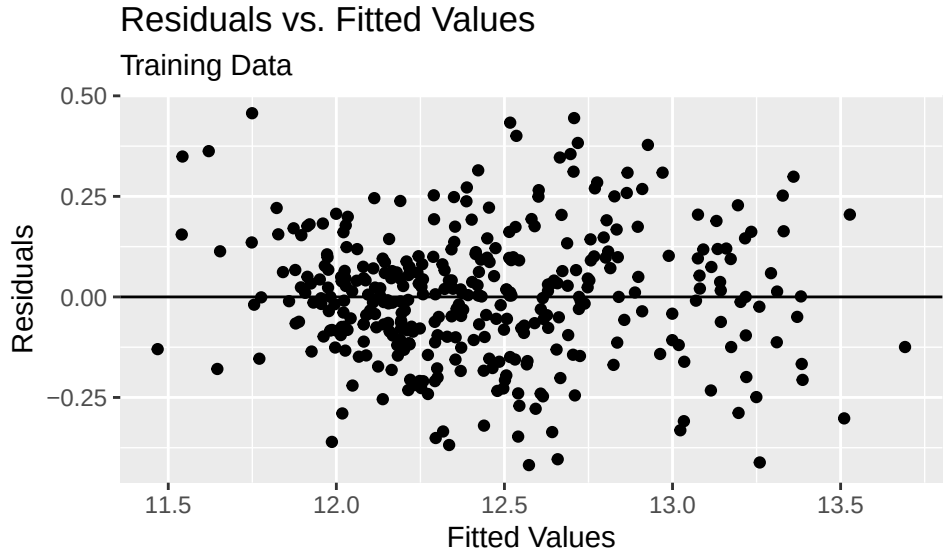
The following bar charts serve as a visualization for the distribution of style when separated by the level of quality of production. A home of quality level 3 has a different distribution of styles than a home of quality levels 1 or 2.



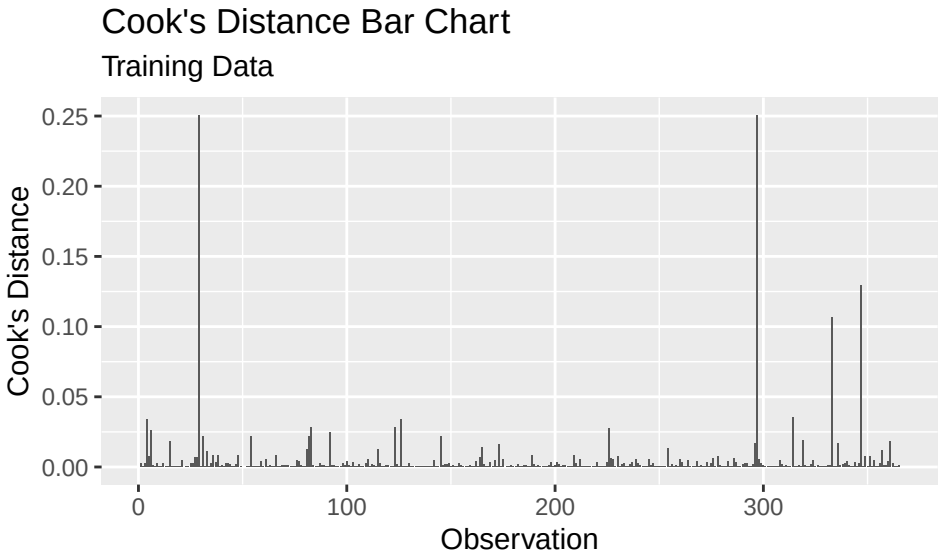
Appendix D: Residuals vs. Fitted Values

The computing capabilities of R limit us to the use of only our suspected interactions when creating a "full" model, not all possible interactions.

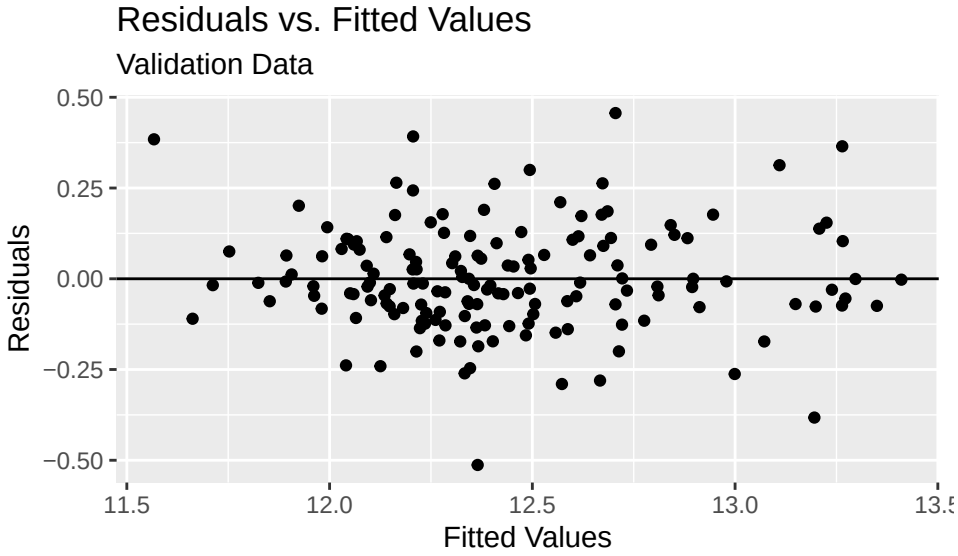
The following scatterplot shows the residuals versus their corresponding fitted values in the training data set based on the tentative (full) model.



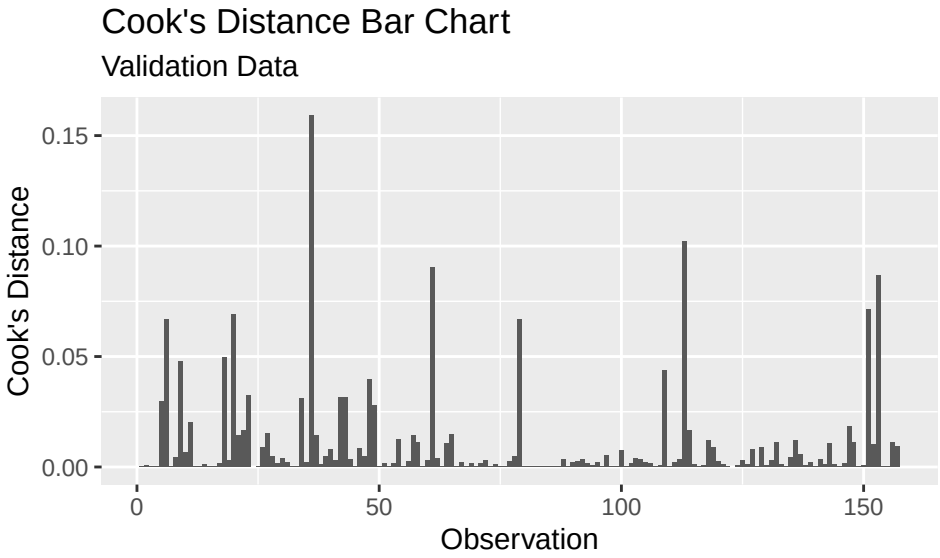
No outliers are obvious, but we look further to be certain. The following is a visualization of the Cook's Distances in the training data set based on the tentative (full) model.



The following scatterplot shows the residuals versus their corresponding fitted values in the validation data set based on the tentative (full) model.



It appears that the point in the bottom right corner may be an outlier. To check, we look again at Cook's Distance. The following is a visualization of the Cook's Distances in the validation data set based on the tentative (full) model.

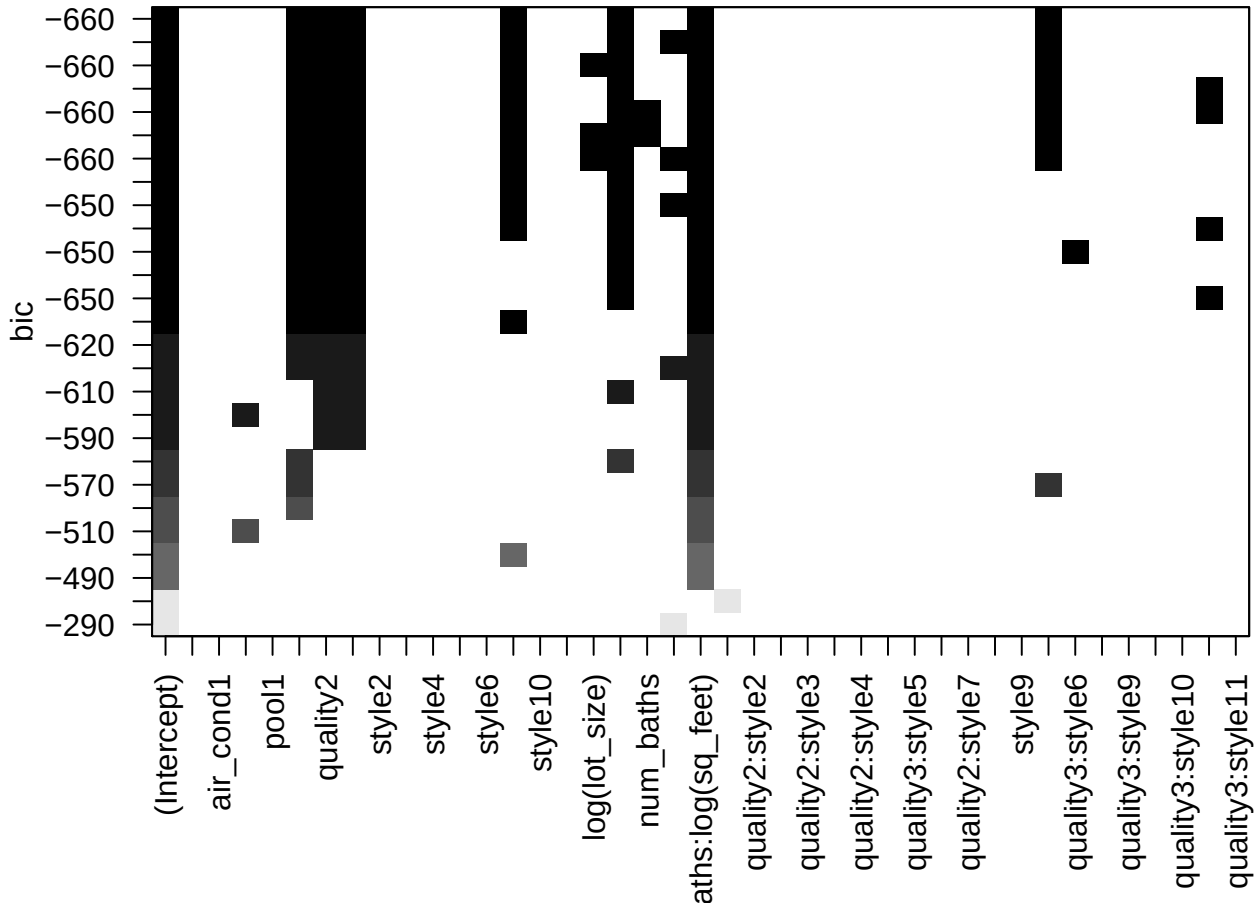


Through these visuals, we can conclude that no observation in either data set are influential outliers because the Cook's Distance is below 1 for every observation.

Appendix E: Model Selection

The following illustrates a series of the best possible versions of the model, specifically the three best models containing each possible number of variables. The shaded boxes denote included variables, and the model with the lowest BIC score was selected as the best fitting model.

```
Reordering variables and trying again:
```



Appendix F: Selected Model Summary

The following information summarizes the best model, selected by considering all possible subsets and ranking by BIC value.

```
Call:
lm(formula = log(sales_price) ~ log(sq_feet) + year_built + quality +
    highway + num_baths:log(sq_feet) + num_beds:log(sq_feet) +
    I(quality == 3 & style == 6), data = validate)

Residuals:
     Min       1Q   Median       3Q      Max
-0.53214 -0.10844 -0.01743  0.10520  0.73494

Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)
(Intercept)                      2.4924367  2.3146393   1.077  0.28332
log(sq_feet)                     0.6899694  0.0945915   7.294 1.69e-11
year_built                       0.0024537  0.0010448   2.349  0.02017
quality2                        -0.3435577  0.0608348  -5.647 8.08e-08
quality3                        -0.4602833  0.0806609  -5.706 6.09e-08
highway1                         0.0937759  0.1083645   0.865  0.38823
I(quality == 3 & style == 6)TRUE  0.1224027  0.0996017   1.229  0.22105
```

```
log(sq_feet):num_baths            0.0083868  0.0030447   2.755  0.00661
log(sq_feet):num_beds            -0.0001292  0.0026784  -0.048  0.96160

(Intercept)
log(sq_feet)                     ***
year_built                       *
quality2                         ***
quality3                         ***
highway1
I(quality == 3 & style == 6)TRUE
log(sq_feet):num_baths           **
log(sq_feet):num_beds
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1825 on 148 degrees of freedom
Multiple R-squared:  0.8105,    Adjusted R-squared:  0.8003
F-statistic: 79.13 on 8 and 148 DF,  p-value: < 2.2e-16
```

Not all variables are significant, so they require further investigation.

Appendix G: Extra-sum-of-squares test

An extra-sum-of squares test was performed to address the hypotheses for the variables Highway, Interaction between Quality 3 and Style 6, and Interaction between log(Square Feet) and Number of Baths:

H_0 = The coefficients associated with the variables are all zero
H_0 = At least one of the coefficients associated with the variables is not zero

The results of the test are shown below.

```
Analysis of Variance Table

Model 1: log(sales_price) ~ log(sq_feet) + year_built + quality + num_baths:log(sq_feet)
Model 2: log(sales_price) ~ log(sq_feet) + year_built + quality + highway +
    num_baths:log(sq_feet) + num_beds:log(sq_feet) + I(quality ==
    3 & style == 6)
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    151 5.0067
2    148 4.9304  3  0.076282 0.7633 0.5164
```

There is not sufficient evidence to reject the null hypothesis.

The following information serves as a summary of reduced model. It highlights the adjusted R^2 values.

```
Call:
lm(formula = log(sales_price) ~ log(sq_feet) + year_built + quality +
    num_baths:log(sq_feet), data = validate)

Residuals:
     Min       1Q   Median       3Q      Max
-0.52912 -0.11000 -0.01181  0.10865  0.73148

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)             2.457776   2.230049   1.102  0.27216
log(sq_feet)            0.704609   0.080505   8.752 3.88e-15 ***
year_built              0.002411   0.001013   2.379  0.01861 *
quality2               -0.336805   0.058855  -5.723 5.47e-08 ***
quality3               -0.440457   0.076874  -5.730 5.29e-08 ***
log(sq_feet):num_baths  0.008320   0.002958   2.813  0.00556 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1821 on 151 degrees of freedom
Multiple R-squared:  0.8076,    Adjusted R-squared:  0.8012
F-statistic: 126.7 on 5 and 151 DF,  p-value: < 2.2e-16
```

The following are the mean squared errors for the models being compared.

$MSE_{full} = 0.031$
$MSE_{red} = 0.032$

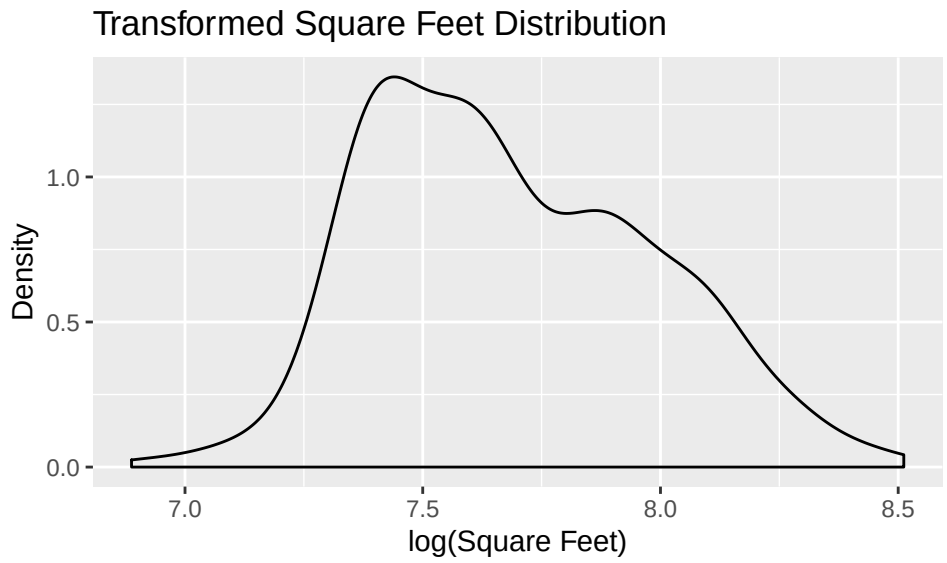
Appendix H: R Code

```
# preamble
library(readr)
library(tidyverse)
library(GGally)
library(leaps)
sales <- read_csv("sales.csv")
```

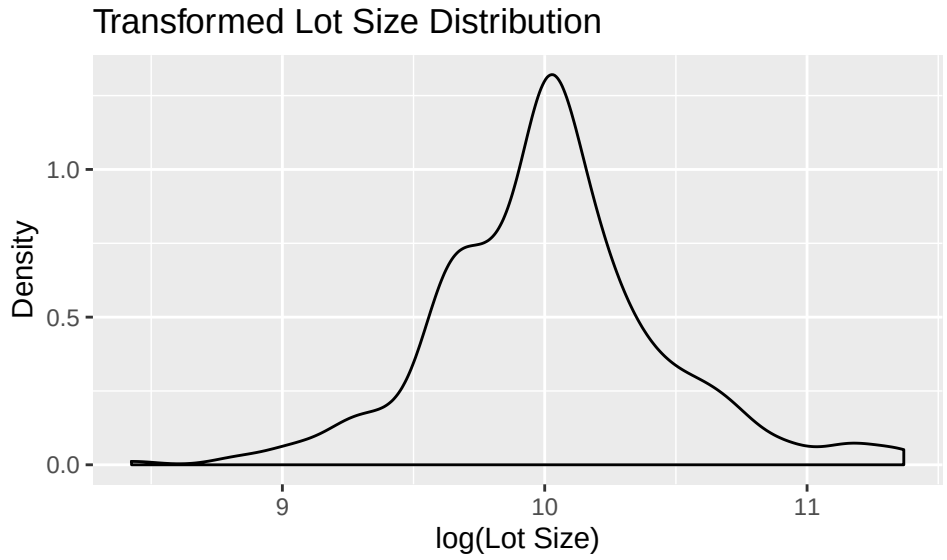
```
Parsed with column specification:
cols(
  ID = col_double(),
  sales_price = col_double(),
  sq_feet = col_double(),
  num_beds = col_double(),
  num_baths = col_double(),
  air_cond = col_double(),
  garage_size = col_double(),
  pool = col_double(),
  year_built = col_double(),
  quality = col_double(),
  style = col_double(),
  lot_size = col_double(),
  highway = col_double()
)
```

```
# correcting int values to factor values
sales <- mutate(sales, air_cond = factor(air_cond),
                pool = factor(pool),
                quality = factor(quality),
                style = factor(style),
                highway = factor(highway))
# dividing into training and validation sets
set.seed(3)
train <- sales %>% sample_frac(.7)
validate <- sales %>% setdiff(train)
```

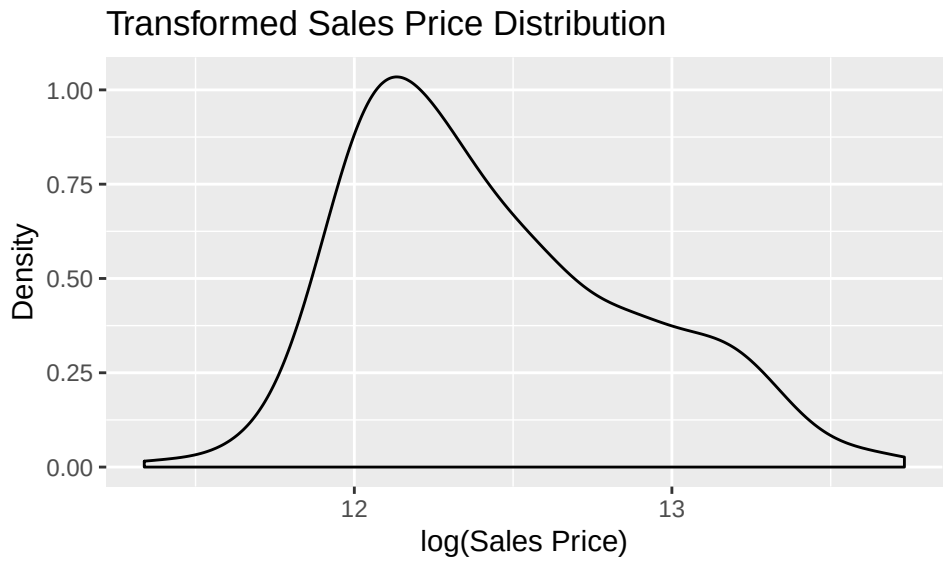
```
# creating a plot of the transformed sq_feet distribution
ggplot(data = train) +
  geom_density(mapping = aes(x = log(sq_feet))) +
  ggtitle("Transformed Square Feet Distribution") +
  xlab("log(Square Feet)") +
  ylab("Density")
```



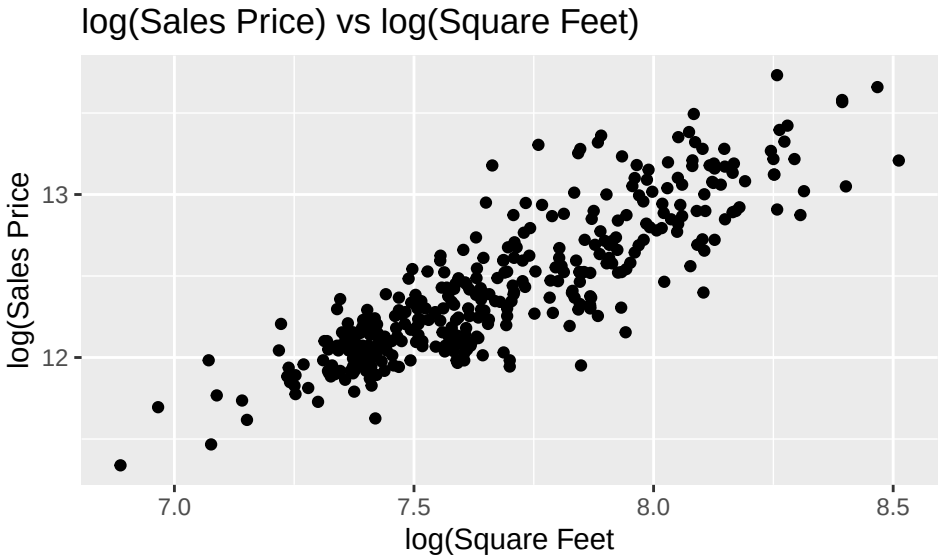

```
# creating a plot of the transformed lot_size distribution
ggplot(data = train) +
  geom_density(mapping = aes(x = log(lot_size))) +
  ggtitle("Transformed Lot Size Distribution") +
  xlab("log(Lot Size)") +
  ylab("Density")
```




```
# creating a plot of the transformed sales_price distribution
ggplot(data = train) +
  geom_density(mapping = aes(x = log(sales_price))) +
  ggtitle("Transformed Sales Price Distribution") +
  xlab("log(Sales Price)") +
  ylab("Density")
```




```
# creating a plot, log(sales_price) vs log(sq_feet)
ggplot(data = train) +
  geom_point(mapping = aes(x = log(sq_feet), y = log(sales_price))) +
  ggtitle("log(Sales Price) vs log(Square Feet)") +
  xlab("log(Square Feet") +
  ylab("log(Sales Price")
```



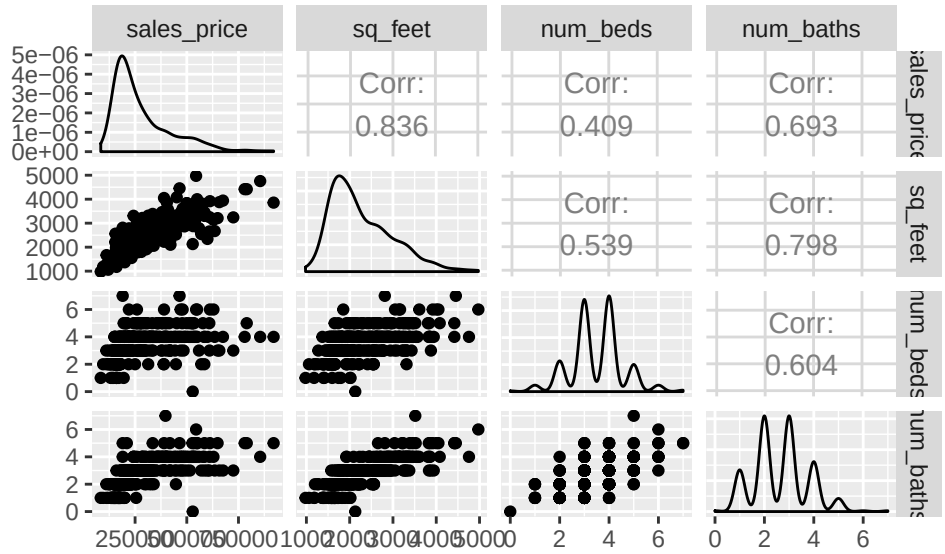

```
# creating the final model for calculation of the MSE
lmred <- lm(log(sales_price) ~ log(sq_feet) + year_built + quality + num_baths:log(sq_feet), data = vali
MSEred <- validate %>%
  mutate(pred = predict(lmred, newdata = ., type = "response")) %>%
  summarise(mse = mean((log(sales_price) - pred)^2))
```

```
# creating the average home to test
avesales <- data_frame("sq_feet" = mean(sales$sq_feet),
                       "year_built" = mean(sales$year_built),
                       "num_baths" = mean(sales$num_baths),
                       "quality" = factor(2))
```

```
# turning off scientific notation for the in-line code describing predicted sales price for the average
options(scipen = 999)
```

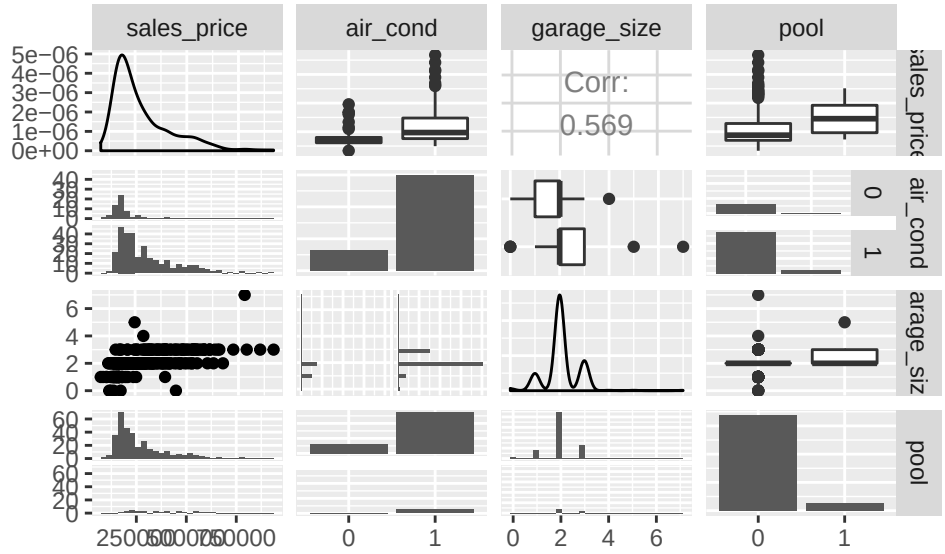
```
# turning back on scientific notation now that we may want it as a defualt again
options(scipen = 0)

# creating matrices of scatterplots
ggpairs(data = train, columns = c(2,3:5))
```




```
ggpairs(data = train, columns = c(2,6:8))
```

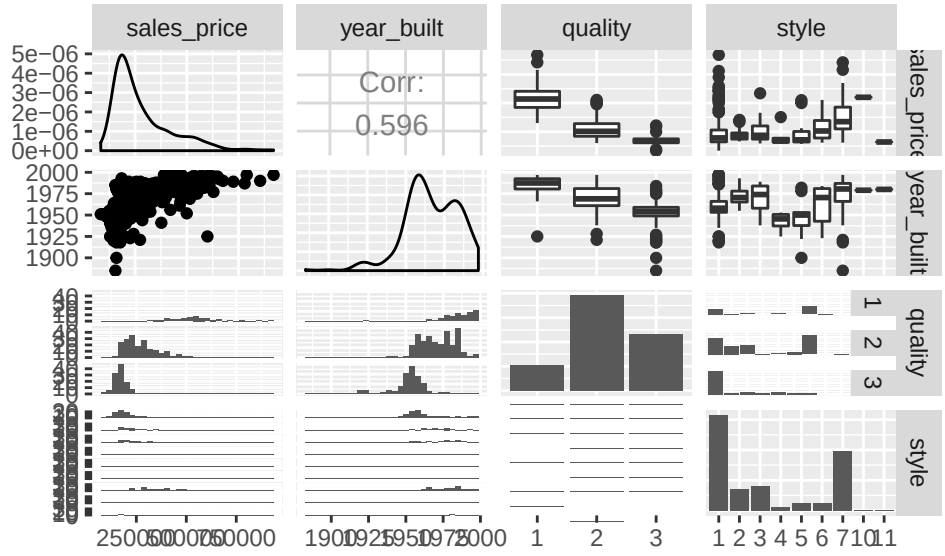
```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggpairs(data = train, columns = c(2,9:11))
```

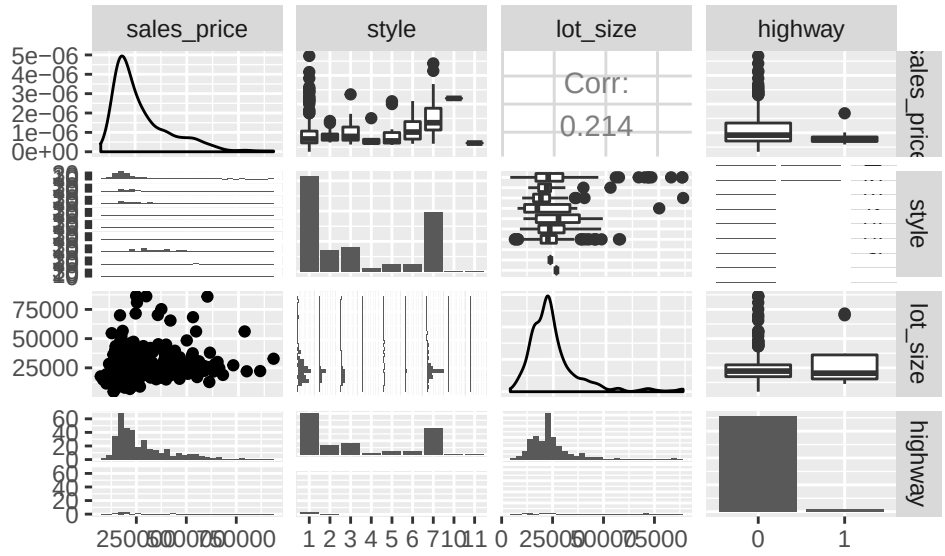
```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

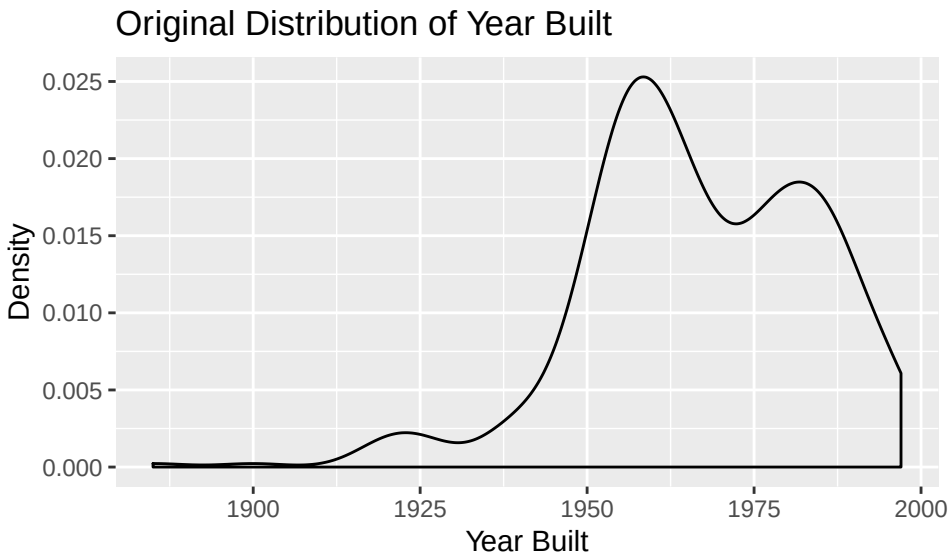



```
ggpairs(data = train, columns = c(2,11:13))
```

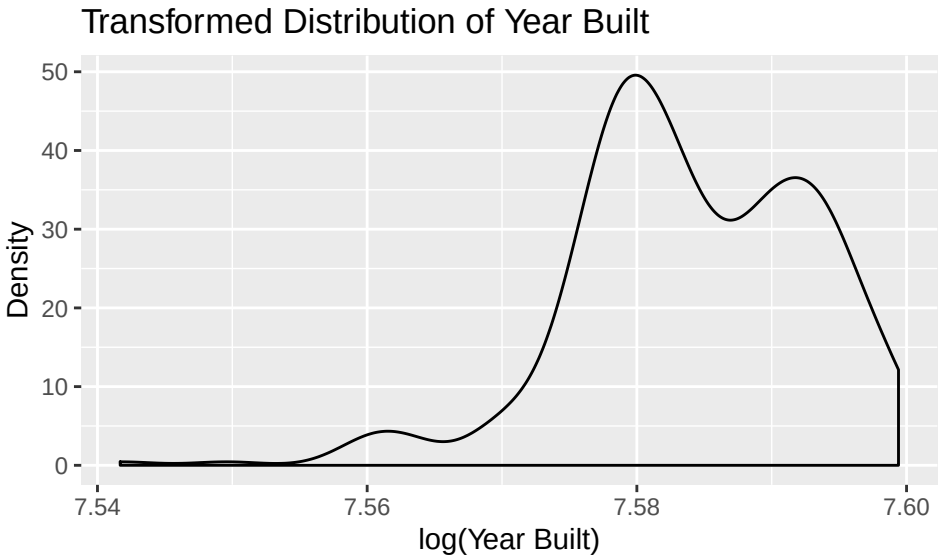
```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



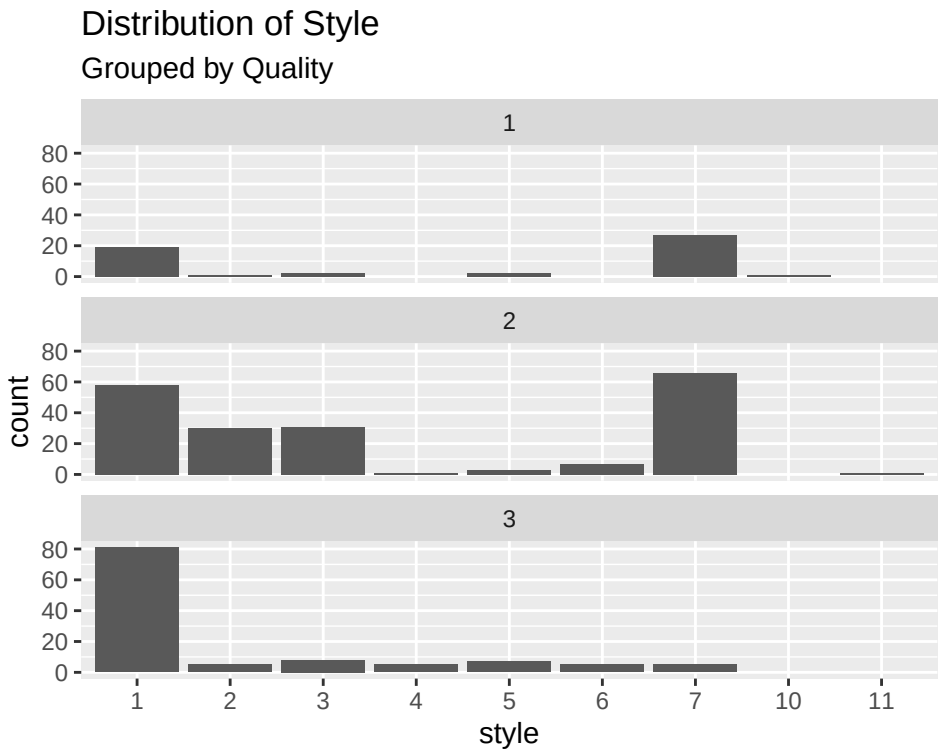

```
# creating a density plot for year_built
ggplot(data = train) +
  geom_density(mapping = aes(x = year_built)) +
  ggtitle("Original Distribution of Year Built") +
  xlab("Year Built") +
  ylab("Density")
```




```
# creating a density plot for log(year_built)
ggplot(data = train) +
  geom_density(mapping = aes(x = log(year_built))) +
  ggtitle("Transformed Distribution of Year Built") +
  xlab("log(Year Built)") +
  ylab("Density")
```

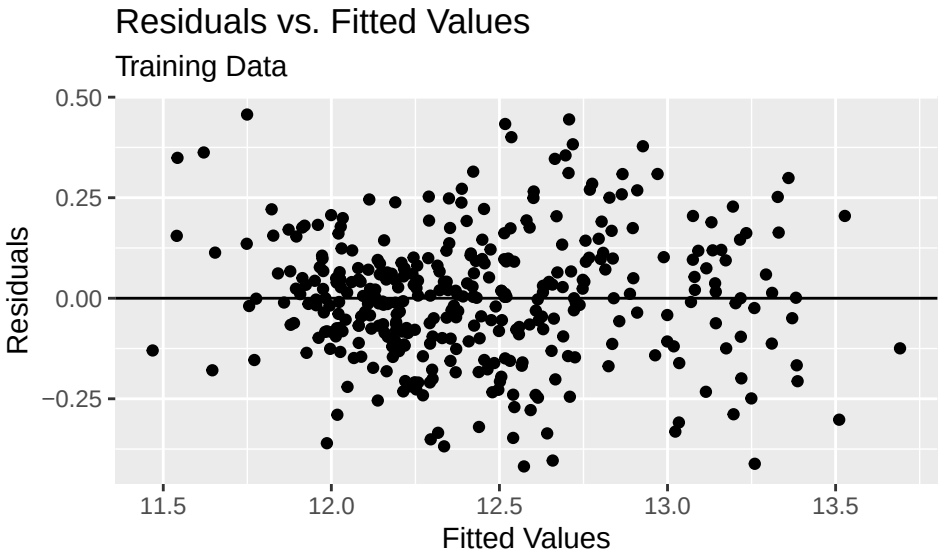



```
# creating a bar chart of style wrapped by quality
ggplot(data = train) +
  geom_bar(mapping = aes(x = style)) +
  facet_wrap(~quality, nrow = 3) +
  ggtitle("Distribution of Style", subtitle = "Grouped by Quality")
```



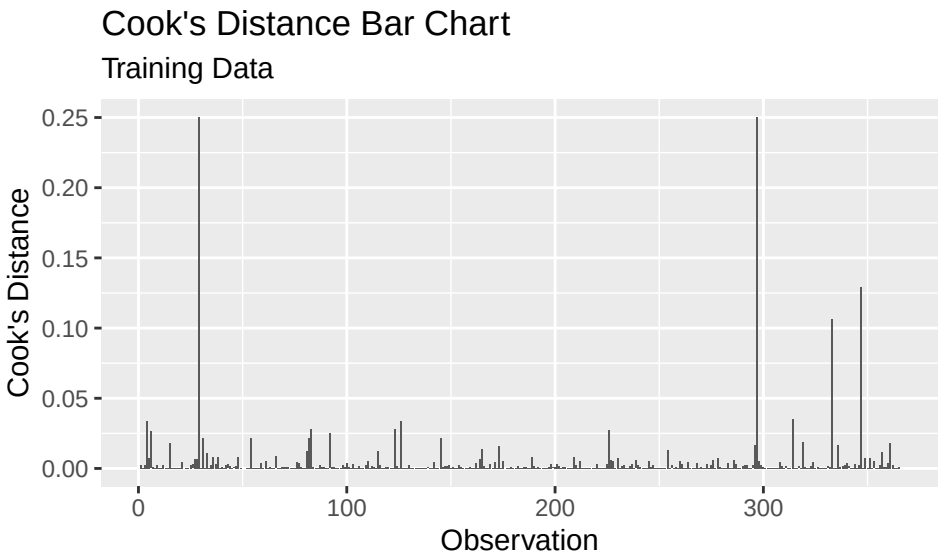

```
# creates the full model based off of the training set
fitfulltrain <- lm(log(sales_price) ~ log(sq_feet) + num_beds + num_baths + air_cond + garage_size + poo

# plot the residuals vs fitted values for this model
ggplot(fitfulltrain) +
  geom_point(mapping = aes(x = .fitted, y = .resid)) +
  geom_hline(yintercept = mean(fitfulltrain$residuals)) +
  ggtitle("Residuals vs. Fitted Values", subtitle = "Training Data") +
  xlab("Fitted Values") +
  ylab("Residuals")
```



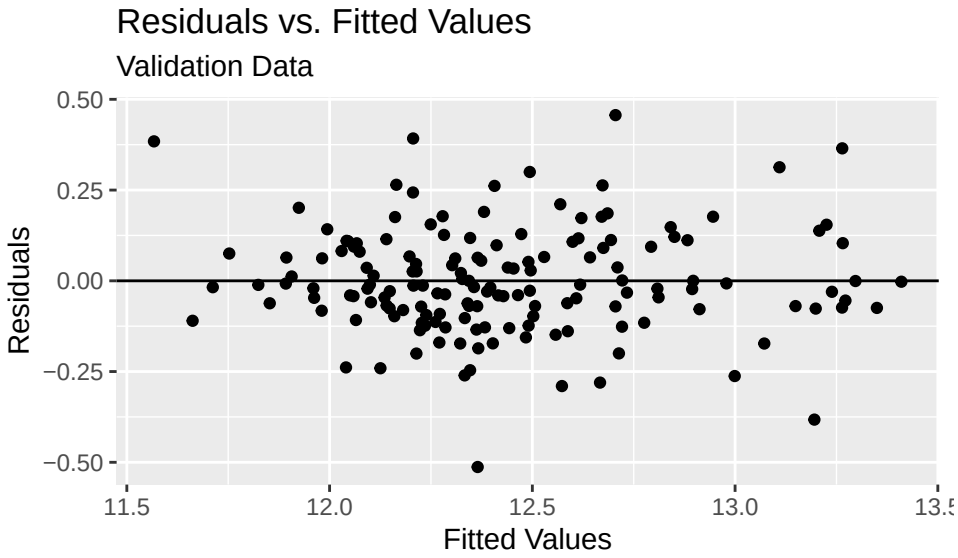
```r
# creates visual for cook's distance for the training set
ggplot(data = fitfulltrain) +
  geom_bar(mapping = aes(x = seq_along(.cooksd), y = .cooksd), stat = "identity") +
  ggtitle("Cook's Distance Bar Chart", subtitle = "Training Data") +
  xlab("Observation") +
  ylab("Cook's Distance")
```

```
Warning: Removed 4 rows containing missing values (position_stack).
```



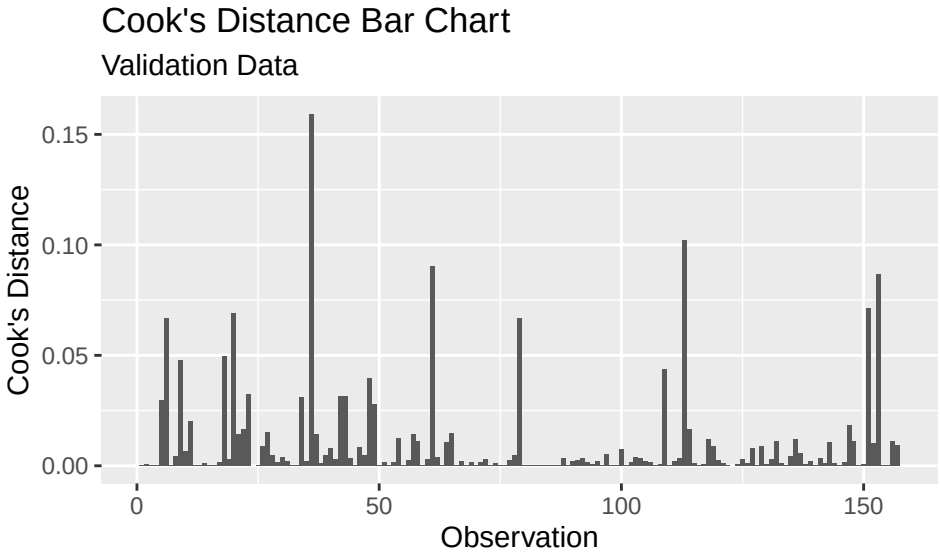

```r
# creates full model for validation set
fitfullvalidate <- lm(log(sales_price) ~ log(sq_feet) + num_beds + num_baths + air_cond + garage_size +

# plots residuals vs fitted values for the validation set
ggplot(fitfullvalidate) +
  geom_point(mapping = aes(x = .fitted, y = .resid)) +
  geom_hline(yintercept = mean(fitfullvalidate$residuals)) +
  ggtitle("Residuals vs. Fitted Values", subtitle = "Validation Data") +
  xlab("Fitted Values") +
  ylab("Residuals")
```




```
# creates visual for cook's distance for the validation set
ggplot(data = fitfullvalidate) +
  geom_bar(mapping = aes(x = seq_along(.cooksd), y = .cooksd), stat = "identity") +
  ggtitle("Cook's Distance Bar Chart", subtitle = "Validation Data") +
  xlab("Observation") +
  ylab("Cook's Distance")
```

```
Warning: Removed 2 rows containing missing values (position_stack).
```

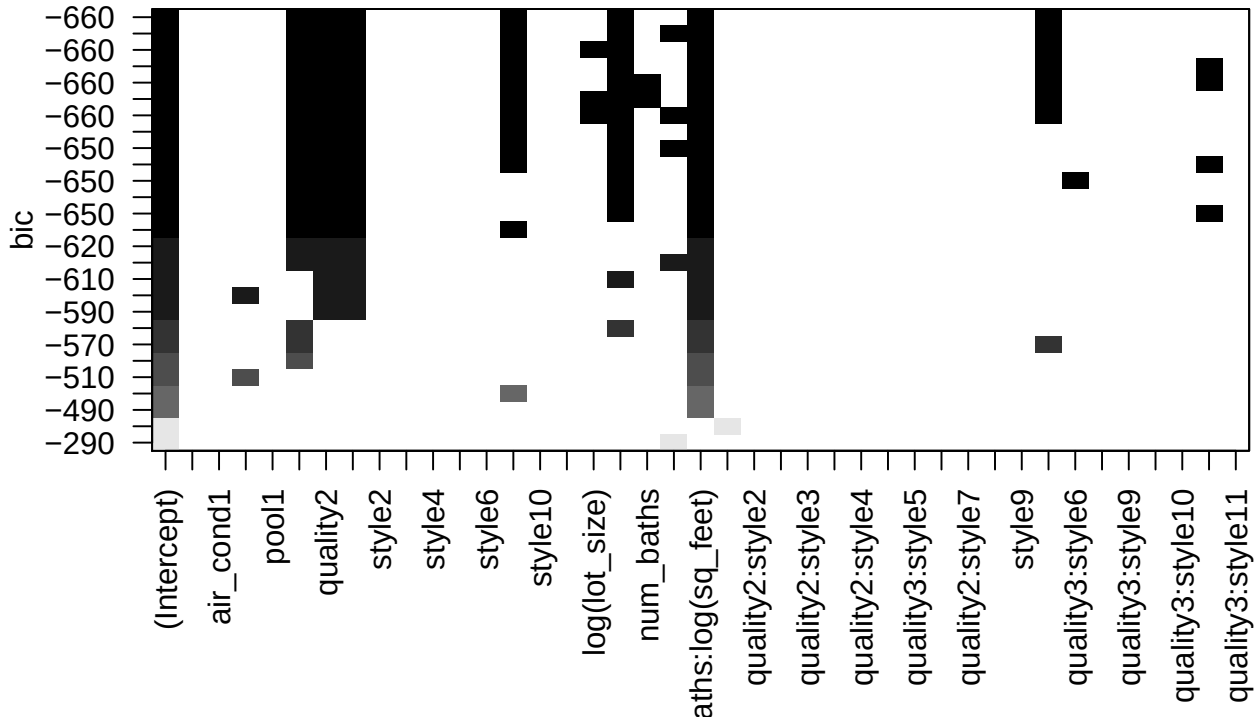



```
# uses regsubsets to obtain the 3 best models of each number of included variables
models <- regsubsets(log(sales_price) ~ num_beds + air_cond + garage_size + pool + year_built + quality
```

```
Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
force.in = force.in, : 9 linear dependencies found
```

```
Reordering variables and trying again:
```

```
# plots the best models
plot(models)
```




```
# creates a reduced model without the insignificant terms
lmred <- lm(log(sales_price) ~ log(sq_feet) + year_built + quality + num_baths:log(sq_feet), data = vali

# runs the extra-sum-of-squares test
anova(lmred, lm)
```

```
Analysis of Variance Table

Model 1: log(sales_price) ~ log(sq_feet) + year_built + quality + num_baths:log(sq_feet)
Model 2: log(sales_price) ~ log(sq_feet) + year_built + quality + highway +
    num_baths:log(sq_feet) + num_beds:log(sq_feet) + I(quality ==
    3 & style == 6)
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    151 5.0067
2    148 4.9304  3  0.076282 0.7633 0.5164
```

```
# calculates the MSE for both models
MSEfull <- validate %>%
  mutate(pred = predict(lm, newdata = ., type = "response")) %>%
  summarise(mse = mean((log(sales_price) - pred)^2))
MSEred <- validate %>%
  mutate(pred = predict(lmred, newdata = ., type = "response")) %>%
  summarise(mse = mean((log(sales_price) - pred)^2))
```

```
# summarizes the final model
summary(lmred)
```

```
Call:
lm(formula = log(sales_price) ~ log(sq_feet) + year_built + quality +
    num_baths:log(sq_feet), data = validate)
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-0.52912 -0.11000 -0.01181  0.10865  0.73148

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)            2.457776   2.230049   1.102  0.27216
log(sq_feet)           0.704609   0.080505   8.752 3.88e-15 ***
year_built             0.002411   0.001013   2.379  0.01861 *
quality2              -0.336805   0.058855  -5.723 5.47e-08 ***
quality3              -0.440457   0.076874  -5.730 5.29e-08 ***
log(sq_feet):num_baths  0.008320   0.002958   2.813  0.00556 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1821 on 151 degrees of freedom
Multiple R-squared:  0.8076,    Adjusted R-squared:  0.8012
F-statistic: 126.7 on 5 and 151 DF,  p-value: < 2.2e-16
```

References

[1] Ghysels E., Plazzi A., Torous W., Valkanov R. (July 1 2012). Forecasting Real Estate Prices. *Handbook of Economic Forecasting, VII*. Retrieved from `https://rady.ucsd.edu/faculty/directory/valkanov/pub/docs`

[2] Nagaraja C.H., Brown L.D., Zhao L.H. (2011). An Autoregressive Approach to House Price Modelling. *The Annals of Applied Statistics, 5, 124-129*. doi:10.1214/10-AOAS380