RegressionR

Logan Quandt

2022-07-22

```
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(purrr)
```

```
##
## Attaching package: 'purrr'
```

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

```
library(readxl)
library(stringr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v readr   2.1.1
## v tibble  3.1.6     v forcats 0.5.1
## v tidyr   1.1.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::arrange()   masks plyr::arrange()
## x purrr::compact()   masks plyr::compact()
## x dplyr::count()     masks plyr::count()
## x dplyr::failwith()  masks plyr::failwith()
## x dplyr::filter()    masks stats::filter()
## x dplyr::id()        masks plyr::id()
## x dplyr::lag()       masks stats::lag()
## x dplyr::mutate()    masks plyr::mutate()
```

```
## x dplyr::rename()    masks plyr::rename()
## x dplyr::summarise() masks plyr::summarise()
## x dplyr::summarize() masks plyr::summarize()
```

```
library(regclass)
```

```
## Warning: package 'regclass' was built under R version 4.1.3
## Loading required package: bestglm
## Warning: package 'bestglm' was built under R version 4.1.3
## Loading required package: leaps
## Loading required package: VGAM
## Warning: package 'VGAM' was built under R version 4.1.3
## Loading required package: stats4
## Loading required package: splines
## Loading required package: rpart
## Loading required package: randomForest
## Warning: package 'randomForest' was built under R version 4.1.3
## randomForest 4.7-1.1
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:ggplot2':
##
##     margin
## The following object is masked from 'package:dplyr':
##
##     combine
## Important regclass change from 1.3:
## All functions that had a . in the name now have an _
## all.correlations -> all_correlations, cor.demo -> cor_demo, etc.
```

```
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
## The following object is masked from 'package:VGAM':
##
##     logit
## The following object is masked from 'package:purrr':
##
##     some
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
setwd('/Users/logan/Documents/GitHub/DSC520LQ')
housing_df <- read_excel('/Users/logan/Documents/GitHub/DSC520LQ/week-6-housing.xlsx')
var_names <- colnames(housing_df)
var_names
```

```
##  [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"
```

```
# I.) Explain any transformations or modifications you are making to the dataset
## I will be removing some variables that are not useful in my analysis or that had missing data.
## I ended up renaming Sale Price due to errors with variables in second equation as well.

refined_cols <- c('Sale Date', 'Sale Price', 'zip5', 'bedrooms', 'sq_ft_lot', 'square_feet_total_living
housing_df_final <- housing_df[refined_cols]
names(housing_df_final)[names(housing_df_final) == 'Sale Price'] <- 'sale_price'

# II.) Create two variables one that will contain the variables Sale Price and Square Foot of Lot (same
# and one that will contain Sale Price and several additional predictors of your choice. Explain the ba

price_lot_lm <- lm(sale_price ~ sq_ft_lot, housing_df_final)
multiple_pred_lm <- lm(sale_price ~ square_feet_total_living + sq_ft_lot + bedrooms + year_built, housi

## Chose predictors zip code, square feet total living and year built.
## These are common factors that go into pricing a house such as price/sq ft total living, new construc

# III.) Execute a summary() function on two variables defined in the previous step to compare the model
summary(price_lot_lm)
```

```
##
## Call:
## lm(formula = sale_price ~ sq_ft_lot, data = housing_df_final)
##
## 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(multiple_pred_lm)
```

```
##
## Call:
## lm(formula = sale_price ~ square_feet_total_living + sq_ft_lot +
##     bedrooms + year_built, data = housing_df_final)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2147040  -119970   -41454    45830  3766517
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -5.333e+06  4.046e+05 -13.181  < 2e-16 ***
## square_feet_total_living  1.717e+02  4.425e+00  38.809  < 2e-16 ***
## sq_ft_lot                 2.729e-01  5.917e-02   4.612 4.02e-06 ***
## bedrooms                 -9.085e+03  4.557e+03  -1.994   0.0462 *
## year_built                2.801e+03  2.031e+02  13.789  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 357100 on 12860 degrees of freedom
## Multiple R-squared:  0.2202, Adjusted R-squared:   0.22
## F-statistic: 907.9 on 4 and 12860 DF,  p-value: < 2.2e-16
```

```
## The R2 for the simple regression for Sale Price/Sq_ft_lot were
## R2 = 0.01435, Adjusted R2 = 0.01428
## The R2 for the multiple predictors were
## R2 = 0.2202, Adjusted R2 = 0.22
## As R2 is a measure of fit for the model we can see neither model is a great fit for the data.
## However, the additional predictor model performed much better (0.014 vs 0.2202)
## likely due to the predictors (sq_foot_total_living, sq_ft_lot, bedrooms and year_built) being more p

# 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?

library(lm.beta)
lm.beta(price_lot_lm)
```

```
##
## Call:
## lm(formula = sale_price ~ sq_ft_lot, data = housing_df_final)
##
## Standardized Coefficients::
## (Intercept)   sq_ft_lot
##   0.0000000   0.1198122
```

```
lm.beta(multiple_pred_lm)
```

```
##
## Call:
## lm(formula = sale_price ~ square_feet_total_living + sq_ft_lot +
##     bedrooms + year_built, data = housing_df_final)
```

```
##
## Standardized Coefficients::
##              (Intercept) square_feet_total_living                sq_ft_lot
##               0.00000000               0.42036390               0.03842553
##                 bedrooms               year_built
##              -0.01968450               0.11928935
```

```
## These standarized betas for the simple regression indicate that for every standard deviation change
## The standardized betas for the multiple predicors indicate that for every std dev change in the pred
## square_feet_total_living = 0.4203, sq_ft_lot = .038, year_built = 0.119, bedrooms = -0.19
## This indicates that square_feet_total_living drives the biggest increase in Sales Price.

# V.) Calculate the confidence intervals for the parameters in your model and explain what the results
confint.lm(price_lot_lm, parameter=0.95)
```

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

```
confint.lm(multiple_pred_lm, parameter=0.95)
```

```
##                                 2.5 %        97.5 %
## (Intercept)              -6.125803e+06 -4.539774e+06
## square_feet_total_living  1.630620e+02  1.804098e+02
## sq_ft_lot                 1.569387e-01  3.889127e-01
## bedrooms                 -1.801844e+04 -1.525239e+02
## year_built                2.403070e+03  3.199472e+03
```

```
## The results indicate that with 95% confidence the results would be within these two bounds for the p
## These results also confirm that sq ft total living is the best predictor due to the smallest margin

# VI.) Assess the improvement of the new model compared to your original model (simple regression model
# whether this change is significant by performing an analysis of variance.
anova(price_lot_lm,multiple_pred_lm)
```

```
## Analysis of Variance Table
##
## Model 1: sale_price ~ sq_ft_lot
## Model 2: sale_price ~ square_feet_total_living + sq_ft_lot + bedrooms +
##     year_built
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
## 1  12863 2.0734e+15
## 2  12860 1.6404e+15  3 4.3302e+14 1131.6 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# The F score of 1131.6 and a p value under 0.05 (2.2e-16) indicate that the changes were statistically

# 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

residuals <- resid(multiple_pred_lm)
stdresiduals <- rstandard(multiple_pred_lm)
stdntresiduals <- rstudent(multiple_pred_lm)
cookscases <- cooks.distance(multiple_pred_lm)
dfbetacases <- dfbeta(multiple_pred_lm)
dffitcases <- dffits(multiple_pred_lm)
```

```r
leveragecases <- hatvalues(multiple_pred_lm)
covratiocases <- covratio(multiple_pred_lm)

casewise_df <- data.frame(residuals, stdresiduals, stdntresiduals, cookscases, dfbetacases, dffitcases,

# VIII.) Calculate the standardized residuals using the appropriate command, specifying those that are
# storing the results of large residuals in a variable you create.

casewise_df$largeres <- casewise_df$stdresiduals > 2 | casewise_df$stdresiduals < -2

# IX.) Use the appropriate function to show the sum of large residuals
sum(casewise_df$largeres_df)
```

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

```r
# X.) Which specific variables have large residuals (only cases that evaluate as TRUE)
colnames(casewise_df)
```

```
##  [1] "residuals"                "stdresiduals"
##  [3] "stdntresiduals"           "cookscases"
##  [5] "X.Intercept."             "square_feet_total_living"
##  [7] "sq_ft_lot"                "bedrooms"
##  [9] "year_built"               "dffitcases"
## [11] "leveragecases"            "covratiocases"
## [13] "largeres"
```

```r
large_residuals_var <- casewise_df$largeres
sum(large_residuals_var)
```

```
## [1] 335
```

```r
## There are 335 variables that have large residuals.

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

problematics <- casewise_df[casewise_df$largeres, c("cookscases", "leveragecases",'covratiocases')]
problematics
```

```
##         cookscases leveragecases covratiocases
## 6     0.0003240312  0.0003401775     0.9988782
## 25    0.0007865132  0.0006715173     0.9987854
## 115   0.0047459701  0.0026231740     0.9995064
## 160   0.0034078340  0.0034820171     1.0019825
## 178   0.0011037879  0.0009282485     0.9990080
## 239   0.0006287502  0.0007246524     0.9994281
## 246   0.0047813247  0.0018843263     0.9973525
## 287   0.0007293987  0.0005447957     0.9983332
## 295   0.0009048370  0.0007087467     0.9986180
## 300   0.0072561886  0.0026945880     0.9978657
## 341   0.0003590686  0.0004155890     0.9991257
## 344   0.0082677601  0.0084873716     1.0070592
## 359   0.0020833468  0.0013002693     0.9985792
## 385   0.0035339447  0.0016722548     0.9979615
## 396   0.0016338162  0.0010402006     0.9983797
## 475   0.0005856695  0.0005399580     0.9988217
```

```
## 482   0.0013294210  0.0013910718     0.9999252
## 508   0.0030437630  0.0023072743     1.0001395
## 528   0.0011173074  0.0010257866     0.9992996
## 576   0.0010115371  0.0012215718     1.0000031
## 661   0.0044081787  0.0035938710     1.0016140
## 670   0.0025062163  0.0006567978     0.9936475
## 679   0.0088317665  0.0029877643     0.9976512
## 784   0.0018092943  0.0014351925     0.9993774
## 802   0.0010063379  0.0012063064     0.9999758
## 811   0.0049123216  0.0009099768     0.9908456
## 853   0.0007058860  0.0008016599     0.9994803
## 877   0.0017697971  0.0012324103     0.9988336
## 916   0.0050484746  0.0024418810     0.9988234
## 1009  0.0002540462  0.0002690618     0.9988232
## 1119  0.0030017253  0.0016510226     0.9985125
## 1142  0.0021206587  0.0013066399     0.9985454
## 1155  0.0172726958  0.0079820273     1.0042369
## 1305  0.0014548006  0.0014532345     0.9998995
## 1368  0.0009025077  0.0010966004     0.9998876
## 1380  0.0003882723  0.0003702544     0.9987217
## 1442  0.0021360702  0.0017628304     0.9998012
## 1492  0.0005239153  0.0005363802     0.9990276
## 1504  0.0072045866  0.0031566648     0.9991260
## 1550  0.0012051683  0.0008878024     0.9986406
## 1633  0.0005019761  0.0005026186     0.9989512
## 1650  0.0217550455  0.0058688344     0.9991063
## 1716  0.0010248719  0.0005472079     0.9972995
## 1745  0.0015079748  0.0008682439     0.9978851
## 1870  0.0083979026  0.0069304211     1.0050160
## 1962  0.0029142766  0.0011253299     0.9964897
## 1963  0.0004509376  0.0003680614     0.9983768
## 1964  0.0019868661  0.0007499854     0.9959982
## 1976  0.0017528192  0.0003797357     0.9918242
## 1977  0.0009112237  0.0001808120     0.9908072
## 1978  0.0009130558  0.0001860725     0.9910685
## 1979  0.0008901442  0.0001774381     0.9908482
## 1980  0.0008930300  0.0001772415     0.9908058
## 1981  0.0035347964  0.0007933420     0.9925481
## 1982  0.0018086844  0.0003807502     0.9915657
## 2022  0.0031718390  0.0015713136     0.9980437
## 2099  0.0007531715  0.0006608847     0.9988361
## 2137  0.0007537450  0.0003737193     0.9968467
## 2157  0.0006808854  0.0004326814     0.9977653
## 2257  0.0002310956  0.0002447384     0.9987988
## 2264  0.0024661868  0.0009148376     0.9960733
## 2302  0.0051196103  0.0045035507     1.0027057
## 2360  0.0008071574  0.0005916226     0.9983306
## 2361  0.0048820278  0.0015557052     0.9958595
## 2469  0.0004786945  0.0003011629     0.9976029
## 2604  0.0001916116  0.0002106094     0.9988315
## 2684  0.0018763871  0.0003877680     0.9914017
## 2685  0.0010264948  0.0002268459     0.9918466
## 2686  0.0018114657  0.0004688356     0.9933666
## 2687  0.0009522420  0.0002029393     0.9914998
```

```
## 2688  0.0008510969  0.0001743425     0.9911054
## 2689  0.0017370598  0.0004342537     0.9930684
## 2690  0.0017636648  0.0004473110     0.9931922
## 2699  0.0483078230  0.0252934719     1.0226399
## 2708  0.0014904737  0.0006646073     0.9967004
## 2709  0.0016364919  0.0008455465     0.9974772
## 2710  0.0017156144  0.0008386260     0.9972564
## 2717  0.0037008622  0.0012609738     0.9959572
## 2742  0.0004904024  0.0003397552     0.9979250
## 2852  0.0374216977  0.0079162189     0.9992115
## 2867  0.0007019562  0.0008701039     0.9996921
## 2934  0.0010239339  0.0005472079     0.9973029
## 2937  0.0009895288  0.0005472079     0.9974248
## 3097  0.0031620235  0.0011311475     0.9960971
## 3102  0.0009529929  0.0006077096     0.9979512
## 3110  0.0007077977  0.0005809253     0.9986031
## 3111  0.0007423807  0.0007385918     0.9991751
## 3168  0.0045449066  0.0004036234     0.9790851
## 3169  0.0045449066  0.0004036234     0.9790851
## 3170  0.0031231487  0.0002876399     0.9797382
## 3171  0.0025586529  0.0002393943     0.9800147
## 3172  0.0026449398  0.0002465421     0.9799451
## 3173  0.0025633861  0.0002398885     0.9800196
## 3174  0.0026442202  0.0002464680     0.9799445
## 3175  0.0034628999  0.0003123700     0.9793270
## 3176  0.0024807347  0.0002351148     0.9802722
## 3177  0.0023159581  0.0002196155     0.9802675
## 3178  0.0023174486  0.0002197765     0.9802695
## 3179  0.0029795741  0.0002737858     0.9796765
## 3180  0.0023195389  0.0002200004     0.9802721
## 3181  0.0029780878  0.0002736378     0.9796755
## 3182  0.0045453479  0.0004036773     0.9790859
## 3183  0.0045445612  0.0004035808     0.9790844
## 3184  0.0045445612  0.0004035808     0.9790844
## 3185  0.0045454010  0.0004036837     0.9790860
## 3186  0.0045449066  0.0004036234     0.9790851
## 3187  0.0045449066  0.0004036234     0.9790851
## 3188  0.0045486003  0.0004040591     0.9790912
## 3189  0.0034606561  0.0003121513     0.9793257
## 3190  0.0026408383  0.0002461186     0.9799413
## 3191  0.0034597595  0.0003120638     0.9793251
## 3192  0.0025591031  0.0002394416     0.9800152
## 3193  0.0026400875  0.0002460407     0.9799405
## 3194  0.0026395949  0.0002459894     0.9799400
## 3195  0.0023166867  0.0002196944     0.9802685
## 3196  0.0023182373  0.0002198612     0.9802705
## 3197  0.0024917909  0.0002362780     0.9802832
## 3198  0.0045463037  0.0004037919     0.9790876
## 3199  0.0045462216  0.0004037822     0.9790875
## 3200  0.0045445612  0.0004035808     0.9790844
## 3201  0.0045445612  0.0004035808     0.9790844
## 3202  0.0045445612  0.0004035808     0.9790844
## 3260  0.0003796139  0.0004697513     0.9992884
## 3424  0.0005659668  0.0003791051     0.9978683
```

```
## 3464  0.0033943936  0.0003120044     0.9797216
## 3465  0.0030458913  0.0002857896     0.9801189
## 3466  0.0022708341  0.0002195401     0.9806538
## 3467  0.0022741613  0.0002199068     0.9806582
## 3468  0.0033967377  0.0003122376     0.9797230
## 3469  0.0029143966  0.0002728801     0.9800633
## 3470  0.0022700349  0.0002194508     0.9806526
## 3471  0.0030475671  0.0002859613     0.9801201
## 3472  0.0025102228  0.0002394315     0.9804048
## 3473  0.0025924575  0.0002463157     0.9803336
## 3474  0.0034022566  0.0003127856     0.9797262
## 3475  0.0025149283  0.0002399324     0.9804097
## 3476  0.0022938659  0.0002214771     0.9806298
## 3477  0.0025133073  0.0002397606     0.9804081
## 3478  0.0044601322  0.0004035808     0.9794842
## 3479  0.0044601322  0.0004035808     0.9794842
## 3480  0.0044601322  0.0004035808     0.9794842
## 3481  0.0044601322  0.0004035808     0.9794842
## 3482  0.0044601322  0.0004035808     0.9794842
## 3483  0.0044601344  0.0004035810     0.9794842
## 3484  0.0044601254  0.0004035799     0.9794842
## 3485  0.0025065718  0.0002433760     0.9807626
## 3486  0.0029136086  0.0002727994     0.9800627
## 3487  0.0023855584  0.0002311270     0.9807078
## 3488  0.0025894149  0.0002459939     0.9803306
## 3489  0.0025084085  0.0002392364     0.9804028
## 3490  0.0025925672  0.0002463272     0.9803337
## 3491  0.0041923786  0.0004022819     0.9806876
## 3492  0.0030461648  0.0002858177     0.9801191
## 3493  0.0022715157  0.0002196158     0.9806547
## 3494  0.0033973201  0.0003122956     0.9797233
## 3495  0.0044601344  0.0004035810     0.9794842
## 3496  0.0044601344  0.0004035810     0.9794842
## 3497  0.0044601344  0.0004035810     0.9794842
## 3523  0.0023044383  0.0010135181     0.9969898
## 3837  0.0007340967  0.0005286531     0.9982204
## 3918  0.0018950937  0.0016510226     0.9998130
## 3919  0.0013819232  0.0013066399     0.9996426
## 4055  0.0006873189  0.0003741231     0.9971956
## 4056  0.0081831731  0.0016843061     0.9926636
## 4248  0.0009505931  0.0011731651     0.9999891
## 4285  0.0003868698  0.0003872595     0.9988351
## 4391  0.0019064458  0.0019959097     1.0005334
## 4435  0.0004215942  0.0003974469     0.9987254
## 4571  0.0008125303  0.0009614349     0.9997091
## 4648  0.1455620065  0.0101978847     0.9832400
## 4649  0.9623861409  0.0441602393     1.0049059
## 4671  0.0039230534  0.0032684268     1.0013372
## 4695  0.0017358361  0.0003452645     0.9909951
## 4696  0.0018647562  0.0009250220     0.9974008
## 4740  0.0003508655  0.0003711812     0.9989234
## 4821  0.0011608990  0.0007145214     0.9979485
## 4834  0.0027139781  0.0015713136     0.9986088
## 4840  0.0180519961  0.0079820273     1.0040477
```

```
## 4934  0.0014439853  0.0010529200     0.9987791
## 5083  0.1050613500  0.0298030096     1.0242819
## 5491  0.0001849418  0.0001961399     0.9987528
## 5494  0.0001788007  0.0001937871     0.9987897
## 5495  0.0001853408  0.0001963767     0.9987513
## 5496  0.0001849829  0.0001961621     0.9987526
## 5497  0.0004875578  0.0005019856     0.9990039
## 5549  0.0005858141  0.0006201300     0.9991738
## 5935  0.0029306766  0.0015412890     0.9982407
## 6055  0.0013451798  0.0013310266     0.9997583
## 6230  0.0013486804  0.0002053038     0.9878838
## 6231  0.0015053454  0.0002387630     0.9884264
## 6232  0.0015600205  0.0002550420     0.9888049
## 6233  0.0014235139  0.0002112187     0.9875619
## 6234  0.0028982573  0.0004385596     0.9880415
## 6235  0.0029265643  0.0004413770     0.9880021
## 6236  0.0028981590  0.0004385549     0.9880418
## 6237  0.0030017656  0.0004539386     0.9880489
## 6238  0.0014586582  0.0002132371     0.9873695
## 6239  0.0015058028  0.0002388307     0.9884263
## 6429  0.0040024484  0.0002028496     0.9628035
## 6430  0.0053953774  0.0002516256     0.9596301
## 6431  0.0047126921  0.0002267872     0.9608507
## 6432  0.0041295604  0.0002037975     0.9618018
## 6433  0.0087623610  0.0004362891     0.9623721
## 6434  0.0092268516  0.0004608433     0.9625139
## 6435  0.0078830222  0.0004104969     0.9640065
## 6436  0.0045683525  0.0002213424     0.9611106
## 6437  0.0049408204  0.0002325778     0.9599842
## 6438  0.0040366167  0.0001893061     0.9597914
## 6439  0.0038327293  0.0001915016     0.9622587
## 6440  0.0040782409  0.0002089087     0.9632047
## 6441  0.0042250210  0.0002073743     0.9615965
## 6442  0.0040454525  0.0002014587     0.9621429
## 6443  0.0094988841  0.0005287937     0.9664680
## 6444  0.0079486689  0.0004572522     0.9674932
## 6445  0.0051811656  0.0002794175     0.9651233
## 6446  0.0038564032  0.0002178753     0.9666548
## 6447  0.0071652884  0.0004231482     0.9683117
## 6448  0.0044312471  0.0002399223     0.9652221
## 6449  0.0069083692  0.0004132157     0.9687085
## 6450  0.0066209979  0.0004087786     0.9696926
## 6451  0.0048883076  0.0002634415     0.9650830
## 6452  0.0038282873  0.0002167251     0.9667214
## 6453  0.0070268516  0.0004223445     0.9688708
## 6454  0.0076594123  0.0004443280     0.9677555
## 6455  0.0099471776  0.0005499928     0.9662572
## 6456  0.0043199242  0.0002366950     0.9656318
## 6457  0.0075608879  0.0004399774     0.9678524
## 6512  0.0028243110  0.0008863531     0.9950948
## 6527  0.0007215151  0.0006294394     0.9987917
## 6739  0.0005688433  0.0004404078     0.9983203
## 6766  0.0061833846  0.0075696246     1.0064316
## 6796  0.0002650325  0.0003116946     0.9990483
```

```
## 6821  0.0010587897  0.0010426478     0.9994597
## 6931  0.0002776294  0.0002878847     0.9988029
## 6938  0.0008703473  0.0003058932     0.9951738
## 6939  0.0012570281  0.0005235826     0.9962526
## 6940  0.0007121605  0.0002155038     0.9941943
## 6941  0.0007078658  0.0002148666     0.9942133
## 6942  0.0012064574  0.0004666648     0.9958383
## 6943  0.0012066695  0.0004654257     0.9958228
## 6944  0.0013030507  0.0005624303     0.9964543
## 6945  0.0008703473  0.0003058932     0.9951738
## 6946  0.0007117867  0.0002159188     0.9942103
## 6947  0.0007150003  0.0002113652     0.9940390
## 6948  0.0007189595  0.0002101517     0.9939635
## 7021  0.0019426974  0.0024164405     1.0012496
## 7039  0.0012872978  0.0012829585     0.9997242
## 7147  0.0010650809  0.0007758639     0.9984988
## 7167  0.0062523105  0.0007759340     0.9855930
## 7210  0.0038159679  0.0010692859     0.9945386
## 7211  0.0031335963  0.0007253833     0.9927420
## 7446  0.0029786093  0.0002536820     0.9780157
## 7447  0.0033096087  0.0002773280     0.9776720
## 7448  0.0029681348  0.0002529710     0.9780310
## 7449  0.0024706536  0.0002084076     0.9777539
## 7450  0.0024726859  0.0002084484     0.9777397
## 7451  0.0024747368  0.0002084914     0.9777257
## 7452  0.0033106530  0.0002774294     0.9776733
## 7453  0.0028772831  0.0002439620     0.9779058
## 7454  0.0028549057  0.0002423727     0.9779328
## 7455  0.0031256836  0.0002617309     0.9776403
## 7456  0.0033094595  0.0002773135     0.9776718
## 7457  0.0030012138  0.0002552784     0.9779884
## 7458  0.0024714611  0.0002082496     0.9777291
## 7459  0.0035306182  0.0002930662     0.9774716
## 7460  0.0024726859  0.0002084484     0.9777397
## 7461  0.0034385220  0.0002864896     0.9775507
## 7462  0.0034840917  0.0002897447     0.9775112
## 7463  0.0034385934  0.0002864964     0.9775508
## 7507  0.0022149927  0.0014351925     0.9988289
## 7649  0.0003629867  0.0004202732     0.9991310
## 7650  0.0003633113  0.0004207235     0.9991317
## 7791  0.0018965629  0.0019607378     1.0004745
## 7871  0.0066823702  0.0031602211     0.9994548
## 8119  0.0008585507  0.0010426478     0.9998327
## 8154  0.0002867721  0.0003039746     0.9988597
## 8232  0.0022315101  0.0014351925     0.9988065
## 8262  0.0321921228  0.0116398653     1.0068032
## 8320  0.0095544493  0.0049550613     1.0016262
## 8377  0.5278303936  0.0639808918     1.0528218
## 8457  0.0004367617  0.0003680614     0.9984516
## 8458  0.0019442656  0.0007499854     0.9961083
## 8535  0.0004539890  0.0004750345     0.9990071
## 8541  0.0005051274  0.0005026186     0.9989390
## 8698  0.0002475346  0.0002223235     0.9984479
## 8710  0.0027974029  0.0006291977     0.9924021
```

```
## 8763  0.0006119122  0.0004740102     0.9983553
## 8887  0.0108275135  0.0017443407     0.9901234
## 8911  0.0013434042  0.0012523507     0.9995589
## 8946  0.0013434042  0.0012523507     0.9995589
## 9215  0.0038411161  0.0028340795     1.0005991
## 9293  0.0011443070  0.0010257866     0.9992485
## 9369  0.0018890072  0.0020387833     1.0006320
## 9420  0.0068031389  0.0052250914     1.0031139
## 9453  0.0005412740  0.0005472079     0.9990144
## 9528  0.0094198048  0.0025857453     0.9959176
## 9546  0.0007032097  0.0005286531     0.9983338
## 9722  0.0006156877  0.0003711812     0.9975387
## 10125 0.0019940892  0.0020851676     1.0006208
## 10318 0.0216423861  0.0223926837     1.0214252
## 10371 0.0008177928  0.0007100602     0.9988619
## 10418 0.0019509311  0.0018592855     1.0002134
## 10623 0.0005587155  0.0005960403     0.9991639
## 10707 0.0001205613  0.0001228020     0.9986039
## 10723 0.0005685941  0.0006111940     0.9991928
## 10741 0.0002882247  0.0003543058     0.9991624
## 10787 0.0036447505  0.0013549077     0.9965256
## 10844 0.0019345571  0.0016035455     0.9996516
## 10958 0.0006936713  0.0005236912     0.9983397
## 10995 0.0037596234  0.0038339254     1.0023334
## 11165 0.0005092099  0.0003552569     0.9979600
## 11289 0.0008315614  0.0002218824     0.9933436
## 11413 0.0039266679  0.0012259344     0.9954032
## 11558 0.0042413983  0.0007451496     0.9901141
## 11586 0.0022977140  0.0010848807     0.9973632
## 11728 0.0095531185  0.0047264310     1.0012150
## 11758 0.0008760885  0.0007100602     0.9987025
## 11772 0.0177302690  0.0020066766     0.9853345
## 11822 0.0034721766  0.0006384908     0.9904968
## 11898 0.0570217579  0.0350849815     1.0336069
## 11899 0.1187767849  0.0370160658     1.0326174
## 11982 0.0095990661  0.0024418810     0.9952161
## 11992 0.1023364898  0.0045587795     0.9620658
## 12212 0.0023446837  0.0016436109     0.9992646
## 12255 0.0031498730  0.0014169344     0.9974926
## 12256 0.0015214308  0.0011177163     0.9988639
## 12392 0.0009154026  0.0008564528     0.9991695
## 12472 0.0038898867  0.0008731282     0.9926289
## 12487 0.0036167660  0.0015660616     0.9974747
## 12577 0.0060419341  0.0016677942     0.9950346
## 12582 0.0287602671  0.0027484092     0.9829597
## 12643 0.0098430722  0.0008252960     0.9782344
## 12686 0.0003446506  0.0003850846     0.9990349
## 12759 0.0003729966  0.0004465431     0.9992124
## 12764 0.0077344820  0.0022281006     0.9958901
```

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

dwt(multiple_pred_lm)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.7209323     0.5581258       0
##  Alternative hypothesis: rho != 0
```

```
## The statistic is less than 1 which means the assumption has not been met.

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

VIF(multiple_pred_lm)
```

```
## square_feet_total_living                sq_ft_lot                 bedrooms
##                 1.934814                 1.144597                 1.607780
##               year_built
##                 1.234185
```

```
1/vif(multiple_pred_lm)
```

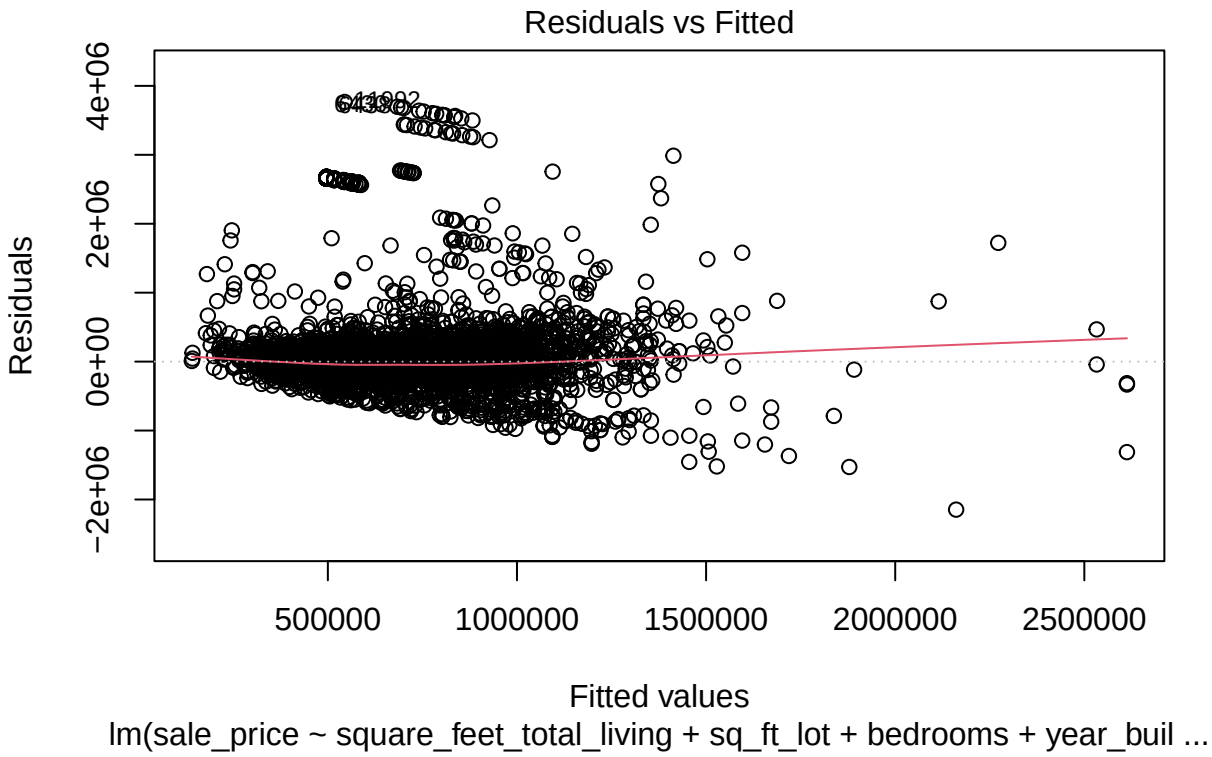
```
## square_feet_total_living                sq_ft_lot                 bedrooms
##                0.5168455                0.8736699                0.6219757
##               year_built
##                0.8102512
```

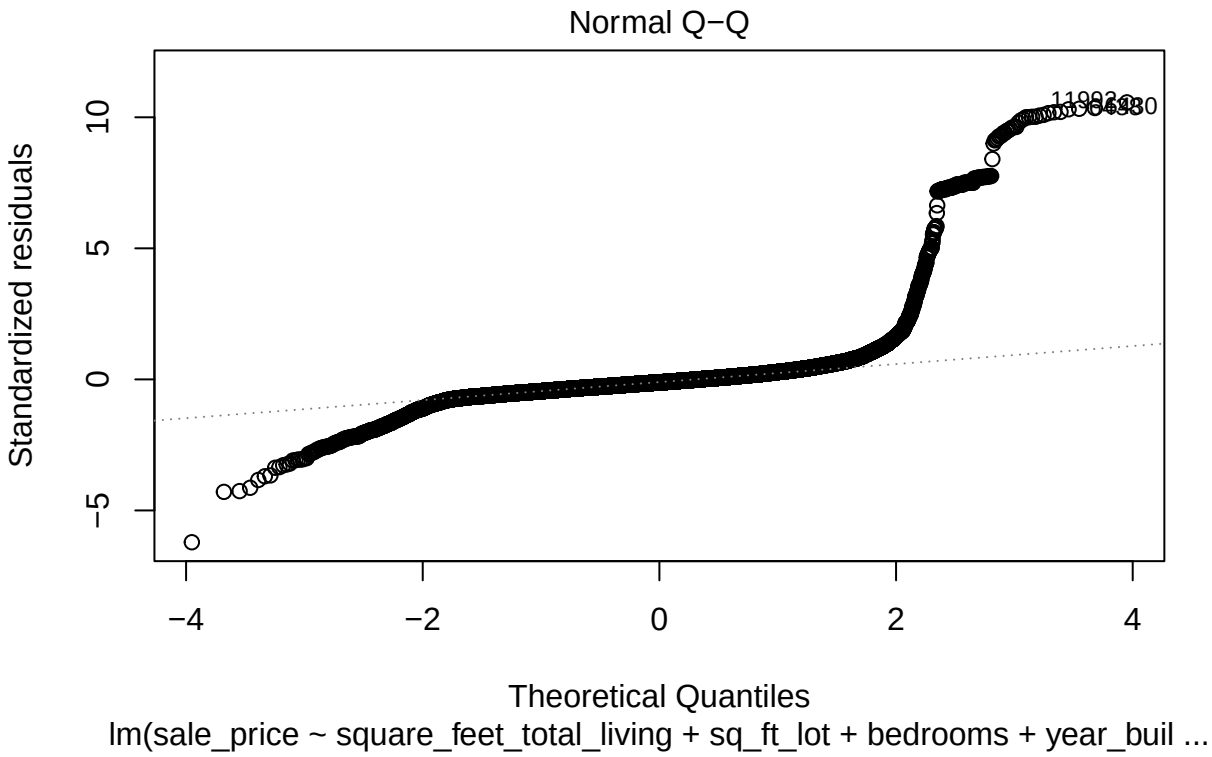
```
mean(vif(multiple_pred_lm))
```

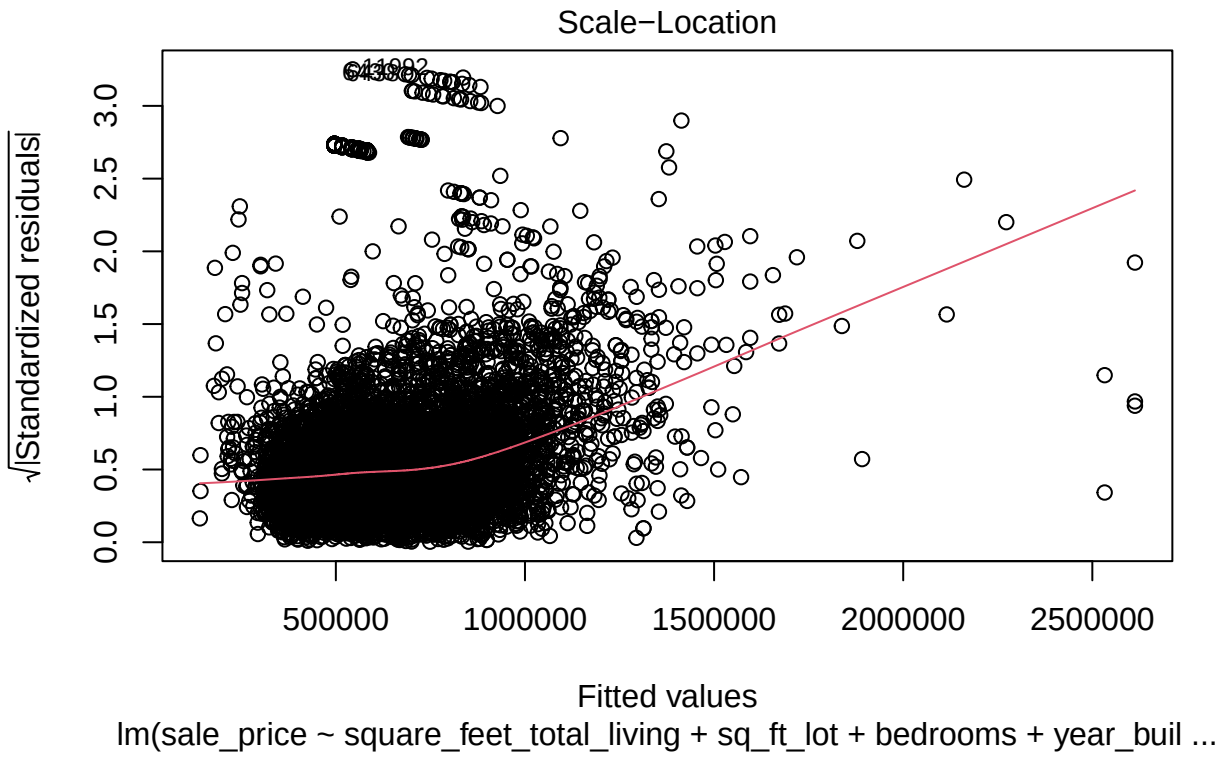
```
## [1] 1.480344
```

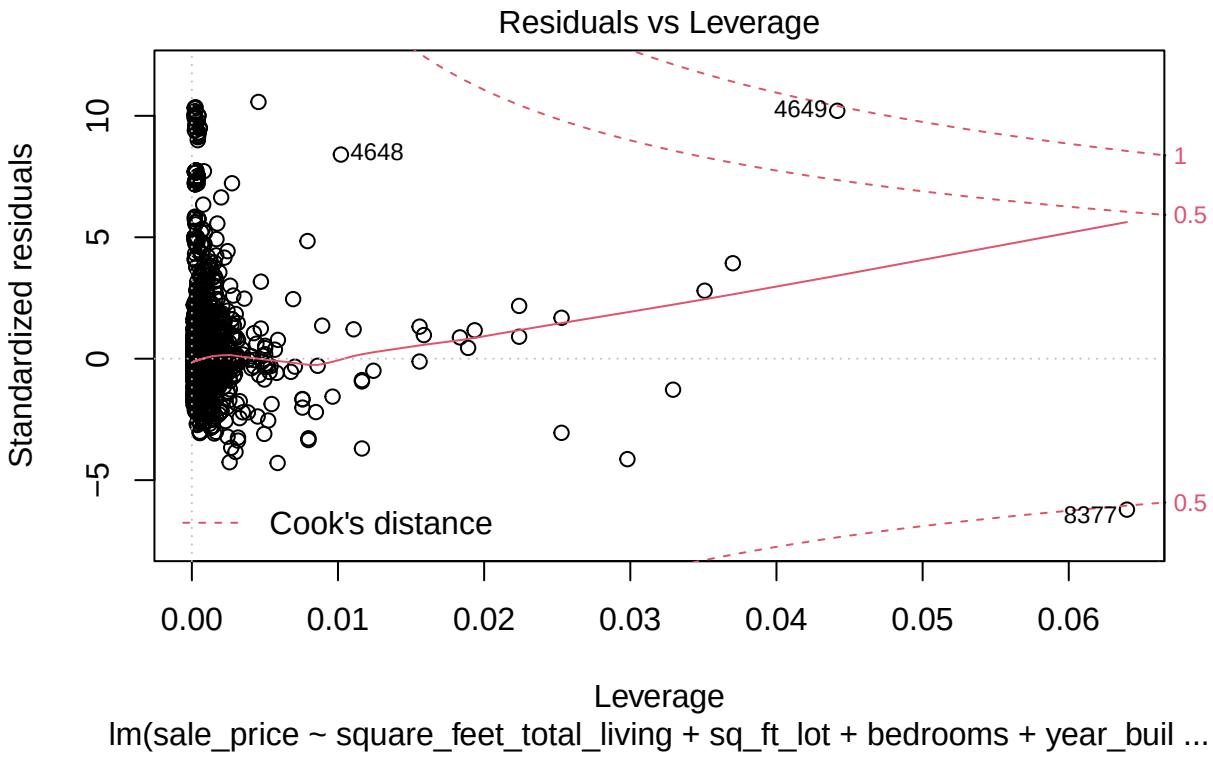
```
## The VIFs are all below 10 and the tolerance statistics are all above 0.2 and the average is only sli

# 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(multiple_pred_lm)
```

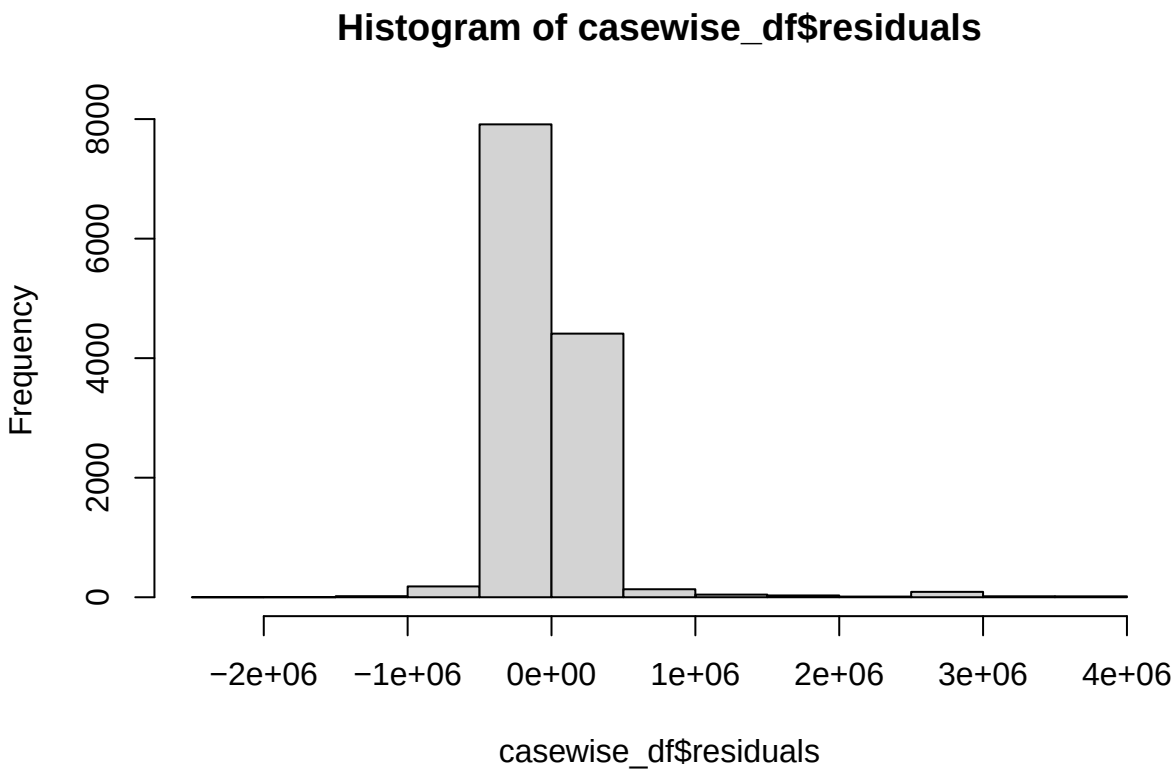

Residuals vs Fitted
Residuals
4e+06
2e+06
0e+00
−2e+06
500000
1000000
1500000
2000000
2500000
Fitted values
lm(sale_price ~ square_feet_total_living + sq_ft_lot + bedrooms + year_buil ...


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

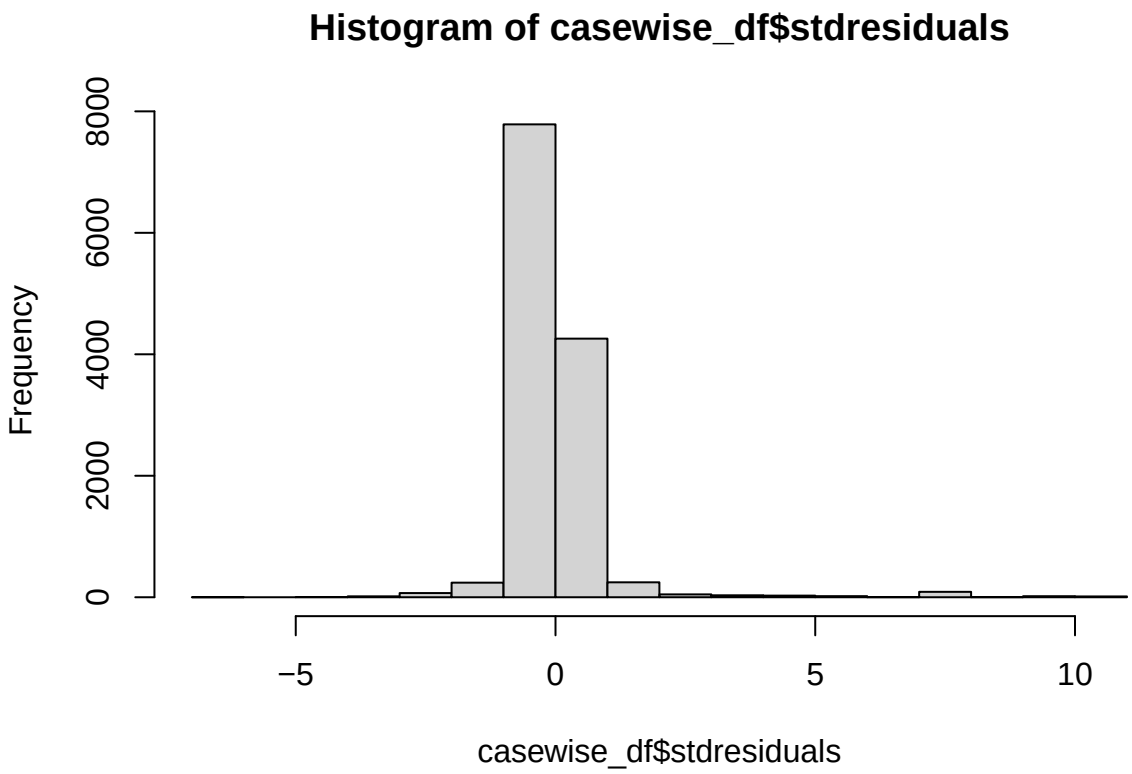

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
500000
1000000
1500000
2000000
2500000
Fitted values
lm(sale_price ~ square_feet_total_living + sq_ft_lot + bedrooms + year_buil ...



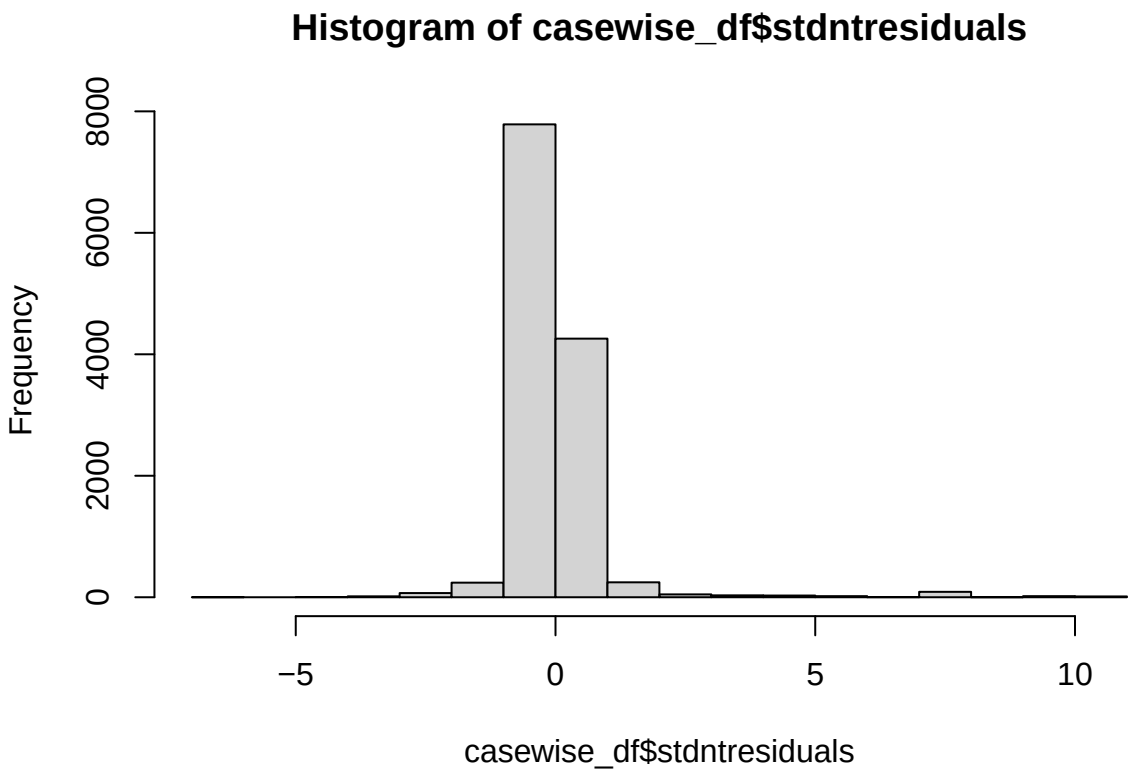

```
hist(casewise_df$residuals)
```




```
hist(casewise_df$stdresiduals)
```




```
hist(casewise_df$stdntresiduals)
```




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

## Overall I would say the regression model is biased as there are 335 outliers and although not by muc
```