Price Prediction of Realtor.com Listings

STAT 4355 - Applied Linear Models

Anveetha Suresh

The University of Texas at Dallas

Table of Contents

1: Introduction

The real estate market is a critical component of the economy, with home prices influencing wealth, lifestyle, and professional opportunities. This report explores a linear regression analysis aimed at predicting home prices using data from Realtor.com. By examining various factors such as location, square footage, and home capacity, the study seeks to identify the key variables driving home values and develop a model to provide accurate price predictions.

The data utilized to create this model was accessed from Kaggle.com. The created by Ahmed Shahriar Sakib, the *USA Real Estate Dataset* [2] collects data from Realtor.com, a real estate listing website based in Santa Clara, California. This dataset contains 2,226,382 entries, with 10 columns (variables) each. Although the data was from one source, it will still be unbiased and diverse, as it covers ~2226382 different zip codes and 2184282 different cities across the nation.

The variables listed below are all of the variables in the dataset:

Response Variable:

- **House Price (price)** is our response variable. The other variables in the dataset ultimately determine the house price, which **ranges from 0 to 2.15 billion dollars** in this dataset.

Predictor Variables:

- **Lot Size (acre_lot):** The size of the property will have an affect on the price of the home. Larger properties will have higher prices.
- **Number of bathrooms (bath):** The number of bathrooms in a home is an indicator of how large the home is and how many people it can accommodate. More bathrooms will indicate a larger home, and an inherently higher priced home.
- **Number of bedrooms (bed): T**he number of beds in a home is an indicator of how large the home is and how many people it can accommodate. More beds will indicate a larger home, and an inherently higher priced home.
- **Property sale status (status):** The property sale status will not tell us information about the price. It will only tell us if the home is sold or not, which is dependent on other factors like age.
- **ZIP Code (zip_code):** The ZIP code of the home determines its proximity to amenities, and more desirable ZIP codes will have higher prices. This variable may be difficult to analyze, as it would require an analysis on which zip codes are more desirable than others.
- **City (city):** The city the home is in may also help determine its price, as larger cities typically have higher cost of living. To use this variable we will have to create a system to show which cities are larger or more desirable than others.
- **State (state):** The state the home is in may also help determine its price, as larger states typically have higher cost of living. To use this variable we will have to create a system to show which states are larger or more desirable than others.

- **Street (street):** The street variables in this dataset are arbitrary numbers that represent the state, zip code, and street the home is on. This data will not help us understand the pricing of a home.
- **Broker/Agency (brokered_by):** The broker variables in this dataset are arbitrary numbers that represent the agency the home is sold by. This data will not help us understand the pricing of a home without knowing exactly which agencies are selling the home. This information is not provided in our dataset.

Through further analysis, state, street, broker/agency, city, and property sale status were deemed inapplicable for the linear regression. Additionally, one the most important factors for house pricing is the home's proximity to resources like schools and offices. In areas with higher population density, the proximity to these resources are higher, and so is the cost of living. So, with the intention of incorporating this aspect of house pricing, the Standard Co. *Population Density (per square mile) for every US Zip Code* [3] dataset was merged into the original to add an extra variable to the dataset. This dataset included the following variables, but only population_density and zip were used.

Variables:

- **zip**: zip code of home in integer format
- **population**: number of people within the zip code in integer format
- **population_density**: number of people within the zip code divided the the total square mileage of the zipcode in double format
- **city**: string containing the city's name
- **state**: string containing the state's name
- **latitude**: double with latitude coordinates of the zip code
- **longitude**: double with longitude coordinates of the zip code

2: Data Cleaning and Preprocessing

Before exploring and analyzing the data, it was important to clean and preprocess the data. To start, I took a look at the unique states and cities represented in the dataset. I noticed that Puerto Rico, Virgin Islands, Guam, New Brunswick, Hawaii and Alaska were included as states in the dataset. To avoid major issues, I removed any data that was from those states, as I wanted to make the data predict the prices of mainland American homes. Additionally, I removed any data points that had NULL values for the relevant variables (bed, bath, acre_lot, and house_size). After doing this, I merged the Population Density dataset with zip_code as the key corresponding to zip in the dataset. For example if a home has a zip_code of 001, the population_density associated with 001 in the density dataset would then be added into a new column in the home dataset. Lastly, after all of the cleaning and data point removal, the dataset still had ~1 million data points, which was difficult to work with due to my device's operating power. So, I took a random subset of 25,000 data points.

3: Working with the Data

3.1 Exploratory Data Analysis (EDA)

After cleaning the data, I conducted exploratory data analysis of the variables. I started by creating histograms of each of the variables, and noticed severe right skewing (Figure 3.1.1).

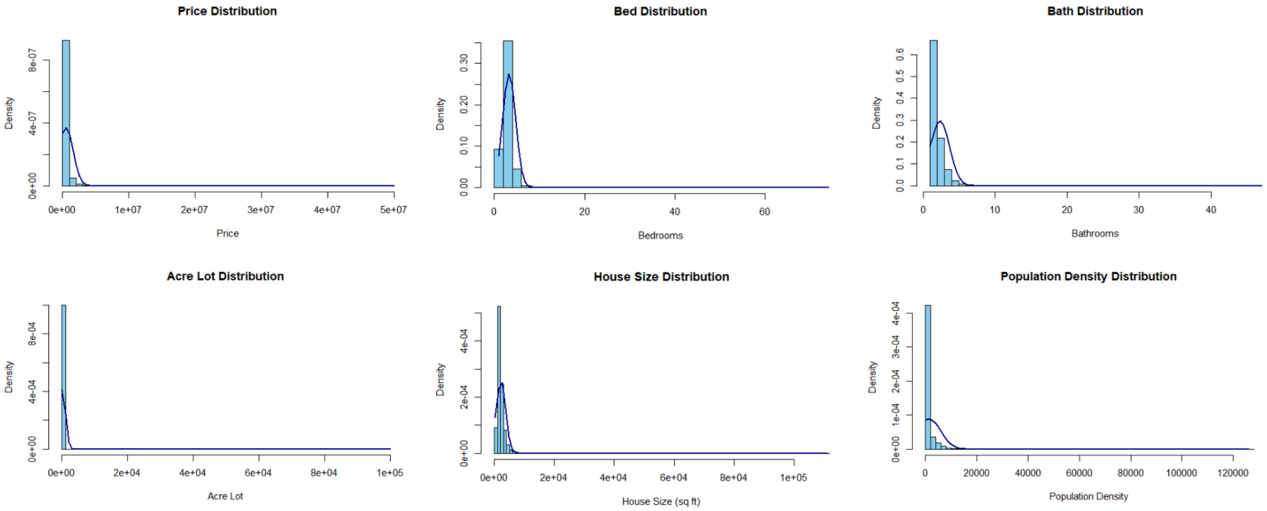


Figure 3.1.1

Looking at these histograms, I noticed that there are many severe outliers that are caused by extremely extravagant and luxurious homes, like castles, mansions, and more. This caused me to redirect my vision for the data and the scope of the project. I decided to cap the variable values to create a more focused dataset. Based on publicly available information and arbitrary limits, I capped the variables accordingly: Price: 0 - 100000000, Bed: 1 - 7, Bath: 1 - 7, Acre Lot: 0 - 1, House Size: 0 - 6000, Population Density: 700 - 4000. Figure 3.1.2 shows the distribution of the predictors after this re-scoping. This scope allowed me to prioritize single family, suburban homes.

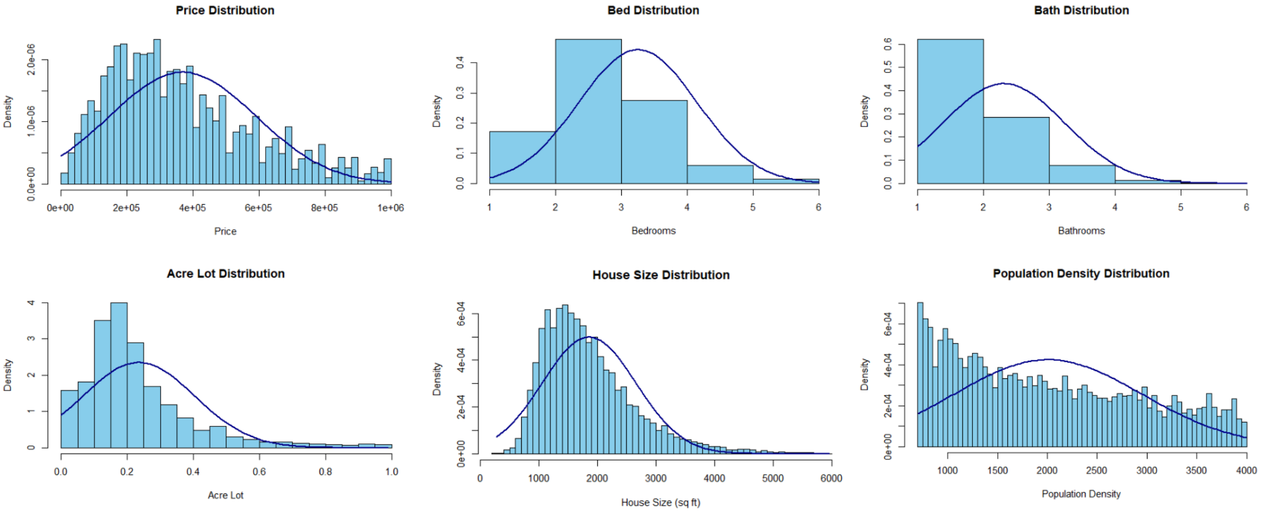


Figure 3.1.2

Lastly, I created a correlation matrix to analyze the relationships between variables and check for multicollinearity. Figure 3.1.3 shows us that there is a fairly strong correlation between the number of bedrooms in a home and the size of the home. Additionally, it shows us that there are somewhat negative correlations between the population density and the other characteristics of a home. As population density goes up, the indicators for a large home decrease.

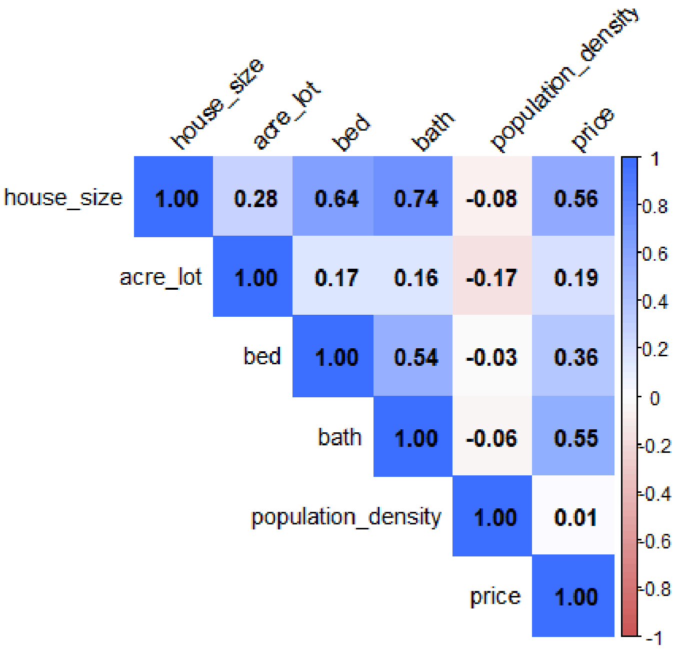
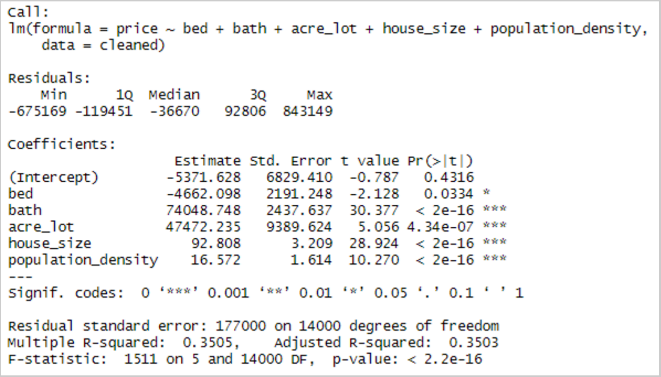


Figure 3.1.3

3.2 Model Fitting

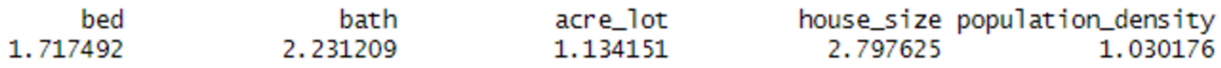
After looking at all of the variables and the EDA, I decided to proceed in using the house_size, acre_lot, bed, bath, population_density, and price as the variables in the model with price being the response variable. I created a full model using these variables. The summary of the model is shown in Figure 3.2.1. Upon looking at the model and the p-values for the predictors, none of them are greater than an α of .05. This demonstrates that leaving the model as it is may be an acceptable decision. But, when observing even further, it is evident that the predictor for the bed variable is a negative value. This, in context, tells us that the addition of a bedroom in a house reduces the price of the home by approximately 4000 dollars. This is a logical issue with the model, but because there are no other indicators that justify



```
Call:
lm(formula = price ~ bed + bath + acre_lot + house_size + population_density,
    data = cleaned)

Residuals:
    Min      1Q  Median      3Q     Max
-675169 -119451  -36670   92806  843149

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)        -5371.628   6829.410  -0.787   0.4316
bed                -4662.098   2191.248  -2.128   0.0334 *
bath               74048.748   2437.637  30.377  < 2e-16 ***
acre_lot           47472.235   9389.624   5.056 4.34e-07 ***
house_size            92.808      3.209  28.924  < 2e-16 ***
population_density    16.572      1.614  10.270  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 177000 on 14000 degrees of freedom
Multiple R-squared:  0.3505,    Adjusted R-squared:  0.3503
F-statistic:  1511 on 5 and 14000 DF,  p-value: < 2.2e-16
```

Figure 3.2.1 (to the left)

removal of the variable, I continued with this model. Additionally, I considered that there may be some form of multicollinearity between bed and bath, as the value of a bath (~$74000 per bathroom) is very high. It is possible that the values of this model exhibit these behaviors because of the fact that both bed and bath are correlated. But when running a colinearity test with the variables, all of the Variance Inflation Factors (Figure 3.2.2) were within reasonable values, so again, there was no statistical indication that justified taking out a variable from this model, although there was a logical one.



```
     bed              bath          acre_lot        house_size population_density
1.717492          2.231209          1.134151          2.797625           1.030176
```

Figure 3.2.2

3.3 Residual Analysis

Using the full model, I proceeded to utilize residual analysis to further examine the model's adequacy. I created scatterplots for Fitted Values, R Student and Standardized residuals. First, when looking at the Residual vs Fitted Value scatterplot, I noticed some issues with the shape of the graph. The original plot is Figure 3.3.1 and the plot with issues highlighted is 3.3.2.

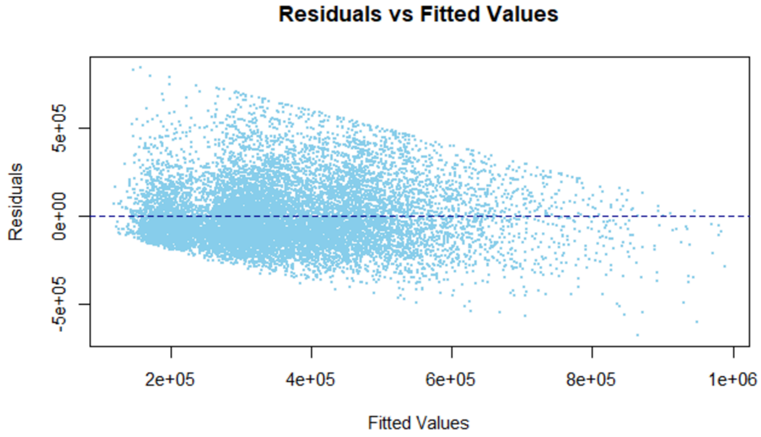


Figure 3.3.1

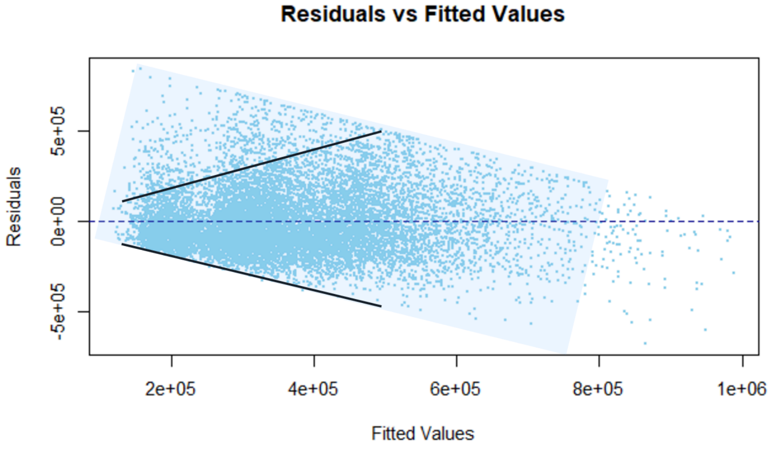
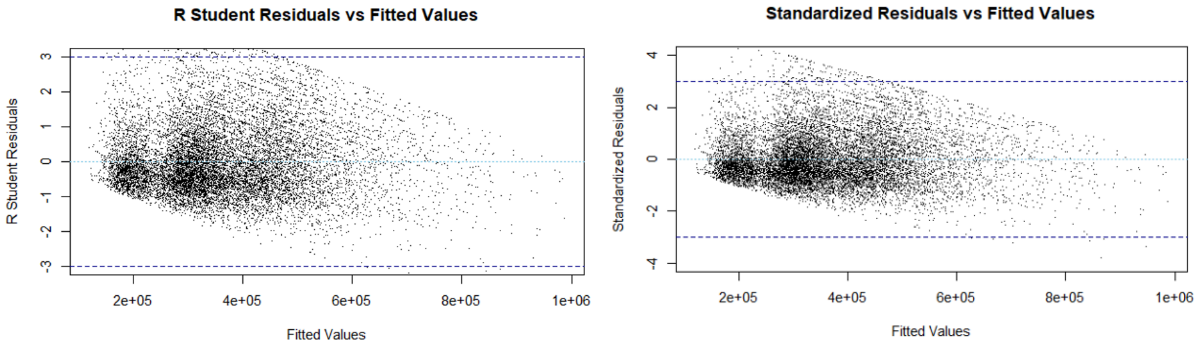


Figure 3.3.2

When taking a look at Figure 3.3.2, the plot is noticeably downward sloping, and it contains a funnel shape, telling that the model needs to be transformed, and that the variance of the residuals is not constant. Both of these issues demonstrate issues in the model that could potentially be fixed using transformation or changing the model entirely. The downward slope of the model tells us that a linear model may not be fit for the data, and that the lower price values are being overfitted. The other two residual models are pictured in Figure(s) 3.3.3

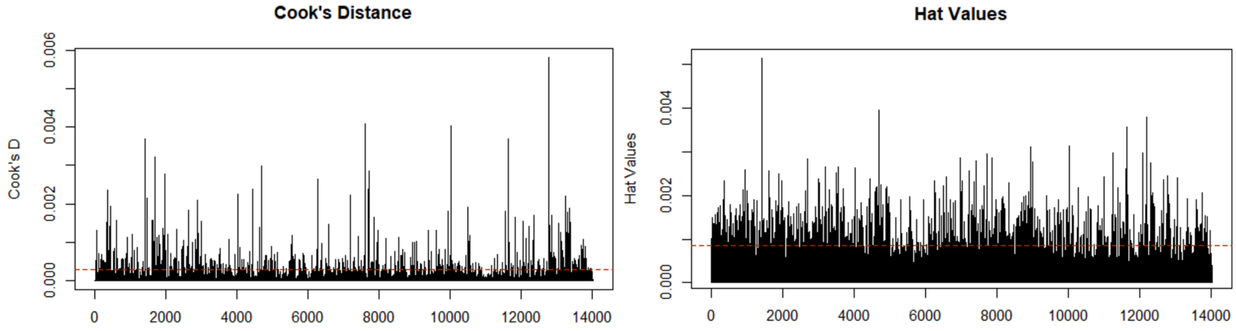


Figure(s) 3.3.3

When taking a look at Figure 3.3.3, the numbers residuals that lie outside of the acceptable threshold are quite high. This is another issue with the model. It lacks accuracy, and the model is likely overfitting – evident by the bulk of residuals that lie *above* the thresholds. This is improved with transformations later.

3.4 Influential Point Analysis

To analyze influential points, Cook's Distance, Hat Values and DFBETA Plots were used. Due to the nature of the data set, all of these plots show a substantial number of influential points. Oftentimes, home price data sets have fluctuations and influential points due to luxury homes (even within single, suburban homes), location, housing market, economy and more [2]. These factors are factors that are difficult to account for in this particular model, which resulted in a high number of influential points. Additionally, removing these points presents a challenge due to the proportion of influential points that are in the dataset, so removal of influential points was not a viable option for this model. Regardless, the Figures 3.4.1(Cook's Distance, Hat Values) & 3.4.2 (DFBETA Plots) are pictured below.



Figure(s) 3.4.1

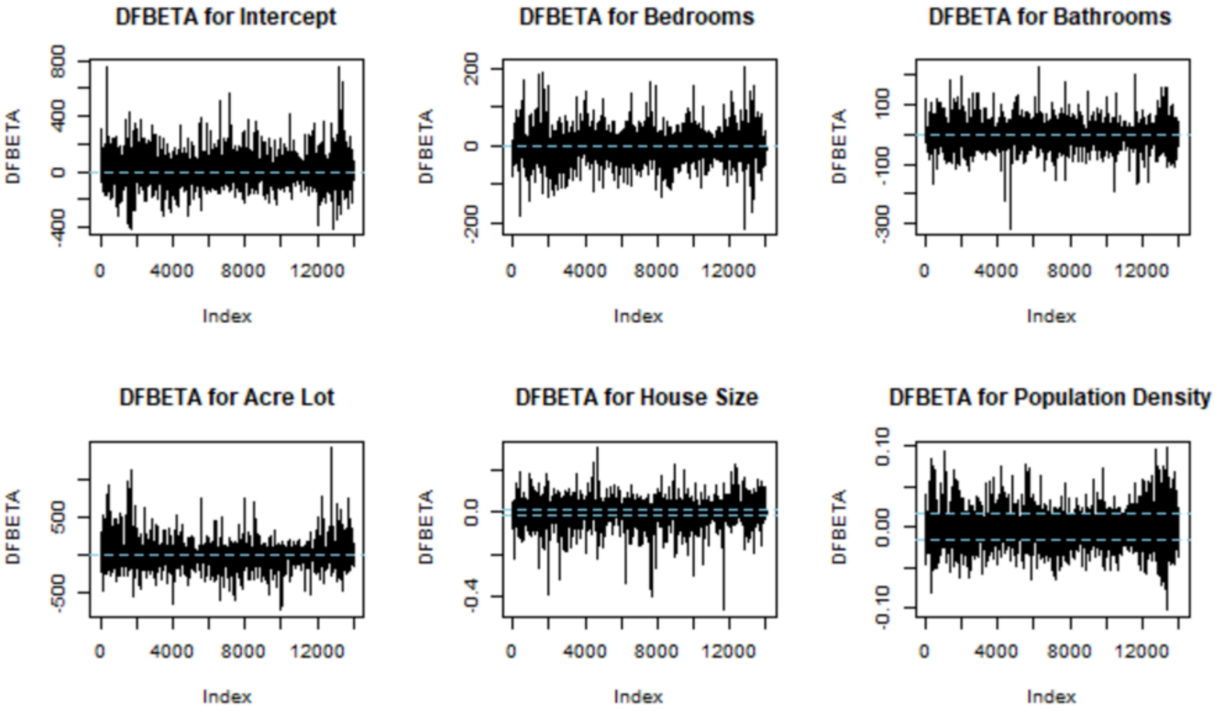


Figure 3.4.2

3.5 Transformation

After all of the considerations presented, I continued on to transformation of the data. To appropriately transform this linear model, I utilized a Box-Cox plot (Figure 3.5.1), and transformed the model with the corresponding lambda value from the plot. The resulting lamba value of the plot was approximately 0.42. This indicates that a square root transformation is ideal.

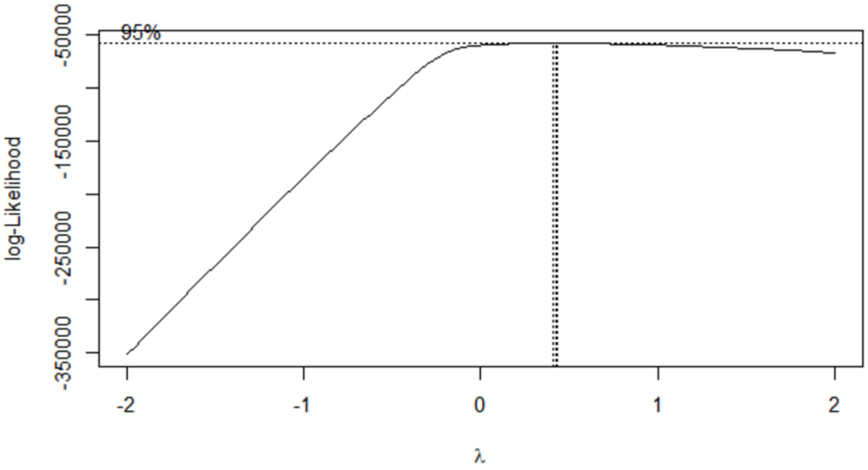


Figure 3.5.1

Prior to conducting a square root transformation, I created a QQ Plot to analyze the accuracy of the original model (Figure 3.5.2) . This QQ Plot, while somewhat aligned with the expected diagonal, has some issues with the peaks towards the left and the right side. After the transformation, there is a noticeable reduction in the peaks of the QQ Plot (Figure 3.5.3).

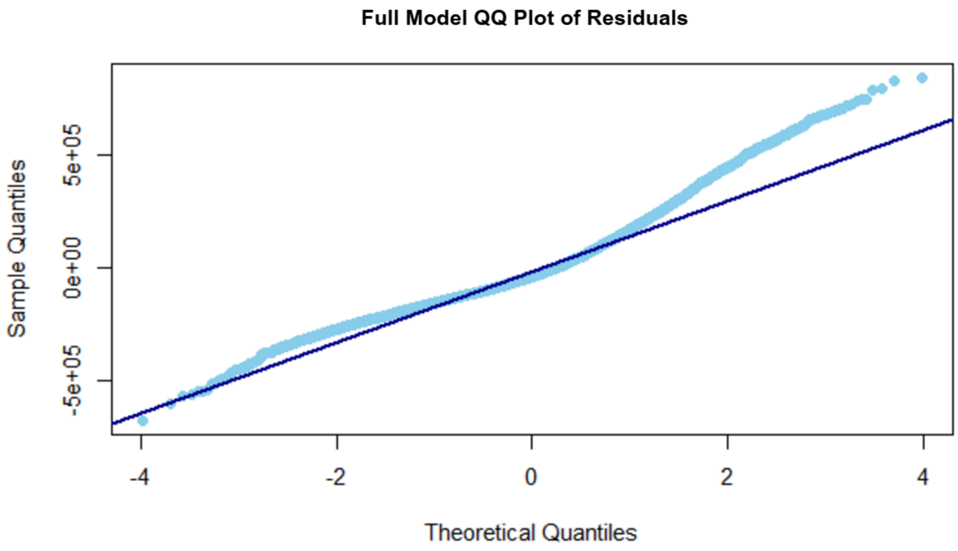


Figure 3.5.2

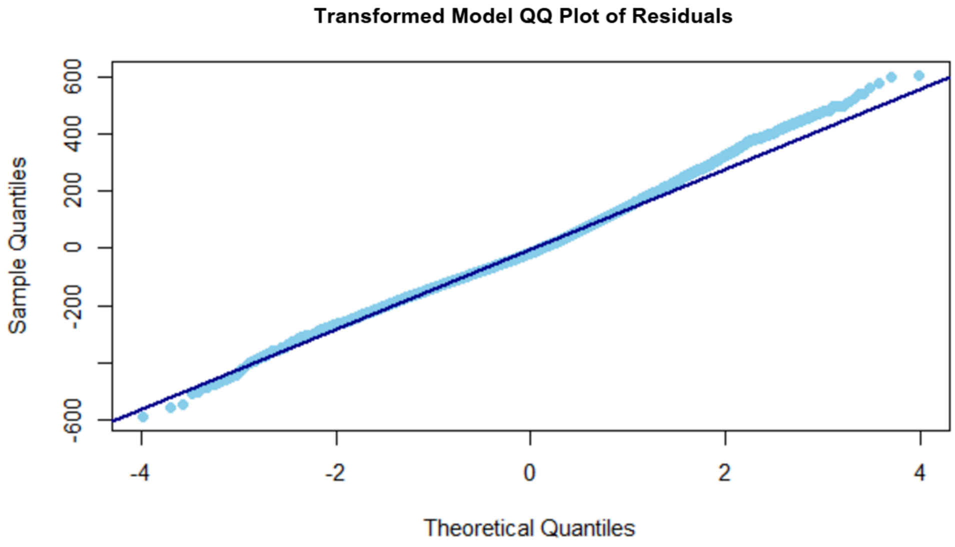


Figure 3.5.3

Additionally, when analyzing the R-Student Residuals for both of the models, the number of residuals outside the threshold reduces significantly. The number of residuals in the full model (Figure 3.4.4), 160, decreases to 66 residuals in the square root model (Figure 3.4.5), and when the total number of data points is ~ 14000, the 66 residuals is somewhat reasonable, especially given the variety of data and problems encountered in the earlier steps of the model.

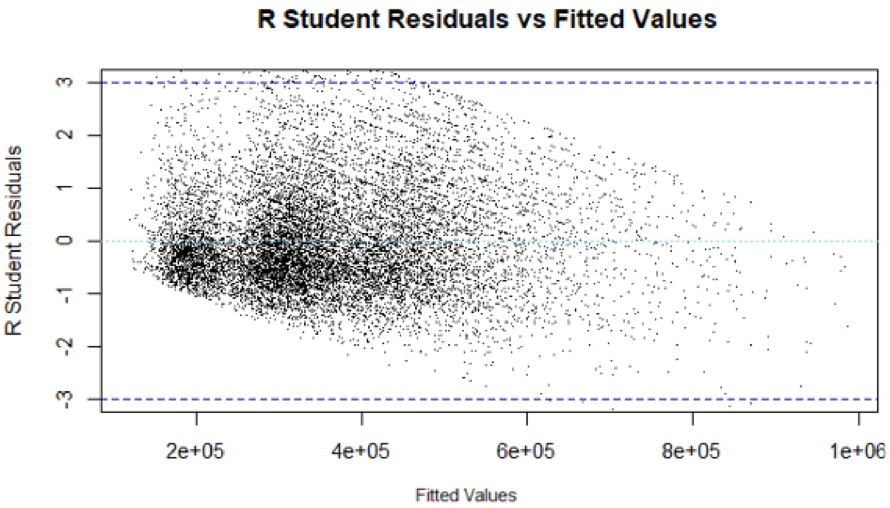


Figure 3.5.4 (Full Model - 160 Residuals)

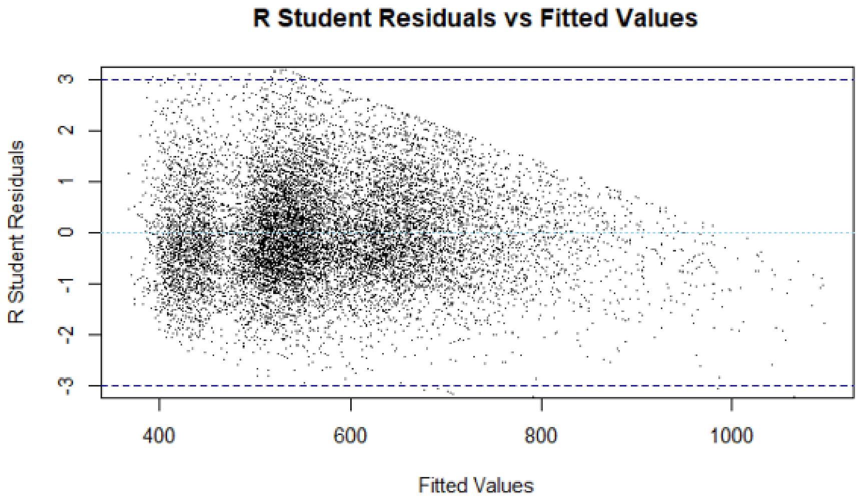
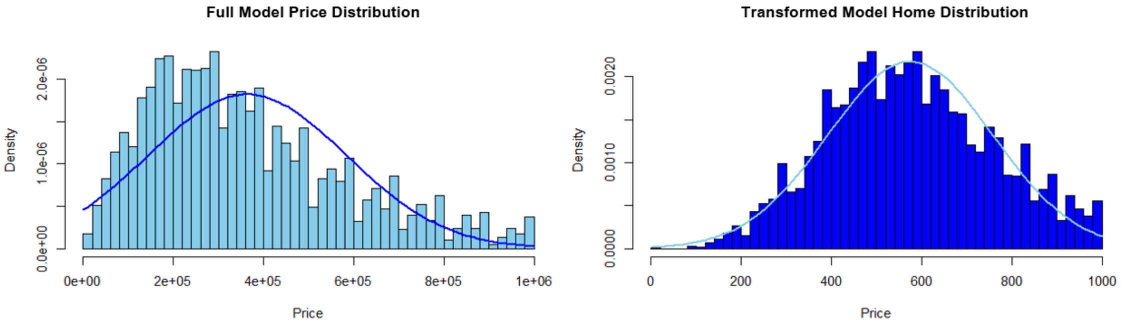


Figure 3.5.5 (Square Root Model - 66 Residuals)

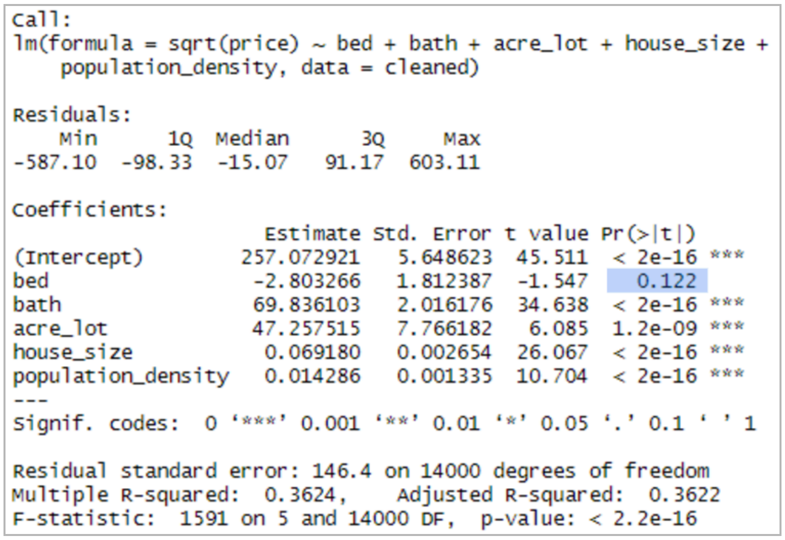
Looking at the distribution of the home prices after transformation, the rightward shift is fairly noticeable, normalizing the curve further. Figure 3.5.6 shows changes in the price distribution before and after transformation.

Figure 3.5.6



3.6 Post-Transformation Observation

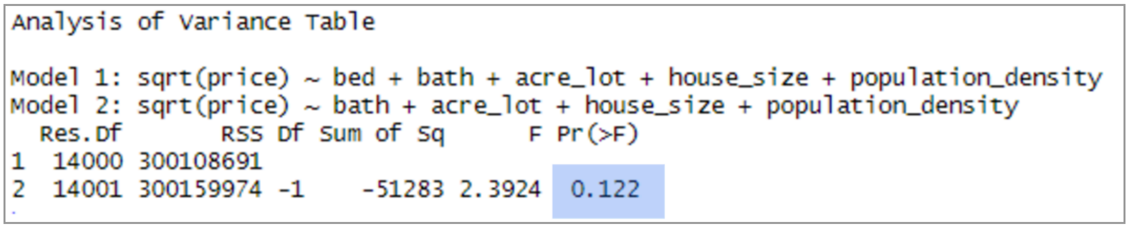
After looking at the QQ Plot and Residual Plots of both models, we are able to establish that the transformed model is more ideal than the original model. Although this is the case, there are other issues that arise during the process of transforming the model. When looking at the summary of the transformed model (Figure 3.6.1), it is evident that the p-value for the bed variable is larger than an α of .05. This demonstrates that it is important to test the model without that variable.



```
Call:
lm(formula = sqrt(price) ~ bed + bath + acre_lot + house_size +
    population_density, data = cleaned)

Residuals:
    Min      1Q  Median      3Q     Max
-587.10  -98.33  -15.07   91.17  603.11

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)        257.072921   5.648623  45.511  < 2e-16 ***
bed                 -2.803266   1.812387  -1.547    0.122
bath                69.836103   2.016176  34.638  < 2e-16 ***
acre_lot            47.257515   7.766182   6.085  1.2e-09 ***
house_size           0.069180   0.002654  26.067  < 2e-16 ***
population_density   0.014286   0.001335  10.704  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 146.4 on 14000 degrees of freedom
Multiple R-squared:  0.3624,	Adjusted R-squared:  0.3622
F-statistic:  1591 on 5 and 14000 DF,  p-value: < 2.2e-16
```

Figure 3.6.1

After making this observation, I conducted an ANOVA Test (Figure 3.6.2) to check how the models compared to one another, and the result was that the reduced model was better fitting than the full model. This makes our final model a transformed, reduced model.

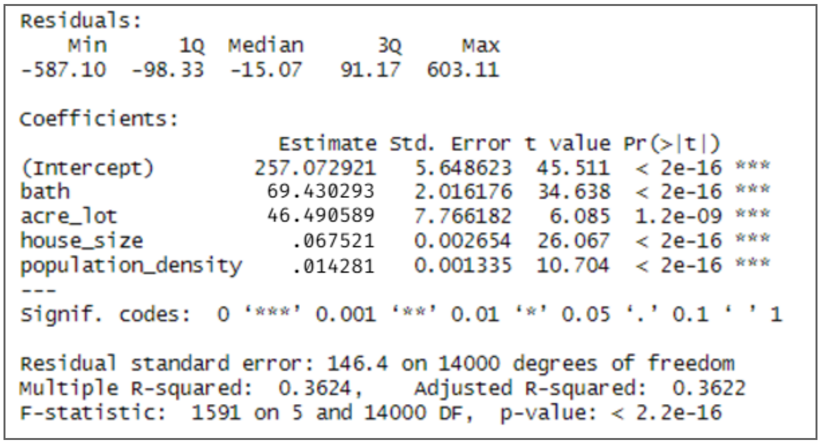


```
Analysis of Variance Table

Model 1: sqrt(price) ~ bed + bath + acre_lot + house_size + population_density
Model 2: sqrt(price) ~ bath + acre_lot + house_size + population_density
  Res.Df       RSS Df Sum of Sq      F Pr(>F)
1  14000 300108691
2  14001 300159974 -1    -51283 2.3924  0.122
```

Figure 3.6.2

4 Final Model

The final model shown in Figure 4.1 is the model that has been derived from all of the methods previously listed.



```
Residuals:
    Min      1Q  Median      3Q     Max
-587.10  -98.33  -15.07   91.17  603.11

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)        257.072921   5.648623  45.511  < 2e-16 ***
bath                69.430293   2.016176  34.638  < 2e-16 ***
acre_lot            46.490589   7.766182   6.085  1.2e-09 ***
house_size            .067521   0.002654  26.067  < 2e-16 ***
population_density    .014281   0.001335  10.704  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 146.4 on 14000 degrees of freedom
Multiple R-squared:  0.3624,    Adjusted R-squared:  0.3622
F-statistic:  1591 on 5 and 14000 DF,  p-value: < 2.2e-16
```

Figure 4.1

This model tells us very little information when analyzed on the surface, and it is also very minimal in nature. There are only 4 predictor variables, which is not realistic to an actual home pricing situation. Home pricing involves factors such as the housing market, economy, proximity to resources, size, previous owners, age, condition, and more. What it does tell us is that bathrooms have a substantial effect on housing price per unit. Additionally, acreage and property size is the second most relevant factor to home pricing.

5 Conclusion

5.1 Conclusion

While this particular linear model has many flaws, it tells us that property size and number of bathrooms are two of the biggest factors in determining the price of a home. The lack of detail and information in this model tells us that many more changes need to be made, and it is possible that a linear regression is not the most ideal model for the situation. This can be seen in our residual plots before and after transformation, which answers the question of whether or not a linear model is most ideal for home price prediction. It also tells us that we can only somewhat accurately predict the price of a home in this manner. For this model, it may have been better to select a different dataset, which will be covered in more detail in 5.2: Reflection.

5.2 Reflection

After creating this linear model, I realized there are many issues with how I approached this project. The primary issue in this model is the selection of the model, but also the variables that were considered in it. Many of the issues in the model arose because of overfitting and incorrect model choice.

To combat these issues, it could be valuable to look into a different dataset and different models. Additionally, creating more specific scopes could create a more accurate model. Creating separate models for suburban, urban, and rural homes would be different than creating models for apartments, mansions, and condos. These nuances make it difficult to create an overarching model to predict home prices. Creating a multitude of models for more specific situations would fix this issue.

When it comes to approaching this particular dataset, I would have liked to incorporate more categorical variables, but the only categorical variable that could be relevant is the sale status of the home, which does not have any significant difference in price (Figure 5.2.1).

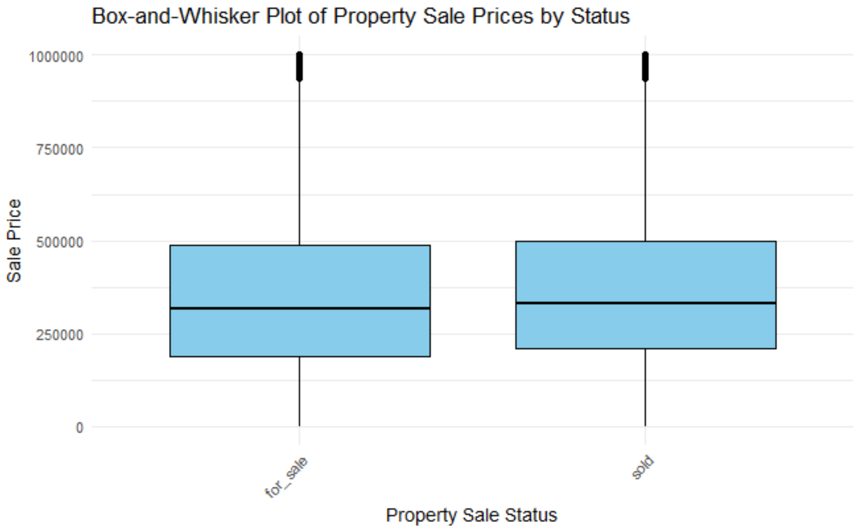


Figure 5.2.1

Another method of variable selection I did not consider was stepwise selection. Because I did not consider stepwise selection from the beginning, I may have missed combinations that may have provided more value than the existing combination.

Lastly, one oversight that may have been missed is feature transformation and scaling. This could have provided more insight and accuracy to the model. Making the house price and property size the same scale could have also made a difference.

On the contrary, this model was very insightful in seeing how certain factors affect home prices. The residual plots and QQ Plots told us a substantial amount of information about the adequacy of the models, which is an aspect of the project I appreciated.

6 Appendix

6.1 R Script

```
library(ggplot2)
library(dplyr)
library(car)
library(MASS)
library(leaps)

setwd("C:/Users/Anveetha Suresh/OneDrive/Desktop/stat 4355/final project")
originalData <- read.csv('realtor-data.csv')
data <- read.csv("homesdata.csv")
density <- read.csv('population-density.csv')
```

```
       colnames(originalData)[colnames(originalData) == "zip_code"] <- "zip"

       originalData <- merge(originalData, density[c("zip", "population_density")],
                             by = "zip", all.x = TRUE)

       originalData <- subset(originalData,!is.na(acre_lot))
       originalData <- subset(originalData,!is.na(bath))
       originalData <- subset(originalData,!is.na(bed))
       originalData <- subset(originalData,!is.na(price))
       originalData <- subset(originalData,!is.na(house_size))
       originalData <- subset(originalData,!is.na(population_density))
       originalData <- subset(originalData,!is.na(brokered_by))
       originalData <- subset(originalData,!is.na(zip))
       originalData <- subset(originalData,!is.na(street))

       states_to_remove <- c("Puerto Rico", "Guam", "New Brunswick", "Virgin Islands", "Hawaii")
       originalHomes <- originalData
       originalData <- subset(originalData, !state %in% states_to_remove)

       # Create a new dataset with one random point from each unique zipcode
       originalData <- originalData %>%
         group_by(zip) %>%
         slice_sample(n = 4) %>%
         ungroup()

       cleaned <- as.data.frame(originalData)

       cleaned <- subset(cleaned, bath <  7)
       cleaned <- subset(cleaned, bed < 7)
       cleaned <- subset(cleaned, acre_lot < 1)
       cleaned <- subset(cleaned, house_size < 6000)
       cleaned <- subset(cleaned, population_density < 4000)
       cleaned <- subset(cleaned, population_density > 700)
       cleaned <- subset(cleaned, price < 1000000)

       # Ensure your dataset has the necessary columns
       if ("status" %in% colnames(cleaned) && "price" %in% colnames(cleaned)) {
         # Create a box-and-whisker plot using ggplot2
         ggplot(cleaned, aes(x = status, y = price)) +
           geom_boxplot(fill = "skyblue", color = "black") + # Boxplot with colors
           theme_minimal() +                              # Minimal theme for clean visuals
           labs(
             title = "Box-and-Whisker Plot of Property Sale Prices by Status",
             x = "Property Sale Status",
             y = "Sale Price"
           ) +
           theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotate x-axis labels if
needed
       }

       hist(originalData$price,
            prob = TRUE,  # Convert to probability density
```

```
     main = "Price Distribution Before Scope Change",
     xlab = "Price",
     col = "skyblue",
     border = "black",
     breaks = 50)

# Add normal curve
x <- seq(min(originalData$price), max(originalHomes$price), length = 100)
curve <- dnorm(x, mean = mean(originalHomes$price), sd = sd(originalHomes$price))
lines(x, curve, col = "darkblue", lwd = 2)

hist(originalHomes$bed,
     prob = TRUE,
     main = "Bed Distribution Before Scope Change",
     xlab = "Bedrooms",
     col = "skyblue",
     border = "black",
     breaks = 50)
x <- seq(min(originalData$bed), max(originalData$bed), length = 100)
curve <- dnorm(x, mean = mean(originalData$bed), sd = sd(originalData$bed))
lines(x, curve, col = "darkblue", lwd = 2)

# Bathrooms histogram with normal curve
hist(originalData$bath,
     prob = TRUE,
     main = "Bath Distribution Before Scope Change",
     xlab = "Bathrooms",
     col = "skyblue",
     border = "black",
     breaks =50)
x <- seq(min(originalData$bath), max(originalData$bath), length = 100)
curve <- dnorm(x, mean = mean(originalData$bath), sd = sd(originalData$bath))
lines(x, curve, col = "darkblue", lwd = 2)

# Acre lot histogram with normal curve
hist(originalData$acre_lot,
     prob = TRUE,
     main = "Acre Lot Distribution Before Scope Change",
     xlab = "Acre Lot",
     col = "skyblue",
     border = "black",
     breaks = 100)
x <- seq(min(originalData$acre_lot), max(originalData$acre_lot), length = 100)
curve <- dnorm(x, mean = mean(originalData$acre_lot), sd = sd(originalData$acre_lot))
lines(x, curve, col = "darkblue", lwd = 2)

# House size histogram with normal curve
hist(originalData$house_size,
     prob = TRUE,
     main = "House Size Distribution Before Scope Change",
     xlab = "House Size (sq ft)",
     col = "skyblue",
     border = "black",
            breaks = 100)
       x <- seq(min(originalData$house_size), max(originalData$house_size), length = 100)
       curve <- dnorm(x, mean = mean(originalData$house_size), sd = sd(originalData$house_size))
       lines(x, curve, col = "darkblue", lwd = 2)

       # Population density histogram with normal curve
       hist(originalData$population_density,
            prob = TRUE,
            main = "Population Density Distribution Before Scope Change",
            xlab = "Population Density",
            col = "skyblue",
            border = "black",
            breaks = 50)
       x <- seq(min(originalData$population_density), max(originalData$population_density), length
= 100)
       curve <- dnorm(x, mean = mean(originalData$population_density), sd =
sd(originalData$population_density))
       lines(x, curve, col = "darkblue", lwd = 2)

       # Bedrooms histogram with normal curve
       # For the first histogram (price distribution)
       hist(cleaned$price,
            prob = TRUE,  # Convert to probability density
            main = "Price Distribution After Scope Change ",
            xlab = "Price",
            col = "skyblue",
            border = "black",
            breaks = 50)

       # Add normal curve
       x <- seq(min(cleaned$price), max(cleaned$price), length = 100)
       curve <- dnorm(x, mean = mean(cleaned$price), sd = sd(cleaned$price))
       lines(x, curve, col = "darkblue", lwd = 2)

       hist(cleaned$bed,
            prob = TRUE,
            main = "Bed Distribution After Scope Change ",
            xlab = "Bedrooms",
            col = "skyblue",
            border = "black",
            breaks = 7)
       x <- seq(min(cleaned$bed), max(cleaned$bed), length = 100)
       curve <- dnorm(x, mean = mean(cleaned$bed), sd = sd(cleaned$bed))
       lines(x, curve, col = "darkblue", lwd = 2)

       # Bathrooms histogram with normal curve
       hist(cleaned$bath,
            prob = TRUE,
            main = "Bath Distribution After Scope Change ",
            xlab = "Bathrooms",
            col = "skyblue",
            border = "black",
            breaks = 7)
```

```
        x <- seq(min(cleaned$bath), max(cleaned$bath), length = 100)
        curve <- dnorm(x, mean = mean(cleaned$bath), sd = sd(cleaned$bath))
        lines(x, curve, col = "darkblue", lwd = 2)

        # Acre lot histogram with normal curve
        hist(cleaned$acre_lot,
             prob = TRUE,
             main = "Acre Lot Distribution After Scope Change ",
             xlab = "Acre Lot",
             col = "skyblue",
             border = "black",
             breaks = 15)
        x <- seq(min(cleaned$acre_lot), max(cleaned$acre_lot), length = 100)
        curve <- dnorm(x, mean = mean(cleaned$acre_lot), sd = sd(cleaned$acre_lot))
        lines(x, curve, col = "darkblue", lwd = 2)

        # House size histogram with normal curve
        hist(cleaned$house_size,
             prob = TRUE,
             main = "House Size Distribution After Scope Change ",
             xlab = "House Size (sq ft)",
             col = "skyblue",
             border = "black",
             breaks = 50)
        x <- seq(min(cleaned$house_size), max(cleaned$house_size), length = 100)
        curve <- dnorm(x, mean = mean(cleaned$house_size), sd = sd(cleaned$house_size))
        lines(x, curve, col = "darkblue", lwd = 2)

        # Population density histogram with normal curve
        hist(cleaned$population_density,
             prob = TRUE,
             main = "Population Density Distribution After Scope Change ",
             xlab = "Population Density",
             col = "skyblue",
             border = "black",
             breaks = 50)
        x <- seq(min(cleaned$population_density), max(cleaned$population_density), length = 100)
        curve <- dnorm(x, mean = mean(cleaned$population_density), sd =
sd(cleaned$population_density))
        lines(x, curve, col = "darkblue", lwd = 2)

        # Select variables
        vars <- cleaned[, c("house_size", "acre_lot", "bed", "bath", "population_density",
"price")]

        # Calculate correlation matrix
        cor_matrix <- cor(vars)

        # Create correlation plot
        corrplot(cor_matrix,
                 method = "color",        # Color squares
                 type = "upper",          # Show upper triangle
                 addCoef.col = "black",   # Add correlation coefficients
```

```
                tl.col = "black",        # Text label color
                tl.srt = 45,            # Rotate text labels
                col = colorRampPalette(c("indianred3", "white", "#3E6EFF"))(200), # Custom color
palette
                diag = TRUE)            # Show diagonal
       pairs(cleaned[c("price", "bed", "bath", "acre_lot", "house_size", "population_density")],
             pch = 16,          # Solid dots
             cex = .01,         # Point size
             col = "skyblue",      # Point color
             main = "Cross-Variable Relationships")

       # Create a 2x3 plotting layout
       par(mfrow = c(2, 3))

       # House Size vs Price
       plot(cleaned$house_size, cleaned$price,
            main = "House Size vs Price",
            xlab = "House Size (sq ft)",
            ylab = "Price",
            pch = 19,
            col = "skyblue")
       abline(lm(price ~ house_size, data = cleaned), col = "darkblue", lwd = 2)

       # Acre Lot vs Price
       plot(cleaned$acre_lot, cleaned$price,
            main = "Acre Lot vs Price",
            xlab = "Acre Lot",
            ylab = "Price",
            pch = 19,
            col = "skyblue")
       abline(lm(price ~ acre_lot, data = cleaned), col = "darkblue", lwd = 2)

       # Bedrooms vs Price
       plot(cleaned$bed, cleaned$price,
            main = "Bedrooms vs Price",
            xlab = "Number of Bedrooms",
            ylab = "Price",
            pch = 19,
            col = "skyblue")
       abline(lm(price ~ bed, data = cleaned), col = "darkblue", lwd = 2)

       # Bathrooms vs Price
       plot(cleaned$bath, cleaned$price,
            main = "Bathrooms vs Price",
            xlab = "Number of Bathrooms",
            ylab = "Price",
            pch = 19,
            col = "skyblue")
       abline(lm(price ~ bath, data = cleaned), col = "darkblue", lwd = 2)

       # Population Density vs Price
       plot(cleaned$population_density, cleaned$price,
            main = "Population Density vs Price",
            xlab = "Population Density",
            ylab = "Price",
            pch = 19,
            col = "skyblue")
       abline(lm(price ~ population_density, data = cleaned), col = "darkblue", lwd = 2)

       fullmodel <- lm(price ~ bed + bath + acre_lot + house_size + population_density , data =
cleaned)

       # Display fullmodel of the model
       summary(fullmodel)

       # Check for multicollinearity
       vif(fullmodel)

       # Create residuals vs fitted plot
       plot(fitted(fullmodel),
            residuals(fullmodel),
            pch = 16,           # Solid dots
            col = "skyblue",       # Blue points
            cex = 0.1,          # Point size
            main = "Residuals vs Fitted Values",
            xlab = "Fitted Values",
            ylab = "Residuals")

       # Add horizontal reference line at y=0
       abline(h = 0, col = "darkblue", lty = 2)
       # Create a data frame of different residuals and fitted values
       residuals_df <- data.frame(
         fitted = fitted(fullmodel),
         rstudent = rstudent(fullmodel),
         rstandard = rstandard(fullmodel),
         residuals = residuals(fullmodel)
       )

       # R Student Residuals Plot
       # Plot 1: R Student Residuals vs Fitted
       plot(residuals_df$fitted, residuals_df$rstudent,
            type = "p",
            pch = ".",
            ylim = c(-3, 3),
            main = "R Student Residuals vs Fitted Values",
            xlab = "Fitted Values",
            ylab = "R Student Residuals")
       abline(h = c(-3, 3), col = "darkblue", lty = 2)
       abline(h = 0, col = "skyblue", lty = 3)

       # Plot 2: Standardized Residuals vs Fitted
       plot(residuals_df$fitted, residuals_df$rstandard,
            type = "p",
            pch = ".",
            ylim = c(-4, 4),
            main = "Standardized Residuals vs Fitted Values",
     xlab = "Fitted Values",
     ylab = "Standardized Residuals")
abline(h = c(-3, 3), col = "darkblue", lty = 2)
abline(h = 0, col = "skyblue", lty = 3)


# Plot 3: Regular Residuals vs Fitted
plot(residuals_df$fitted, residuals_df$residuals,
     type = "p",
     pch = ".",
     main = "Regular Residuals vs Fitted Values",
     xlab = "Fitted Values",
     ylab = "Residuals")
abline(h = 0, col = "skyblue", lty = 3)

# Identify outliers using Cook's distance
cooks_d <- cooks.distance(fullmodel)
influential <- which(cooks_d > 4/length(cooks_d))
cleaned_subset <- cleaned[-influential, ]

# For your existing model
dfbeta_values <- dfbeta(fullmodel)


plot(cooks.distance(fullmodel),
     type="h",
     main="Cook's Distance",
     ylab="Cook's D",
     xlab="")
cooksThresh <- 4/nrow(cleaned)
abline(h=cooksThresh, col="red", lty=2)  # adds dashed red line at threshold
# Add zipcode labels for influential points
influential_cook <- which(cooks.distance(fullmodel) > 4/nrow(cleaned))
#text(influential_cook, cooks.distance(fullmodel)[influential_cook],
#    labels=cleaned$zip[influential_cook], pos=3, cex=0.7)

plot(rstandard(fullmodel),
     type="p",
     main="Standardized Residuals",
     ylab="Standardized Residuals",
     xlab="")
abline(h=c(-2,2), col="red", lty=2)
# Add zipcode labels for outliers
outliers_std <- which(abs(rstandard(fullmodel)) > 2)
#text(outliers_std, rstandard(fullmodel)[outliers_std],
#    labels=cleaned$zip[outliers_std], pos=3, cex=0.7)

hat_values <- hatvalues(fullmodel)
plot(hat_values,
     type="h",
     main="Hat Values",
     ylab="Hat Values",
     xlab="Index")
```

```
        # Add threshold line
        threshold <- 2*(6/nrow(cleaned))
        abline(h=threshold, col="red", lty=2)  # adds dashed red line at threshold

        # Add zipcode labels for high leverage points
        high_leverage <- which(hat_values > threshold)
        #text(high_leverage, hat_values[high_leverage],
        #labels=cleaned$zip[high_leverage], pos=3, cex=0.7)
        # Create Q-Q plot
        qqnorm(residuals(fullmodel),
               main="Q-Q Plot of Residuals (NORMAL MODEL)",
               pch=16,
               col="skyblue")
        qqline(residuals(fullmodel),
               col="darkblue",
               lwd=2)


        sqrtmodel <- lm(sqrt(price) ~ bed + bath + acre_lot + house_size + population_density, data
= cleaned)
        # Q-Q Plot
        qqnorm(residuals(sqrtmodel),
               main="Q-Q Plot of Residuals (SQRT MODEL)",
               pch=16,
               col="skyblue")
        qqline(residuals(sqrtmodel),
               col="darkblue",
               lwd=2)

        # Create residuals dataframe
        sqrtresiduals_df <- data.frame(
          fitted = fitted(sqrtmodel),
          rstudent = rstudent(sqrtmodel),
          rstandard = rstandard(sqrtmodel),
          residuals = residuals(sqrtmodel)
        )

        # R Student Residuals Plot
        plot(sqrtresiduals_df$fitted, sqrtresiduals_df$rstudent,
             type = "p",
             pch = ".",
             ylim = c(-3, 3),
             main = "R Student Residuals vs Fitted Values (Square Root Model)",
             xlab = "Fitted Values",
             ylab = "R Student Residuals")
        abline(h = c(-3, 3), col = "darkblue", lty = 2)
        abline(h = 0, col = "skyblue", lty = 3)

        plot(sqrtresiduals_df$fitted, sqrtresiduals_df$residuals,
             type = "p",
             pch = ".",
             main = "Regular Residuals vs Fitted Values (Square Root Model)",
             xlab = "Fitted Values",
             ylab = "Residuals")
        abline(h = 0, col = "skyblue", lty = 3)


        # Add horizontal reference line at y=0
        abline(h = 0, col = "darkblue", lty = 2)


        hist(a,
             prob = TRUE,  # Convert to probability density
             main = "Price Distribution",
             xlab = "Price",
             col = "blue",
             border = "black",
             breaks = 50)
        x <- seq(min(a), max(a), length = 100)
        curve <- dnorm(x, mean = mean(a), sd = sd(a))
        lines(x, curve, col = "skyblue", lwd = 2)

        sqrtmodel2 <- lm(sqrt(price) ~ bath + acre_lot + house_size + population_density, data =
 cleaned)

        summary(sqrtmodel)

        anova(sqrtmodel,sqrtmodel2)
```

6.2 References

[1] Comptsf. (2024, June 17). What determines the value of a home? Five factors that matter. Compass Mortgage. https://www.compmort.com/what-determines-the-value-of-a-home/#:~:text=Multiple%20factors%20determine%20a%20home's,market%20conditions%20and%20comparable%20sales.

[2] Sakib, A. S. (2024, March 30). USA Real Estate Dataset. Kaggle. https://www.kaggle.com/datasets/ahmedshahriarsakib/usa-real-estate-dataset