Week 8 Exercise 8.2

Javier Corpus

2025-02-01

Exercise 8.2

```r
# Importing the required libraries

library(ggplot2) # To create plots
library(readxl) # To read the Excel file
library(Metrics) # To calculate RMSE
library(dplyr, warn.conflicts = FALSE) # To transform the dataframe

# Load the dataset
df <- read_excel("week-6-housing.xlsx")
```

Filling blank city names based on zip code.

```r
# Fill city name (if blank) based on zip code:
df <- df %>%
  mutate(ctyname = ifelse(ctyname == "" & zip5 == "98052", "REDMOND",
                          ifelse(ctyname == "" & zip5 == "98053", "REDMOND",
                                 ifelse(ctyname == "" & zip5 == "98059",
"RENTON",
                                        ifelse(ctyname == "" & zip5 ==
"98074", "SAMMAMISH", ctyname)))))
```

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

Filled out missing city names

Using the `mutate` and `ifelse` functions, the blank city names were filled based on the zip code.

Added to new columns

For the second model, two new predictors were derived:

- price_sq_ft_lot - Price of the lot per square feet. This was obtained dividing `Sale Price` by `sq_ft_lot`.
- price_sq_total_living - Price of the living space per square feet. This was obtained dividing `Sale Price` by `square_feet_total_living`.

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

```
model01 <- lm(`Sale Price` ~ sq_ft_lot, data = df)
```

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

```
summary(model01)

##
## Call:
## lm(formula = `Sale Price` ~ sq_ft_lot, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2016064  -194842   -63293    91565  3735109
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.418e+05  3.800e+03  168.90   <2e-16 ***
## sq_ft_lot   8.510e-01  6.217e-02   13.69   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 401500 on 12863 degrees of freedom
## Multiple R-squared:  0.01435,    Adjusted R-squared:  0.01428
## F-statistic: 187.3 on 1 and 12863 DF,  p-value: < 2.2e-16
```

Interpretation of summary results for model 1

For this model, the value of `R-squared` is 0.01435. Because there is only one predictor, this value represents the square of the simple correlation between the size of the lot and the price. This means that the variation in lot size can account only for 1.43% of the variation in sales price. There must be other variables that influence 98.57% of the variations in sale price.

`Adjusted R-squared`: 0.01428. Since this model only has one predictor, the value of the Adjusted R-squared (0.01428 in this example) also means that only about 1.43% of the variations in sale prize can be explained by the variation in the size of the lot.

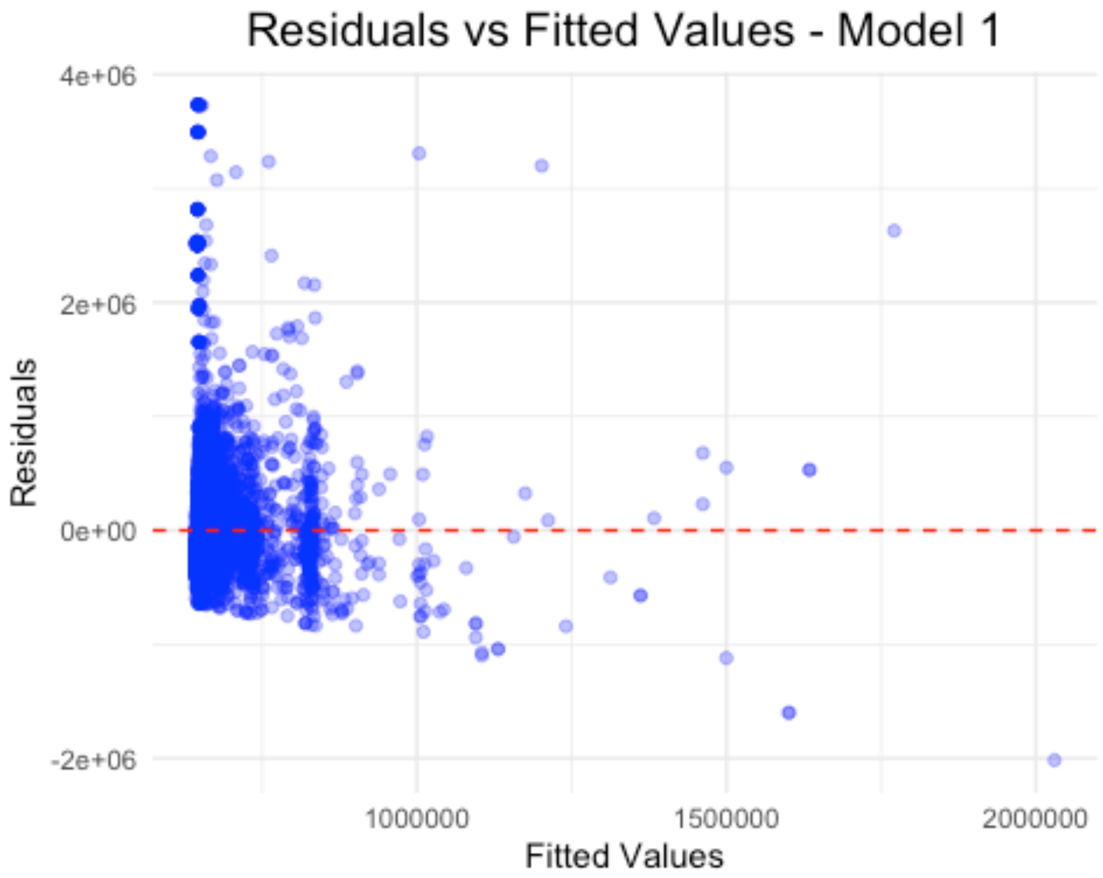
The `F-statistic` of 187.3 is the ratio of variance explained by the model to the variance not explained by the model.

The `Degrees of Freedom` (DF, 1 and 12863) indicate the number of predictors (1) and the residual degrees of freedom. This is the total of observations minus the number of predictors minus 1. In other words: `12865 - 1 - 1 = 12863`

Finally, the p-value of < 2.2e-16 is extremely small. This means that the probability of observing an F-statistic of 187.3 under the null hypothesis is extremely low.

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?

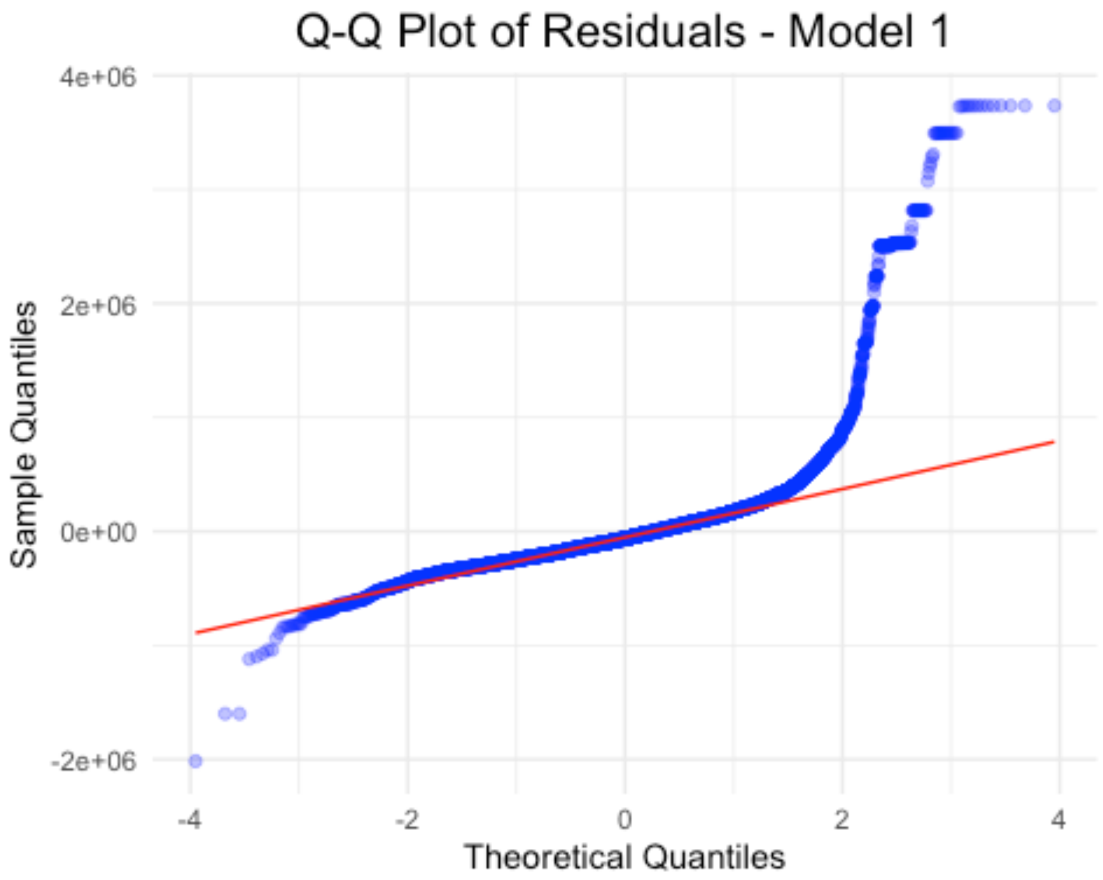
```
residuals01 <- residuals(model01)
fitted_values01 <- fitted(model01)

ggplot(df, aes(x = fitted_values01, y = residuals01)) +
  geom_point(color = "blue", alpha = 0.3) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red", linewidth =
0.5) +
  labs(title = "Residuals vs Fitted Values - Model 1",
       x = "Fitted Values",
       y = "Residuals") +
  theme_minimal() +
  theme(
          plot.title = element_text(size = 15, hjust = 0.5),
       )
```



The scatter plot shows a cluster of points. This indicates that the model is missing key variables to successfully explain the `Sales Price` variable. There are also a few outliers.

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

```
ggplot(df, aes(sample = residuals01)) +
  geom_qq(color = "blue", alpha = 0.3) +
  geom_qq_line(color = "red") +
  labs(title = "Q-Q Plot of Residuals - Model 1",
       x = "Theoretical Quantiles",
       y = "Sample Quantiles") +
  theme_minimal() +
  theme(
          plot.title = element_text(size = 15, hjust = 0.5),
       )
```



The Q-Q plot shows that the residuals of the first are not normally distributed, since the data points deviate from the diagonal line.

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.

Variables that may add value to the model

- **sq_ft_lot**: The biggest lot size, the more expensive a property could be.
- **building_grade**: A property could be more expensive if it has a better building grade.
- **square_feet_total_living**: The size of the actual construction affects the price.
- **bedrooms**: Typically, a property with more bedrooms is more expensive.
- **bath_full_count**: As with bedrooms, the more full bathrooms, the more expensive a property could be.
- **bath_half_count**: The number of half-bathrooms also affects the price of properties.
- **year_built**: In a typical property, the newest it is, the more expensive it could be.
- **year_renovated**: As with the previous variable, a property recently renovated can have a higher price.
- **price_sq_ft_lot**: The price per square foot of the property has a direct impact on the price.
- **price_sq_total_living**: The price per square foot of the constructed space in the property has a direct impact on the price.

```r
df$price_sq_ft_lot = df$`Sale Price` / df$sq_ft_lot
df$price_sq_total_living = df$`Sale Price` / df$square_feet_total_living

model02 <- lm(`Sale Price` ~ sq_ft_lot + building_grade + 
square_feet_total_living +
                bedrooms + bath_full_count + bath_half_count +
                year_built + year_renovated + price_sq_ft_lot + 
price_sq_total_living, data = df)
```

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

```
summary(model02)

##
## Call:
## lm(formula = `Sale Price` ~ sq_ft_lot + building_grade +
square_feet_total_living +
##     bedrooms + bath_full_count + bath_half_count + year_built +
##     year_renovated + price_sq_ft_lot + price_sq_total_living,
##     data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4900561   -41500      310    39600  2288585
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                2.453e+06  2.389e+05  10.267  < 2e-16 ***
## sq_ft_lot                  2.680e-01  2.886e-02   9.288  < 2e-16 ***
## building_grade             5.082e+04  2.116e+03  24.018  < 2e-16 ***
## square_feet_total_living   1.876e+02  2.817e+00  66.615  < 2e-16 ***
## bedrooms                   1.310e+04  2.184e+03   5.997 2.06e-09 ***
## bath_full_count            8.446e+03  2.862e+03   2.951  0.00317 **
## bath_half_count            7.904e+03  2.963e+03   2.668  0.00764 **
## year_built                -1.609e+03  1.214e+02 -13.260  < 2e-16 ***
## year_renovated            -1.269e+00  6.738e+00  -0.188  0.85067
## price_sq_ft_lot            1.093e+03  2.723e+01  40.154  < 2e-16 ***
## price_sq_total_living      1.293e+03  1.217e+01 106.248  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 166200 on 12854 degrees of freedom
## Multiple R-squared:  0.8311, Adjusted R-squared:  0.831
## F-statistic:  6326 on 10 and 12854 DF,  p-value: < 2.2e-16
```

Interpretation of summary results for model 2

For this model, the value of `R-squared` is 0.8311. This indicates that approximately 83.11% of the variance in Sale Price can be explained by the 10 predictors used in the model

`Adjusted R-squared`: 0.831. Since this model has a different numbers of predictors than the first one (10 vs 1), the adjusted R-squared value makes a more reliable metric when comparing both models. This second model is better at explaining the variability of Sale Price (83.10 % vs 1.43%)

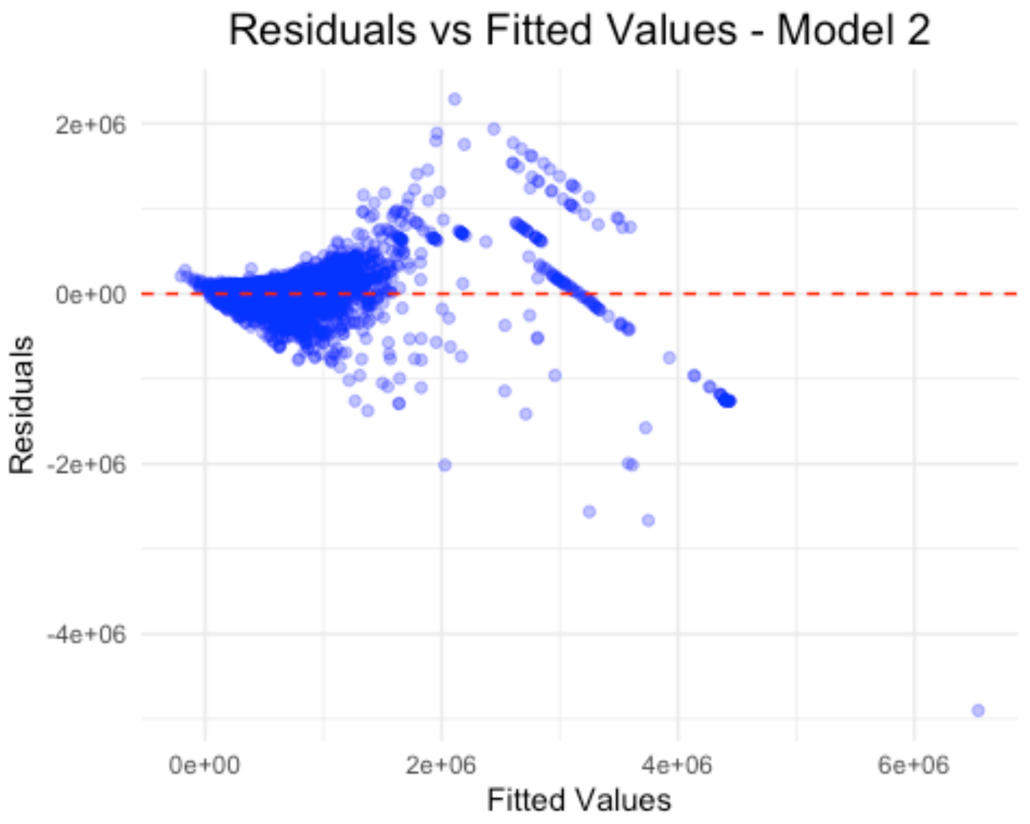
The `F-statistic` of 6326 is the ratio of variance explained by the model to the variance # not explained by the model.

The `Degrees of Freedom` (DF, 10 and 12854) indicate the number of predictors (10) and the residual degrees of freedom. This is the total of observations minus the number of predictors minus 1. In other words: `12865 - 10 - 1 = 12854`

Finally, the `p-value` of < 2.2e-16 is extremely small. This means that the probability of observing an F-statistic of 6326 under the null hypothesis is extremely low.

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?

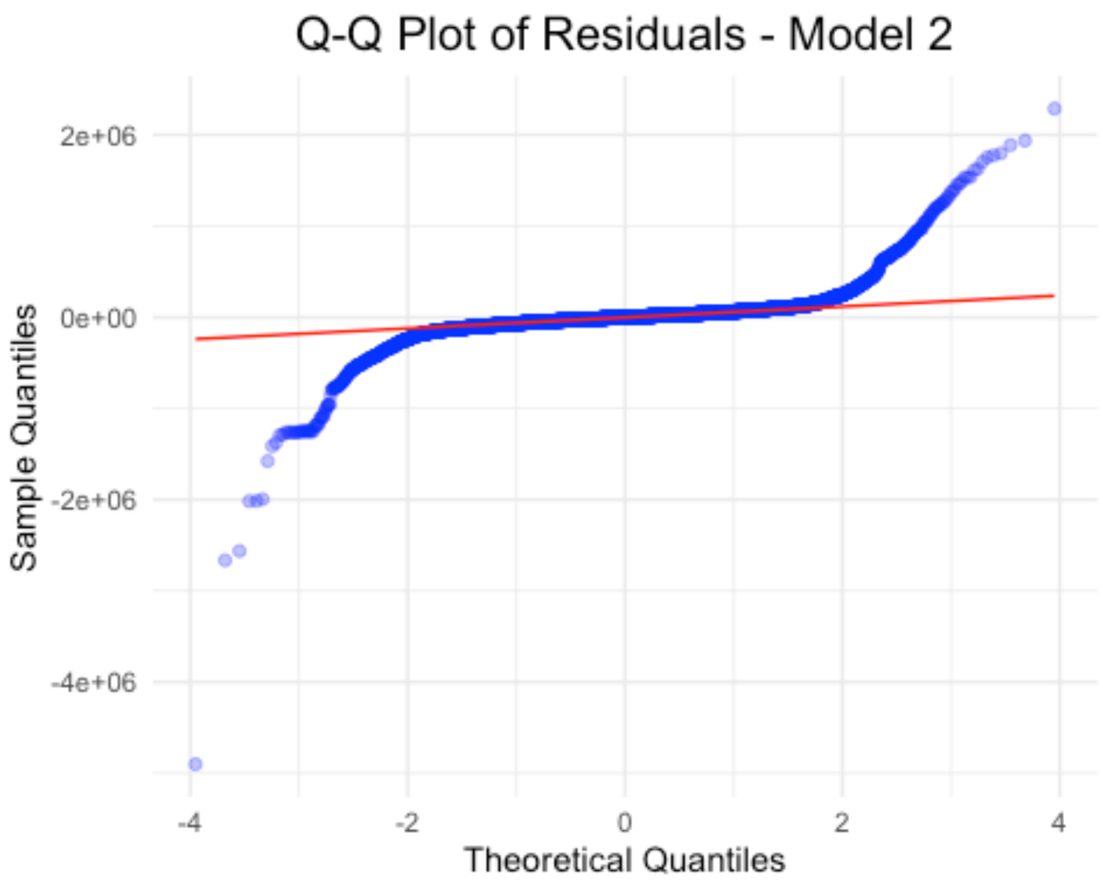
```
residuals02 <- residuals(model02)
fitted_values02 <- fitted(model02)

ggplot(df, aes(x = fitted_values02, y = residuals02)) +
  geom_point(color = "blue", alpha = 0.3) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red", linewidth =
0.5) +
  labs(title = "Residuals vs Fitted Values - Model 2",
       x = "Fitted Values",
       y = "Residuals") +
  theme_minimal() +
  theme(
          plot.title = element_text(size = 15, hjust = 0.5),
       )
```



The scatter plot shows that all the data points are slightly better aligned, when comparing them to the first model. However there are still cluster of points and outliers. The second model is still missing key variables to successfully explain the `Sales Price` variable.

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

```r
ggplot(df, aes(sample = residuals02)) +
  geom_qq(color = "blue", alpha = 0.3) +
  geom_qq_line(color = "red") +
  labs(title = "Q-Q Plot of Residuals - Model 2",
       x = "Theoretical Quantiles",
       y = "Sample Quantiles") +
  theme_minimal() +
  theme(
        plot.title = element_text(size = 15, hjust = 0.5),
       )
```



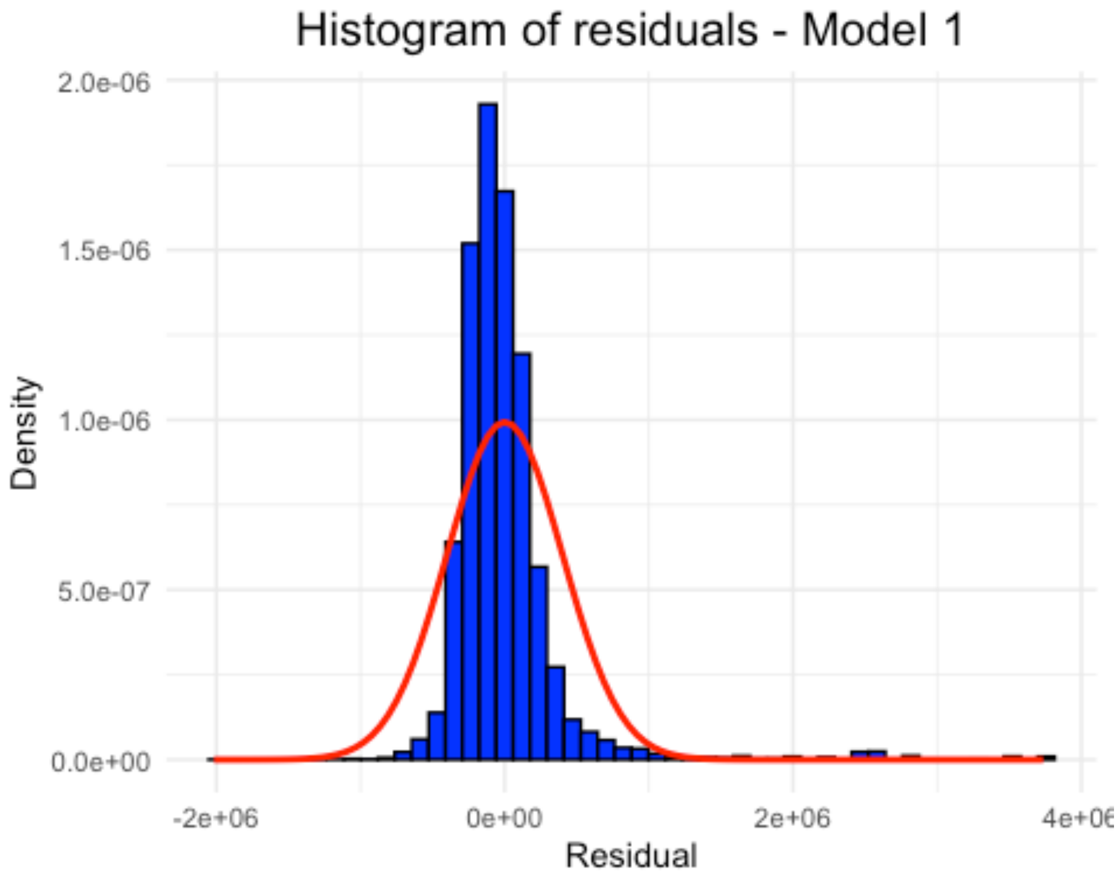
The Q-Q plot shows that the residuals of the second model are still not normally distributed, since the data points deviate from the diagonal line.

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?

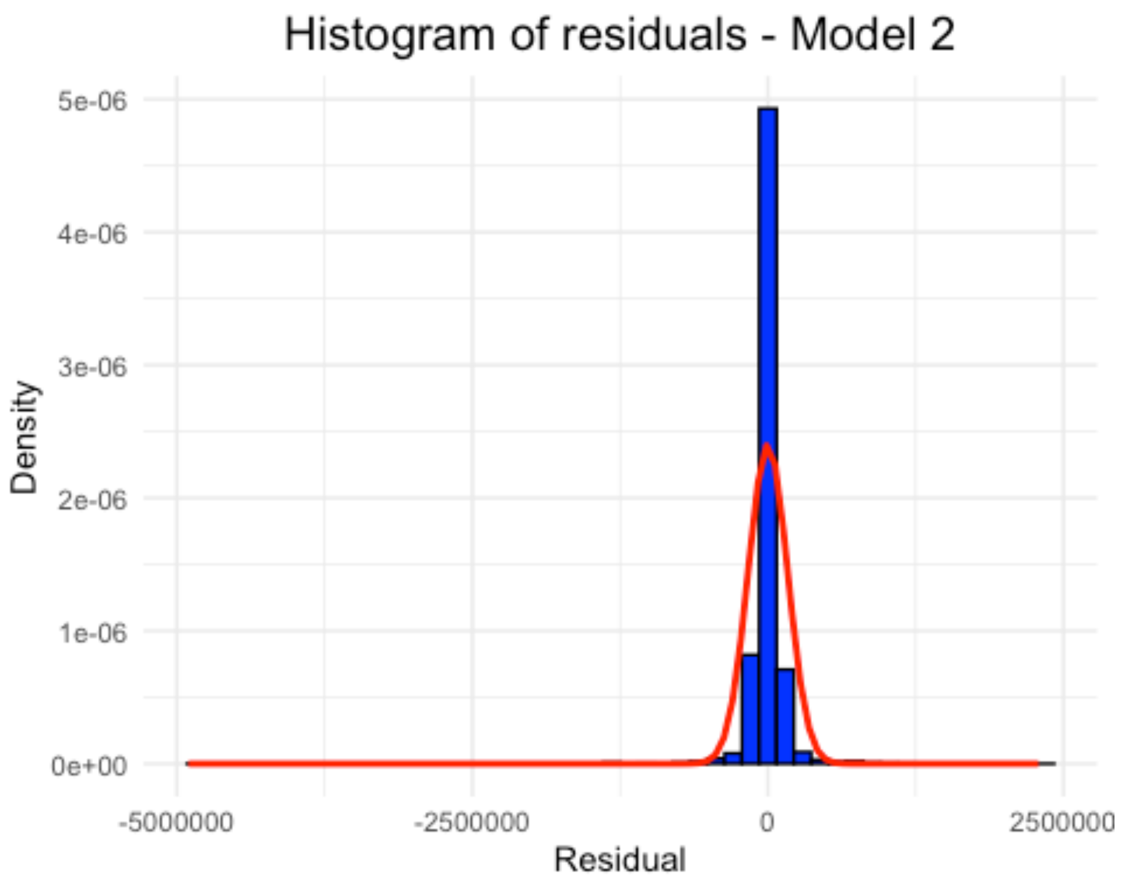
```
anova(model01, model02)

## Analysis of Variance Table
##
## Model 1: `Sale Price` ~ sq_ft_lot
## Model 2: `Sale Price` ~ sq_ft_lot + building_grade +
square_feet_total_living +
##     bedrooms + bath_full_count + bath_half_count + year_built +
##     year_renovated + price_sq_ft_lot + price_sq_total_living
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
## 1  12863 2.0734e+15
## 2  12854 3.5524e+14  9 1.7181e+15 6907.6 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
# Histogram for Model 1
residuals_model1 <- residuals(model01)

ggplot(df, aes(x = residuals_model1)) +
  geom_histogram(aes(y = after_stat(density)), bins = 50, fill = "blue",
color = "black") +
  stat_function(fun = dnorm, args = list(mean = mean(residuals_model1), sd =
sd(residuals_model1)),
                color = "red", linewidth = 1) +
  labs(title = "Histogram of residuals - Model 1",
       x = "Residual",
       y = "Density") +
  theme_minimal() +
    theme(
          plot.title = element_text(size = 15, hjust = 0.5),
       )
```



```
# Histogram for Model 2
residuals_model2 <- residuals(model02)

ggplot(df, aes(x = residuals_model2)) +
  geom_histogram(aes(y = after_stat(density)), bins = 50, fill = "blue",
color = "black") +
  stat_function(fun = dnorm, args = list(mean = mean(residuals_model2), sd =
sd(residuals_model2)),
                color = "red", linewidth = 1) +
  labs(title = "Histogram of residuals - Model 2",
       x = "Residual",
       y = "Density") +
  theme_minimal() +
    theme(
          plot.title = element_text(size = 15, hjust = 0.5),
       )
```



Histograms for residuals of both models are not normally distributed. This is an indication of a biased model.

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

12.1. Install the 'Metrics' package in R Studio

12.2. Using the first model, we will make predictions on the dataset using the predict function.

```
preds01 <- predict(model01)
```

12.3. What is the RMSE for the first model?

```
rmse(df$`Sale Price`, preds01)

## [1] 401452.5
```

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

```
preds02 <- predict(model02)
rmse(df$`Sale Price`, preds02)

## [1] 166171.7
```

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

Yes, the second model (RMSE: 166,171.70) is significantly better than the first model (RMSE: 401,452.50).

The difference is 235,280.80.