EXERCISE 8.2.3

Jahedur Rahman

2/3/2022

```
knitr::opts_chunk$set(echo = TRUE)
```

#i
Explain any transformations or modifications you made to the dataset

```
#load housing df
housing_dataset_df <- read_xlsx("C:/Users/jahed/OneDrive/Documents/GitHub/dsc520/data/week-7-housing.xls
#Data changes
colnames(housing_dataset_df)[c(1,2)]<-c("sale_date", "sale_price")
colnames(housing_dataset_df)
```

```
##  [1] "sale_date"                "sale_price"
##  [3] "sale_reason"              "sale_instrument"
##  [5] "sale_warning"             "sitetype"
##  [7] "addr_full"                "zip5"
##  [9] "ctyname"                  "postalctyn"
## [11] "lon"                      "lat"
## [13] "building_grade"           "square_feet_total_living"
## [15] "bedrooms"                 "bath_full_count"
## [17] "bath_half_count"          "bath_3qtr_count"
## [19] "year_built"               "year_renovated"
## [21] "current_zoning"           "sq_ft_lot"
## [23] "prop_type"                "present_use"
```

#ii
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.

```
saleprice_sqftlot <- (housing_dataset_df)[c(2,22)]
head(saleprice_sqftlot)
```

```
## # A tibble: 6 x 2
##   sale_price sq_ft_lot
##        <dbl>     <dbl>
## 1     698000      6635
## 2     649990      5570
## 3     572500      8444
## 4     420000      9600
## 5     369900      7526
## 6     184667      7280
```

```
saleprice_predictors <- (housing_dataset_df)[c(2,13,14,15,16,22)]
head(saleprice_predictors)
```

```
## # A tibble: 6 x 6
##   sale_price building_grade square_feet_tota~ bedrooms bath_full_count sq_ft_lot
##        <dbl>          <dbl>             <dbl>    <dbl>           <dbl>     <dbl>
## 1     698000              9              2810        4               2      6635
## 2     649990              9              2880        4               2      5570
## 3     572500              8              2770        4               1      8444
## 4     420000              8              1620        3               1      9600
## 5     369900              7              1440        3               1      7526
## 6     184667              7              4160        4               2      7280
```

#iii
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_lm1 <- lm(sale_price ~ sq_ft_lot, data = saleprice_sqftlot)
housing_lm2 <- lm(sale_price ~ building_grade
                  + square_feet_total_living + bedrooms
                  + bath_full_count + sq_ft_lot, data = saleprice_predictors)
summary(housing_lm1)
```

```
##
## Call:
## lm(formula = sale_price ~ sq_ft_lot, data = saleprice_sqftlot)
##
## 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_lm2)
```

```
##
## Call:
## lm(formula = sale_price ~ building_grade + square_feet_total_living +
##     bedrooms + bath_full_count + sq_ft_lot, data = saleprice_predictors)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -1915140  -113296   -42511    40683  3755382
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -3.726e+04  3.148e+04  -1.184   0.2366
## building_grade            3.781e+04  4.435e+03   8.526  < 2e-16 ***
## square_feet_total_living  1.493e+02  5.912e+00  25.255  < 2e-16 ***
## bedrooms                 -1.794e+04  4.494e+03  -3.991 6.61e-05 ***
## bath_full_count           3.705e+04  5.738e+03   6.456 1.11e-10 ***
## sq_ft_lot                 1.360e-01  5.771e-02   2.356   0.0185 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 358000 on 12859 degrees of freedom
## Multiple R-squared:  0.2166, Adjusted R-squared:  0.2163
## F-statistic: 711.2 on 5 and 12859 DF,  p-value: < 2.2e-16
```

The first model, "housing_lm1", has an R2 statistic which is the square of the correlation of the sale price and the lot size. This is a single variable regression. This shows us the variability in sale price that is affected by the lot size. The adjusted R2 statistics has the same measure. The difference is that it gives us information on how it could change if it was given more data. Since the values of the R2 statistic and the adjusted R2 statistic are close, there is good generalizability. The second model, "housing_lm2", has an R2 statistic and average R2 statistic that are higher than the first model. These results tell us there is about 21% variability in the sale price. Just like in the first model the R2 and adjusted R2 values are very close so there is good generalizability.

#iv
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?

```
lm.beta(housing_lm1)
```

```
## sq_ft_lot
## 0.1198122
```

```
lm.beta(housing_lm2)
```

```
##           building_grade square_feet_total_living                bedrooms
##               0.10216546               0.36546358              -0.03886233
##          bath_full_count                sq_ft_lot
##               0.05961918               0.01914434
```

The standardized betas for the linear model "housing_lm1" indicates the sale price increases by 0.11 standard deviations when there is an increase in standard deviations for the property's lot size by square feet. The standardized betas for the linear model "housing_lm2" indicates that the sale price is affected by each variable in a similar way. Square feet of total living space has the most predictive impact compared to all the variables.

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

```
confint(housing_lm1)
```

```
##                    2.5 %       97.5 %
## (Intercept) 6.343730e+05 6.492698e+05
## sq_ft_lot   7.291208e-01 9.728641e-01
```

```
confint(housing_lm2)
```

```
##                                 2.5 %        97.5 %
## (Intercept)              -9.896334e+04 24443.2970173
## building_grade            2.911905e+04 46504.0252857
## square_feet_total_living  1.377186e+02   160.8951316
## bedrooms                 -2.674661e+04 -9127.6129384
## bath_full_count           2.579848e+04 48291.8566980
## sq_ft_lot                 2.284809e-02     0.2491056
```

The confidence intervals calculated for "housing_lm1" have a small range. This indicates that the predictor's b value is close to the real b value. The confidence intervals calculated for "housing_lm2" have a larger range. In addition, these values cross zero and include negative values. This indicates that the sale price can increase or decrease depending on the number of bedrooms. This makes the output for sale price not consistent. However, the other variables do have better consistency and shorter range.

#vi
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.

```
anova(housing_lm1, housing_lm2)
```

```
## Analysis of Variance Table
##
## Model 1: sale_price ~ sq_ft_lot
## Model 2: sale_price ~ building_grade + square_feet_total_living + bedrooms +
##     bath_full_count + sq_ft_lot
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
## 1  12863 2.0734e+15
## 2  12859 1.6479e+15  4 4.2548e+14 830.03 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The analysis shows that the second model, "housing_lm2", has a p value below 0.001 so it did perform better with 4 degrees of freedom.

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

```
saleprice_predictors$standardized.residuals<-rstandard(housing_lm2)
saleprice_predictors$studentized.residuals<-rstudent(housing_lm2)
saleprice_predictors$cooks.distance<-cooks.distance(housing_lm2)
```

```
saleprice_predictors$dfbeta<-dfbeta(housing_lm2)
saleprice_predictors$dffit<-dffits(housing_lm2)
saleprice_predictors$leverage<-hatvalues(housing_lm2)
saleprice_predictors$covariance.ratios<-covratio(housing_lm2)
```

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

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

#ix
Use the appropriate function to show the sum of large residuals.

```
sum(saleprice_predictors$large.residual)
```

```
## [1] 322
```

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

```
large_res<-saleprice_predictors%>%
  filter(large.residual == 1)
```

#xi
Investigate further by calculating the leverage, cooks distance, and covariance ratios. Comment on all cases that are problematic.

```
#Percentage of sample with residuals over (+/-)2
nrow(housing_dataset_df)
```

```
## [1] 12865
```

```
nrow(large_res)
```

```
## [1] 322
```

```
322/12865*100
```

```
## [1] 2.502915
```

```
#calculate average leverage for comparison with 4 parameters
avg_leverage = (4+1)/12865
avg_leverage
```

```
## [1] 0.0003886514
```

```
#calculate limit(s) leverage should not exceed
leverage_limit= avg_leverage*2
leverage_limit
```

```
## [1] 0.0007773028
```

```
leverage_limit3 = avg_leverage*3
leverage_limit3
```

```
## [1] 0.001165954
```

```
#get count of samples over leverage limit
large_res%>%
  filter(leverage > leverage_limit)%>%
  nrow()
```

```
## [1] 111
```

```
large_res%>%
  filter(leverage > leverage_limit3)%>%
  nrow()
```

```
## [1] 87
```

After calculating the average leverage, I doubled it to create a boundary. 111 cases from the large residuals dataframe go passed this boundary. After tripling the average leverage and applying it to the large residuals dataframe, 87 cases are beyond it. Compared to other cases, these unusually large residuals are not appropriate to our model's outcome.

```
#Search for cook's distance values that exceed 1.0
large_res%>%
  filter(cooks.distance > 1)%>%
  nrow()
```

```
## [1] 0
```

```
#Calculate upper CVR boundary
upper_cvr = 1 + (3*(4+1)/12865)
upper_cvr
```

```
## [1] 1.001166
```

```
#Calculate lower CVR boundary
lower_cvr = 1 - (3*(4+1)/12865)
lower_cvr
```

```
## [1] 0.998834
```

```
#Check for values outside of CVR boundaries
large_res%>%
  filter(covariance.ratios > upper_cvr)%>%
  nrow()
```

```
## [1] 20
```

```
large_res%>%
  filter(covariance.ratios < lower_cvr)%>%
  nrow()
```

```
## [1] 250
```

After calculating cook's distance, I found no cases problematic since they are not equal to or larger than 1. The covariance ratios were calculated to be between 0.998 and 1.001.

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

```
dwt(housing_lm1)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.6309692     0.7380424       0
##  Alternative hypothesis: rho != 0
```

```
dwt(housing_lm2)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.7381742     0.5236481       0
##  Alternative hypothesis: rho != 0
```

The D-W Statistic for both "housing_lm1" and "housing_lm2" are values that are less than 1 and not even close to a value of 2. Since the values aren't close to 2 that means the assumption of independence for both models are not met.

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

```
#VIF
vif(housing_lm2)
```

```
##           building_grade square_feet_total_living                 bedrooms
##                 2.356699                 3.437335                 1.556365
##          bath_full_count                sq_ft_lot
##                 1.399634                 1.083811
```

```
#Tolerance
1/vif(housing_lm2)
```

```
##           building_grade square_feet_total_living                 bedrooms
##                0.4243223                0.2909231                0.6425229
##          bath_full_count                sq_ft_lot
##                0.7144727                0.9226703
```

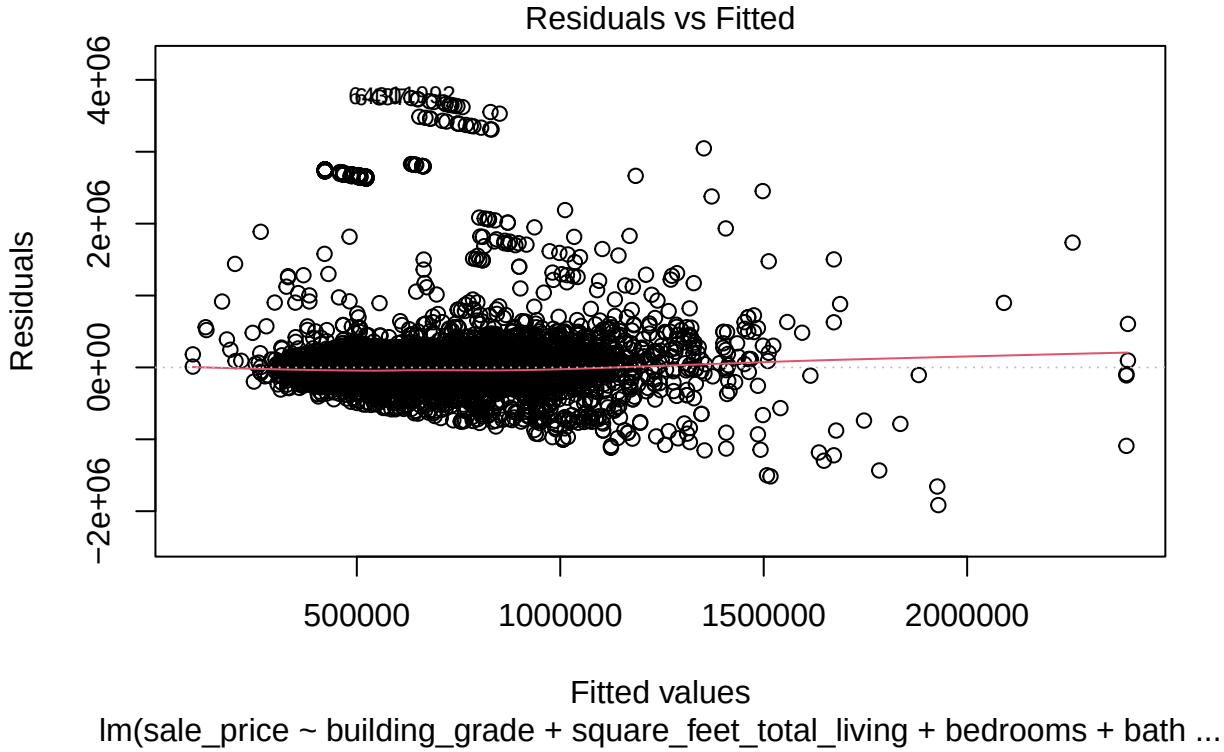
```
#Average VIF
mean(vif(housing_lm2))
```

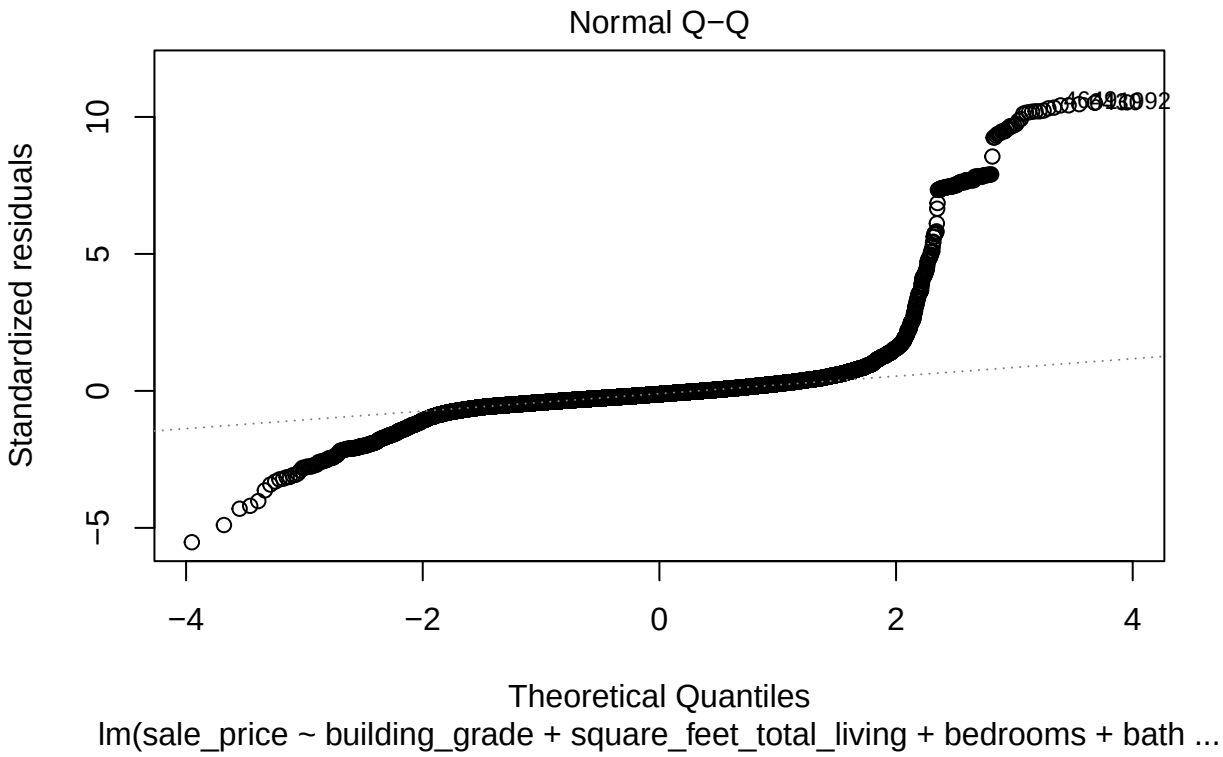
```
## [1] 1.966769
```

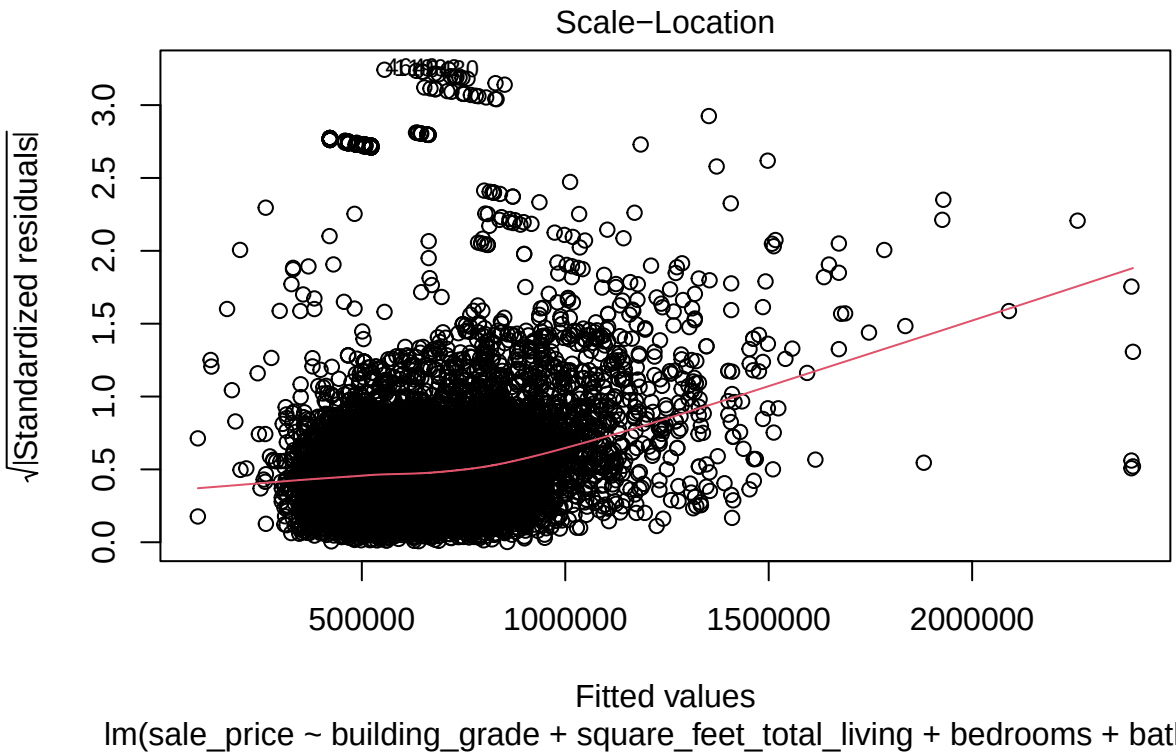
The VIF values are below 10. Also, the tolerance statistics are above 0.2. In addition the average VIF value is is not that much greater than 1 so it shouldn't be biased. Using these values we can conclude that there is no collinearity with the data.

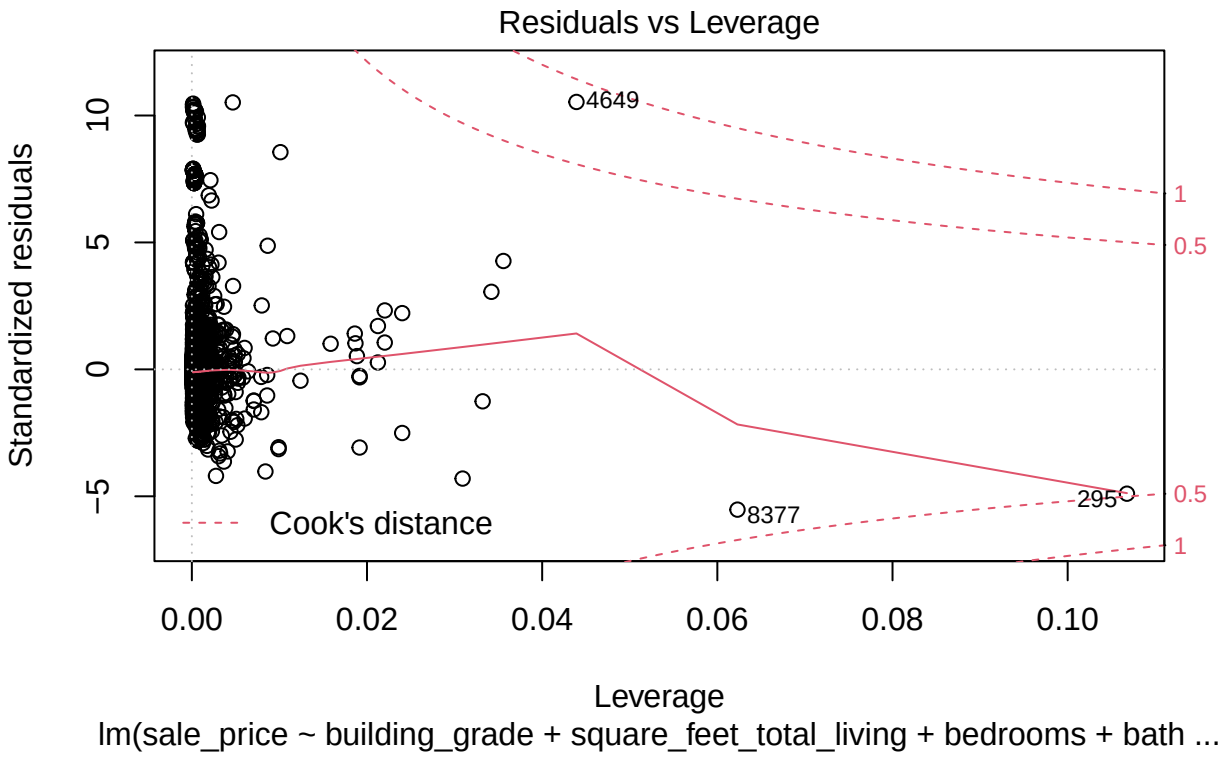
#xiv
Visually check the assumptions related to the residuals using the plot() and hist() functions. Summarize what each graph is informing you of and if any anomalies are present.

```
plot(housing_lm2)
```

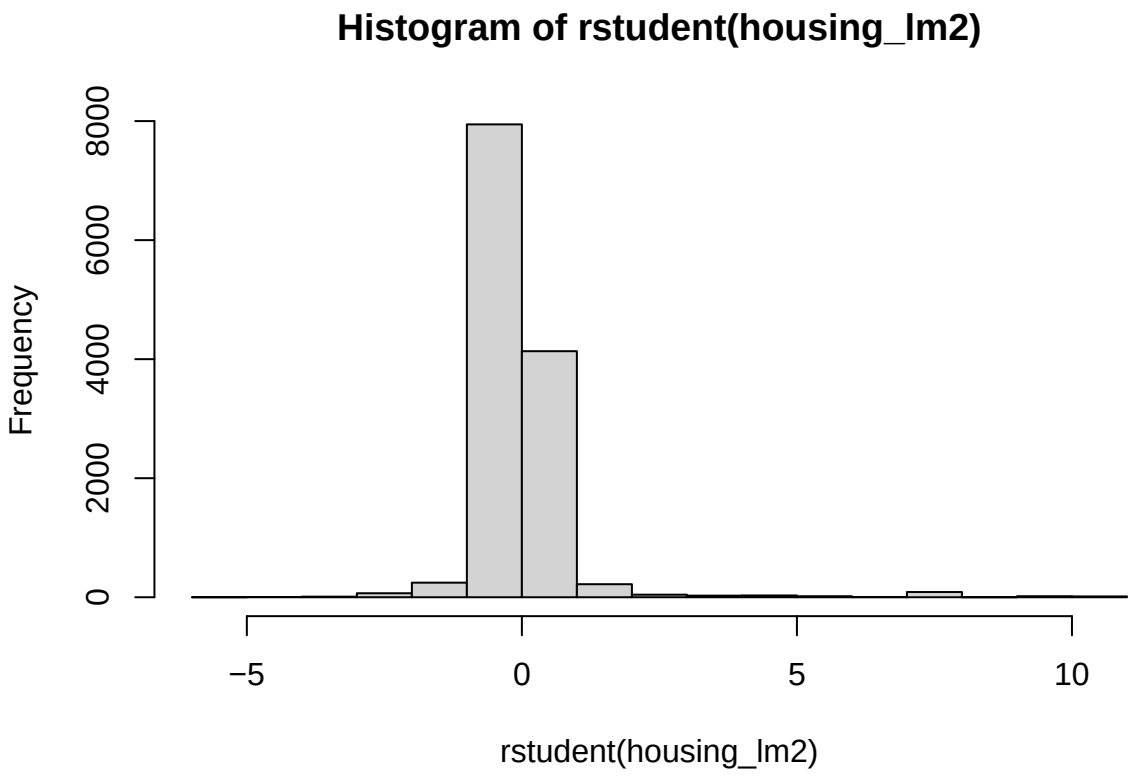



Normal Q−Q
Standardized residuals
10
5
0
−5
−4
−2
0
2
4
Theoretical Quantiles
lm(sale_price ~ building_grade + square_feet_total_living + bedrooms + bath ...


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
500000
1000000
1500000
2000000
Fitted values
lm(sale_price ~ building_grade + square_feet_total_living + bedrooms + bath ...




```
hist(rstudent(housing_lm2))
```



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

Based on the average VIF value, the model should be unbiased.