Real Estate House Pricing Analysis

# Abstract

Buying or selling a house is one of the important events faced by many in their lifetime. As such understanding what factors drive sale prices is both interesting and beneficial. In this project I used the popular Ames, Iowa housing data set to investigate which factors influence the final sale price. The research showed that location and condition of the house are the biggest attributes. A bigger house will not increase the final sale price as much as a smaller, nicer house in a better neighborhood.

# Key Words

house prices, regression, linear models, assessed value

# Introduction

This paper is the result of the final project in applying regression methods to real-world data. I chose housing data because I work as a Quant on Investment Real Estate portfolio for a Commercial Bank. In addition, this research is based on a well-studied data set that allowed me to employ many topics discussed during the course.

The data set describes the sale of individual residential properties in Ames, Iowa from 2006 to 2010. Founded in 1864 as a station stop, Ames has a population of approximately 60,000 people, covers about 24.27 square miles and was ranked ninth on the *Best Places to Live*. The original data came directly from the Ames Assessor’s Office recordkeeping database and included information for calculation of assessed value using the city’s assessment process. Each of over 2,900 total observations in the data represent attributes of an individual residential property sold. For properties that exchanged ownership multiple times during the collection period (2006 through 2010). There are about 80 variables included that cover a multitude of property attributes. Most variables describe the physical appearance or features of the property but other variables provide measures of quality and condition. The data types are varied and include discrete, continuous, and categorical (both nominal and ordinal) data.

The data was originally published in the Journal of Statistics Education. The data used for regression was downloaded from Kaggle.com, which allowed me the opportunity to compete against other kaggle users to find the best-fitting model for this data.

# Literature Review

Building regression models to predict house prices is not a new undertaking.There is also a lot of data readily available with some cleanup work. Additionally, in large part thanks to the information revolution, data are easily accessible via many multiple listing services such as MLS.com.

There are many attributes that factor into a house price. For example, environmental attributes can impact the price substantially. Gardens, pleasant views, and attractive landscaping can all increase the house value. Neighborhood attributes such as schools and public services also play a factor.

The data set deals mostly with physical characteristics of the house itself.One key characteristic that should be taken into account is the correlation between the prices of neighboring houses. This will not be addressed by my project because of complexity and since I do not have specific locations for our data set.

Another interesting approach to modeling house prices is hedonic regression. This method breaks up an item into its constituent parts and tries to predict the target value based on how much individual parts contribute to it. A house is a perfect heterogeneous good to be predicted by hedonic regression. It can be broken up into various characteristics such as number of bedrooms, distance to the city center, environmental features, etc.

# Methodology

### Data Description

The data set includes 2,910 observation and 79 independent variables. Out of those 36 are numeric, such as lot area or pool area in square feet, and 43 are categorical, such as garage type (attached to home, built-in, carport, etc.) or utilities (gas, sewer, both, etc.). The data set is split into 1,460 observations comprising the training set and 1,459 observations representing the testing set.

### Data Imputation

Original data set included no complete observations (*see table 3*). However, according to the data dictionary, many of theNA values have a specific meaning. For example, NA in the PoolQC variable (pool quality) implies that the property has no pool. Often this logic carried across multiple variables - for example, NA in GarageQual (garage quality), GarageCond (garage condition) and GarageType variables all indicate that the property has no garage. This type of missing value was replaced with a new category - *No Pool*, *No Garage* or similar. This work was accomplished using the forcats R package.

After this substitution the number of complete observations increased significantly to 2,861 or about 98% of all observations. There remained only 58 observations with true missing values. These observations contained 180 missing values in 32 variables. None of the variables contained a large number of missing values. The top one was MasVnrType with 24 observations containing NA (0.8% of all observations). None of the variables were close to the 5% missing threshold that would suggest that I should drop them from analysis.

There are three ways to deal with missing values :

* **Deleting the variables:** If the missingness is concentrated in a relatively small number of variables, then deleting the variables may be a good option. The downside to this approach is that I lose the opportunity to include the observed values in the model.
* **Imputation via mean, median and mode:** An expedient way to retain all of the cases and variables is to insert the mean or median for continuous variables or the mode for categorical or discrete variables. This approach may suffice for a small number of values, but has the potential to introduce bias in the form of decreasing the variance.
* **Prediction:** This more advanced approach involves using the other variables to predict the missing values. Without validating the results, this method also has the potential to introduce bias.

For my data set I used multiple imputation by chained equations (MICE). The technique involves imputing multiple iterations of values in order to account for statistical uncertainty with standard errors. Since it uses chained equations, MICE has the ability to impute both numerical and categorical variables. The ideal scenario to use MICE is when less than 5% of the values are missing and when values are missing at random. I used the mice R package with the cart (classification and regression trees) method. cart is one of the five mice methods that can impute both numerical and categorical variables. Figure 3 shows the density plots of the observed and imputed values. The imputed distributions have more variance and extremes than the observed distributions.

### Additional Data Preparation

All categorical variables were inspected and their order (or order of levels in R) was changed to match the most likely low-to-high order. These variables for the most part do not rely on the order of categories, so this step was not critical to modeling; however, it makes the modeling output more readable and easier to interpret.

As is the case with most data sets, I found several values that were clearly typos and input errors. For instance, one observation had the year when garage was built listed as 2207. There were 6 negative values in age related variables (see data transformations below). Those were set to 0.

### Data Transformation

Prior to modeling, I have extensively analyzed available variables and took a few approaches to variable transformations. They were meant to both simplify existing variables and add new variables that may be helpful in modeling.

Generally, it is more common to think about the age of the house than the year it was built. Each age related variable was stored in the data set in two related variables - year built and year sold.For house age the value was .

Because I am not dealing with a time series data set, I have converted YrSold and MoSold variables from numeric to nominal.

Using the side-by-side box plots in Figure 4, I examined the categorical variables with more than two values to see if the variable can be simplified by combining the values into two groups. My criteria for this simplification was if the variables’ inner quartile ranges of the response variable distinctly and logically bifurcate. For example, in FireplaceQu (fireplace quality), HeatingQC (heating quality) and PoolQC (pool quality), I noticed that the inner quartiles are bifurcated into two groups that do not overlap: the highest *Excellent* value and all other lesser quality conditions. Additional values that are distinct from other values in the same variables are the *Wood Shingle* value in the roof material variable (RoofMat1), the above average values in the garage quality variable (GarageQual), the gas-related values in the heating variable (Heating), and the *Partial* value in the sale condition variable (SaleCondition).

I tried to examine whether the modeling would benefit from transforming any of the predictor variables. I took natural logarithms, square roots and squares of the numerical variables, and then I calculated every possible pairwise interaction between these transformations, the original numerical variables and categorical variables. I then calculated all pairwise correlations between the interactions and the response variable SalePrice. The top correlations can be seen in table 6, which is sorted in descending order by R-squared. I clearly observed that there are several correlation values higher than the highest correlation between the original predictors and the response, which is OverallQual at 0.79 (*see table 5*). I added the top five interactions to our training data set.

I created several training data sets. The **first** of the three training data sets I created includes only the original variables with the missing values imputed. In model building and selection this set is referred to as the *original* data set. The **second** training data set includes seven “simplified” dummy variables instead of original variables. It also includes the five highly-correlated interactions. This set is referred to as the *transformed* data set. The **third** training data set includes the same predictor variables as in the second set with a transformed response variable. While creating all interactions, I noticed that the correlation values appeared to increase vis-a-vis the square root of the response variable. Consequently, since the response variable contains only positive values, I created a simple BIC step model and used it to calculate the Box-Cox value and transform the response variable. According to Box-Cox, a value of approximately 0.184 should help the final model meet the normality assumption. This set is referred as the *Box-Cox* data set.

### Modeling

Since I am dealing with trying to predict a continuous variable, house sale price, I relied upon building and optimizing general linear model.

After fitting three baseline (all k-parameters) models to all three training data sets, ANOVA demonstrated statistical significance between the original data set and the transformed data set. While all multiple values were within some negligible deviation of each other, adding a Box-Cox transformation of the response variable improved the beyond the model based on the original data set.

I took the strongest model, and applied step-wise regression. I applied backward elimination in order to settle on a model with the lowest Akaike information criterion (AIC) value.

I ended up with six representative models:

1. **Model 1** is based on the fully transformed data set with Box-Cox transformed response variable. It includes all available predictor variables including any interactions created in data preparation. This model explains nearly 94% of variability of the response variable. A good starting point, however, I can remove some insignificant variables for a more parsimonious model with a decreased risk of overfitting.
2. **Model 2** is based on Model 1 modified with step-wise regression (backward elimination). It is an improvement with lower number of parameters (156 comparing to 237). The multiple value is similar. Comparing two models using ANOVA indicates that they are not significantly different.
3. **Model 3** selects only statistically highly significant variables from the previous model (p-value is nearly 0). drops and F-statistic rises, so even though the model is simpler with only 58 parameters, it may not be an improvement. Comparing this model with the first one using ANOVA, shows that there is significant difference between the two.
4. **Model 4** expands on the previous model by using statistically significant variables, but with less strict criteria (p-value < 0.01). The number of parameters is increased, but is also increased. Similarly, per ANOVA, this model is significantly different from models 1 and 3.
5. **Model 5** takes variables identified in the previous model, but it is trained on the original data set without interactions. It uses only log-transformation of LotArea predictor variable and SalePrice response variable. This model represents the best results based on for any model I have tried using the original data set.
6. **Model 6** is based on Model 4, but it is trained on the transformed data set that includes interactions, but not the Box-Cox transformation of the response variable. Similarly to model 5, this model uses log-transformed LotArea and SalePrice.

For all models the F-statistic’s p-value shows a drastic improvement over an intercept-only model, so I can infer that these models are statistically significant.

The table below summarizes the models. I can see a steady improvement in AIC numbers for models 1 through 6. Adjusted fluctuates, but it remains high, and the values are close between various models. The fluctuation in is not enough to be the deciding factor in selecting a model.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Multiple R^2 | Adjusted R^2 | AIC | Kaggle Score |
| Model 1 (Box-Cox) | 0.9359 | 0.9241 | -531 | NA |
| Model 2 (Box-Cox) | 0.9330 | 0.9252 | -617 | NA |
| Model 3 (Box-Cox) | 0.8934 | 0.8890 | -126 | NA |
| Model 4 (Box-Cox) | 0.9193 | 0.9131 | -440 | NA |
| Model 5 (Original) | 0.8935 | 0.8857 | -1604 | 0.1475 |
| Model 6 (Transformed) | 0.9183 | 0.9120 | -1982 | 0.1385 |

Models 1 through 4 rely on the Box-Cox transformation of the response variable with value of 0.184. Although this may slightly improve a model, it makes conversion of prediction difficult and confusing. Log-transformation of the response variable used by models 5 and 6 is significantly easier to implement. As such slight improvement of the Box-Cox model is not enough to justify added complexity of making predictions. It is important to consider how the model will be implemented and simplicity matters.

Model 6 is our primary linear regression model to predict house sale prices.

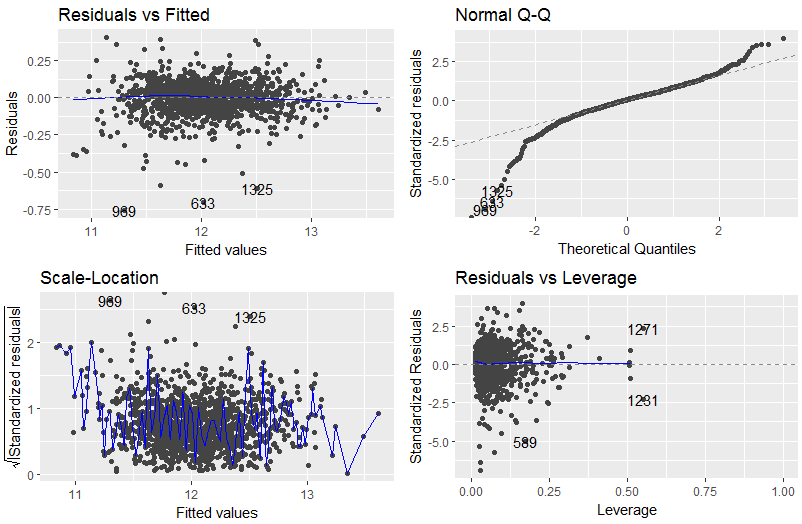
The model was tuned using k-fold validation (with 10 folds). It was run through full diagnostic and four leverage points have been identified. These observations were removed from the training set as it is very plausible that the sales data set has some uncharacteristic outliers.

Additionally, two categorical variables - Condition2 and Utilities - were removed from the model because they did not have enough samples in each category. These variables may be meaningful for predicting the response variable, but there is not enough information in our small data set for these variable to train on.

The final model had multiple of 0.9276, adjusted of 0.9225, AIC of -2172 . These are the best values in all categories.

Figure 1 shows the model diagnostics. There is no discernible pattern among residuals. The Q-Q plot shows some problems with distribution tails; however, this is not unexpected with sales data - there are bound to be some really good and some really bad deals out there. Major leverage points have been accounted for.

Data preparation and transformation, including response variable transformation, model building and re-building as well as model diagnostics resulted in a linear regression model to predict house sale prices.



Model diagnostics.

# Experimentation and Results

My model has 98 parameters. The majority of those are dummy variables created for categorical variables. Table 2 lists all parameters and corresponding coefficients. Please note that our response variable is log-transformed, so coefficients will only give a general idea of the impact on the sale price, and not a specific amount.

Linear regression takes the following form:

Here, is the predicted outcome, through are selected parameters, through are corresponding coefficients, is the intercept and is the error or noise term. The formula used in R is as follows:

log(SalePrice) ~ OverallCond + Condition1 + MSZoning + X1stFlrSF + X2ndFlrSF +   
 LowQualFinSF + Neighborhood + KitchenQual + Fireplaces + WoodDeckSF +   
 Functional + FullBath + BsmtFullBath + BsmtFinType1 + BsmtExposure + BsmtQual +  
 LandSlope + LandContour + log(LotArea) + LotFrontage + LotConfig + HouseStyle +   
 RoofStyle + MasVnrArea + ScreenPorch + House\_Age\_Yrs + RoofMatl\_WdShngl +  
 GarageQual\_abv\_avg + OverallQual2\_x\_GrLivArea + OverallQual2\_x\_TotRmsAbvGrd\_log +   
 OverallQual2\_x\_GarageCars

Linear model parameters and coefficients (sorted by category and coefficient).

|  |  |  |
| --- | --- | --- |
| Category | Details | Coefficient |
|  |  | 9.736404 |
| Amenities | Basement exposure - Good | 0.051112 |
| Amenities | Full bathrooms - Basement | 0.040449 |
| Amenities | Number of fireplaces | 0.030267 |
| Amenities | Full bathrooms - Above grade | 0.007889 |
| Amenities | Basement exposure - Minimum | 0.007124 |
| Amenities | Size of garage | 0.000961 |
| Amenities | Basement exposure - None | -0.000683 |
| Amenities | Basement exposure - No basement | NA |
| Condition | Normal | 0.079479 |
| Condition | Near park, greenbelt, etc. | 0.078508 |
| Condition | Within 200’ of East-West railroad | 0.058670 |
| Condition | Adjacent to North-South railroad | 0.056679 |
| Condition | Overall condition of the house | 0.052166 |
| Condition | Within 200’ of North-South railroad | 0.048134 |
| Condition | Adjacent to feeder street | 0.034262 |
| Condition | Adjacent to park, greenbelt, etc. | 0.008202 |
| Condition | Age | -0.003162 |
| Condition | Adjacent to East-West railroad | -0.062048 |
| Functionality | Typical | 0.112427 |
| Functionality | Minor deductions 2 | 0.072990 |
| Functionality | Minor deductions 1 | 0.062419 |
| Functionality | Moderate deductions | 0.007538 |
| Functionality | Major deductions 2 | -0.144502 |
| Functionality | Severely damaged | -0.390187 |
| House | Roof material - Wood shingles | 0.072151 |
| House | 2.5 story: 2nd level unfinished | 0.059545 |
| House | Roof - Mansard | 0.045818 |
| House | 1.5 story: 2nd level unfinished | 0.023669 |
| House | Total rooms above grade | 0.001667 |
| House | Above grade living area | -0.000001 |
| House | 1 story | -0.001891 |
| House | Roof - Hip | -0.005659 |
| House | Split foyer | -0.010301 |
| House | Roof - Gable | -0.010775 |
| House | 2 story | -0.016187 |
| House | Roof - Shed | -0.017076 |
| House | Split level | -0.022143 |
| House | Roof - Gambrel | -0.024712 |
| House | 2.5 story: 2nd level finished | -0.073665 |
| Land | Land contour - Hillside | 0.038803 |
| Land | Land contour - Near flat | 0.028476 |
| Land | Land contour - Depression | 0.026734 |
| Land | Land slope - Moderate | 0.007487 |
| Land | Land slope - Severe | -0.085362 |
| Lot | Lot area | 0.091273 |
| Lot | Cul-de-sac | 0.022886 |
| Lot | Feet of street connected to property | 0.000066 |
| Lot | Inside lot | -0.004728 |
| Lot | Frontage on 2 sides | -0.026650 |
| Lot | Frontage on 3 sides | -0.075301 |
| Neighborhood | Stone Brooke | 0.096771 |
| Neighborhood | Crawford | 0.086667 |
| Neighborhood | Northridge Heights | 0.039785 |
| Neighborhood | Somerset | 0.030671 |
| Neighborhood | South & West of Iowa State U | 0.013564 |
| Neighborhood | Clear Creek | 0.011456 |
| Neighborhood | Brookside | 0.010711 |
| Neighborhood | Northridge | 0.007573 |
| Neighborhood | Veenker | 0.003404 |
| Neighborhood | Northpark Villa | -0.019407 |
| Neighborhood | College Creek | -0.020850 |
| Neighborhood | Bluestem | -0.027089 |
| Neighborhood | Iowa DOT and Rail Road | -0.030277 |
| Neighborhood | Gilbert | -0.037993 |
| Neighborhood | Old Town | -0.039175 |
| Neighborhood | North Ames | -0.047091 |
| Neighborhood | Timberland | -0.048063 |
| Neighborhood | Sawyer West | -0.051714 |
| Neighborhood | Sawyer | -0.059436 |
| Neighborhood | Briardale | -0.059884 |
| Neighborhood | Edwards | -0.082427 |
| Neighborhood | Northwest Ames | -0.083887 |
| Neighborhood | Mitchell | -0.086347 |
| Neighborhood | Meadow Village | -0.151122 |
| Quality | Garage - above average | 0.076615 |
| Quality | Finished basement - Good living quarters | 0.008018 |
| Quality | Finished basement - Below average | -0.010040 |
| Quality | Finished basement - Low quality | -0.013167 |
| Quality | Finished basement - Average rec room | -0.018644 |
| Quality | Finished basement - Unfinished | -0.042924 |
| Quality | Kitchen - Good | -0.051366 |
| Quality | Basement - Good | -0.051999 |
| Quality | Basement - Fair | -0.052521 |
| Quality | Basement - Typical | -0.074473 |
| Quality | Kitchen - Average | -0.074507 |
| Quality | Kitchen - Fair | -0.124179 |
| Quality | Finished basement - None | -0.249115 |
| Quality | Basement - None | NA |
| Sq Footage | First floor | 0.000363 |
| Sq Footage | Second floor | 0.000290 |
| Sq Footage | Screen porch | 0.000238 |
| Sq Footage | Low quality | 0.000200 |
| Sq Footage | Wood deck | 0.000048 |
| Sq Footage | Masonry veneer | -0.000004 |
| Zoning | Floating Village Residential | 0.511221 |
| Zoning | Residential Low Density | 0.469410 |
| Zoning | Residential High Density | 0.449200 |
| Zoning | Residential Medium Density | 0.442153 |

The attribute with the most negative effect on the sale price is severely damaged functionality of the house. Major deductions in functionality (a step above *Severely Damaged*) is also included in the top five negative attributes. Interestingly, typical functionality has one of the most positive influences on the sale price. Clearly and not surprisingly, general house condition that ranges from *Typical* to *Salvage Only*, is very important to the final sale price.

Another attribute that influences the sale price negatively is the Meadow Village neighborhood. Reviewing Ames, Iowa information, I can quickly notice that the Meadow Village despite its peaceful name is actually on the edge of town next to the airport. This would be a less desirable neighborhood in any town. On the opposite side of town, next to a golf club and to the Ada Hayden Heritage park and lake, is the Stone Brooke neighborhood. This is one of the most positive attributes. Again not surprisingly, the most positive and negative neighborhoods are on completely opposite sides of town.

Other major negative attributes are lack of finished basement and fair quality of kitchen. Zoning attributes make up other major positive attributes.

The model confirms a few attributes that can be considered common sense. For example, location adjacent to the railroad has a negative influence on the sale price. However, this only applies to the East-West railroad. The impact of the North-South railroad is positive. It is important to note that the East-West railroad cuts through the middle of the town while the North-South railroad only affects a small portion, and it is clearly a less used railroad.

Other notable factors are the following:

* House age has a negative influence, but not as great as some other attributes.
* Being on a cul-de-sac is the most positive lot location. Inside lot has a slightly negative effect, and a lot with streets on 2 sides has even more negative coefficient. Finally, a lot with streets on 3 sides has the biggest negative impact.
* An unfinished basement generally has a negative impact as does a low-quality kitchen.
* The square footage of several characteristics, such as first or second floor, wood deck, porch, is not important (but the impact is slightly positive).
* 1.5 and 2.5 story houses seem to be preferred; however, this variable is probably highly dependent on the type of houses available in Ames, and it could be representative of some other features.

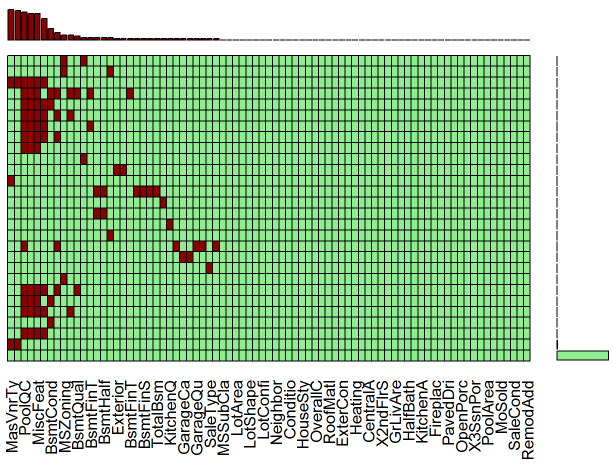
# Discussion and Conclusion

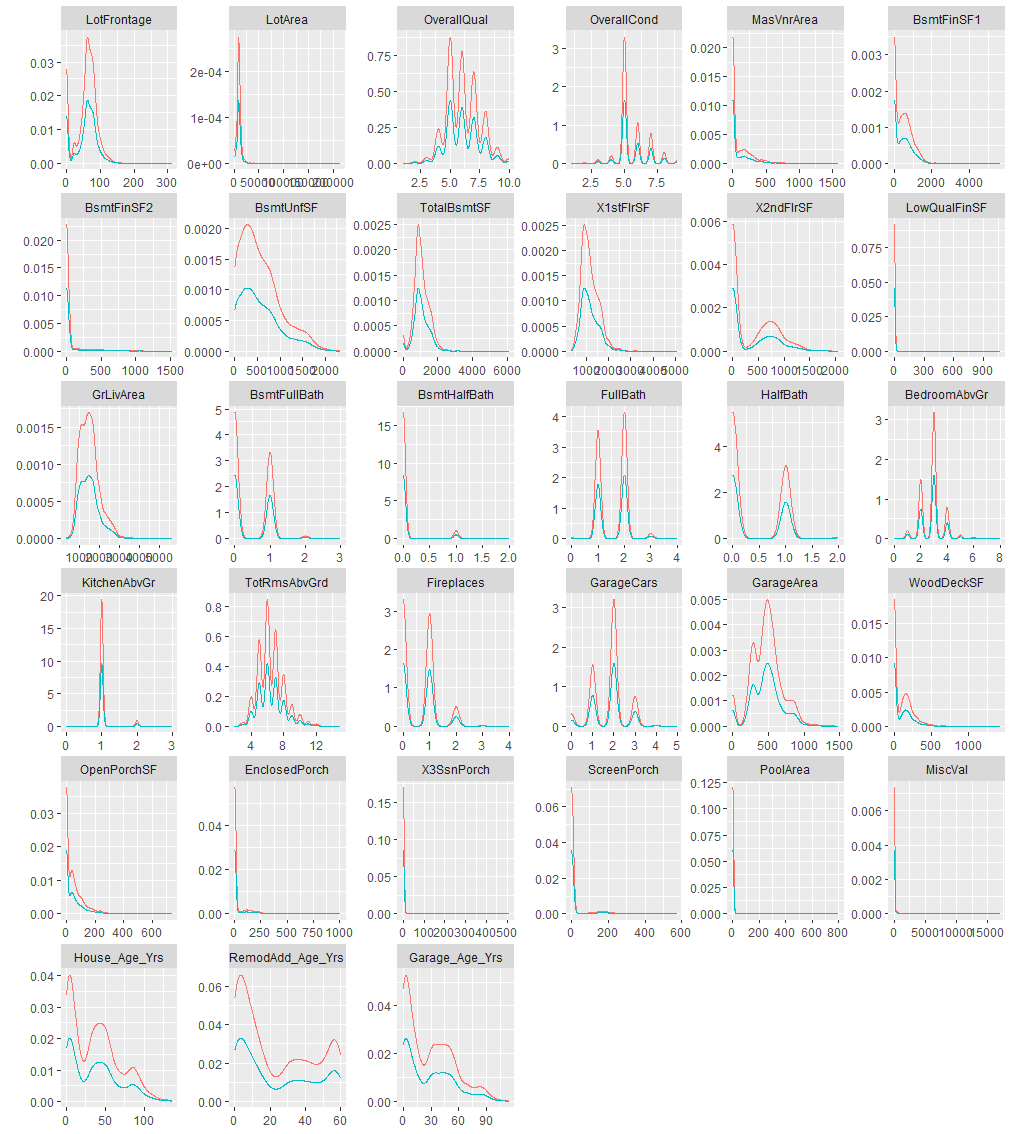
Location and condition seem to be major attributes in house sale prices. There is seasonality to number of houses sold, especially considering that Ames is a college town; however, it does not appear to be a major factor.

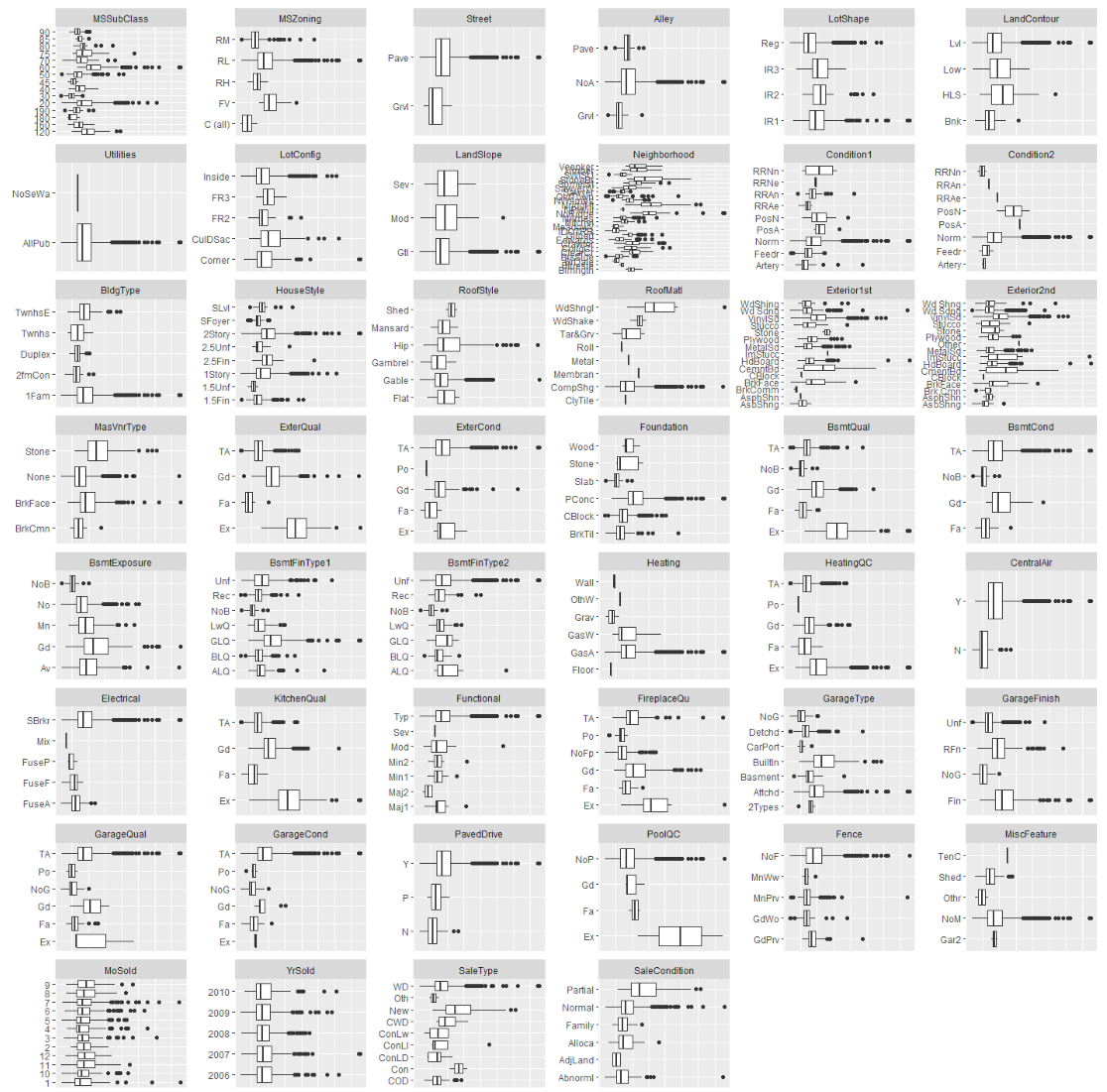
The biggest limiting factor of my study is that it is based on one Midwest town’s data. This model may not transfer well to other locations. After all it is not hard to imagine that the driving factors in Ames, Iowa are quite different from factors in New York, NY or Los Angeles, California. It is a helpful starting point, but may not be widely applicable.

My research is based exclusively on linear regression. Other methods, such as Support Vector Machines or Random Forrest, may be able to achieve much better results. This data set provides a rich opportunity for research, and it would be beneficial to try other methods for comparison purposes.

# Appendix A. Figures







# Appendix B. Tables

Number of NA values in original data.

|  |  |  |
| --- | --- | --- |
| Variable | No of NAs | Percent of Total Obs |
| PoolQC | 2909 | 99.66 |
| MiscFeature | 2814 | 96.40 |
| Alley | 2721 | 93.22 |
| Fence | 2348 | 80.44 |
| FireplaceQu | 1420 | 48.65 |
| LotFrontage | 486 | 16.65 |
| GarageYrBlt | 159 | 5.45 |
| GarageFinish | 159 | 5.45 |
| GarageQual | 159 | 5.45 |
| GarageCond | 159 | 5.45 |
| Garage\_Age\_Yrs | 159 | 5.45 |
| GarageType | 157 | 5.38 |
| BsmtCond | 82 | 2.81 |
| BsmtExposure | 82 | 2.81 |
| BsmtQual | 81 | 2.77 |
| BsmtFinType2 | 80 | 2.74 |
| BsmtFinType1 | 79 | 2.71 |
| MasVnrType | 24 | 0.82 |
| MasVnrArea | 23 | 0.79 |
| MSZoning | 4 | 0.14 |
| Utilities | 2 | 0.07 |
| BsmtFullBath | 2 | 0.07 |
| BsmtHalfBath | 2 | 0.07 |
| Functional | 2 | 0.07 |
| Exterior1st | 1 | 0.03 |
| Exterior2nd | 1 | 0.03 |
| BsmtFinSF1 | 1 | 0.03 |
| BsmtFinSF2 | 1 | 0.03 |
| BsmtUnfSF | 1 | 0.03 |
| TotalBsmtSF | 1 | 0.03 |
| Electrical | 1 | 0.03 |
| KitchenQual | 1 | 0.03 |
| GarageCars | 1 | 0.03 |
| GarageArea | 1 | 0.03 |
| SaleType | 1 | 0.03 |

Descriptive statistics for numerical variables.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Variable | Count | Mean | SD | Median | Min | Max | Kurtosis |
| LotFrontage | 2919 | 57.77 | 33.48 | 63 | 0 | 313 | 2.169 |
| LotArea | 2919 | 10168 | 7887 | 9453 | 1300 | 215245 | 264.3 |
| OverallQual | 2919 | 6.089 | 1.41 | 6 | 1 | 10 | 0.06295 |
| OverallCond | 2919 | 5.565 | 1.113 | 5 | 1 | 9 | 1.472 |
| YearBuilt | 2919 | 1971 | 30.29 | 1973 | 1872 | 2010 | -0.5142 |
| YearRemodAdd | 2919 | 1984 | 20.89 | 1993 | 1950 | 2010 | -1.347 |
| MasVnrArea | 2896 | 102.2 | 179.3 | 0 | 0 | 1600 | 9.228 |
| BsmtFinSF1 | 2918 | 441.4 | 455.6 | 368.5 | 0 | 5644 | 6.884 |
| BsmtFinSF2 | 2918 | 49.58 | 169.2 | 0 | 0 | 1526 | 18.79 |
| BsmtUnfSF | 2918 | 560.8 | 439.5 | 467 | 0 | 2336 | 0.3985 |
| TotalBsmtSF | 2918 | 1052 | 440.8 | 989.5 | 0 | 6110 | 9.125 |
| X1stFlrSF | 2919 | 1160 | 392.4 | 1082 | 334 | 5095 | 6.936 |
| X2ndFlrSF | 2919 | 336.5 | 428.7 | 0 | 0 | 2065 | -0.4254 |
| LowQualFinSF | 2919 | 4.694 | 46.4 | 0 | 0 | 1064 | 174.5 |
| GrLivArea | 2919 | 1501 | 506.1 | 1444 | 334 | 5642 | 4.108 |
| BsmtFullBath | 2917 | 0.4299 | 0.5247 | 0 | 0 | 3 | -0.738 |
| BsmtHalfBath | 2917 | 0.06136 | 0.2457 | 0 | 0 | 2 | 14.81 |
| FullBath | 2919 | 1.568 | 0.553 | 2 | 0 | 4 | -0.5409 |
| HalfBath | 2919 | 0.3803 | 0.5029 | 0 | 0 | 2 | -1.035 |
| BedroomAbvGr | 2919 | 2.86 | 0.8227 | 3 | 0 | 8 | 1.933 |
| KitchenAbvGr | 2919 | 1.045 | 0.2145 | 1 | 0 | 3 | 19.73 |
| TotRmsAbvGrd | 2919 | 6.452 | 1.569 | 6 | 2 | 15 | 1.162 |
| Fireplaces | 2919 | 0.5971 | 0.6461 | 1 | 0 | 4 | 0.07213 |
| GarageYrBlt | 2918 | 2412 | 1816 | 1984 | 1895 | 9999 | 13.51 |
| GarageCars | 2918 | 1.767 | 0.7616 | 2 | 0 | 5 | 0.2335 |
| GarageArea | 2918 | 472.9 | 215.4 | 480 | 0 | 1488 | 0.9334 |
| WoodDeckSF | 2919 | 93.71 | 126.5 | 0 | 0 | 1424 | 6.721 |
| OpenPorchSF | 2919 | 47.49 | 67.58 | 26 | 0 | 742 | 10.91 |
| EnclosedPorch | 2919 | 23.1 | 64.24 | 0 | 0 | 1012 | 28.31 |
| X3SsnPorch | 2919 | 2.602 | 25.19 | 0 | 0 | 508 | 149 |
| ScreenPorch | 2919 | 16.06 | 56.18 | 0 | 0 | 576 | 17.73 |
| PoolArea | 2919 | 2.252 | 35.66 | 0 | 0 | 800 | 297.9 |
| MiscVal | 2919 | 50.83 | 567.4 | 0 | 0 | 17000 | 562.7 |
| MoSold | 2919 | 6.213 | 2.715 | 6 | 1 | 12 | -0.4574 |
| YrSold | 2919 | 2008 | 1.315 | 2008 | 2006 | 2010 | -1.156 |
| House\_Age\_Yrs | 2919 | 36.48 | 30.34 | 35 | -1 | 136 | -0.5058 |
| RemodAdd\_Age\_Yrs | 2919 | 23.53 | 20.89 | 15 | -2 | 60 | -1.339 |
| Garage\_Age\_Yrs | 2918 | 28.07 | 25.8 | 25 | -200 | 114 | 1.614 |

Predictor variables most correlated with original response variable.

|  |  |  |  |
| --- | --- | --- | --- |
| Predictor Variable | Response Variable | Correlation | R^2 |
| OverallQual | SalePrice | 0.79 | 0.63 |
| GrLivArea | SalePrice | 0.71 | 0.50 |
| GarageCars | SalePrice | 0.64 | 0.41 |
| GarageArea | SalePrice | 0.62 | 0.39 |
| TotalBsmtSF | SalePrice | 0.61 | 0.38 |
| X1stFlrSF | SalePrice | 0.61 | 0.37 |
| FullBath | SalePrice | 0.56 | 0.31 |
| TotRmsAbvGrd | SalePrice | 0.53 | 0.28 |
| House\_Age\_Yrs | SalePrice | -0.52 | 0.27 |
| RemodAdd\_Age\_Yrs | SalePrice | -0.51 | 0.26 |

Predictor transformations most correlated with transformed response variable.

|  |  |  |  |
| --- | --- | --- | --- |
| Response Variable | Predictor Transformation | Correlation | R^2 |
| SalePrice\_sqrt | LotArea\_log:OverallQual | 0.856 | 0.732 |
| SalePrice\_sqrt | GrLivArea\_log:OverallQual | 0.852 | 0.727 |
| SalePrice\_sqrt | OverallQual\_2:GarageCars | 0.851 | 0.724 |
| SalePrice\_sqrt | OverallQual\_sqrt:GarageCars | 0.851 | 0.724 |
| SalePrice\_sqrt | OverallQual\_2:TotRmsAbvGrd\_log | 0.851 | 0.724 |
| SalePrice\_sqrt | OverallQual\_sqrt:TotRmsAbvGrd\_log | 0.851 | 0.724 |
| SalePrice\_sqrt | X1stFlrSF\_log:OverallQual | 0.851 | 0.724 |
| SalePrice\_sqrt | OverallQual\_2:LotArea\_log | 0.850 | 0.723 |
| SalePrice\_sqrt | OverallQual\_sqrt:LotArea\_log | 0.850 | 0.723 |
| SalePrice\_sqrt | OverallQual\_2:GrLivArea\_log | 0.847 | 0.717 |
| SalePrice\_sqrt | OverallQual\_sqrt:GrLivArea\_log | 0.847 | 0.717 |
| SalePrice\_sqrt | OverallQual\_2:X1stFlrSF\_log | 0.844 | 0.713 |
| SalePrice\_sqrt | OverallQual\_sqrt:X1stFlrSF\_log | 0.844 | 0.713 |
| SalePrice\_sqrt | TotRmsAbvGrd\_log:OverallQual | 0.841 | 0.707 |
| SalePrice\_sqrt | OverallQual\_log:GrLivArea\_log | 0.837 | 0.700 |
| SalePrice\_sqrt | OverallQual\_2:TotRmsAbvGrd | 0.835 | 0.698 |
| SalePrice\_sqrt | OverallQual\_sqrt:TotRmsAbvGrd | 0.835 | 0.698 |
| SalePrice\_sqrt | OverallQual\_log:X1stFlrSF\_log | 0.834 | 0.695 |
| SalePrice\_sqrt | OverallQual\_2 | 0.828 | 0.685 |
| SalePrice\_sqrt | OverallQual\_sqrt | 0.828 | 0.685 |
| SalePrice\_sqrt | OverallQual:GrLivArea | 0.827 | 0.685 |
| SalePrice\_sqrt | UtilitiesAllPub:OverallQual\_2 | 0.827 | 0.684 |
| SalePrice\_sqrt | UtilitiesAllPub:OverallQual\_sqrt | 0.827 | 0.684 |
| SalePrice\_sqrt | OverallQual\_2:OverallQual\_log | 0.827 | 0.684 |
| SalePrice\_sqrt | OverallQual\_sqrt:OverallQual\_log | 0.827 | 0.684 |
| SalePrice\_sqrt | LotArea\_log:OverallQual\_log | 0.827 | 0.683 |
| SalePrice\_sqrt | OverallQual:GarageCars | 0.826 | 0.682 |
| SalePrice\_sqrt | OverallQual\_2:GrLivArea | 0.826 | 0.682 |
| SalePrice\_sqrt | OverallQual\_sqrt:GrLivArea | 0.826 | 0.682 |
| SalePrice\_sqrt | StreetPave:OverallQual\_2 | 0.825 | 0.680 |
| SalePrice\_sqrt | StreetPave:OverallQual\_sqrt | 0.825 | 0.680 |
| SalePrice\_sqrt | OverallQual\_2:GarageArea | 0.823 | 0.678 |
| SalePrice\_sqrt | OverallQual\_sqrt:GarageArea | 0.823 | 0.678 |
| SalePrice\_sqrt | OverallQual\_log:OverallQual | 0.823 | 0.677 |
| SalePrice\_sqrt | OverallQual\_2:OverallQual | 0.822 | 0.676 |
| SalePrice\_sqrt | OverallQual\_sqrt:OverallQual | 0.822 | 0.676 |
| SalePrice\_sqrt | OverallQual\_2:TotalBsmtSF\_log | 0.822 | 0.675 |
| SalePrice\_sqrt | OverallQual\_sqrt:TotalBsmtSF\_log | 0.822 | 0.675 |
| SalePrice\_sqrt | OverallQual\_2:FullBath | 0.821 | 0.674 |
| SalePrice\_sqrt | OverallQual\_sqrt:FullBath | 0.821 | 0.674 |
| SalePrice\_sqrt | CentralAirY:OverallQual\_2 | 0.817 | 0.667 |
| SalePrice\_sqrt | CentralAirY:OverallQual\_sqrt | 0.817 | 0.667 |
| SalePrice\_sqrt | OverallQual | 0.816 | 0.666 |
| SalePrice\_sqrt | UtilitiesAllPub:OverallQual | 0.813 | 0.660 |
| SalePrice\_sqrt | Condition2Norm:OverallQual\_2 | 0.812 | 0.659 |
| SalePrice\_sqrt | Condition2Norm:OverallQual\_sqrt | 0.812 | 0.659 |
| SalePrice\_sqrt | OverallQual\_2:OverallCond\_log | 0.810 | 0.656 |
| SalePrice\_sqrt | OverallQual\_sqrt:OverallCond\_log | 0.810 | 0.656 |
| SalePrice\_sqrt | OverallQual\_2:GarageCars\_2 | 0.809 | 0.655 |
| SalePrice\_sqrt | OverallQual\_2:GarageCars\_sqrt | 0.809 | 0.655 |

# Appendix C. R Code

install\_load <- function(pkg){  
 new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]  
 if (length(new.pkg)) install.packages(new.pkg, dependencies = TRUE)  
 sapply(pkg, require, character.only = TRUE, quietly = TRUE, warn.conflicts = FALSE)  
}  
# required packages  
packages <- c("tidyverse","knitr", "mice", "VIM", "RCurl", "knitcitations",   
 "janitor", "missForest", "DMwR", "splitstackshape", "car")  
install\_load(packages)  
# Read data  
url\_train <- paste0("https://raw.githubusercontent.com/omerozeren/DATA621/master/final\_project/train.csv")  
url\_test <- paste0("https://raw.githubusercontent.com/omerozeren/DATA621/master/final\_project/test.csv")  
stand\_read <- function(url){  
 return(read.csv(text = getURL(url)))  
}  
o\_train <-   
 stand\_read(url\_train) %>%   
 mutate(d\_name = 'train')  
o\_test <- stand\_read(url\_test) %>%   
 mutate(SalePrice = NA, d\_name = 'test')  
full\_set <- rbind(o\_train, o\_test)  
na\_review <- function(df){  
 # returns df of vars w/ NA qty desc.  
 na\_qty <- colSums(is.na(df)) %>% as.data.frame(stringsAsFactors=F)  
 colnames(na\_qty) <- c("NA\_qty")  
 na\_qty <- cbind('Variable' = rownames(na\_qty), na\_qty) %>%   
 select(Variable, NA\_qty)  
 rownames(na\_qty) <- NULL  
   
 na\_qty <- na\_qty %>%   
 arrange(desc(NA\_qty)) %>% filter(NA\_qty > 0) %>%   
 mutate(Variable = as.character(Variable)) %>%   
 mutate(Pct\_of\_Tot = round(NA\_qty/nrow(df), 4) \* 100)  
   
 return(na\_qty)  
}  
first\_pass <-   
 full\_set %>%   
 # first\_pass is train.csv and test.csv combined for NA reviews   
 # and imputation planning and calculated columns  
 mutate(House\_Age\_Yrs = YrSold - YearBuilt,   
 RemodAdd\_Age\_Yrs = YrSold - YearRemodAdd,   
 Garage\_Age\_Yrs = YrSold - GarageYrBlt)   
naVars <- na\_review(first\_pass %>% select(-SalePrice))  
naVars  
set\_aside <- c(2600, 2504, 2421, 2127, 2041, 2186, 2525, 1488, 949, 2349, 2218, 2219, 333)  
set\_asideA <- '2600|2504|2421|2127|2041|2186|2525|1488|949|2349|2218|2219|333' # 13  
set\_asideB <- '|2550|524|2296|2593' # negative values in '\_Age' columns  
x <- first\_pass %>%   
 # exclude set\_aside observations to fill in known NA's  
 filter(!grepl(paste0(set\_asideA, set\_asideB), Id))  
   
naVarsx <- na\_review(x %>% select(-SalePrice))  
naVarsx  
obtain\_data <- function(df){  
 # like first\_pass but with imputation that addresses   
 # observations that have known NA's  
 df %>%  
 mutate(PoolQC = fct\_explicit\_na(PoolQC, na\_level='NoP'),  
 MiscFeature = fct\_explicit\_na(MiscFeature, na\_level='NoM'),  
 Alley = fct\_explicit\_na(Alley, na\_level='NoA'),  
 Fence = fct\_explicit\_na(Fence, na\_level = 'NoF'),  
 FireplaceQu = fct\_explicit\_na(FireplaceQu, na\_level = 'NoFp'),   
 LotFrontage = ifelse(is.na(LotFrontage), 0, LotFrontage),  
   
 # Note GarageYrBlt set to 9999 may be a problem  
 GarageYrBlt = ifelse(is.na(GarageYrBlt), 9999, GarageYrBlt),   
 GarageFinish = fct\_explicit\_na(GarageFinish, na\_level = 'NoG'),   
 GarageQual = fct\_explicit\_na(GarageQual, na\_level = 'NoG'),   
 GarageCond = fct\_explicit\_na(GarageCond, na\_level = 'NoG'),   
 # NOTE: Garage\_Age\_Yrs: 0 doesn't seem appropriate...   
 Garage\_Age\_Yrs = ifelse(is.na(Garage\_Age\_Yrs), 0, Garage\_Age\_Yrs),  
 GarageType = fct\_explicit\_na(GarageType, na\_level = 'NoG'),   
   
 BsmtQual = fct\_explicit\_na(BsmtQual, na\_level = 'NoB'),  
 BsmtCond = fct\_explicit\_na(BsmtCond, na\_level = 'NoB'),  
 BsmtExposure = fct\_explicit\_na(BsmtExposure, na\_level = 'NoB'),  
 BsmtFinType1 = fct\_explicit\_na(BsmtFinType1, na\_level = 'NoB'),  
 BsmtFinType2 = fct\_explicit\_na(BsmtFinType2, na\_level = 'NoB')  
 )  
}  
probl\_obs <- full\_set %>%   
 mutate(House\_Age\_Yrs = YrSold - YearBuilt,   
 RemodAdd\_Age\_Yrs = YrSold - YearRemodAdd,   
 Garage\_Age\_Yrs = YrSold - GarageYrBlt) %>%   
 filter(grepl(paste0(set\_asideA, set\_asideB), Id))  
known\_obs <- full\_set %>%   
 filter(!grepl(paste0(set\_asideA, set\_asideB), Id)) %>%   
 mutate(House\_Age\_Yrs = YrSold - YearBuilt,   
 RemodAdd\_Age\_Yrs = YrSold - YearRemodAdd,   
 Garage\_Age\_Yrs = YrSold - GarageYrBlt)  
full\_set\_clean <- rbind(obtain\_data(known\_obs), probl\_obs) %>% arrange(Id)  
str(full\_set\_clean)  
#View(full\_set\_clean)  
#summary(full\_set\_clean)  
naVarsy <- na\_review(full\_set\_clean %>% select(-SalePrice))  
sum(naVarsy$NA\_qty) # 176  
# ord\_vars per the Data Dictionary.   
ord\_vars <- c("LotShape","Utilities", "LandSlope", "ExterQual",   
 "ExterCond", "BsmtQual", "BsmtCond", "BsmtExposure",  
 "BsmtFinType1", "BsmtFinType2", "HeatingQC", "Electrical",  
 "KitchenQual", "Functional", "FireplaceQu", "GarageFinish",  
 "GarageQual", "GarageCond", "PavedDrive", "PoolQC", "Fence")  
# Order of levels for ordinal variables   
# all are ordered most favorible to least favorible, below  
LotShape\_ <- c("Reg", "IR1", "IR2", "IR3")   
Utilities\_ <- c("AllPub", "NoSeWa")   
LandSlope\_ <- c("Gtl","Mod", "Sev")   
ExterQual\_ <- c("Ex", "Gd", "TA", "Fa")   
ExterCond\_ <- c("Ex", "Gd", "TA", "Fa", "Po")   
BsmtQual\_ <- c("Ex", "Gd", "TA", "Fa", "NoB")   
BsmtCond\_ <- c("Gd", "TA", "Fa", "NoB")   
BsmtExposure\_ <- c("Gd", "Av", "Mn", "No", "NoB")   
BsmtFinType1\_ <- c("GLQ", "ALQ", "BLQ",   
 "Rec", "LwQ", "Unf", "NoB")   
BsmtFinType2\_ <- c("GLQ", "ALQ", "BLQ",   
 "Rec", "LwQ", "Unf", "NoB")   
HeatingQC\_ <- c("Ex", "Gd", "TA", "Fa", "Po")   
Electrical\_ <- c("SBrkr", "FuseA", "FuseF",  
 "FuseP", "Mix")   
KitchenQual\_ <- c("Ex", "Gd", "TA", "Fa")   
Functional\_ <- c("Typ", "Min1", "Min2", "Mod",  
 "Maj1", "Maj2", "Sev")   
FireplaceQu\_ <- c("Ex", "Gd", "TA", "Fa",   
 "Po", "NoFp")   
GarageFinish\_ <- c("Fin", "RFn", "Unf", "NoG")   
GarageQual\_ <- c("Ex", "Gd", "TA", "Fa", "Po",   
 "NoG")   
GarageCond\_ <- c("Ex", "Gd", "TA", "Fa", "Po",   
 "NoG")   
PavedDrive\_ <- c("Y", "P", "N")   
PoolQC\_ <- c("Ex", "Gd", "Fa", "NoP")   
Fence\_ <- c("GdPrv", "MnPrv", "GdWo", "MnWw",  
 "NoF")   
# list of lists of the correct factor levels  
n\_levels <- list(LotShape\_, Utilities\_, LandSlope\_, ExterQual\_,   
 ExterCond\_, BsmtQual\_, BsmtCond\_, BsmtExposure\_,   
 BsmtFinType1\_, BsmtFinType2\_, HeatingQC\_, Electrical\_,   
 KitchenQual\_, Functional\_, FireplaceQu\_, GarageFinish\_,   
 GarageQual\_, GarageCond\_, PavedDrive\_, PoolQC\_, Fence\_)  
names(n\_levels) <- ord\_vars # name vars so I can index  
relevel\_data <- function(df, ord\_list, new\_lvls){  
 # updates factor cols df[ord\_list] with new\_lvls (list of lists)  
 i = sapply(colnames(full\_set\_clean),   
 function (x) x %in% ord\_list) # obtain order list cols  
 df[i] = lapply(df[i], as.character) # convert factors to char  
 for(s\_var in ord\_list){ # correct levels   
 df[[s\_var]] = factor(df[[s\_var]], rev(new\_lvls[[s\_var]]))  
 }  
 return(df)  
}  
full\_set\_clean <- relevel\_data(full\_set\_clean, ord\_vars, n\_levels)  
var\_types <- function(df){  
 # returns df of Variable name and Type from df  
 var\_df <- sapply(df, class) %>% as.data.frame()  
 colnames(var\_df) <- c("Var\_Type")  
 var\_df <- cbind(var\_df, 'Variable' = rownames(var\_df)) %>%   
 select(Variable, Var\_Type) %>%   
 mutate(Variable = as.character(Variable),Var\_Type = as.character(Var\_Type))  
 return(var\_df)  
}  
var\_review <-   
 var\_types(full\_set\_clean %>%   
 select(-c(Id,SalePrice,d\_name)))  
fac\_vars <- var\_review %>%   
 filter(Var\_Type == 'factor') %>%   
 select(Variable) %>%   
 t() %>%   
 as.character()   
# 43 total length(fac\_vars)  
num\_vars <- var\_review %>%   
 filter(grepl('character|integer|numeric', Var\_Type)) %>%   
 select(Variable) %>% t() %>% as.character() # 39 total but see GarageYrBlt   
sum(complete.cases(full\_set %>% select(-SalePrice))) # 0  
sum(complete.cases(full\_set\_clean %>% select(-SalePrice))) # 2,861 ~ 98%  
nrow(full\_set\_clean) - 2861 # 58 NA  
stat\_info <- psych::describe(full\_set\_clean %>% select(num\_vars, -Id, -d\_name))  
stat\_info[c(2:nrow(stat\_info)),c(2:5,8:9,13:ncol(stat\_info)-1)]  
train\_data <- full\_set\_clean %>% filter(d\_name == 'train') %>% select(-d\_name)  
test\_data <- full\_set\_clean %>% filter(d\_name == 'test') %>% select(-d\_name)  
##View(train\_data)  
dim(train\_data)  
dim(test\_data)  
dplyr::filter(full\_set\_clean,   
 House\_Age\_Yrs < 0 | RemodAdd\_Age\_Yrs < 0 | Garage\_Age\_Yrs < 0) %>%   
 dplyr::select(YrSold, YearBuilt, YearRemodAdd, House\_Age\_Yrs, GarageYrBlt,   
 RemodAdd\_Age\_Yrs, Garage\_Age\_Yrs) %>%   
 kable(caption = "Table 3.1: Invalid Negative Values")  
# Mutute Variables  
# bc of the new Age vars, remove the YearBuilt, YearRemodAdd, GarageYrBlt   
# set negative Ages to zero, scaled the YrSold, MoSold as a factor  
full\_set\_clean\_kyle <-   
 full\_set\_clean %>%   
 arrange(desc(d\_name)) %>%   
 dplyr::select(-c(Id, YearBuilt, YearRemodAdd, GarageYrBlt, d\_name)) %>%   
 mutate(  
 House\_Age\_Yrs = pmax(0, House\_Age\_Yrs),  
 RemodAdd\_Age\_Yrs = pmax(0, RemodAdd\_Age\_Yrs),  
 Garage\_Age\_Yrs = pmax(0, Garage\_Age\_Yrs),  
 YrSold = as.ordered(YrSold),  
 MoSold = as.ordered(MoSold),  
 MSSubClass = as.factor(MSSubClass)  
 )  
factor\_differences <-   
 full\_set\_clean %>%   
 mutate(d\_name = factor(d\_name)) %>%   
 select\_if(is.factor) %>%   
 #na.omit() %>%   
 reshape2::melt(id.var = "d\_name") %>%   
 group\_by(d\_name, variable) %>%   
 summarise(unique\_values = length(na.omit(unique(value)))) %>%   
 spread(key = d\_name, value = unique\_values) %>%   
 dplyr::filter(test != train) %>%   
 left\_join(  
 gather(full\_set\_clean) %>%   
 group\_by(key) %>%   
 summarize(NAs = sum(as.integer(is.na(value)))) %>%   
 dplyr::select(variable = key, NAs)  
 )  
kable(factor\_differences,   
 caption = "Table 3.2: Differences in Factor Values between Test & Training Sets")  
#combine data sets for imputation  
predictors\_for\_imputation <-   
 full\_set\_clean\_kyle %>%   
 dplyr::select(-SalePrice)  
# https://www.rdocumentation.org/packages/VIM/versions/4.7.0/topics/aggr  
missing\_plot <- VIM::aggr(predictors\_for\_imputation,  
 #numbers = T,  
 sortVars = T,  
 combine = T,  
 col = c("lightgreen", "darkred", "orange"),  
 labels=str\_sub(names(predictors\_for\_imputation), 1, 8),  
 ylab="Figure 3.1: Missing Values in Train Set"  
 )  
kable(data.frame(complete\_cases\_pct = missing\_plot$percent[1]),  
 caption = "Table 3.3 % of Complete Cases",  
 digits = 1)  
dtypes <- rapply(predictors\_for\_imputation, class)  
dtypes <- data.frame(  
 Variable = names(dtypes),  
 dtype = dtypes  
)  
missing\_summary <-   
 missing\_plot$missings %>%   
 arrange(-Count) %>%   
 janitor::adorn\_totals() %>%   
 mutate(  
 pct\_missing = Count / nrow(predictors\_for\_imputation) \* 100  
 ) %>%  
 filter(pct\_missing > 0) %>%   
 left\_join(dtypes)   
missing\_summary[nrow(missing\_summary), "pct\_missing"] <- NA  
kable(missing\_summary, digits = 3, row.names = T,   
 caption = "Table 3.4 Missing Values by Variable")   
if (!exists("predictors\_imputed")){  
 #https://www.rdocumentation.org/packages/mice/versions/2.46.0/topics/mice  
 mice\_mod <- mice(predictors\_for\_imputation, m = 1, method = "cart", seed = 5)   
 predictors\_imputed <- mice::complete(mice\_mod)  
}  
full\_set\_imputed <-   
 predictors\_imputed %>%   
 mutate(SalePrice = full\_set\_clean\_kyle$SalePrice) %>%   
 droplevels()  
   
train\_data\_imputed <-   
 full\_set\_imputed[1:nrow(train\_data), ]   
test\_data\_imputed <-   
 full\_set\_imputed[nrow(train\_data) + 1:nrow(test\_data), ] %>%   
 dplyr::select(-SalePrice)   
# Visualize the imputations  
# SOURCE: https://stackoverflow.com/questions/12056989/  
# density-plots-with-multiple-groups?utm\_medium=organic&  
# utm\_source=google\_rich\_qa&utm\_campaign=google\_rich\_qa  
# Melt into long format  
# Add a variable for the plot legend  
mice\_data <- mice::complete(mice\_mod, "long", include = TRUE)  
mice\_mod\_viz <-   
 mice\_data %>%   
 select\_if(is.numeric) %>%   
 mutate(Imputed = ifelse(mice\_data$.imp == "0", "Observed", "Imputed")) %>%   
 reshape2::melt("Imputed") %>%   
 na.omit()  
if (!exists("mice\_density\_plot")){  
mice\_density\_plot <-   
 ggplot(mice\_mod\_viz, aes(x=value, colour = factor(Imputed))) +   
 stat\_density(geom = "path") +  
 facet\_wrap(~variable, scales="free") +  
 labs(title = "Figure 3.2: Denisity plots of Observed & Imputed Values")  
}  
mice\_density\_plot  
stripplot(mice\_mod, pch = 20, cex = 1.2,   
 main = "Figure 3.3: Strip Plots of Observed & Imputed Values")  
# http://web.maths.unsw.edu.au/~dwarton/missingDataLab.html  
### Side-by-Side Boxplots of Categorical Variables  
# create data  
boxplot\_data <-   
 train\_data\_imputed %>%   
 select\_if(function(x) !is.numeric(x)) %>%   
 mutate(SalePrice = train\_data\_imputed$SalePrice) %>%   
 reshape2::melt(id.vars = "SalePrice")  
### Boxplots  
ggplot(data = boxplot\_data, aes(x = value, y = SalePrice)) +  
 geom\_boxplot() +  
 facet\_wrap( ~ variable, scales = "free") +  
 coord\_flip() +  
 labs(title = paste0("Figure 3.4: Side-by-Side Box Plots of the Categorical Variables ",  
 "versus the Response"))  
# Reference: https://stackoverflow.com/questions/14604439/  
# plot-multiple-boxplot-in-one-graph?utm\_medium=organic&utm\_source=  
# google\_rich\_qa&utm\_campaign=google\_rich\_qa  
## CORRELATIONS  
# correlation matrix  
train\_data\_numeric <-   
 train\_data\_imputed %>%   
 select\_if(is.numeric)  
cm <- cor(train\_data\_numeric, use = "pairwise.complete.obs")  
#plot  
corrplot::corrplot(cm, method = "square", type = "upper")  
#find the top correlations  
correlation\_df <- function(cm){  
 #Creates a df of pairwise correlations  
 correlations <- c(cm[upper.tri(cm)])  
 cor\_df <- data.frame(  
 Var1 = rownames(cm)[row(cm)[upper.tri(cm)]],  
 Var2 = colnames(cm)[col(cm)[upper.tri(cm)]],  
 Correlation = correlations,  
 Rsquared = correlations^2  
 ) %>%   
 arrange(-Rsquared)  
 return(cor\_df)  
}  
cor\_df <- correlation\_df(cm)  
kable(head(cor\_df, 10), digits = 2, row.names = T,   
 caption = "Top Correlated Variable Pairs")  
kable(head(dplyr::filter(cor\_df, Var1 == "SalePrice" | Var2 == "SalePrice" ), 10),   
 digits = 2, row.names = T, caption = "Top Correlated Variable Pairs")  
# Reference: https://stackoverflow.com/questions/28035001/  
# transform-correlation-matrix-into-dataframe-with-records-for-each-row-column-pai  
### CORRELATIONS WITH RESPONSE  
pred\_vars <- dplyr::select(train\_data\_numeric, -SalePrice)  
# categorical\_dummy\_vars  
categorical\_vars <-  
 train\_data\_imputed %>%  
 select\_if(function(x) !is.numeric(x)) %>%  
 mutate(SalePrice = train\_data\_imputed$SalePrice)  
categorical\_dummy\_vars <-  
 model.matrix(SalePrice ~ ., data = categorical\_vars) %>%  
 data.frame() %>%  
 dplyr::select(-X.Intercept.)  
#squared variables  
squared\_vars <-  
 apply(pred\_vars, 2, function(x) x^2) %>%  
 as.data.frame()  
colnames(squared\_vars) <- paste0(names(squared\_vars), "\_2")  
#square root variables  
sqrt\_vars <-  
 apply(pred\_vars, 2, function(x) x^2) %>%  
 as.data.frame()  
colnames(sqrt\_vars) <- paste0(names(sqrt\_vars), "\_sqrt")  
#log variables  
log\_vars <-  
 apply(pred\_vars, 2, function(x) log(x + .01)) %>%  
 as.data.frame()  
colnames(log\_vars) <- paste0(names(log\_vars), "\_log")  
#combine all transformed variables  
individual\_vars <- cbind(categorical\_dummy\_vars,   
 squared\_vars,   
 sqrt\_vars,   
 log\_vars,   
 pred\_vars)   
# create interaction variables  
# https://stackoverflow.com/questions/2080774/  
# generating-interaction-variables-in-r-dataframes?  
# utm\_medium=organic&utm\_source=google\_rich\_qa&utm\_campaign=google\_rich\_qa  
if (!exists("all\_interactions")){  
 all\_interactions <- data.frame(t(apply(individual\_vars, 1, combn, 2, prod)))  
 colnames(all\_interactions) <- combn(names(individual\_vars), 2, paste, collapse=":")  
}  
# combine the individual variables and interactions  
all\_predictors <- cbind(individual\_vars, all\_interactions)  
# response variable transformations  
response\_transformed <-   
 train\_data\_numeric %>%   
 transmute(  
 SalePrice = SalePrice,  
 SalePrice\_2 = SalePrice^2,  
 SalePrice\_sqrt = sqrt(SalePrice),  
 SalePrice\_log = log(SalePrice)  
 )  
# create pairwise correlation df  
if (!exists("response\_correlations")){  
 response\_correlations <-   
 cor(response\_transformed, all\_predictors, use = "pairwise.complete.obs") %>%   
 correlation\_df() %>%   
 na.omit()  
}  
n\_rows <- 50  
kable(head(dplyr::filter(response\_correlations, Var1 == "SalePrice\_sqrt"), n\_rows),   
 digits = 3,   
 caption = "Table 3.5: Top Correlations with the Original Response Variable")  
# 1. Original Variables Imputed  
# divide into training & test  
train\_orig\_vars\_imputed <- full\_set\_imputed[1:nrow(train\_data), ]   
test\_orig\_vars\_imputed <-   
 full\_set\_imputed[nrow(train\_data) + 1:nrow(test\_data), ]  
# 2. Several Predictor Transformations, including  
# 7 categorical re-classifications & 5 interactions  
full\_set\_predictors\_transformed <-   
 full\_set\_imputed %>%   
 mutate(  
 RoofMatl\_WdShngl = as.integer(RoofMatl == "WdShngl"),  
 FireplaceQu\_Ex = as.integer(FireplaceQu == "Ex"),  
 HeatingQC\_Ex = as.integer(HeatingQC == "Ex"),  
 GarageQual\_abv\_avg = as.integer(GarageQual %in% c("TA", "Gd", "Ex")),  
 PoolQC\_Ex = as.integer(PoolQC == "Ex"),  
 Heating\_Gas = as.integer(Heating %in% c("GasA", "GasW")),  
 SaleCondition\_Partial = as.integer(SaleCondition == "Partial"),  
 OverallQual2\_x\_GarageCars = OverallQual^2 \* GarageCars,  
 OverallQual2\_x\_TotRmsAbvGrd\_log = OverallQual^2 \* log(TotRmsAbvGrd),  
 OverallQual2\_x\_GrLivArea = OverallQual^2 \* GrLivArea,  
 OverallQual2\_x\_LotArea\_log = OverallQual^2 \* log(LotArea),  
 OverallQual\_2 = OverallQual^2  
 ) %>%   
 dplyr::select(-c(RoofMatl, FireplaceQu, HeatingQC, GarageQual, PoolQC,   
 SaleCondition, Heating))  
#divide into training & test  
train\_predictors\_transformed <- full\_set\_predictors\_transformed[1:nrow(train\_data), ]   
test\_predictors\_transformed <-   
 full\_set\_predictors\_transformed[nrow(train\_data) + 1:nrow(test\_data), ]   
#3. Box-cox response transformation added to the existing predictor transformations  
lmod <- lm(SalePrice ~ ., data = train\_predictors\_transformed)  
n <- nrow(train\_predictors\_transformed)  
if (!exists("BIC\_lmod")) BIC\_lmod <- step(lmod, trace = 0, k = log(n))  
PT <- car::powerTransform(as.formula(BIC\_lmod$call), data = train\_predictors\_transformed)  
train\_BC\_transformed <-   
 train\_predictors\_transformed %>%   
 mutate(SalePrice\_BC = SalePrice^PT$lambda) %>%   
 dplyr::select(-SalePrice)  
  
# write.csv(train\_orig\_vars\_imputed, "train\_orig\_vars\_imputed.csv")  
# write.csv(train\_predictors\_transformed, "train\_predictors\_transformed.csv")  
# write.csv(train\_BC\_transformed, "train\_BC\_transformed.csv")  
# write.csv(test\_orig\_vars\_imputed, "test\_orig\_vars\_imputed.csv")  
# write.csv(test\_predictors\_transformed, "test\_predictors\_transformed.csv")  
# Read prepared data  
bcData = read.csv(paste0('https://raw.githubusercontent.com/omerozeren/DATA621/master/final\_project/train\_BC\_transformed.csv'))  
bcData$X = NULL  
imputedData = read.csv(paste0('https://raw.githubusercontent.com/omerozeren/DATA621/master/final\_project/train\_orig\_vars\_imputed.csv'))  
imputedData$X = NULL  
transformedData = read.csv(paste0('https://raw.githubusercontent.com/omerozeren/DATA621/master/final\_project/train\_predictors\_transformed.csv'))  
transformedData$X = NULL  
library(psych)  
describe(bcData)  
m1BC = lm(data=bcData,formula =SalePrice\_BC~. )  
m1IMP = lm(data = imputedData, formula = SalePrice~.)  
anova(m1IMP,m1TD)  
m1TD = lm(data=transformedData,formula = SalePrice~.)  
m2BCstep =step(m1BC,direction = 'backward', trace=0)  
summary(m2BCstep)  
m3BC = lm(data = bcData, formula = SalePrice\_BC~OverallCond+Condition2+Condition1+  
 Neighborhood+MSZoning +X1stFlrSF+X2ndFlrSF+LowQualFinSF+KitchenQual+  
 Fireplaces +ScreenPorch+House\_Age\_Yrs+RoofMatl\_WdShngl+  
 GarageQual\_abv\_avg +OverallQual2\_x\_GrLivArea+  
 OverallQual2\_x\_TotRmsAbvGrd\_log+OverallQual2\_x\_GarageCars)   
m4BC = lm(data = bcData, formula = SalePrice\_BC~OverallCond+Condition2+Condition1+  
 Neighborhood+MSZoning +X1stFlrSF+X2ndFlrSF+LowQualFinSF+KitchenQual+  
 Fireplaces+WoodDeckSF+Functional+FullBath+BsmtFullBath+BsmtFinType1+  
 BsmtExposure +BsmtQual +LandSlope +LandContour+LotArea +LotFrontage+   
 LotConfig + Utilities + HouseStyle + RoofStyle + MasVnrArea +  
 ScreenPorch+House\_Age\_Yrs+RoofMatl\_WdShngl+GarageQual\_abv\_avg +  
 OverallQual2\_x\_GrLivArea+OverallQual2\_x\_TotRmsAbvGrd\_log+  
 OverallQual2\_x\_GarageCars)   
m5imp = lm(data = imputedData, formula = log(SalePrice)~OverallCond+Condition2+  
 Condition1+Neighborhood+MSZoning +X1stFlrSF+X2ndFlrSF+LowQualFinSF+  
 KitchenQual+(Fireplaces)^2+WoodDeckSF+Functional+FullBath+  
 BsmtFullBath+BsmtFinType1 + BsmtExposure +BsmtQual +LandSlope +  
 LandContour+log(LotArea) + LotFrontage+ LotConfig + Utilities +   
 HouseStyle + RoofStyle + MasVnrArea +ScreenPorch+House\_Age\_Yrs)   
m6TD = lm(log(SalePrice)~OverallCond+Condition2+  
 Condition1+Neighborhood+MSZoning +X1stFlrSF+X2ndFlrSF+LowQualFinSF+  
 KitchenQual+Fireplaces+WoodDeckSF+Functional+FullBath+BsmtFullBath+  
 BsmtFinType1 + BsmtExposure +BsmtQual +LandSlope +LandContour+  
 log(LotArea) + LotFrontage+ LotConfig + Utilities + HouseStyle +   
 RoofStyle + MasVnrArea +ScreenPorch+House\_Age\_Yrs+RoofMatl\_WdShngl+  
 GarageQual\_abv\_avg +OverallQual2\_x\_GrLivArea+  
 OverallQual2\_x\_TotRmsAbvGrd\_log+OverallQual2\_x\_GarageCars,  
 data = transformedData)   
# Get AIC  
AIC (m1BC, m2BCstep, m3BC, m4BC, m5imp, m6TD)  
summary(m1BC)  
summary(m2BCstep)  
summary(m3BC)  
summary(m4BC)  
summary(m5imp)  
summary(m6TD)  
# Read test data  
transformedTest = read.csv(paste0('https://raw.githubusercontent.com/omerozeren/DATA621/master/final\_project/test\_predictors\_transformed.csv'))  
index <- transformedTest$X  
transformedTest$X <- NULL  
# Tune model and run prediction  
library(caret)  
ctrl <- trainControl(method = "repeatedcv", number = 10, savePredictions = TRUE)  
model\_fit <- train(log(SalePrice)~OverallCond+Condition2+  
 Condition1+Neighborhood+MSZoning +X1stFlrSF+X2ndFlrSF+LowQualFinSF+  
 KitchenQual+Fireplaces+WoodDeckSF+Functional+FullBath+BsmtFullBath+  
 BsmtFinType1 + BsmtExposure +BsmtQual +LandSlope +LandContour+  
 log(LotArea) + LotFrontage+ LotConfig + Utilities + HouseStyle +   
 RoofStyle + MasVnrArea +ScreenPorch+House\_Age\_Yrs+RoofMatl\_WdShngl+  
 GarageQual\_abv\_avg +OverallQual2\_x\_GrLivArea+  
 OverallQual2\_x\_TotRmsAbvGrd\_log+OverallQual2\_x\_GarageCars,   
 data=transformedData, method="lm", trControl = ctrl, tuneLength = 5)  
pred <- predict(model\_fit, newdata=transformedTest)  
results <- cbind(index, exp(pred))  
write.csv(results, "c://temp//results\_tune.csv", row.names = FALSE)  
summary(model\_fit)  
library(ggplot2)  
library(ggfortify)  
autoplot(m6TD2)  
library(car)  
vif(m6TD2)  
alias(m6TD2)  
transformedData2 <- transformedData[-c(826,524,1299,89),]  
m6TD2 = lm(log(SalePrice)~OverallCond+ # Condition2+  
 Condition1+  
 MSZoning +X1stFlrSF+X2ndFlrSF+LowQualFinSF+Neighborhood+  
 KitchenQual+Fireplaces+WoodDeckSF+Functional+FullBath+BsmtFullBath+  
 BsmtFinType1 + BsmtExposure +BsmtQual +  
 LandSlope +LandContour+  
 log(LotArea) + LotFrontage+ LotConfig + HouseStyle + #Utilities +   
 RoofStyle + MasVnrArea +ScreenPorch+House\_Age\_Yrs + RoofMatl\_WdShngl+  
 GarageQual\_abv\_avg +OverallQual2\_x\_GrLivArea+  
 OverallQual2\_x\_TotRmsAbvGrd\_log+OverallQual2\_x\_GarageCars,  
 data = transformedData2)   
summary(m6TD2)  
pred <- predict(m6TD2, newdata=transformedTest)  
results <- cbind(index, exp(pred))  
write.csv(results, "c://temp//results\_m5TD2.csv", row.names = FALSE)  
AIC(m6TD2)  
table(transformedData2$OverallCond)  
table(transformedData2$Condition1)  
table(transformedData2$Condition2) # Removed  
table(transformedData2$MSZoning)  
table(transformedData2$Neighborhood)  
table(transformedData2$KitchenQual)  
table(transformedData2$Fireplaces)  
table(transformedData2$WoodDeckSF)  
table(transformedData2$FullBath)  
table(transformedData2$BsmtFullBath)  
table(transformedData2$BsmtFinType1)  
table(transformedData2$BsmtExposure)  
table(transformedData2$BsmtQual)  
table(transformedData2$LandSlope)  
table(transformedData2$LandContour)  
table(transformedData2$LotConfig)  
table(transformedData2$Utilities) # Removed  
table(transformedData2$HouseStyle)  
table(transformedData2$RoofStyle)  
table(transformedData2$RoofMatl\_WdShngl)  
table(transformedData2$GarageQual\_abv\_avg)  
table(transformedData2$MasVnrArea)  
ce <- as.data.frame(m6TD2$coefficients)  
colnames(ce) <- c("Coefficient")  
write.csv(round(ce, 6), "c://temp//embedded\_table2\_coef.csv", row.names = TRUE)