Practicum II: Douglas County, CO Home Sales Price Predictions

Odes Mullins

Regis University

Douglas County, CO Home Sales Price Predictions

**Introduction**

The goal of this project was to predict home sale prices for Douglas County, Colorado. Having a good predictive model would be a great tool to use to predict future home sale prices of properties for Douglas County. I extracted home sales data for years 2014 through 2018 from the Douglas County assessor’s office. I used multiple regression models with machine learning to train and predict home sale prices. The models were analyzed and modified to determine which model had the greatest predictive accuracy.

I evaluated the data and concluded that other types of housing properties as well as all properties other than single family detached homes had too many varying factors and would be difficult to evaluate and accurately predict so I chose to only predict prices for single family homes. There were over 28,000 single family home sales in Douglas County over the five-year period. I performed the required tasks in R and created a companion R Markdown file.

**Data Selection and Cleaning**

The data used for this project was extracted from the Douglas County assessor’s office website under the Online Services/Advance Search option. There are a series of reports which can be generated in a .csv format and I chose to generate three reports which included the features required to build the models. The three reports I used were **Property Improvement (Segments) Report**, **Building Summary Report**, and **Sales Information Report**. I downloaded the three .csv files to my RStudio project working directory. There were 69 columns for the three files with 8 columns being keys across the three files leaving 45 columns to evaluate to determine which columns were needed for the predictive models.

I read the three .csv files into a data frame in RStudio then I cleaned the data. I evaluated the data and converted some columns to numeric initially and then I evaluated all the columns and converted several columns to factor depending on the values. I also found some odd inconsistencies in the lower home prices so I decided to remove all sales with a sales price less than $150,000 which was mostly values of $0 for properties being placed in trusts so they technically were not sales. I removed duplicate rows when the values in all columns were the same. I removed columns from the three files which were not needed for the predictive models and then I combined them into the final file. I kept some identifying key columns in the final file in case there was a need to compare the file back to the original three files but that information was not used in the predictive models.

I chose 15 features for predicting home sale prices. These features are described below in Table 1. I extracted Sale Year from Sale Date to use in my regression models for predictions. I changed Style and Quality to a number range for easier comparison and imputed missing values for Walkout Basement due to values missing when there is no basement. There were only eleven records with missing values requiring them to be deleted. I converted several numeric features to factor for model processing as shown in the Table 1. There were over 100 values for “Year Built” which I initially tried modifying to factor but the models got errors so I determined it was easier to leave it as a number value instead.

|  |  |  |
| --- | --- | --- |
| **FEATURES** | **DESCRIPTION** | **UNIT** |
| Acres | Lot Size | acres |
| Improvmnt SF | above ground home size | square feet |
| Garage SF | garage size | square feet |
| Basement SF | total basement size | square feet |
| Finished Basement SF | basement finished size | square feet |
| Total Porch SF | total porch size | Square feet |
| Year Built | year home built | number |
| Situs Zip Code | property zip code | factor |
| Sale Year | property sale year | factor |
| Style | type of home such as ranch | factor |
| Stories | number of stories tall | factor |
| Bedroom Count | number of bedrooms | factor |
| Bathroom Count | number of bathrooms | factor |
| Quality | home quality rating | factor |
| Walkout Basement | is there a walkout basement | factor |

Table 1: Summary of Features

I decided against using the assessor’s property estimate used for property taxation as the estimate is only made every two years and it tends to not be an accurate predictor of a property sale price especially with the rate of property value increases within Douglas County.

**Exploratory Data Analysis**

The initial data files had to be extracted, cleaned, combined, and columns not used in the regression models were removed. The final data frame consisted of 15 independent variables along with the one dependent variable (home sale price) and a total of 28,523 rows. Home prices are between $150,000.00 and $6,750,000.00.

I wanted a visual of the home sales locations in Douglas County so I mapped the sales in Douglas County, Colorado which required obtaining the latitude and longitude of each address. This task required a lot of effort but it was very beneficial to see where the actual sales were instead of just looking at numbers even though the latitude and longitude were not used in the predictive models. I used Bing to obtain the coordinates as there is a free license available for students from Bing. I created a general account at link <https://www.bingmapsportal.com/#> and then sent an email to Bing Maps Enterprise Account Administration requesting the special education key. I ran the process using a location address which I created by concatenating each property street address, city, state, and zip code and extracted the latitude and longitude for property address of each property sold. I commented out the code in my R Markdown file which extracts the Bing coordinates as the extraction only needed to be performed once. I read the file I saved with the latitude and longitude and substitute the values into my newly created file when I reran the R Markdown code. I created a map using census information in Federal Information Processing Standards (FIPS) format from the United States Census Bureau, extracting the coordinates necessary to create the Douglas County, Colorado map. This took some additional effort since all the coordinates used to draw the entire Colorado map along with all the counties were in the initial file and Douglas County was broken into voting districts so I had to remove unneeded coordinates for the internal lines and only create the Douglas County map with the external boundary lines. The Douglas County coordinates for the map outline were out of sequence due to how the original map drew divisions within the map so I had to modify sequence numbers to get the map to print correctly. I decided to map the entire five years home sales (shown in Figure 1 below) but also only 2018 home sales (shown in Figure 2 below) to see if there were specific areas of Douglas County which were more active for that year. I found the maps relatively similar with the 2018 map sales being much sparser in some areas indicating that sales locations have been consistent over the five-year period.

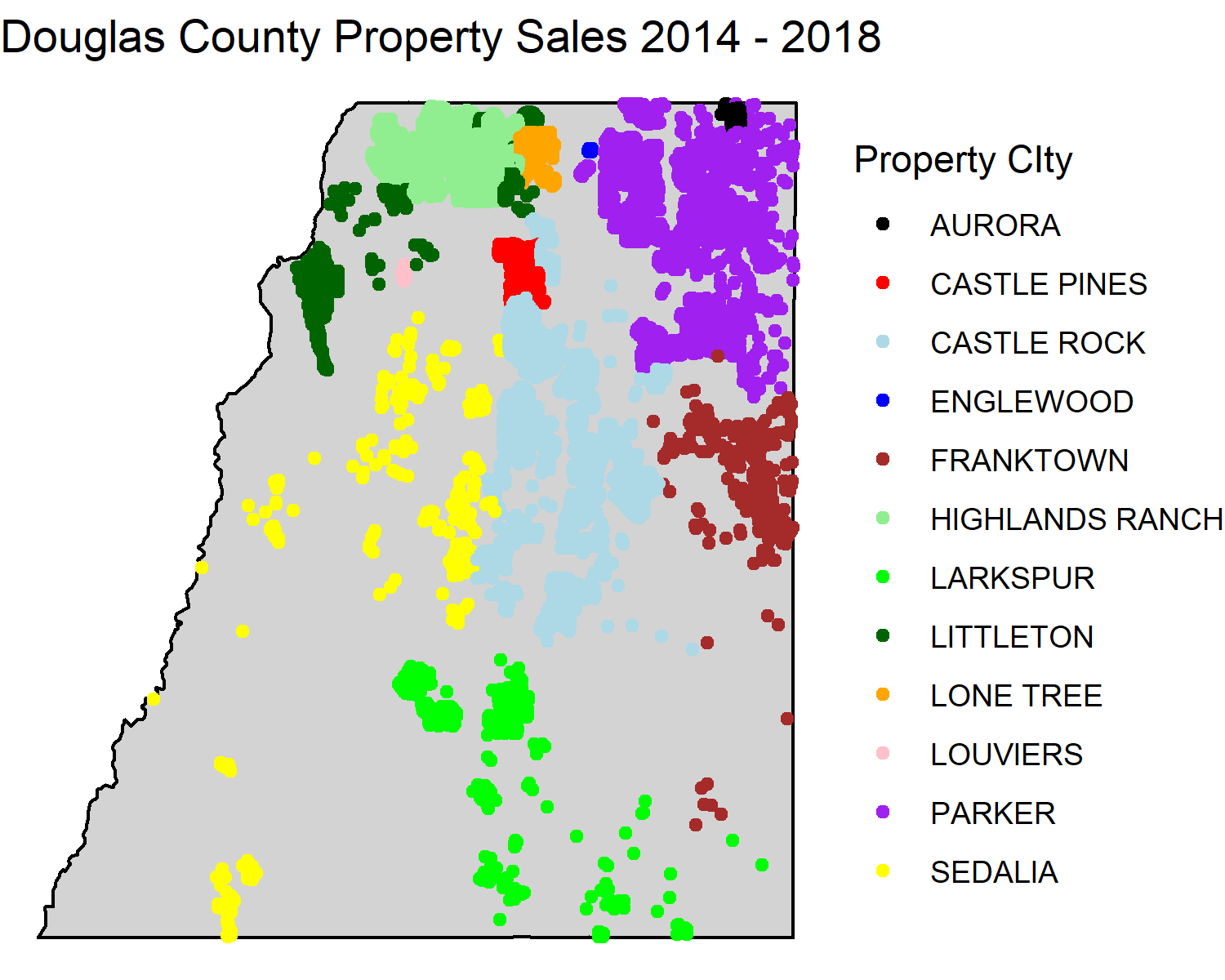


Figure 2: Home Sales 2014 – 2018

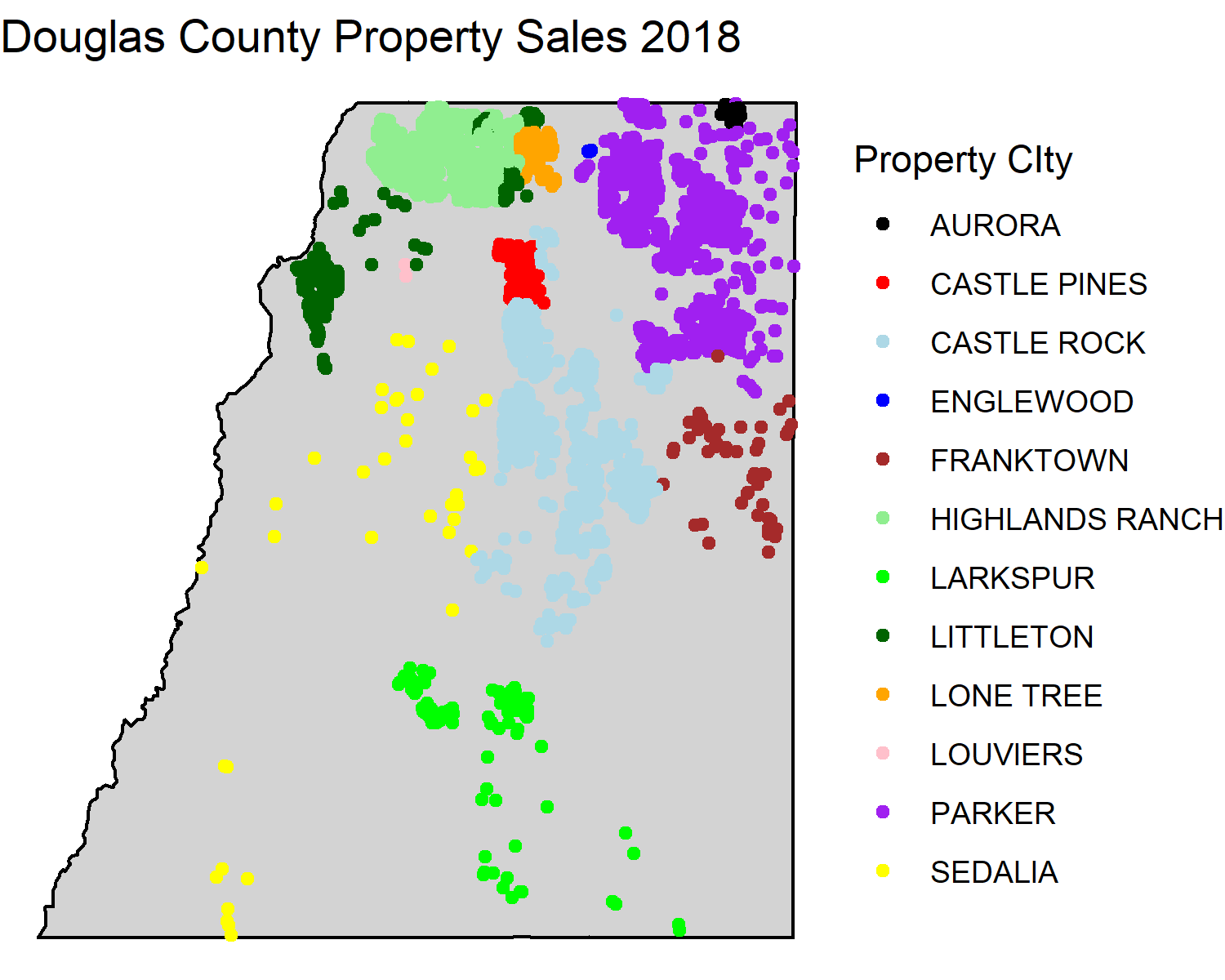


Figure 2: Home Sales 2018

I graphed several of the independent variables against the home sale price to evaluate how they compared. This gave a good look at each variable compared to the sale price alone. Figure 3 below shows the Sale Year with Sale Price. It is expected that with inflation the home prices will increase over time. This graph may show home prices upward trend over time but it is difficult to determine with these two variables only.

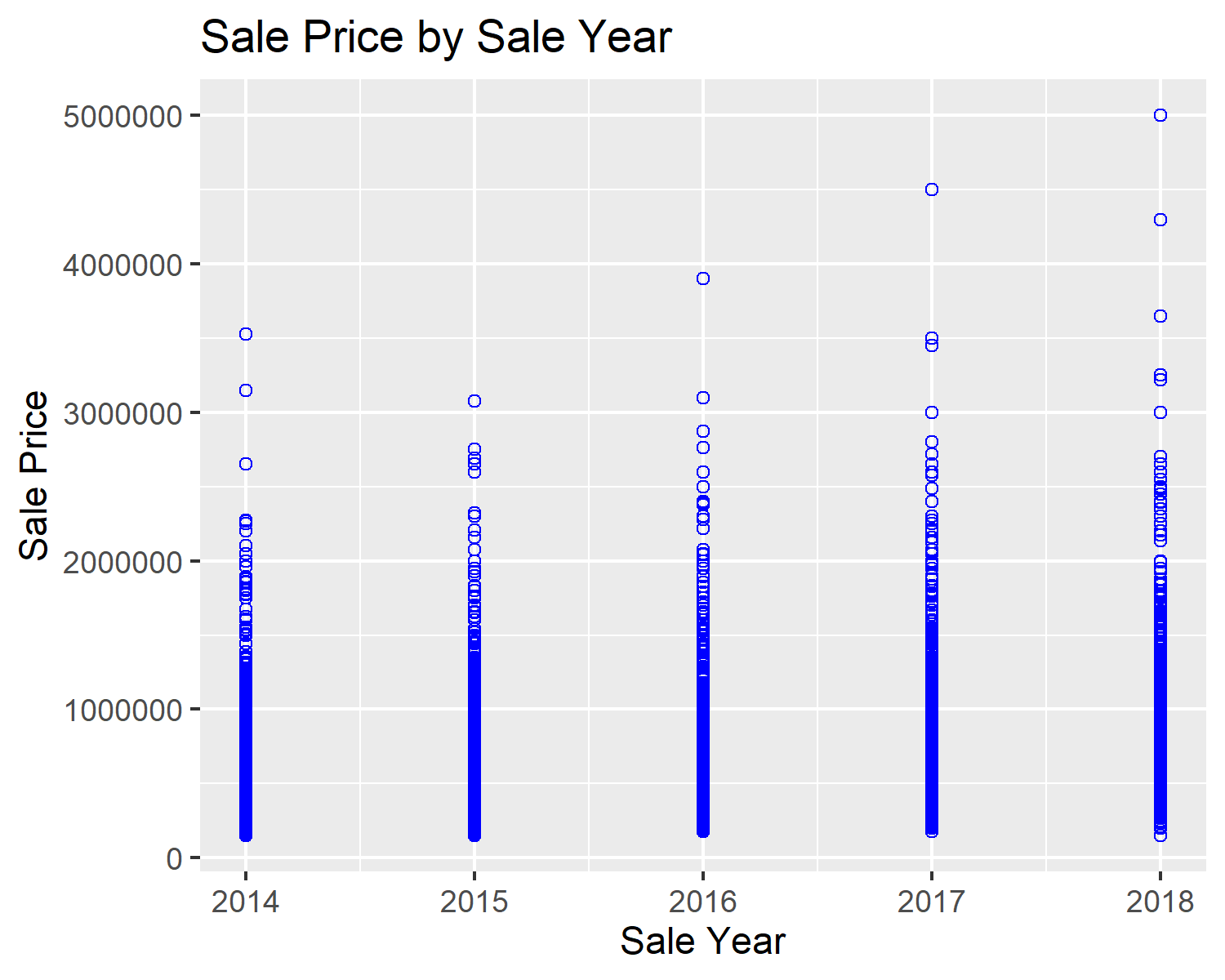


Figure 3: Home Sales 2018

Figure 4 below shows a comparison of home square feet (not including basement square feet) and sale price. This graph shows a correlation between the two variables. Larger homes require more building resources and time so it is expected they would also cost more so the correlation is as expected.

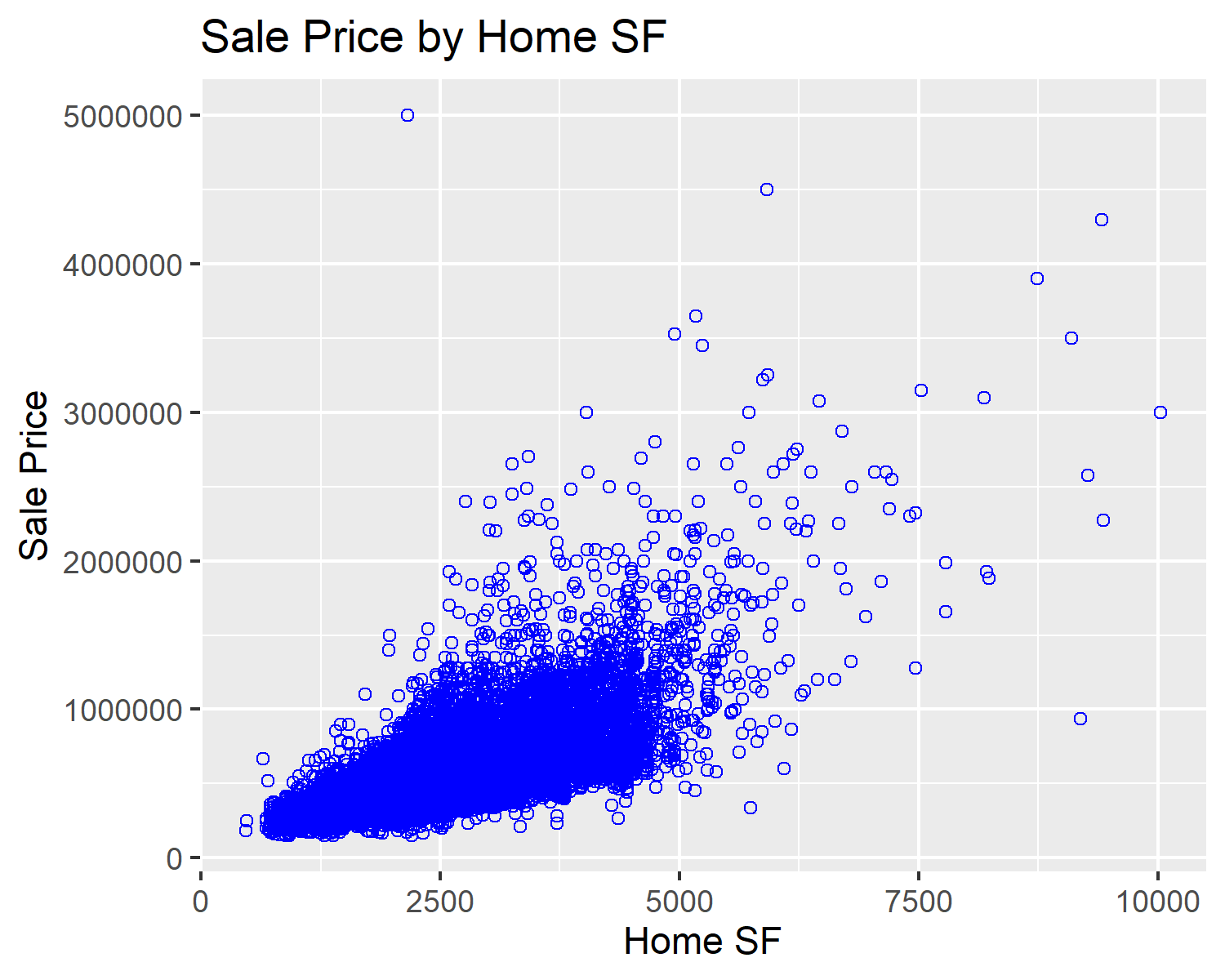


Figure 4: Home Finished SF to Sale Price

Figure 5 below shows a comparison of total home finished square feet (includes basement finished square feet) and sale price. This graph shows a similar correlation between the two variables like Figure 4 above.

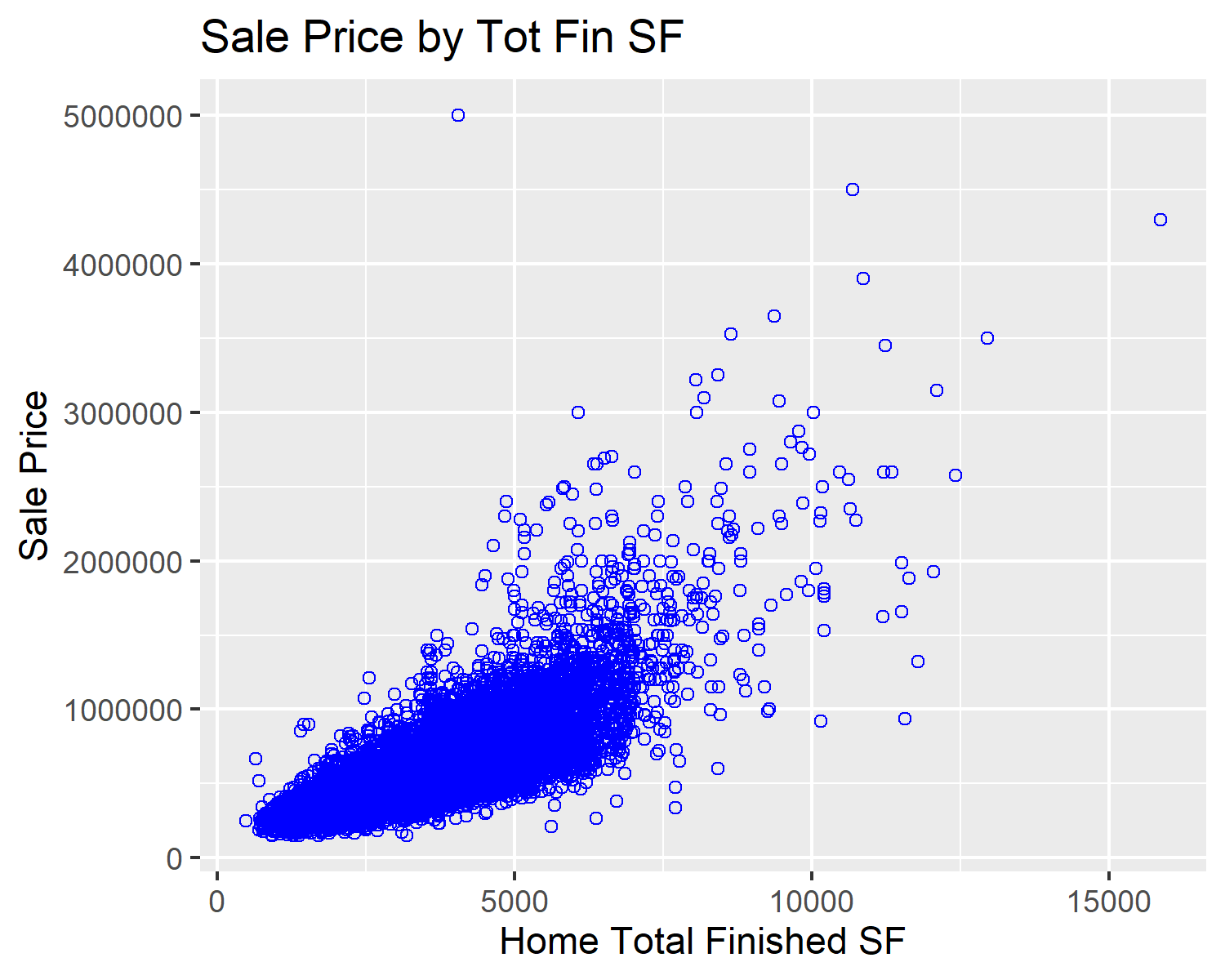


Figure 5: Total Home Finished SF to Sale Price

Figure 6 below shows a comparison of home quality and sale price. This graph shows a trend for increased home price as quality increases. There are some outliers like the $5,000,000.00 price for a property with quality level 2 but investigation shows that is due to the land size which is multiple acres causing the unexpected high price.

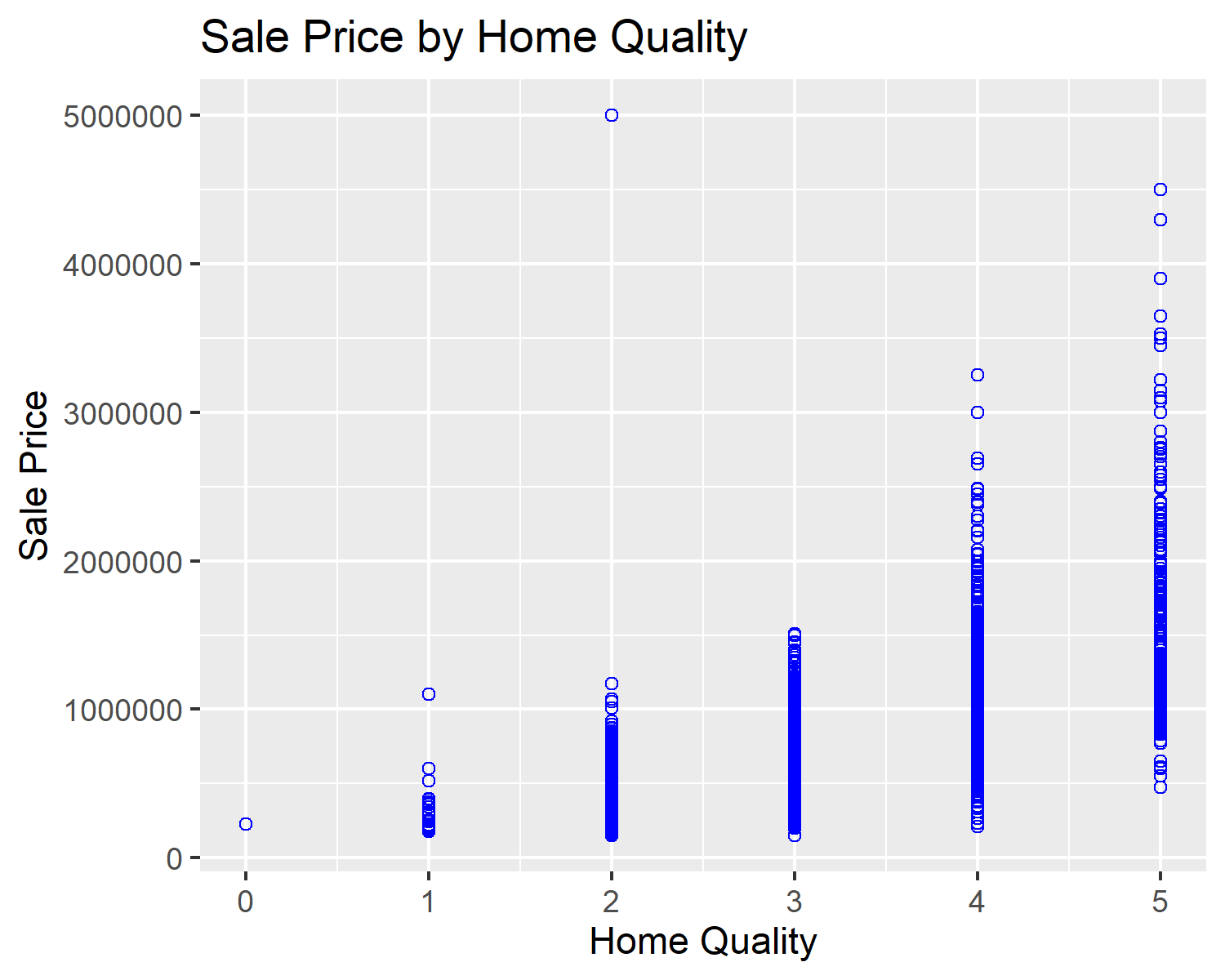


Figure 6: Home Quality to Sale Price

Figure 7 below shows a comparison of home year built and sale price. This graph shows that the newer the home have a higher sale price. I noticed one outlier from the 1950’s for $5,000,000.00 price which is most likely due to the land value as shown before in several other slides. It is not obvious if the price is due to the newer homes being larger or if newer homes are more desirable but it is probably a combination of both.

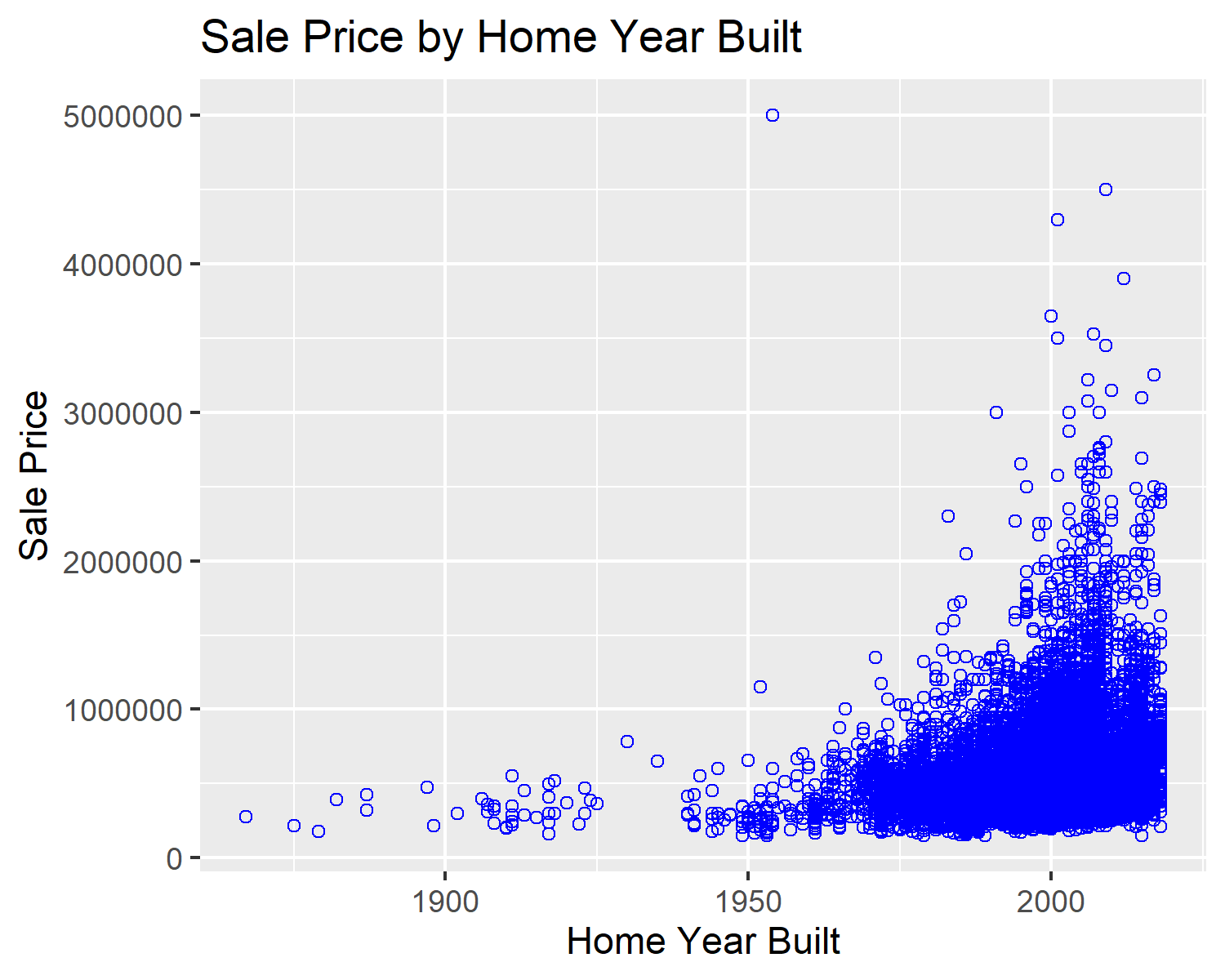


Figure 7: Home Quality to Sale Price

I created several other graphs of pairs which are in my R Markdown file but they did not show any noticeable correlation by themselves so I am not including them here. I also decided to calculate median sale price and graph based on sale year as shown in Figure 8 below. This graph gave very useful information showing a median price increase of over $120,000.00 over the 5-year period.

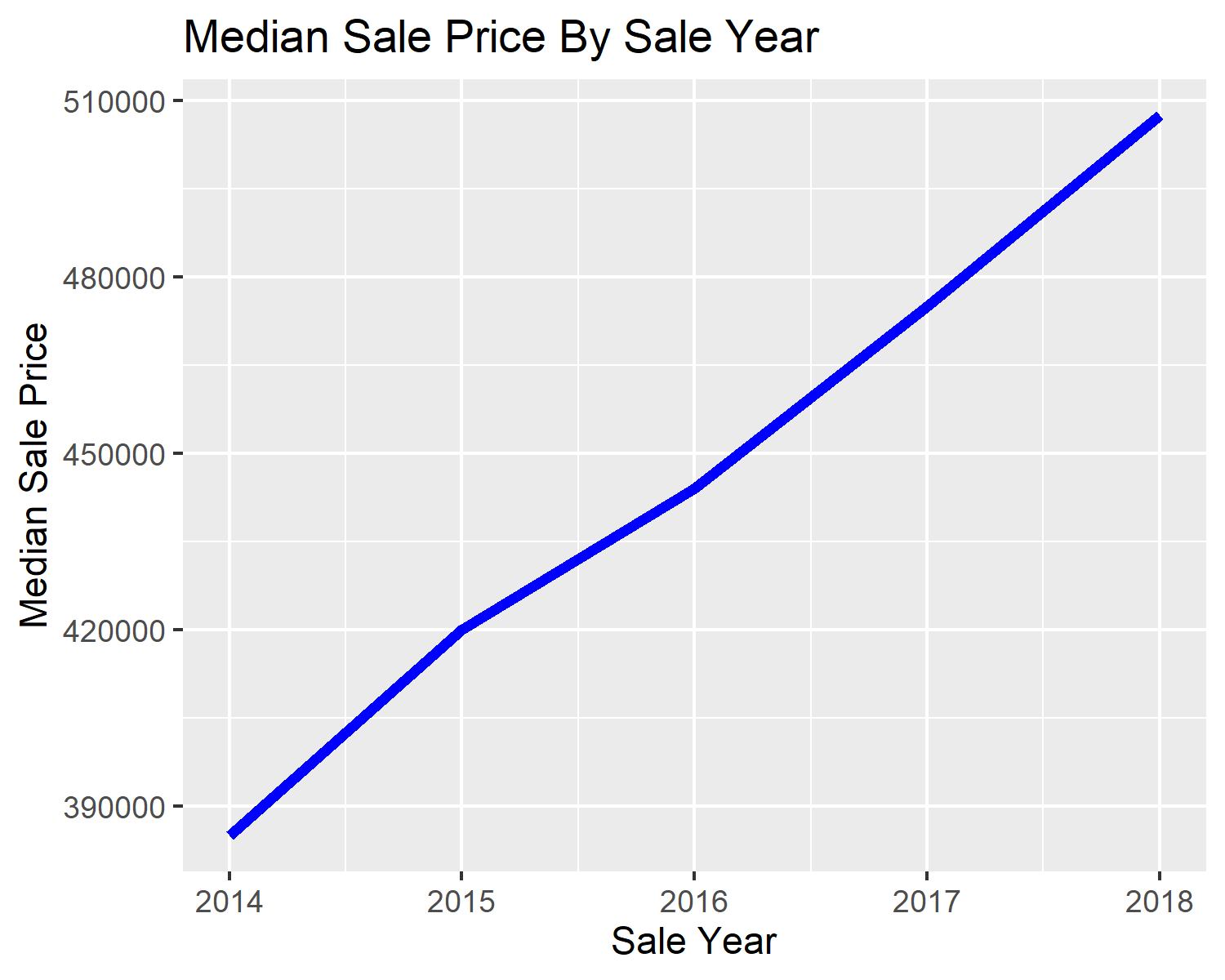


Figure 8: Median Home Sale Price to Sale Year

I calculated the price per square foot for each home then I calculated the mean of the price per square foot values. I graphed the mean value by sale year as given in Figure 9 below. The home increase per square foot is over time is obvious with home prices increasing from just under $185/SF to over $250/SF.

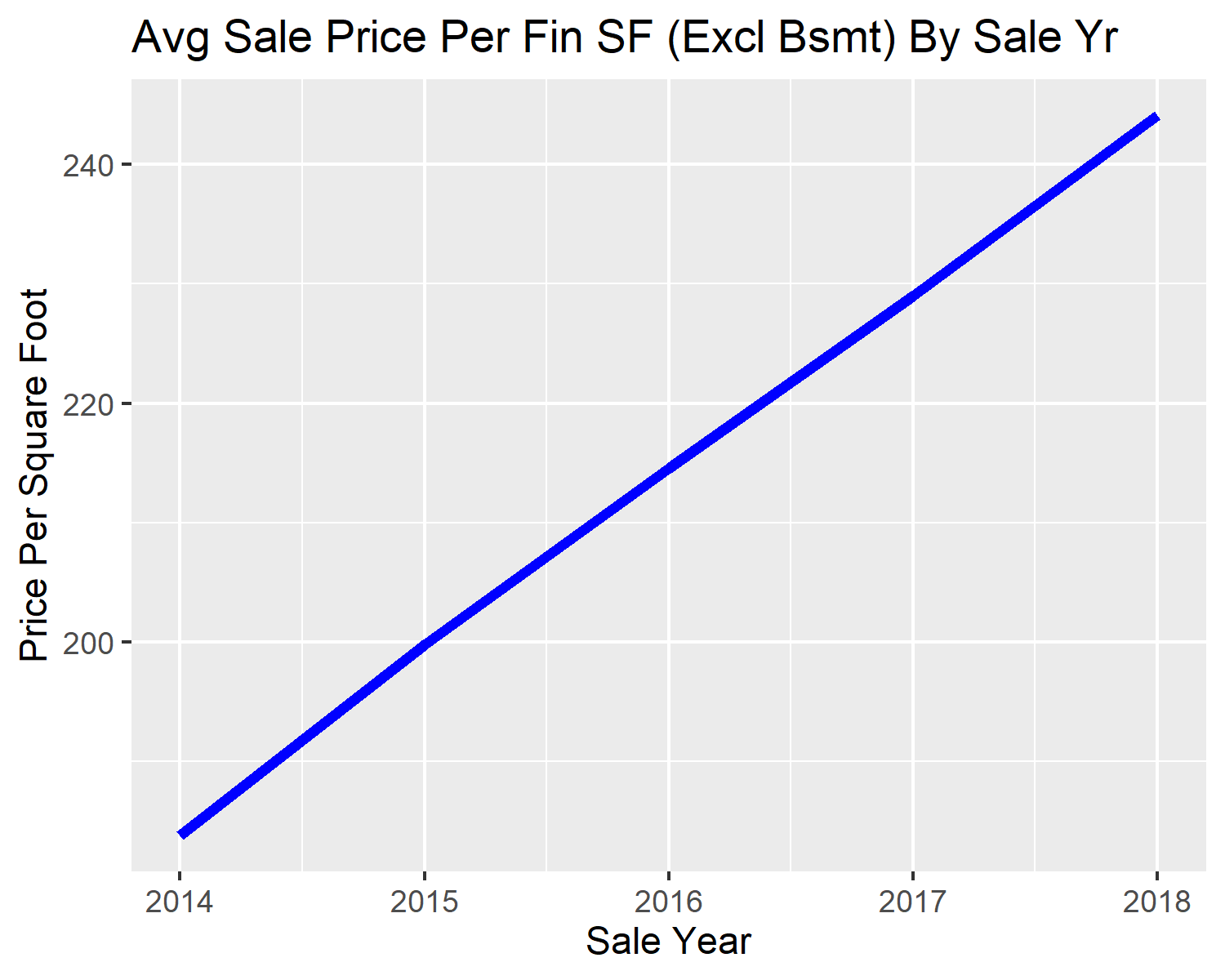


Figure 9: Average Home Sale Price Per Square Foot to Sale Year

I combined above ground finished square feet with finished basement square feet and then I calculated the price per square foot for the combined value. I then calculated the mean of the combined values. I graphed the mean value by sale year as given in Figure 10 below. This graph shows a similar trend to Figure 9 above with a price per square foot increasing from around $148/SF to over $195/SF.

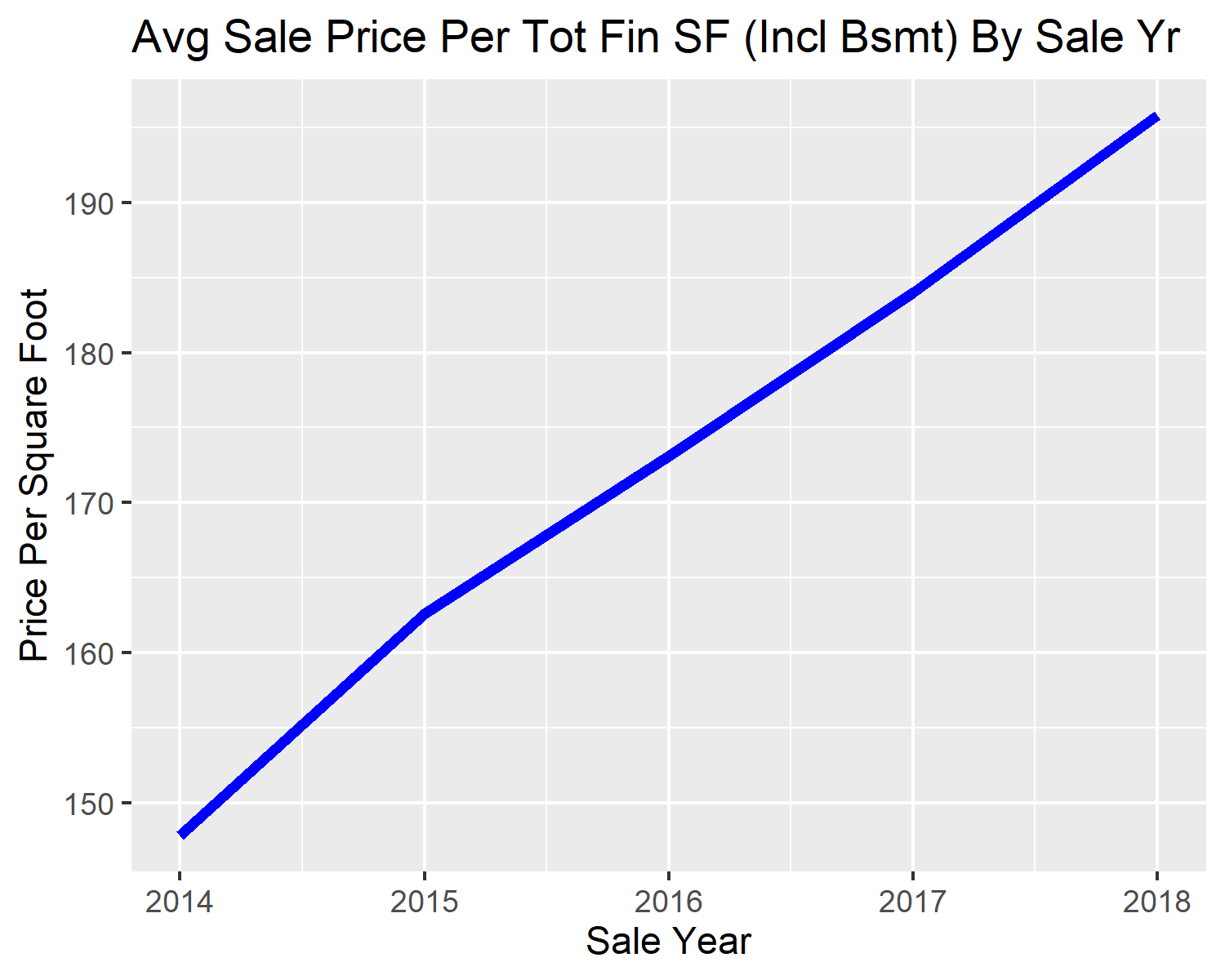


Figure 10: Average Home Sale Price Per Total Square Foot to Sale Year

I also created three boxplot graphs for evaluation. The first boxplot graph compares home style to sale price. Some home styles vary quite a bit on sales price but in general most categories have the same average sale price except home style 25 which appears to be lower than the other prices but there are only 99 sales for style 25 which is a three-story home. There were only 3 sales for home type 11 (A Frame) and 17 sales for style 9 (2 ½ story). I did not see anything which stood out for this comparison.

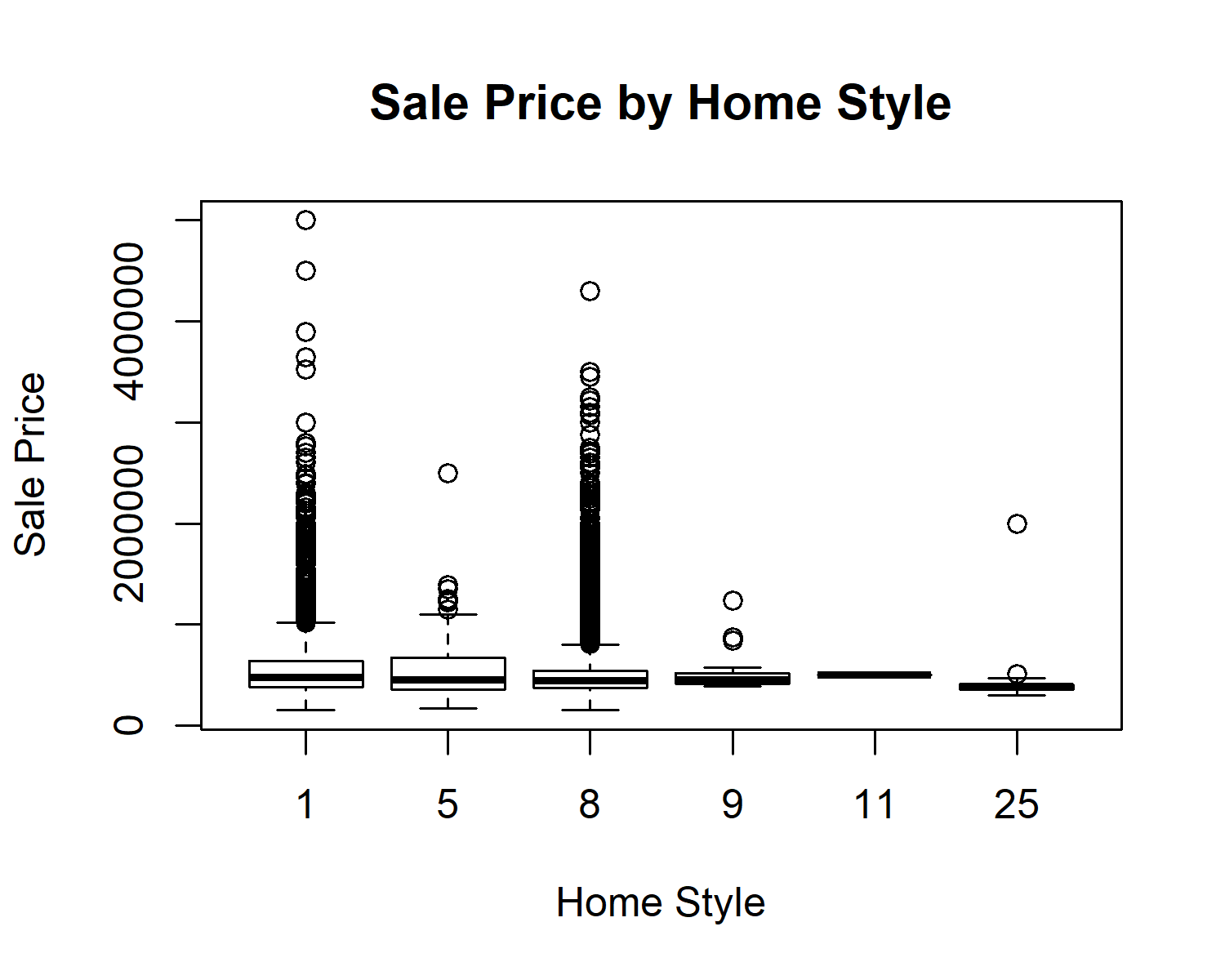


Figure 11: Home Style to Home Sale Price

I also created a boxplot for property city to sale price. There are several cities where the average sale price is quite a bit more than others but there is a lot of variety and it is difficult to determine the average difference based on the graph as there are some very expensive homes and some city’s home sale prices were spread over several million dollars therefore the graph scale caused the city sales differences to not be as clear as shown in Figure 12 below.

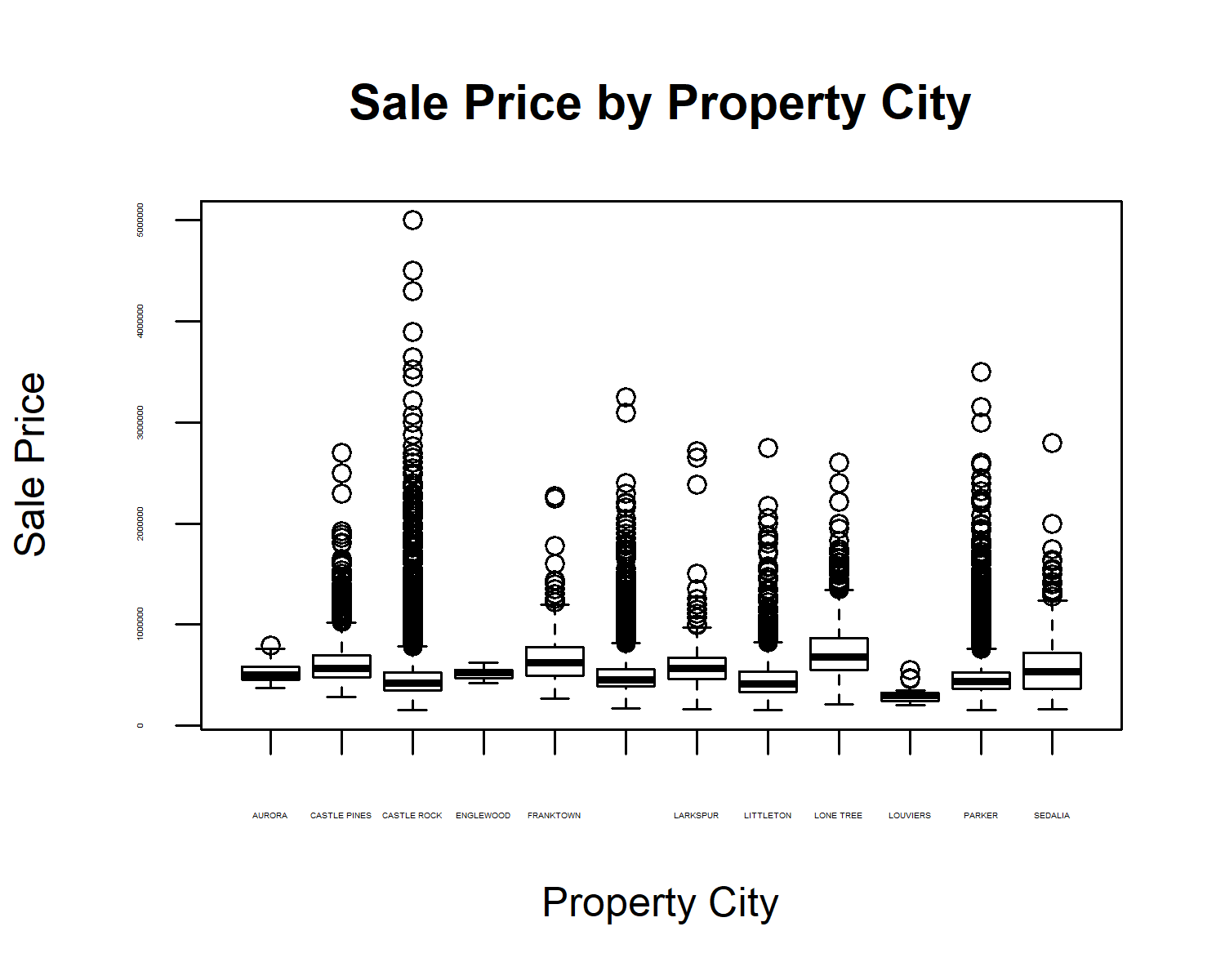


Figure 12: Property City to Home Sale Price

The next boxplot shows the relation of the property city to the price per square foot as indicated in Figure 13 below. There is a little difference for some cities but not as much as Figure 12 shows above which may be due to there being larger houses for those cities but the price per square foot is similar between the cities.

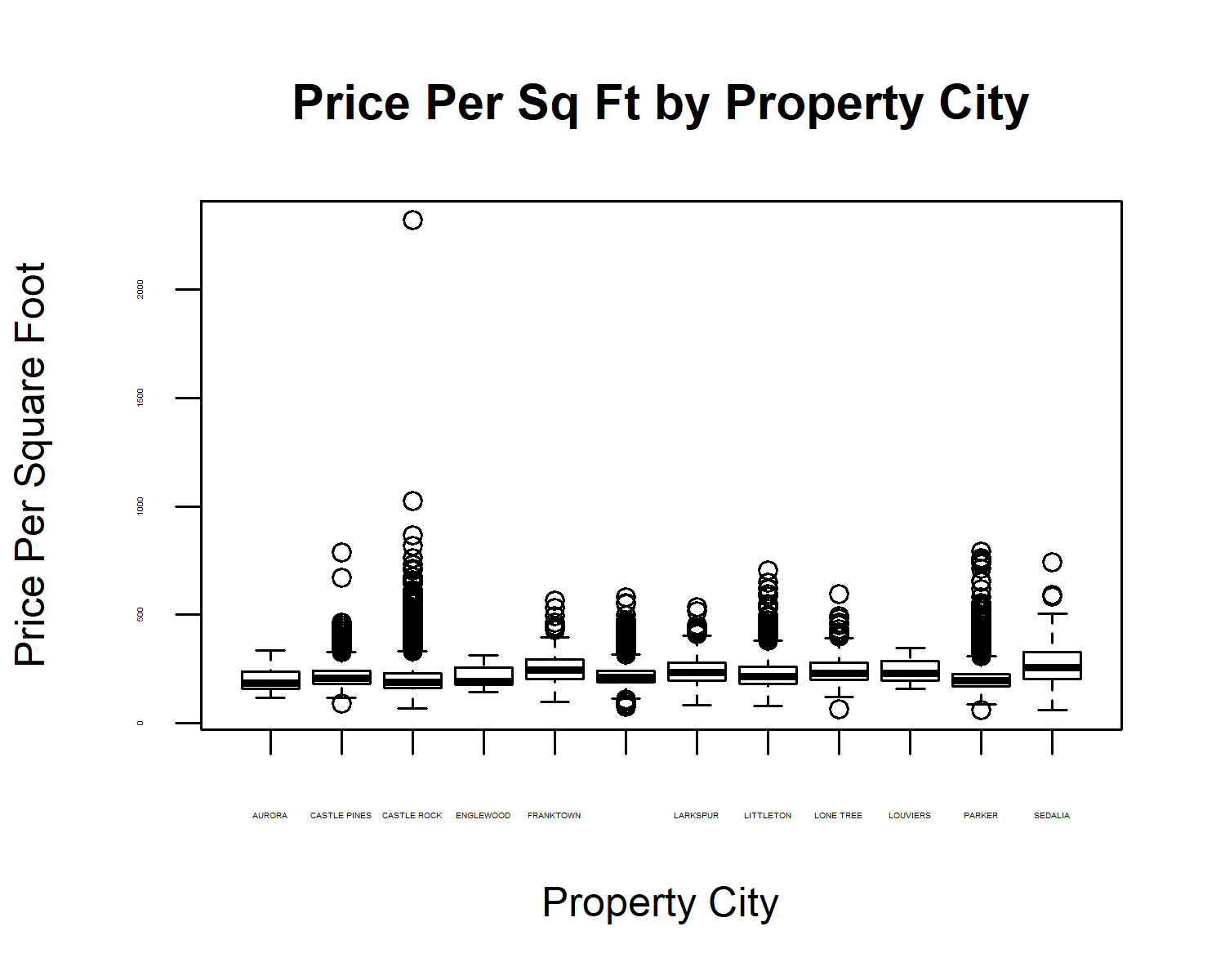


Figure 13: Property City to Price Per Square Foot

The final boxplot graph compares home quality to price per square foot as shown in Figure 14. I was curious if there would be a noticeable trend for increased price as the home quality increased. There was one property that showed a home quality zero. I am not sure if that was intentional or by accident but I left it in the data frame for evaluation. There were only 25 sales with a one home quality so that is probably not a good indicator of price compared to the other quality levels as the price difference may be because the low-quality houses are multiple acres which can be very expensive. The quality levels of interest are 2-5. Levels 2 and 3 appear to be similar in range and price but when evaluating levels 4 and 5 it is obvious that better quality homes sell for a higher price per square foot.

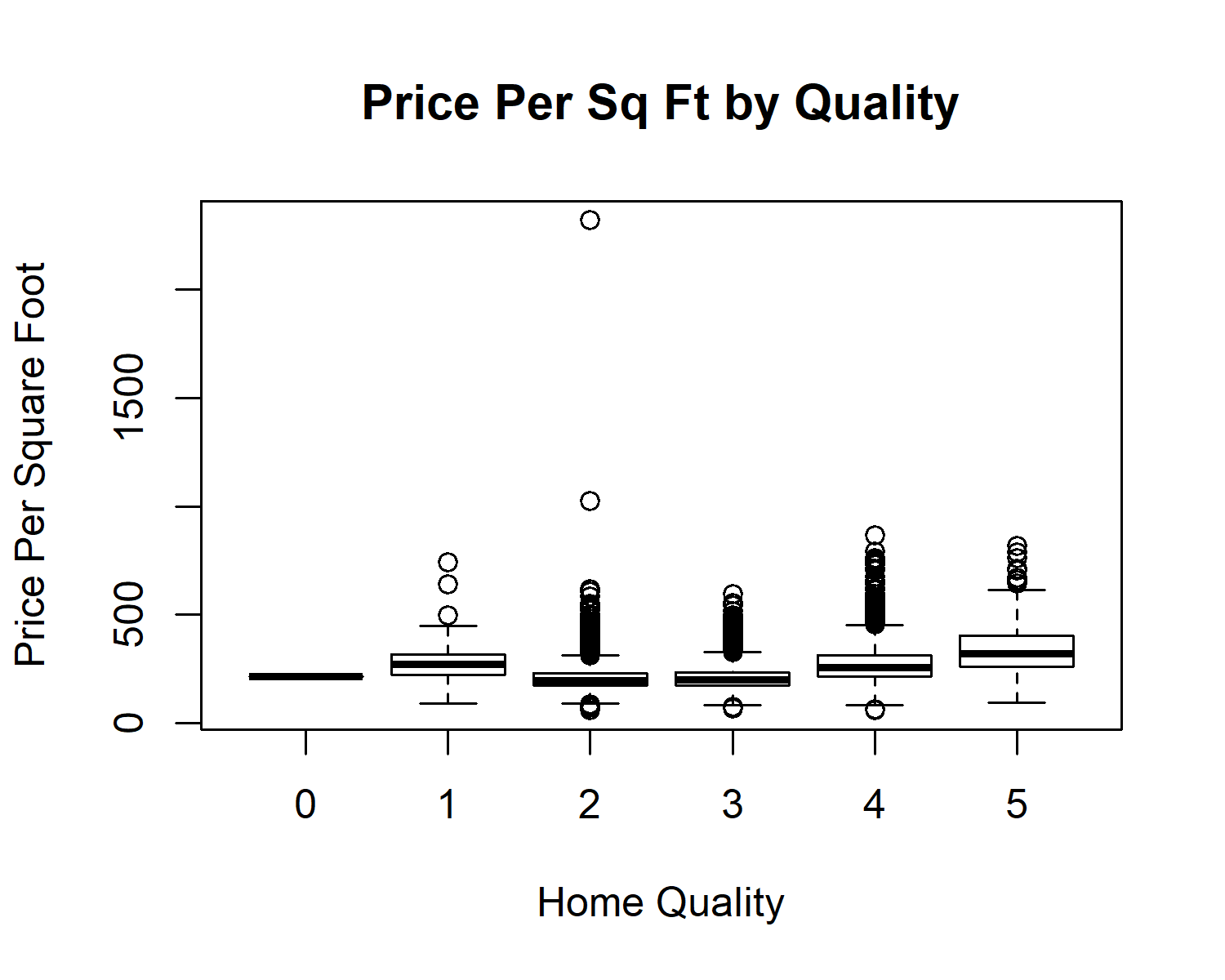


Figure 14: Home Quality to Price Per Square Foot

**Regression Models**

I ran a couple of tests on the entire data model to get some perspective on the best approach for testing with the data frame. I used set.seed(123) so I could repeat the testing results. I initially debated whether leaving several of the independent variables as numeric or converting them to factor would give better results so I tried both ways to compare results with a linear model. I got an R-squared value of 0.7706 and an Adjusted R-squared of 0.7705 for all independent variables as numbers. I then converted 8 of the 15 independent variables to factors and ran the same model and I got an R-squared of 0.8226 and an Adjusted R-squared of 0.8223. This convinced me to use the data frame with the 8 factors instead of numbers for all the models. One interesting observation is that some of the factor variables created showed p-values greater than 0.05 indicating they are not statistically significant. Some of the results were zip codes which had very few sales so volume was the issue but when the values were not converted to factor and left as a number then the p-value was extremely small for all variables except Home Style which had a p-value equal to 0.13 indicating it is not significant to the model and could have been removed from the model. I also tested a normalized version of the data frame with factors to see if the results were the same as the data frame which did not have normalized data and the R-squared and Adjusted R-squared were the same so I opted not to use the normalized version of the file for my models.

I split the data frame into train and test data frames for training and predicting at an 80/20 ratio. I created five different models using Linear Regression, Random Forest, Gradient Boosting, Extreme Gradient Boosting, and Support Vector Machine utilizing assignments from prior classes as a guide to create the models. Note that I used Support Vector Machine out of curiosity to see how it would perform but it is a great duel classifier and it was expected to perform poorly compared to other models. I tried using Cross Validation with Support Vector Machine but it hung when executing so I used the svm() function instead. I ultimately used Cross Validation with the Random Forest, Gradient Boosting, and Extreme Gradient Boosting models. I trained the 5 models using the training data frame and then I used the testing data frame to test the accuracy of the trained model. I used the t.test() function to perform the t-test on the trained models for the three Cross Validation models. The small p-value from the t.test() function supports that the statistical tests are significant. I used the function cor() to calculate the prediction percent for the models from the predictions on the test data frame. I also used the regr.eval() function on the predictions from the test data frame but I only looked at mape as it was a better indicator of error since I did not normalize the data frame making the mae, mse, and rmse values were large and I felt the mape values were easier to evaluate and compare since they are independent of the unit of measurement and results for all models were less than one. Mape stands for **mean absolute percentage error** and it is a measure of prediction accuracy of a forecasting method. Mape doesn’t do well when it approaches zero but since I removed the few rows less than $150,000.00 due to the questionable validity of them being sales record (they were rolling into trusts or other activities instead of sales) then all values were $150,000.00 or greater so mape worked well. Mape doesn’t work with negative values and there were no negative values so using mape was a great choice. I noticed that since I had over 28,000 rows of data that there didn’t appear to be any signs of overfitting of the data for the five models I ran.

The first model was a Linear Regression model. Below is the code used to create, train, and test the accuracy of the model. Please refer to the R Markdown file for execution output for the various commands. There was an R-squared value of 0.8322 and an Adjusted R-squared values of 0.8318 from the trained model. The accuracy of the model on the test data frame was 88.5% and the mape value was 9.9%. The testing results to actuals are graphed as shown below in Figure 15. The graph shows some consistency but tends to show less consistency in predictions at the higher home sale prices.

**modelLm <- lm(`Sale Price` ~ ., data = dcSalesTrain)**

**summary (modelLm)**

**modelRf**

**t.test(dcSalesTrain$"Sale Price", modelRf$predictions)**

**modelLmPred <- predict(modelLm, newdata = dcSalesTest)**

**modelLmPredActuals <- data.frame(cbind(actuals=dcSalesTest$`Sale Price`, preds=modelLmPred)) # create modelLmPredActuals dataframe.**

**corrAccLm <- cor(modelLmPredActuals) # 88.5% Accuracy**

**corrAccLm**

**regr.eval(modelLmPredActuals$actuals, modelLmPredActuals$preds)**

**head(modelLmPredActuals)**

**plot(modelLmPredAct,col="blue")**

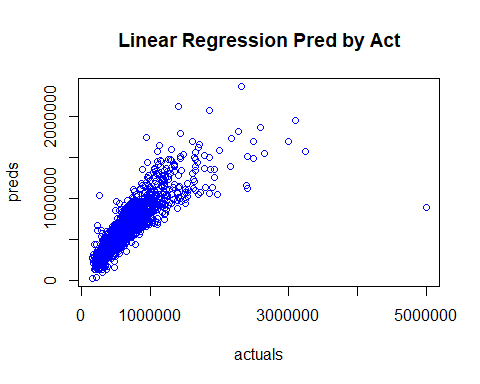


Figure 15: Linear Regression Actuals to Predictions

The second model was a Random Forest model. Below is the code used to create, train, and test the accuracy of the model. Note that the trainControl() function statement included in this model R code below is used by models three and four as well. The accuracy of the model on the test data frame was 91% and the mape value was 0.08905. The testing results to actuals are graphed as shown below in Figure 16. The graph shows a little better consistency including the higher price home sales except for a few of the highest priced sales. The lower price home sales also show better consistency and more even distribution.

**myControl = trainControl(method = "cv", number = 5, verboseIter = FALSE)**

**modelRf = train(`Sale Price` ~ ., data = dcSalesTrain, tuneLength = 1, method = "ranger", importance = 'impurity', trControl = myControl)**

**summary(modelRf)**

**modelRfPred <- predict(modelRf, newdata = dcSalesTest)**

**modelRfPredAct <- data.frame(cbind(actuals=dcSalesTest$`Sale Price`, preds=modelRfPred)) # create modelRfPredAct dataframe.**

**corrAccRf <- cor(modelRfPredAct) # 91.0% Accuracy**

**corrAccRf**

**regr.eval(modelRfPredAct$actuals, modelRfPredAct$preds)**

**head(modelRfPredAct)**

**plot(modelRfPredAct,col="blue")**

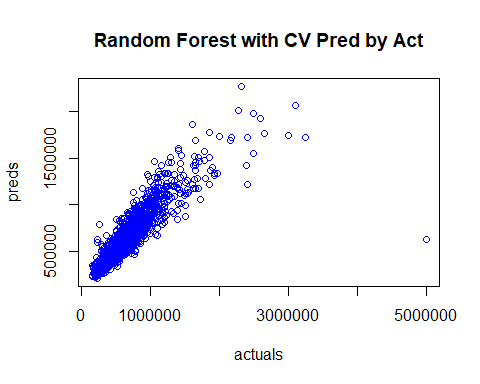


Figure 16: Random Forest with CV Actuals to Predictions

The third model was a Gradient Boosting model. Below is the code used to create, train, and test the accuracy of the model. The accuracy of the model on the test data frame was 88.9% and the mape value was 11.29%. The testing results to actuals are graphed as shown below in Figure 17. It is interesting to compare Figure 15 with Figure 17. The points from Figure 15 seem to create a slight upwards arch and the group of points from Figure 17 seem to create a slight downwards arch but both have a similar accuracy with the mape being slightly better for the Linear Regression model.

**modelSgb = train(`Sale Price` ~ ., data = dcSalesTrain, tuneLength = 2, method = "gbm", trControl = myControl)**

**summary(modelSgb)**

**modelSgbPred <- predict(modelSgb, newdata = dcSalesTest)**

**modelSgbPredAct <- data.frame(cbind(actuals=dcSalesTest$`Sale Price`, preds=modelSgbPred)) # create modelSgbPredAct dataframe.**

**corrAccSgb <- cor(modelSgbPredAct) # 88.9% Accuracy**

**corrAccSgb**

**regr.eval(modelSgbPredAct$actuals, modelSgbPredAct$preds)**

**head(modelSgbPredAct)**

**plot(modelSgbPredAct,col="blue")**

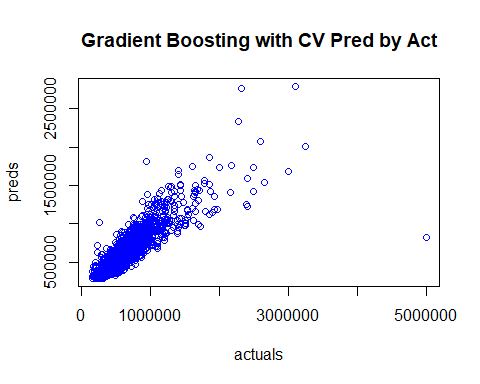


Figure 17: Gradient Boosting with CV Actuals to Predictions

The fourth model was an Extreme Gradient Boosting with Cross Validation model. Below is the code used to create, train, and test the accuracy of the model. The accuracy of the model on the test data frame was 92.2% and the mape value was 7.45%. The testing results to actuals are graphed as shown below in Figure 18. This graph shows more of a similarity to the Random Forest model as shown in Figure 16 and it only has a few outlier points but in general it is rather consistent. The accuracy and mape value is the best so far with 7.45%

**xgbTuneGrid = expand.grid(nrounds = c(50, 100), lambda = seq(0.1, 0.5, 0.1), alpha = seq(0.1, 0.5, 0.1), eta = c(0.3, 0.4))**

**modelXgb = train(`Sale Price` ~ ., data = dcSalesTrain, tuneLength = 3, method = "xgbLinear", trControl = myControl, tunegrid = xgbTuneGrid)**

**summary(modelXgb)**

**modelXgbPred <- predict(modelXgb, newdata = dcSalesTest)**

**modelXgbPredAct <- data.frame(cbind(actuals=dcSalesTest$`Sale Price`, preds=modelXgbPred)) # create modelXgbPredAct dataframe.**

**corrAccXgb <- cor(modelXgbPredAct) # 92.2% Accuracy**

**corrAccXgb**

**regr.eval(modelXgbPredAct$actuals, modelXgbPredAct$preds)**

**head(modelXgbPredAct)**

**plot(modelXgbPredAct,col="blue")**

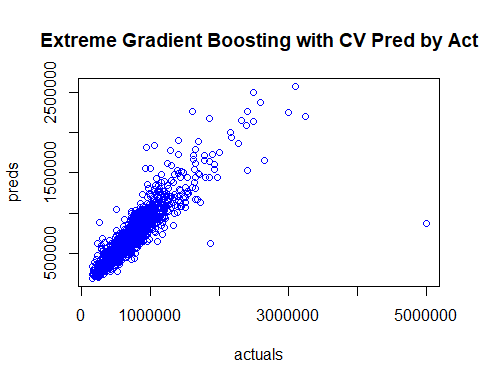


Figure 18: Extreme Gradient Boosting with CV Actuals to Predictions

The fifth, and final, model was a Support Vector Machine model. Following is the code used to create, train, and test the accuracy of the model. The accuracy of the model on the test data frame was 80.8% and the mape value was 14.65%. The testing results to actuals are graphed as shown below in Figure 19. This model shows a larger arch upwards than Figure 16 above confirming the reason for the 80.8% predicting accuracy which is the lowest accuracy rate of all the models and the worst mape value of 14.65%.

**modelSvm = svm(`Sale Price`~., data = dcSalesTrain, kernel="linear", scale = FALSE)**

**summary(modelSvm)**

**modelSvmPred <- predict(modelSvm, newdata = dcSalesTest)**

**modelSvmPredAct <- data.frame(cbind(actuals=dcSalesTest$`Sale Price`, preds=modelSvmPred)) # create modelSvmPredAct dataframe.**

**corrAccSvm <- cor(modelSvmPredAct) # 80.8% Accuracy**

**corrAccSvm**

**regr.eval(modelSvmPredAct$actuals, modelSvmPredAct$preds)**

**head(modelSvmPredAct)**

**plot(modelSvmPredAct,col="blue")**

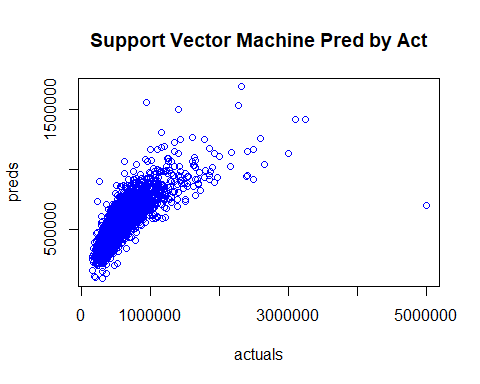


Figure 19: Support Vector Machine with CV Actuals to Predictions

Below is a table detailing the predictive match rate and the mape value. Extreme Gradient Boosting, which had the best match rate of 92.20% and the best mape value of 7.45%, is highlighted for easy identification. The RSME rates were smaller for the models with higher accuracy and lower for models with worse accuracy with Extreme Gradient Boosting RMSE being the lower and best rate and Support Vector Machine being the highest and worse rate. It helps to see the statistics from the 5 models to compare them.

|  |  |  |
| --- | --- | --- |
| **Regression Model** | **Match Rate** | **mape** |
| **Extreme Gradient Boosting** | **92.20%** | **7.45%** |
| Random Forest | 91.00% | 8.91% |
| Gradient Boosting | 88.91% | 11.29% |
| Linear Regression | 88.52% | 9.90% |
| Support Vector Machine | 80.78% | 14.65% |

Table 2: Regression Model Evaluation

**Possible Future Projects**

The 92.2% match rate was great but the model could be more accurate. A future project could be to expand on this project and find a way to increase home sales price approximation. One modification would be to remove the homes with higher sales price from the model. There are only a few hundred higher priced home sales which could be removed and that may make the model predict more accurately although it was interesting seeing the results of leaving those homes in the model. The trees models could be expanded to try to improve accuracy. More effort can be included to tune the models even further by modifying the number of folds and number of repeats for those models. I performed several models but there are a lot more options for parameters which could be made. Another change could be to remove properties over a determined acreage size from the model as this can drastically increase the sale price but it may be that removing the very high-priced home sales would resolve this issue as well. The assessor’s office home price estimator did not show consistency and was low in most cases so it was not used for this project. For a future project there could be research comparing these values to the home sales prices to see if there is a pattern to the approximations where each assessor value could be multiplied by a factor which is calculated by possible identified features and therefore better approximates the home sale prices. This suggested project will require a lot of research and evaluation but it would be great to find a model which works even better than the Extreme Gradient Boosting model from this project.

**Conclusion**

It was challenging at times to work through fixing bad data and identifying the values for converting to a number scale. It took a lot of effort evaluating, cleaning, and combining the data. It also took a lot of effort to figure out how to create the map and get the key from Bing to extract coordinates from each home location. I feel both processes were very important and produced better visuals and smoother processing. It was interesting comparing the results from the various types of models and seeing how they performed on the data frames. Decision trees class of regression models appear to perform better on home sales prices with the types of features used to classify and predict the prices. The Extreme Gradient Boosting model was the decision tree model which performed best. Extreme Gradient Boosting is a modified version of Gradient Boosting which more accurately approximates the best tree learner plus it trains fast. The models performed well and could be a useful tool for approximating home sale prices in Douglas County, Colorado which is the goal of this project.

References

Douglas County, Colorado Assessor’s Office. Advanced Search and Reports. Retrieved on 11/01/2019 from <https://apps.douglas.co.us/assessor/advanced-search/>

University of Iowa, College of Liberal Arts and Sciences. Maps and Geographical Data. Statistics & Actuarial Science. Retrieved on 11/29/2019 from <https://homepage.divms.uiowa.edu/~luke/classes/STAT4580/maps.html>

United States Census Bureau. State and county district maps. Retrieved on 11/30/2019 from <https://www2.census.gov/geo/tiger/GENZ2010/>

Microsoft. Bing Maps Dev Center. Retrieved on 11/25/2019 from <https://www.microsoft.com/en-us/maps/create-a-bing-maps-key>

Microsoft. Bing Maps. Retrieved on 11/25/2019 from <https://www.microsoft.com/en-us/maps/create-a-bing-maps-key>