Assignment 8.2 Housing Dataset

Kausik Chattapadhyay

October 20 2022

Housing Data

Data for this assignment is focused on real estate transactions recorded from 1964 to 2016.

Loading the Housing Dataset

```
## 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 ...
##  $ 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 ...
```

Question A:

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

Answer for A

Here are the variables and data points I chose to remove:

- Removed rows whose sale price is > 2 million and square foot lot > 20000 as they are outliers and would skew the data
- Removed properties with sale warning and no bedrooms as those are mostly land and not a house.
- Removed columns Sale_date, sale_reason, sale_instrument, sale_warning, site_type as they are not relevant in predicting the prices.
- Removed Address, ctyname, postalcty, lon, lat as they are redundant info and can be indered from Zip5.
- Removed current_zoning, prop_type and present_use

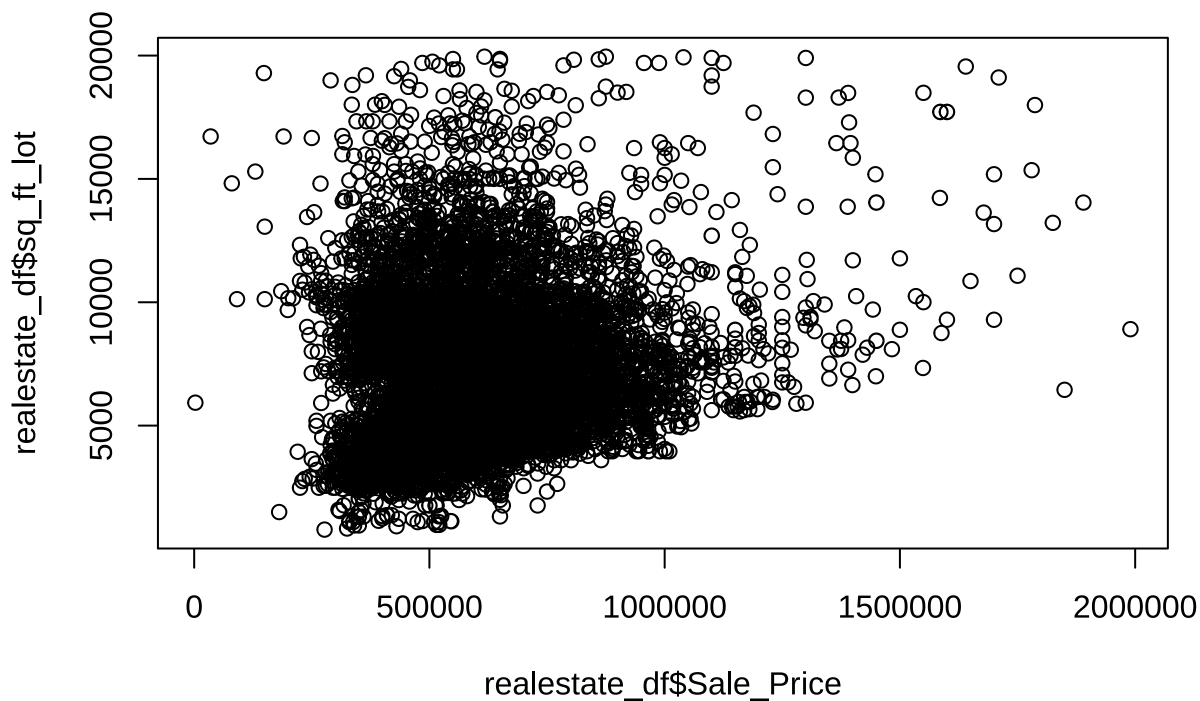
Code

```
## Set the working directory to the root of your DSC 520 directory
setwd("/Users/kausik/desktop/MS Data Science/DSC 520/dsc520-stats-r-assignments")

library(ggplot2)
library(plyr)
library(dplyr)
library(readxl)

realestate_df <- read_excel("data/week-7-housing.xlsx", sheet="Sheet2")

# Renamimg the field names
colnames(realestate_df)[2] <- "Sale_Price"
colnames(realestate_df)[1] <- "Sale_Date"
## Add a calculated column total_bath which provides no of bathroom in total
realestate_df <- within(realestate_df, total_bath <- bath_full_count + (bath_half_count/2) + (bath_3qtr

##Select relevant data points, sale price < 2000000 and square foot lot < 20000
realestate_df = realestate_df[realestate_df$Sale_Price < 2000000 & realestate_df$sq_ft_lot < 20000, ]
realestate_df <- realestate_df[(is.na(realestate_df$sale_warning)) & (realestate_df$bedrooms != 0), ]

##selecting only relevant columns for our calculation

realestate_df <- realestate_df[, c(2,8,13, 14,15,19,20, 22, 25)]

summary(realestate_df)
```

```
##    Sale_Price           zip5       building_grade   square_feet_total_living
##  Min.   :   2500   Min.   :98052   Min.   : 5.000   Min.   : 530
##  1st Qu.: 474800   1st Qu.:98052   1st Qu.: 8.000   1st Qu.:1800
##  Median : 584000   Median :98052   Median : 8.000   Median :2310
##  Mean   : 610864   Mean   :98052   Mean   : 8.116   Mean   :2396
##  3rd Qu.: 719950   3rd Qu.:98053   3rd Qu.: 9.000   3rd Qu.:2930
##  Max.   :1990000   Max.   :98074   Max.   :12.000   Max.   :7980
##     bedrooms        year_built    year_renovated   sq_ft_lot
##  Min.   : 1.000   Min.   :1900   Min.   :   0   Min.   :  785
##  1st Qu.: 3.000   1st Qu.:1979   1st Qu.:   0   1st Qu.: 4998
##  Median : 3.000   Median :2003   Median :   0   Median : 6973
##  Mean   : 3.439   Mean   :1995   Mean   :  17   Mean   : 7329
```

```
##  3rd Qu.: 4.000   3rd Qu.:2008   3rd Qu.:   0   3rd Qu.: 9055
##  Max.   :11.000   Max.   :2016   Max.   :2016   Max.   :19954
##    total_bath
##  Min.   :0.3333
##  1st Qu.:1.8333
##  Median :2.5000
##  Mean   :2.2363
##  3rd Qu.:2.5000
##  Max.   :6.6667
```

```
plot(realestate_df$Sale_Price,realestate_df$sq_ft_lot)
```



summary plot

Question 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.

Answer for B:

- After looking at the corelation between sale price and other variables, I noticed that the variables building_grade, square_feet_total_living, bedrooms, year_built and total_bath have a signification impact on the sale price of the property, so I chose used them as predictors.

- For total bathrooms i have calculated it as bath_full_count + (bath_half_count/3) + (bath_3qtr_count/3) to make it on equal as the customers usually look at the total baths and it is calculated this basis.

```
cor(realestate_df)
```

```
##                         Sale_Price        zip5 building_grade
## Sale_Price              1.00000000  0.04946348     0.64853955
## zip5                    0.04946348  1.00000000     0.07739962
## building_grade          0.64853955  0.07739962     1.00000000
## square_feet_total_living 0.73280440  0.06064458     0.66728632
## bedrooms                0.37791091 -0.07349727     0.29690360
## year_built              0.38819417  0.16130642     0.43988990
## year_renovated          0.05191527 -0.01782266    -0.01084515
## sq_ft_lot               0.11916511  0.02336914     0.06007563
## total_bath              0.52925631  0.07702720     0.50144470
##                         square_feet_total_living     bedrooms   year_built
## Sale_Price                            0.73280440  0.377910910  0.388194175
## zip5                                  0.06064458 -0.073497274  0.161306421
## building_grade                        0.66728632  0.296903602  0.439889897
## square_feet_total_living              1.00000000  0.628011451  0.420570192
## bedrooms                              0.62801145  1.000000000 -0.009455569
## year_built                            0.42057019 -0.009455569  1.000000000
## year_renovated                        0.03958108  0.024417942 -0.199569889
## sq_ft_lot                             0.11737705  0.217320060 -0.528780889
## total_bath                            0.67634670  0.392656869  0.533229220
##                         year_renovated   sq_ft_lot  total_bath
## Sale_Price                  0.05191527  0.11916511  0.52925631
## zip5                       -0.01782266  0.02336914  0.07702720
## building_grade             -0.01084515  0.06007563  0.50144470
## square_feet_total_living    0.03958108  0.11737705  0.67634670
## bedrooms                    0.02441794  0.21732006  0.39265687
## year_built                 -0.19956989 -0.52878089  0.53322922
## year_renovated              1.00000000  0.12678523  0.02289362
## sq_ft_lot                   0.12678523  1.00000000 -0.13015370
## total_bath                  0.02289362 -0.13015370  1.00000000
```

```
## Fit a linear model using the `Square foot of Lot` variable as the predictor and Sale Price` as the o
salepricebysqft_lm <-  lm(realestate_df$Sale_Price~realestate_df$sq_ft_lot,data = realestate_df)

## Fit a linear model using several predictors variable and `Sale Price` as the outcome
salepricebymultiplevar_lm <-  lm(realestate_df$Sale_Price~realestate_df$square_feet_total_living+reales
                                  ,data = realestate_df)
```

Question C:

** Choose the type of correlation test to perform, explain why you chose this test, and make a prediction if the test yields a positive or negative correlation? **

Answer For C

- Looking at the R2 statistics at the bottom of each summary. This value describes the overall model and tells us whether the model is successful in predicting the outcome and If the difference between R2 and adjusted R2 values is small this would indicate that the sample taken is a good representation of the population.
- looking at the first regression model, R2 is 0.0142 so this indicated that sq_ft_lot accounted for only 1.42% of the variation in sale price.
- Whereas in the multiple regression model, the value of R2 is 0.5874, so this multiple predictor model accounted for 54.98% of the variation in sale price.
- So the inclusion of the new predictors has explained quite a large amount of the variation in sale price, went up from 1.42% to 54.98%
- The adjusted R2 gives us an idea of how well our model generalizes. In our summary the difference for the final model the difference between the R2 and adjusted R2 values is (0.5874 minus 0.5872) = .0002 or 0.02%. This shrinkage means that if the model were derived from the population rather than a sample it would account for approximately 0.02% less variance in the outcome.

```r
## View the summary of your model using `summary()`
summary(salepricebysqft_lm)
```

```
##
## Call:
## lm(formula = realestate_df$Sale_Price ~ realestate_df$sq_ft_lot,
##     data = realestate_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -645897 -136979  -24938  106739 1367351
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             5.562e+05  5.335e+03  104.26   <2e-16 ***
## realestate_df$sq_ft_lot 7.457e+00  6.708e-01   11.12   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 191900 on 8579 degrees of freedom
## Multiple R-squared:  0.0142, Adjusted R-squared:  0.01409
## F-statistic: 123.6 on 1 and 8579 DF,  p-value: < 2.2e-16
```

```r
## View the summary of your new model using `summary()`
summary(salepricebymultiplevar_lm)
```

```
##
## Call:
## lm(formula = realestate_df$Sale_Price ~ realestate_df$square_feet_total_living +
##     realestate_df$year_built + realestate_df$bedrooms + realestate_df$total_bath +
##     realestate_df$building_grade, data = realestate_df)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -881746  -75243  -12843   58597 1292098
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       -4.686e+05  2.063e+05  -2.272  0.02314
## realestate_df$square_feet_total_living  1.428e+02  3.309e+00  43.173  < 2e-16
## realestate_df$year_built           1.471e+02  1.053e+02   1.397  0.16237
## realestate_df$bedrooms            -1.650e+04  2.150e+03  -7.674 1.85e-14
## realestate_df$total_bath           9.044e+03  3.389e+03   2.669  0.00762
## realestate_df$building_grade       5.919e+04  2.161e+03  27.394  < 2e-16
##
## (Intercept)                       *
## realestate_df$square_feet_total_living ***
## realestate_df$year_built
## realestate_df$bedrooms            ***
## realestate_df$total_bath          **
## realestate_df$building_grade      ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 124200 on 8575 degrees of freedom
## Multiple R-squared:  0.5874, Adjusted R-squared:  0.5872
## F-statistic:  2442 on 5 and 8575 DF,  p-value: < 2.2e-16
```

Question 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?

Answer to D

```r
library('QuantPsyc')
```

```
## Loading required package: boot

## Warning: package 'boot' was built under R version 4.0.2

## Loading required package: purrr

## Warning: package 'purrr' was built under R version 4.0.2

##
## Attaching package: 'purrr'

## The following object is masked from 'package:plyr':
##
##     compact

## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

##
## Attaching package: 'QuantPsyc'

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

```
##standardized betas for each parameter
lm.beta(salepricebymultiplevar_lm)
```

```
## realestate_df$square_feet_total_living              realestate_df$year_built
##                             0.57954215                            0.01267996
##                  realestate_df$bedrooms              realestate_df$total_bath
##                            -0.07528203                            0.02723217
##            realestate_df$building_grade
##                             0.26493730
```

- As we know, the standardized beta 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.
- Looking at the outcome, we can figure out that square_feet_total_living and building_grade have more degree of importance in prediction, whereas bedrooms, year_built and total_bath have a comparably less degree of importance.

Question E:

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

Answer for E

```
confint(salepricebymultiplevar_lm)
```

```
##                                                2.5 %      97.5 %
## (Intercept)                             -872966.50303 -64223.3607
## realestate_df$square_feet_total_living     136.36233    149.3343
## realestate_df$year_built                   -59.25909    353.4163
## realestate_df$bedrooms                  -20717.95186 -12287.1934
## realestate_df$total_bath                  2402.06681  15686.7220
## realestate_df$building_grade             54953.31011  63423.9444
```

Lets look at the output generated from the confidence interval:

- square__feet__total__living 136.36 - 149.3343, this has very tight confidence interval, indicating that the estimates for the current model are likely to be representative of the true population values.
- building__grade 54953.31011 63423.9444, this is a good predictor but has more gap
- bedrooms -20717.95186 -12287.1934, this is a good predictor but has more gap
- total__bath 2402.06681 15686.7220, this is a good predictor but has more gap
- year__built -59.25909 353.4163, Confidence intervals that cross zero, indicating that in some samples the predictor has a negative relationship to the outcome whereas in others it has a positive relationship

Question 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.

Answer for F:

```
anova(salepricebysqft_lm, salepricebymultiplevar_lm)
```

```
## Analysis of Variance Table
##
## Model 1: realestate_df$Sale_Price ~ realestate_df$sq_ft_lot
## Model 2: realestate_df$Sale_Price ~ realestate_df$square_feet_total_living +
##     realestate_df$year_built + realestate_df$bedrooms + realestate_df$total_bath +
##     realestate_df$building_grade
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
## 1   8579 3.1584e+14
## 2   8575 1.3219e+14  4 1.8365e+14 2978.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

for salepricebymultiplevar__lm the variance table analysis shows: $F(4, 8575) = 2978.2$ with $p < 0.001$ hence we can conclude that the multiple regression model significantly improved the fit of the model to the data compared to salepricebysqft__lm.

Question 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.

Answer for G

Outliers: Residuals can be obtained with the resid() function, standardized residuals with the rstandard() function and studentized residuals with the rstudent() function.

Influential cases: Cook's distances can be obtained with the cooks.distance() function, DFBeta with the dfbeta() function, DFFit with the dffits() function, hat values (leverage) with the hatvalues() function, and the covariance ratio with the covratio() function.

Below is the detailed diagnostics of outliers and influential cases

```
## outliers
realestate_df$residuals <- resid(salepricebymultiplevar_lm)
realestate_df$studentized.residuals <- rstudent(salepricebymultiplevar_lm)
realestate_df$standardized.residuals <- rstandard(salepricebymultiplevar_lm)

## Influential cases

realestate_df$dffit <- dffits(salepricebymultiplevar_lm)
realestate_df$leverage <- hatvalues(salepricebymultiplevar_lm)
realestate_df$covariance.ratios <- covratio(salepricebymultiplevar_lm)
realestate_df$cooks.distance <- cooks.distance(salepricebymultiplevar_lm)
realestate_df$dfbeta <- dfbeta(salepricebymultiplevar_lm)

summary(realestate_df)
```

```
##    Sale_Price          zip5       building_grade   square_feet_total_living
##  Min.   :   2500   Min.   :98052   Min.   : 5.000   Min.   : 530
##  1st Qu.: 474800   1st Qu.:98052   1st Qu.: 8.000   1st Qu.:1800
##  Median : 584000   Median :98052   Median : 8.000   Median :2310
##  Mean   : 610864   Mean   :98052   Mean   : 8.116   Mean   :2396
##  3rd Qu.: 719950   3rd Qu.:98053   3rd Qu.: 9.000   3rd Qu.:2930
##  Max.   :1990000   Max.   :98074   Max.   :12.000   Max.   :7980
##     bedrooms        year_built   year_renovated   sq_ft_lot
##  Min.   : 1.000   Min.   :1900   Min.   :   0   Min.   :  785
##  1st Qu.: 3.000   1st Qu.:1979   1st Qu.:   0   1st Qu.: 4998
##  Median : 3.000   Median :2003   Median :   0   Median : 6973
##  Mean   : 3.439   Mean   :1995   Mean   :  17   Mean   : 7329
##  3rd Qu.: 4.000   3rd Qu.:2008   3rd Qu.:   0   3rd Qu.: 9055
##  Max.   :11.000   Max.   :2016   Max.   :2016   Max.   :19954
##    total_bath       residuals       studentized.residuals
##  Min.   :0.3333   Min.   :-881746   Min.   :-7.129288
##  1st Qu.:1.8333   1st Qu.: -75243   1st Qu.:-0.606191
##  Median :2.5000   Median : -12843   Median :-0.103465
##  Mean   :2.2363   Mean   :      0   Mean   : 0.000084
##  3rd Qu.:2.5000   3rd Qu.:  58598   3rd Qu.: 0.472080
##  Max.   :6.6667   Max.   :1292098   Max.   :10.478545
##  standardized.residuals     dffit               leverage
##  Min.   :-7.108665      Min.   :-0.6485180   Min.   :0.0001761
##  1st Qu.:-0.606213      1st Qu.:-0.0140526   1st Qu.:0.0004280
##  Median :-0.103471      Median :-0.0025020   Median :0.0006049
##  Mean   : 0.000006      Mean   : 0.0002524   Mean   :0.0006992
##  3rd Qu.: 0.472101      3rd Qu.: 0.0112635   3rd Qu.:0.0008253
##  Max.   :10.412695      Max.   : 0.5580787   Max.   :0.0121037
##  covariance.ratios cooks.distance
##  Min.   :0.9282    Min.   :0.000e+00
##  1st Qu.:1.0007    1st Qu.:6.310e-06
##  Median :1.0010    Median :2.845e-05
##  Mean   :1.0007    Mean   :1.883e-04
##  3rd Qu.:1.0013    3rd Qu.:8.825e-05
##  Max.   :1.0096    Max.   :6.971e-02
##   dfbeta.(Intercept)   dfbeta.realestate_df$square_feet_total_living  dfbeta.realestate_df$year_buil
##  Min.   :-63277.76     Min.   :-0.9072594    Min.   :-48.61350     Min.   :-621.2461     Min.   :-16
```

```
##  1st Qu.:  -692.11     1st Qu.:-0.0098428    1st Qu.: -0.34156     1st Qu.:  -6.2076     1st Qu.:  -
##  Median :     6.47     Median :-0.0000643    Median : -0.00241     Median :   0.3097     Median :
##  Mean   :     0.08     Mean   :-0.0000013    Mean   : -0.00005     Mean   :  -0.0003     Mean   :
##  3rd Qu.:   686.12     3rd Qu.: 0.0104603    3rd Qu.:  0.36343     3rd Qu.:   7.5582     3rd Qu.:
##  Max.   : 98912.34     Max.   : 0.9312512    Max.   : 34.30849     Max.   : 292.5238     Max.   : 10
```

Question 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.

Answer for G

> Outliers: Residuals can be obtained with the resid() function, standardized residuals with the rstandard() function and studentized residuals with the rstudent() function.
>
> Influential cases: Cook's distances can be obtained with the cooks.distance() function, DFBeta with the dfbeta() function, DFFit with the dffits() function, hat values (leverage) with the hatvalues() function, and the covariance ratio with the covratio() function.
>
> Below is the detailed diagnostics of outliers and influential cases

```
## outliers
realestate_df$residuals <- resid(salepricebymultiplevar_lm)
realestate_df$studentized.residuals <- rstudent(salepricebymultiplevar_lm)
realestate_df$standardized.residuals <- rstandard(salepricebymultiplevar_lm)

## Influential cases

realestate_df$dffit <- dffits(salepricebymultiplevar_lm)
realestate_df$leverage <- hatvalues(salepricebymultiplevar_lm)
realestate_df$covariance.ratios <- covratio(salepricebymultiplevar_lm)
realestate_df$cooks.distance <- cooks.distance(salepricebymultiplevar_lm)
realestate_df$dfbeta <- dfbeta(salepricebymultiplevar_lm)

summary(realestate_df)
```

```
##    Sale_Price          zip5       building_grade   square_feet_total_living
##  Min.   :   2500   Min.   :98052   Min.   : 5.000   Min.   : 530
##  1st Qu.: 474800   1st Qu.:98052   1st Qu.: 8.000   1st Qu.:1800
##  Median : 584000   Median :98052   Median : 8.000   Median :2310
##  Mean   : 610864   Mean   :98052   Mean   : 8.116   Mean   :2396
##  3rd Qu.: 719950   3rd Qu.:98053   3rd Qu.: 9.000   3rd Qu.:2930
##  Max.   :1990000   Max.   :98074   Max.   :12.000   Max.   :7980
##     bedrooms        year_built    year_renovated   sq_ft_lot
##  Min.   : 1.000   Min.   :1900   Min.   :   0   Min.   :  785
##  1st Qu.: 3.000   1st Qu.:1979   1st Qu.:   0   1st Qu.: 4998
##  Median : 3.000   Median :2003   Median :   0   Median : 6973
##  Mean   : 3.439   Mean   :1995   Mean   :  17   Mean   : 7329
##  3rd Qu.: 4.000   3rd Qu.:2008   3rd Qu.:   0   3rd Qu.: 9055
##  Max.   :11.000   Max.   :2016   Max.   :2016   Max.   :19954
```

```
##    total_bath       residuals       studentized.residuals
##  Min.   :0.3333   Min.   :-881746   Min.   :-7.129288
##  1st Qu.:1.8333   1st Qu.: -75243   1st Qu.:-0.606191
##  Median :2.5000   Median : -12843   Median :-0.103465
##  Mean   :2.2363   Mean   :      0   Mean   : 0.000084
##  3rd Qu.:2.5000   3rd Qu.:  58598   3rd Qu.: 0.472080
##  Max.   :6.6667   Max.   :1292098   Max.   :10.478545
##  standardized.residuals     dffit              leverage
##  Min.   :-7.108665      Min.   :-0.6485180   Min.   :0.0001761
##  1st Qu.:-0.606213      1st Qu.:-0.0140526   1st Qu.:0.0004280
##  Median :-0.103471      Median :-0.0025020   Median :0.0006049
##  Mean   : 0.000006      Mean   : 0.0002524   Mean   :0.0006992
##  3rd Qu.: 0.472101      3rd Qu.: 0.0112635   3rd Qu.:0.0008253
##  Max.   :10.412695      Max.   : 0.5580787   Max.   :0.0121037
##  covariance.ratios cooks.distance
##  Min.   :0.9282    Min.   :0.000e+00
##  1st Qu.:1.0007    1st Qu.:6.310e-06
##  Median :1.0010    Median :2.845e-05
##  Mean   :1.0007    Mean   :1.883e-04
##  3rd Qu.:1.0013    3rd Qu.:8.825e-05
##  Max.   :1.0096    Max.   :6.971e-02
##   dfbeta.(Intercept)   dfbeta.realestate_df$square_feet_total_living  dfbeta.realestate_df$year_buil
##  Min.   :-63277.76     Min.   :-0.9072594    Min.   :-48.61350     Min.   :-621.2461     Min.   :-16
##  1st Qu.:  -692.11     1st Qu.:-0.0098428    1st Qu.: -0.34156     1st Qu.:  -6.2076     1st Qu.:  -
##  Median :     6.47     Median :-0.0000643    Median : -0.00241     Median :   0.3097     Median :
##  Mean   :     0.08     Mean   :-0.0000013    Mean   : -0.00005     Mean   :  -0.0003     Mean   :
##  3rd Qu.:   686.12     3rd Qu.: 0.0104603    3rd Qu.:  0.36343     3rd Qu.:   7.5582     3rd Qu.:
##  Max.   : 98912.34     Max.   : 0.9312512    Max.   : 34.30849     Max.   : 292.5238     Max.   : 10
```

Question 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.

Answer for G

> Outliers: Residuals can be obtained with the resid() function, standardized residuals with the rstandard() function and studentized residuals with the rstudent() function.
>
> Influential cases: Cook's distances can be obtained with the cooks.distance() function, DFBeta with the dfbeta() function, DFFit with the dffits() function, hat values (leverage) with the hatvalues() function, and the covariance ratio with the covratio() function.
>
> Below is the detailed diagnostics of outliers and influential cases

```r
## outliers
realestate_df$residuals <- resid(salepricebymultiplevar_lm)
realestate_df$studentized.residuals <- rstudent(salepricebymultiplevar_lm)
realestate_df$standardized.residuals <- rstandard(salepricebymultiplevar_lm)

## Influential cases
```

```r
realestate_df$dffit <- dffits(salepricebymultiplevar_lm)
realestate_df$leverage <- hatvalues(salepricebymultiplevar_lm)
realestate_df$covariance.ratios <- covratio(salepricebymultiplevar_lm)
realestate_df$cooks.distance <- cooks.distance(salepricebymultiplevar_lm)
realestate_df$dfbeta <- dfbeta(salepricebymultiplevar_lm)

summary(realestate_df)
```

```
##    Sale_Price          zip5       building_grade   square_feet_total_living
##  Min.   :   2500   Min.   :98052   Min.   : 5.000   Min.   : 530
##  1st Qu.: 474800   1st Qu.:98052   1st Qu.: 8.000   1st Qu.:1800
##  Median : 584000   Median :98052   Median : 8.000   Median :2310
##  Mean   : 610864   Mean   :98052   Mean   : 8.116   Mean   :2396
##  3rd Qu.: 719950   3rd Qu.:98053   3rd Qu.: 9.000   3rd Qu.:2930
##  Max.   :1990000   Max.   :98074   Max.   :12.000   Max.   :7980
##     bedrooms        year_built   year_renovated   sq_ft_lot
##  Min.   : 1.000   Min.   :1900   Min.   :   0   Min.   :  785
##  1st Qu.: 3.000   1st Qu.:1979   1st Qu.:   0   1st Qu.: 4998
##  Median : 3.000   Median :2003   Median :   0   Median : 6973
##  Mean   : 3.439   Mean   :1995   Mean   :  17   Mean   : 7329
##  3rd Qu.: 4.000   3rd Qu.:2008   3rd Qu.:   0   3rd Qu.: 9055
##  Max.   :11.000   Max.   :2016   Max.   :2016   Max.   :19954
##    total_bath       residuals       studentized.residuals
##  Min.   :0.3333   Min.   :-881746   Min.   :-7.129288
##  1st Qu.:1.8333   1st Qu.: -75243   1st Qu.:-0.606191
##  Median :2.5000   Median : -12843   Median :-0.103465
##  Mean   :2.2363   Mean   :      0   Mean   : 0.000084
##  3rd Qu.:2.5000   3rd Qu.:  58598   3rd Qu.: 0.472080
##  Max.   :6.6667   Max.   :1292098   Max.   :10.478545
##  standardized.residuals     dffit                leverage
##  Min.   :-7.108665      Min.   :-0.6485180   Min.   :0.0001761
##  1st Qu.:-0.606213      1st Qu.:-0.0140526   1st Qu.:0.0004280
##  Median :-0.103471      Median :-0.0025020   Median :0.0006049
##  Mean   : 0.000006      Mean   : 0.0002524   Mean   :0.0006992
##  3rd Qu.: 0.472101      3rd Qu.: 0.0112635   3rd Qu.:0.0008253
##  Max.   :10.412695      Max.   : 0.5580787   Max.   :0.0121037
##  covariance.ratios cooks.distance
##  Min.   :0.9282    Min.   :0.000e+00
##  1st Qu.:1.0007    1st Qu.:6.310e-06
##  Median :1.0010    Median :2.845e-05
##  Mean   :1.0007    Mean   :1.883e-04
##  3rd Qu.:1.0013    3rd Qu.:8.825e-05
##  Max.   :1.0096    Max.   :6.971e-02
##   dfbeta.(Intercept)   dfbeta.realestate_df$square_feet_total_living  dfbeta.realestate_df$year_buil
##  Min.   :-63277.76     Min.   :-0.9072594     Min.   :-48.61350     Min.   :-621.2461     Min.   :-16
##  1st Qu.:  -692.11     1st Qu.:-0.0098428     1st Qu.: -0.34156     1st Qu.:  -6.2076     1st Qu.:  -
##  Median :     6.47     Median :-0.0000643     Median : -0.00241     Median :   0.3097     Median :
##  Mean   :     0.08     Mean   :-0.0000013     Mean   : -0.00005     Mean   :  -0.0003     Mean   :
##  3rd Qu.:   686.12     3rd Qu.: 0.0104603     3rd Qu.:  0.36343     3rd Qu.:   7.5582     3rd Qu.:
##  Max.   : 98912.34     Max.   : 0.9312512     Max.   : 34.30849     Max.   : 292.5238     Max.   : 10
```

Question 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.

Answer for H

```
realestate_df$large.residual <- realestate_df$standardized.residuals > 2 | realestate_df$standardized.r
summary(realestate_df)
```

```
##    Sale_Price           zip5       building_grade   square_feet_total_living
##  Min.   :   2500   Min.   :98052   Min.   : 5.000   Min.   : 530
##  1st Qu.: 474800   1st Qu.:98052   1st Qu.: 8.000   1st Qu.:1800
##  Median : 584000   Median :98052   Median : 8.000   Median :2310
##  Mean   : 610864   Mean   :98052   Mean   : 8.116   Mean   :2396
##  3rd Qu.: 719950   3rd Qu.:98053   3rd Qu.: 9.000   3rd Qu.:2930
##  Max.   :1990000   Max.   :98074   Max.   :12.000   Max.   :7980
##     bedrooms        year_built    year_renovated   sq_ft_lot
##  Min.   : 1.000   Min.   :1900   Min.   :   0   Min.   :  785
##  1st Qu.: 3.000   1st Qu.:1979   1st Qu.:   0   1st Qu.: 4998
##  Median : 3.000   Median :2003   Median :   0   Median : 6973
##  Mean   : 3.439   Mean   :1995   Mean   :  17   Mean   : 7329
##  3rd Qu.: 4.000   3rd Qu.:2008   3rd Qu.:   0   3rd Qu.: 9055
##  Max.   :11.000   Max.   :2016   Max.   :2016   Max.   :19954
##    total_bath       residuals       studentized.residuals
##  Min.   :0.3333   Min.   :-881746   Min.   :-7.129288
##  1st Qu.:1.8333   1st Qu.: -75243   1st Qu.:-0.606191
##  Median :2.5000   Median : -12843   Median :-0.103465
##  Mean   :2.2363   Mean   :      0   Mean   : 0.000084
##  3rd Qu.:2.5000   3rd Qu.:  58598   3rd Qu.: 0.472080
##  Max.   :6.6667   Max.   :1292098   Max.   :10.478545
##  standardized.residuals     dffit               leverage
##  Min.   :-7.108665      Min.   :-0.6485180   Min.   :0.0001761
##  1st Qu.:-0.606213      1st Qu.:-0.0140526   1st Qu.:0.0004280
##  Median :-0.103471      Median :-0.0025020   Median :0.0006049
##  Mean   : 0.000006      Mean   : 0.0002524   Mean   :0.0006992
##  3rd Qu.: 0.472101      3rd Qu.: 0.0112635   3rd Qu.:0.0008253
##  Max.   :10.412695      Max.   : 0.5580787   Max.   :0.0121037
##  covariance.ratios cooks.distance
##  Min.   :0.9282    Min.   :0.000e+00
##  1st Qu.:1.0007    1st Qu.:6.310e-06
##  Median :1.0010    Median :2.845e-05
##  Mean   :1.0007    Mean   :1.883e-04
##  3rd Qu.:1.0013    3rd Qu.:8.825e-05
##  Max.   :1.0096    Max.   :6.971e-02
##   dfbeta.(Intercept)   dfbeta.realestate_df$square_feet_total_living  dfbeta.realestate_df$year_buil
##  Min.   :-63277.76     Min.   :-0.9072594    Min.   :-48.61350     Min.   :-621.2461     Min.   :-16
##  1st Qu.:  -692.11     1st Qu.:-0.0098428    1st Qu.: -0.34156     1st Qu.:  -6.2076     1st Qu.:  -
##  Median :     6.47     Median :-0.0000643    Median : -0.00241     Median :   0.3097     Median :
##  Mean   :     0.08     Mean   :-0.0000013    Mean   : -0.00005     Mean   :  -0.0003     Mean   :
##  3rd Qu.:   686.12     3rd Qu.: 0.0104603    3rd Qu.:  0.36343     3rd Qu.:   7.5582     3rd Qu.:
##  Max.   : 98912.34     Max.   : 0.9312512    Max.   : 34.30849     Max.   : 292.5238     Max.   : 10
```

```
##   large.residual
##   Mode :logical
##   FALSE:8297
##   TRUE :284
##
##
##
```

Question I:

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

Answer for I

```r
sum(realestate_df$large.residual)
```

```
## [1] 284
```

Question J:

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

Answer for J

```r
realestate_df[realestate_df$large.residual, c("Sale_Price", "building_grade", "square_feet_total_living
```

```
## # A tibble: 284 x 8
##     Sale_Price building_grade square_fe~1 bedro~2 total~3 year_~4 sq_ft~5 stand~6
##          <dbl>          <dbl>       <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##   1    1392000              9        3740       4    4.33    1998   17291    4.25
##   2    1053649              9        2680       2    2.5     2005    8517    2.60
##   3    1080135              9        2700       3    2.33    2006    7694    2.93
##   4     732500              9        5710       5    4.33    1977   10200   -3.19
##   5     370000              9        4000       4    3.5     2014   11780   -4.25
##   6    1588359              9        3360       2    2.5     2005    8752    6.12
##   7    1450000              8        3480       3    2.5     1972   14043    5.52
##   8    1450000              6         900       2    1       1918   14043    9.49
##   9    1369900             11        4630       5    2.67    2005   18297    2.33
##  10    1174477              9        2800       3    2.5     2006   11071    3.56
## # ... with 274 more rows, and abbreviated variable names
## #   1: square_feet_total_living, 2: bedrooms, 3: total_bath, 4: year_built,
## #   5: sq_ft_lot, 6: standardized.residuals
```

Question K:

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

Answer for K

```
realestate_df[realestate_df$large.residual, c("cooks.distance", "leverage", "covariance.ratios")]
```

```
## # A tibble: 284 x 3
##    cooks.distance leverage covariance.ratios
##             <dbl>    <dbl>             <dbl>
##  1       0.00717  0.00238              0.990
##  2       0.000993 0.000883             0.997
##  3       0.000478 0.000334             0.995
##  4       0.00926  0.00544              0.999
##  5       0.00266  0.000883             0.989
##  6       0.00990  0.00158              0.976
##  7       0.0103   0.00203              0.982
##  8       0.0514   0.00341              0.942
##  9       0.00173  0.00190              0.999
## 10       0.000710 0.000335             0.992
## # ... with 274 more rows
```

> the above generated 284 rows but there is no row where cooks distance is greater than 1, so there are no problematic rows

Question L:

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

Answer for L

```
library("car")
```

```
## Loading required package: carData

## Warning: package 'carData' was built under R version 4.0.5

##
## Attaching package: 'car'

## The following object is masked from 'package:purrr':
##
##     some

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

## The following object is masked from 'package:dplyr':
##
##     recode
```

```r
dwt(salepricebymultiplevar_lm)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.4054537      1.189018       0
##  Alternative hypothesis: rho != 0
```

> We can test the assumption of independent errors using the Durbin–Watson test. We can obtain this statistic along with a measure of autocorrelation and a p-value in R using the durbinWatsonTest().The statistic should be between 1 and 3 and should be closer to 2, in our case, it is 1.18. The p-value of 0 confirms this conclusion.

Question M:

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

Answer for M

```r
## vif
vif(salepricebymultiplevar_lm)
```

```
## realestate_df$square_feet_total_living                 realestate_df$year_built
##                               3.745030                                 1.711507
##                 realestate_df$bedrooms                 realestate_df$total_bath
##                               2.000033                                 2.163369
##           realestate_df$building_grade
##                               1.943865
```

```r
## 1/vif
1/vif(salepricebymultiplevar_lm)
```

```
## realestate_df$square_feet_total_living                 realestate_df$year_built
##                              0.2670206                                0.5842806
##                 realestate_df$bedrooms                 realestate_df$total_bath
##                              0.4999917                                0.4622420
##           realestate_df$building_grade
##                              0.5144390
```

```r
## mean
mean(vif(salepricebymultiplevar_lm))
```

```
## [1] 2.312761
```

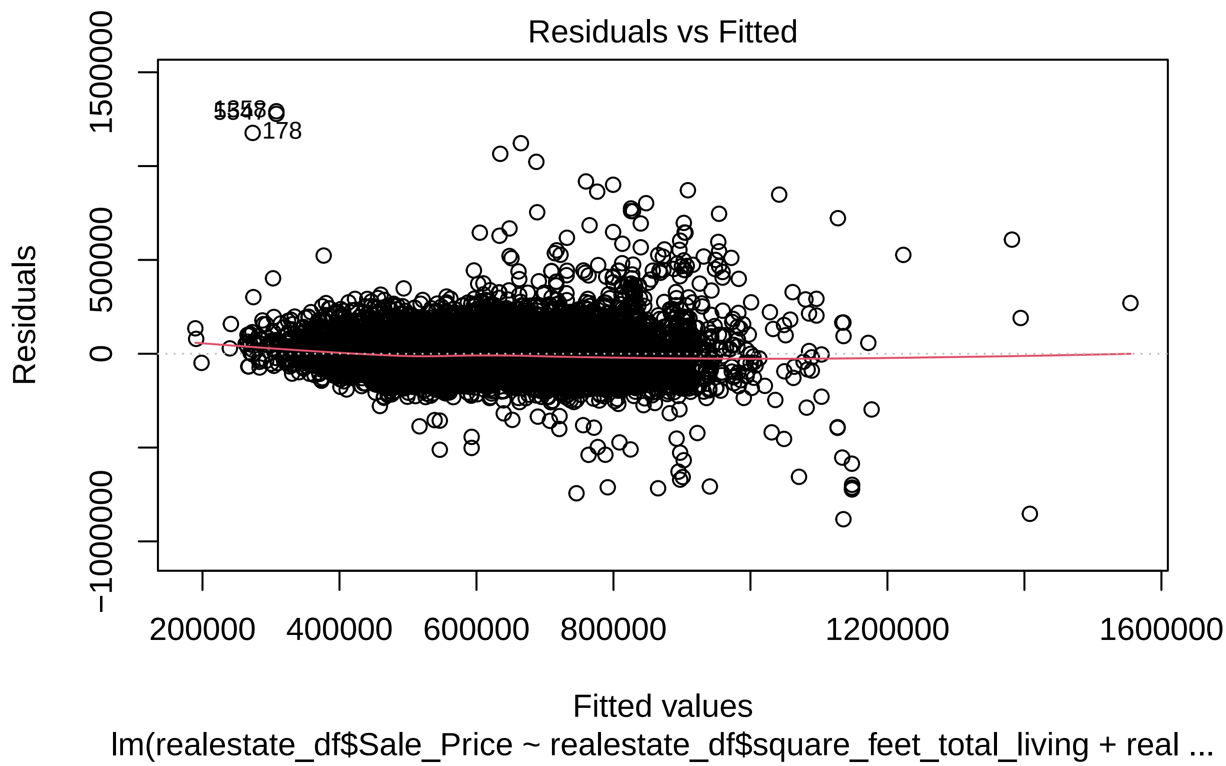
> For our current model the VIF values are all well below 10 and the tolerance statistics all well above 0.2. Also, the average VIF is very close to 1. Based on these measures we can safely conclude that there is no collinearity within our data.

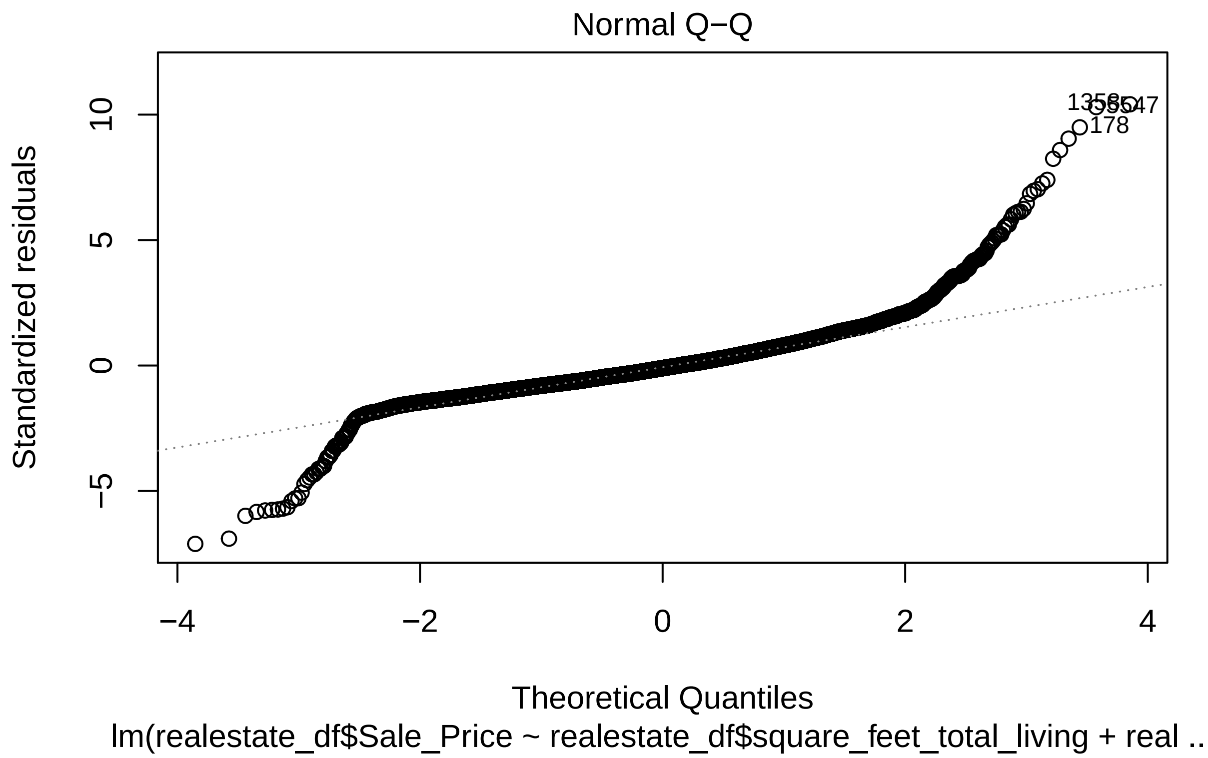
Question N:

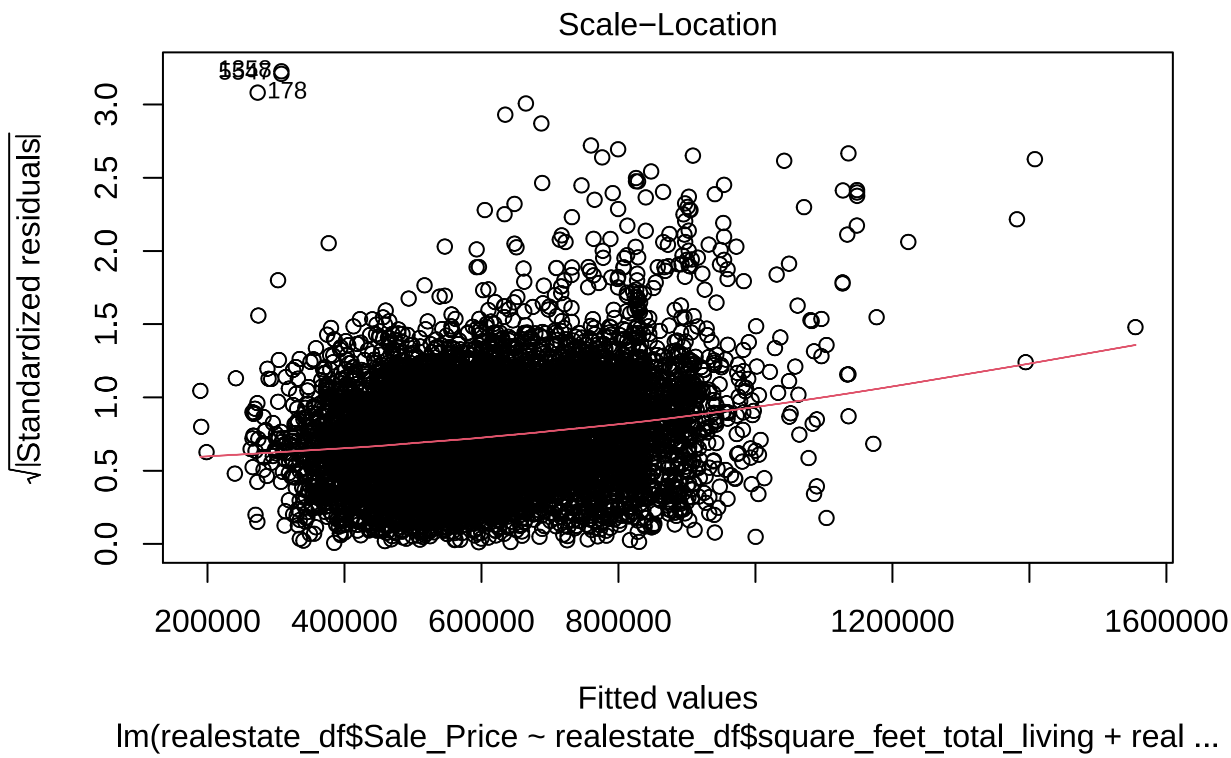
Visually check the assumptions related to the residuals using the plot() and hist() functions. Summarize what each graph is informing you of and if any anomalies are present.

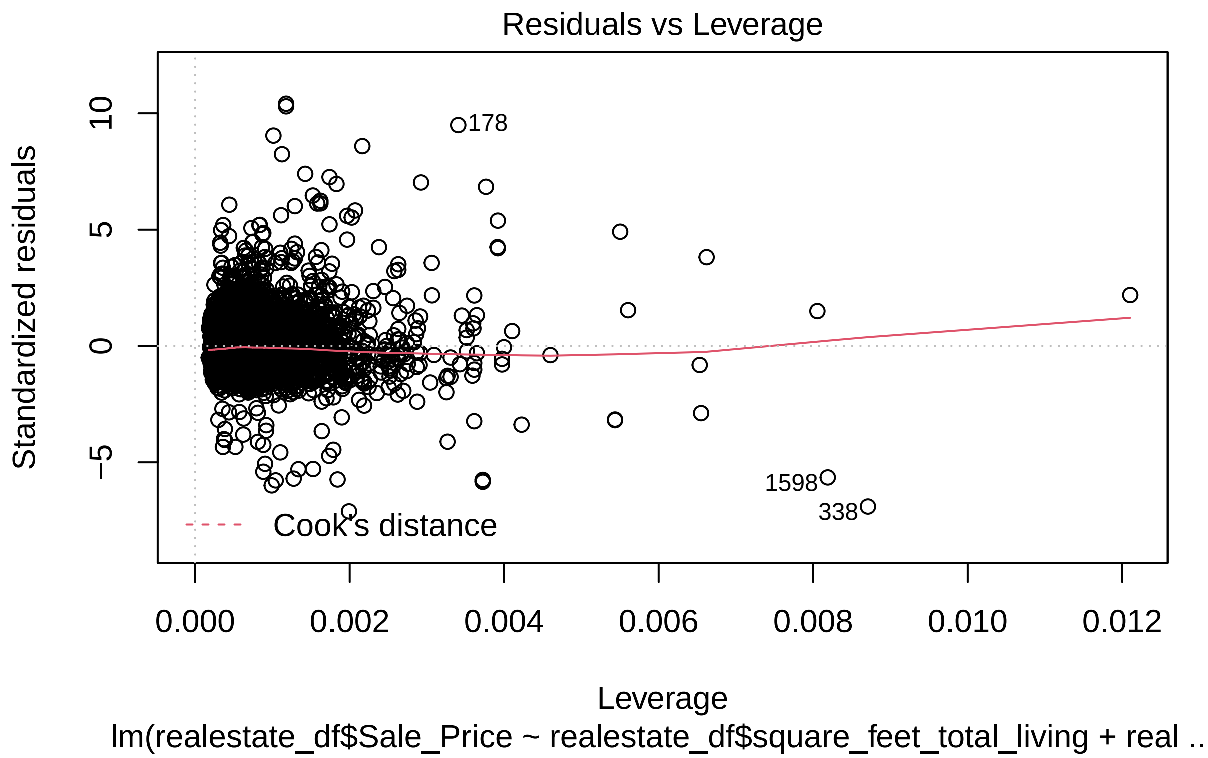
Answer for N

```
library(ggplot2)
plot(salepricebymultiplevar_lm)
```

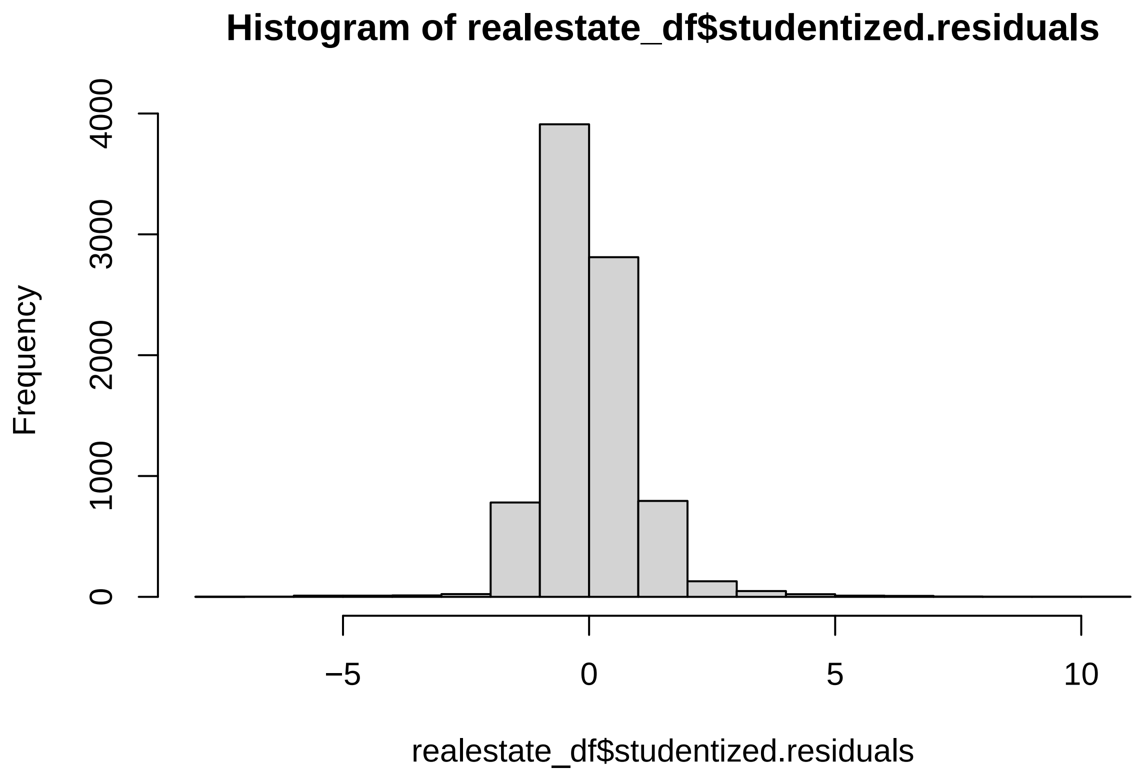



Normal Q−Q
Standardized residuals
10
5
0
−5
1358
5547
178
−4
−2
0
2
4
Theoretical Quantiles
lm(realestate_df$Sale_Price ~ realestate_df$square_feet_total_living + real ...


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
178
200000
400000
600000
800000
1200000
1600000
Fitted values
lm(realestate_df$Sale_Price ~ realestate_df$square_feet_total_living + real ...




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




```
scatter <- ggplot(realestate_df, aes(fitted, studentized.residuals)) + geom_point() + geom_smooth(metho
```

The first graph shows the plot of fitted values against residuals. looking like a random array of dots evenly dispersed around zero. the graph is not funneling out, so there are no chances that there is heteroscedasticity in the data. There is no curve in the graph, so it is not violating any assumptions of linearity.

The Normal Q-Q plot should show deviations from normality. In the plot above, it deviates from both the ends of the line, which indicates deviation of normality at the extreme values.

Question 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?

Answer for O

Looking at all the ouputs and calculations performed on the data model after removing the outliers, we can safely conclude that the regression model is unbiased. The sample is a good representation of the entire population model.