Price Prediction of Realtor.com Listings

Anveetha Suresh

2024-12-10

For further analysis, please view HomeValueLinearRegressionReport.pdf.

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.3.3
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.3.3
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.3.3
```

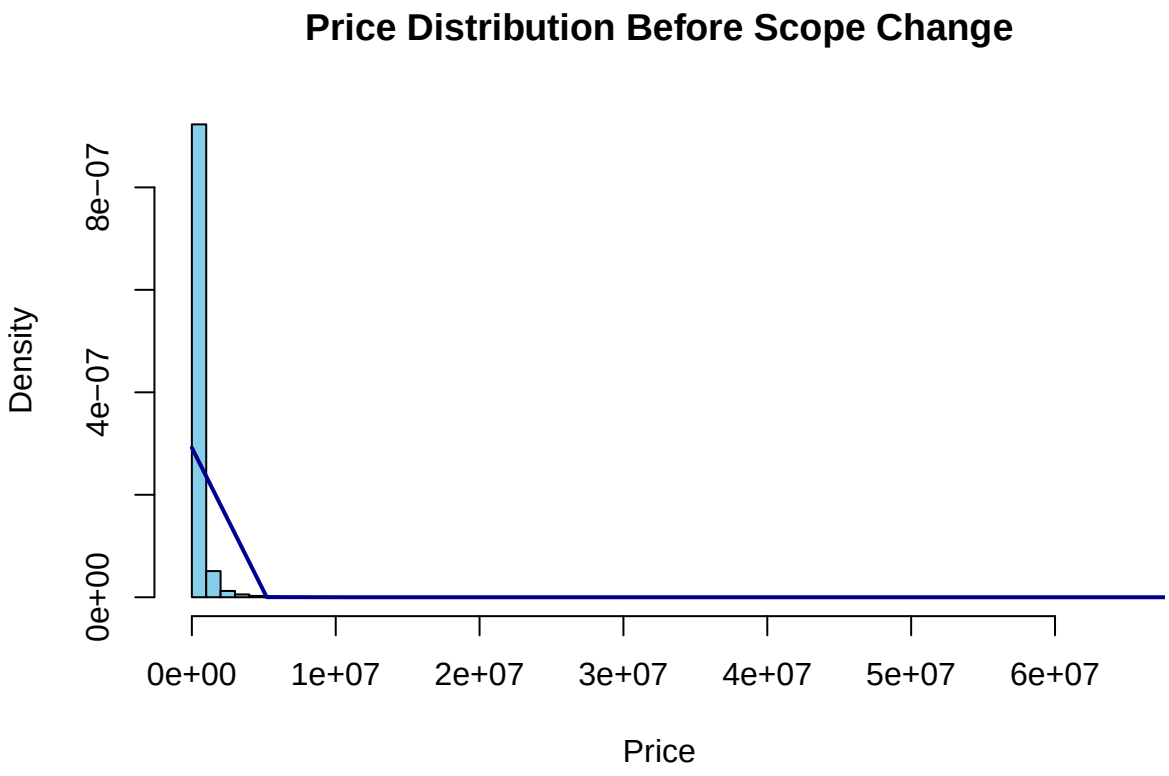
```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.3.3
```

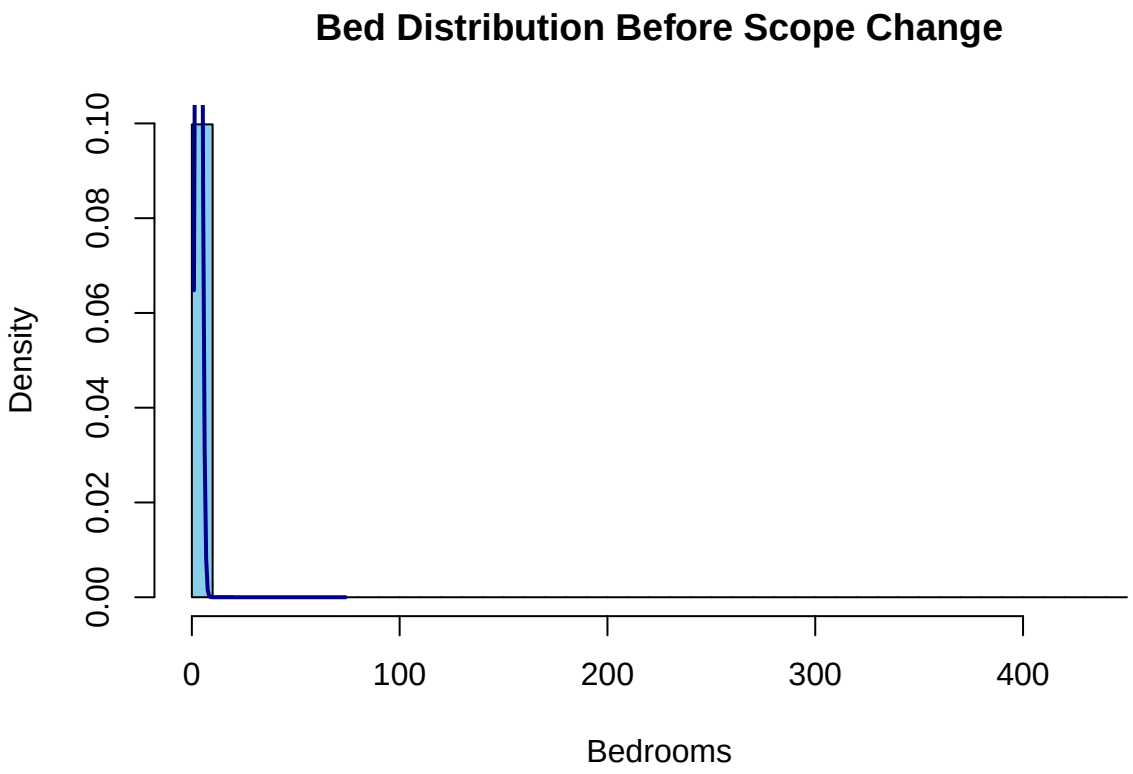
```
## corrplot 0.95 loaded
```

```r
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)
```

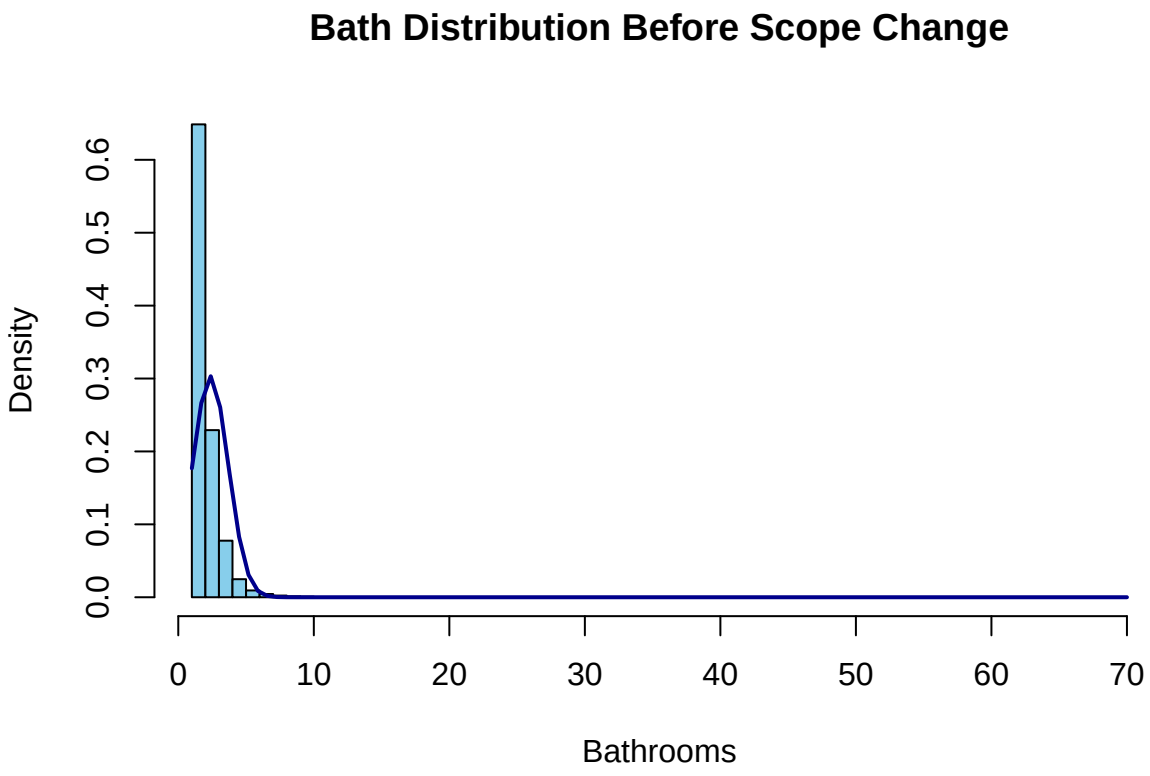
```r
hist(originalData$price,
     prob = TRUE,  # Convert to probability density
     main = "Price Distribution Before Scope Change",
     xlab = "Price",
     col = "skyblue",
     border = "black",
     breaks = 50)

# 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)
```



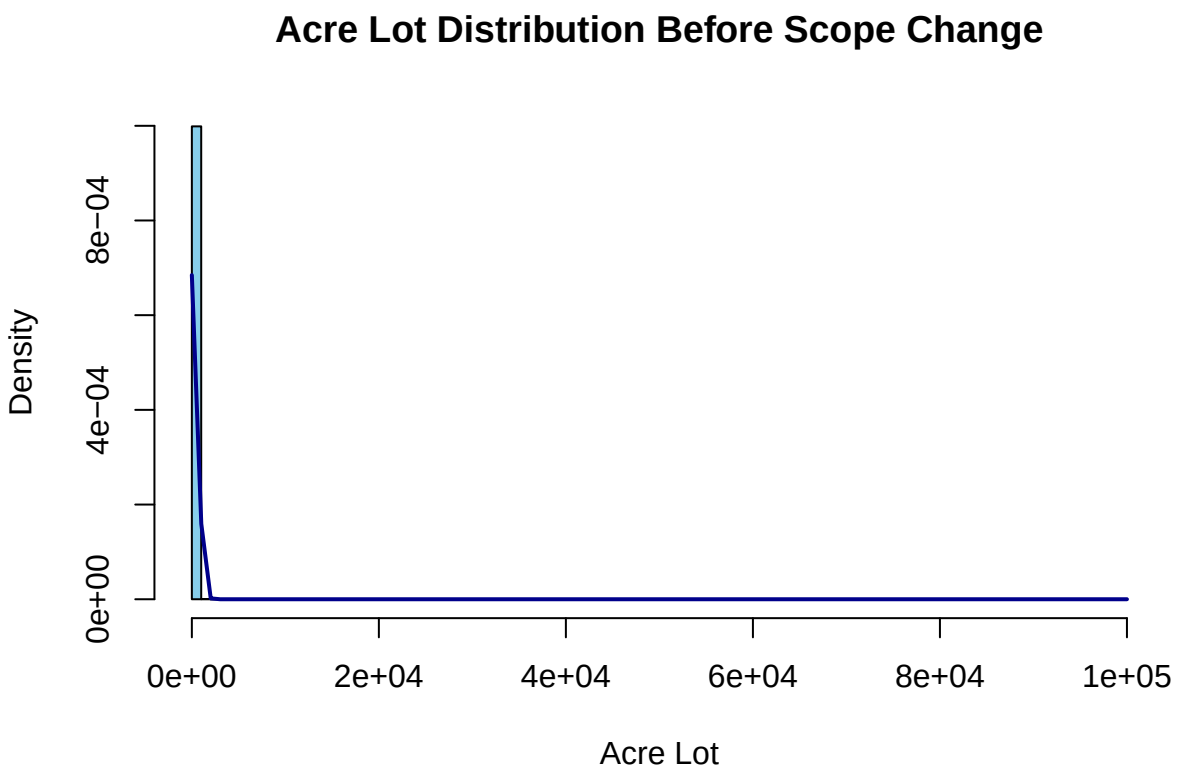

```r
hist(originalHomes$bed,
     prob = TRUE,
     main = "Bed Distribution Before Scope Change",
     xlab = "Bedrooms",
     col = "skyblue",
     border = "black",
     breaks = 50)
x <- seq(min(originalData$bed), max(originalData$bed), length = 100)
curve <- dnorm(x, mean = mean(originalData$bed),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



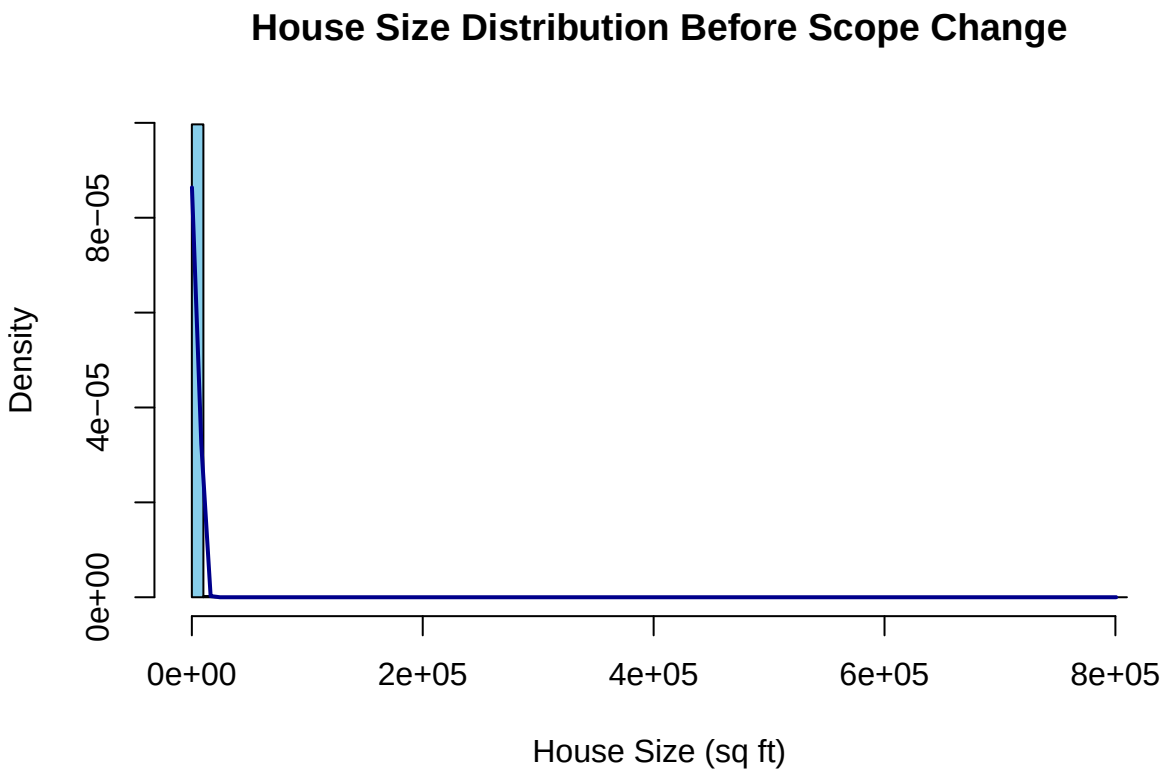

```r
# Bathrooms histogram with normal curve
hist(originalData$bath,
     prob = TRUE,
     main = "Bath Distribution Before Scope Change",
     xlab = "Bathrooms",
     col = "skyblue",
     border = "black",
     breaks =50)
x <- seq(min(originalData$bath), max(originalData$bath), length = 100)
curve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



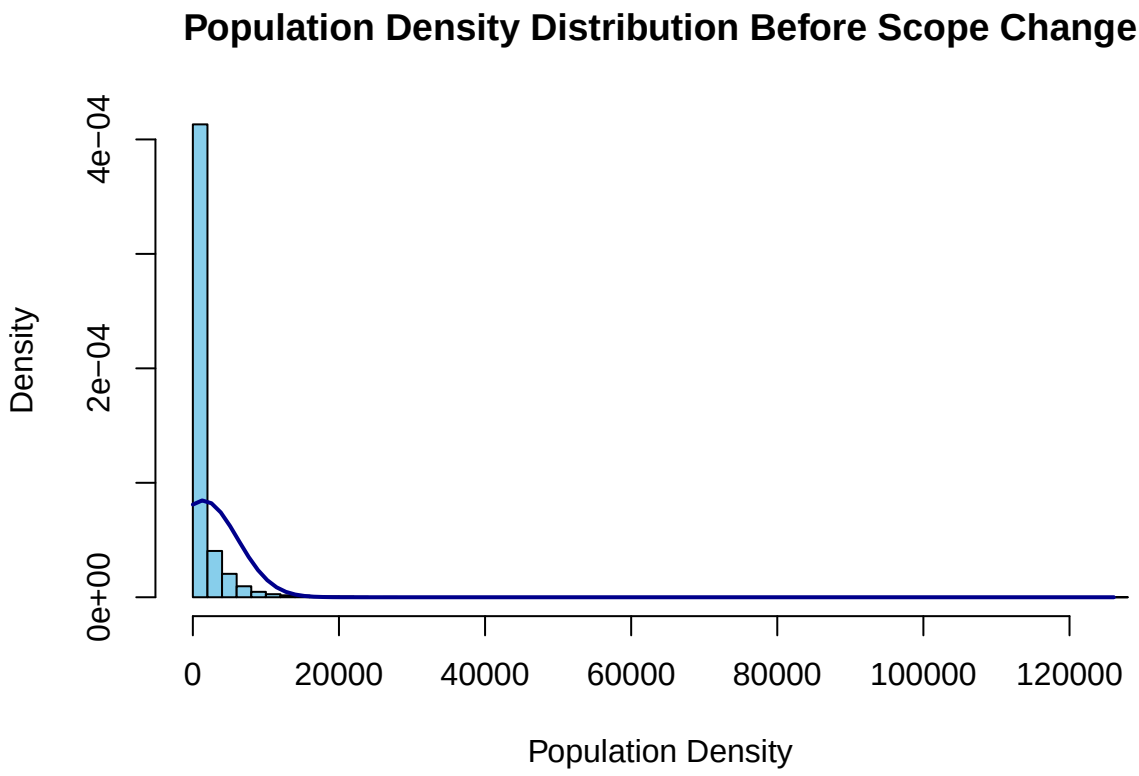

```r
# Acre lot histogram with normal curve
hist(originalData$acre_lot,
     prob = TRUE,
     main = "Acre Lot Distribution Before Scope Change",
     xlab = "Acre Lot",
     col = "skyblue",
     border = "black",
     breaks = 100)
x <- seq(min(originalData$acre_lot), max(originalData$acre_lot), length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



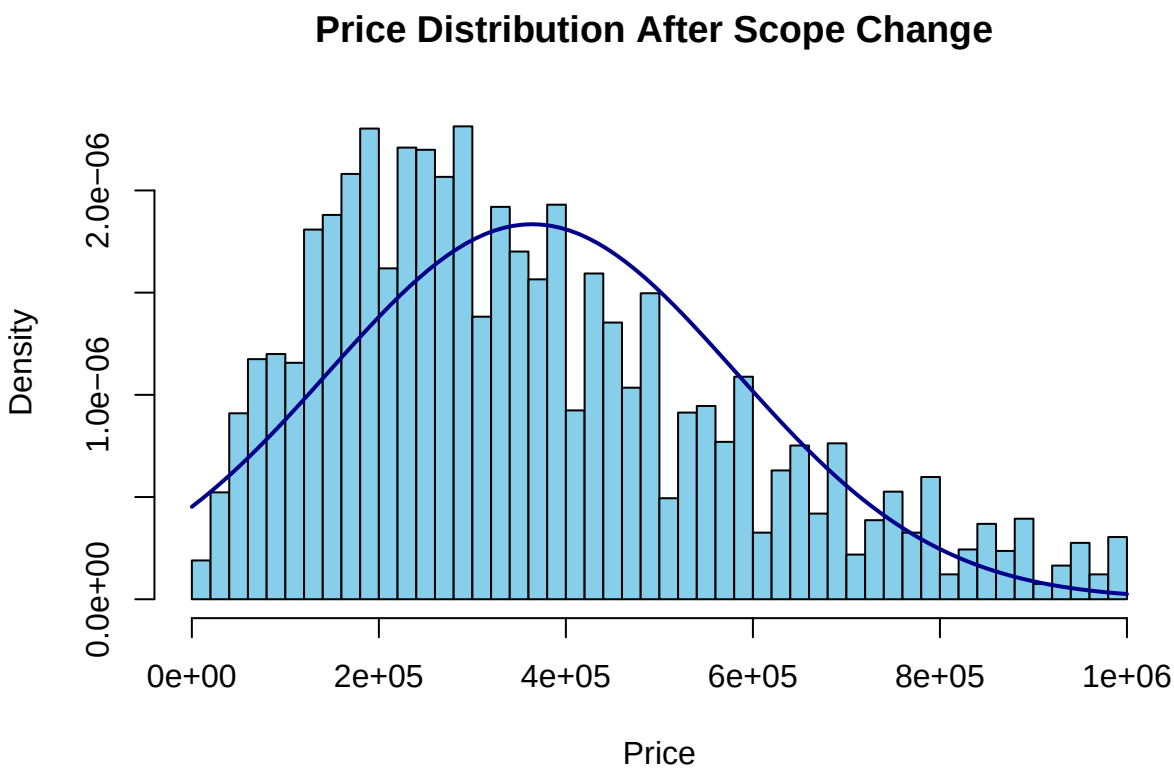

```r
# House size histogram with normal curve
hist(originalData$house_size,
     prob = TRUE,
     main = "House Size Distribution Before Scope Change",
     xlab = "House Size (sq ft)",
     col = "skyblue",
     border = "black",
     breaks = 100)
x <- seq(min(originalData$house_size), max(originalData$house_size),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



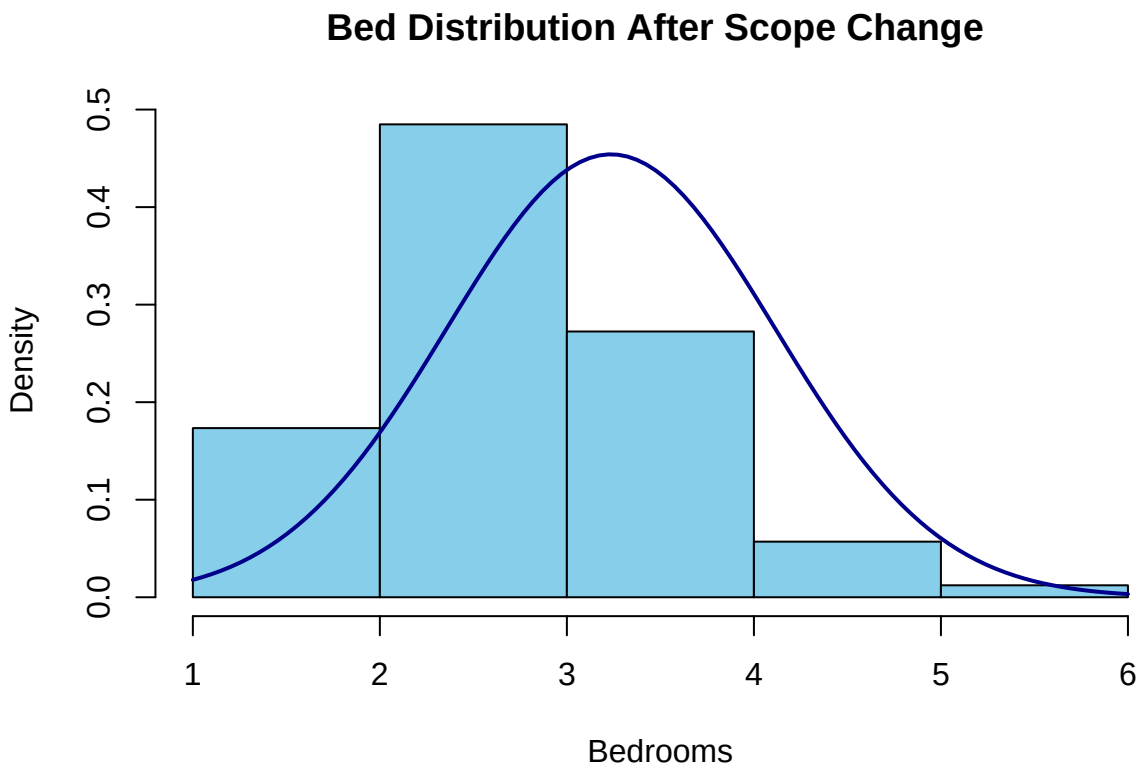

```r
# Population density histogram with normal curve
hist(originalData$population_density,
     prob = TRUE,
     main = "Population Density Distribution Before Scope Change",
     xlab = "Population Density",
     col = "skyblue",
     border = "black",
     breaks = 50)
x <- seq(min(originalData$population_density), 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)
```



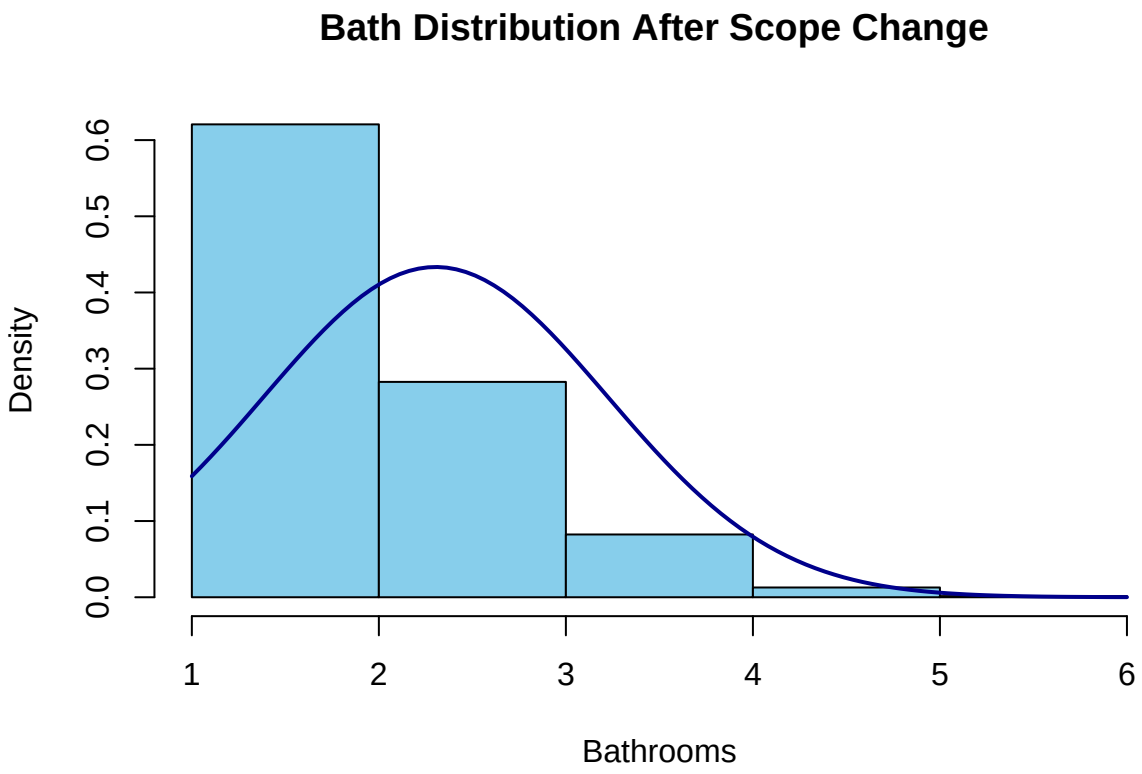

```r
# Bedrooms histogram with normal curve
# For the first histogram (price distribution)
hist(cleaned$price,
     prob = TRUE,  # Convert to probability density
     main = "Price Distribution After Scope Change ",
     xlab = "Price",
     col = "skyblue",
     border = "black",
     breaks = 50)

# 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)
```



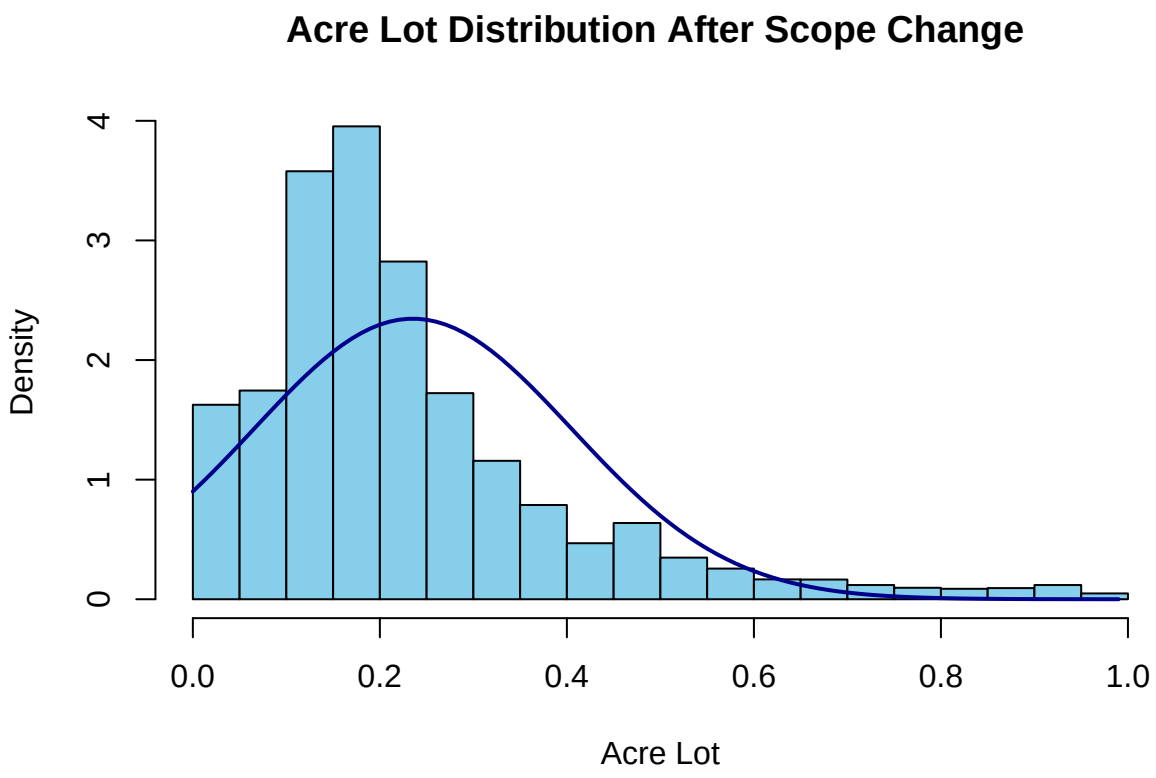

```r
hist(cleaned$bed,
     prob = TRUE,
     main = "Bed Distribution After Scope Change ",
     xlab = "Bedrooms",
     col = "skyblue",
     border = "black",
     breaks = 7)
x <- seq(min(cleaned$bed), max(cleaned$bed), length = 100)
curve <- dnorm(x, mean = mean(cleaned$bed), sd = sd(cleaned$bed))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



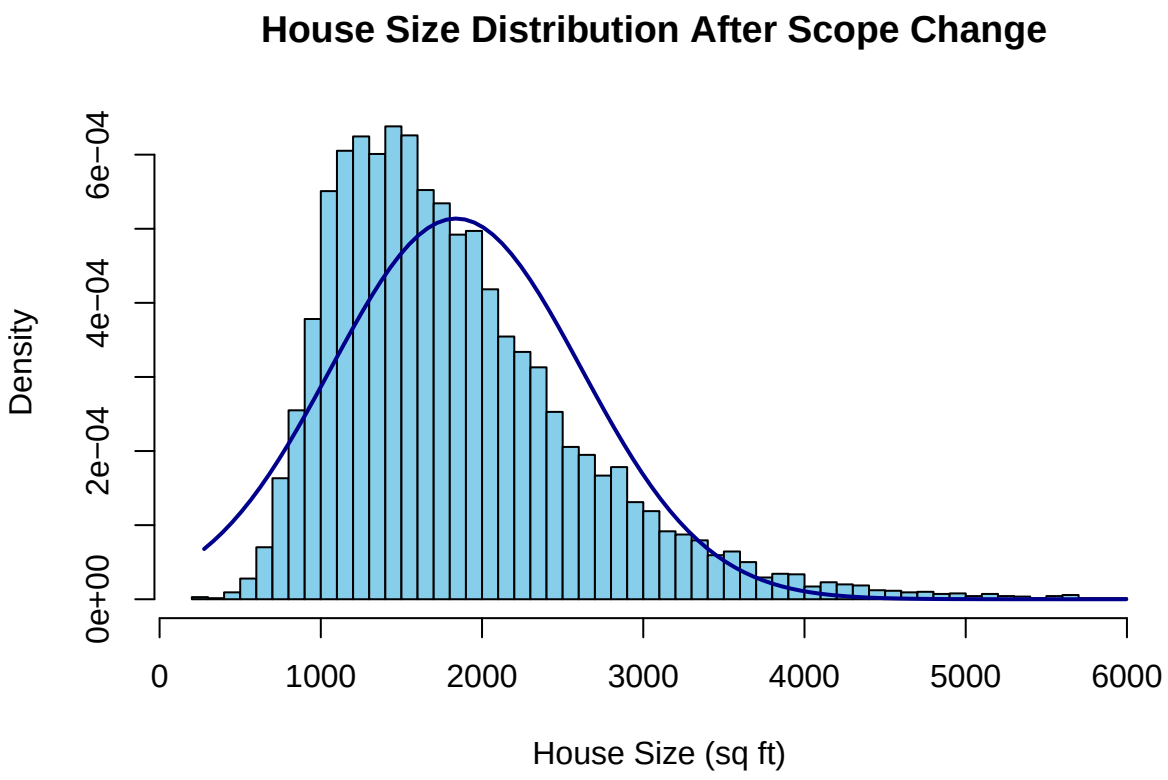

```
# Bathrooms histogram with normal curve
hist(cleaned$bath,
     prob = TRUE,
     main = "Bath Distribution After Scope Change ",
     xlab = "Bathrooms",
     col = "skyblue",
     border = "black",
     breaks = 7)
x <- seq(min(cleaned$bath), max(cleaned$bath), length = 100)
curve <- dnorm(x, mean = mean(cleaned$bath), sd = sd(cleaned$bath))
lines(x, curve, col = "darkblue", lwd = 2)
```



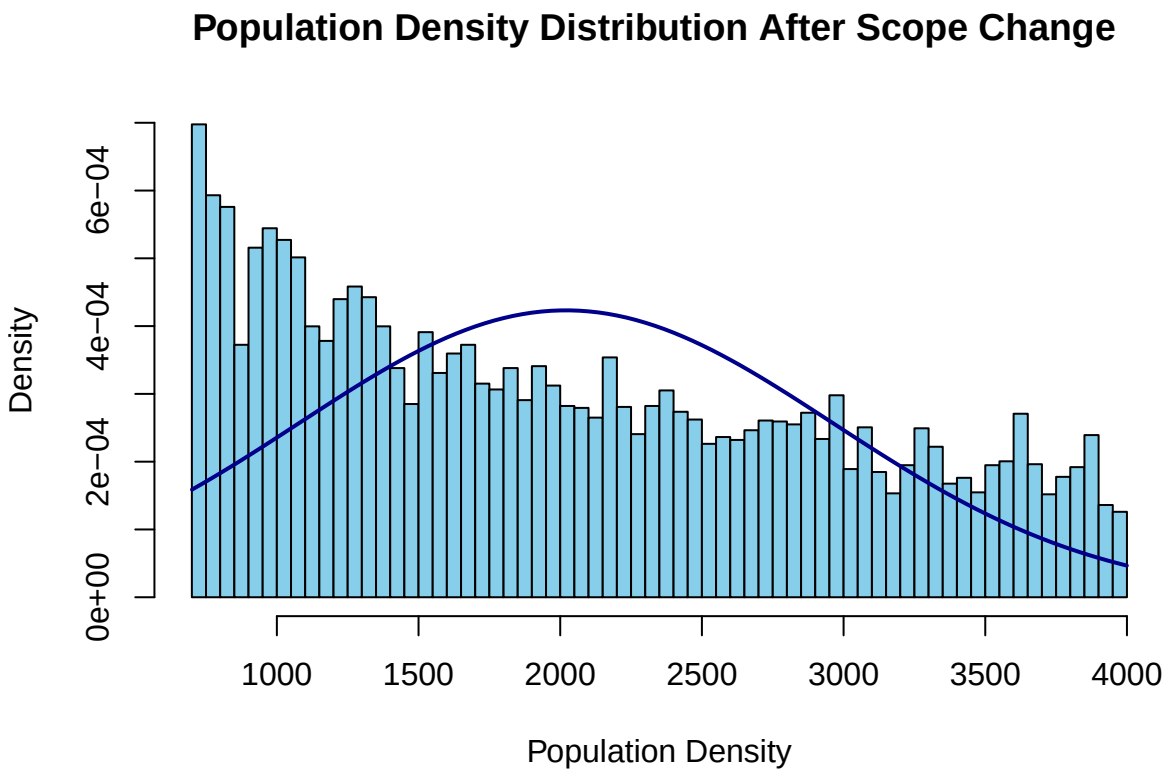

```
# Acre lot histogram with normal curve
hist(cleaned$acre_lot,
     prob = TRUE,
     main = "Acre Lot Distribution After Scope Change ",
     xlab = "Acre Lot",
     col = "skyblue",
     border = "black",
     breaks = 15)
x <- seq(min(cleaned$acre_lot), max(cleaned$acre_lot), length = 100)
curve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




```r
# 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)
```

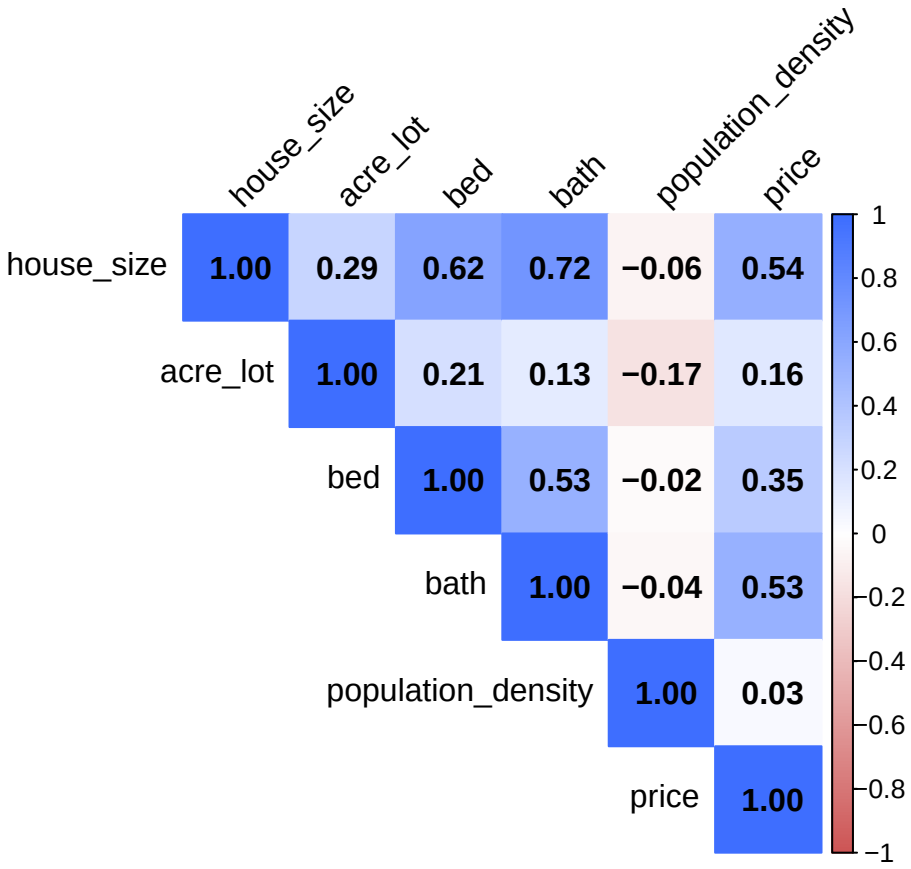



```r
# Population density histogram with normal curve
hist(cleaned$population_density,
     prob = TRUE,
     main = "Population Density Distribution After Scope Change ",
     xlab = "Population Density",
     col = "skyblue",
     border = "black",
     breaks = 50)
x <- seq(min(cleaned$population_density), 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)
```

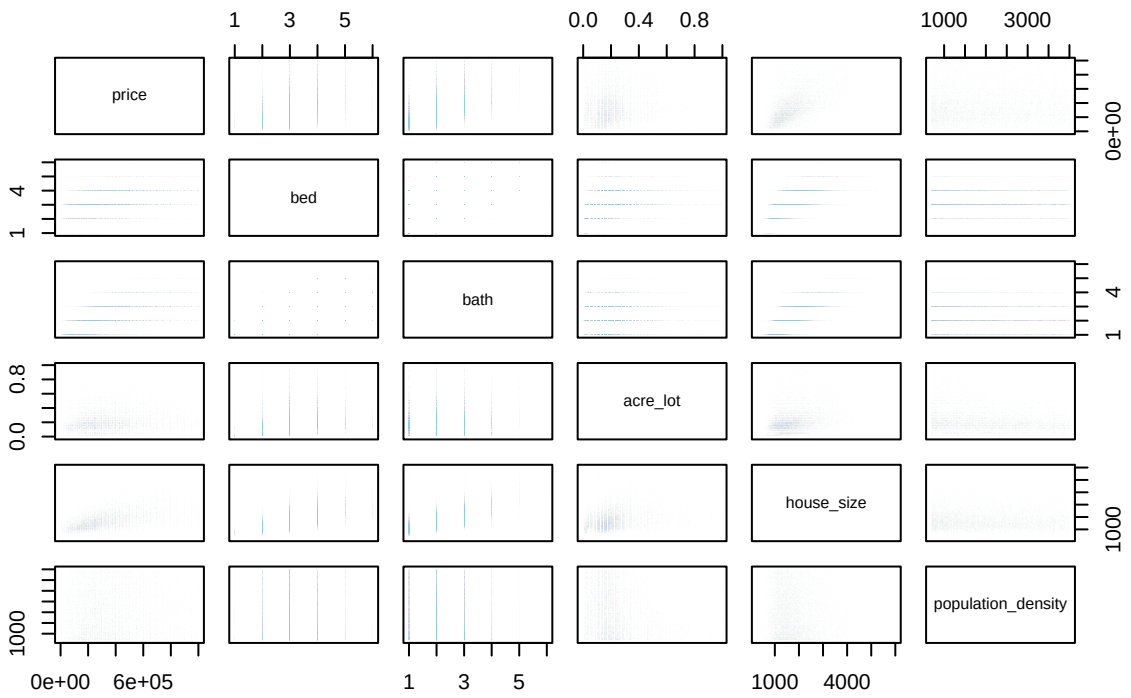



```
ggplot(cleaned, aes(x = status, y = price)) +
    geom_boxplot(fill = "skyblue", color = "black") + # Boxplot with colors
    theme_minimal() +                                 # Minimal theme for clean visuals
    labs(
      title = "Box-and-Whisker Plot of Property Sale Prices by Status",
      x = "Property Sale Status",
      y = "Sale Price"
    ) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotate x-axis labels if needed
```


```r
# 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
```

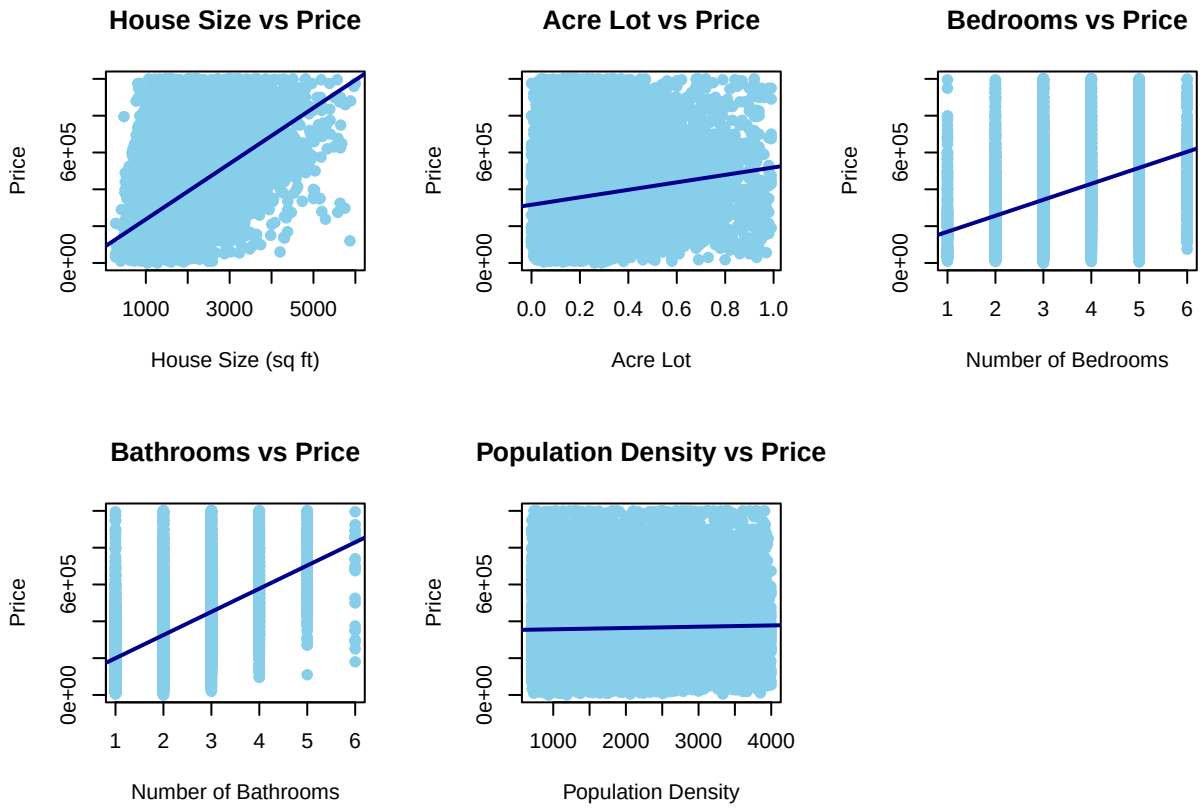



```r
pairs(cleaned[c("price", "bed", "bath", "acre_lot", "house_size", "population_density")],
      pch = 16,          # Solid dots
      cex = .01,         # Point size
      col = "skyblue",      # Point color
      main = "Cross-Variable Relationships")
```




```r
# 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)
```



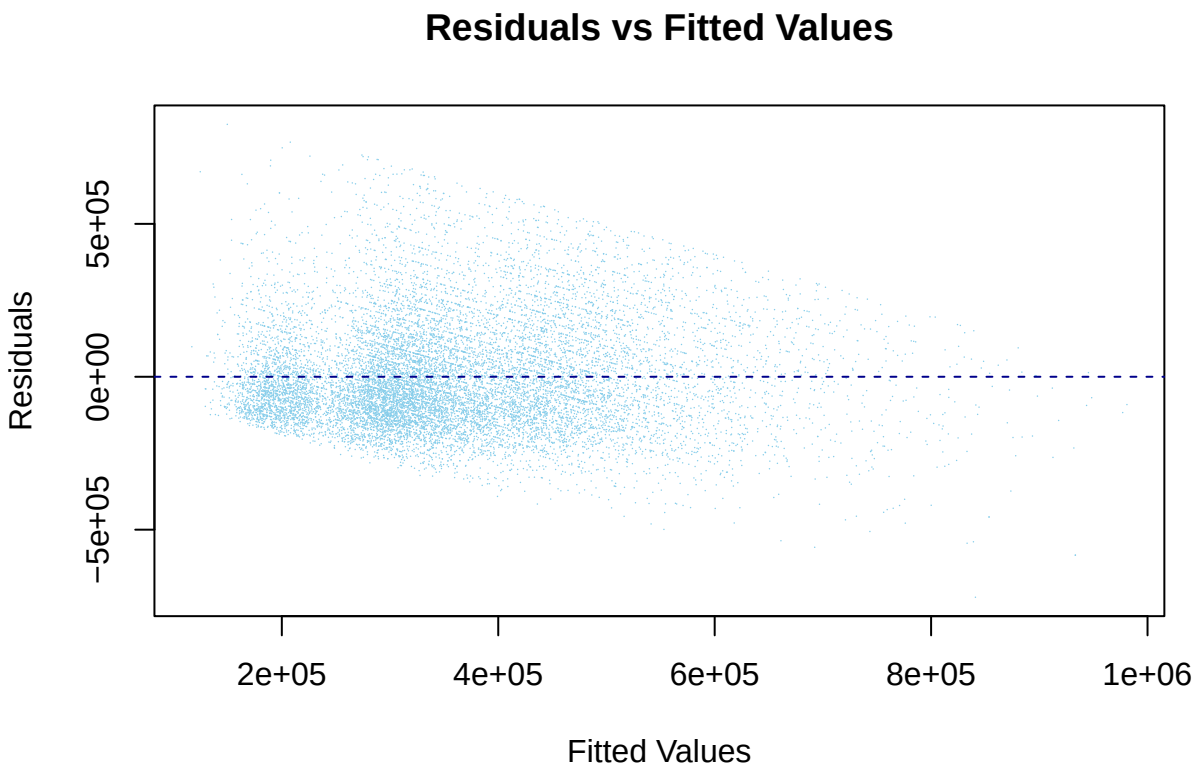

```
fullmodel <- lm(price ~ bed + bath + acre_lot + house_size + population_density , data = cleaned)

# Display fullmodel of the model
summary(fullmodel)
```

```
##
## Call:
## lm(formula = price ~ bed + bath + acre_lot + house_size + population_density,
##     data = cleaned)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -720960 -121128  -36262   91402  825408
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         2042.094   6886.337   0.297   0.7668
## bed                -5668.805   2206.874  -2.569   0.0102 *
## bath               72937.261   2389.168  30.528  < 2e-16 ***
## acre_lot           53016.341   9366.537   5.660 1.54e-08 ***
## house_size            90.832      3.129  29.027  < 2e-16 ***
## population_density    16.142      1.610  10.027  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 176600 on 13955 degrees of freedom
## Multiple R-squared:  0.3405, Adjusted R-squared:  0.3403
## F-statistic:  1441 on 5 and 13955 DF,  p-value: < 2.2e-16
```

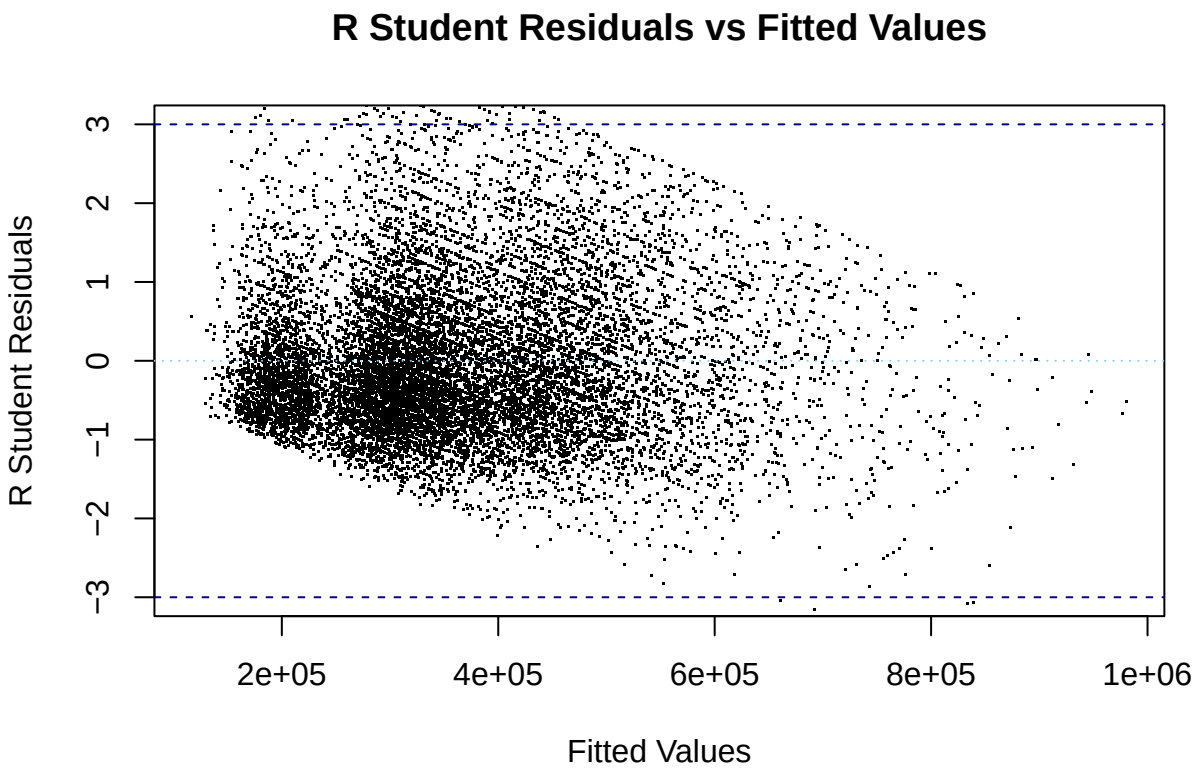
```r
# Check for multicollinearity
vif(fullmodel)
```

```
##                bed               bath           acre_lot         house_size
##           1.682335           2.164322           1.135805           2.642325
## population_density
##           1.030596
```

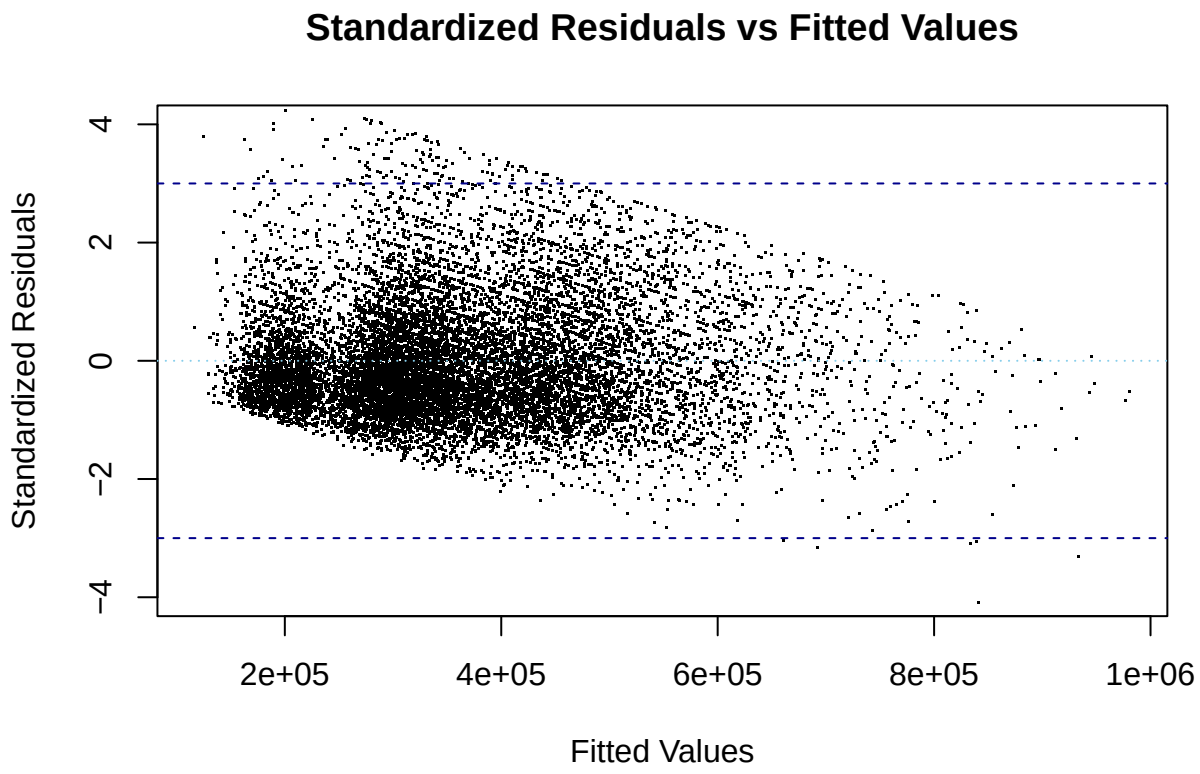
```r
# Create residuals vs fitted plot
plot(fitted(fullmodel),
     residuals(fullmodel),
     pch = 16,           # Solid dots
     col = "skyblue",       # Blue points
     cex = 0.1,          # Point size
     main = "Residuals vs Fitted Values",
     xlab = "Fitted Values",
     ylab = "Residuals")

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



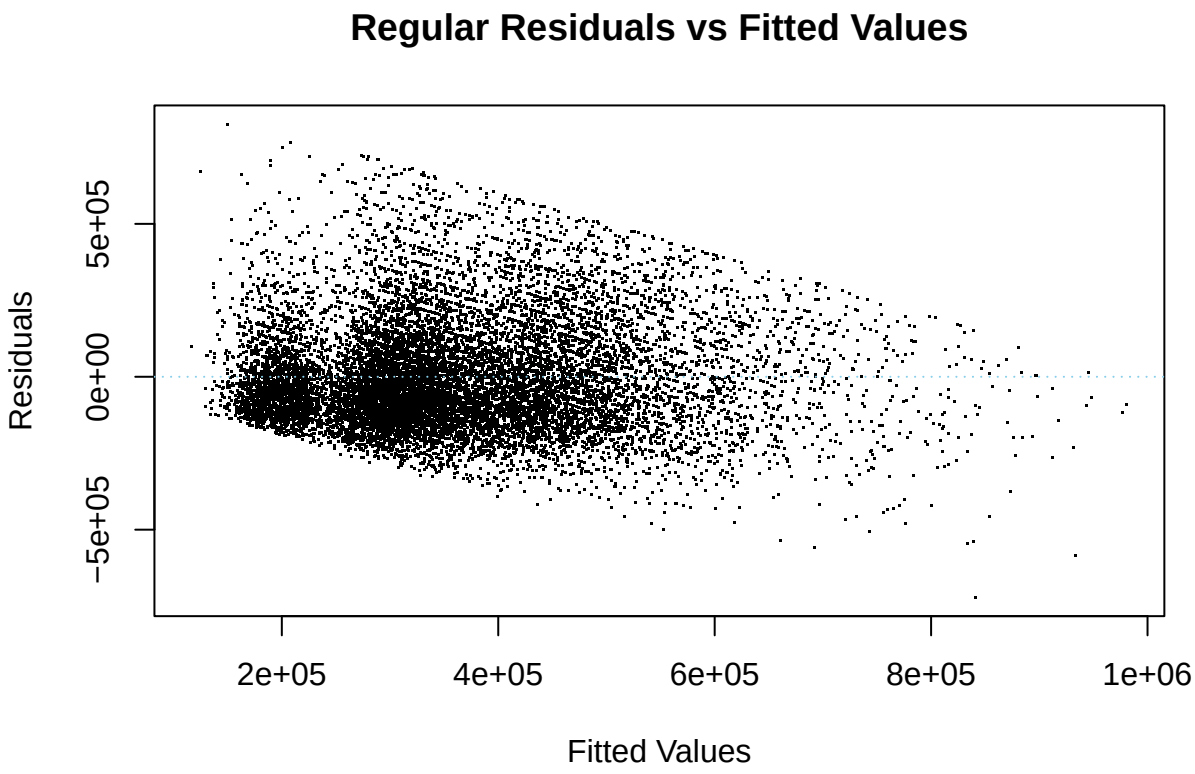

```r
# Create a data frame of different residuals and fitted values
residuals_df <- data.frame(
  fitted = fitted(fullmodel),
  rstudent = rstudent(fullmodel),
  rstandard = rstandard(fullmodel),
  residuals = residuals(fullmodel)
)

# 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)
```



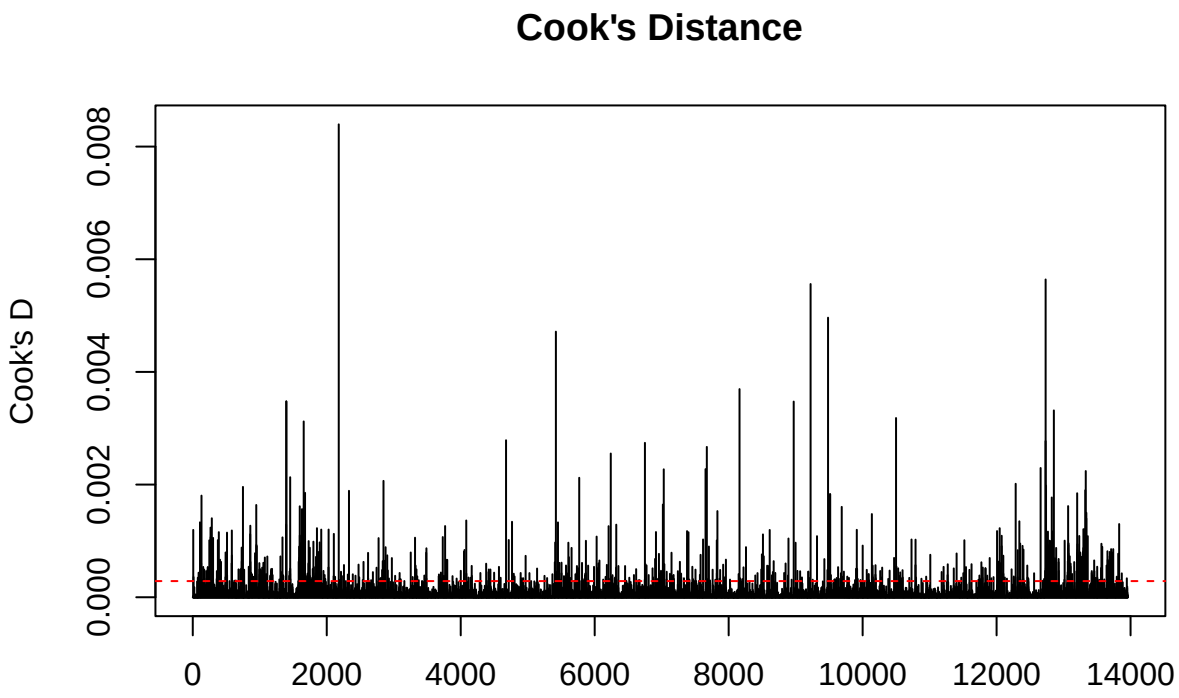

```r
# Plot 2: Standardized Residuals vs Fitted
plot(residuals_df$fitted, residuals_df$rstandard,
     type = "p",
     pch = ".",
     ylim = c(-4, 4),
     main = "Standardized Residuals vs Fitted Values",
     xlab = "Fitted Values",
     ylab = "Standardized Residuals")
abline(h = c(-3, 3), col =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



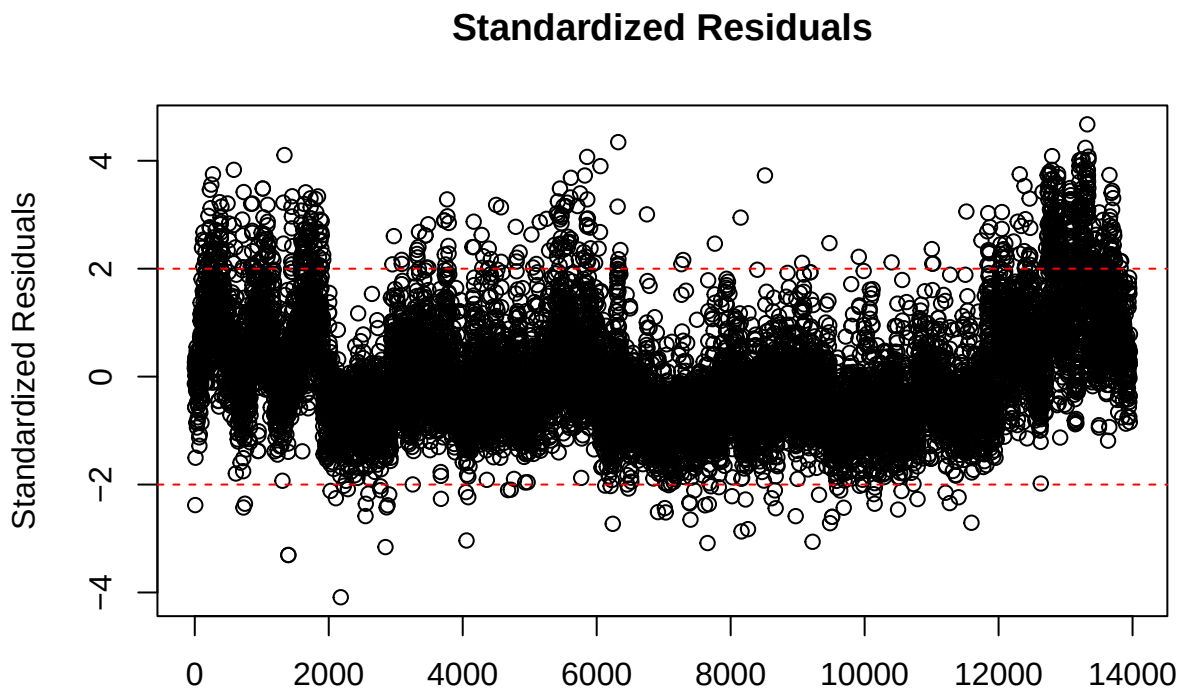

```r
# Plot 3: Regular Residuals vs Fitted
plot(residuals_df$fitted, residuals_df$residuals,
     type = "p",
     pch = ".",
     main = "Regular Residuals vs Fitted Values",
     xlab = "Fitted Values",
     ylab = "Residuals")
abline(h = 0, col = "skyblue", lty = 3)
```



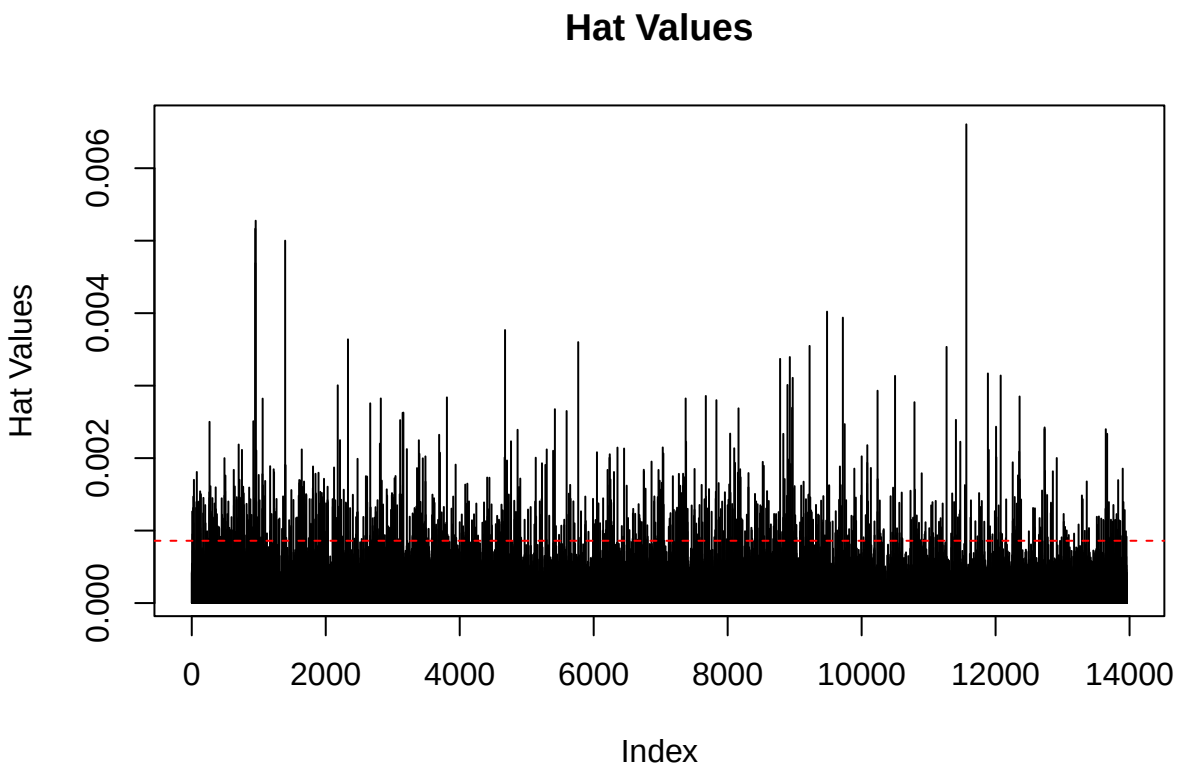

```r
# Identify outliers using Cook's distance
cooks_d <- cooks.distance(fullmodel)
influential <- which(cooks_d > 4/length(cooks_d))
cleaned_subset <- cleaned[-influential, ]

# 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
```



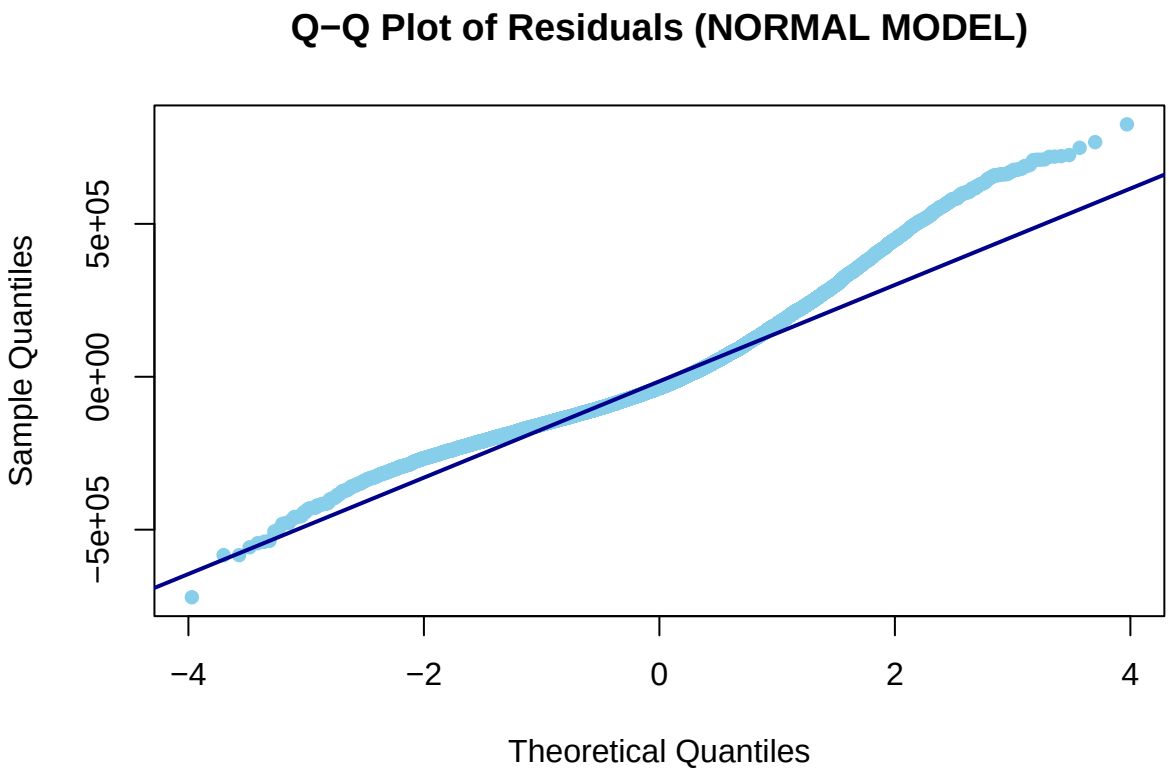

```r
# Add zipcode labels for influential points
influential_cook <- which(cooks.distance(fullmodel) > 4/nrow(cleaned))
#text(influential_cook, cooks.distance(fullmodel)[influential_cook],
#    labels=cleaned$zip[influential_cook], pos=3, cex=0.7)

plot(rstandard(fullmodel),
     type="p",
     main="Standardized Residuals",
     ylab="Standardized Residuals",
     xlab="")
abline(h=c(-2,2), col="red", lty=2)
```



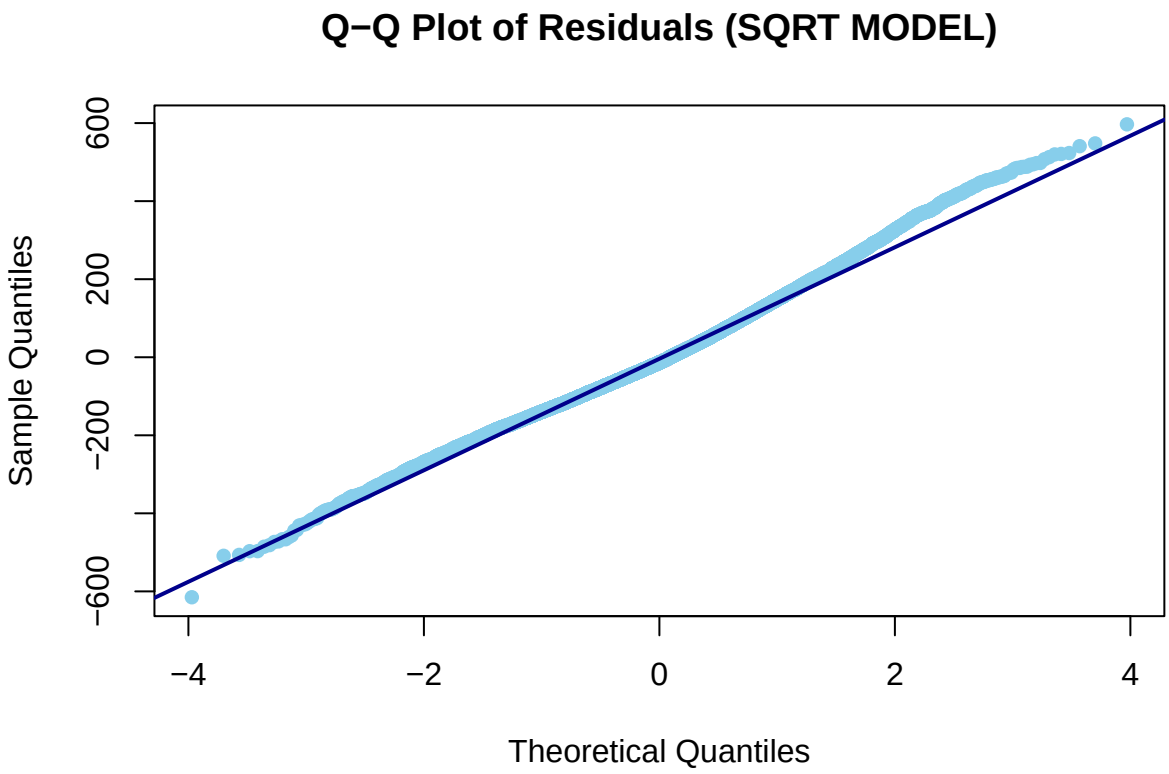

```r
# Add zipcode labels for outliers
outliers_std <- which(abs(rstandard(fullmodel)) > 2)
#text(outliers_std, rstandard(fullmodel)[outliers_std],
#    labels=cleaned$zip[outliers_std], pos=3, cex=0.7)

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
```



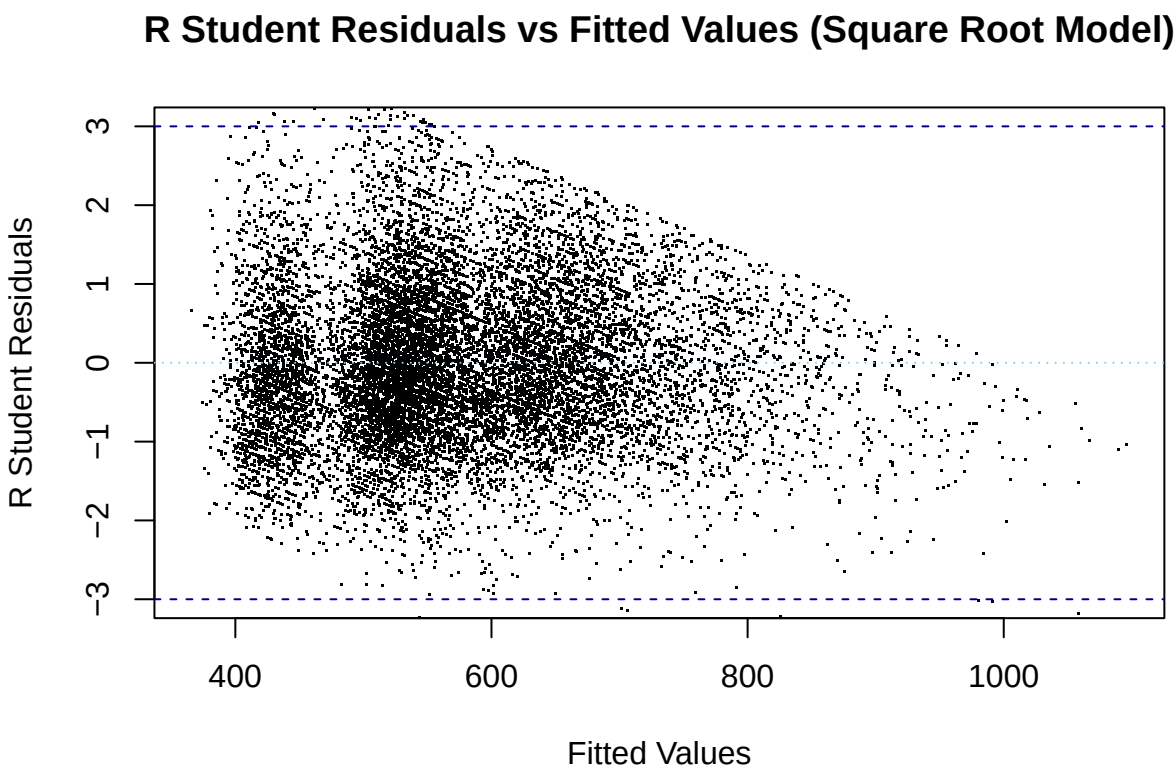

```r
# Add zipcode labels for high leverage points
high_leverage <- which(hat_values > threshold)
#text(high_leverage, hat_values[high_leverage],
#labels=cleaned$zip[high_leverage], pos=3, cex=0.7)
# Create Q-Q plot
qqnorm(residuals(fullmodel),
       main="Q-Q Plot of Residuals (NORMAL MODEL)",
   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



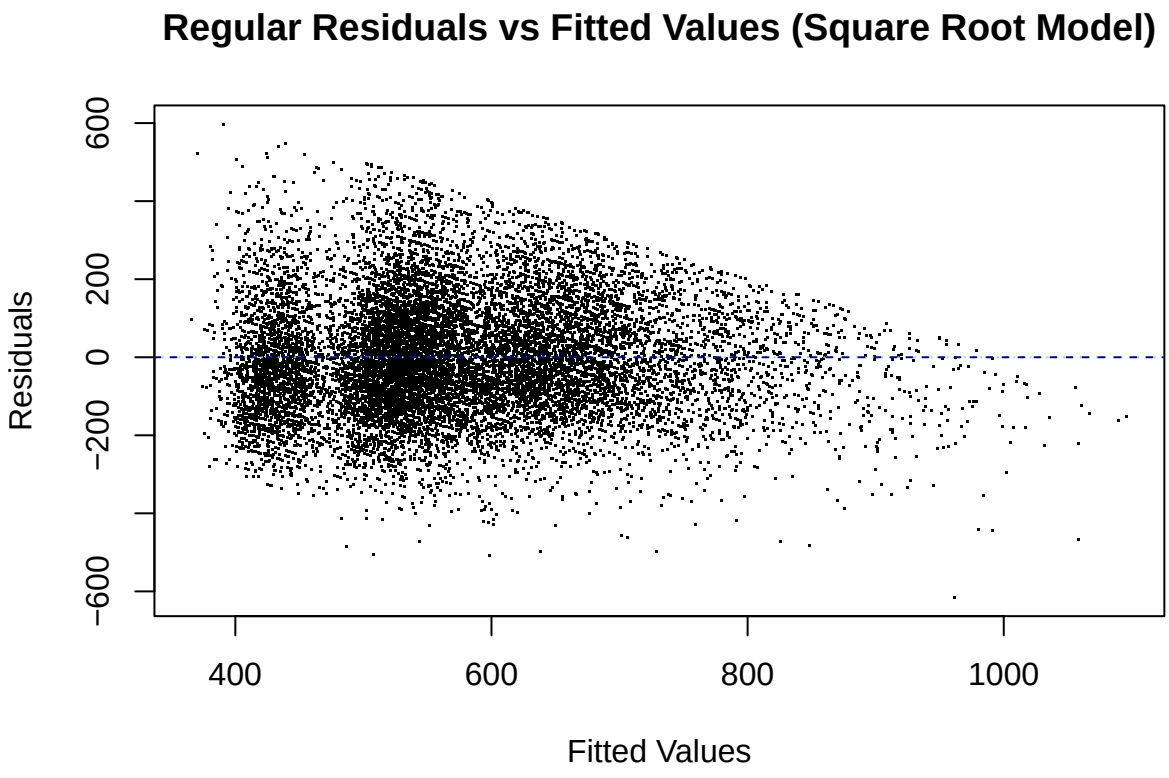

```
sqrtmodel <- lm(sqrt(price) ~ bed + bath + acre_lot + house_size + population_density, data = cleaned)
# Q-Q Plot
qqnorm(residuals(sqrtmodel),
       main="Q-Q Plot of Residuals (SQRT MODEL)",
       pch=16,
       col="skyblue")
qqline(residuals(sqrtmodel),
       col="darkblue",
       lwd=2)
```



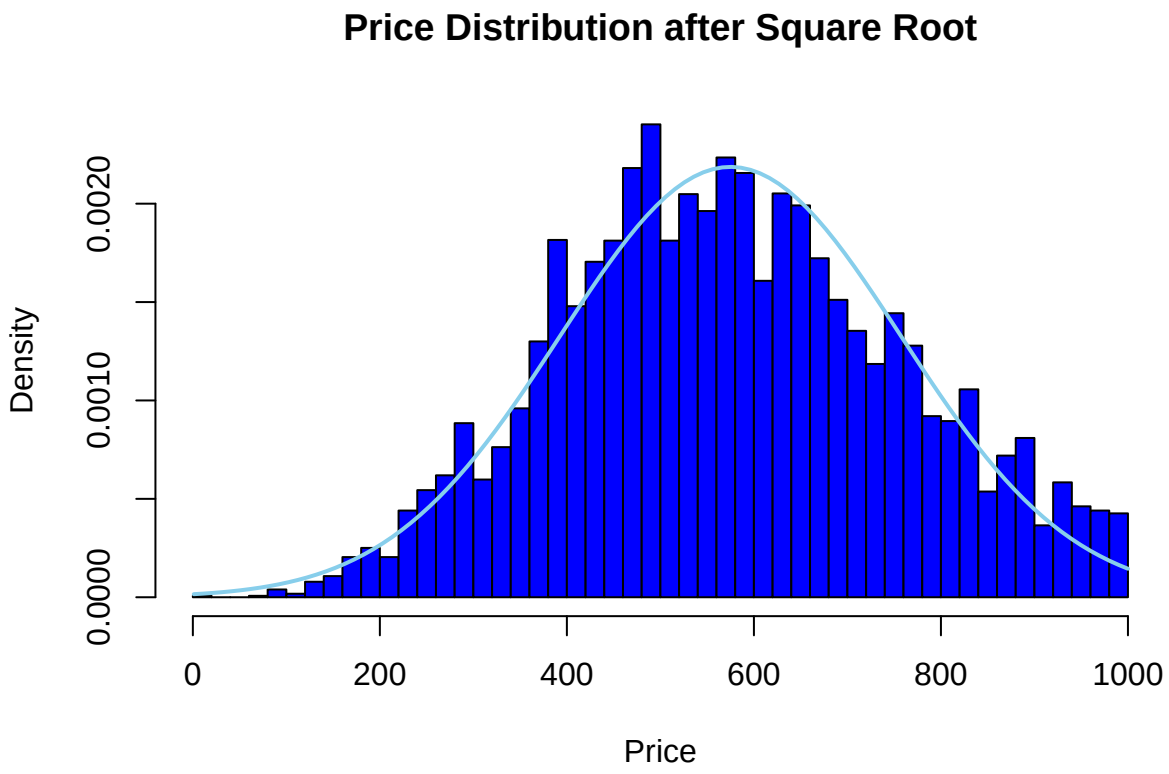

```r
# Create residuals dataframe
sqrtresiduals_df <- data.frame(
  fitted = fitted(sqrtmodel),
  rstudent = rstudent(sqrtmodel),
  rstandard = rstandard(sqrtmodel),
  residuals = residuals(sqrtmodel)
)

# 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)
```




```r
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)
```




```r
a <- sqrt(cleaned$price)
hist(a,
     prob = TRUE,  # Convert to probability density
     main = "Price Distribution after Square Root",
     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)
```

```
##
## Call:
## lm(formula = sqrt(price) ~ bed + bath + acre_lot + house_size +
##     population_density, data = cleaned)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -614.98 -100.26  -13.85   92.46  596.76
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        262.335970   5.722256  45.845  < 2e-16 ***
## bed                 -3.227337   1.833820  -1.760   0.0784 .
## bath                68.811450   1.985298  34.661  < 2e-16 ***
## acre_lot            47.003913   7.783198   6.039 1.59e-09 ***
## house_size           0.068580   0.002600  26.375  < 2e-16 ***
## population_density   0.013543   0.001338  10.123  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 146.8 on 13955 degrees of freedom
## Multiple R-squared:  0.3531, Adjusted R-squared:  0.3529
```

```
## F-statistic:  1523 on 5 and 13955 DF,  p-value: < 2.2e-16
```

```
# Using ANOVA to compare full and reduced model
anova(sqrtmodel,sqrtmodel2)
```

```
## 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  13955 300533308
## 2  13956 300600010 -1    -66702 3.0972 0.07845 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```