Exercise 12 - Housing Data

Daniel Angel

2021-1-31

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.

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

I first removed obvious outliers or potential problem data points which in my eyes were sales that had zero bedrooms, sales that had a non-empty 'Sale Warning' entry, and sales with excessive bathrooms(over 20) or no bathrooms. Then, I removed all Character Vectors or Columns which had no potential statistical significance or discernible meaning. Subsequently, I limited building grade to values between 5 and 10. Finally, I converted 5 digit zip codes (Zip5) into a character vector so as not to be confused by any results later on.

Lastly, I combined the bathroom counts for simplicity's sake.

The result of all these changes are that I reduced my data frame from 12865 objects of 24 variables into a much cleaner 10226 objects of 8 variables.

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.

I based my selections on intuition, knowledge of real estate and higher correlation scores.

```
corr <- cor(subset(clean_housing_df, select = -c(zip5)))

simp_lm <- lm(formula = clean_housing_df$`Sale Price` ~ clean_housing_df$sq_ft_lot, data = clean_housing

mult_lm <- lm(formula = clean_housing_df$`Sale Price` ~ clean_housing_df$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?

```r
summary(simp_lm)
```

```
##
## Call:
## lm(formula = clean_housing_df$`Sale Price` ~ clean_housing_df$sq_ft_lot,
##     data = clean_housing_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1118336  -138761   -24395   111120  3327672
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 6.080e+05  2.216e+03  274.36   <2e-16 ***
## clean_housing_df$sq_ft_lot 8.829e-01  4.859e-02   18.17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 206200 on 10224 degrees of freedom
## Multiple R-squared:  0.03128,    Adjusted R-squared:  0.03118
## F-statistic: 330.1 on 1 and 10224 DF,  p-value: < 2.2e-16
```

```r
summary(mult_lm)
```

```
##
## Call:
## lm(formula = clean_housing_df$`Sale Price` ~ clean_housing_df$square_feet_total_living +
##     clean_housing_df$sq_ft_lot + clean_housing_df$building_grade +
##     clean_housing_df$bedrooms + clean_housing_df$total_bath,
##     data = clean_housing_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1249731   -75779   -12889    60921  3659962
##
## Coefficients:
##                                           Estimate Std. Error t value
## (Intercept)                             -1.705e+05  1.635e+04 -10.431
## clean_housing_df$square_feet_total_living  1.183e+02  3.493e+00  33.858
## clean_housing_df$sq_ft_lot                 3.792e-01  3.611e-02  10.501
## clean_housing_df$building_grade            5.753e+04  2.238e+03  25.705
## clean_housing_df$bedrooms                 -1.407e+04  2.212e+03  -6.363
## clean_housing_df$total_bath                2.987e+04  3.836e+03   7.786
##                                           Pr(>|t|)
## (Intercept)                                < 2e-16 ***
## clean_housing_df$square_feet_total_living  < 2e-16 ***
## clean_housing_df$sq_ft_lot                 < 2e-16 ***
## clean_housing_df$building_grade            < 2e-16 ***
## clean_housing_df$bedrooms                 2.06e-10 ***
## clean_housing_df$total_bath               7.61e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 149100 on 10220 degrees of freedom
## Multiple R-squared:  0.4933, Adjusted R-squared:  0.493
## F-statistic:  1990 on 5 and 10220 DF,  p-value: < 2.2e-16
```

The multiple factor model is better the R2 statistics are better and it does help explain some of the large variations in Sale Price.

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?

```
## clean_housing_df$square_feet_total_living
##                                0.45955043
##                clean_housing_df$sq_ft_lot
##                                0.07595856
##           clean_housing_df$building_grade
##                                0.25078753
##                 clean_housing_df$bedrooms
##                               -0.05749711
##               clean_housing_df$total_bath
##                                0.08222064
```

The beta tells us what number or portion of standard deviations the outcome will be altered by the change of one standard deviation in the predicting variable. Based on this, the square footage of the living space and the grade of the building are the best variables that the help the model perform a successful prediction.

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

```
##                                                 2.5 %        97.5 %
## (Intercept)                                -2.025343e+05 -1.384542e+05
## clean_housing_df$square_feet_total_living   1.114167e+02  1.251105e+02
## clean_housing_df$sq_ft_lot                  3.084131e-01  4.499854e-01
## clean_housing_df$building_grade             5.314238e+04  6.191659e+04
## clean_housing_df$bedrooms                  -1.840726e+04 -9.737290e+03
## clean_housing_df$total_bath                 2.234614e+04  3.738451e+04
```

The confidence interval means that at these percentiles the confidence that the model is making a good prediction is within this range sort of like when a survey lists a plus/minus error.

None of these variables are horrible at helping the predictive model but the best are certainly square footage of the living space and the lot size. When the confidence interval is lower it means the value of the beta for the model is close to the actual value of beta from the true data.

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.

```
## Analysis of Variance Table
```

```
##
## Model 1: clean_housing_df$`Sale Price` ~ clean_housing_df$sq_ft_lot
## Model 2: clean_housing_df$`Sale Price` ~ clean_housing_df$square_feet_total_living +
##     clean_housing_df$sq_ft_lot + clean_housing_df$building_grade +
##     clean_housing_df$bedrooms + clean_housing_df$total_bath
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
## 1  10224 4.3463e+14
## 2  10220 2.2735e+14  4 2.0728e+14 2329.4 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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.

```r
# Outliars
clean_housing_df$residuals <- resid(mult_lm)
clean_housing_df$standardized.residuals <- rstandard(mult_lm)
clean_housing_df$studentized.residuals <- rstudent(mult_lm)
# Influential Cases
clean_housing_df$cooks.distance <- cooks.distance(mult_lm)
clean_housing_df$dfbeta <- dfbeta(mult_lm)
clean_housing_df$dffit <- dffits(mult_lm)
clean_housing_df$leverage <- hatvalues(mult_lm)
clean_housing_df$covariance.ratios <- covratio(mult_lm)
summary(clean_housing_df)
```

```
##    Sale Price          zip5           building_grade   square_feet_total_living
##  Min.   :   2500   Length:10226       Min.   : 5.000   Min.   : 430
##  1st Qu.: 483000   Class :character   1st Qu.: 8.000   1st Qu.:1850
##  Median : 598750   Mode  :character   Median : 8.000   Median :2410
##  Mean   : 623743                      Mean   : 8.183   Mean   :2471
##  3rd Qu.: 735000                      3rd Qu.: 9.000   3rd Qu.:3040
##  Max.   :4311000                      Max.   :10.000   Max.   :9360
##     bedrooms        year_built     sq_ft_lot         total_bath
##  Min.   : 1.000   Min.   :1900   Min.   :    785   Min.   :0.500
##  1st Qu.: 3.000   1st Qu.:1980   1st Qu.:   5318   1st Qu.:2.250
##  Median : 4.000   Median :2000   Median :   7697   Median :2.500
##  Mean   : 3.465   Mean   :1993   Mean   :  17861   Mean   :2.453
##  3rd Qu.: 4.000   3rd Qu.:2007   3rd Qu.:  11250   3rd Qu.:2.750
##  Max.   :10.000   Max.   :2016   Max.   :1166246   Max.   :8.250
##    residuals        standardized.residuals studentized.residuals
##  Min.   :-1249731   Min.   :-8.44440       Min.   :-8.473604
##  1st Qu.:  -75779   1st Qu.:-0.50813       1st Qu.:-0.508115
##  Median :  -12889   Median :-0.08643       Median :-0.086423
##  Mean   :       0   Mean   : 0.00002       Mean   : 0.000165
##  3rd Qu.:   60921   3rd Qu.: 0.40855       3rd Qu.: 0.408537
##  Max.   : 3659962   Max.   :24.66478       Max.   :25.432114
##  cooks.distance
##  Min.   :0.0000000
##  1st Qu.:0.0000031
##  Median :0.0000142
##  Mean   :0.0005194
```

```
##  3rd Qu.:0.0000483
##  Max.   :1.0430054
##   dfbeta.(Intercept)    dfbeta.clean_housing_df$square_feet_total_living  dfbeta.clean_housing_df$sq
##  Min.   :-7543.671      Min.   :-3.542816      Min.   :-0.06108803    Min.   :-768.5224      Min.
##  1st Qu.:  -37.739      1st Qu.:-0.006624      1st Qu.:-0.00003524    1st Qu.:  -5.9454      1st Qu.
##  Median :    1.002      Median : 0.000542      Median : 0.00000050    Median :  -0.2170      Median
##  Mean   :   -0.050      Mean   :-0.000024      Mean   : 0.00000058    Mean   :   0.0105      Mean
##  3rd Qu.:   39.987      3rd Qu.: 0.009560      3rd Qu.: 0.00003152    3rd Qu.:   4.7787      3rd Qu.
##  Max.   : 5864.613      Max.   : 1.656415      Max.   : 0.08843672    Max.   :1113.8359      Max.
##      dffit               leverage         covariance.ratios
##  Min.   :-1.7284683   Min.   :0.0001284   Min.   :0.6995
##  1st Qu.:-0.0099649   1st Qu.:0.0002802   1st Qu.:1.0006
##  Median :-0.0016968   Median :0.0004291   Median :1.0008
##  Mean   : 0.0002582   Mean   :0.0005867   Mean   :1.0006
##  3rd Qu.: 0.0080928   3rd Qu.:0.0006245   3rd Qu.:1.0010
##  Max.   : 2.5794325   Max.   :0.0786096   Max.   :1.0475
```

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.

```
clean_housing_df$large.residual <- clean_housing_df$standardized.residuals > 2 | clean_housing_df$standa

big_boiz <- clean_housing_df$large.residual
```

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

```
sum(big_boiz)
```

```
## [1] 327
```

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

```
clean_housing_df[big_boiz, c("Sale Price", "square_feet_total_living", "bedrooms", "building_grade", "t
```

```
## # A tibble: 327 x 6
##    `Sale Price` square_feet_total_~ bedrooms building_grade total_bath sq_ft_lot
##           <dbl>               <dbl>    <dbl>          <dbl>      <dbl>     <dbl>
##  1       165000                1850        3              9        2       278891
##  2       265000                4920        4             10        4.5     112650
##  3      1392000                3740        4              9        4.75     17291
##  4      1053649                2680        2              9        2.5       8517
##  5      1080135                2700        3              9        2.75      7694
##  6       732500                5710        5              9        4.75     10200
##  7      1390000                3280        3             10        2.75    225640
```

```
##  8       650000                3960        4              9       4       217800
##  9       370000                4000        4              9       3.5      11780
## 10      1588359                3360        2              9       2.5       8752
## # ... with 317 more rows
```

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

```
clean_housing_df[big_boiz, c("cooks.distance", "leverage", "covariance.ratios")]
```

```
## # A tibble: 327 x 3
##    cooks.distance leverage covariance.ratios
##             <dbl>    <dbl>             <dbl>
##  1       0.00981  0.00472             0.998
##  2       0.0110   0.00205             0.984
##  3       0.00424  0.00216             0.996
##  4       0.000594 0.000686            0.998
##  5       0.000286 0.000278            0.997
##  6       0.00279  0.00278             1.00
##  7       0.00491  0.00293             0.998
##  8       0.00232  0.00317             1.00
##  9       0.00114  0.000598            0.995
## 10       0.00562  0.00119             0.985
## # ... with 317 more rows
```

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

```
library(car)
durbinWatsonTest(mult_lm)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.3112329      1.377493       0
##  Alternative hypothesis: rho != 0
```

By the Durbin Watson Test,because the value is in the range of 1 to 3 it is alright.

m) Perform the necessary calculations to assess the assumption of no multi-collinearity and state if the condition is met or not.

```
print("Variance inflation factors")
```

```
## [1] "Variance inflation factors"
```

```
vif(mult_lm)
```

```
## clean_housing_df$square_feet_total_living
##                                  3.715590
##                clean_housing_df$sq_ft_lot
##                                  1.055331
##           clean_housing_df$building_grade
##                                  1.919820
##                 clean_housing_df$bedrooms
##                                  1.646692
##               clean_housing_df$total_bath
##                                  2.249276
```

```
print("Tolerance = 1/Variance inflation factor")
```

```
## [1] "Tolerance = 1/Variance inflation factor"
```

```
1/vif(mult_lm)
```

```
## clean_housing_df$square_feet_total_living
##                                  0.2691363
##                clean_housing_df$sq_ft_lot
##                                  0.9475700
##           clean_housing_df$building_grade
##                                  0.5208823
##                 clean_housing_df$bedrooms
##                                  0.6072782
##               clean_housing_df$total_bath
##                                  0.4445874
```

```
print("Mean Variance inflation factor")
```

```
## [1] "Mean Variance inflation factor"
```

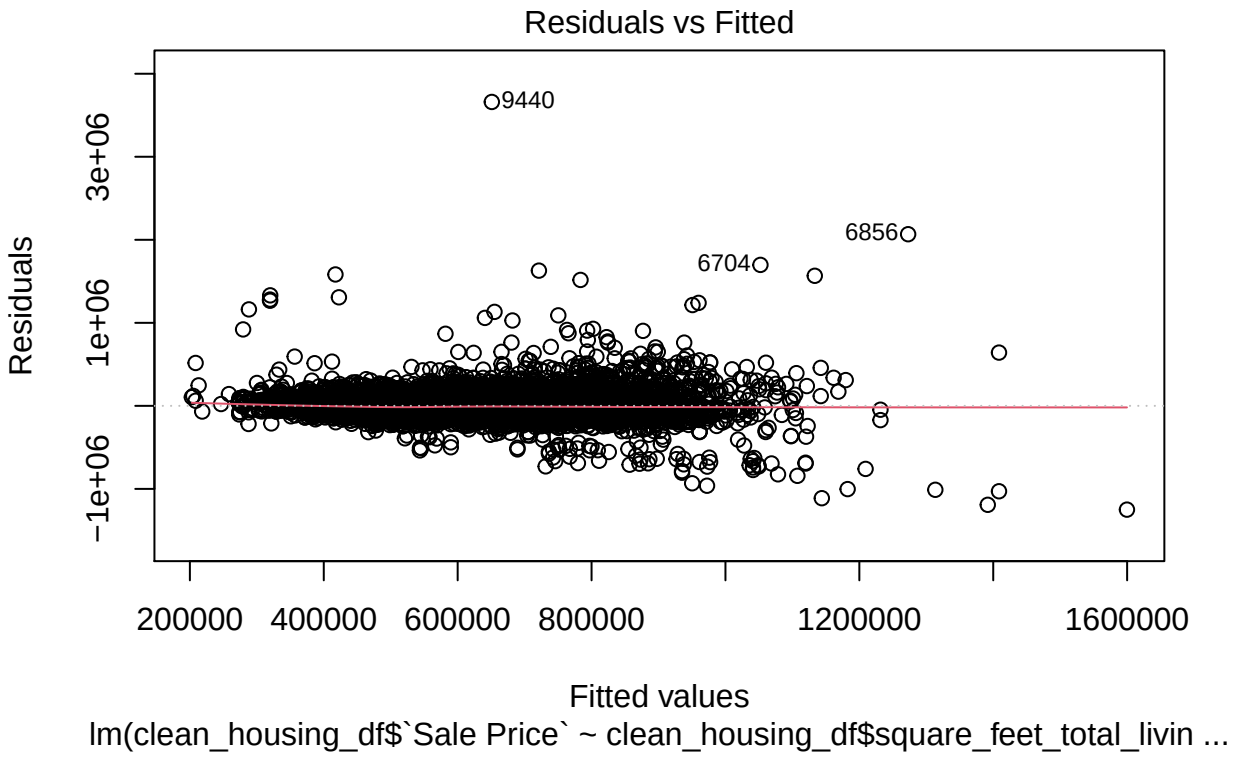
```
mean(vif(mult_lm))
```

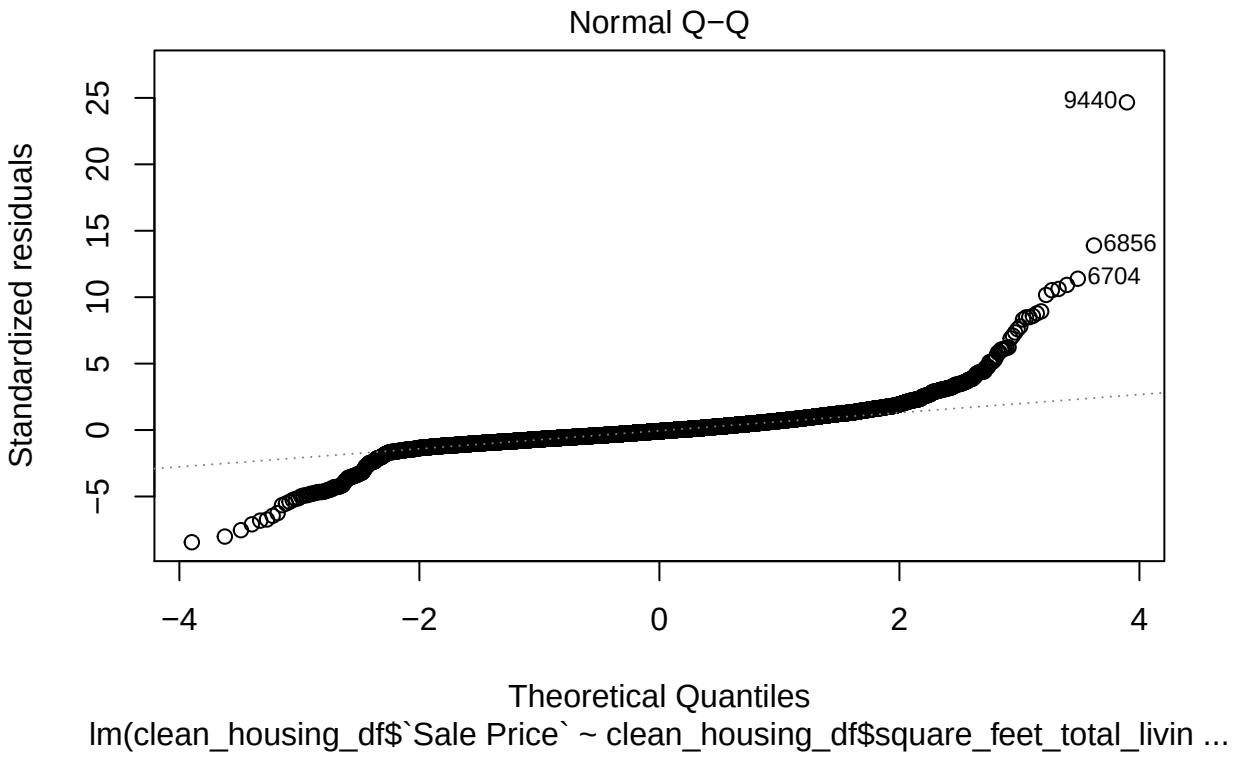
```
## [1] 2.117342
```

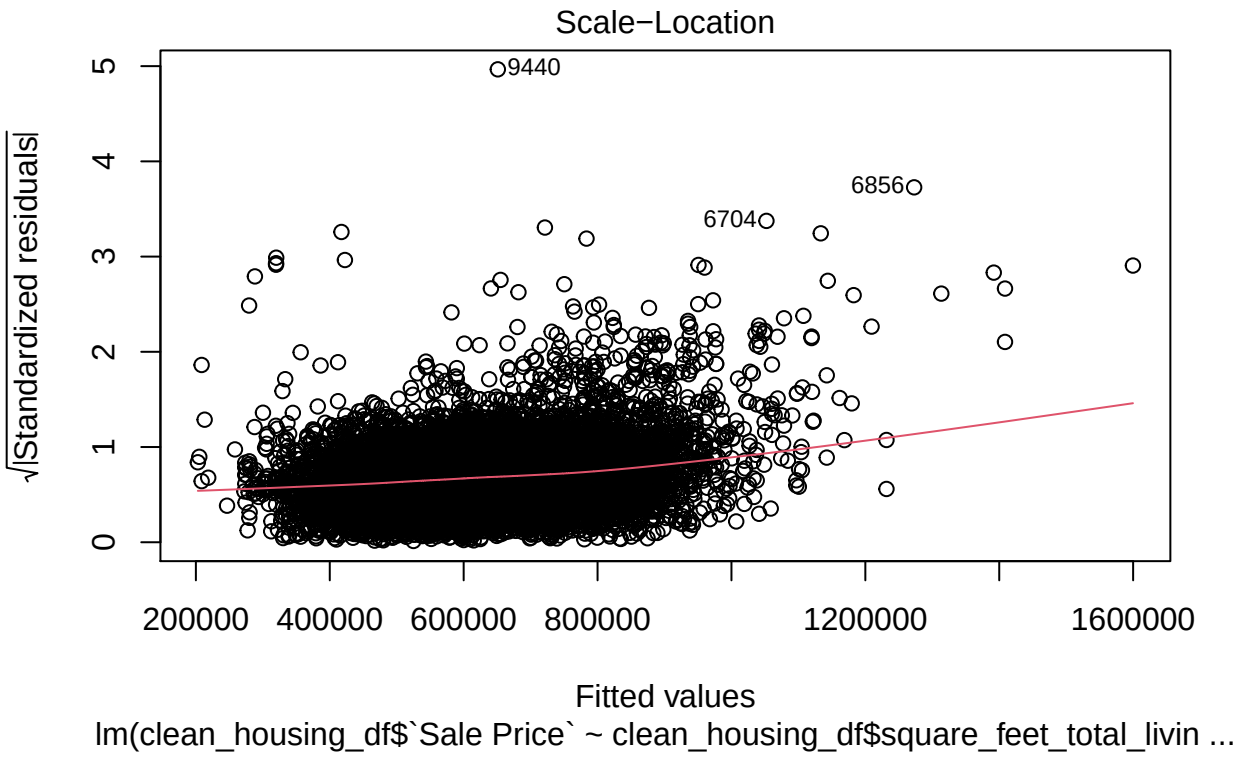
The largest VIF value is more than 10 and the least more than 0.2. Furthermore, the average of the Variance inflation factor is

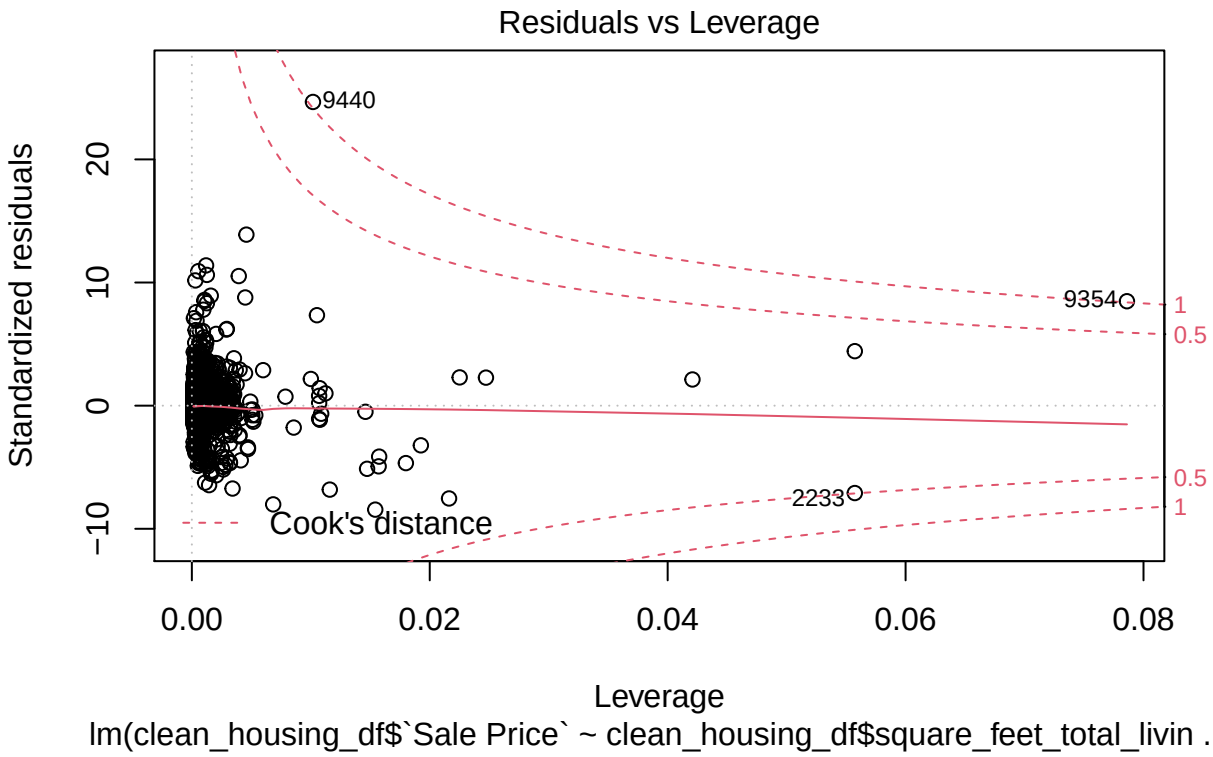
n) Visually check the assumptions related to the residuals using the plot() and hist() functions. Summarize what each graph is informing you of and if any anomalies are present.

```
plot(mult_lm)
```

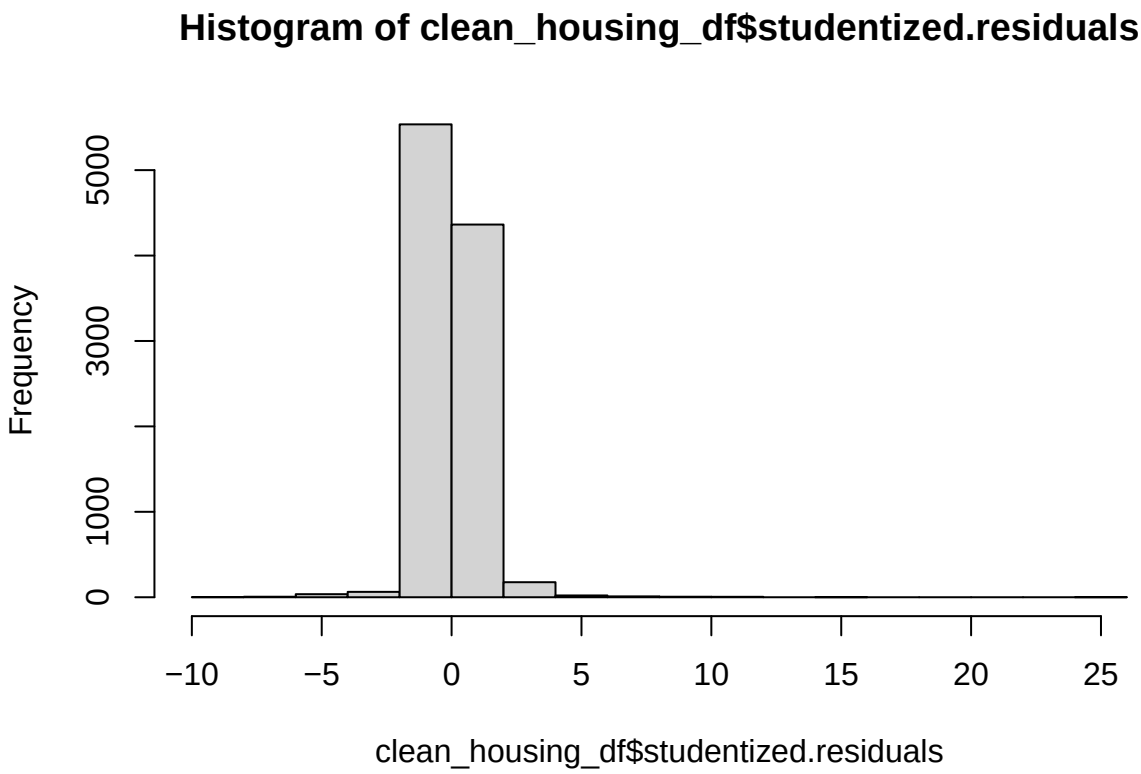

Residuals vs Fitted
9440
6856
6704
Residuals
3e+06
1e+06
−1e+06
200000
400000
600000
800000
1200000
1600000
Fitted values
lm(clean_housing_df$`Sale Price` ~ clean_housing_df$square_feet_total_livin ...


Normal Q−Q
Standardized residuals
25
20
15
10
5
0
−5
9440
6856
6704
−4
−2
0
2
4
Theoretical Quantiles
lm(clean_housing_df$`Sale Price` ~ clean_housing_df$square_feet_total_livin ...


Scale−Location
9440
6704
6856
√|Standardized residuals|
0
1
2
3
4
5
200000
400000
600000
800000
1200000
1600000
Fitted values
lm(clean_housing_df$`Sale Price` ~ clean_housing_df$square_feet_total_livin ...




```
hist(clean_housing_df$studentized.residuals)
```



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?

I don't think the model is biased. This means it should be pretty good at making predictions and also somewhat reflects what you might actual found out in society.