Exercise 12

Abhigyan Misra

October 12th 2020

Housing Data

Problem Statement : Work individually on this assignment. You are encouraged to collaborate on ideas and strategies pertinent to this assignment. Data for this assignment is focused on real estate transactions recorded from 1964 to 2016 and can be found in Week 6 Housing.xlsx. Using your skills in statistical correlation, multiple regression and R programming, you are interested in the following variables: Sale Price and several other possible predictors.

Using your ‘clean’ data set from the previous week complete the following:

```
## Set the working directory to the root of your DSC 520 directory
setwd("C:/git-bellevue/dsc520-fork")

## Load the ‘readxl‘ library
library(readxl)

## Load the ‘completed/Exercise 12/week-6-housing.xlsx‘ to
housing_df <- read_excel(path = ’completed/Exercise_12/week-6-housing.xlsx’ , skip = 0, sheet = ’Sheet2
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
##  $ zip5                    : num [1:12865] 98052 98052 98052 98052 98052 ...
##  $ ctyname                 : 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 1 0 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 ...
```

a. Explain why you chose to remove data points from your 'clean' dataset.

Removing all datasets which have Sales Warnings as they may tell if a sale was not correct or the price mentioned may be wrong. The warnings might be legitimate and not reflect the correct values. We may need more understanding on the Sales Warning codes, if we want to use those datasets.

```r
summary(housing_df)
```

```
##    Sale Date                     Sale Price       sale_reason
##  Min.   :2006-01-03 00:00:00   Min.   :    698   Min.   : 0.00
##  1st Qu.:2008-07-07 00:00:00   1st Qu.: 460000   1st Qu.: 1.00
##  Median :2011-11-17 00:00:00   Median : 593000   Median : 1.00
##  Mean   :2011-07-28 15:07:32   Mean   : 660738   Mean   : 1.55
##  3rd Qu.:2014-06-05 00:00:00   3rd Qu.: 750000   3rd Qu.: 1.00
##  Max.   :2016-12-16 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         ctyname           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                                         Mean   :-122.1
##  3rd Qu.:98053                                         3rd Qu.:-122.0
##  Max.   :98074                                         Max.   :-121.9
##       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                      Mean   :  22229
##  3rd Qu.:   0.00                      3rd Qu.:  12632
##  Max.   :2016.00                      Max.   :1631322
##   present_use
##  Min.   :  0.000
##  1st Qu.:  2.000
##  Median :  2.000
##  Mean   :  6.598
##  3rd Qu.:  2.000
##  Max.   :300.000
```

```r
cleaned_housing_df <- housing_df[(is.na(housing_df$sale_warning)),]
```

```r
summary(cleaned_housing_df)
```

```
##    Sale Date                      Sale Price       sale_reason
##  Min.   :2006-01-03 00:00:00   Min.   :   2500   Min.   : 0.000
##  1st Qu.:2008-05-27 00:00:00   1st Qu.: 485075   1st Qu.: 1.000
##  Median :2012-01-24 00:00:00   Median : 605000   Median : 1.000
##  Mean   :2011-08-17 23:50:44   Mean   : 645051   Mean   : 1.107
##  3rd Qu.:2014-07-29 00:00:00   3rd Qu.: 749950   3rd Qu.: 1.000
##  Max.   :2016-12-16 00:00:00   Max.   :4311000   Max.   :18.000
##  sale_instrument  sale_warning         sitetype          addr_full
##  Min.   : 0.000   Length:10568       Length:10568       Length:10568
##  1st Qu.: 3.000   Class :character   Class :character   Class :character
##  Median : 3.000   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 3.147
##  3rd Qu.: 3.000
##  Max.   :26.000
##       zip5          ctyname            postalctyn             lon
##  Min.   :98052   Length:10568       Length:10568       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                                         Mean   :-122.1
##  3rd Qu.:98053                                         3rd Qu.:-122.0
##  Max.   :98074                                         Max.   :-121.9
##       lat        building_grade   square_feet_total_living    bedrooms
##  Min.   :47.46   Min.   : 2.000   Min.   :  240            Min.   : 0.000
##  1st Qu.:47.67   1st Qu.: 8.000   1st Qu.: 1870            1st Qu.: 3.000
##  Median :47.69   Median : 8.000   Median : 2450            Median : 4.000
##  Mean   :47.68   Mean   : 8.273   Mean   : 2545            Mean   : 3.482
##  3rd Qu.:47.71   3rd Qu.: 9.000   3rd Qu.: 3110            3rd Qu.: 4.000
##  Max.   :47.73   Max.   :13.000   Max.   :13540            Max.   :11.000
##  bath_full_count  bath_half_count  bath_3qtr_count    year_built
##  Min.   : 0.000   Min.   :0.0000   Min.   :0.0000   Min.   :1900
##  1st Qu.: 1.000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:1980
##  Median : 2.000   Median :1.0000   Median :0.0000   Median :1999
##  Mean   : 1.803   Mean   :0.6175   Mean   :0.5006   Mean   :1993
##  3rd Qu.: 2.000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:2007
##  Max.   :23.000   Max.   :6.0000   Max.   :8.0000   Max.   :2016
##  year_renovated    current_zoning       sq_ft_lot        prop_type
##  Min.   :   0.00   Length:10568       Min.   :    785   Length:10568
##  1st Qu.:   0.00   Class :character   1st Qu.:   5400   Class :character
##  Median :   0.00   Mode  :character   Median :   7850   Mode  :character
##  Mean   :  21.93                     Mean   :  19921
##  3rd Qu.:   0.00                     3rd Qu.:  12037
##  Max.   :2016.00                     Max.   :1631322
##   present_use
##  Min.   :  0.000
##  1st Qu.:  2.000
##  Median :  2.000
##  Mean   :  6.546
##  3rd Qu.:  2.000
##  Max.   :300.000
```

b. 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.

```
# This is Simple Linear Regression Model
saleprice_slm <- lm(cleaned_housing_df$‘Sale Price‘ ~ cleaned_housing_df$sq_ft_lot, cleaned_housing_df)

print("Correlation of Sale Price and square_feet_total_living ")
```

```
## [1] "Correlation of Sale Price and square_feet_total_living "
```

```
cor(cleaned_housing_df$‘Sale Price‘,cleaned_housing_df$square_feet_total_living)
```

```
## [1] 0.707278
```

```
print("Correlation of Sale Price and bedrooms")
```

```
## [1] "Correlation of Sale Price and bedrooms"
```

```
cor(cleaned_housing_df$‘Sale Price‘,cleaned_housing_df$bedrooms)
```

```
## [1] 0.3299898
```

```
print("Correlation of Sale Price and bath_full_count")
```

```
## [1] "Correlation of Sale Price and bath_full_count"
```

```
cor(cleaned_housing_df$‘Sale Price‘,cleaned_housing_df$bath_full_count)
```

```
## [1] 0.3827874
```

```
print("Correlation of Sale Price and bath_half_count")
```

```
## [1] "Correlation of Sale Price and bath_half_count"
```

```
cor(cleaned_housing_df$‘Sale Price‘,cleaned_housing_df$bath_half_count)
```

```
## [1] 0.2246326
```

```
print("Correlation of Sale Price and bath_3qtr_count")
```

```
## [1] "Correlation of Sale Price and bath_3qtr_count"
```

```
cor(cleaned_housing_df$‘Sale Price‘,cleaned_housing_df$bath_3qtr_count)
```

```
## [1] 0.09751304
```

```
print("Correlation of Sale Price and year_built")
```

```
## [1] "Correlation of Sale Price and year_built"
```

```
cor(cleaned_housing_df$‘Sale Price‘,cleaned_housing_df$year_built)
```

```
## [1] 0.2595616
```

```
print("Correlation of Sale Price and year_renovated")
```

```
## [1] "Correlation of Sale Price and year_renovated"
```

```
cor(cleaned_housing_df$‘Sale Price‘,cleaned_housing_df$year_renovated)
```

```
## [1] 0.05747795
```

Based on the Correlation between Sales price and other variables, I am picking the one’s with correlation over 0.2 and feeding them into the model

```
# This is Multiple Linear Regression Model
saleprice_mlm <- lm(cleaned_housing_df$‘Sale Price‘ ~ cleaned_housing_df$square_feet_total_living + clea
```

c. 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?

```
summary(saleprice_slm)
```

```
##
## Call:
## lm(formula = cleaned_housing_df$‘Sale Price‘ ~ cleaned_housing_df$sq_ft_lot,
##     data = cleaned_housing_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2615922  -151493   -35572   106230  3293158
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  6.205e+05  2.598e+03   238.9   <2e-16 ***
## cleaned_housing_df$sq_ft_lot 1.232e+00  4.830e-02    25.5   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 248100 on 10566 degrees of freedom
## Multiple R-squared:  0.05799,    Adjusted R-squared:  0.0579
## F-statistic: 650.5 on 1 and 10566 DF,  p-value: < 2.2e-16
```

```
summary(saleprice_mlm)
```

```
##
## Call:
## lm(formula = cleaned_housing_df$‘Sale Price‘ ~ cleaned_housing_df$square_feet_total_living +
##     cleaned_housing_df$bedrooms + cleaned_housing_df$bath_full_count +
##     cleaned_housing_df$bath_half_count + cleaned_housing_df$year_built)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1882432   -82773   -13207    63887  3832295
##
## Coefficients:
##                                               Estimate Std. Error t value
## (Intercept)                                 500325.803 244451.214   2.047
## cleaned_housing_df$square_feet_total_living    208.013      2.677  77.695
## cleaned_housing_df$bedrooms                 -35931.629   2540.274 -14.145
## cleaned_housing_df$bath_full_count           11867.822   3399.441   3.491
## cleaned_housing_df$bath_half_count            9595.764   3548.283   2.704
## cleaned_housing_df$year_built                 -143.926    123.407  -1.166
##                                             Pr(>|t|)
## (Intercept)                                 0.040709 *
## cleaned_housing_df$square_feet_total_living  < 2e-16 ***
## cleaned_housing_df$bedrooms                  < 2e-16 ***
## cleaned_housing_df$bath_full_count          0.000483 ***
## cleaned_housing_df$bath_half_count          0.006855 **
## cleaned_housing_df$year_built               0.243531
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 178800 on 10562 degrees of freedom
## Multiple R-squared:  0.5109, Adjusted R-squared:  0.5107
## F-statistic:  2207 on 5 and 10562 DF,  p-value: < 2.2e-16
```

The R2 of model tells how successfully we are predicting the model. Higher the R2 value, means better the Correlation coefficient, which is square root of R2. So based on the values from two models, we may say that the first model which has value of 0.05799, which means square foot of the lot only contributes 5.8% to the sales price. However in the other model, other attributes together the R2 value is 0.5109 contribute approx 51% towards the sale price.

The Adjusted R2 gives an idea how well our model generalizes, and ideally we expect a similar value or close to R2. And in both our models, this value is very minimal. This difference tells if the model was derived from the population rather than sample, it would account for (diffX100)% less variance in the outcome. For both of our models R2 and Adjusted R2 is very similar which indicates that cross-validity of the model is good.

d. 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?

```
library(QuantPsyc)
```

```
## Loading required package: boot

## Loading required package: MASS

##
## Attaching package: 'QuantPsyc'

## The following object is masked from 'package:base':
##
##     norm
```

```
lm.beta(saleprice_mlm)
```

```
## cleaned_housing_df$square_feet_total_living
##                                 0.760671787
##                  cleaned_housing_df$bedrooms
##                                -0.122206339
##           cleaned_housing_df$bath_full_count
##                                 0.029636829
##           cleaned_housing_df$bath_half_count
##                                 0.019305856
##                cleaned_housing_df$year_built
##                                -0.009325044
```

In general, it tells that if the specific attribute changes by one standard deviation, then the sales price(or outcome variable) increase by the Standardized Beta times(the value it displays) the standard deviation. If Beta is -ve, it means decreases by same factor of Standard Deviation.

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

```r
confint(saleprice_mlm)
```

```
##                                                   2.5 %      97.5 %
## (Intercept)                                  21155.3173 979496.2895
## cleaned_housing_df$square_feet_total_living    202.7650    213.2611
## cleaned_housing_df$bedrooms                 -40911.0449 -30952.2126
## cleaned_housing_df$bath_full_count            5204.2765  18531.3674
## cleaned_housing_df$bath_half_count            2640.4590  16551.0688
## cleaned_housing_df$year_built                 -385.8282     97.9753
```

From the confidence interval values here we can say that 1. square_feet_total_living 2. bedrooms 3. bath_full_count 4. bath_half_count

are on the same side of Zero be it, 2.5 percentile value or 97.5 percentile. So these are fine.

The gap between square_feet_total_living is tight, so seems its estimates using this are more likely representing the true population. However the bedrooms, bath_full_count and batch_half_count are less representatives.

The last value that is year_built is crossing the zero from 2.5 percentile to 97.5 percentile, so this may be a bad attribute to predict.

f. 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.

```r
anova(saleprice_slm, saleprice_mlm)
```

```
## Analysis of Variance Table
##
## Model 1: cleaned_housing_df$‘Sale Price‘ ~ cleaned_housing_df$sq_ft_lot
## Model 2: cleaned_housing_df$‘Sale Price‘ ~ cleaned_housing_df$square_feet_total_living +
##     cleaned_housing_df$bedrooms + cleaned_housing_df$bath_full_count +
##     cleaned_housing_df$bath_half_count + cleaned_housing_df$year_built
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
## 1  10566 6.5023e+14
## 2  10562 3.3758e+14  4 3.1265e+14 2445.5 < 2.2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

The F(4,10562) = 2445.5 for p<0.001 So the Fit of the model has significantly improved from the original model.

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

```
# Outliers
cleaned_housing_df$residuals <- resid(saleprice_mlm)
cleaned_housing_df$standardized.residuals <- rstandard(saleprice_mlm)
cleaned_housing_df$rstudent <- rstudent(saleprice_mlm)

# Influential Cases
cleaned_housing_df$cooks.distance <- cooks.distance(saleprice_mlm)
cleaned_housing_df$dfbeta <- dfbeta(saleprice_mlm)
cleaned_housing_df$dffits <- dffits(saleprice_mlm)
cleaned_housing_df$leverage <- hatvalues(saleprice_mlm)
cleaned_housing_df$covariance.ratios <- covratio(saleprice_mlm)
```

h. Calculate the standardized residuals using the appropriate command, specifying those that are +-2, storing the results of large residuals in a variable you create.

```
cleaned_housing_df$large.residuals<-cleaned_housing_df$standardized.residuals > 2 | cleaned_housing_df$
```

i. Use the appropriate function to show the sum of large residuals.

```
sum(cleaned_housing_df$large.residuals)
```

```
## [1] 376
```

j. Which specific variables have large residuals (only cases that evaluate as TRUE)?

```
cleaned_housing_df[cleaned_housing_df$large.residuals, c("Sale Price", "square_feet_total_living", "bed
```

```
## # A tibble: 376 x 6
##    `Sale Price` square_feet_tot~ bedrooms bath_full_count bath_half_count
##           <dbl>            <dbl>    <dbl>           <dbl>           <dbl>
##  1       265000             4920        4               4               1
##  2      1392000             3740        4               3               2
##  3      1080135             2700        3               2               0
##  4       732500             5710        5               3               2
##  5      1390000              660        0               1               0
##  6      1390000             3280        3               2               0
##  7       370000             4000        4               3               1
##  8       390000             5800        5               4               1
##  9      1588359             3360        2               2               1
## 10      1450000             3480        3               2               1
## # ... with 366 more rows, and 1 more variable: year_built <dbl>
```

k. Investigate further by calculating the leverage, cooks distance, and covariance rations. Comment on all cases that are problematics.

```
cleaned_housing_df[cleaned_housing_df$large.residuals, c("cooks.distance", "leverage", "covariance.rati
```

```
## # A tibble: 376 x 3
##    cooks.distance leverage covariance.ratios
##             <dbl>    <dbl>             <dbl>
##  1       0.00632  0.00155              0.988
##  2       0.00131  0.00104              0.997
##  3       0.000298 0.000373             0.998
##  4       0.00344  0.00220              0.997
##  5       0.0130   0.00238              0.984
##  6       0.000920 0.000525             0.995
##  7       0.000852 0.000496             0.995
##  8       0.00747  0.00174              0.988
##  9       0.00238  0.000888             0.992
## 10       0.00153  0.000861             0.995
## # ... with 366 more rows
```

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

```
library(car)
```

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

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

```
## The following object is masked from 'package:boot':
##
##     logit
```

```
durbinWatsonTest(saleprice_mlm)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.2572229      1.485527       0
##  Alternative hypothesis: rho != 0
```

As per the Durbin Watson Test, if the values is in between 1-3, the model is considered good. Closer the value to 2, better the model.

m. Perform the necessary calculations to assess the assumption of no multicollinearity and state if the condition is met or not.

```
vif(saleprice_mlm)
```

```
## cleaned_housing_df$square_feet_total_living
##                                    2.070122
##                   cleaned_housing_df$bedrooms
##                                    1.612053
##            cleaned_housing_df$bath_full_count
##                                    1.556399
##            cleaned_housing_df$bath_half_count
##                                    1.100627
##                 cleaned_housing_df$year_built
##                                    1.380662
```

```
print("Tolerance = 1/VIF")
```

```
## [1] "Tolerance = 1/VIF"
```

```
1/vif(saleprice_mlm)
```

```
## cleaned_housing_df$square_feet_total_living
##                                   0.4830634
##                   cleaned_housing_df$bedrooms
##                                   0.6203271
##            cleaned_housing_df$bath_full_count
##                                   0.6425089
##            cleaned_housing_df$bath_half_count
##                                   0.9085731
##                 cleaned_housing_df$year_built
##                                   0.7242903
```

```
print("Mean VIF")
```

```
## [1] "Mean VIF"
```

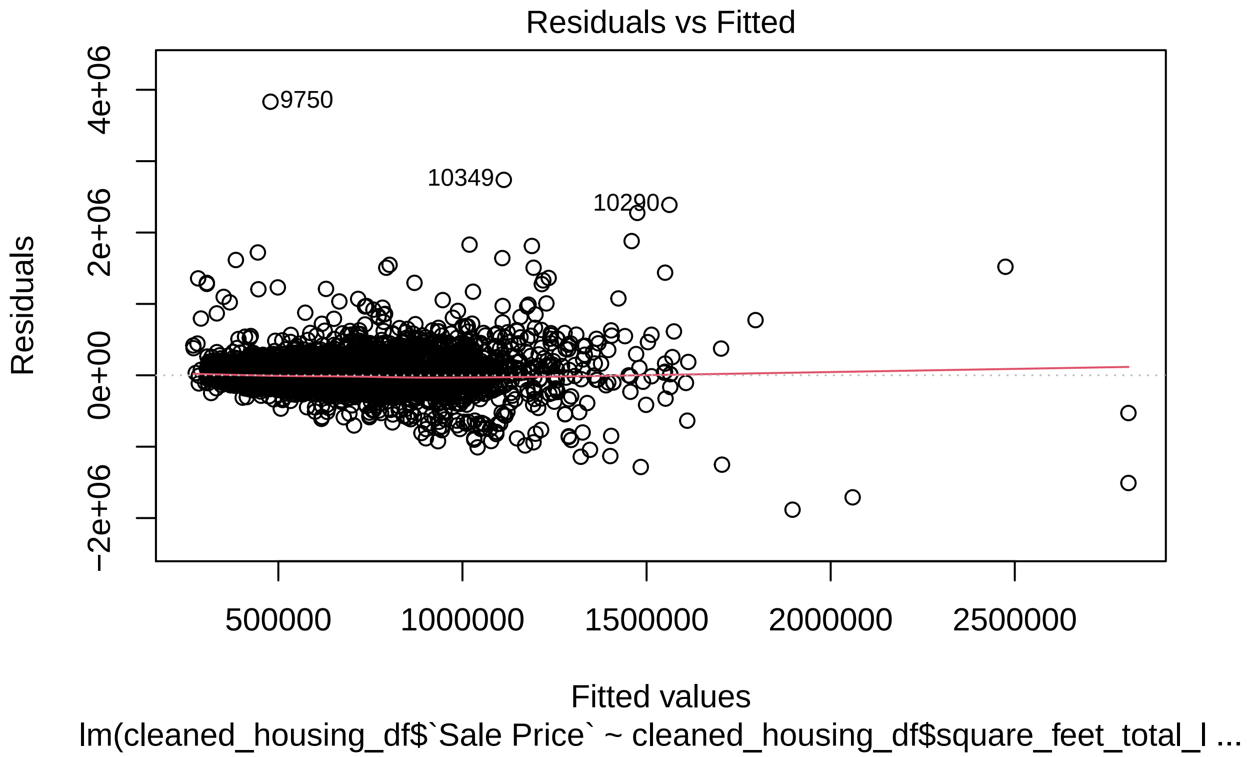
```
mean(vif(saleprice_mlm))
```

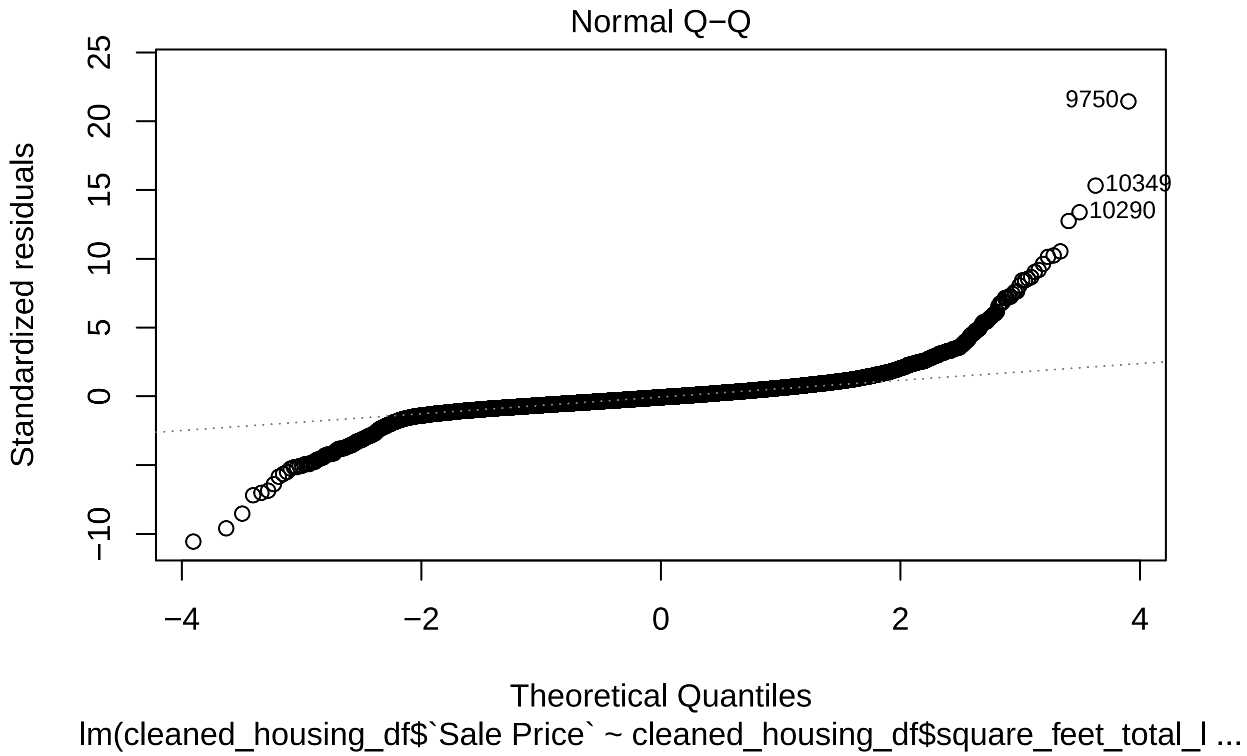
```
## [1] 1.543972
```

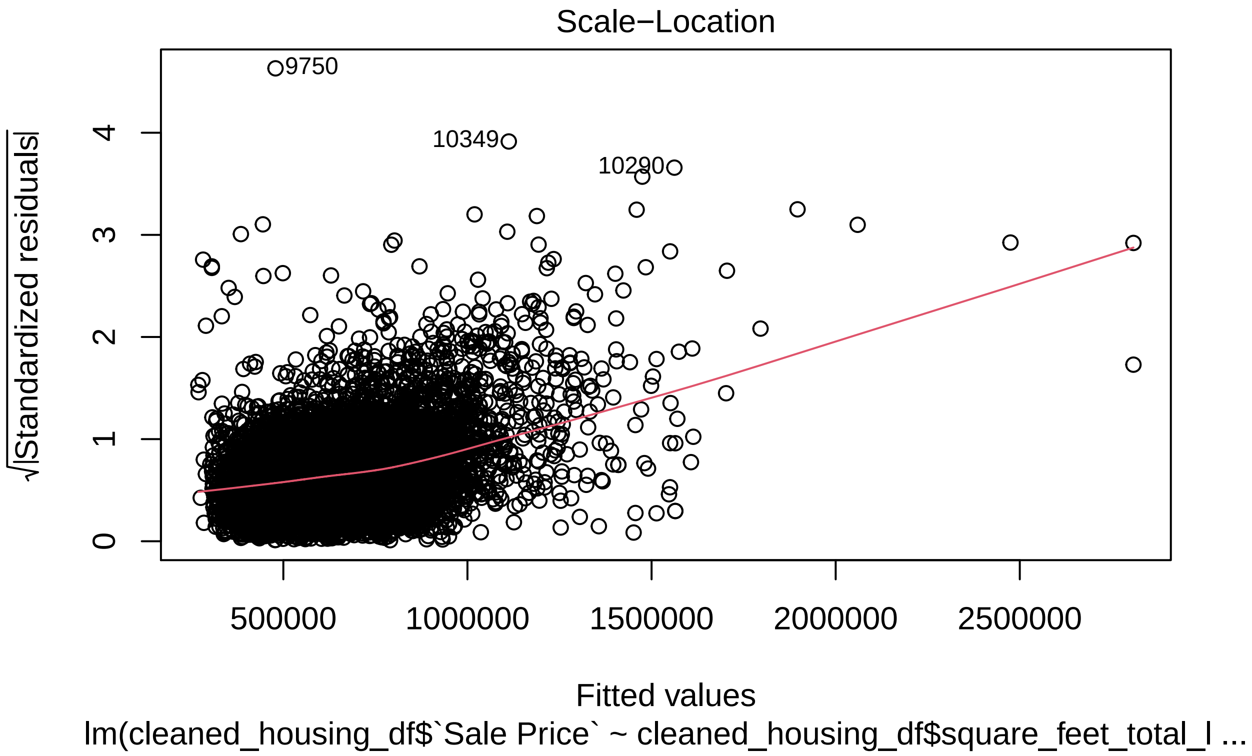
Is Largest VIF > 10 ? NO - So no cause for concern Avg VIF is 1.54, which is not substantially greater than 1. (Substantially more is considered more than 2.5, as from https://statisticalhorizons.com/multicollinearity) All Tolerance are above 0.2, meaning it should be fine.(Less than 0.2 is potential problem, less than 0.1 is significant problem. Its same as VIF >10, as tolerance = 1/VIF)

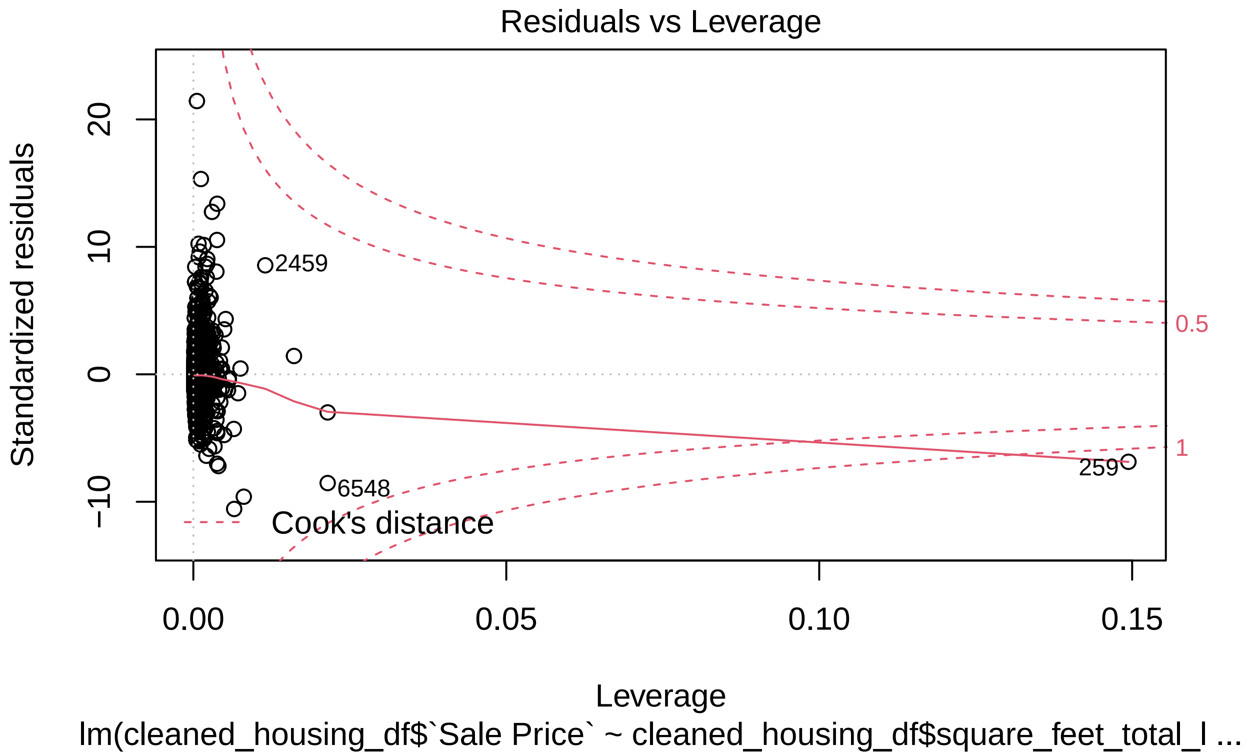
n. Visually check the assumptions related to the residuals using the plot() and hist() functions. Summarize what each graph is informing you of and if any anomalies are present.

```
plot(saleprice_mlm)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(cleaned_housing_df$`Sale Price` ~ cleaned_housing_df$square_feet_total_l ...
25
20
15
10
5
0
−10
−4
−2
0
2
4
9750
10349
10290

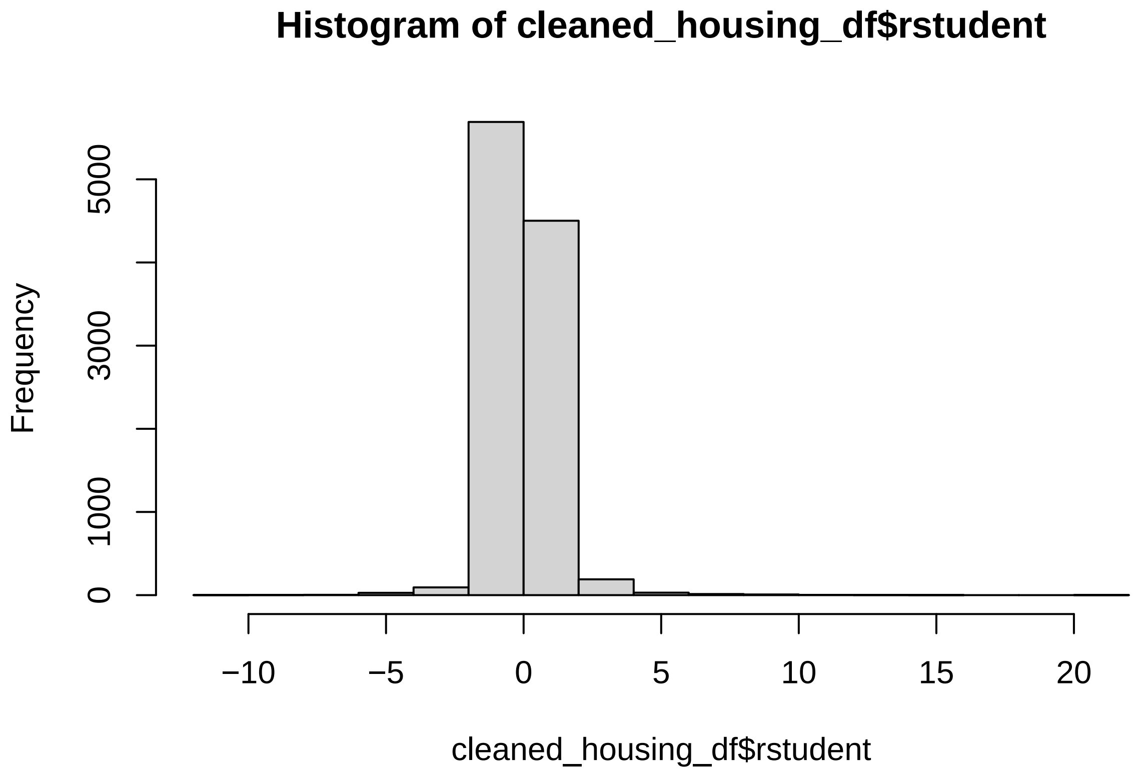

Scale−Location
√|Standardized residuals|
0
1
2
3
4
9750
10349
10290
500000
1000000
1500000
2000000
2500000
Fitted values
lm(cleaned_housing_df$`Sale Price` ~ cleaned_housing_df$square_feet_total_l ...



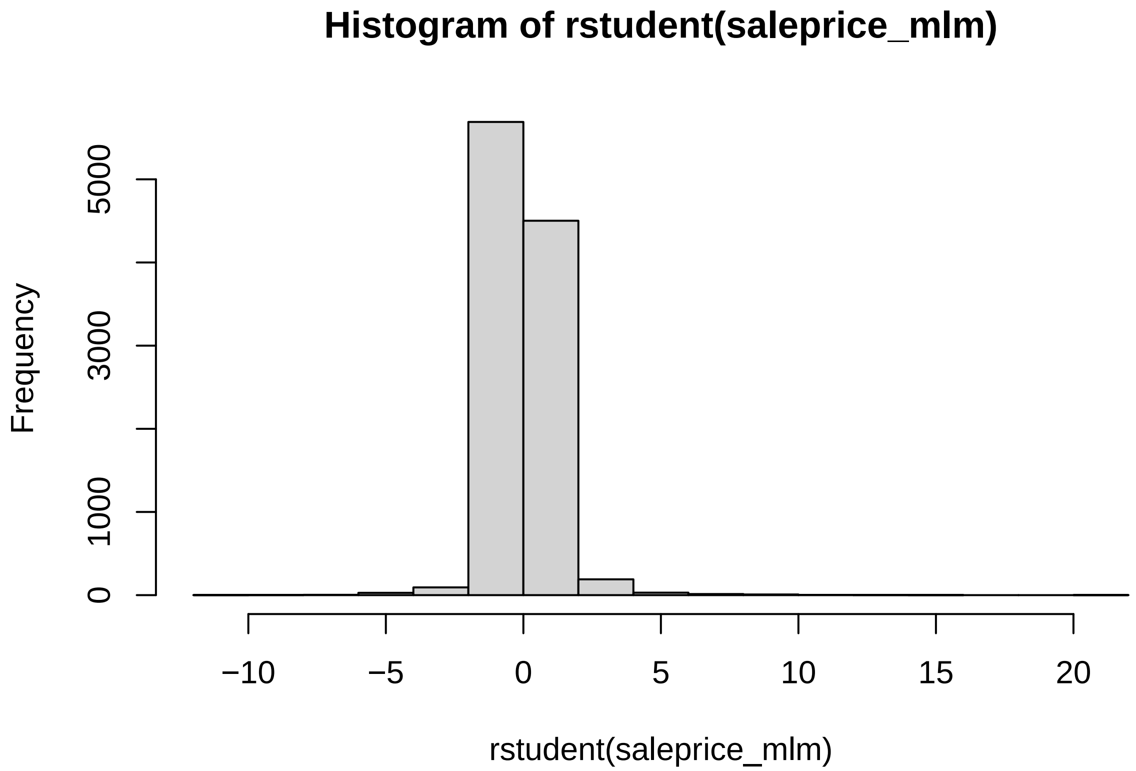
The Residuals Vs Fitted Graph shows random dots evenly dispersed around 0. Though not fully dispersed but evenly dispersed. It does not funnel out, so there is no heteroscedasticity. The data points also dont form a curve, so should be linear.

With the QQ plot we see that the plot curves of at extremes, so it means has more extreme values than would be expected if they truly came from a Normal distribution.

```
hist(cleaned_housing_df$rstudent)
```



```
hist(rstudent(saleprice_mlm))
```



Looks like a Bell slight skewed towards right. Could be assumed Normal.

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

As we see from the QQ plot that the plot curves away in opposite directions when approaching extreme values. This means there are outliers present at extremes. This tells that the model could be biased.

Secondly, as we saw with year_built attribute the confint() output shows to affect the model in a bad way.

So based on these two, we can say that we have bias present in this model.

If the model is unbiased, it means that it holds true for both sample as well as it could be used confidently over the entire population.

To make this model better 1. We should try to clean the outliers based on the analysis so far. 2. We should also try to re-look at the parameters being used in the model. The one's which have bad effect on the model, should be removed. Additional parameters should also be added, if needed to improve the model.