Author: Rebeca Ansar
Date: May 5, 2021

Predicting List Prices of Homes in Round Rock, Texas

Research Question and Motivation

Round Rock is a burgeoning city in the greater Austin, Texas area. According to the US Census Bureau, it is also one of the fastest-growing cities in the United States with a population growth of 33.5% in the last 10 years.[2] Round Rock is expected to continue seeing an influx of new residents. In addition to Dell Headquarters being based in the city, other major technology companies have solidified plans to build offices in the area. Thus, workers in this sector will continue to move into Round Rock from other states and Texas cities.

This analysis aims to identify the best predictors of a home's list price in this city. Thus, the research question is: What predictors, if any, are significantly associated with the list price of a home in Round Rock, Texas?

The motivation for this analysis is a desire to educate both homeowners and homebuyers about the greater Austin housing market. This information can be particularly helpful to a seller if they are considering home renovations before listing their property. They can prioritize adding features that are associated with a higher list price. Homebuyers can also utilize this analysis to decide their offer on a house by comparing the listed price of the home to the price predicted by the chosen model.

Hypotheses

The first hypothesis is that from the list of 11 home features in consideration, the number of bedrooms, the number of bathrooms, the square footage of the home, the acreage of the land, and the age of the house will be predictive of the home list price. These features are commonly discussed among homebuyers to fit their various needs.

The second hypothesis is that the features heating, cooling, and the number of days the home remains on the market will not be significant predictors of the home list price. Heating and cooling options are common home features in Texas, and all homes in the dataset have at least one of each of these features. Additionally, the Greater Austin area is presently a seller's market, so it is not expected that home listings will vary much in how long they stay on the market.

Methods

Data Sourcing

As a homebuyer, I have access to new home listings in the Greater Austin area through the RealtyAustin website.[1] RealtyAustin, one of the biggest real-estate brokerages in central Texas, has 12 home features listed on the first page of each listing that the buyer sees. Thus, I was confident that this data source could be utilized to address my research question.

I selected the 100 first-available listings on the RealtyAustin website in one sitting with one restriction: the listing must be located in Round Rock, Texas. I saved the 100 selected listings as PDF files to keep a record of the source data. Next, I transferred the data for each listing to an Excel spreadsheet. Missing values were left blank in the spreadsheet. One feature, called 'Parking' was not included in the data transfer because it was highly unstructured and it repeated data captured in the 'Garages' feature. Thus, 11 features were included for each home listing in the final spreadsheet.

Data Cleaning

Originally, the real estate dataset contained 100 observations. There were 5 rows with missing data. 2 of these rows were commercial listings, which were deleted because such listings were outside of the research question's scope. The remaining 3 rows were residential listings. I chose to remove those listings rather than impute the missing values because this was a simple solution that still allowed a retention of 95 observations, which is not significantly lower than the original number of observations.

I ensured that out of the 11 features, two features, namely property type and stories were converted to factor variables, each with two levels. *Property_Type* had the levels 'Home' and 'Condo'. *Stories* had the levels '1' and '2'. I ensured that the remaining 9 features were of numeric data type. Lastly, I converted the variable *Year_Built* to *House_Age* to improve the interpretability of that predictor within a regression model. This was done by subtracting the values in *Year_Built* from 2021 as this is the present year.

The 11 home features used as potential predictors in the model_building phase of this analysis are included in Table 1.

Feature Name	Description
Bedrooms	Number of bedrooms in the residence
Bathrooms	Number of bathrooms in the residence
Square_Feet	The square footage
Acres	The acreage of the residence
Property_Type	Whether the residence is a house or a condo
Heating	Number of heating options in the residence
Stories	Number of stories in the residence
Cooling	Number of cooling options in the residence
Days_on_Market	Number of days the residence has been listed for sale
Garages	Number of garages
House_Age	The current age of the home in years

Table 1: The 11 features used in the model-building phase

Exploratory Analysis

I plotted histograms to assess the distribution of my response variable, home list price, for both the entire dataset and the training data. Both histograms were right-skewed, as seen in Figure 1. I contemplated log transformation of the response variable to normalize the distribution. While this does mitigate the distribution's skewness, it also makes the model less translatable. Further, both the entire dataset and the training set used to build models are larger than the cutoff of 30 observations. The full dataset has 95 observations and the training data consists of 54 observations. Thus, normality of the response is not necessary to build my regression models. For these reasons, I decided to move forward without transforming the data.

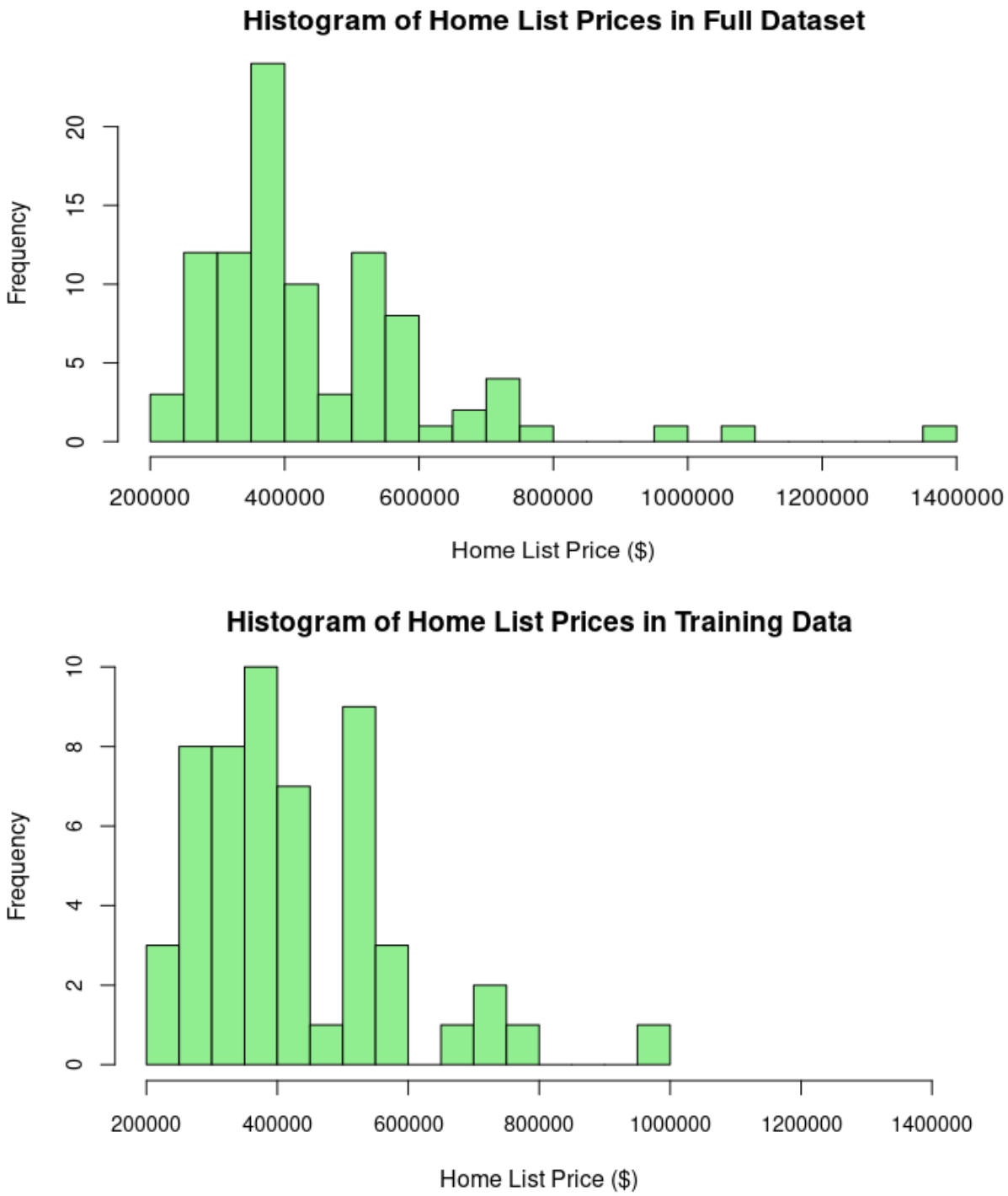


Figure 1: Histograms of Home List Prices in the entire dataset and the training data

Model Building

The dataset was partitioned with 60% of the data being allocated to the training set and 40% being allocated to the testing set. Thus, the train data consisted of 54 observations and the test data consisted of 41 observations. A seed was used to ensure reproducibility of results. All model building utilized the training data.

The first multiple linear regression model, model_1, was constructed with all 11 features as predictors of home list price. The Adjusted R-squared for this model was 0.8753, and 6 of the 11 predictors were statistically significant at the $p < 0.05$ level.

Next, the stepAIC function was utilized to refine model_1. This resulting model, named aic_model_1, retained only those 7 predictors that were statistically significant at the $p < 0.01$ level in model_1. All 7 predictors were statistically significant at the $p < 0.05$ level in aic_model_1. The adjusted R-squared value for this model was slightly higher than that of model_1 at 0.8838.

The next model, model_2, was also a direct modification of model_1. This model did not utilize the stepAIC function. Instead, model_2 included only those 6 predictors from model_1 that were statistically significant at the $p < 0.05$ level. This model had a lower R-squared of 0.8762 when compared to aic_model_1. Additionally, two of the predictors, *Bedrooms* and *Stories*, were statistically significant at the $p < 0.05$ level in model_1 but lost their statistical significance at this level in model_2.

At this point, I had created three multiple linear regression models. Next, I evaluated the variance inflation factors for the predictors in each of the three models. There were no concerning VIF values across all models, with all VIF values being less than 6.

Although there were no concerning variance inflation factor values, I still evaluated multicollinearity and the statistical significance of pairwise correlation coefficients for the predictors to assess the need for further model refinement. The reasoning behind this is that generally the number of bedrooms and bathrooms in a residence is connected to its square footage. Since these predictors were included in all three regression models thus far, the goal was to evaluate if removing one or more of them could lead to a improved model.

The following observations were made based on the rule that multicollinearity may pose a serious issue when pairwise correlations have an absolute value greater than 0.5. The correlation chart demonstrated that there was a high collinearity between the number of bedrooms and the number of bathrooms, the number of bedrooms and square footage, and the number of bathrooms and square footage. Square footage had the highest correlation of 0.85 with home list price when compared to the other two features, whose correlations with the response were 0.54 and 0.58 respectively, Similarly, acreage of a home and days on market of the listing were also high correlated predictors. Acreage had a correlation of 0.61 with the home list price while days on market had a correlation of 0.51. Lastly, the number of garages and the age of the house were also highly correlated, with the number of garages showing a

higher correlation of 0.42 with home list price while the age of the house showed a very small correlation with the predictor.

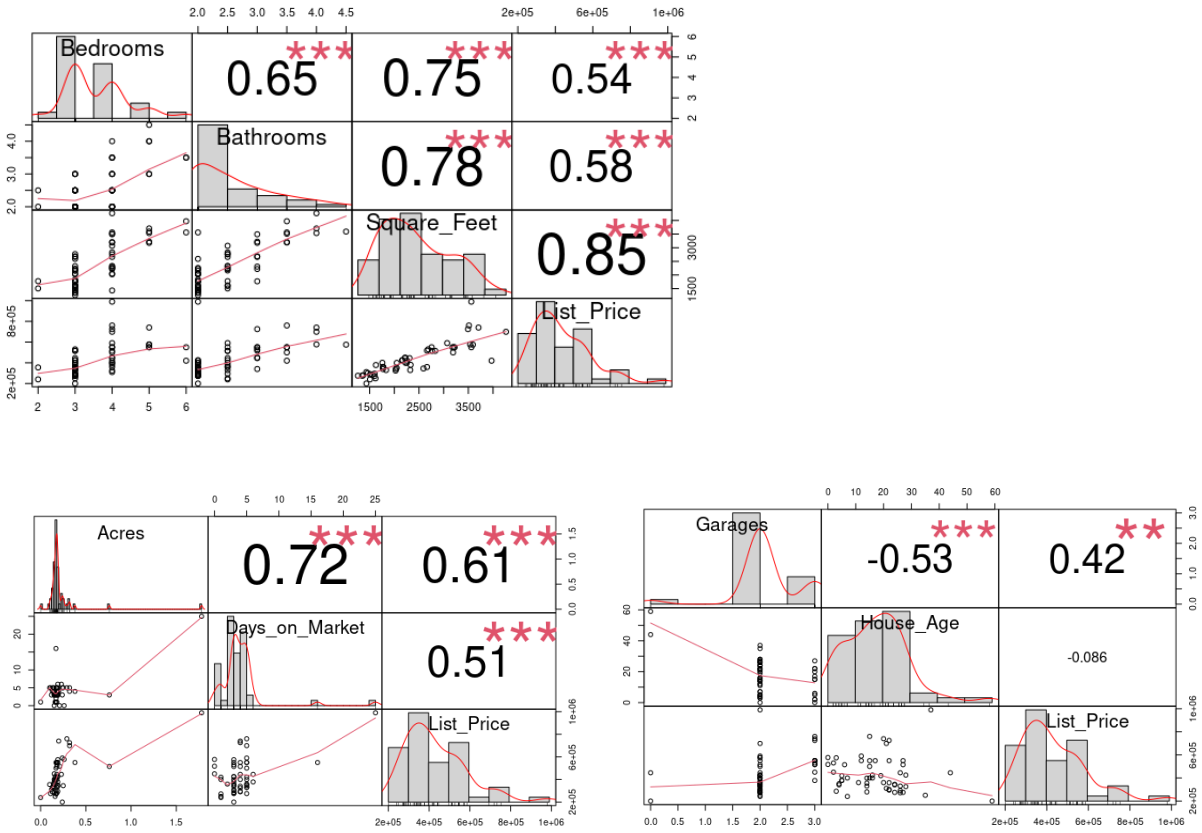


Figure 2: Correlation charts showing correlations among predictor sets

I decided to modify aic_model_1 based on these observations. I selected this model because it had the highest adjusted R-squared value of all three prior models. I eliminated the following predictors from the model: bedrooms, bathrooms, and days on market, keeping only those predictors that showed the highest correlation with the home list price. The age of the house had already been excluded in aic_model_1. This modified model was named model_3, and it had an adjusted R-squared value of 0.8612. Only one predictor, *Stories*, was not statistically significant at the $p < 0.05$ level.

The final multiple regression model was built by applying the stepAIC function to model_3. This model, titled aic_model_3, had a slightly higher adjusted R-squared value of 0.8625, and it included three features, all of which were statistically significant at the $p < 0.05$ level.

This component of model building resulted in 5 multiple linear regression models.

A random forest model, titled forest, was built for the second component of model building to evaluate if another type of model could improve upon the results of the multiple linear regressions. The random forest model resulted in a training R-squared value of 0.6777.

Assumptions

Since model_3 had the highest R-squared for the test data (discussed below), it was selected as the final multiple linear regression model. Thus, I ensured that the four assumptions of linear regression are satisfied for model_3.

It is reasonable to assume that the observations are independent of each other because they were collected from independent home listings. Next, I observed the Residuals vs Fitted plot in order to assess if the assumption of linearity holds. The plot shows a roughly constant band of points with no apparent pattern across the predicted values. However, a few points do appear to stand out and there is visible clumping of points. Overall, the assumption of linearity is reasonably satisfied, though further investigation is warranted.

To assess homoscedasticity, I observed the Scale-Location plot. This plot did not show significant variation in the spread of points around the red line and across the predicted fitted values. The red line does not significantly diverge from a horizontal line, but there is room for improvement. There are a few points that stand out in this plot as well as some visible clumping. Here too, though the assumption of homoscedasticity is overall satisfied, further investigation is warranted.

Finally, I observed the Normal Q-Q plot to assess normality. While there is divergence from the straight line at both the top and bottom, the residuals show overall adherence to the line. Thus, the assumption of normality is also satisfied.

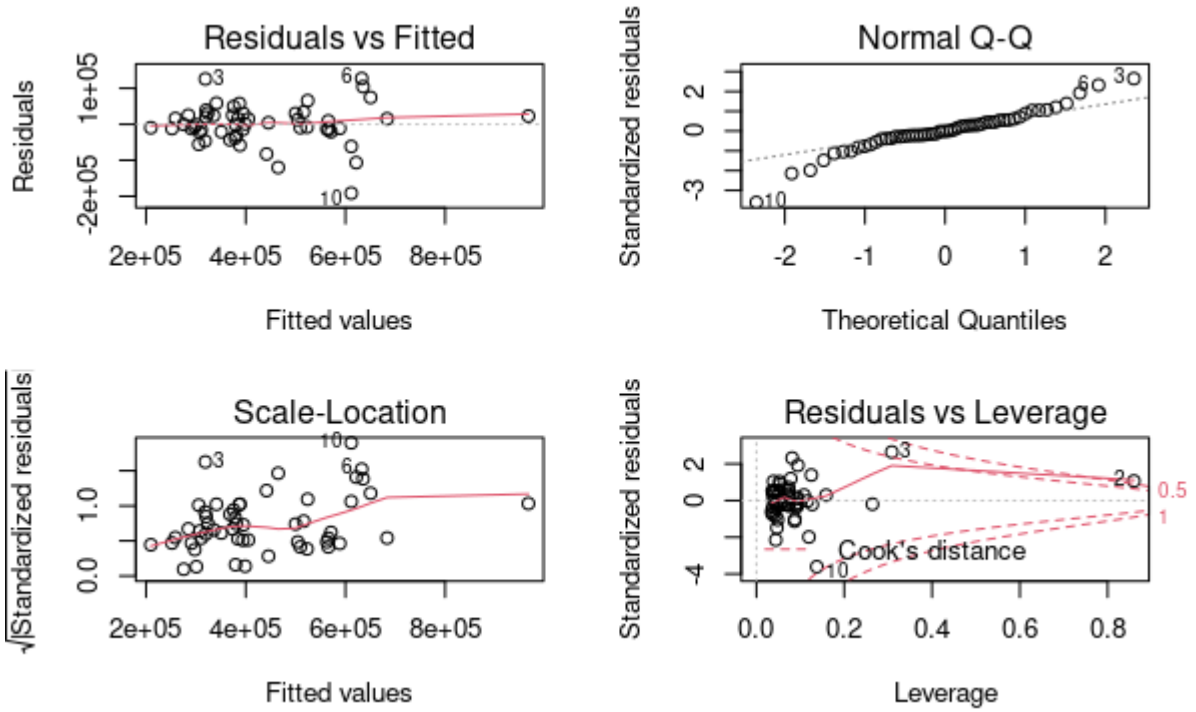


Figure 3: Diagnostics plots to assess assumptions for model_3

Though these three assumptions were roughly satisfied, the Residuals vs Leverage diagnostics plot did indicate influential points. Points 2, 3, and 10 were marked as influential on the plot. I removed those points and re-evaluated the diagnostics plots to evaluate if the influential points were responsible for the divergences from normality and homoscedasticity.

As expected, the removal of the influential points resulted in all three plots, Residuals vs Fitted, Normal Q-Q, and Scale-Location, conforming more closely with the assumptions of linearity, normality, and homoscedasticity, respectively. The Residuals vs Fitted plot showed more pattern-less spread and less clumping. The Normal Q-Q plot showed much more adherence to the straight line at the bottom and top. Finally, the Scale-Location plot's red line was more horizontal, and the points showed less variation in their spread.

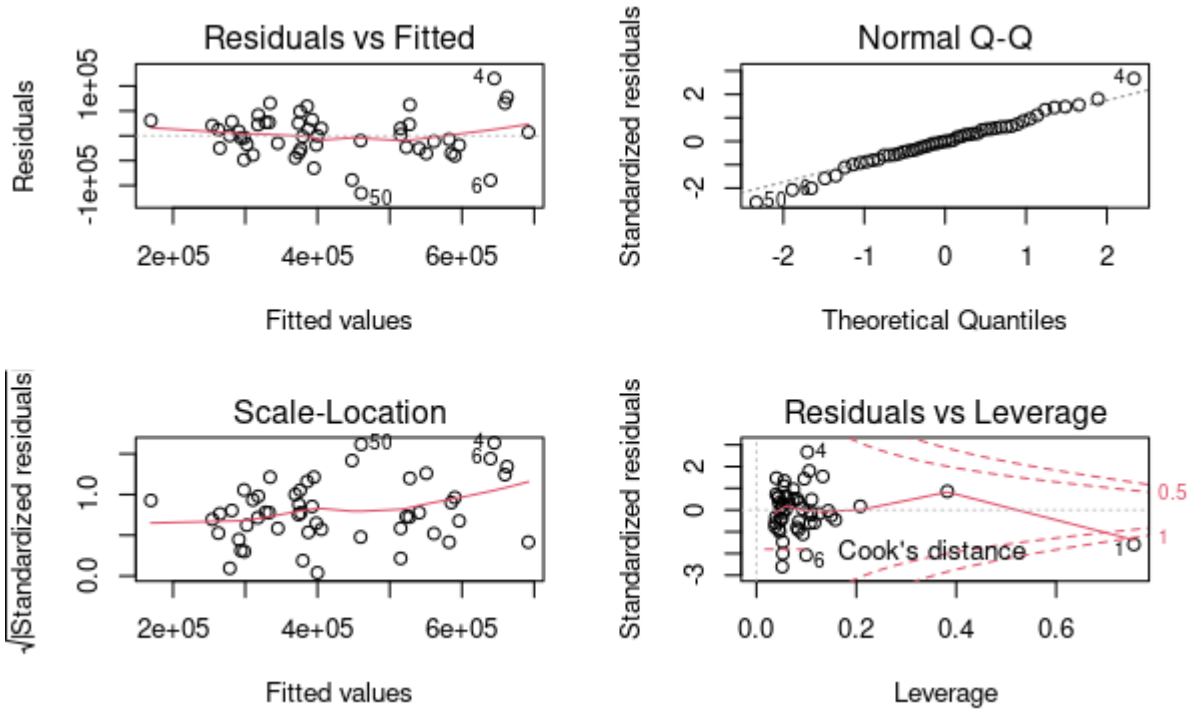


Figure 4: Diagnostics plots for model_3 without influential points 2, 3, and 10

The influential points did facilitate understanding as to why there was some divergence in the initial diagnostics plots, but they were retained in the final model to preserve the veracity of the dataset.

Results and Conclusions

Final Model

The table below shows the results of this analysis. The table includes model names, predictors included in each model, multiple R-squared values, adjusted R-squared values, and lastly, the R-squared values derived from the correlation between the model predictions and the home list prices in the test data.

Model name	Predictors included	Multiple R-squared	Adjusted R-squared	Test R-squared
model_1	All 11 features	0.9012	0.8753	0.8242
aic_model_1	*Bedrooms, Bathrooms, Square_Feet, Acres, Stories, Days_on_Market, Garages*	0.8992	0.8838	0.8288
model_2	*Bedrooms, Bathrooms, Square_Feet, Acres, Stories, Garages*	0.8902	0.8762	0.8357
model_3	*Square_Feet, Acres, Stories, Garages*	0.8717	0.8612	0.8420
aic_model_3	*Square_Feet, Acres, Garages*	0.8703	0.8625	0.8334
forest	*Square_Feet, Acres, Bathrooms, Bedrooms, Garages,* and *Days_on_Market* have variable importance greater than 10	% Variance explained: 67.77 % proportion is 0.6777	-	0.6107

Table 2: Results from the model building phase of the analysis

Performance on the test data was used to select the best model, and the model with the highest test R-squared was the multiple linear regression model, model_3. The row for this model is highlighted in green in the table above.

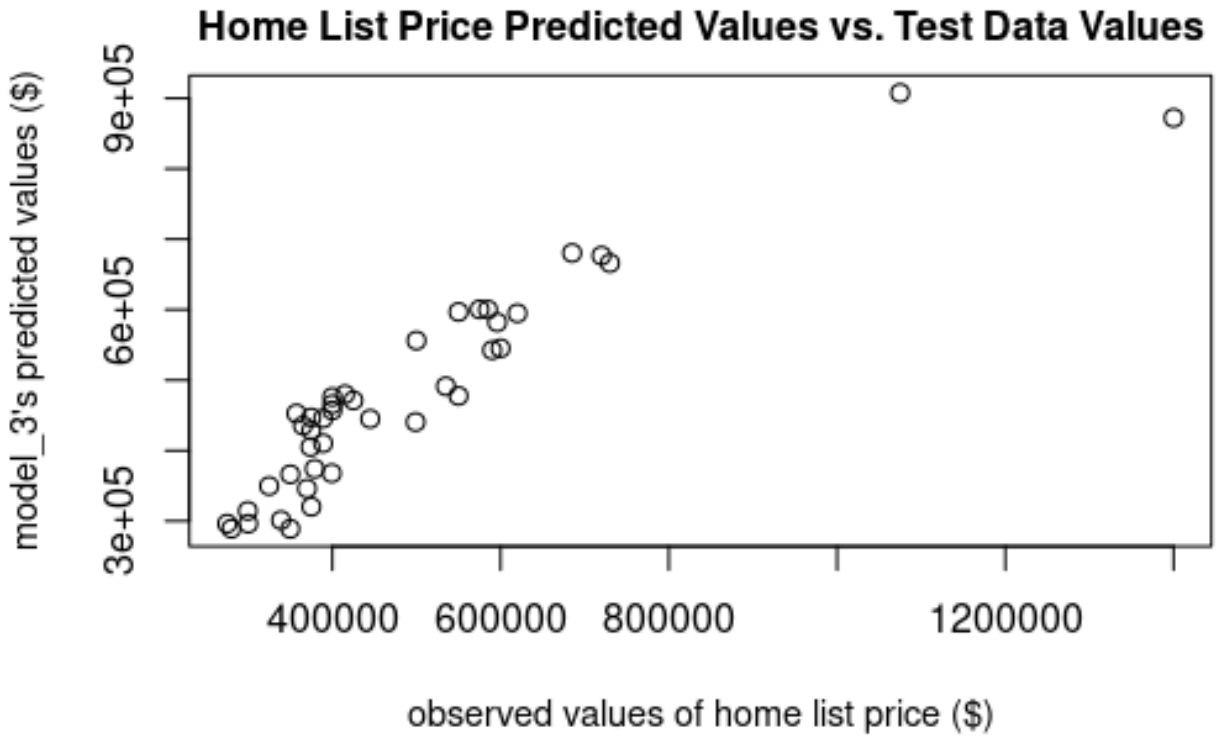


Figure 5: A scatter plot illustrating the R-squared between model_3 predictions and test data

Below is the summary for the final model, model_3, for the predictors of home list price in the city of Round Rock, Texas for 2021. This model boasts a high R-squared value of 0.8420 on the test data, meaning that the model explains 84.2% of the variance in the test data.

```
Call:
lm(formula = List_Price ~ Square_Feet + Acres + Stories +
Garages,
    data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-191713  -19752    -730   25142  127193

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -33034.15   33040.29  -1.000 0.322312
Square_Feet    127.93      12.44  10.281 7.96e-14 ***
Acres       246933.33   38042.00   6.491 4.08e-08 ***
Stories2    -12826.77   17837.20  -0.719 0.475494
Garages      52552.93   14312.31   3.672 0.000595 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 57280 on 49 degrees of freedom
Multiple R-squared:  0.8717,  Adjusted R-squared:  0.8612
F-statistic: 83.21 on 4 and 49 DF,  p-value: < 2.2e-16
```

The multiple linear regression equation for model_3 is:

$$\widehat{List\_Price} = -33034.15 + 127.93 * Square\_Feet + 246933.33 * Acres - 12826.77 * Stories2 + 52552.93 * Garages$$

3 of the 4 predictors in the final model are statistically significant at the $p < 0.05$ level. These are: *Square_Feet, Acres*, and *Garages*. *Stories* is not statistically significant, with a relatively high p-value.

Assessing the Hypotheses

The number of bedrooms and bathrooms and the age of the house were not included as statistically significant predictors of home list price in the final model. The square footage of the home and the acreage of land were both statistically significant predictors at the $p < 0.05$ level in the final model. Thus, we reject the null hypothesis that *Square_Feet* and *Acres* are statistically insignificant at the $p < 0.05$ level. The first hypothesis is partially correct.

The features *Heating, Cooling,* and *Days_on_Market* were not statistically significant predictors in the final model. In fact, Heating and Cooling were eliminated as predictors early in the model building phase. The predictor *Days_on_Market* was also excluded from the final model because it showed strong multicollinearity with *Acres*, with the latter being more strongly correlated with the response variable. Thus, the second hypothesis is correct.

Furthermore, although the random forest model did not have a competitive test R-squared value, it did provide support for the variables deemed to be statistically significant predictors of home list price in the final model. The variable importance plot below shows that *Square_Feet* is the most significant predictor in the random forest model, followed by *Acres*. The predictor *Garages* also makes it in the top 5 most important predictors.

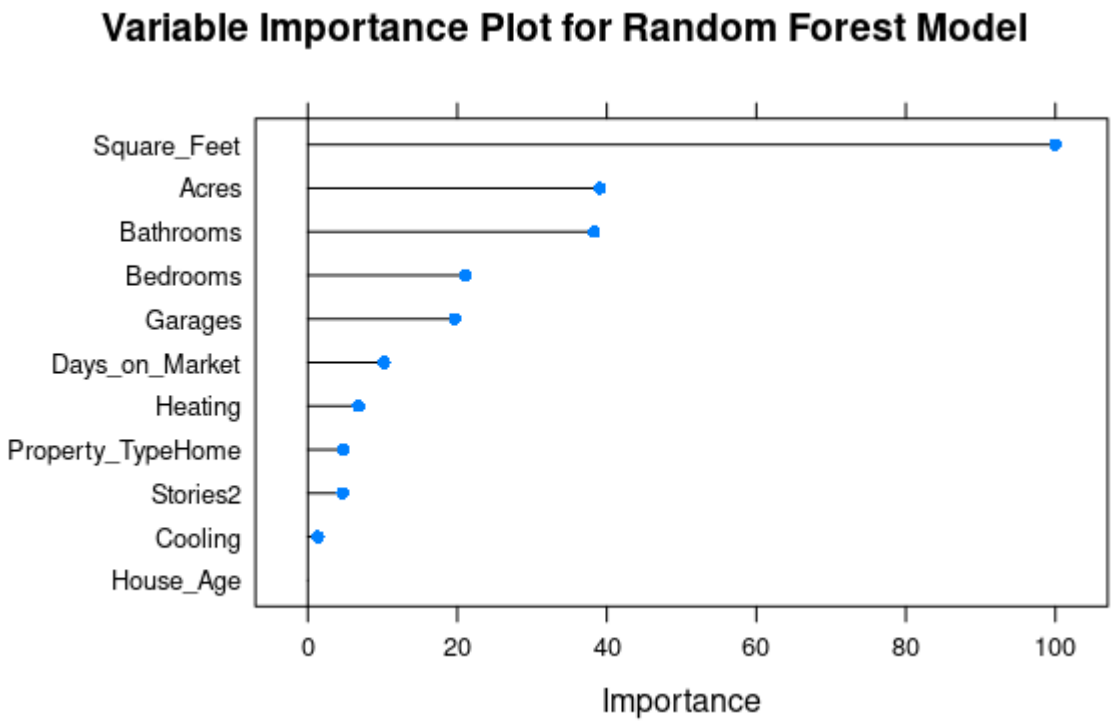


Figure 6: Variable importance plot showing the significant predictors of home list price

Ultimately, this analysis can provide direction to real-estate buyers and sellers. Buyers can utilize the final model equation to calculate a predicted home list price based on home features and compare it with the actual list price to help them make a decision on the amount to offer. Further, the analysis highlights the features that contribute to a higher home list price. Thus,

sellers could use the results of this analysis to understand and maximize the most marketable home features. For example, they could increase the square footage of their home in order to increase its list price.

Limitations and Challenges

The data for this analysis was available on individual webpages of the RealtyAustin website. Because of this, I had to carefully manually transfer the data onto an Excel spreadsheet from 100 listing pages. Of course, the time required to use this type of data source poses a limitation on how many independent observations can be collected in a reasonable amount of time. This in turn limits how big a sample size can be, which could have ramifications on how well the model performs in a real-world setting. Additionally, this approach increases the chances of human error.

A further limitation of this analysis was that only the 11 features available on the listing's first page were collected while there is a much larger pool of home features that could attract buyers or be factored into a home list price by sellers.

There were also a few challenges. After the spreadsheet was loaded into RStudio Cloud, I began to see two issues with the data frame that was created. Firstly, most of the inherently numerical columns were auto-assigned character data type by R. Secondly, functions to exclude incomplete cases were not successful on each re-run. After investigating this issue, I realized that both problems would be solved if instead of marking missing values as 'NA' in my original spreadsheet, I left the cells blank. As expected, the resulting data frame did correctly identify column data types and correctly excluded incomplete cases on each run. Finally, only 3 of the 95 observations were condos while the other 92 were homes. This disbalance may be somewhat alleviated with a bigger dataset, but it is likely that more sophisticated modeling tools would be required to truly address the issue.

Further Discussion and Research

This analysis can certainly be expanded by using web scraping to automate the process of moving data from the RealtyAustin web pages directly into a data frame. There would be several benefits to this approach:

- Elimination of the possibility of human error in transferring data from the web page onto a spreadsheet.
- The ability to create a much bigger sample size, leading to more training and test data to build and assess the models.
- The ability to collect more types of data than just the 11 features on the first page.

Collectively, this approach could result in a model that has a higher coefficient of determination with the test data, which would indicate better potential performance on real-world data. Further, automating data reformatting in this way would allow analysts to repurpose the analysis for other cities in the Greater Austin area to compare and contrast important predictors of home list price for different cities in the region.

References

[1] *RealtyAustin*. www.realtyaustin.com. Accessed 30 March 2021.

[2] Buchanan, Taylor. "Round Rock among fastest-growing big cities in the nation, new census data shows." *Community Impact Newspaper*, 20 May 2020. https://communityimpact.com/austin/round-rock-pflugerville-hutto/data-reference/2020/05/20/round-rock-among-fastest-growing-big-cities-in-the-nation-new-census-data-shows/.

Code Appendix

```
library(PerformanceAnalytics)
library(olsrr)
library(MASS)
library(car)
library(psych)
library(dplyr)
library("readxl")
library(mlbench)
library(caret)
library(e1071)
library(lime)
```

Data Cleaning

```
#store excel data in data frame
mydata <- read_excel("stat109_realestate_data.xlsx")
```

```
#evaluate data frame
str(mydata)
```

```
## tibble [100 x 13] (S3: tbl_df/tbl/data.frame)
##  $ Address       : chr [1:100] "4 Ridge Run, Round Rock, TX" "17 Stillmeadow, Round Rock, TX" "118 Ja
##  $ Bedrooms      : num [1:100] 4 4 4 NA 4 3 4 NA 5 3 ...
##  $ Bathrooms     : num [1:100] 4 2.5 2 NA 3 2 2 NA 3 2.5 ...
##  $ Square_Feet   : num [1:100] 3518 2829 3560 4000 2308 ...
##  $ Acres         : num [1:100] 0.48 0.76 1.78 1.62 0.23 0.13 0.21 0.31 0.17 0.14 ...
##  $ Property_Type : chr [1:100] "Home" "Home" "Home" "Commercial" ...
##  $ Heating       : num [1:100] 3 1 2 5 2 1 2 2 1 2 ...
##  $ Year_Built    : num [1:100] 1995 1994 1984 1984 1977 ...
##  $ Stories       : num [1:100] 2 1 1 NA 1 1 1 1 2 2 ...
##  $ Cooling       : num [1:100] 3 1 3 3 1 1 1 1 1 3 ...
##  $ Days_on_Market: num [1:100] 7 3 25 4 6 5 2 5 16 5 ...
##  $ Garages       : num [1:100] 3 2 2 NA 0 2 2 2 3 2 ...
##  $ List_Price    : num [1:100] 685000 515000 989900 925000 445000 ...
```

```
#evaluate data frame
head(mydata, 5)
```

```
## # A tibble: 5 x 13
##   Address  Bedrooms Bathrooms Square_Feet Acres Property_Type Heating Year_Built
##   <chr>       <dbl>     <dbl>       <dbl> <dbl> <chr>           <dbl>      <dbl>
## 1 4 Ridge~        4       4          3518  0.48 Home                3       1995
## 2 17 Stil~        4       2.5        2829  0.76 Home                1       1994
## 3 118 Jac~        4       2          3560  1.78 Home                2       1984
## 4 123 Jac~       NA      NA          4000  1.62 Commercial          5       1984
## 5 303 Dov~        4       3          2308  0.23 Home                2       1977
## # ... with 5 more variables: Stories <dbl>, Cooling <dbl>,
## #   Days_on_Market <dbl>, Garages <dbl>, List_Price <dbl>
```

```
#list cases with NAs
missing_values <- mydata[!complete.cases(mydata), ]
head(missing_values, 10)
```

```
## # A tibble: 5 x 13
##   Address  Bedrooms Bathrooms Square_Feet Acres Property_Type Heating Year_Built
```

```
##   <chr>       <dbl>     <dbl>       <dbl> <dbl> <chr>           <dbl>      <dbl>
## 1 123 Jac~       NA        NA        4000  1.62 Commercial          5       1984
## 2 601 Mis~       NA        NA          NA  0.31 Multi-Family        2       1982
## 3 1001 Ch~       NA        NA          NA  0.16 Multi-Family        2       1997
## 4 3001 Jo~       NA        NA        4980  0.12 Commercial         NA       2019
## 5 5813 Mi~        5         3        2690 NA    Home                1       2017
## # ... with 5 more variables: Stories <dbl>, Cooling <dbl>,
## #   Days_on_Market <dbl>, Garages <dbl>, List_Price <dbl>
```

```
#number of rows with incomplete data
print(paste('Number of rows with missing data:', nrow(mydata[!complete.cases(mydata), ])))
```

```
## [1] "Number of rows with missing data: 5"
```

```
#check how many commercial properties in dataset
print(paste('Number of Commercial properties in dataset:',
            nrow(mydata[mydata$Property_Type == 'Commercial', ])))
```

```
## [1] "Number of Commercial properties in dataset: 2"
```

```
#check how many multi-family homes in dataset
print(paste('Number of Multi-Family homes in dataset:',
            nrow(mydata[mydata$Property_Type == 'Multi-Family', ])))
```

```
## [1] "Number of Multi-Family homes in dataset: 2"
```

```
nrow(mydata)
```

```
## [1] 100
```

```
#only keep complete cases, which also removes 'Commercial' and #'Multi-family' property type
df <- mydata[complete.cases(mydata), ]

#check that 95 observations remain
nrow(df)
```

```
## [1] 95
```

```
#change feature variables with categorical values to factor data type
df$Property_Type <- as.factor(df$Property_Type)
df$Stories <- as.factor(df$Stories)

#use House_Age predictor instead of Year_Built
df$Year_Built <- as.numeric(df$Year_Built)
df$House_Age <- 2021 - df$Year_Built

#check that Property_Type is only 'Home' or 'Condo'
unique(df$Property_Type)
```

```
## [1] Home  Condo
## Levels: Condo Home
```

```
#check that Stories is only 1 or 2
unique(df$Stories)
```

```
## [1] 2 1
## Levels: 1 2
```

```
#drop address column
df <- subset(df, select = -Address)
```

```
df <- subset(df, select = -Year_Built)

#ensure that all variables have correct type
str(df)
```

```
## tibble [95 x 12] (S3: tbl_df/tbl/data.frame)
##  $ Bedrooms      : num [1:95] 4 4 4 4 3 4 5 3 4 3 ...
##  $ Bathrooms     : num [1:95] 4 2.5 2 3 2 2 3 2.5 3.5 2.5 ...
##  $ Square_Feet   : num [1:95] 3518 2829 3560 2308 1540 ...
##  $ Acres         : num [1:95] 0.48 0.76 1.78 0.23 0.13 0.21 0.17 0.14 0.29 0.13 ...
##  $ Property_Type : Factor w/ 2 levels "Condo","Home": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Heating       : num [1:95] 3 1 2 2 1 2 1 2 2 1 ...
##  $ Stories       : Factor w/ 2 levels "1","2": 2 1 1 1 1 1 2 2 2 2 ...
##  $ Cooling       : num [1:95] 3 1 3 1 1 1 1 3 3 1 ...
##  $ Days_on_Market: num [1:95] 7 3 25 6 5 2 16 5 5 5 ...
##  $ Garages       : num [1:95] 3 2 2 0 2 2 3 2 3 2 ...
##  $ List_Price    : num [1:95] 685000 515000 989900 445000 339000 ...
##  $ House_Age     : num [1:95] 26 27 37 44 17 42 15 11 15 26 ...
```

```
#check number of rows and columns in data frame
print(paste("row number:", nrow(df), "and", "column number:", ncol(df)))
```

```
## [1] "row number: 95 and column number: 12"
```

```
#Check if predictor Stories is balanced
print(paste("Number of 1-story homes in df:", nrow(df[df$Stories == 1,])))
```

```
## [1] "Number of 1-story homes in df: 49"
```

```
print(paste("Number of 2-story homes in df:", nrow(df[df$Stories == 2,])))
```

```
## [1] "Number of 2-story homes in df: 46"
```

```
#Check if predictor Property_Type is balanced
print(paste("Number of homes in df:", nrow(df[df$Property_Type == 'Home',])))
```

```
## [1] "Number of homes in df: 92"
```

```
print(paste("Number of condos in df:", nrow(df[df$Property_Type == 'Condo',])))
```

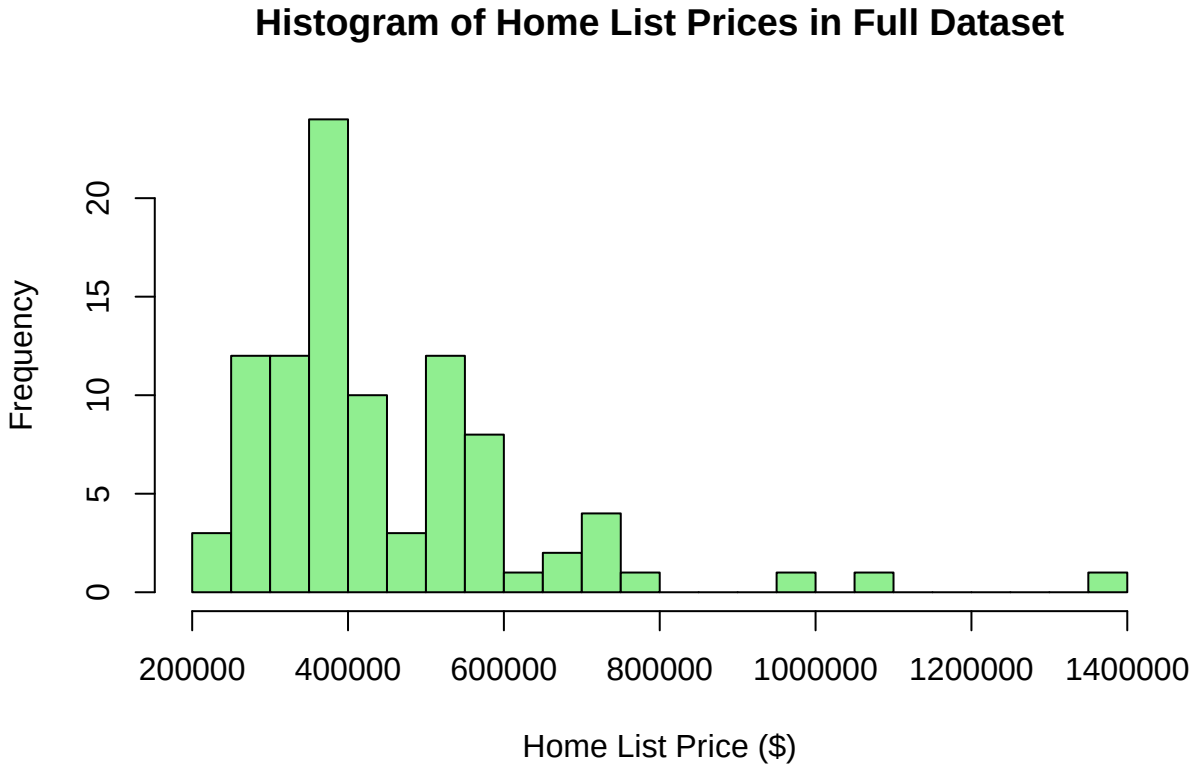
```
## [1] "Number of condos in df: 3"
```

```
print(paste("Number of property types that are neither home nor condo in df:",
            nrow(df[df$Property_Type != 'Home'& df$Property_Type !=
                          'Condo',])))
```

```
## [1] "Number of property types that are neither home nor condo in df: 0"
```

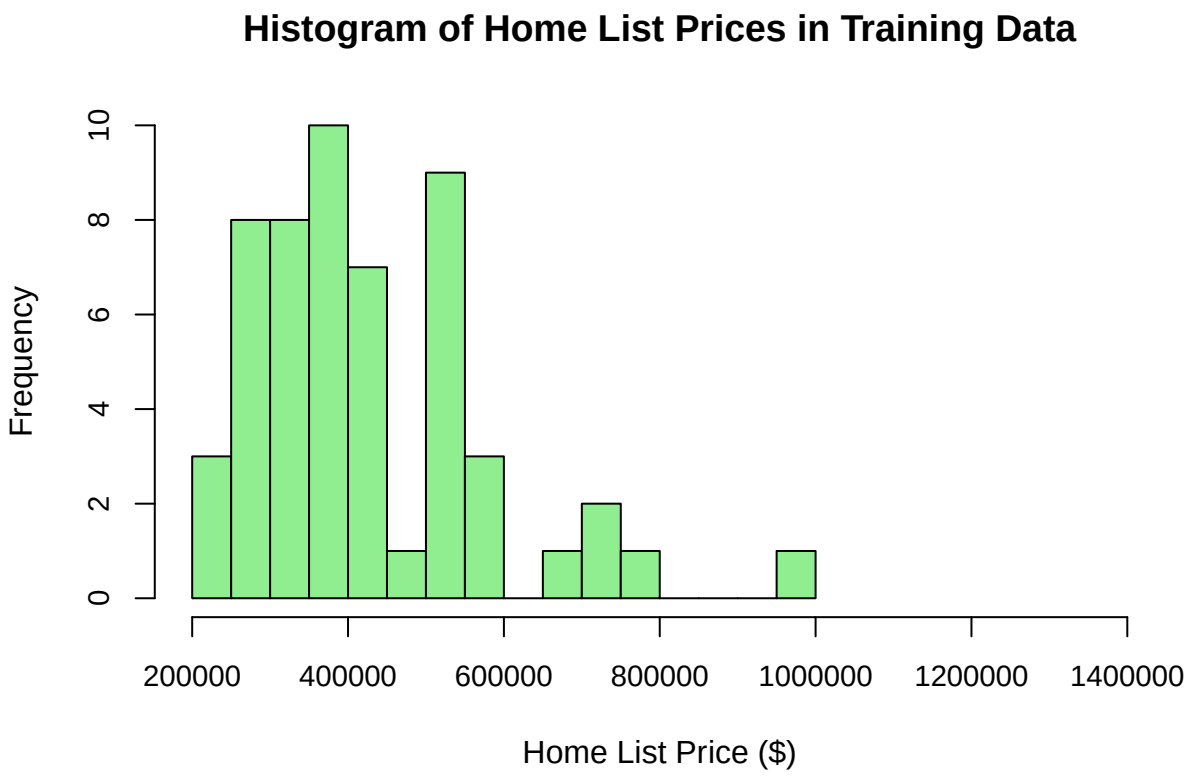
Data Visualization

```
#check if home list price is normally distributed in df
hist(df$List_Price,
     breaks = 20,
     main = 'Histogram of Home List Prices in Full Dataset',
     xlab = 'Home List Price ($)',
     col = 'lightgreen')
```




```r
#partition data set into 60% training and 40% testing
set.seed(222)
ind <- sample(2, nrow(df), replace = T, prob = c(0.6, 0.4))
train <- df[ind == 1, ]
test <- df[ind == 2, ]
```

```r
#check if home list price is normally distributed in train
hist(train$List_Price,
     breaks = 20,
     main = 'Histogram of Home List Prices in Training Data',
     xlab = 'Home List Price ($)',
     xlim = c(200000, 1400000),
     col = 'lightgreen',
     cex.axis = 0.9)
```



Model Building

```
#Create first linear model with all 11 predictors
model_1 <- lm(List_Price ~ ., train)
summary(model_1)
```

```
##
## Call:
## lm(formula = List_Price ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -139034  -29612  -10761   31108  102734
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -114678.42   68023.49  -1.686 0.099238 .
## Bedrooms           -32857.44   14375.72  -2.286 0.027389 *
## Bathrooms           75791.82   26460.47   2.864 0.006497 **
## Square_Feet           105.05      23.04   4.559 4.39e-05 ***
## Acres              205687.04   53582.52   3.839 0.000410 ***
## Property_TypeHome   27525.69   35544.92   0.774 0.443037
## Heating              3571.85   13286.98   0.269 0.789382
## Stories2           -52151.45   21375.53  -2.440 0.018999 *
## Cooling              1375.92    7807.59   0.176 0.860961
## Days_on_Market       6224.77    3226.43   1.929 0.060467 .
## Garages             62605.35   16974.91   3.688 0.000643 ***
## House_Age             256.33     852.57   0.301 0.765157
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 54290 on 42 degrees of freedom
## Multiple R-squared:  0.9012, Adjusted R-squared:  0.8753
## F-statistic: 34.83 on 11 and 42 DF,  p-value: < 2.2e-16
```

```
#use stepAIC function to improve model_1
aic_model_1 <- stepAIC(model_1, direction = 'both', trace = F)
summary(aic_model_1)
```

```
##
## Call:
## lm(formula = List_Price ~ Bedrooms + Bathrooms + Square_Feet +
##     Acres + Stories + Days_on_Market + Garages, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -140689  -30814  -11209   28375  110280
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -71750.56   46098.53  -1.556 0.126453
## Bedrooms       -28919.79   13065.74  -2.213 0.031869 *
## Bathrooms       70427.16   24600.95   2.863 0.006303 **
## Square_Feet       107.12      21.91   4.889 1.27e-05 ***
## Acres          210786.31   50575.40   4.168 0.000134 ***
## Stories2       -48881.00   19757.62  -2.474 0.017110 *
## Days_on_Market   6179.45    3060.12   2.019 0.049300 *
## Garages         57071.59   13333.31   4.280 9.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52410 on 46 degrees of freedom
## Multiple R-squared:  0.8992, Adjusted R-squared:  0.8838
## F-statistic:  58.6 on 7 and 46 DF,  p-value: < 2.2e-16
```

```
#remove predictors not significant at the p < 0.05 level in model_1
model_2 <- lm(List_Price ~ . -Property_Type - Heating - Cooling
              -Days_on_Market -House_Age, train)
summary(model_2)
```

```
##
## Call:
## lm(formula = List_Price ~ . - Property_Type - Heating - Cooling -
##     Days_on_Market - House_Age, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -144069  -24848   -8748   34061  110096
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -53826.65   46693.57  -1.153 0.254837
## Bedrooms    -25455.95   13370.02  -1.904 0.063048 .
## Bathrooms    55765.56   24262.53   2.298 0.026032 *
## Square_Feet    112.83      22.42   5.031 7.56e-06 ***
## Acres       275885.37   40225.02   6.859 1.34e-08 ***
```

```
## Stories2    -34771.21   19076.34  -1.823 0.074708 .
## Garages      56263.94   13756.76   4.090 0.000168 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54100 on 47 degrees of freedom
## Multiple R-squared:  0.8902, Adjusted R-squared:  0.8762
## F-statistic: 63.52 on 6 and 47 DF,  p-value: < 2.2e-16
```

```
#check VIF for model_1
vif(model_1)
```

```
##       Bedrooms      Bathrooms    Square_Feet          Acres  Property_Type
##       2.737469       5.512714       5.997373       2.896338       1.214691
##        Heating        Stories        Cooling Days_on_Market        Garages
##       1.248028       2.067215       1.351394       2.545148       1.839619
##      House_Age
##       1.736051
```

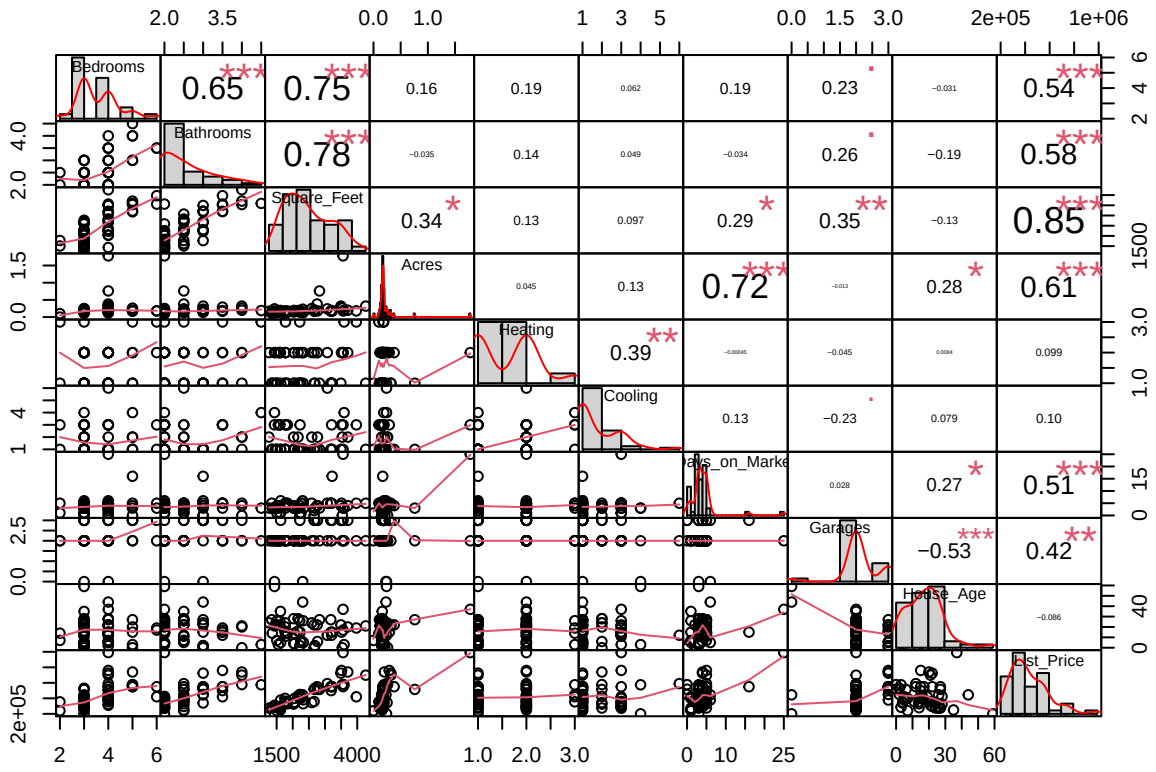
```
#check VIF for aic_model_1
vif(aic_model_1)
```

```
##       Bedrooms      Bathrooms    Square_Feet          Acres        Stories
##       2.426417       5.113071       5.817990       2.768786       1.895085
## Days_on_Market        Garages
##       2.456709       1.217859
```

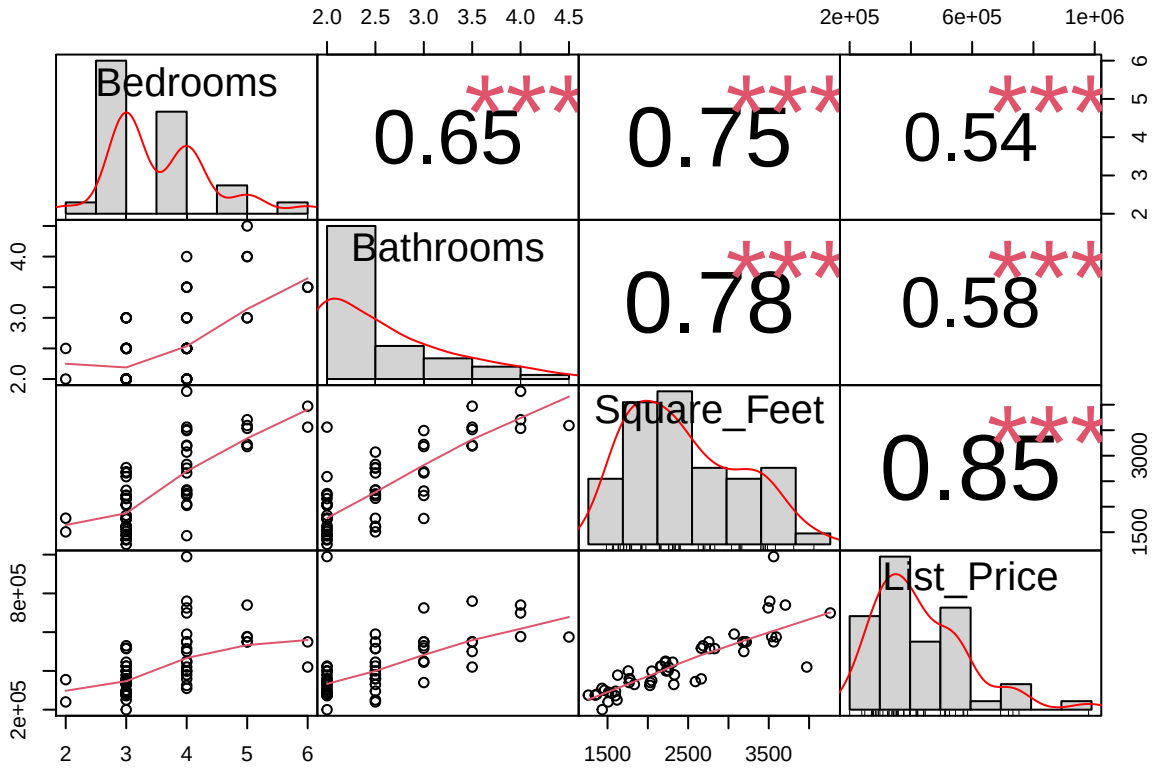
```
#check VIF for model_2
vif(model_2)
```

```
##    Bedrooms   Bathrooms Square_Feet       Acres     Stories     Garages
##    2.384596    4.667704    5.721081    1.643826    1.658069    1.216763
```

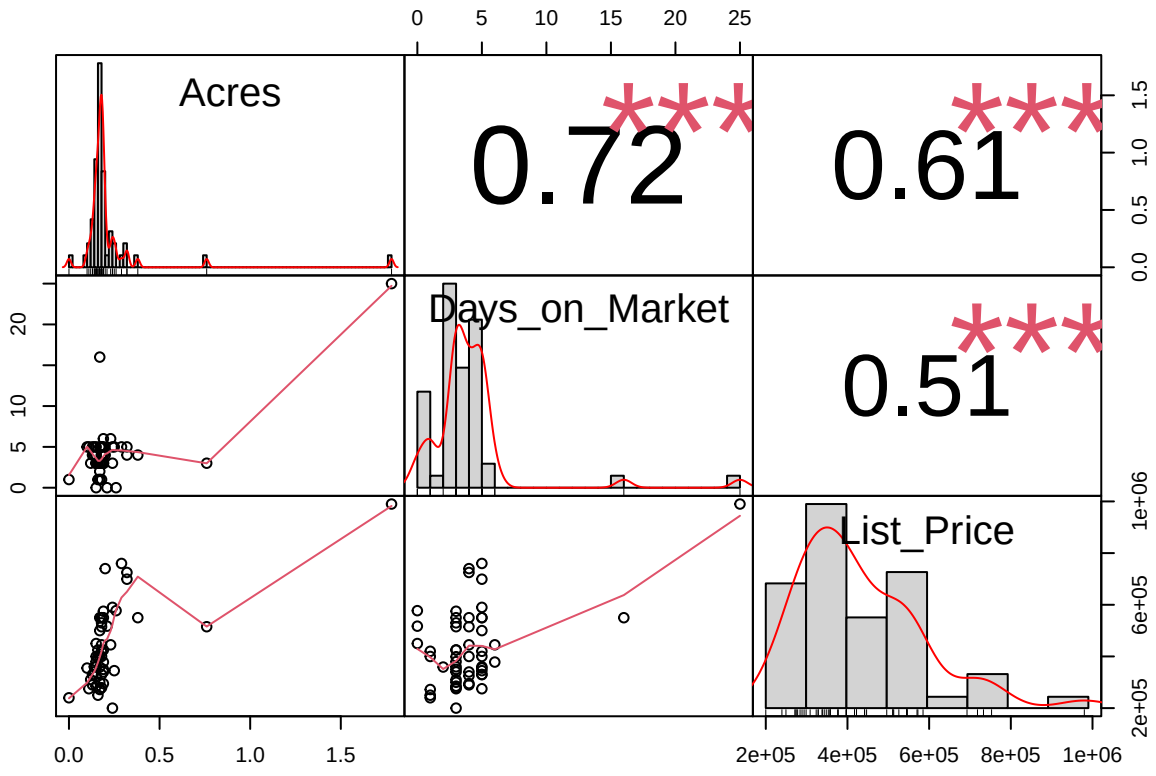
```
#check multi-collinearity
chart.Correlation(train %>% select(Bedrooms, Bathrooms, Square_Feet,
                                   Acres, Heating, Cooling, Days_on_Market,
                                   Garages, House_Age, List_Price))
```



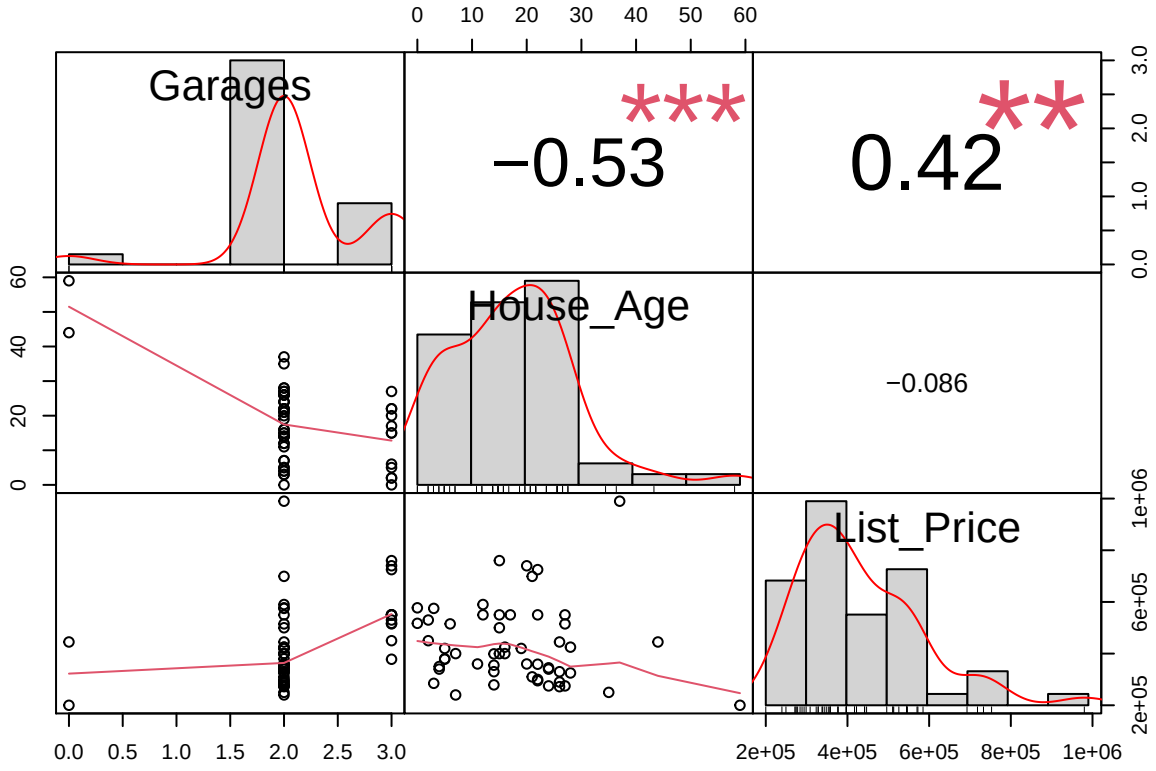

```
#multi-collinearity charts for predictors showing high collinearity
chart.Correlation(train %>% select(Bedrooms, Bathrooms, Square_Feet, List_Price))
```




```
chart.Correlation(train %>% select(Acres, Days_on_Market, List_Price))
```




```
chart.Correlation(train %>% select(Garages, House_Age, List_Price))
```




```
#Based on the correlation chart, I will drop
#bedrooms, bathrooms, and days on market to create model_3
model_3 <- lm(List_Price ~ Square_Feet + Acres + Stories + Garages, train)
```

```
summary(model_3)
```

```
##
## Call:
## lm(formula = List_Price ~ Square_Feet + Acres + Stories + Garages,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -191713  -19752    -730   25142  127193
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -33034.15   33040.29  -1.000 0.322312
## Square_Feet    127.93      12.44  10.281 7.96e-14 ***
## Acres       246933.33   38042.00   6.491 4.08e-08 ***
## Stories2    -12826.77   17837.20  -0.719 0.475494
## Garages      52552.93   14312.31   3.672 0.000595 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 57280 on 49 degrees of freedom
## Multiple R-squared:  0.8717, Adjusted R-squared:  0.8612
## F-statistic: 83.21 on 4 and 49 DF,  p-value: < 2.2e-16
```

Final model equation:

$$\widehat{List\_Price} = -33034.15 + 127.93 * Square\_Feet + 246933.33 * Acres - 12826.77 * Stories2 + 52552.93 * Garages$$

```
#use stepAIC on model_3 to create next model
aic_model_3 <- stepAIC(model_3, direction = 'both', trace = F)
summary(aic_model_3)
```

```
##
## Call:
## lm(formula = List_Price ~ Square_Feet + Acres + Garages, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -192914  -20101    2202   28335  129977
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -31014.60   32761.35  -0.947 0.348354
## Square_Feet    124.40      11.38  10.935 7.29e-15 ***
## Acres       256215.63   35611.87   7.195 2.98e-09 ***
## Garages      51914.75   14215.62   3.652 0.000623 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 57000 on 50 degrees of freedom
## Multiple R-squared:  0.8703, Adjusted R-squared:  0.8625
## F-statistic: 111.9 on 3 and 50 DF,  p-value: < 2.2e-16
```

```
#run models on test data
pred_val_model_1 <- predict(model_1, test)
pred_val_aic_model_1 <- predict(aic_model_1, test)
pred_val_model_2 <- predict(model_2, test)
pred_val_model_3 <- predict(model_3, test)
pred_val_aic_model_3 <- predict(aic_model_3, test)

#obtain R-squared using testing data
print(paste("model_1 R-squared:", (cor(pred_val_model_1, test$List_Price))^2))
```

```
## [1] "model_1 R-squared: 0.824154919411823"
```

```
print(paste("aic_model_1 R-squared:", (cor(pred_val_aic_model_1, test$List_Price))^2))
```

```
## [1] "aic_model_1 R-squared: 0.828809051853922"
```

```
print(paste("model_2 R-squared:", (cor(pred_val_model_2, test$List_Price))^2))
```

```
## [1] "model_2 R-squared: 0.835705449183693"
```

```
print(paste("model_3 R-squared:", (cor(pred_val_model_3, test$List_Price))^2))
```

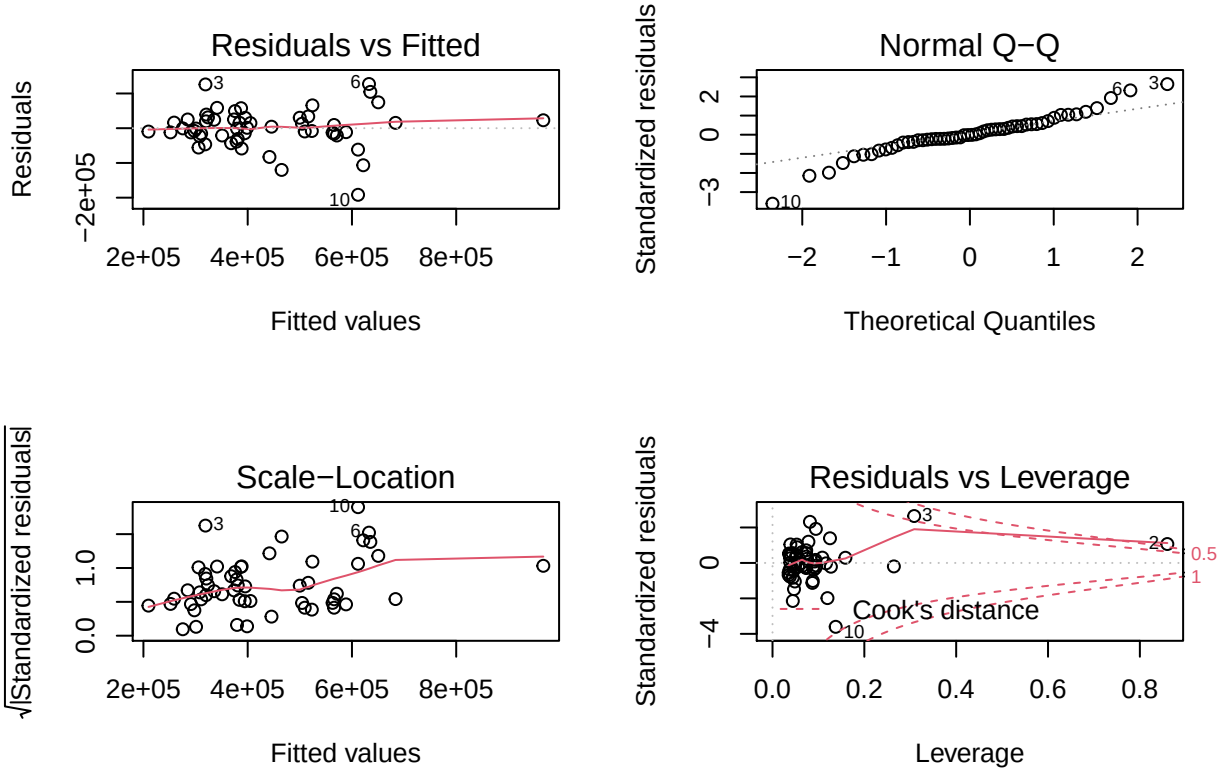
```
## [1] "model_3 R-squared: 0.841993699449823"
```

```
print(paste("aic_model_3 R-squared:", (cor(pred_val_aic_model_3, test$List_Price))^2))
```

```
## [1] "aic_model_3 R-squared: 0.83344879888398"
```

Assumptions

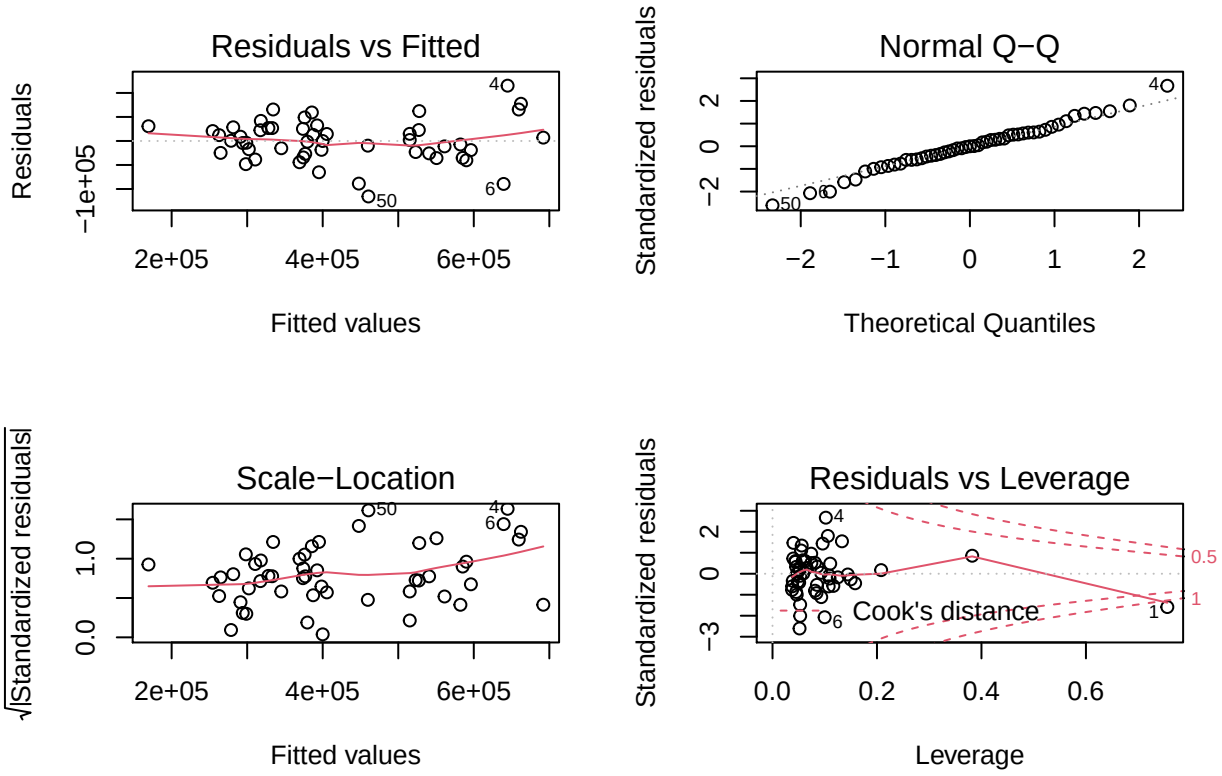
```
#Diagnostics of least squares regression fit
par(mfrow = c(2,2))
plot(model_3)
```



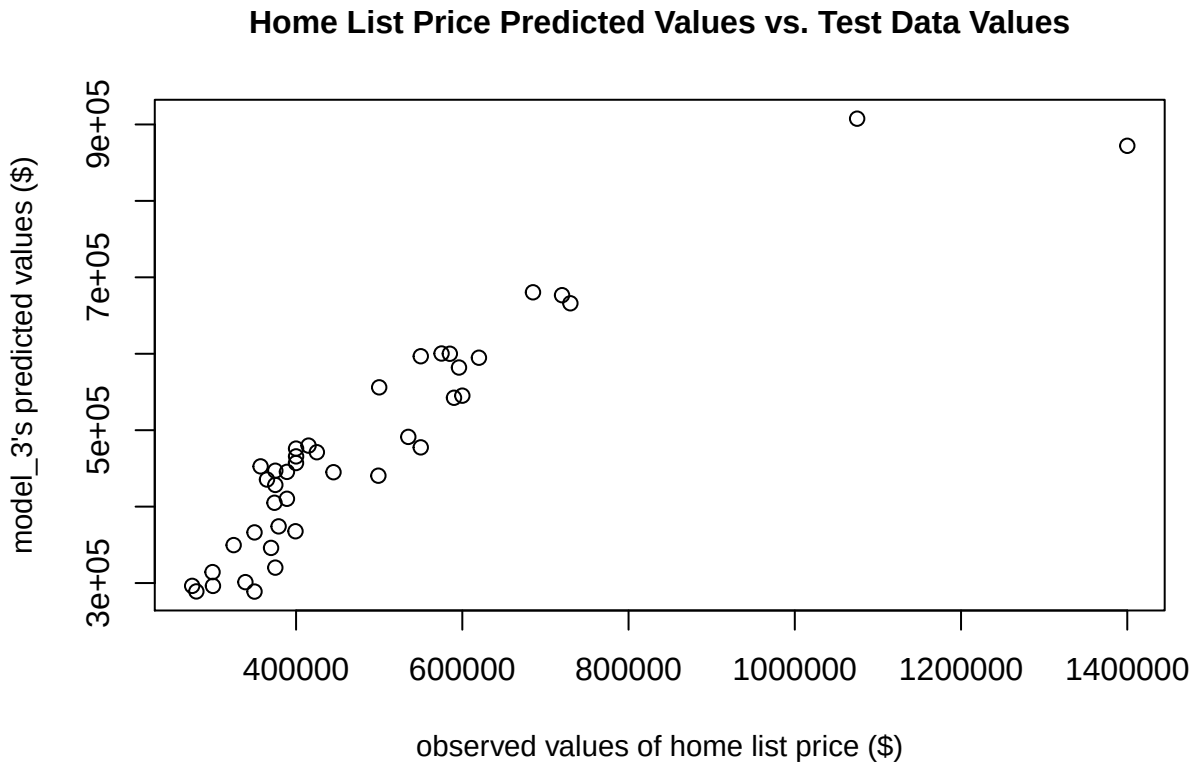
```r
#evaluate diagnostics after removal of influential points 2, 3, and 10
train_2 <- train[-c(2, 3, 10),]
train_2_model_3 <- lm(List_Price ~ Square_Feet + Acres + Stories + Garages, train_2)
summary(train_2_model_3)
```

```
##
## Call:
## lm(formula = List_Price ~ Square_Feet + Acres + Stories + Garages,
##     data = train_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -115540  -25355     -65   26006  115123
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -61031.76   29643.69  -2.059   0.0452 *
## Square_Feet    142.22      11.63  12.229 4.66e-16 ***
## Acres       110087.43   77247.04   1.425   0.1609
## Stories2    -14144.30   14899.12  -0.949   0.3474
## Garages      62851.67   13760.05   4.568 3.69e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 45500 on 46 degrees of freedom
## Multiple R-squared:  0.8984, Adjusted R-squared:  0.8896
## F-statistic: 101.7 on 4 and 46 DF,  p-value: < 2.2e-16
```

```r
#Diagnostics of least squares regression fit
par(mfrow = c(2,2))
plot(train_2_model_3)
```




```
#plot predicted values vs. observed values from testing data
plot(pred_val_model_3 ~ test$List_Price,
    main = 'Home List Price Predicted Values vs. Test Data Values',
    cex.main = 1,
    cex.lab = 0.9,
    ylab = 'model_3\'s predicted values ($)',
    xlab ='observed values of home list price ($)')
```




```
#build random forest model
set.seed(1234)

#bagging
cvcontrol <- trainControl(method="repeatedcv",
                          number = 5,
                          repeats = 2,
                          allowParallel=TRUE)
```

```
#create Random Forest model
forest <- train(List_Price ~ .,
             data=train,
             method="rf",
             trControl=cvcontrol,
             importance=TRUE)
```

```
forest
```

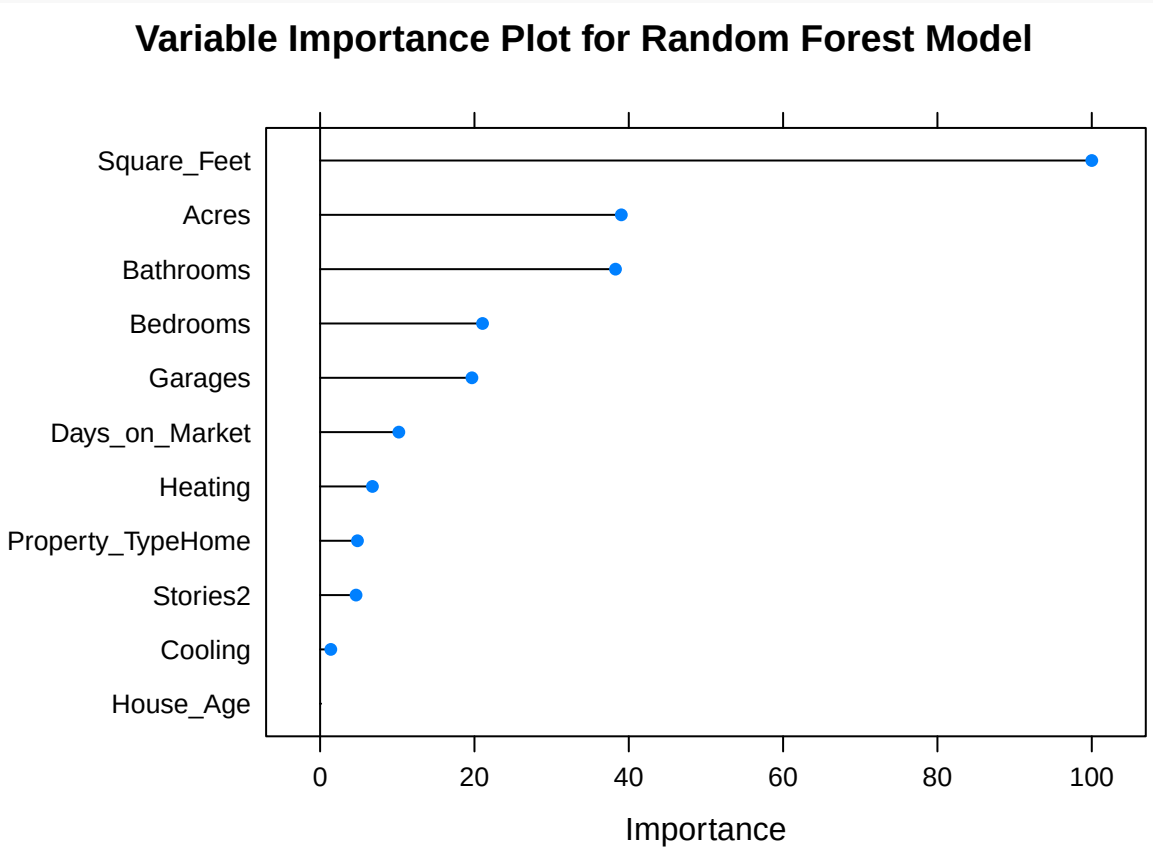
```
## Random Forest
##
## 54 samples
## 11 predictors
##
## No pre-processing
## Resampling: Cross-Validated (5 fold, repeated 2 times)
## Summary of sample sizes: 43, 45, 42, 43, 43, 44, ...
## Resampling results across tuning parameters:
##
##   mtry  RMSE      Rsquared   MAE
##    2    91940.76  0.7300187  66048.10
##    6    85950.48  0.7788343  59157.12
```

```
##   11    87643.97  0.7716018  61040.09
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 6.
```

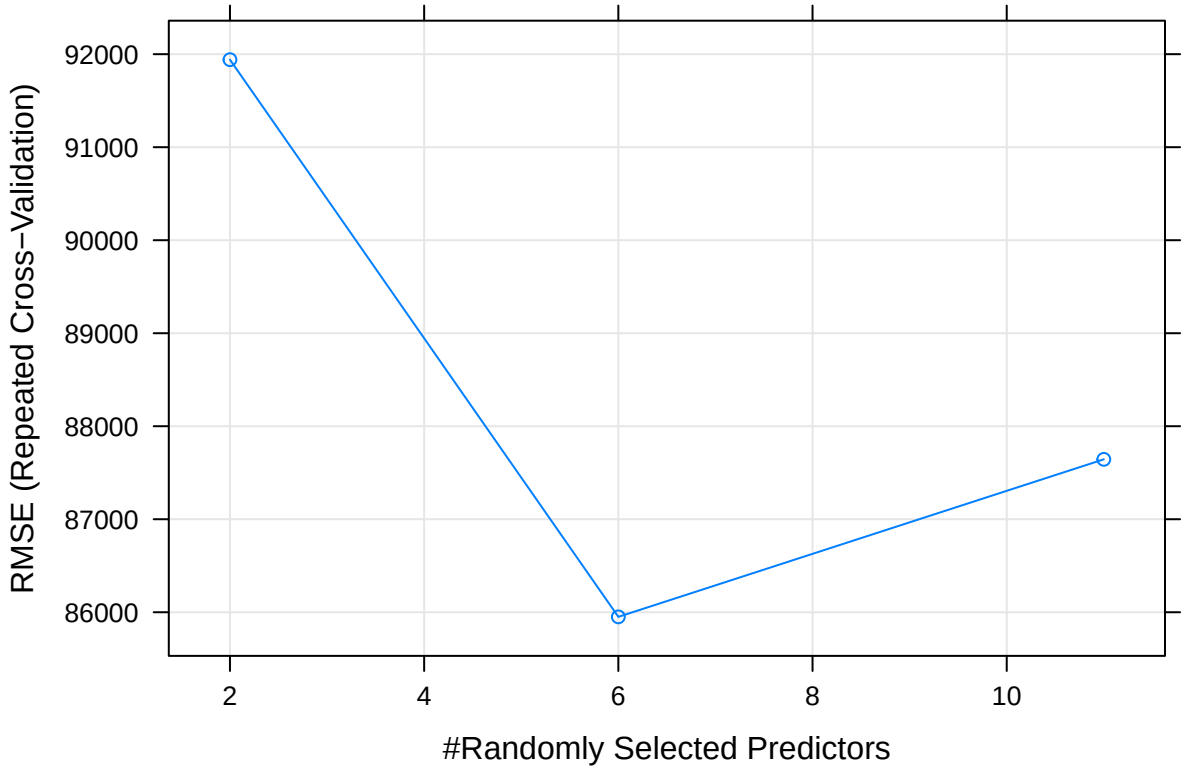
```
#table with variable importances
varImp(forest)
```

```
## rf variable importance
##
##                   Overall
## Square_Feet       100.000
## Acres              39.031
## Bathrooms          38.280
## Bedrooms           21.047
## Garages            19.673
## Days_on_Market     10.202
## Heating             6.779
## Property_TypeHome   4.841
## Stories2            4.658
## Cooling             1.383
## House_Age           0.000
```

```
#plot variable importance
plot(varImp(forest),
     main = 'Variable Importance Plot for Random Forest Model')
```



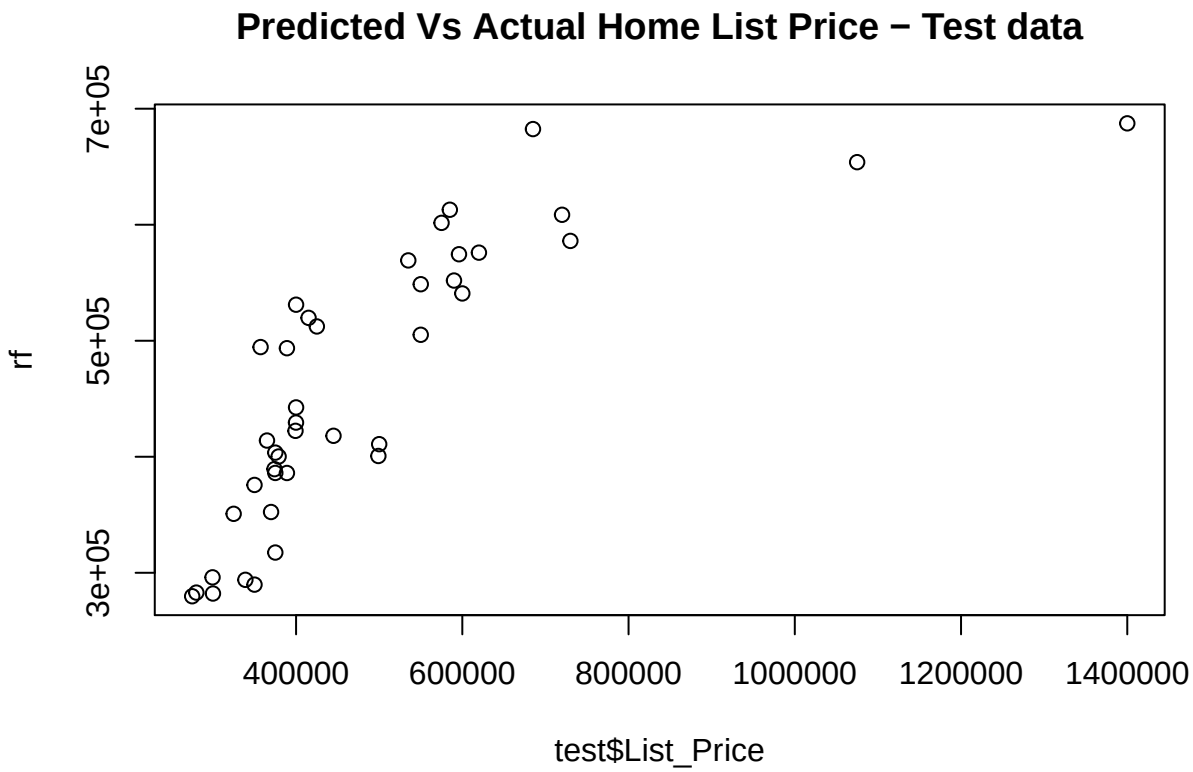

```
plot(forest)
```




```
#evaluate %variance explained
forest$finalModel
```

```
##
## Call:
##  randomForest(x = x, y = y, mtry = param$mtry, importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 6
##
##           Mean of squared residuals: 7478447173
##                     % Var explained: 67.77
```

```
#plot, RMSE, R-square
rf <-  predict(forest,  test)
plot(rf ~ test$List_Price, main = 'Predicted Vs Actual Home List Price - Test data')
```




```
sqrt(mean((test$List_Price - rf)^2))
```

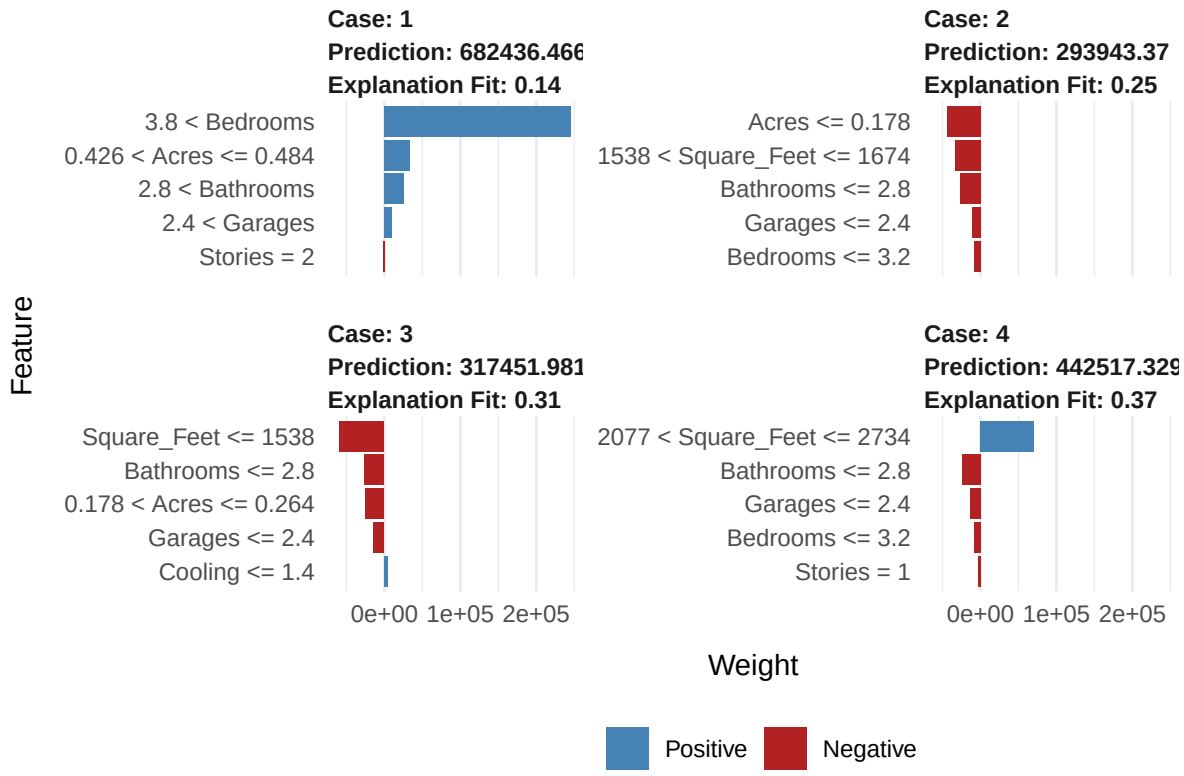
```
## [1] 142488.8
```

```
test_r_squared <- cor(test$List_Price, rf) ^2
print(paste("The test R-squared for the random forest model is",
            test_r_squared))
```

```
## [1] "The test R-squared for the random forest model is 0.61068835052904"
```

```
# explain predictions
explainer <- lime(test[1:4,], forest, n_bins = 5)
explanation <- explain( x = test[1:4,],
                        explainer = explainer,
                        n_features = 5)

plot_features(explanation)
```




```
plot_explanations(explanation)
```

```
## Warning: Unknown or uninitialised column: `label`.
```