**Predicting Future House Prices**

**Advanced Linear Regression and Feature Engineering Techniques**



**SYST 468-001**

**Group:**

**Anderson Barreal, Murat Goturk, O’Ryan Lattin**

Table of Contents

[Project Introduction 2](#_Toc40192703)

[Project Description 2](#_Toc40192704)

[Preliminary Data Analysis 2](#_Toc40192705)

[Preprocessing 2](#_Toc40192706)

[Categorical Preprocessing 2](#_Toc40192707)

[Numerical Preprocessing 3](#_Toc40192708)

[Data Partitioning 5](#_Toc40192709)

[Regression Models 5](#_Toc40192710)

[Linear Regression Models 5](#_Toc40192711)

[Penalized Regression Models 7](#_Toc40192712)

[Nonlinear and Tree Regression 8](#_Toc40192713)

[Discussion 10](#_Toc40192714)

[Conclusions 11](#_Toc40192715)

[Problems Encountered 11](#_Toc40192716)

[Conclusion 12](#_Toc40192717)

[How to Run Code 12](#_Toc40192718)

[References 13](#_Toc40192719)

[Appendices 14](#_Toc40192720)

[Appendix A: Data Visualization 14](#_Toc40192721)

[Appendix B: Regression Graphs 15](#_Toc40192722)

[Appendix C: Kaggle Results 19](#_Toc40192723)

# **Project Introduction**

## Project Description

Kaggle’s, “House Prices: Advanced Regression Techniques”, challenges data scientists to use regression techniques to predict the final price of a home in Aimes, Iowa given 80 variables that describe the many characteristics of a home. This is the official description as seen on Kaggle, “Ask a home buyer to describe their dream house, and they probably won't begin with the height of the basement ceiling or the proximity to an east-west railroad. But this playground competition's dataset proves that much more influences price negotiations than the number of bedrooms or a white-picket fence. With 80 explanatory variables describing (almost) every aspect of residential homes in Ames, Iowa, this competition challenges you to predict the final price of each home.” The goals of this project are to use preprocessing and modeling techniques learned in class to tackle this problem. The goal of this report is to go through all steps of how data was analyzed, preprocessed, and modeled before entering the Kaggle competition for predicting house prices from a test set provided by Kaggle.

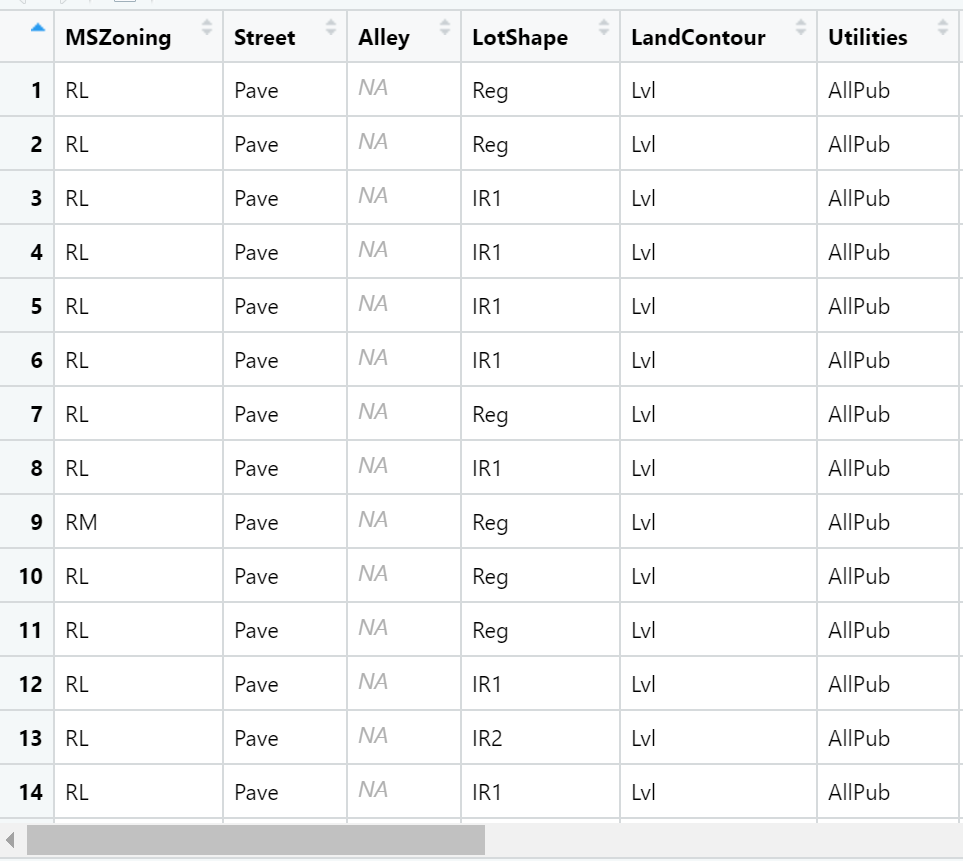
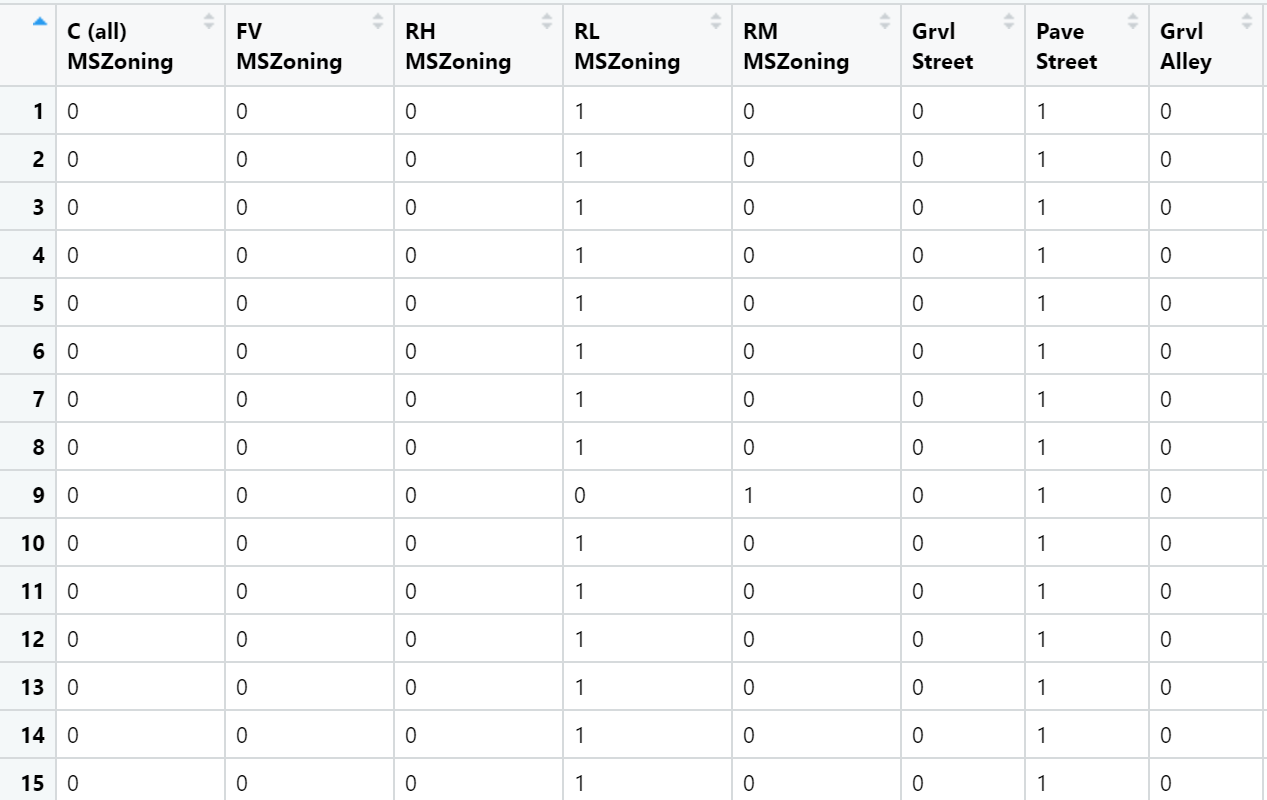
## Preliminary Data Analysis

Two datasets were provided on Kaggle, a training set and a test set. Both sets contained 79 predictor variables describing aspects of homes in Ames, Iowa and the training set contained an 80th variable containing the sales price of the house for training the models. The dataset predictor variables consist of two data types, numerical and categorical. The categorical data had a lot of nominal variables, such as street type( paved, gravel, etc.) and whether there is a central air conditioning unit (Y, N). There were also ordinal categorical variables, such as overall house quality (1-10) and overall house condition (1-10). There were a lot of different numerical variables, consisting of things like square footage measurements, lot areas, room counts, and bathroom counts. Examples from the dataset are basement square foot, total full baths, and lot area. The sales price variable, the response to be predicted, consisted of the sales price of the home based on the predictors. From looking at the predictor variables available to us, it was decided that the categorical and numerical data be split into two separate datasets for separate preprocessing techniques, then be later combined back into one set for creating predictive models.

# **Preprocessing**

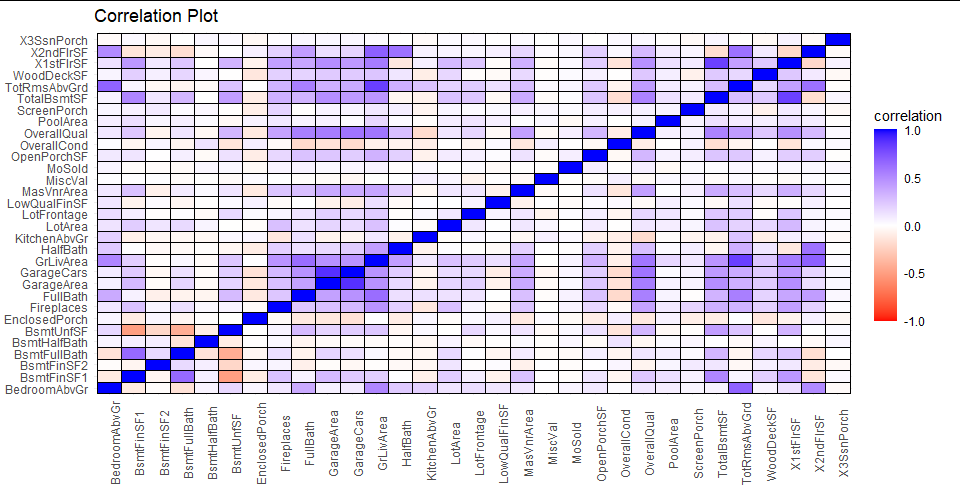
## Categorical Preprocessing

The training and test datasets provided by Kaggle were split into two separate datasets for preprocessing, categorical and numerical. This was done by first separating the two sets by variable data type, where all variables casted as factor type were put in the category set. Then, numerical categories not initially casted as a factor had to manually be removed from the numerical set and placed in the category set. This included variables for years (year built, etc.) and zoning districts ranked by numerical classification. Once the categories were all in the set, there were a total of 48 categorical predictors. To address concerns of overfitting, ANOVA was performed on each categorical variable to remove variables with little variance between levels of each category. This was done in a created R function that made use of the aov function provided in R. A low significance level of .01 was used because many of the categories contained very high variance between the different levels. This reduced the number of categorical variables to 33 predictors. After this, each level of each variable was converted into its own column, creating a column for every category level in each variable. This was done using a created R function that iterated through each column and assigned a new column to each level of that column in a new dataset. Then, if that level occurred in a sample row, a value was assigned a 1 to that column, and if not a 0. This allowed us to capture every level in the training set so that there would not be factors in the test set that were not in the training set. This resulted in a transformed categorical dataset of 277 predictor variables, all either a 0 or 1 in value indicating on whether it occurred or not.

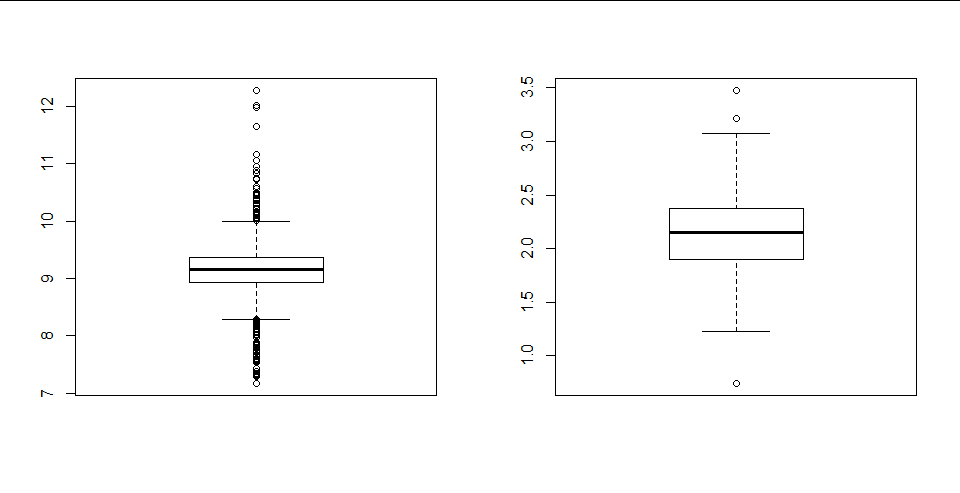
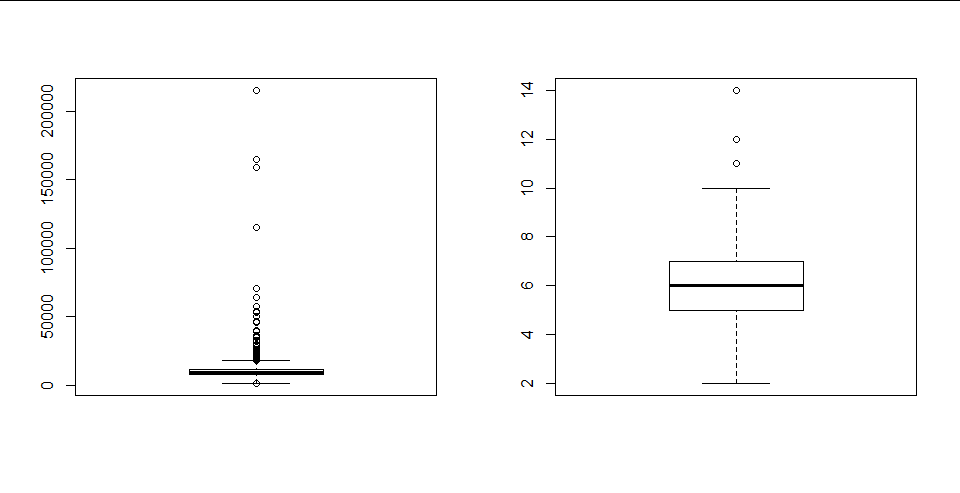
 

## Numerical Preprocessing

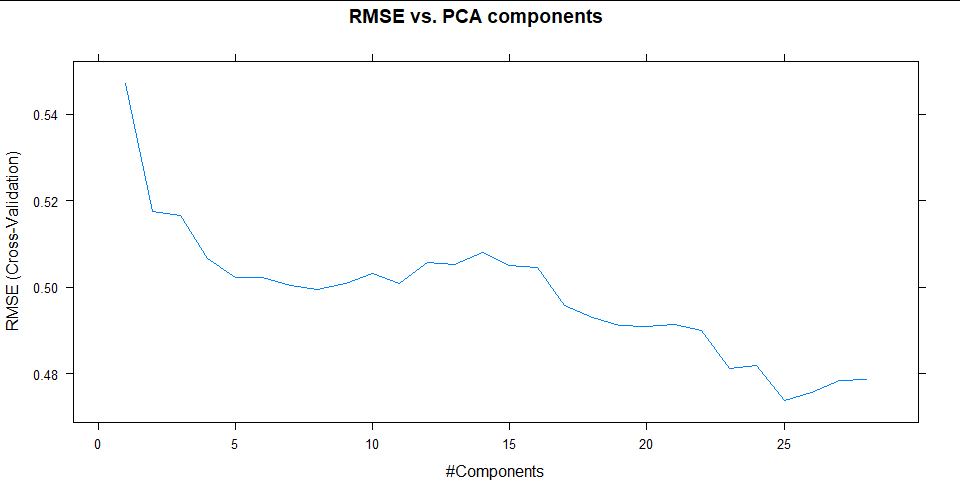
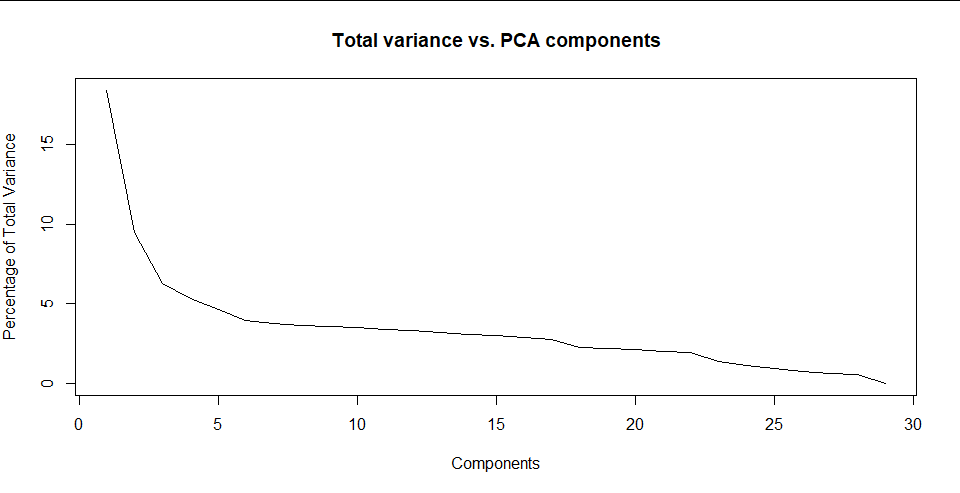
There were 31 numerical predictor variables in the dataset, separated from the training and test set by selecting variables that were of numeric data type. The first preprocessing step was testing for correlation between each of the numeric predictors, and to remove correlated predictors above a threshold level of 80 percent. There was not a lot of correlation between the predictors, and only three predictors were removed due to high correlation. Below is a plot generated in R of correlation between all the predictors.



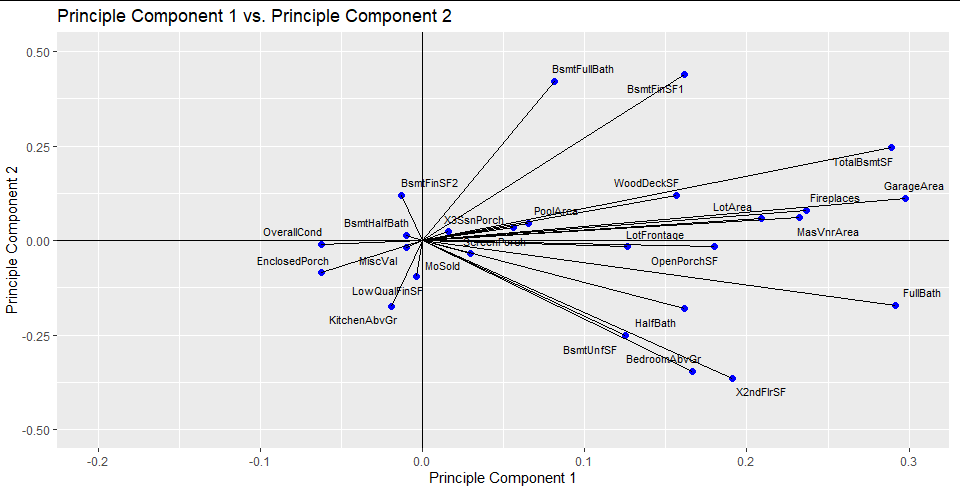
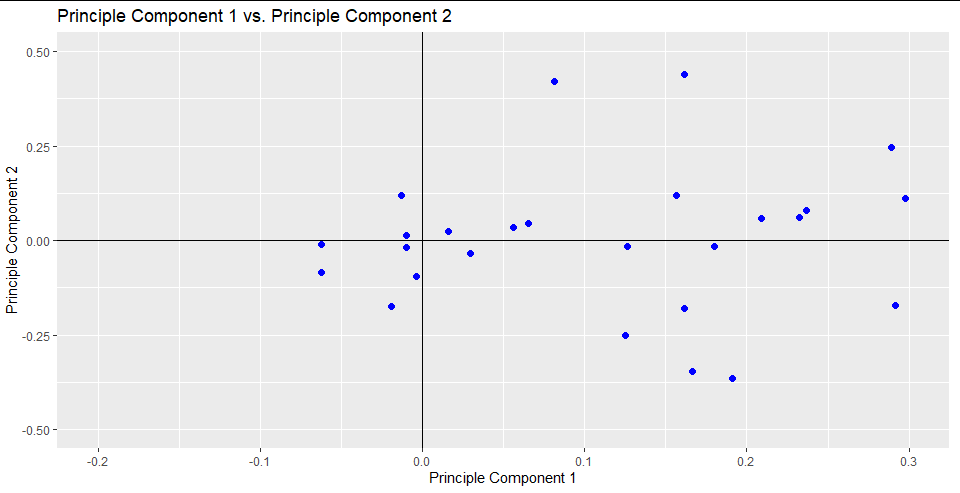
After removing correlated predictors, the numerical predictors were tested for skewness. There was found to be skewness in many of the predictors. To address this, BoxCox transformation was applied to the dataset to remove the skewness. Below is a before and after Box and Whisker of two predictors showing how BoxCox transformation removed skewness.



After skewness was addressed, PCA transformation was done in efforts to reduce the dimension of the numerical dataset while also capturing most of the variability in the predictors. It was also the intention to perform more preliminary data analysis on the transformed dataset to search for any trends in the principal components. The function prcomp was used to perform PCA transformation on the predictors. Looking at a plot of the percent of total variance captured by each principal component, roughly 25 components effectively captures most of the total variance from all the predictors.



The train function was used to tune the model for finding the best number of components to use in building a regression model. After tuning across 28 predictors, the optimal number of predictors chosen was 25 predictors. The graph on the right shows that 25 principal components reduces the RMSE value to an acceptable minimum value. Both graphs show that roughly 3-5 predictors capture a lot of variance from the predictors and reduce RMSE to a low level. Because of this, looking at a plot of principal component one vs. principal component two was conducted to provide insight for which predictors from the original dataset contribute most to the principal components that capture most of the variance. GGplot was used for generating both scatterplots produced below. The plot on the right includes labels for viewing each predictor that contributes most to each principal component.



From the results of the plot, the first PCA component is composed primarily of areas, square footage, and bathroom count measurements with the one exception being fireplace count. This makes sense however as bigger and more expensive houses probably have more fireplaces than smaller cheaper houses. From this it was hypothesized that regarding the numerical predictors, predictors measuring square footage, area, bathroom count, and fireplace count can be used to provide a reasonable estimate of the sale price of the home. It was also assumed that these predictors would show up later as important variables when building the different models for predicting home price. After performing PCA analysis, all preprocessed numerical predictors were centered and scaled using the scale function.

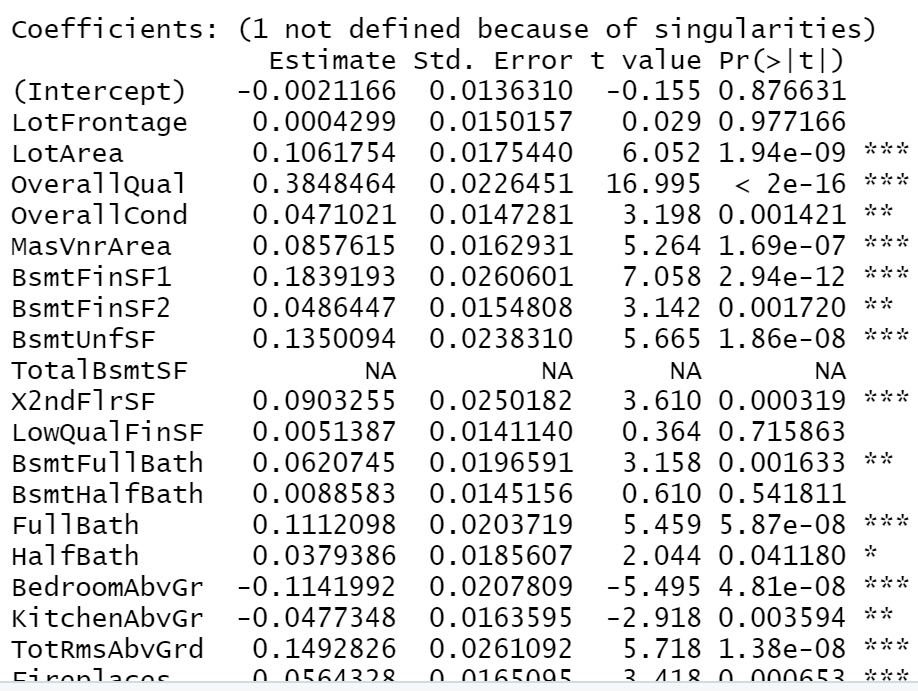
## Data Partitioning

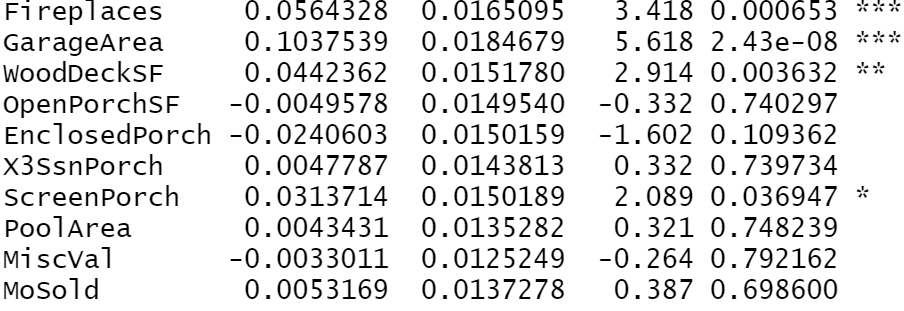
Predictive models were built from datasets made up of the preprocessed numerical and categorical predictors. There were three datasets used for creating the models. One included only the preprocessed numerical predictors, one included both the preprocessed numerical and categorical predictors combined, and the last one included the principal components and preprocessed categorical predictors combined. 25 principal components were used in the last dataset as suggested in the tuning model. Each of these datasets was split into its own training and test set using the createDataPartition function. 80 percent of the samples were used in the training sets, with the remainder used in the test sets.

# **Regression Models**

## Linear Regression Models

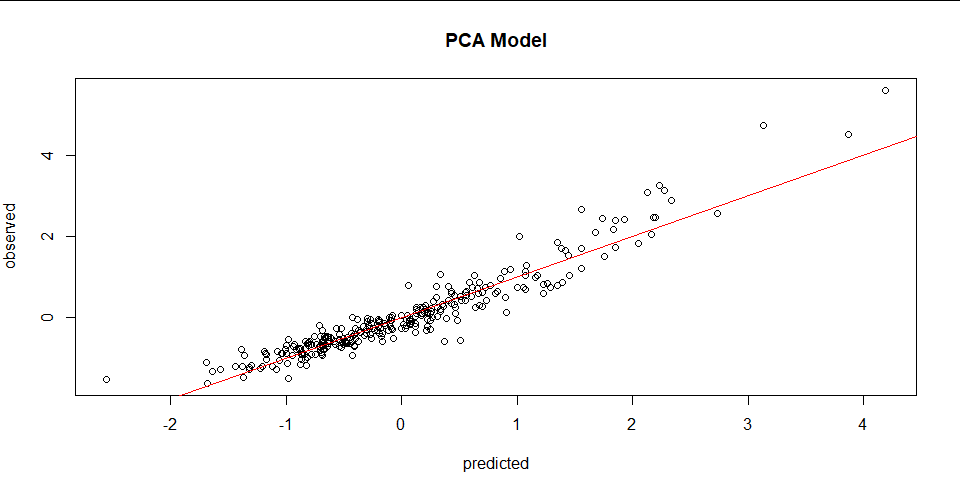
Linear Regression Models were used to identify the relationship between our response variable Sales Price and our training sets. The lm function was used to create ordinary linear regression models, with the response variable sales price measured against all the predictor variables. Ordinary linear models were built using only numerical predictors, with both numerical and categorical predictors, and with PCA transformed numerical predictors and categorical predictors. Out of the three training sets, the PCA component training set produced the best linear fit for our training set. Our PCA regression model was tuned to 25 principal components to best capture the variance of our numerical dataset and cross validated 10 times using trControl. Below is the graph of the results obtained by using the PCA regression model to predict the sale price for the test set. PCA regression does an excellent job with an exception of outliers for when the price is extremely high. The results of our graph also show a strong positive relationship between our predictors and the response variable. This makes sense given that the more attributes such as number of bathrooms, amount of square footage, and areas within the home increase, the more it will cost. The PCA regression model had an R2 of 0.9 and RMSE of 0.32. PLS regression was also performed, however the tuning process was very inconsistent. At times it was recommended to use 8 PLS components and at times it was recommended to use 118, and overall the fit was not any better than regular ordinary linear regression. Whether this was due to a tuning error or to not being the right model fit is unknown, however it was discarded early on as a candidate for best model fit and provided little insight into any data relationships. However, all ordinary linear regression models produced decent results. They all appeared to capture most of the variability in the response for most of the sales price range, however they all seemed to not accurately predict higher prices. It was found that this was the case for only using the numerical predictors, so it is believed that the lower to mid-range priced homes can accurately be predicted using only the numerical predictors that have a linear relationship with the response. Below is a summary of the R code output from the ordinary linear regression model.





As can be seen from the large t value scores and low p values on the majority of the numerical predictors, there is statistically significant evidence to assume that the majority of the numerical predictors have a linear relationship with the sales price response, and as seen from the ordinary linear model graphs and model fit values, that the sales price can be adequately be predicted from low to upper mid-priced homes off of these numerical values. The categorical values improved the model fit and PCA most likely captures most of this variance in less predictors, however there must be some sort of linear relationship between some of the categorical predictors and the higher priced homes that the ordinary linear regression models were not capturing.

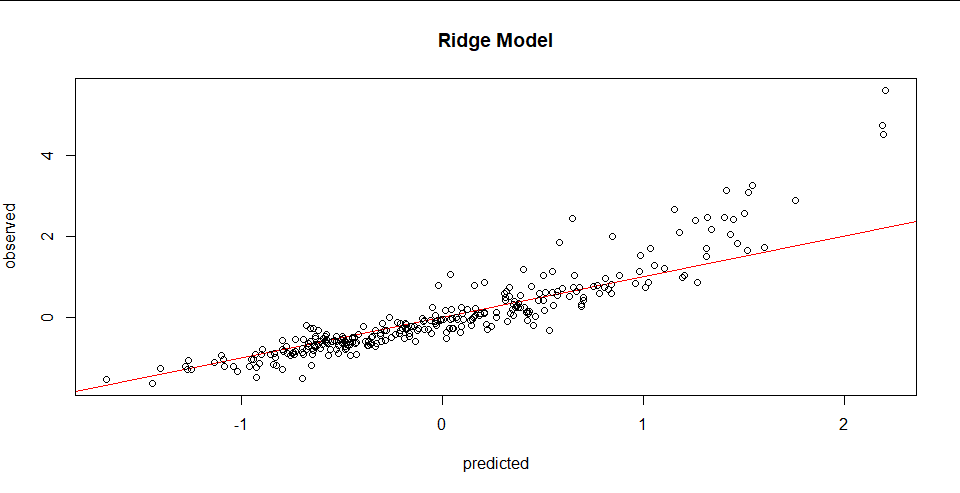
|  |  |  |  |
| --- | --- | --- | --- |
| **Models** | **Tuning** | **R^2** | **RMSE** |
| **Linear Regression** |  | **0.7927591** | **0.4393719** |
| **PCA Regression** | **Ncomp = 28**  **CV = 10** | **0.9015692** | **0.3205605** |
| **PLS Regression** | **NComp = 118**  **CV = 10** | **0.7951528** | **0.4358829** |



## 

## Penalized Regression Models

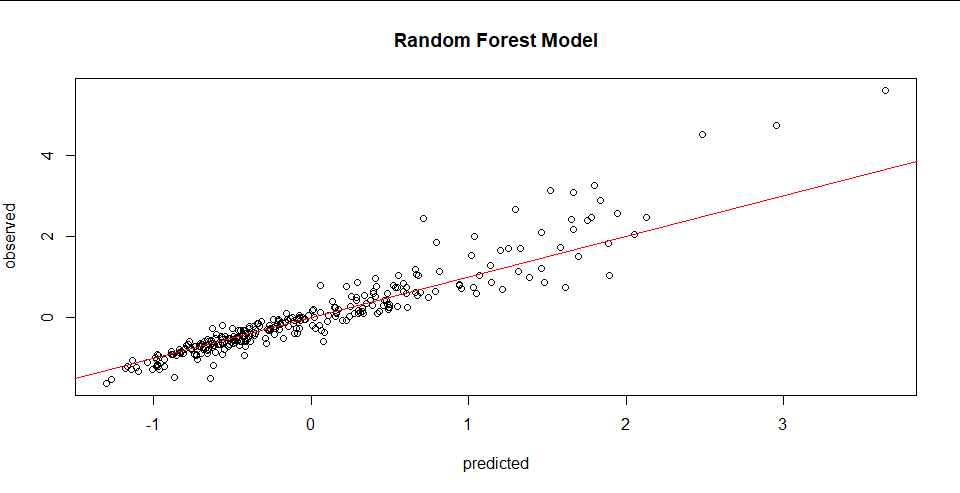
The ridge regression model performed the best out of all the penalized models. By adding bias to the results, ridge regression finds the “sweet spot” where variance and bias meet which produces a model that finds a general pattern in the data. Through tuning, the optimal lambda of 1.57 was found using the cv.glmnet function and cross validated the model 10 times. The R2 came out to be 0.79 with an RMSE of 0.47. However, ridge regression models are known for not overfitting to a dataset and this is proven with the results from the Kaggle submission. Even though the ridge regression model did not perform the best when testing from the fitted model created by the training set, for the Kaggle submission the ridge results performed the best. Below is the graph for the ridge regression model. It captures the general trend of the data without overfitting like the PCA and SVM model. It still appears that there is an issue with predicting values of high cost, which is a pattern seen in most of the models built. This is an indication that some nonlinear approaches may be needed for further improving the model fit.

****

## Nonlinear and Tree Regression

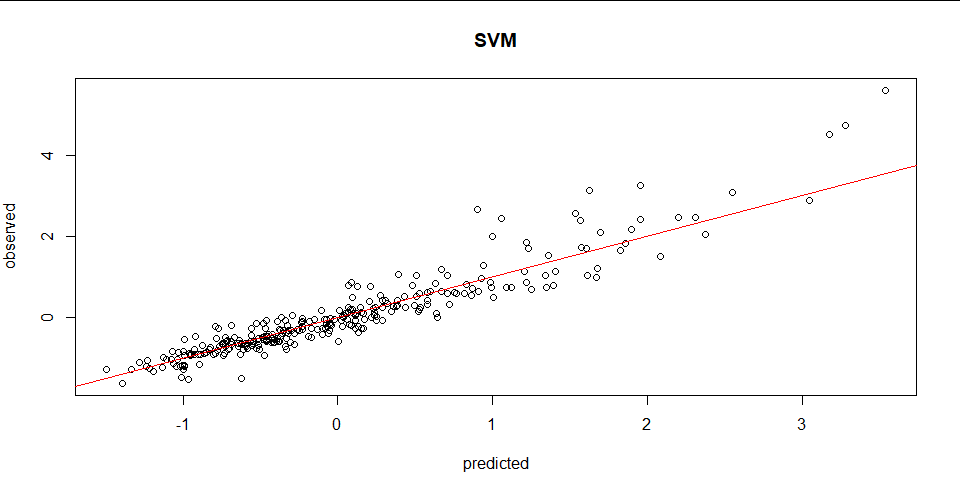
While there were multiple attempts at creating nonlinear models and tree regression models, there were many errors encountered regarding tuning. The train function tuning against parameters often would continue running over an hour, at which point it was uncertain whether it was still running or if the program crashed. Due to inexperience and bad timing some of these errors prevented the use of more nonlinear models in the report, however some were created with good results.

The two models built were random forest and support vector machine models (SVM). Random forest model was expected to produce good results as it could possibly partition effectively to produce a subset of better fits at different price ranges to address the sales price range issue, however the tuning errors mentioned above prevented building a good fitting model. The Mtry value used was the default value produced by the random forest function in R, which was ⅓ the number of predictors. Not tuning this value produced a fit not much better than the PCA regression models. As seen, there seems to be a small improvement in the residual error from higher priced homes, however there are still some issues here regarding a better fit. It is believed that if this function was properly tuned, it may have produced a better fit than ordinary linear regression models using PCA components.



The radial basis function kernel was chosen for tuning the model given its use for SVM and in-class use. A grid was built for tuning the model for cost values from 0.25 to 8 and sigma values from 0.001 to 0.1. This was cross validated 10 times and the optimal decision for the model was a cost value of 8 and a sigma of 0.001. A high cost value and sigma value will introduce more bias in the results and lower variance which from the ridge regression analysis is the right decision for the data due to its size and complexity. The R2 value is 0.895 with an RMSE of 0.318. Here is a graph of the SVM model and as you can see, it performs well at predicting prices until prices are much higher than expected given the predictors. Again, there seems to be an improvement in the residual error on the higher priced homes, and SVM was one of the top candidates for best fitted model using the partitioned training and test set. Exploring different kernel functions and revisiting tuning values may have produced an even better fit.

|  |  |  |  |
| --- | --- | --- | --- |
| **Models** | **Tuning** | **R^2** | **RMSE** |
| **Random Forest Regression** |  | **0.8592079** | **0.3625748** |
| **Support Vector Machine** | **Cost = 8**  **Epsilon = .001** | **0.8951272** | **0.3181096** |



## Discussion

Included in this section is a table for all the models built for this project, tuning parameters, R2, RMSE, and important predictors. From looking at the table, you can see that PCA regression, support vector machine , and random forest have the highest R2  value with the lowest RMSE when testing the partitioned test set on models built from the training set. However, the results from the Kaggle submissions ([Appendix C](#_uq4y2whve13w)) shows that they did not perform as well in predicting the future housing prices for the test set provided by Kaggle. The Ridge and PLS Regression models performed the best for predicting future housing prices for the test set. There were a lot of errors in tuning and building the PLS model so it is still uncertain as to exactly why this performed better, although it may have captured some of the total variance not accounted for in PCA regression due to the fact that the categorical predictors were processed in the PLS transformation, however there is too much uncertainty here to go into more detail. The ridge and PLS had a higher RMSE for the Training set and a lower R2 value. This suggests that the PCA regression, random forest, and support vector machine models were overfitting for our training set which is why it was not performing well on the Kaggle test set. Ridge and PLS regression may have been more effective in capturing the general trend for the data without predicting it exactly. This could be due to too many categorical predictors being used, which needs to be reduced for any future efforts on this project.

The important Predictors for the data are Principal Component 1, Overall Quality, and BsmtFinSF1. Principal Component 1 consists of areas, square footage, and bathroom count measurements. Overall Quality refers to the material finish of the house and BsmtFinSF1 refers to the size in feet of the basement area. As shown earlier in the PCA component analysis, principal component one was primarily influenced by variables consisting of areas, square footage, and bathroom counts, all of which showed up as important variables in other models. It also agrees with Homeguide which says that the amount of square feet and areas in the house are big factors in determining the final price of a house. Moreover, according to Homeguide, 41.5% to 50% of the final sale prices is determined by the materials used to build the house. The data set description describes overall quality as “rates the overall material and finish of the house”, which directly agrees with Homeguide and makes sense as to why it was the most important variable in some of the models.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Models** | **Tuning** | **R2** | **RMSE** | **Important Predictors** | **Overall/Node Impurity** |
| **Linear Regression** | **CV = 10** | **0.7927591** | **0.4393719** | **BsmtFinSF1** | **11.36** |
| **PCA Regression** | **Ncomp = 28**  **CV = 10** | **0.9015692** | **0.3205605** | **PC 1** | **26.2** |
| **PLS Regression** | **NComp = 118**  **CV = 10** | **0.7951528** | **0.4358829** | **OverallQual** | **0.13** |
| **Ridge Regression** | **Alpha = 0**  **CV = 10**  **Lamda = 1.570689** | **0.7902284** | **0.4781130** |  |  |
| **Random Forest Regression** |  | **0.8592079** | **0.3625748** | **OverallQual** | **3.31e + 02** |
| **Support Vector Machine** | **Cost = 8**  **Epsilon = .001** | **0.8951272** | **0.3181096** |  |  |
| **Lasso Regression** |  | **0.7486046** | **0.5453981** |  |  |

# Conclusions

## Problems Encountered

There were some problems encountered throughout the project that are worth noting. In hindsight, the biggest concern is that there are too many preprocessed categorical binary variables included in the model. After converting each category level to a new column, there are 270 categorical predictor variables. Including this many binary predictors raises concern regarding model performance and overfitting. ANOVA was used to reduce categorical variables with minimal variability between levels, however this did not reduce enough of the variables and other categorical reduction methods were not explored. More efforts are needed here for reducing the number of category variables used. Problems were also encountered during tuning certain models, causing tuning errors when model building. Neural net was not included in the final project because the tuning function did not run properly, it was either running for too long or crashing. This same problem was encountered when building random forest models, so we were not able to effectively tune the parameters for the best model fit. Random forest was still included with the default values, as it was believed to be a good candidate had it been properly tuned. The last problem encountered worth noting was rank deficient warnings. When building the linear regression model prediction set using a test set with the model built, warning messages were encountered saying that a rank deficient fit may be misleading. The only efforts to try to prevent this were modifying the size of the training and test set, but more efforts may be needed here to see if it really was affecting model performance.

## Conclusion

Overall, the project was successful in building models to predict house prices. It was proven with the ordinary linear regression models that there is definitely a strong linear relationship between the numerical predictors and the sales price response. There is still more work to be done in effectively capturing the categorical predictor’s variability and how they influence the sales price. The model that performed the best in the Kaggle competition was the ridge regression model. Following close behind is the PLS regression model. On Kaggle, our best rank for our submission is 4711 with an RMSE score of 0.53513, see results [here](#_uq4y2whve13w). PCA, SVM, and Random Forest were overfit for the training set, so they performed well in predicting the partitioned training data set but not for the Kaggle Test set. More efforts are needed here in reducing the dimension of the categorical predictors so that they can be more effectively used with the Kaggle competition set, which when done so properly would be expected to produce much better results than what we achieved.

# How to Run Code

1. **Download** and **Extract** zip file into a specified directory.
2. **Open** housePricesFinal.R
3. Scroll down to “directory set sourcing” and **set your directory** to the location of the extracted files
4. **CTRL + A, CTRL + Enter**

*\*For additional assistance, see readme file in folder containing R code file*

# References

<https://homeguide.com/costs/cost-to-build-a-house>

<https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html>

<https://towardsdatascience.com/ridge-regression-for-better-usage-2f19b3a202db>

<https://www.statisticshowto.com/ridge-regression/>

# 

# 

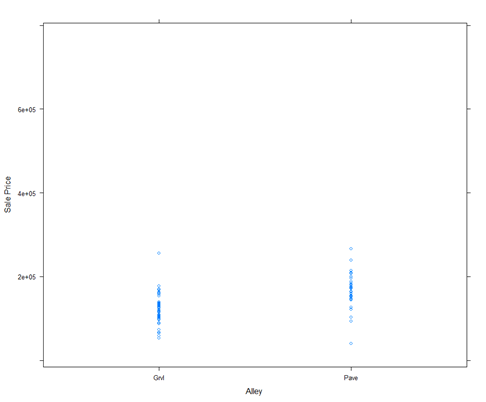
# 

# 

# Appendices

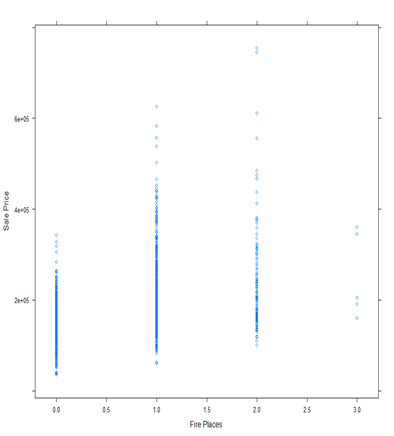
## Appendix A: Data Visualization

*Plot 1: Alley vs Sale Price*



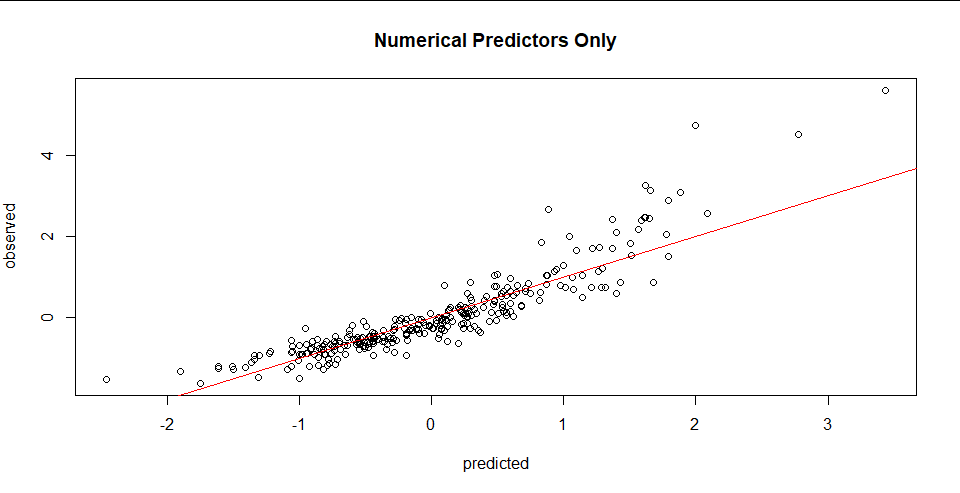
This is an xy graph of Alley vs Sale Price. Here you can see how “Paved” alleys have higher sale prices than “Gravel” alleys.

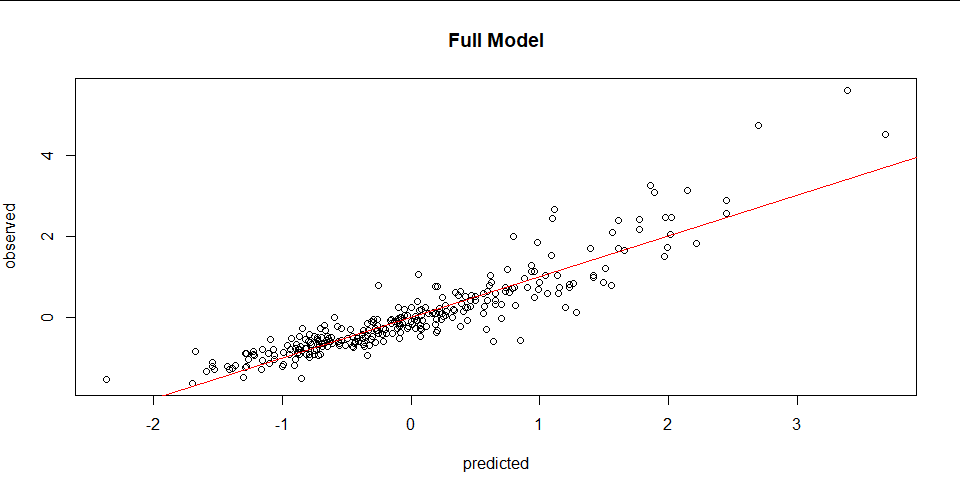
*Plot 2: Fire Places vs Sale Price*

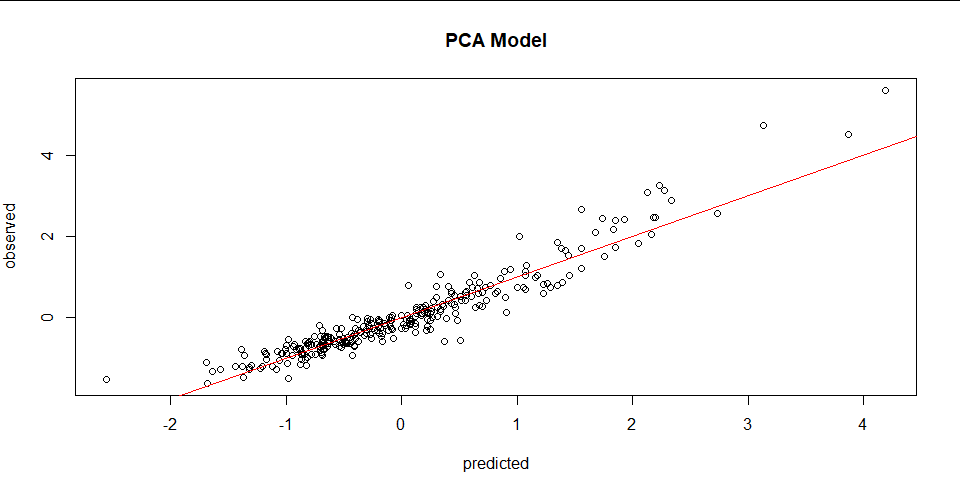


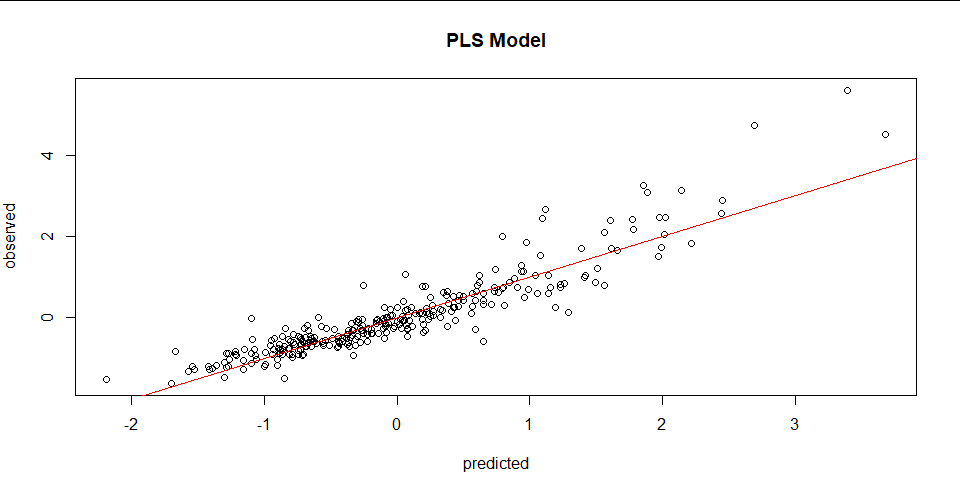
This is an xy plot graph of Fire Places vs Sales Price. Houses with 2 fireplaces will cost more than a house with 0 or 1. Moreover, you can see that in Iowa, there aren’t a lot of homes with more than 2 fireplaces which could be an indicator of consumer demands and the wealth of the area.

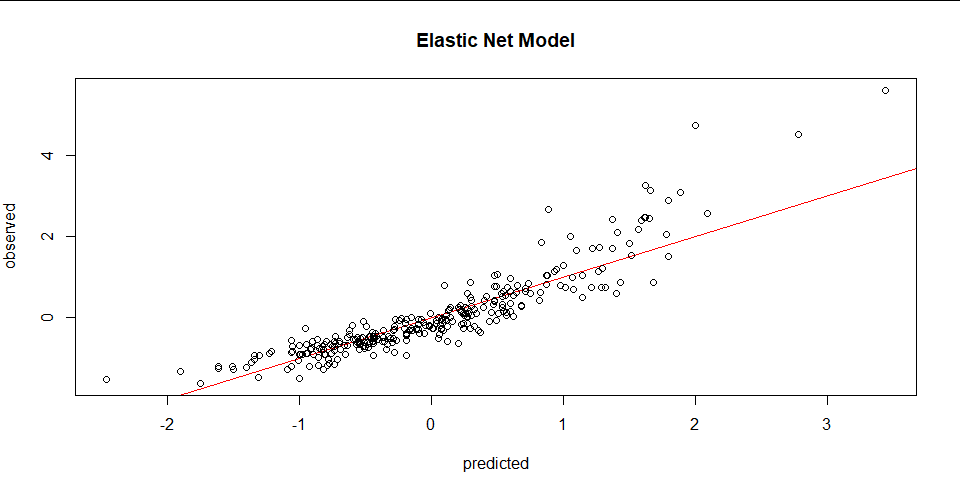
## Appendix B: Regression Graphs

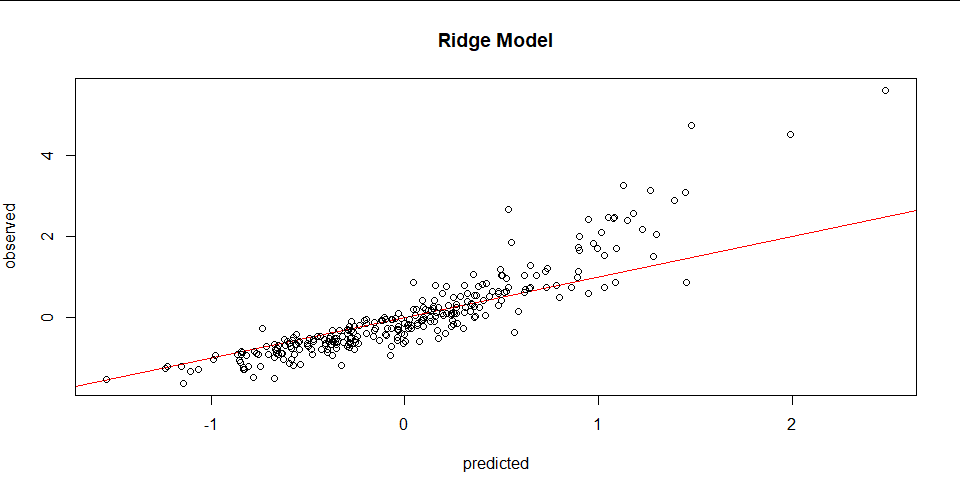


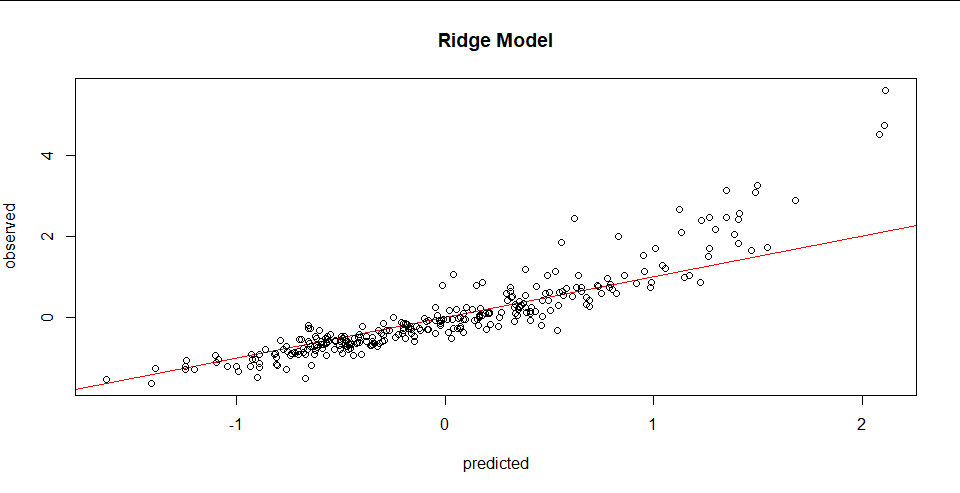


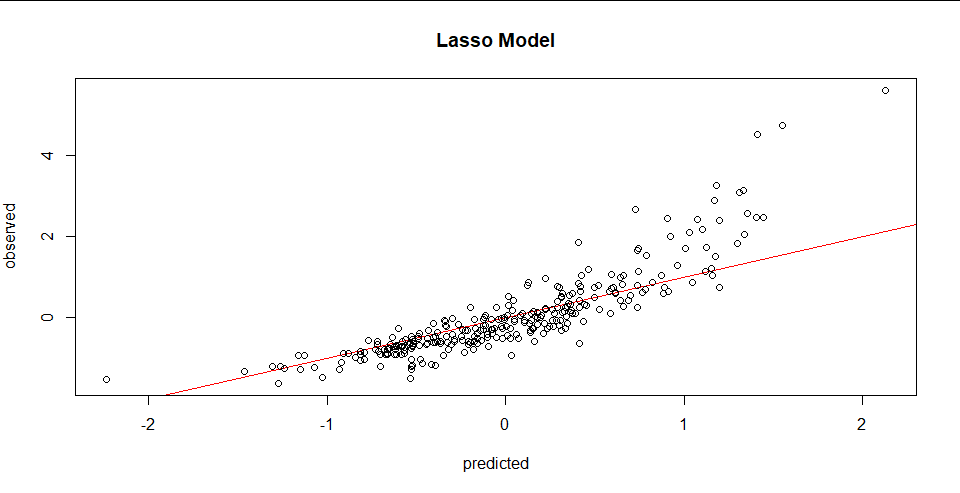


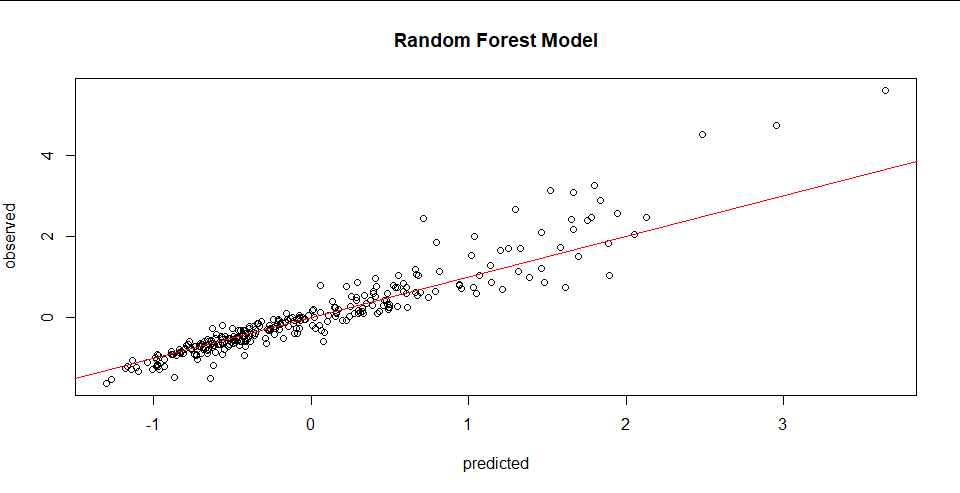
****

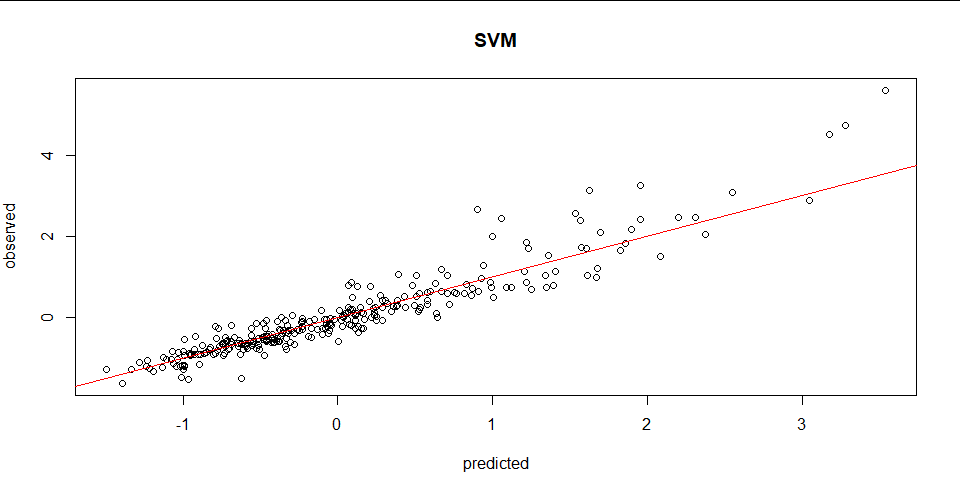
****

****

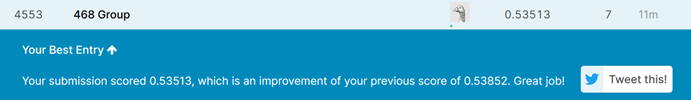
****

****

****

****

## Appendix C: Kaggle Results



|  |  |
| --- | --- |
| **Models** | **RMSE** |
| **Linear Regression** | **4.98345** |
| **PCA Regression** | **1.35564** |
| **PLS Regression** | **0.53852** |
| **Ridge Regression** | **0.53513** |
| **Random Forest Regression** | **0.95172** |
| **Support Vector Machine** | **0.95830** |