Determining Factors Influencing House Sale Prices

Kyle Gilde, Jaan Bernberg, Kai Lukowiak, Michael Muller, Ilya Kats
DATA 621, Master of Science in Data Science, City University of New York

Abstract

Buying or selling a house is one of the important events faced by many in their lifetime. It is also a big part of our economy. As such understanding what factors drive sale prices is both interesting and beneficial. In this project we use popular Ames, Iowa housing data set to investigate what factors may influence the final sale price the most. Our 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 neighrborhood. It definitely doesn't pay to have the biggest house on the block.

Key Words

house prices, regression, linear models, assessed value

Introduction

This project stems out of the Business Analytics and Data Mining class in the Master of Science in Data Science program at CUNY. This paper is the result of the final class group project in applying regression methods to real-world data. Our team chose housing data because it promissed to be an interesting and useful subject. In addition, this research is based on a well studied data set which makes it an excellent educational resource allowing our team to study various approaches.

The data set was prepared by Dean De Cock in an effort to create a real-world data set to test and practice regression methods (De Cock 2011). It describes the sale of individual residential property in Ames, Iowa from 2006 to 2010. Ames, Iowa was founded in 1864 as a station stop. It has the population of about 60,000 people and covers about 24.27 sq mi. It was ranked ninth on the *Best Places to Live* list (CNNMoney 2010).

The data came directly from the Assessor's Office in the form of a data dump from their records system and it included information for calculation of assessed values in the city's assessment process. The data is recent and it covers the period of housing bubble collapse that led to the subprime mortgage crisis. 2008 saw one of the largest housing price drops in history.

Each of over 2,900 total observations in the data represent attributes of a residental property sold. For properties that exchanged ownership multiple times during the collection period (2006 through 2010), only the last sale is included in the data since it represents the most current value of the property. The attributes that make up the sale price of a house can seem daunting given a myriad of factors that can impact its value. There are about 80 variables included in the data set. Most variables describe physical attributes of the property. There is a variety of variable types - discrete, continous, categorical (both nominal and ordinal).

The data was originally published in the Journal of Statistics Education (Volume 19, Number 3). Data set was downloaded from Kaggle.com which gave us the ability to compare our results with results of other teams working with this data set (Kaggle 2016).

Literature Review

Building regression models to predict house prices is not a new undertaking. Quite the opposite, a lot of research went into this area. There is a clear financial benefit to buyers, sellers and other parties in knowing which attributes influence final sale price. There is also a lot of data readily available with some cleanup work. Data is kept by local governments to be used in the assessment process for property taxes. There is a lot of data captured by realors when a property is listed on the market. Additionally, in large part thanks to information revolution, data is easily accessible via many aggregators such as MLS.

There are many attributes that factor into a house price. For example, environmental attributes can impact the price substantially. A garden facing water, a pleasant view whether it overlooks water or open space, attractive landscaping all increase house prices (Luttik 2000). Neighborhood attributes such as schools and public services also play a factor.

Our data set deals mostly with physical characteristics of the house itself. Even here there is a lot of room for variation. For example, one study counted half-bathrooms as 0.1 out of belief that buyers do not value them as much as full bathrooms (Pardoe 2008).

One key characteristic that should be taken into account is the correlation between the prices of neighboring houses (Dubin 1998). This will not be addressed by our project because of complexity and because we do not have specific locations for our data set.

Another interesting approach to modeling house prices is hedonic regression. It 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. It is possible to study just one specific area, such as school characteristics (Downes 2002), and its impact on house prices.

A few examples we state here is just scratching the surface of this particular topic. In addition to determining factors that go into house sale prices, our goal is to apply a methodical approach to linear regression. As such we have decided to study the data set without referencing specific works. We intentionally avoided discussions on Kaggle, the source of our data set, so that we can treat this research as a learning exercise as much as it is a scientific inquiry.

Methodology

Data Description

The data set includes 2,910 observation and 79 indepedent 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, many `NA` values found in the data carry useable information. 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 imply that the property has no garage. This type of missing values 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 went up significantly to 2,861 or about 98% of all observations. There remained only 58 observations with true missing values (about 2% of the total observations). 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 we should drop them from analysis.

Consider the pattern to the missing values. In addition to the quantity of missingness being important, why and how the values are missing can give us insight into whether we have a biased sample. There are three types of missing data (Faraway 2014): 1) Missing Completely at Random (MCAR), 2) Missing at Random (MAR), and 3) Missing Not at Random (MNAR). MCAR is when the probability of missingness is the same for all cases. This is the ideal type of missingness because we could delete these cases without incurring bias. MAR occurs when the probability of a value being missing depends upon a known mechanism. In this scenario, we could delete these observations and compensate by weighting by group membership. Finally, MNAR occurs when the values are missing because of an unknown variable. This is the type of missingness that is most likely to bias our sample. Faraway asserts that ascertaining the exact nature of the missingness is not possible and must be inferred. Figure 2 displays the combinations of missing values in the predictor variables. We may not have MCAR because we can see that the missingness is not more dispersed across all variables and cases. Only 32 of the 79 predictors have a missing value, and we notice that the missingness occurs most often in some of the masonry, basement and garage variables. There is no indication that values are missing not at random and given the small number of missing values, we believe the bias, if any, will be limited.

There are four ways to deal with missing values (Prabhakaran 2017):

- **Deleting the cases:** This is not a preferred method because one could introduce bias or the model could lose power from being based upon fewer cases.
- **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 we 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.

For our data set we 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 (Azur 2012). 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. We 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. If we were to run, multiple imputations, hopefully we would begin to see more convergence between the imputed and observed values.

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 modeling output more readable and easier to interpret.

As is the case with most data sets, we 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, we have extensively analyzed available variables and took a few approaches to variable transformations. They were ment 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. Rather than trying to work with original variables we have converted them to a single *age* variable. For house age the value was $YrSold - YearBuilt$. Similarly the age of garage and remodeling was added to the data set. Original variables were dropped from analysis.

Because we are not dealing with a time series data set, we have converted `YrSold` and `MoSold` variables from numeric to nominal. It is important to catch seasonality, but does not make sense to regress on these variables as continous variables.

Using the side-by-side box plots in Figure 4, we examined the categorical variables with more than two values to see if the variable can be simplified by combining the values into two groups. Our criteria for this simplification is 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), we can notice that only 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`). Consequently, we transformed these into dummy variables with appropriate names. This allowed us to preserve some degrees of freedom that would otherwise be subtracted if each and every one of the original values were turned into dummy variables.

We examined whether our modeling would benefit from transforming any of the predictor variables. To do so, we have automated creation of several different versions of the predictor variables using `R`. We took natural logarithms, square roots and squares of the numerical variables, and then we calculated every possible pairwise interaction between these transformations, the original numerical variables and categorical variables. We 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 descendingly by R-squared. We observed that there are several correlation values higher than the highest correlation between the original predictor and the response, which is `OverallQual` at 0.79 (*see table 5*). Most promising transformations involved taking the square of `OverallQual` and multiplying it by the log-transformed or square-root-transformed one of the area variables. We added top five interactions to our training data set.

We have created several potential training sets to give use flexibility in training the model. The **first** of the three training data sets we 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 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, we 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, we created a simple BIC step model and used it to calculate the Box-Cox λ value and transform the response variable. According to Box-Cox, a λ 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 we are dealing with trying to predict a continous variable, house sale price, we relied on 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 R^2 values were within some neglible deviation of each other, adding a Box-Cox transformation of the response variable improved the R^2 beyond the model based on the original data set.

We took the strongest model, and applied stepwise regression. Since we started with the baseline model containing all variables we applied backward elimination in order to settle on a model with the lowest Akaike information criterion (AIC) value.

For non-transformed response variable, we expetimented with applying log-transformation as it tends to bring sales data closer to normal distribution.

We 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, but we can remove some insignificant variables for a more parsimonious model and lower chances of overfitting.
2. **Model 2** is based on Model 1 modified with stepwise regression (backward elimination). It is an improvement with lower number of parameters (156 comparing to 237). The multiple R^2 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). R^2 drops and F-statistic rises, so even though the model is simplier 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). Number of parameters is increased, but R^2 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 R^2 for any model we 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 we can infer that these models are statistically significant.

The table below summarizes the models. We can see a steady improvement in AIC numbers for models 1 through 6. Adjusted R^2 fluctuates, but it remains high and the values are close between various models. Fluctuation in R^2 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

Since the data set comes from Kaggle, we have an easy way to validate test these models by predicting the sale price for the testing set and submitting our predictions. Kaggle provides a score that lets you judge the performance of our models. We submitted predictions from models 5 and 6. Model 6 was a clear favorite.

Models 1 through 4 rely on the Box-Cox transformation of the response variable with λ value of 0.184.

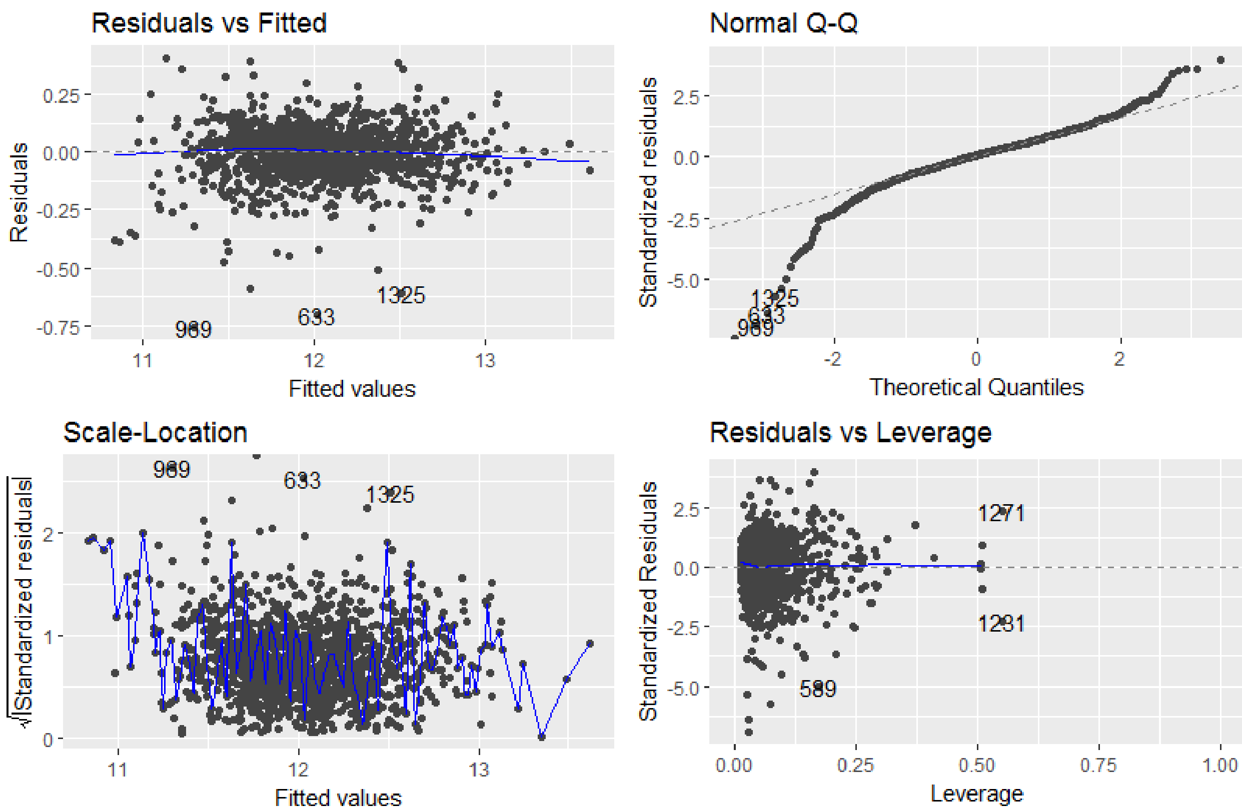

Figure 1: Model diagnostics.

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 signficantly 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 plausable 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 response variable, but there is not enough information in our small data set in these variable to train on.

Final model had multiple R^2 of 0.9276, adjusted R^2 of 0.9225, AIC of -2172 and Kaggle score of 0.13376. These are the best values in all categories.

Figure 1 shows typical model diagnostic. There is no dicernible 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.

Experimentation and Results

Having developed a model and successfully tested it against avaible test data, we can now review the results.

Our model has 98 parameters. 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:

$$\hat{y} = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p + \epsilon$$

Here, $\hat{y}$ is the predicted outcome, x_1 through x_p are selected parameters, β_1 through β_p are corresponding coefficients, β_0 is the intercept and ϵ is the error or noise term. We have too many parameters to efficiently write out our linear model; however, using the coefficient table we can easily assemble the final formula if desired. The formula used in `R` is as follows:

```r
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
```

Table 2: 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

Category	Details	Coefficient
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

Category	Details	Coefficient
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

Attribute with the most negative influence 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 import to the final sale price.

Other attribute that influences the sale price negatively is the Meadow Village neighborhood. Reviewing Ames, Iowa information, we 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 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 attribues.

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:

- House age has a negative influence, but not as great as some other attributes.
- Cul-de-sac is the most positive lot location. Inside lot has a slightly negative influence. A lot with streets on 2 sides has even more negative influence. Finally, a lot with streets on 3 sides has the biggest negative impact.
- Unfinished basement is generall a negative impact. As is low quality kitchen.
- 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, Iowa 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 our 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.

Our 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 many opportunity for research, and it would be beneficial to try other methods for comparison purposes. If moving to other similar data sets, then increasing complexity by trying to account for other factors should produce the most benefit.

Appendix A. Figures

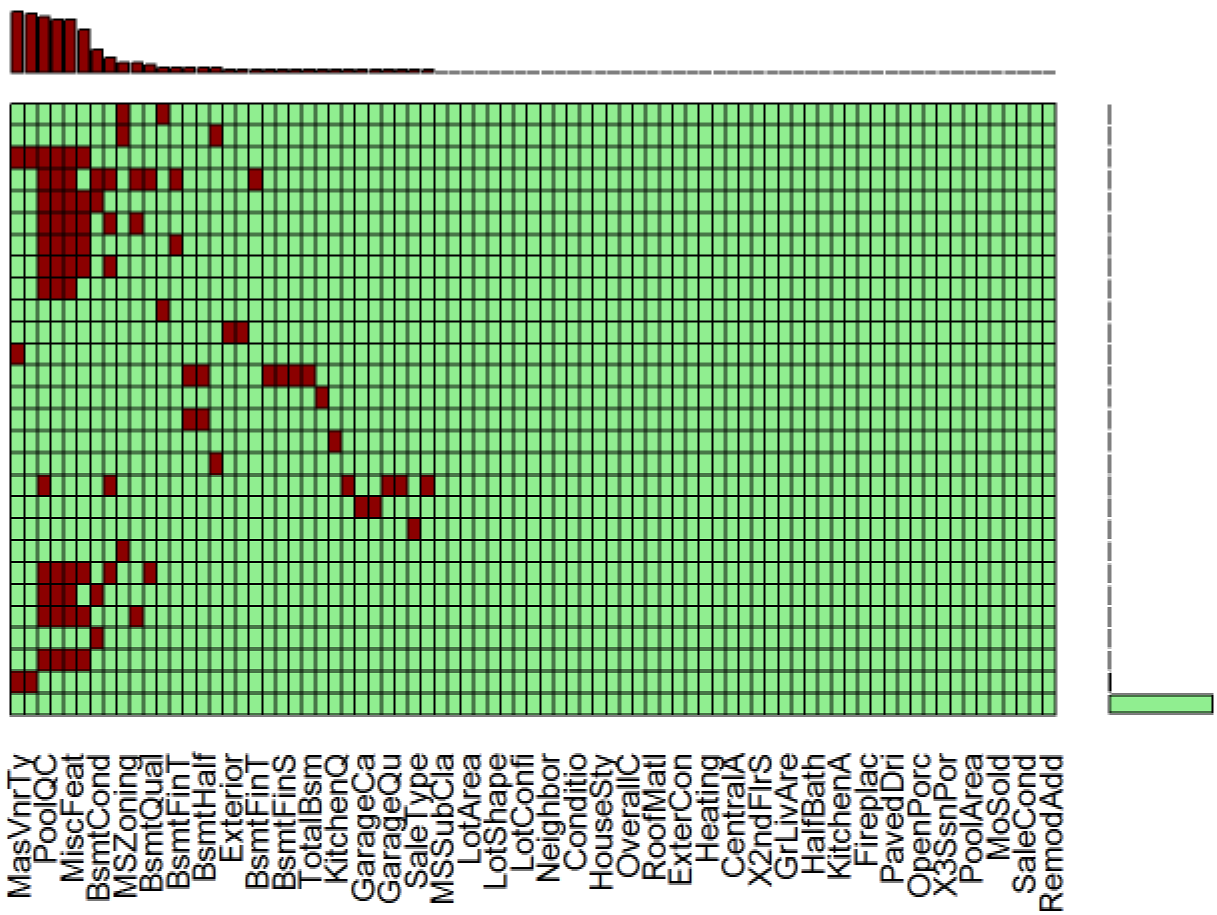

Figure 2. Missing values.

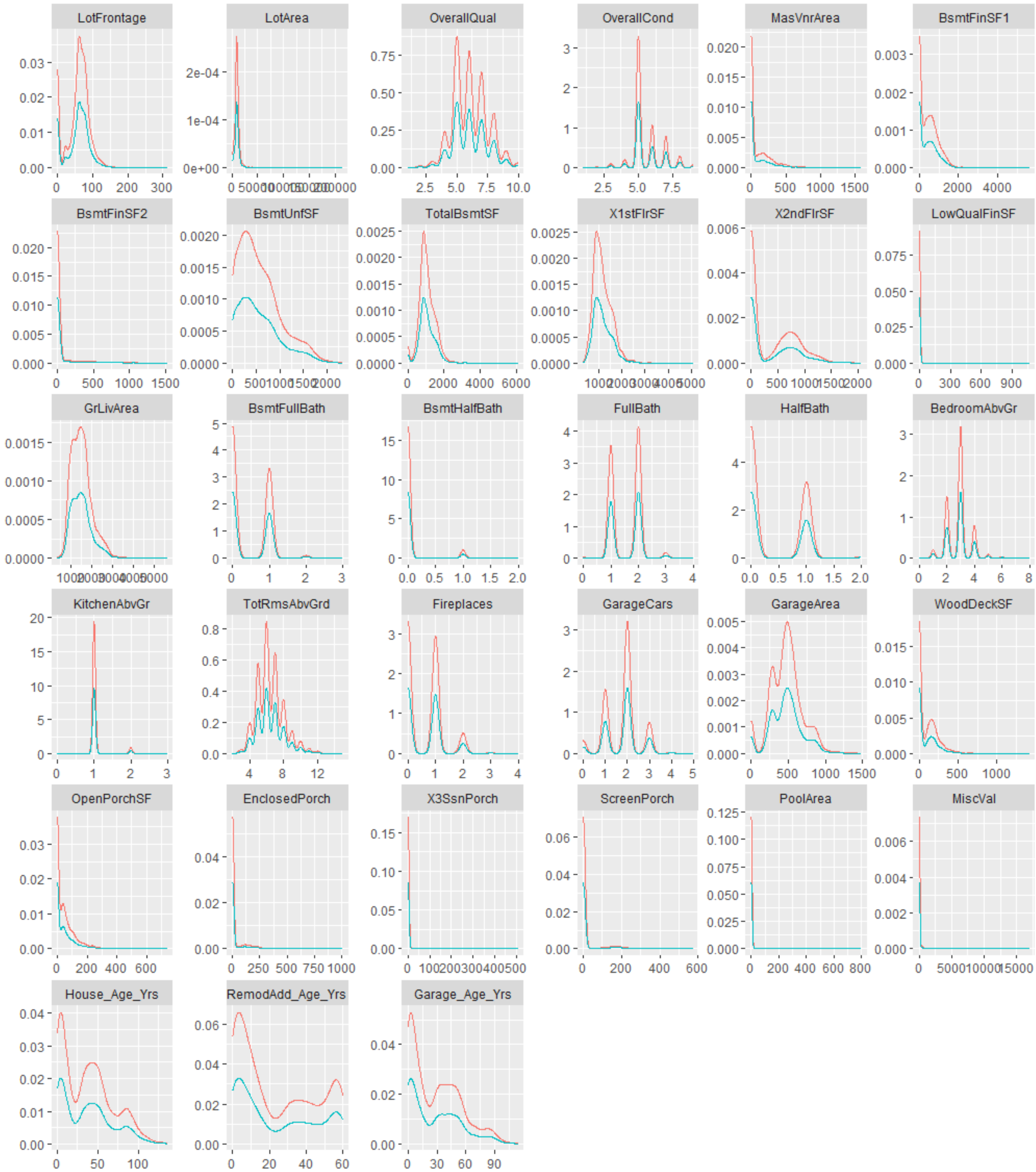

Figure 3. Density plots of observed (blue) and imputed (red) values.

Figure 4. Box plots of categorical variables against the response variable.

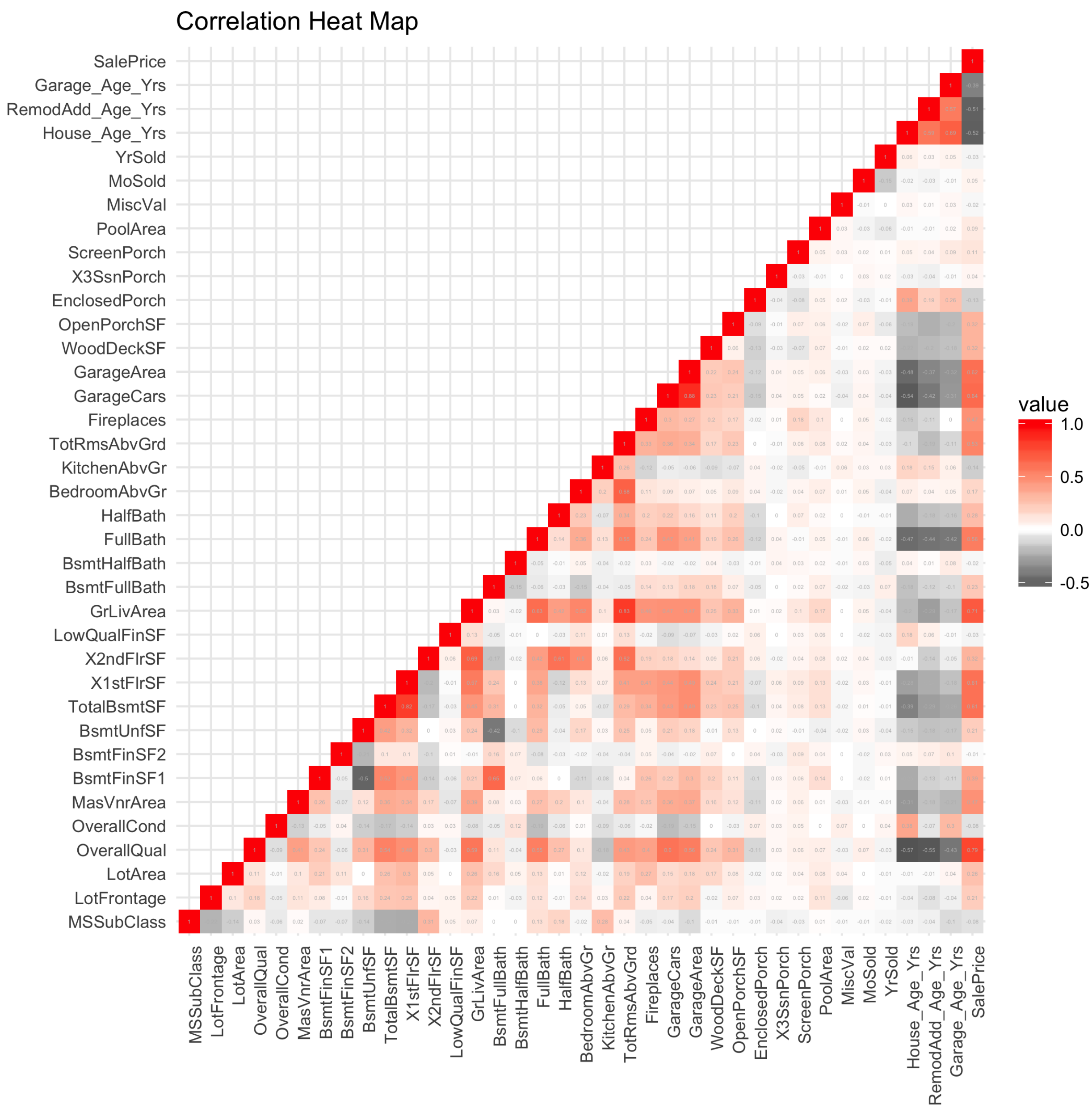

Figure 5. Correlation heat map.

Appendix B. Tables

Table 3: 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

Table 4: 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

Table 5: 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

Table 6: 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

Response Variable	Predictor Transformation	Correlation	R^2
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){
  # Load packages & Install them if needed.
  # CODE SOURCE: https://gist.github.com/stevenworthington/3178163
  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/kaiserxc/DATA621FinalProject/",
                    "master/house-prices-advanced-regression-techniques/train.csv")
url_test <-  paste0("https://raw.githubusercontent.com/kaiserxc/DATA621FinalProject/",
                    "master/house-prices-advanced-regression-techniques/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
```

```r
  # 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))
```

```r
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")        # needs repair
Utilities_ <- c("AllPub", "NoSeWa")               # ok - No "NoSewr", "ELO"
LandSlope_ <- c("Gtl","Mod", "Sev")               # ok
ExterQual_ <- c("Ex", "Gd", "TA", "Fa")           # needs repair - No "Po"

ExterCond_ <- c("Ex", "Gd", "TA", "Fa", "Po")     # needs repair
BsmtQual_ <- c("Ex", "Gd", "TA", "Fa", "NoB")     # needs repair
BsmtCond_ <- c("Gd", "TA", "Fa", "NoB")           # needs repair
BsmtExposure_ <- c("Gd", "Av", "Mn", "No", "NoB") # needs repair

BsmtFinType1_ <- c("GLQ", "ALQ", "BLQ",
                   "Rec", "LwQ", "Unf", "NoB")    # needs repair
BsmtFinType2_ <- c("GLQ", "ALQ", "BLQ",
                   "Rec", "LwQ", "Unf", "NoB")    # needs repair
HeatingQC_ <- c("Ex", "Gd", "TA", "Fa", "Po")     # needs repair
Electrical_ <- c("SBrkr", "FuseA", "FuseF",
                 "FuseP", "Mix")                  # needs repair

KitchenQual_ <- c("Ex", "Gd", "TA", "Fa")         # needs repair - no "Po"
Functional_ <- c("Typ", "Min1", "Min2", "Mod",
                 "Maj1", "Maj2", "Sev")           # needs repair - no "Sal"
FireplaceQu_ <- c("Ex", "Gd", "TA", "Fa",
                  "Po", "NoFp")                   # needs repair
GarageFinish_ <- c("Fin", "RFn", "Unf", "NoG")    # ok

GarageQual_ <- c("Ex", "Gd", "TA", "Fa", "Po",
                 "NoG")                           # needs repair
GarageCond_ <- c("Ex", "Gd", "TA", "Fa", "Po",
                 "NoG")                           # needs repair
```

```r
PavedDrive_ <- c("Y", "P", "N")                         # needs repair
PoolQC_ <- c("Ex", "Gd", "Fa", "NoP")                   # needs repair - no "TA"
Fence_ <- c("GdPrv", "MnPrv", "GdWo", "MnWw",
            "NoF")                                      # needs repair

# 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
```

```r
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")
```

```r
#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), ]
```

```r
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
```

```r
# 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")
```

```r
#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")
```

```r
# 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)

# setwd("C:\\Users\\kyleg\\DATA621FinalProject\\data-imputed-transformed\\")
# 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")
```

```r
# write.csv(test_predictors_transformed, "test_predictors_transformed.csv")


# Read prepared data
bcData = read.csv(paste0('https://raw.githubusercontent.com/kaiserxc/',
                         'DATA621FinalProject/',
                         'master/data-imputed-transformed/train_BC_transformed.csv'))
bcData$X = NULL
imputedData = read.csv(paste0('https://raw.githubusercontent.com/kaiserxc/',
                              'DATA621FinalProject/master/data-imputed-transformed/',
                              'train_orig_vars_imputed.csv'))
imputedData$X = NULL
transformedData = read.csv(paste0('https://raw.githubusercontent.com/kaiserxc/',
                                  'DATA621FinalProject/master/data-imputed-transformed/',
                                  '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,
```

```r
          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/kaiserxc/',
                                  'DATA621FinalProject/master/data-imputed-transformed/',
                                  '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)
```

References

Azur, Stuart, M. 2012. “Multiple Imputation by Chained Equations: What Is It and How Does It Work?” March. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3074241/.

Chang, W., J. Cheng, JJ. Allaire, Y. Xie, and J. McPherson. 2015. “Shiny: Web Application Framework for R. R Package Version 0.12.1.” Computer Program. http://CRAN.R-project.org/package=shiny.

CNNMoney. 2010. “Best Places to Live.” http://money.cnn.com/magazines/moneymag/bplive/2010/snapshots/PL1901855.html.

De Cock, D. 2011. “Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project.” *Journal of Statistics Education, Vol. 19, No. 3.* https://ww2.amstat.org/publications/jse/v19n3/decock.pdf.

Downes, Zabel, T. 2002. “The Impact of School Characteristics on House Prices: Chicago 1987-1991.” *Journal of Urban Economics, Vol. 52, Issue 1*, July. https://www.sciencedirect.com/science/article/pii/S0094119002000104.

Dubin, R. 1998. “Predicting House Prices Using Multiple Listings Data.” *The Journal of Real Estate Finance and Economics, Vol. 17, Issue 1*, July. https://link.springer.com/article/10.1023/A:1007751112669.

Faraway, J. 2014. *Linear Models with R.* 2nd Edition. New York, NY: Chapman; Hall/CRC.

Kaggle. 2016. “House Prices: Advanced Regression Techniques,” August. https://www.kaggle.com/c/house-prices-advanced-regression-techniques.

Luttik, J. 2000. “The Value of Trees, Water and Open Space as Reflected by House Prices in the Netherlands.” *Landscape and Urban Planning, Vol. 48, Issues 3-4*, May. https://ww2.amstat.org/publications/jse/v16n2/datasets.pardoe.html.

Pardoe, I. 2008. “Modeling Home Prices Using Realtor Data.” *Journal of Statistics Education, Vol. 16, No. 2.* https://ww2.amstat.org/publications/jse/v16n2/datasets.pardoe.html.

Prabhakaran, S. 2017. “Missing Value Treatment,” April. https://datascienceplus.com/missing-value-treatment/.

R Core Team. 2015. “R: A Language and Environment for Statistical Computing.” Journal Article. http://www.R-project.org.

Wickham, H. 2009. *Ggplot2: Elegant Graphics for Data Analysis (Use R!).* New York, NY. http://ggplot2.org.