

assignment_exercise8.2_BezawadaSashidhar

Sashidhar Bezawada

10/29/2022

R Markdown

```
#Exploring the housing dataset  
str(housing_df)
```

```
## # tibble [12,865 x 24] (S3: tbl_df/tbl/data.frame)  
## $ Sale Date : POSIXct[1:12865], format: "2006-01-03" "2006-01-03" ...  
## $ Sale Price : num [1:12865] 698000 649990 572500 420000 369900 ...  
## $ sale_reason : num [1:12865] 1 1 1 1 1 1 1 1 1 1 ...  
## $ sale_instrument : num [1:12865] 3 3 3 3 3 15 3 3 3 3 ...  
## $ sale_warning : chr [1:12865] NA NA NA NA ...  
## $ sitetype : chr [1:12865] "R1" "R1" "R1" "R1" ...  
## $ addr_full : chr [1:12865] "17021 NE 113TH CT" "11927 178TH PL NE" "13315 174TH AVE N" ...  
## $ zip5 : num [1:12865] 98052 98052 98052 98052 98052 ...  
## $ ctynname : chr [1:12865] "REDMOND" "REDMOND" NA "REDMOND" ...  
## $ postalctyn : chr [1:12865] "REDMOND" "REDMOND" "REDMOND" "REDMOND" ...  
## $ lon : num [1:12865] -122 -122 -122 -122 -122 ...  
## $ lat : num [1:12865] 47.7 47.7 47.7 47.6 47.7 ...  
## $ building_grade : num [1:12865] 9 9 8 8 7 7 10 10 9 8 ...  
## $ square_feet_total_living: num [1:12865] 2810 2880 2770 1620 1440 4160 3960 3720 4160 2760 ...  
## $ bedrooms : num [1:12865] 4 4 4 3 3 4 5 4 4 4 ...  
## $ bath_full_count : num [1:12865] 2 2 1 1 1 2 3 2 2 1 ...  
## $ bath_half_count : num [1:12865] 1 0 1 0 0 1 0 1 1 0 ...  
## $ bath_3qtr_count : num [1:12865] 0 1 1 1 1 1 0 1 1 1 ...  
## $ year_built : num [1:12865] 2003 2006 1987 1968 1980 ...  
## $ year_renovated : num [1:12865] 0 0 0 0 0 0 0 0 0 0 ...  
## $ current_zoning : chr [1:12865] "R4" "R4" "R6" "R4" ...  
## $ sq_ft_lot : num [1:12865] 6635 5570 8444 9600 7526 ...  
## $ prop_type : chr [1:12865] "R" "R" "R" "R" ...  
## $ present_use : num [1:12865] 2 2 2 2 2 2 2 2 2 2 ...
```

```
summary(housing_df)
```

```
##      Sale Date           Sale Price       sale_reason  
## Min.   :2006-01-03 00:00:00.00  Min.   : 698   Min.   : 0.00  
## 1st Qu.:2008-07-07 00:00:00.00  1st Qu.: 460000  1st Qu.: 1.00  
## Median :2011-11-17 00:00:00.00  Median : 593000  Median : 1.00  
## Mean    :2011-07-28 15:07:32.48  Mean    : 660738  Mean    : 1.55  
## 3rd Qu.:2014-06-05 00:00:00.00  3rd Qu.: 750000  3rd Qu.: 1.00
```

```

##  Max.   :2016-12-16 00:00:00.00  Max.   :4400000  Max.   :19.00
##  sale_instrument  sale_warning      sitetype      addr_full
##  Min.   : 0.000  Length:12865      Length:12865      Length:12865
##  1st Qu.: 3.000  Class :character  Class :character  Class :character
##  Median : 3.000  Mode   :character  Mode   :character  Mode   :character
##  Mean   : 3.678
##  3rd Qu.: 3.000
##  Max.   :27.000
##    zip5      ctynname      postalctyn      lon
##  Min.   :98052  Length:12865      Length:12865      Min.   :-122.2
##  1st Qu.:98052  Class :character  Class :character  1st Qu.:-122.1
##  Median :98052  Mode   :character  Mode   :character  Median :-122.1
##  Mean   :98053
##  3rd Qu.:98053
##  Max.   :98074
##    lat      building_grade  square_feet_total_living  bedrooms
##  Min.   :47.46  Min.   : 2.00  Min.   : 240          Min.   : 0.000
##  1st Qu.:47.67  1st Qu.: 8.00  1st Qu.: 1820        1st Qu.: 3.000
##  Median :47.69  Median : 8.00  Median : 2420        Median : 4.000
##  Mean   :47.68  Mean   : 8.24  Mean   : 2540        Mean   : 3.479
##  3rd Qu.:47.70  3rd Qu.: 9.00  3rd Qu.: 3110        3rd Qu.: 4.000
##  Max.   :47.73  Max.   :13.00  Max.   :13540       Max.   :11.000
##  bath_full_count  bath_half_count  bath_3qtr_count  year_built
##  Min.   : 0.000  Min.   :0.0000  Min.   :0.000  Min.   :1900
##  1st Qu.: 1.000  1st Qu.:0.0000  1st Qu.:0.000  1st Qu.:1979
##  Median : 2.000  Median :1.0000  Median :0.000  Median :1998
##  Mean   : 1.798  Mean   :0.6134  Mean   :0.494  Mean   :1993
##  3rd Qu.: 2.000  3rd Qu.:1.0000  3rd Qu.:1.000  3rd Qu.:2007
##  Max.   :23.000  Max.   :8.0000  Max.   :8.000  Max.   :2016
##  year_renovated  current_zoning      sq_ft_lot      prop_type
##  Min.   : 0.00  Length:12865      Min.   : 785  Length:12865
##  1st Qu.: 0.00  Class :character  1st Qu.: 5355  Class :character
##  Median : 0.00  Mode   :character  Median : 7965  Mode   :character
##  Mean   : 26.24
##  3rd Qu.: 0.00
##  Max.   :2016.00
##  present_use
##  Min.   : 0.000
##  1st Qu.: 2.000
##  Median : 2.000
##  Mean   : 6.598
##  3rd Qu.: 2.000
##  Max.   :300.000

head(housing_df)

```

```

## # A tibble: 6 x 24
##   `Sale Date`      `Sale Price`  sale_~1 sale_~2 sale_~3 sitet~4 addr_~5  zip5
##   <dttm>           <dbl>     <dbl>     <dbl> <chr>    <chr>    <dbl>
## 1 2006-01-03 00:00:00 698000      1        3 <NA>     R1      17021 ~ 98052
## 2 2006-01-03 00:00:00 649990      1        3 <NA>     R1      11927 ~ 98052
## 3 2006-01-03 00:00:00 572500      1        3 <NA>     R1      13315 ~ 98052
## 4 2006-01-03 00:00:00 420000      1        3 <NA>     R1      3303 1~ 98052
## 5 2006-01-03 00:00:00 369900      1        3 15      R1      16126 ~ 98052

```

```

## 6 2006-01-03 00:00:00      184667      1      15 18 51   R1      8101 2~ 98053
## # ... with 16 more variables: ctynname <chr>, postalctyn <chr>, lon <dbl>,
## #   lat <dbl>, building_grade <dbl>, square_feet_total_living <dbl>,
## #   bedrooms <dbl>, bath_full_count <dbl>, bath_half_count <dbl>,
## #   bath_3qtr_count <dbl>, year_built <dbl>, year_renovated <dbl>,
## #   current_zoning <chr>, sq_ft_lot <dbl>, prop_type <chr>, present_use <dbl>,
## #   and abbreviated variable names 1: sale_reason, 2: sale_instrument,
## #   3: sale_warning, 4: sitetype, 5: addr_full

## Converting numerical categorical columns to factors
housing_df$sale_reason <- factor(housing_df$sale_reason)

housing_df$sale_instrument <- factor(housing_df$sale_instrument)

housing_df$building_grade <- factor(housing_df$building_grade)

housing_df$zip5 <- factor(housing_df$zip5)

#Handling any NA values, mostly in ctynname
#sale_warning
#ctynname
sum(is.na(housing_df$sale_warning))

## [1] 10568

sum(is.na(housing_df$ctynname))

## [1] 6078

#Take out NA citynames
housing_df$ctynname[is.na(housing_df$ctynname)] <- 'Not Stated'

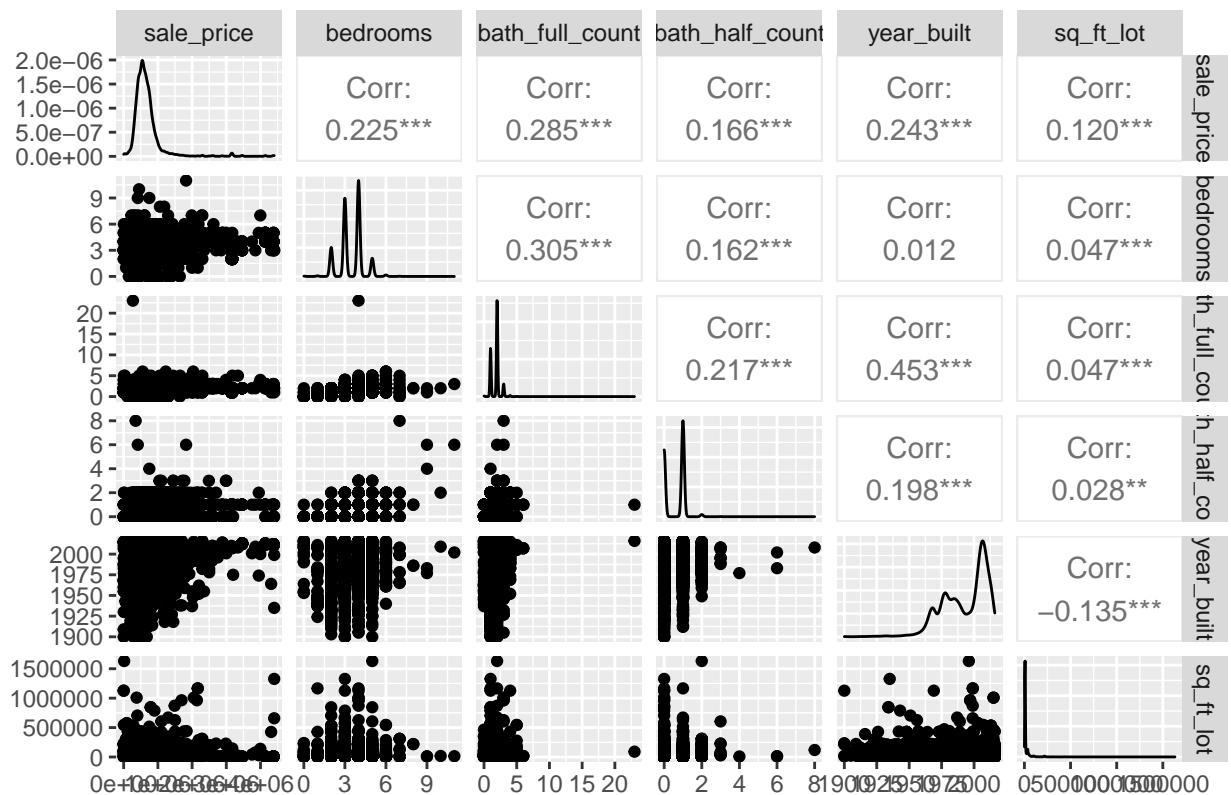
#Handling NA sale_warning, 0 if does not exist
housing_df$sale_warning[is.na(housing_df$sale_warning)] <- 0

```

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

- From first using this dataset, I found that there were categorical variables which were labeled as being of type ‘numeric’. Since I wanted the categories to be recognized, I converted these variables into factors to allow that sort of manipulation and typing. These variables are: sale_reason, sale_instrument and building_grade.
- Also, there were NA values in two of the columns, sale_warning and ctynname. In order to remove these, I set default values in both of the fields.
- I also renamed the ‘Sale Date’ & ‘Sale Price’ columns, so they wouldn’t require the use of quotes when referencing.

correlogram with ggpairs()



Highest positive correlation with sales_price \rightarrow bath_full_count with 0.285

Lowest correlation \rightarrow sq_ft_lot with 0.120

2. Create two variables; one that will contain the variables Sale Price and Square Foot of Lot (same variables used from previous assignment on simple regression) and one that will contain Sale Price and several additional predictors of your choice. Explain the basis for your additional predictor selections.

The additional predictors I selected (apart from sq_ft_lot) are: bedrooms, bath_full_count, bath_half_count and year_built. Using personal experience, location, age of the house and number of rooms is critical for determining the sales price of a house. These are factors that most prospective renters/buyers ask about when looking at places, and definitely play a role at determining the value of the property. This value is usually conveyed through sales price.

3. Execute a summary() function on two variables defined in the previous step to compare the model results. What are the R² and Adjusted R² statistics? Explain what these results tell you about the overall model. Did the inclusion of the additional predictors help explain any large variations found in Sale Price?

```
housing.1 <- lm(var1, data=housing_df)
housing.2 <- lm(var2, data=housing_df)

summary(housing.1)
```

```

## 
## Call:
## lm(formula = var1, data = housing_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

```

```
summary(housing.2)
```

```

## 
## Call:
## lm(formula = var2, data = housing_df)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2342084 -148542 - 48527  63149 3657521
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -8.499e+06 4.402e+05 -19.308 <2e-16 ***
## sq_ft_lot    9.163e-01 5.889e-02 15.558 <2e-16 ***
## bedrooms     7.682e+04 4.022e+03 19.100 <2e-16 ***
## bath_full_count 7.998e+04 6.091e+03 13.130 <2e-16 ***
## bath_half_count 5.404e+04 6.510e+03  8.301 <2e-16 ***
## year_built    4.363e+03 2.223e+02 19.623 <2e-16 ***
## ---      
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 373700 on 12859 degrees of freedom
## Multiple R-squared:  0.1464, Adjusted R-squared:  0.1461
## F-statistic: 441.1 on 5 and 12859 DF, p-value: < 2.2e-16

```

- Model 1:
 - R2: 0.01435, Square feet of the lot accounts for 1.435% of the variation in sales price.
 - Adjusted R-squared: 0.01428, Difference from R2 is 0.00007 (very small), if the model were derived from the population rather than a sample, it would account for approx. 0.007% less variance in the outcome.
- Model 2:
 - R2: 0.1464, If Square feet of the lot accounts for 1.435%, the other predictors account for an additional 10% of the variation in sales price, which is much more than just square feet by itself.

- Adjusted R-squared: 0.1461, Difference from R² is 0.0003, if the model were derived from the population rather than the sample, it would account for approx. 0.03% less variance in the outcome.

The inclusion of the additional predictors helped explain an additional 10% of the variation in sales price, which is quite a jump from the 1.435% that was only explained by square feet of the lot.

4. Considering the parameters of the multiple regression model you have created. What are the standardized betas for each parameter and what do the values indicate?

```
##          sq_ft_lot      bedrooms bath_full_count bath_half_count      year_built
##        0.1290002     0.1664462      0.1287159      0.0703343     0.1857869
## [1] "Standard deviation of sales price"
## [1] 404381.1
```

These estimates tell us the number of standard deviations by which the outcome will change as a result of one standard deviation change in the predictor.

Examples:

- sq_ft_lot (.129): this value indicates that as the square feet of the lot increases by one standard deviation, sales prices increases by .129 standard deviations. The standard deviation for sales price is 404,381, and so this constitutes a change of 52,165 (404381*.129) dollars.
- bedrooms (.166): this value indicates that as the number of bedrooms in the house increases by one standard deviation, sales price increases by .166 standard deviations. This constitutes a change of \$67,127 dollars.

5. Calculate the confidence intervals for the parameters in your model and explain what the results indicate.

```
##                  2.5 %      97.5 %
## (Intercept) -9.361834e+06 -7.636159e+06
## sq_ft_lot    8.008167e-01  1.031688e+00
## bedrooms     6.893989e+04  8.470829e+04
## bath_full_count 6.803975e+04  9.191888e+04
## bath_half_count 4.128221e+04  6.680497e+04
## year_built    3.927031e+03  4.798632e+03
```

For this model, there are no predictors which have zero in their confidence intervals, which is an indicator that the model is not extremely poor in its predictions.

sq_ft_lot and year_built seem to have the highest confidence intervals, indicating that the estimates for the current model are likely to be representative of the true population values.

The interval for the other variables are a bit wider, indicating that the parameter is less representative.

6. Assess the improvement of the new model compared to your original model (simple regression model) by testing whether this change is significant by performing an analysis of variance.

```
## Analysis of Variance Table
##
## Model 1: sale_price ~ sq_ft_lot
## Model 2: sale_price ~ sq_ft_lot + bedrooms + bath_full_count + bath_half_count +
##           year_built
##   Res.Df      RSS Df  Sum of Sq      F    Pr(>F)
## 1 12863 2.0734e+15
## 2 12859 1.7956e+15  4 2.7776e+14 497.28 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Value of F is 497.28. The p-value for F is very small, 2.2e-16, which is definitely statistically significant given its magnitude.

From this assessment, the new model significantly improved the fit of the model to the data compared to the simple regression model with just sq_ft_lot.

7. Perform casewise diagnostics to identify outliers and/or influential cases, storing each function's output in a dataframe assigned to a unique variable name.

```
housing_df$standardized.residuals <- rstandard(housing.2)
housing_df$studentized.residuals <- rstudent(housing.2)
housing_df$cooks.distance <- cooks.distance(housing.2)
housing_df$dfbeta <- dfbeta(housing.2)
housing_df$dffit <- dffits(housing.2)
housing_df$leverage <- hatvalues(housing.2)
housing_df$covariance.ratios <- covratio(housing.2)

housing_df %>% dplyr::select(standardized.residuals, studentized.residuals, cooks.distance, dfbeta, dffit, lever, covariance.ratios)
```

	standardized.residuals	studentized.residuals	cooks.distance	dfbeta	dffit	leverage	covariance.ratios
1	-0.185	-0.185	9.20e-7	332.	-2.35e-3	1.61e-4	1.00
2	-0.201	-0.201	2.14e-6	725.	-3.58e-3	3.16e-4	1.00
3	-0.124	-0.124	9.04e-7	107.	-2.33e-3	3.50e-4	1.00
4	0.0367	0.0367	7.54e-8	144.	6.72e-4	3.36e-4	1.00
5	-0.232	-0.232	2.40e-6	-188.	-3.79e-3	2.66e-4	1.00
6	-1.58	-1.58	7.18e-5	3683.	-2.08e-2	1.72e-4	0.999
7	0.376	0.376	1.94e-5	392.	1.08e-2	8.21e-4	1.00
8	0.405	0.405	4.29e-6	744.	5.08e-3	1.57e-4	1.00
9	-0.0832	-0.0832	2.82e-7	-359.	-1.30e-3	2.44e-4	1.00
10	0.144	0.144	1.71e-6	-11.4	3.20e-3	4.93e-4	1.00

... with 12,855 more rows, 1 more variable: dfbeta[2:6] <dbl>, and
abbreviated variable names 1: studentized.residuals, 2: cooks.distance,
3: dfbeta[, "(Intercept)"], 4: leverage, 5: covariance.ratios

```

head(housing_df)

## # A tibble: 6 x 31
##   sale_date      sale_price sale_re~1 sale_~2 sale_~3 sitet~4 addr_~5 zip5
##   <dttm>          <dbl> <fct>    <fct>  <chr>  <chr>  <fct>
## 1 2006-01-03 00:00:00 698000 1        3       0       R1     17021 ~ 98052
## 2 2006-01-03 00:00:00 649990 1        3       0       R1     11927 ~ 98052
## 3 2006-01-03 00:00:00 572500 1        3       0       R1     13315 ~ 98052
## 4 2006-01-03 00:00:00 420000 1        3       0       R1     3303 1~ 98052
## 5 2006-01-03 00:00:00 369900 1        3       15      R1     16126 ~ 98052
## 6 2006-01-03 00:00:00 184667 1        15      18      51     R1     8101 2~ 98053
## # ... with 23 more variables: ctynname <chr>, postalctyn <chr>, lon <dbl>,
## #   lat <dbl>, building_grade <fct>, square_feet_total_living <dbl>,
## #   bedrooms <dbl>, bath_full_count <dbl>, bath_half_count <dbl>,
## #   bath_3qtr_count <dbl>, year_built <dbl>, year_renovated <dbl>,
## #   current_zoning <chr>, sq_ft_lot <dbl>, prop_type <chr>, present_use <dbl>,
## #   standardized.residuals <dbl>, studentized.residuals <dbl>,
## #   cooks.distance <dbl>, dfbeta <dbl[,6]>, dffit <dbl>, leverage <dbl>, ...
## #   ... with 335 more rows

housing_df$large.residual <- housing_df$standardized.residuals > 2 | housing_df$standardized.residuals < -2
sum(housing_df$large.residual)

## [1] 345

housing_df[housing_df$large.residual,c("sale_date","sale_price","sq_ft_lot","bedrooms","standardized.residuals")]

## # A tibble: 345 x 5
##   sale_date      sale_price sq_ft_lot bedrooms standardized.residuals
##   <dttm>          <dbl>    <dbl>     <dbl>           <dbl>
## 1 2006-01-04 00:00:00 165000    278891     3            -2.02
## 2 2006-01-11 00:00:00 265000    112650     4            -2.08
## 3 2006-02-01 00:00:00 1900000   37017      4             2.90
## 4 2006-02-13 00:00:00 1520000   19173      5             2.52
## 5 2006-02-15 00:00:00 1390000   225640     0             2.88
## 6 2006-03-20 00:00:00 1588359    8752      2             2.58
## 7 2006-03-21 00:00:00 1450000   14043      3             2.38
## 8 2006-03-21 00:00:00 1450000   14043      2             3.57
## 9 2006-03-28 00:00:00 270000    89734      4            -6.56
## 10 2006-03-29 00:00:00 200000    288367     5            -2.26
## # ... with 335 more rows

```

345 cases had a large residual. We would expect about 643 cases (5% of 12865) to have standardized residuals outside of the limits from -2 to 2. Therefore, our sample was underneath what we expected to be outside of the limits!

However, there are quite a few residuals with values further away from 2 and -2. For example, there is one case with a standarized residual of ~8, which could be a case that we need to do more analysis on.

Case 341 has a standardized residual of 8.21 which is concerningly above the upper bound of 2.

```

## # A tibble: 345 x 3
##   cooks.distance leverage covariance.ratios

```

```

##          <dbl>      <dbl>      <dbl>
## 1      0.00145  0.00212    1.00
## 2      0.000851 0.00118    1.00
## 3      0.000647 0.000463   0.997
## 4      0.00102  0.000962   0.998
## 5      0.00369  0.00267    0.999
## 6      0.000471 0.000424   0.998
## 7      0.000398 0.000422   0.998
## 8      0.00423  0.00198    0.996
## 9      0.915     0.113     1.11
## 10     0.00246  0.00288    1.00
## # ... with 335 more rows

```

- No Cook's distance values greater than 1, no undue influence on the model
- Average leverage = $5 + 1/12865 = 0.0005$
 - Looking for values either twice as large as this (0.001) or three times as large (0.0015)
 - Case 9 has a leverage quite larger than both limits
 - Case 17 is higher than the limit
 - There are quite a few other cases with leverage amounts greater than the average limits, may need to investigate further
- $.9986 > CVR > 1.0014$
- Most of the values are right around these limits, but there are some which fall below and above.
 - Case 9 has a covariance ratio greater than 1.0014 by $\sim .1$
 - Case 17 is just slightly above the upper bound for covariance ratio, $\sim .007$
 - Case 341 has a covariance ratio below the specified lower bound by $\sim .03$, but its cook distance and leverage values are OK.

Perform the necessary calculations to assess the assumption of independence and state if the condition is met or not.

```

##   lag Autocorrelation D-W Statistic p-value
##   1      0.6474656     0.7050626      0
## Alternative hypothesis: rho != 0

```

Testing the assumption of independent errors using the Durbin-Watson test

Our D-W Statistic is 0.705, which is less than 1 and could definitely raise some alarm bells according to the book author. In this case, we could state that the assumption of independence has not been met. Also, with the p-value equal to 0 which is less than 0.05, we could say that this contradiction of the assumption is statistically significant.

Perform the necessary calculations to assess the assumption of no multicollinearity and state if the condition is met or not. VIF & tolerance statistics

```

## [1] "VIF"

##          sq_ft_lot      bedrooms bath_full_count bath_half_count      year_built
## 1      1.035628       1.144039        1.447635         1.081471       1.350335

## [1] "TOLERANCE"

```

```
##          sq_ft_lot      bedrooms bath_full_count bath_half_count      year_built
## 0.9655978        0.8740957        0.6907819        0.9246666        0.7405570

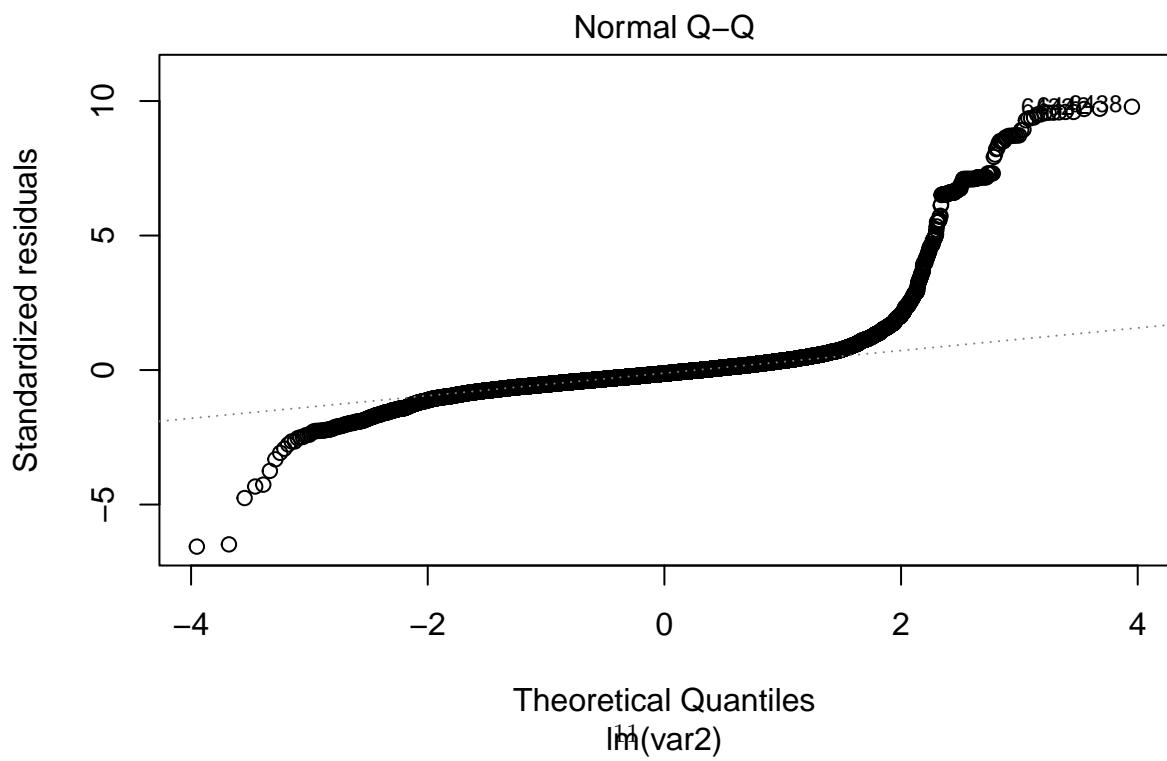
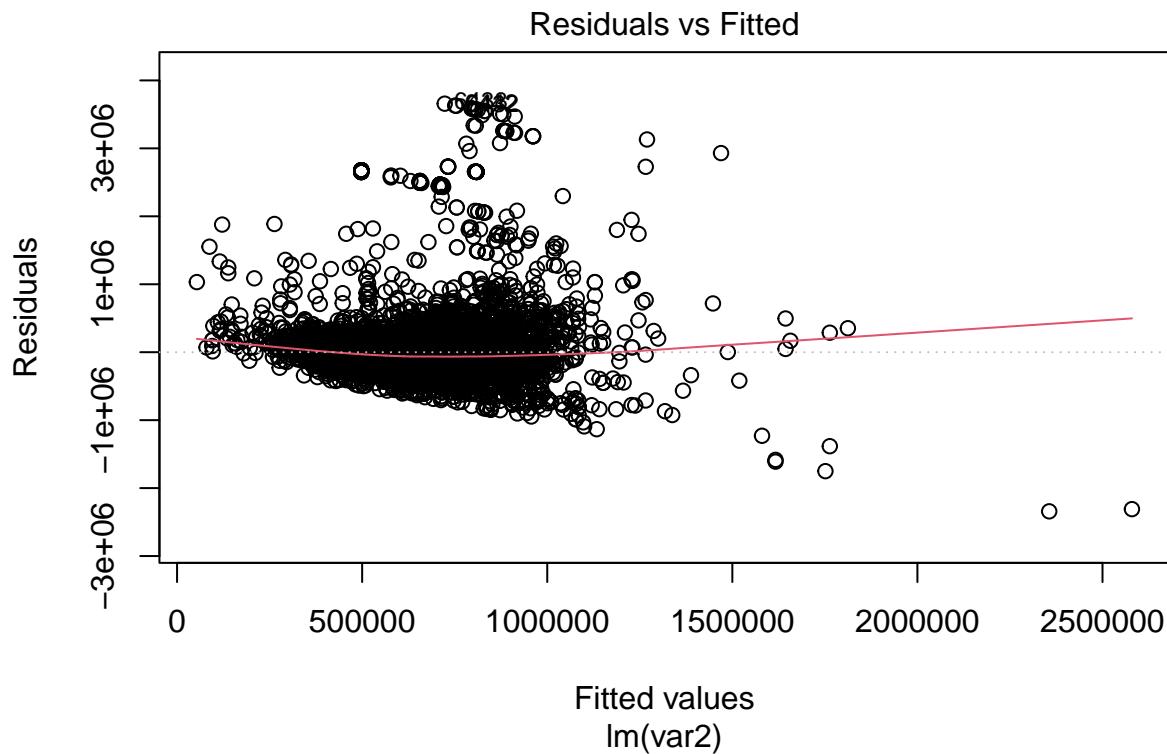
## [1] "AVERAGE VIF"

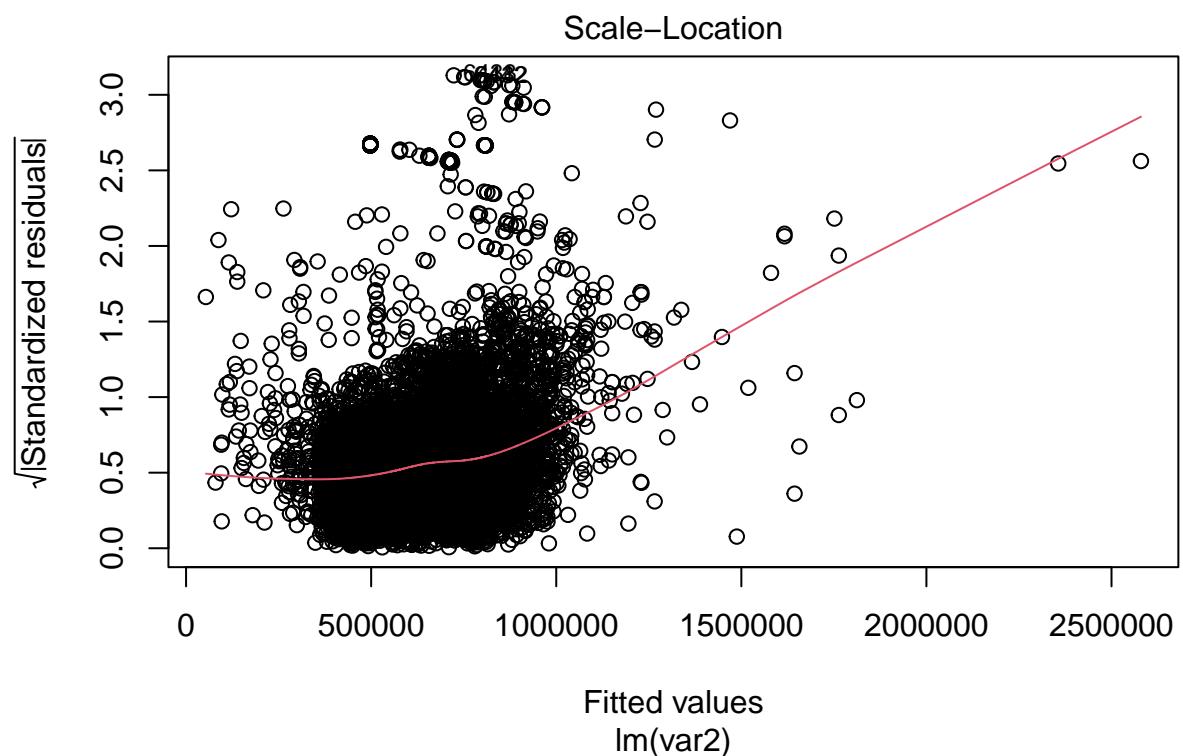
## [1] 1.211822
```

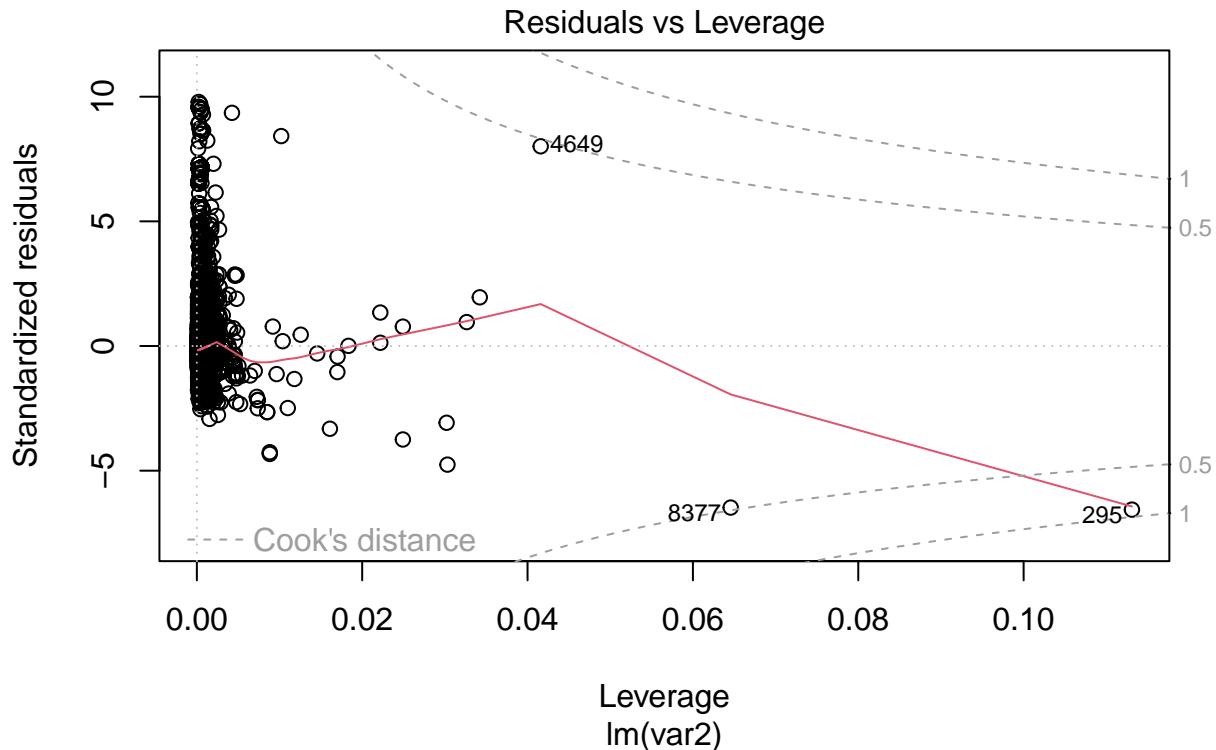
- The largest VIF is not greater than 10, so we are good there!
- The average VIF is just slightly greater than 1 by ~.2, so I wouldn't say it is substantial enough to bias the regression.
- All tolerances are greater than 0.1 and 0.2

Therefore with these measures, I would conclude that there is no collinearity within this housing data.

Visually check the assumptions related to the residuals using the `plot()` and `hist()` functions. Summarize what each graph is informing you of and if any anomalies are present.



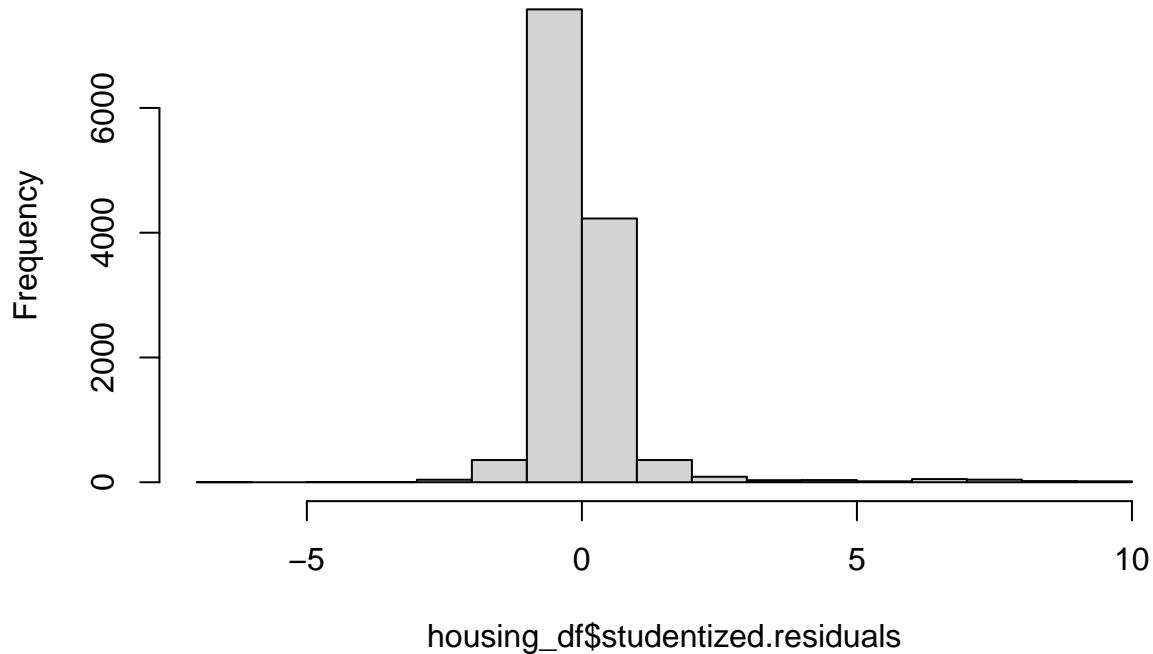




From looking at the plot of fitted values against residuals, I can see some clear funneling out on the graph. This increases the chances that there is heteroscedasticity in the data.

For the Q-Q Plot, the points quite obviously deviate from the line, expressing non-normality for the distribution of the data. There is skew.

Histogram of housing_df\$studentized.residuals



The histogram of the studentized residuals shows a left skew to the data, more points at the lower end of the scale. There's also some outliers in the plot, which should probably be investigated.

Overall, is this regression model unbiased? If an unbiased regression model, what does this tell us about the sample vs. the entire population model?

From all of these assessments, I would conclude that the model does not appear to be both accurate for the sample and the entire population. The regression model does not appear to be unbiased unfortunately. We may need to assess our predictors further.