Housing Dataset Linear Regression Assignment

David Berberena

02-04-2024

Housing Dataset Linear Regression Assignment

```
# Assignment Start

# Upload readr library for file importing

library(readr)

# set working directory for smooth file importing

setwd("C:/Users/dbzda/Documents/School/DSC 520 Statistics for Data Science")

# Import the converted Housing CSV file to view its properties

housing <- read_csv("Housing.csv", show_col_types = FALSE)
```

1. Explain any transformations or modifications you made to the dataset.

In order to work with the dataset, I needed to omit the NA values present. While the na.omit() function in base R omits any row with a NA value present, it does not input any average or chosen values for the NA values, so the function inadvertently creates a subset of the dataset to work with, which could be rather small or large depending on the number of NA values present in the parent dataset. Making a linear regression model with the current “Sale Price” variable name prompts an error, so I will be changing the variable name using the names() function in base R so the error will be handled and the simple linear regression model can be created.

```
transformed_housing <- na.omit(housing)

names(transformed_housing)[names(transformed_housing) == "Sale Price"] <- "sale_price"
```

Simple Linear Regression Model

2. Create a linear regression model where “sq_ft_lot” predicts Sale Price.

```
# Accomplishing the creation of a simple linear regression model requires
# the use of the built-in lm() function

simple_housing_model <- lm(sale_price ~ sq_ft_lot, data = transformed_housing)
```

3, Get a summary of your first model and explain your results (i.e., R2, adj. R2, etc.)

```
summary(simple_housing_model)
```

```
##
## Call:
## lm(formula = sale_price ~ sq_ft_lot, data = transformed_housing)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -704178 -372433 -206483   22288 3674990
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.096e+05  2.807e+04  25.283   <2e-16 ***
## sq_ft_lot   -3.075e-01  1.438e+00  -0.214    0.831
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 767100 on 1106 degrees of freedom
## Multiple R-squared:  4.134e-05,  Adjusted R-squared:  -0.0008628
## F-statistic: 0.04572 on 1 and 1106 DF,  p-value: 0.8307
```

My results can be explained as the following:

Residuals: This statistic is meant to explain the error between the prediction of the model and the actual results in the original dataset. However, as the residual values are very large in range, this model is not a very good one to use a predictor for the sale price of a home based on the square footage of the lot.

Intercept: The model intercept estimate here is $709,600, meaning that when the square footage of the lot is 0, the sale price of a house is $709,600. This fact about the linear regression model doesn't make any sense in real life, yet that is what the regression model states.

Coefficients: The estimate for the linear regression model here is a negative number, indicating that with the increase of square footage, the sale price actually goes down. This completely destroys the validity of the model as that is not logically sound. The standard error shows how accurately the estimate was computed. 1.438 is relatively low, so the model's accuracy in finding the estimate is high. The t-value is very close to 0 and is negative, which suggests that there is not much statistical significance to the deviation from zero (supporting the null hypothesis) and that the relationship of the variables is negative, further supporting the above claim that the estimate carries. The Pr(>|t|) value, or p-value, signifies the probability that the observed model t-value when computed again will be as extreme as the one seen here. As the p-value is very close to 1, the probability of this is rather high, and shows that the model is accepting of the null hypothesis that the coefficient is 0.

Residual Standard Error: The figure listed in my results shows the standard deviation of the residuals and gives an estimate of how much the model's predicted values are different from the actual values. As the value is an extremely large number, this backs the previous evidence that this linear regression model is not a good fit for the data.

Multiple R-Squared: This statistic is supposed to show much much of the data the model explains regarding variance. As the value in this model is almost zero (0.00004134), not very much of the dataset's variance is explained by the model created.

Adjusted R-Squared: While this statistic here in a simple linear regression model doesn't mean much, this value helps determine whether adding addition predictor variables helps improve the model. Here in the model created, the value is negative and very close to zero, meaning that additional variables wouldn't help the model's performance much at all.

F-Statistic: Here is where the significance of the model is shown. The p-value is included once again as it also is a measure of model significance, and with the F-Statistic value being very low and close to zero, both values drive home the point that the simple linear regression model that has been created here is not a good fit for the data at all. The model has very little significance to the original dataset, as indicated by the high p-value in relation to its scale of 0 to 1, and the low F-Statistic value.

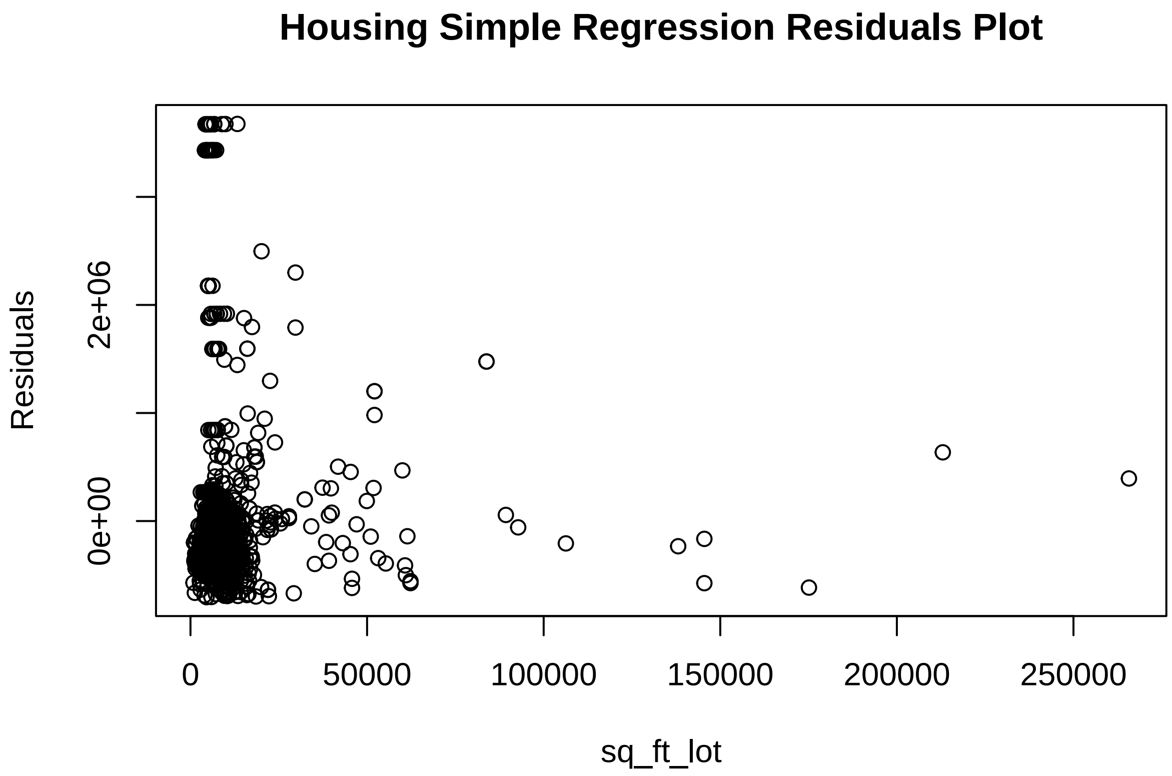
4. Get the residuals of your model (you can use 'resid' or 'residuals' functions) and plot them. What the does the plot tell you about your predictions?

```r
# Storing the residuals in a variable allows them to be plotted

residual_simple_model <- residuals(simple_housing_model)

# I am using the base R plot() function to plot the residuals with proper labeling

plot(transformed_housing$sq_ft_lot, residual_simple_model,
     main = "Housing Simple Regression Residuals Plot", xlab = "sq_ft_lot",
     ylab = "Residuals")
```

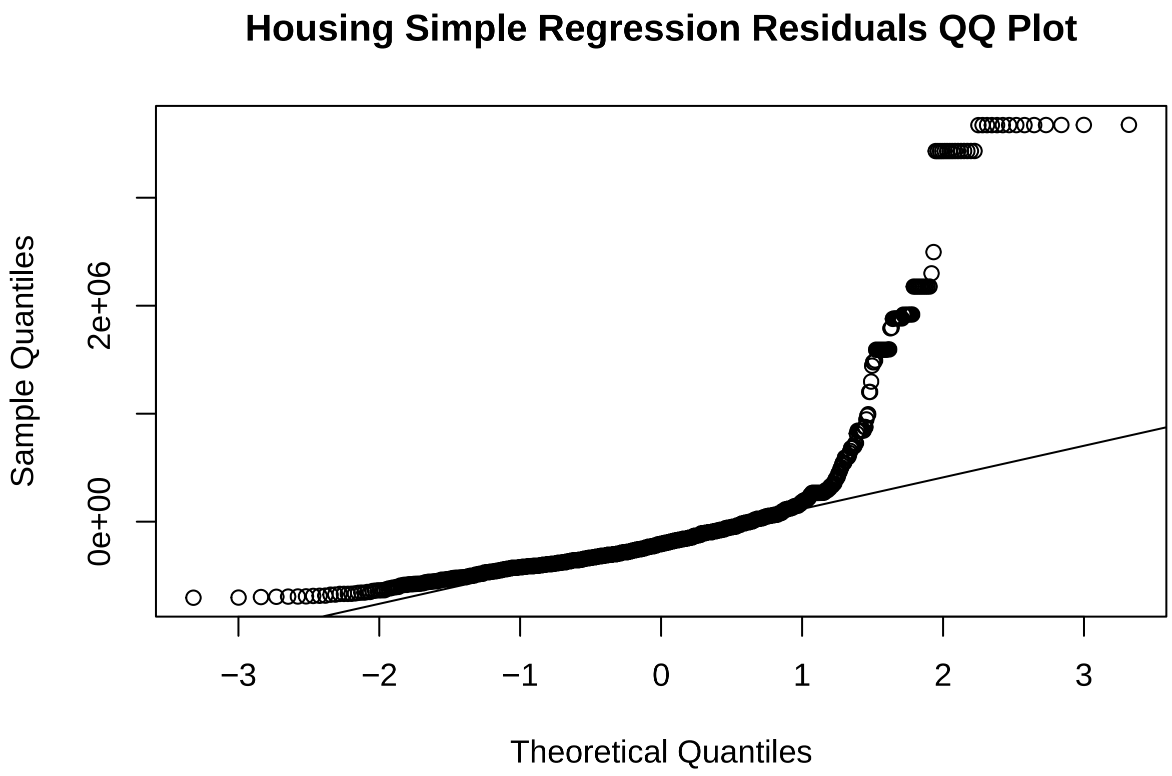


Looking at the residuals plot, we can see that there is not constant variance (even spread around the x-axis), indicating a problem in the model's fit to the data. We also can see that there is no linearity in the residuals. In a good fitting model, we should be observing a linear pattern in the residuals with those points being scattered around zero, yet in this current plot we are visualizing the residuals clustered around zero with outliers spreading across the plot with increased distance from one another. With this plot, we can assess that the model is not a good fit for the data.

5. Use a qq plot to observe your residuals. Do your residuals meet the normality assumption?

```
# Creating a qqplot involves the use of the qqnorm() function in base R and
# for a better visual, I will also use the qqline() function to add a
# line for reference

qqnorm(residual_simple_model, main = "Housing Simple Regression Residuals QQ Plot")
qqline(residual_simple_model)
```



The residuals as shown in the qq plot do not meet the normality assumption as the data points deviate heavily from the reference line after 1 on the x-axis. Normality assumption is visualized by seeing the residual data points keeping a constant linearity in step with the reference line with very little spread across that line. The qq plot here does not show the normality assumption, as it actually looks to be skewed to the left (negatively skewed).

6. Now, create a linear regression model that uses multiple predictor variables to predict Sale Price (feel free to derive new predictors from existing ones). Explain why you think each of these variables may add explanatory value to the model.

```
# Creation of multiple linear regression model using bedrooms and year_built

multiple_housing_model <- lm(sale_price ~ bedrooms +year_built, data = transformed_housing)
```

Thinking logically, using two more commonly looked at variables when it comes to the sale price of a home should yield more explanation to the model than the previous predictor variable. Most individuals shopping for a home (myself included) do not tend to think of the size of the lot first when browsing around. Most commercial home-buying websites do not even list that statistic, so hopefully these two more popular figures will formulate a better model.

7. Get a summary of your next model and explain your results.

```
summary(multiple_housing_model)
```

```
##
## Call:
## lm(formula = sale_price ~ bedrooms + year_built, data = transformed_housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1288113  -315862  -137541    45469  3459720
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -26594047    2140535 -12.424  < 2e-16 ***
## bedrooms       165993      24903   6.666 4.15e-11 ***
## year_built      13435       1088  12.343  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 691700 on 1105 degrees of freedom
## Multiple R-squared:  0.1878, Adjusted R-squared:  0.1863
## F-statistic: 127.7 on 2 and 1105 DF,  p-value: < 2.2e-16
```

The results for this linear regression model are more promising than the first model, as there are a few key differences to note. I will list each element of the results as I did above.

Residuals: The residuals for this multiple regression model are smaller than the simple linear regression model, indicating a better fit for the data than the first. However, the residuals are still large values, showing that even though this model is a better fit than the first, it still doesn't fit the data ideally.

Coefficients: For both bedrooms and year_built, the estimate is a positive number, which shows a positive relationship between both variable pairs involved (bedrooms and Sale Price and year_built and Sale Price). The standard error for both variables is a large positive number, indicating a low accuracy of the model finding the estimates for both variables. The t-values for bedrooms and year_built are positive and not too close to zero, and can explain the model's statistical significance to the deviation from zero (rejecting the null hypothesis). The values being positive for both variables supports the estimate showing a positive relationship between the variables. The p-values for both variables are extremely close to zero, suggesting that the null hypothesis of the coefficient being to zero should be rejected and that the chance of the multiple regression model computing an estimate as extreme as the one seen here is very low.

Residual Standard Error: This number is very large just like the simple regression model, meaning that the model is not a good fit for the data. Of course this residual standard error is a smaller value than the first model, so the model here is a slightly better fit than the simple model.

Multiple R-Squared: Not very much of the dataset's variance is explained by the model created as shown by the value being close to zero, yet a little more variance is explained here than in the simple model as this value is larger in the multiple regression model than in the simple model.

Adjusted R-Squared: As the addition of variables increases, this value that is close to zero shows that there wouldn't be much model performance enhancement. This value is larger than the previous model, which does show growth in the multiple regression model's performance.

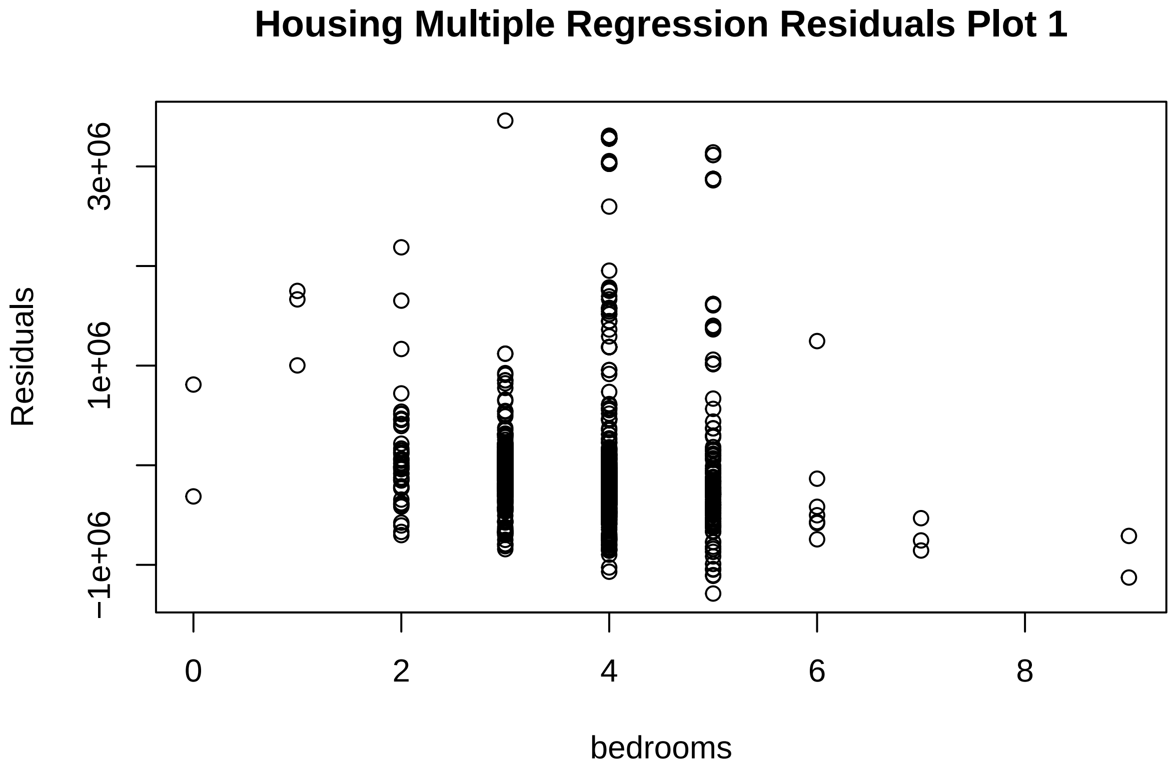
F-Statistic: The significance of the model is shown to be much higher here in the multiple regression model than in the simple model by the larger F-Statistic value and the p-value being extremely close to zero. While this statistic shows the model's significance, it does not combat the other statistics that show that this model is not a good fit for the data.

8. Get the residuals of your second model (you can use 'resid' or 'residuals' functions) and plot them. What the does the plot tell you about your predictions?

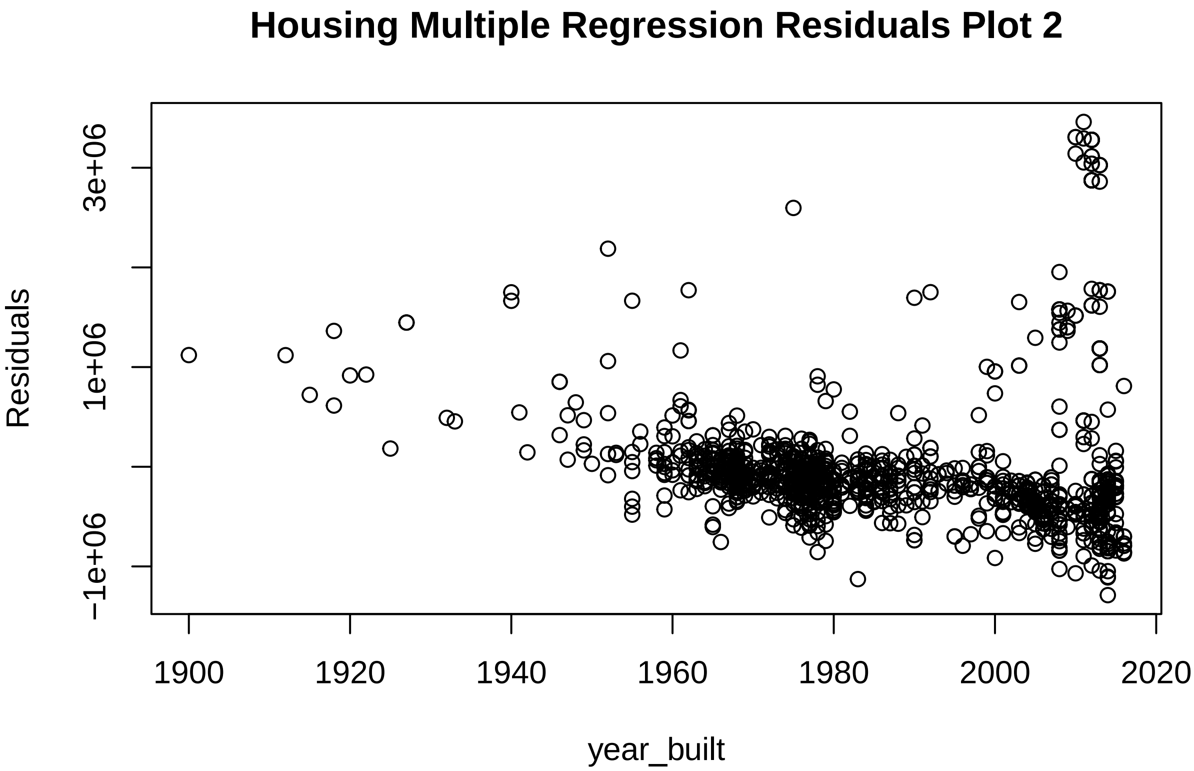
```r
residual_multiple_model <- residuals(multiple_housing_model)

# To plot both variables with the residuals of the model, two plots will need
# to be made (one for each independent variable)

plot(transformed_housing$bedrooms, residual_multiple_model,
     main = "Housing Multiple Regression Residuals Plot 1", xlab = "bedrooms",
     ylab = "Residuals")
```



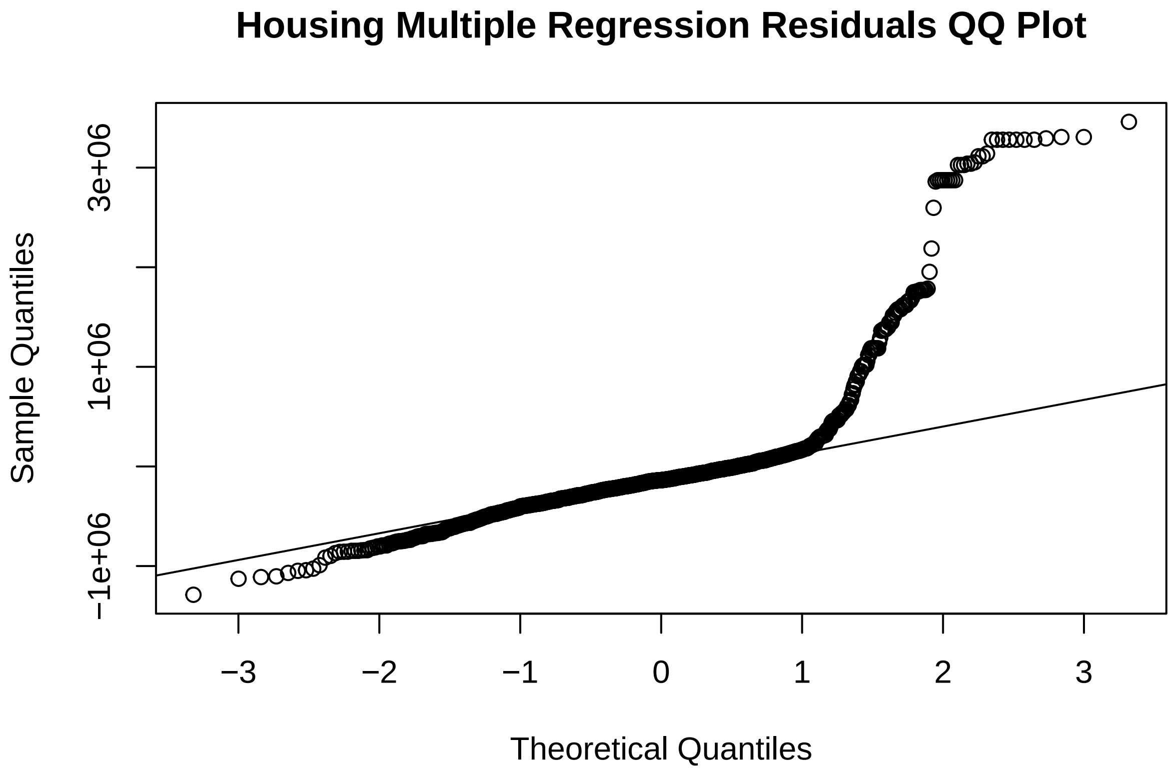
```r
plot(transformed_housing$year_built, residual_multiple_model,
     main = "Housing Multiple Regression Residuals Plot 2", xlab = "year_built",
     ylab = "Residuals")
```



For the bedrooms plot, the clear absence of scattered points around the x-axis, the absence of any linearity, and the presence of certain outlier points shows that the model is not a good fit for the data. Concerning the year_built plot, this one is much more promising. There seems to be some linearity in the data points as well as some consistent variance. While this plot is in show indicative of an extremely good fit for the data, it is the best looking residuals plot we have seen thus far.

9. Use a qq plot to observe your residuals. Do your residuals meet the normality assumption?

```r
qqnorm(residual_multiple_model, main = "Housing Multiple Regression Residuals QQ Plot")
qqline(residual_multiple_model)
```



Just like the simple regression QQ plot, the residuals as shown in this qq plot do not meet the normality assumption as the data points deviate heavily from the reference line after 1 on the x-axis almost identically to the first QQ plot. Normality assumption is visualized by seeing the residual data points keeping a constant linearity in step with the reference line with very little spread across that line. The qq plot here does not show the normality assumption, as it actually looks to be skewed to the left (negatively skewed), following in the same footsteps as its simple regression predecessor. It is very intriguing that both models do not follow the normality assumption in almost the exact same way.

10. Compare the results (i.e., R2, adj R2, etc) between your first and second model. Does your new model show an improvement over the first? To confirm a 'significant' improvement between the second and first model, use ANOVA to compare them. What are the results?

Looking back at my explanation of the multiple regression model results in the seventh question, almost every statistical value shows an improvement over the simple regression model. R-Squared, Adjusted R-Squared, Standard Error, and F-Statistics all show a slightly better performance, fit, and significance.

```
anova_test <- anova(multiple_housing_model)
anova_test
```

```
## Analysis of Variance Table
##
## Response: sale_price
##              Df     Sum Sq    Mean Sq F value    Pr(>F)
## bedrooms      1 4.9337e+13 4.9337e+13  103.12 < 2.2e-16 ***
## year_built    1 7.2889e+13 7.2889e+13  152.34 < 2.2e-16 ***
## Residuals  1105 5.2869e+14 4.7845e+11
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Seeing the ANOVA test results for the multiple linear regression model, the p-value is the measurement of statistical significance. With the value being almost zero, we can most likely reject the null hypothesis and consider the model to be statistically significant, as the chance that the model will predict an estimate as large as the current one is extremely low.

11. After observing both models (specifically, residual normality), provide your thoughts concerning whether the model is biased or not.

The model created is biased when reviewing the QQ plots, due to the large deviation of data points from the residuals reference line after 1 on the x-axis. This spread of the data points shows the lack of constant variance and nonlinearity, supporting the idea of the model being biased.

12. Another important aspect of regression tasks is determining the accuracy of your predictions. For this section, we will look at root mean square error (RMSE), a common accuracy metric for regression models.

- Install the 'Metrics' package in R Studio
- Using the first model, we will make predictions on the dataset using the predict function. An example would look like this (will vary for you based on variable names): – 'preds <- predict(object = modelName, newdata = dataset)' – Use the 'rmse' function to get RMSE for the model ('rmse(actual, predicted)')

A. What is the RMSE of the first model?

```
# Load the Metrics package

library(Metrics)

# Making predictions for the simple linear regression model

predicted_simple_model <- predict(object = simple_housing_model,
                                  newdata = transformed_housing)

# Obtaining the RMSE from the actual versus predicted model values for sq_ft_lot

rmse(transformed_housing$sq_ft_lot, predicted_simple_model)
```

```
## [1] 695388.3
```

B. Perform the same task for the second model. Provide the RMSE for the second model.

```
# Making predictions for the multiple linear regression model

predicted_multiple_model <- predict(object = multiple_housing_model,
                                    newdata = transformed_housing)

# Obtaining the RMSE from the actual versus predicted model values for both
# bedrooms and year_built requires two RMSE values to be calculated

rmse(transformed_housing$bedrooms, predicted_multiple_model)
```

```
## [1] 780411.7
```

```
rmse(transformed_housing$year_built, predicted_multiple_model)
```

```
## [1] 778610.6
```

C. Did the second model's RMSE improve upon the first model? By how much?

In this case, the simple linear regression model's RMSE value was better than the multiple regression model's two RMSE values. A lower RMSE value suggests less distance between predicted and observed values. The simple model's RMSE value is more than 80,000 lower than both multiple model's RMSE values. Now this may seem like the simple model is a better fit for the data, and this is true according to the story this statistic alone tells. However, all of the RMSE values are extremely high, indicating that even though the simple model looks better than the multiple model, they both are not a good fit for the data.