assignment_8.2_VenkidusamyKesavAdithya

Kesav Adithya Venkidusamy

10/24/2021

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 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.

```
#Question a
#If you worked with the Housing dataset in previous week - you are in luck, you likely have already fou

#Load required libraries for analysis
library(ggplot2)
library(dplyr)
```

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

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

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

```
library(pastecs)
```

```
##
## Attaching package: 'pastecs'

## The following objects are masked from 'package:dplyr':
##
##     first, last
```

```
library("readxl")

#Create data frame for the input survey file
raw_housing_df <- read_excel("E:/Personal/Bellevue University/Course/github/dsc520/data/week-7-housing.x

#Print column and row names
print(colnames(raw_housing_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"
```

```
#Print the dimension of the dataframe
print(dim(raw_housing_df))
```

```
## [1] 12865    24
```

```
#Calculate the str for the dataframe
print(str(raw_housing_df))
```

```
## tibble [12,865 x 24] (S3: tbl_df/tbl/data.frame)
##  $ Sale Date               : POSIXct[1:12865], format: "2006-01-03" "2006-01-03" ...
##  $ Sale Price              : num [1:12865] 698000 649990 572500 420000 369900 ...
##  $ sale_reason             : num [1:12865] 1 1 1 1 1 1 1 1 1 1 ...
##  $ sale_instrument         : num [1:12865] 3 3 3 3 3 15 3 3 3 3 ...
##  $ sale_warning            : chr [1:12865] NA NA NA NA ...
##  $ sitetype                : chr [1:12865] "R1" "R1" "R1" "R1" ...
##  $ addr_full               : chr [1:12865] "17021 NE 113TH CT" "11927 178TH PL NE" "13315 174TH AVE 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 ...
## NULL
```

```
# Question b
# Explain any transformations or modifications you made to the data set

#Removing unwanted columns from data frame
remove_cols <- c(3,4,5,6,7,9,13,21,23,24)
clean_housing_df <- select(raw_housing_df, -all_of(remove_cols))
colnames(clean_housing_df)
```

```
##  [1] "Sale Date"               "Sale Price"
##  [3] "zip5"                    "postalctyn"
##  [5] "lon"                     "lat"
##  [7] "square_feet_total_living" "bedrooms"
##  [9] "bath_full_count"         "bath_half_count"
## [11] "bath_3qtr_count"         "year_built"
## [13] "year_renovated"          "sq_ft_lot"
```

```
#Renaming the column names to remove space in the column names
rename_housing_df <- rename(clean_housing_df, "sale_price" = "Sale Price","sale_date" = "Sale Date")
colnames(rename_housing_df)
```

```
##  [1] "sale_date"               "sale_price"
##  [3] "zip5"                    "postalctyn"
##  [5] "lon"                     "lat"
##  [7] "square_feet_total_living" "bedrooms"
##  [9] "bath_full_count"         "bath_half_count"
## [11] "bath_3qtr_count"         "year_built"
## [13] "year_renovated"          "sq_ft_lot"
```

```
head(rename_housing_df,10)
```

```
## # A tibble: 10 x 14
##    sale_date           sale_price  zip5 postalctyn   lon   lat square_feet_tota~
##    <dttm>                   <dbl> <dbl> <chr>      <dbl> <dbl>             <dbl>
##  1 2006-01-03 00:00:00     698000 98052 REDMOND    -122.  47.7              2810
##  2 2006-01-03 00:00:00     649990 98052 REDMOND    -122.  47.7              2880
##  3 2006-01-03 00:00:00     572500 98052 REDMOND    -122.  47.7              2770
##  4 2006-01-03 00:00:00     420000 98052 REDMOND    -122.  47.6              1620
##  5 2006-01-03 00:00:00     369900 98052 REDMOND    -122.  47.7              1440
##  6 2006-01-03 00:00:00     184667 98053 REDMOND    -122.  47.7              4160
##  7 2006-01-04 00:00:00    1050000 98053 REDMOND    -122.  47.7              3960
##  8 2006-01-04 00:00:00     875000 98053 REDMOND    -122.  47.7              3720
##  9 2006-01-04 00:00:00     660000 98053 REDMOND    -122.  47.7              4160
## 10 2006-01-04 00:00:00     650000 98052 REDMOND    -122.  47.6              2760
## # ... with 7 more variables: bedrooms <dbl>, bath_full_count <dbl>,
## #   bath_half_count <dbl>, bath_3qtr_count <dbl>, year_built <dbl>,
## #   year_renovated <dbl>, sq_ft_lot <dbl>
```

```r
# Create two variables; one that will contain the variables Sale Price and Square Foot of Lot (same var
#Add addition columns to calculate price_per_sq_ft and housing size

housing_df <- transform(rename_housing_df,
          price_per_sq_ft=round(sale_price/square_feet_total_living,2),
          housing_size=case_when(square_feet_total_living <= 1000 ~ "Tiny",
          square_feet_total_living > 1000 & square_feet_total_living <= 2000 ~ "Small",
          square_feet_total_living > 2000 & square_feet_total_living <= 3000 ~ "Medium",
          square_feet_total_living > 3000 ~ "Large"))


colnames(housing_df)
```

```
##  [1] "sale_date"                "sale_price"
##  [3] "zip5"                     "postalctyn"
##  [5] "lon"                      "lat"
##  [7] "square_feet_total_living" "bedrooms"
##  [9] "bath_full_count"          "bath_half_count"
## [11] "bath_3qtr_count"          "year_built"
## [13] "year_renovated"           "sq_ft_lot"
## [15] "price_per_sq_ft"          "housing_size"
```

```r
head(housing_df,10)
```

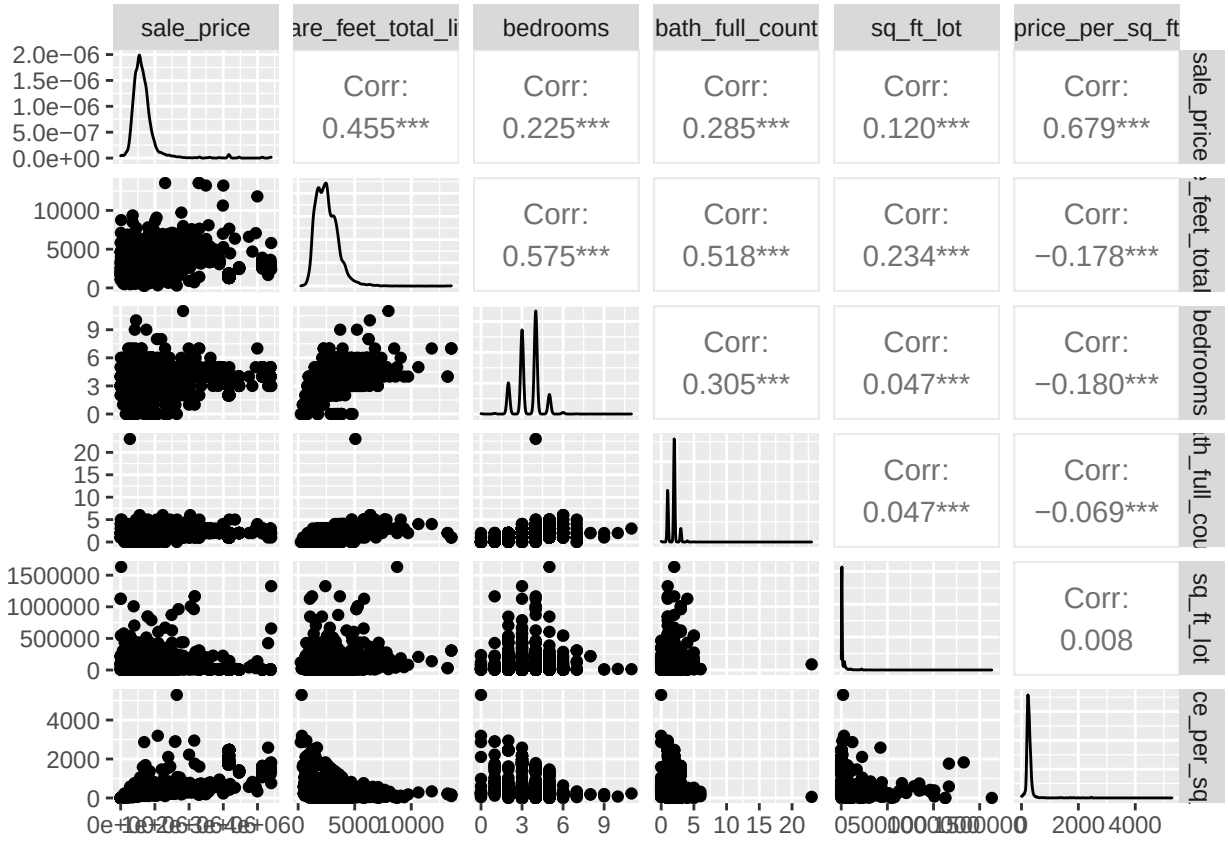
```
##     sale_date sale_price  zip5 postalctyn       lon      lat
## 1  2006-01-03     698000 98052    REDMOND -122.1124 47.70139
## 2  2006-01-03     649990 98052    REDMOND -122.1022 47.70731
## 3  2006-01-03     572500 98052    REDMOND -122.1085 47.71986
## 4  2006-01-03     420000 98052    REDMOND -122.1037 47.63914
## 5  2006-01-03     369900 98052    REDMOND -122.1242 47.69748
## 6  2006-01-03     184667 98053    REDMOND -122.0341 47.67545
## 7  2006-01-04    1050000 98053    REDMOND -122.0507 47.68053
## 8  2006-01-04     875000 98053    REDMOND -122.0555 47.66510
## 9  2006-01-04     660000 98053    REDMOND -122.0227 47.67208
## 10 2006-01-04     650000 98052    REDMOND -122.1039 47.63341
##    square_feet_total_living bedrooms bath_full_count bath_half_count
## 1                      2810        4               2               1
## 2                      2880        4               2               0
## 3                      2770        4               1               1
## 4                      1620        3               1               0
## 5                      1440        3               1               0
## 6                      4160        4               2               1
## 7                      3960        5               3               0
## 8                      3720        4               2               1
## 9                      4160        4               2               1
## 10                     2760        4               1               0
##    bath_3qtr_count year_built year_renovated sq_ft_lot price_per_sq_ft
## 1                0       2003             0      6635          248.40
## 2                1       2006             0      5570          225.69
## 3                1       1987             0      8444          206.68
## 4                1       1968             0      9600          259.26
## 5                1       1980             0      7526          256.88
```

```
## 6                1        2005               0       7280           44.39
## 7                1        1993               0      97574          265.15
## 8                0        1988               0      30649          235.22
## 9                1        1978               0      42688          158.65
## 10               1        1976               0      94889          235.51
##    housing_size
## 1        Medium
## 2        Medium
## 3        Medium
## 4         Small
## 5         Small
## 6         Large
## 7         Large
## 8         Large
## 9         Large
## 10       Medium
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
ggpairs(data=housing_df,columns=c(2,7,8,9,14,15))
```



```
#Observation: From the above chart, We see square_feet_total_living is having strong correlation with "

sapply(housing_df, class)
```

```
## $sale_date
## [1] "POSIXct" "POSIXt"
##
## $sale_price
## [1] "numeric"
##
## $zip5
## [1] "numeric"
##
## $postalctyn
## [1] "character"
##
## $lon
## [1] "numeric"
##
## $lat
## [1] "numeric"
##
## $square_feet_total_living
## [1] "numeric"
##
## $bedrooms
## [1] "numeric"
##
## $bath_full_count
## [1] "numeric"
##
## $bath_half_count
## [1] "numeric"
##
## $bath_3qtr_count
## [1] "numeric"
##
## $year_built
## [1] "numeric"
##
## $year_renovated
## [1] "numeric"
##
## $sq_ft_lot
## [1] "numeric"
##
## $price_per_sq_ft
## [1] "numeric"
##
## $housing_size
## [1] "character"
```

```r
# Simple Linear Regression

price_lm <- lm(sale_price ~ sq_ft_lot, housing_df)
price_lm
```

```
##
## Call:
## lm(formula = sale_price ~ sq_ft_lot, data = housing_df)
##
## Coefficients:
## (Intercept)    sq_ft_lot
##   6.418e+05    8.510e-01
```

```r
zip_lm <- lm(sale_price ~ zip5, housing_df)
zip_lm
```

```
##
## Call:
## lm(formula = sale_price ~ zip5, data = housing_df)
##
## Coefficients:
## (Intercept)         zip5
##  -1.406e+09    1.435e+04
```

```r
sqft_lm <- lm(sale_price ~ square_feet_total_living, housing_df)
sqft_lm
```

```
##
## Call:
## lm(formula = sale_price ~ square_feet_total_living, data = housing_df)
##
## Coefficients:
##              (Intercept)  square_feet_total_living
##                 189106.6                     185.7
```

```r
#Multiple Regression

multi_lm <- lm(sale_price ~ sq_ft_lot + bath_full_count + bedrooms, housing_df)
multi_lm
```

```
##
## Call:
## lm(formula = sale_price ~ sq_ft_lot + bath_full_count + bedrooms,
##     data = housing_df)
##
## Coefficients:
##     (Intercept)        sq_ft_lot  bath_full_count         bedrooms
##       1.429e+05        7.227e-01        1.458e+05        6.887e+04
```

```
multitotalsqft_lm <- lm(sale_price ~ sq_ft_lot + bath_full_count + bedrooms + square_feet_total_living,
multitotalsqft_lm
```

```
##
## Call:
## lm(formula = sale_price ~ sq_ft_lot + bath_full_count + bedrooms +
##     square_feet_total_living, data = housing_df)
##
## Coefficients:
##              (Intercept)                 sq_ft_lot           bath_full_count
##                2.031e+05                 1.048e-01                 4.323e+04
##                 bedrooms  square_feet_total_living
##               -2.432e+04                 1.820e+02
```

```
multisqftprice_lm <- lm(sale_price ~ sq_ft_lot + bath_full_count + bedrooms + square_feet_total_living +
multisqftprice_lm
```

```
##
## Call:
## lm(formula = sale_price ~ sq_ft_lot + bath_full_count + bedrooms +
##     square_feet_total_living + price_per_sq_ft, data = housing_df)
##
## Coefficients:
##              (Intercept)                 sq_ft_lot           bath_full_count
##               -4.721e+05                -1.452e-01                 2.507e+04
##                 bedrooms  square_feet_total_living           price_per_sq_ft
##                1.582e+04                 2.282e+02                 1.670e+03
```

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

summary(price_lm)
```

```
##
## Call:
## lm(formula = sale_price ~ sq_ft_lot, data = housing_df)
##
## 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(zip_lm)
```

```
##
## Call:
## lm(formula = sale_price ~ zip5, data = housing_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -666635 -200429  -67984   87016 3747016
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.406e+09  2.059e+08  -6.831 8.82e-12 ***
## zip5         1.435e+04  2.100e+03   6.834 8.62e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 403700 on 12863 degrees of freedom
## Multiple R-squared:  0.003618,   Adjusted R-squared:  0.00354
## F-statistic: 46.71 on 1 and 12863 DF,  p-value: 8.621e-12
```

```
summary(sqft_lm)
```

```
##
## Call:
## lm(formula = sale_price ~ square_feet_total_living, data = housing_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1800136  -120257   -41547    44028  3811745
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.891e+05  8.745e+03   21.62   <2e-16 ***
## square_feet_total_living 1.857e+02  3.208e+00   57.88   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 360200 on 12863 degrees of freedom
## Multiple R-squared:  0.2066, Adjusted R-squared:  0.2066
## F-statistic:  3351 on 1 and 12863 DF,  p-value: < 2.2e-16
```

```
summary(multi_lm)
```

```
##
## Call:
## lm(formula = sale_price ~ sq_ft_lot + bath_full_count + bedrooms,
##     data = housing_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -3566287  -153218   -51375    69545  3736381
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.429e+05  1.481e+04   9.649   <2e-16 ***
## sq_ft_lot       7.227e-01  5.910e-02  12.229   <2e-16 ***
## bath_full_count 1.458e+05  5.422e+03  26.888   <2e-16 ***
## bedrooms        6.887e+04  4.027e+03  17.099   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 381000 on 12861 degrees of freedom
## Multiple R-squared:  0.1127, Adjusted R-squared:  0.1125
## F-statistic: 544.3 on 3 and 12861 DF,  p-value: < 2.2e-16
```

```
summary(multitotalsqft_lm)
```

```
##
## Call:
## lm(formula = sale_price ~ sq_ft_lot + bath_full_count + bedrooms +
##     square_feet_total_living, data = housing_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1917207  -117118   -40819    44256  3789150
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               2.031e+05  1.404e+04  14.471  < 2e-16 ***
## sq_ft_lot                 1.048e-01  5.776e-02   1.814   0.0697 .
## bath_full_count           4.323e+04  5.708e+03   7.574 3.87e-14 ***
## bedrooms                 -2.432e+04  4.444e+03  -5.473 4.50e-08 ***
## square_feet_total_living  1.820e+02  4.515e+00  40.307  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 359000 on 12860 degrees of freedom
## Multiple R-squared:  0.2122, Adjusted R-squared:  0.2119
## F-statistic: 865.9 on 4 and 12860 DF,  p-value: < 2.2e-16
```

```
summary(multisqftprice_lm)
```

```
##
## Call:
## lm(formula = sale_price ~ sq_ft_lot + bath_full_count + bedrooms +
##     square_feet_total_living + price_per_sq_ft, data = housing_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6789489   -29012     1576    27353  2280000
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)              -4.721e+05  7.782e+03 -60.665  < 2e-16 ***
## sq_ft_lot                -1.452e-01  2.879e-02  -5.045 4.60e-07 ***
## bath_full_count           2.507e+04  2.844e+03   8.816  < 2e-16 ***
## bedrooms                  1.582e+04  2.222e+03   7.121 1.13e-12 ***
## square_feet_total_living  2.282e+02  2.260e+00 100.945  < 2e-16 ***
## price_per_sq_ft           1.670e+03  8.455e+00 197.494  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 178800 on 12859 degrees of freedom
## Multiple R-squared:  0.8047, Adjusted R-squared:  0.8046
## F-statistic: 1.059e+04 on 5 and 12859 DF,  p-value: < 2.2e-16
```

```
summary(price_lm)$r.squared
```

```
## [1] 0.01435497
```

```
summary(price_lm)$adj.r.squared
```

```
## [1] 0.01427835
```

```
summary(zip_lm)$r.squared
```

```
## [1] 0.003617862
```

```
summary(zip_lm)$adj.r.squared
```

```
## [1] 0.003540401
```

```
summary(sqft_lm)$r.squared
```

```
## [1] 0.2066499
```

```
summary(sqft_lm)$adj.r.squared
```

```
## [1] 0.2065882
```

```
summary(multi_lm)$r.squared
```

```
## [1] 0.1126625
```

```
summary(multi_lm)$adj.r.squared
```

```
## [1] 0.1124555
```

```
summary(multitotalsqft_lm)$r.squared
```

```
## [1] 0.2121903
```

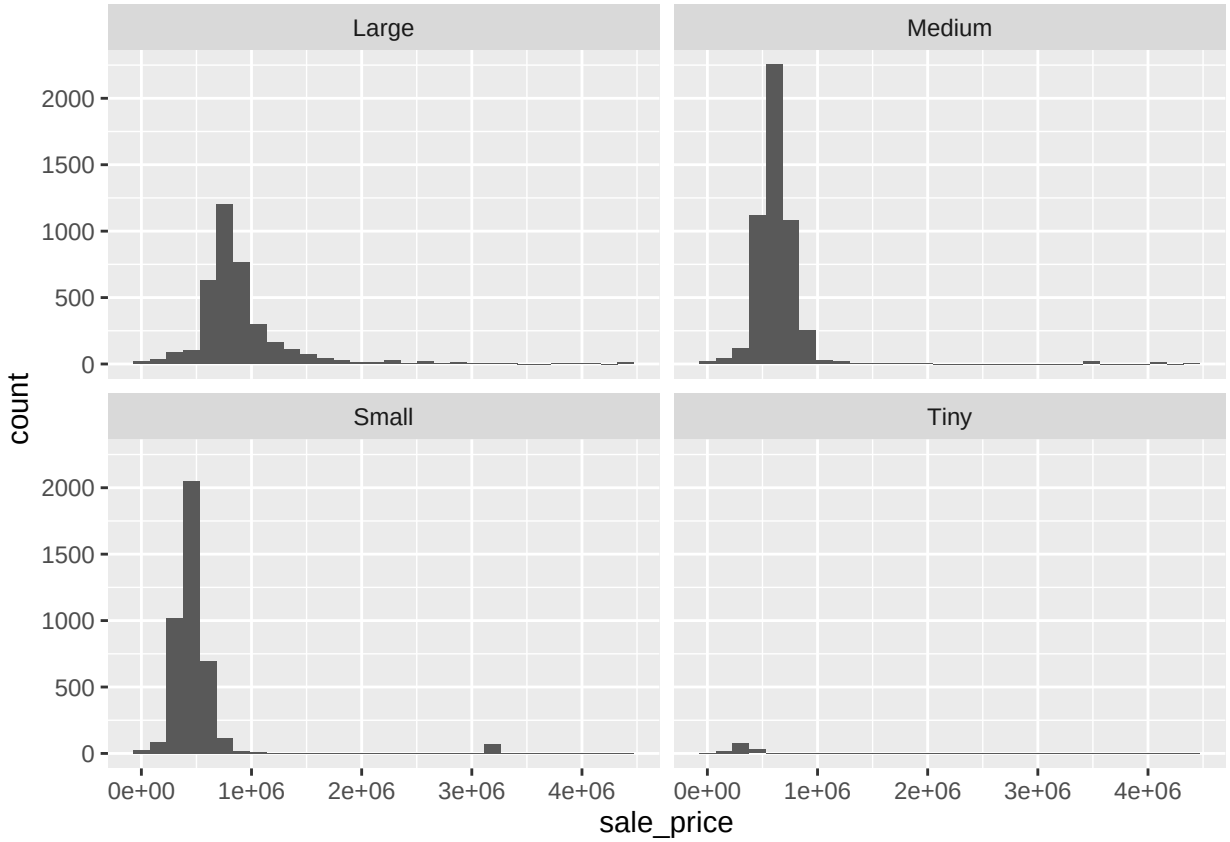
```
summary(multitotalsqft_lm)$adj.r.squared
```

```
## [1] 0.2119453
```

```
price_summary <- housing_df |> group_by(housing_size) |> summarize(mean_by_size=mean(sale_price))
price_summary
```

```
## # A tibble: 4 x 2
##   housing_size mean_by_size
##   <chr>               <dbl>
## 1 Large             890808.
## 2 Medium            636017.
## 3 Small             495161.
## 4 Tiny              432586.
```

```
ggplot(housing_df,aes(sale_price)) + geom_histogram(bins=30) + facet_wrap(vars(housing_size))
```



```
#Oberservation: From r2 and adjusted r2, there is none of the models performed well. However, some perf

#Out of above 2, I see square_feet_total_living (0.206 for sqft_lm) performs significantly well compare

#multisqft_lm is performing well with r^2 as .2121 and adj.r^2 as .219

# 4. Considering the parameters of the multiple regression model you have created. What are the standar

library(lm.beta)

models <- list(price_lm, zip_lm, sqft_lm, multi_lm, multisqftprice_lm, multitotalsqft_lm)

sapply(models,lm.beta)
```

```
##                            [,1]                   [,2]
## coefficients               numeric,2              numeric,2
## residuals                  numeric,12865          numeric,12865
## effects                    numeric,12865          numeric,12865
## rank                       2                      2
## fitted.values              numeric,12865          numeric,12865
## assign                     integer,2              integer,2
## qr                         qr,5                   qr,5
## df.residual                12863                  12863
## xlevels                    list,0                 list,0
## call                       expression             expression
## terms                      sale_price ~ sq_ft_lot sale_price ~ zip5
## model                      data.frame,2           data.frame,2
## standardized.coefficients numeric,2               numeric,2
##                            [,3]
## coefficients               numeric,2
## residuals                  numeric,12865
## effects                    numeric,12865
## rank                       2
## fitted.values              numeric,12865
## assign                     integer,2
## qr                         qr,5
## df.residual                12863
## xlevels                    list,0
## call                       expression
## terms                      sale_price ~ square_feet_total_living
## model                      data.frame,2
## standardized.coefficients numeric,2
##                            [,4]
## coefficients               numeric,4
## residuals                  numeric,12865
## effects                    numeric,12865
## rank                       4
## fitted.values              numeric,12865
## assign                     integer,4
## qr                         qr,5
## df.residual                12861
## xlevels                    list,0
## call                       expression
```

```
## terms                       sale_price ~ sq_ft_lot + bath_full_count + bedrooms
## model                       data.frame,4
## standardized.coefficients numeric,4
##                             [,5]
## coefficients                numeric,6
## residuals                   numeric,12865
## effects                     numeric,12865
## rank                        6
## fitted.values               numeric,12865
## assign                      integer,6
## qr                          qr,5
## df.residual                 12859
## xlevels                     list,0
## call                        expression
## terms                       terms,3
## model                       data.frame,6
## standardized.coefficients numeric,6
##                             [,6]
## coefficients                numeric,5
## residuals                   numeric,12865
## effects                     numeric,12865
## rank                        5
## fitted.values               numeric,12865
## assign                      integer,5
## qr                          qr,5
## df.residual                 12860
## xlevels                     list,0
## call                        expression
## terms                       sale_price ~ sq_ft_lot + bath_full_count + bedrooms + square_feet_total_li
## model                       data.frame,5
## standardized.coefficients numeric,5
```

```
print("Multisq_lm betas")
```

```
## [1] "Multisq_lm betas"
```

```
print(lm.beta(multisqftprice_lm))
```

```
##
## Call:
## lm(formula = sale_price ~ sq_ft_lot + bath_full_count + bedrooms +
##     square_feet_total_living + price_per_sq_ft, data = housing_df)
##
## Standardized Coefficients::
##              (Intercept)                sq_ft_lot          bath_full_count
##               0.00000000              -0.02044829               0.04034341
##                 bedrooms square_feet_total_living          price_per_sq_ft
##               0.03428435               0.55848919               0.78689830
```

```
print(lm.beta(multitotalsqft_lm))
```

```
##
```

```
## Call:
## lm(formula = sale_price ~ sq_ft_lot + bath_full_count + bedrooms +
##     square_feet_total_living, data = housing_df)
##
## Standardized Coefficients::
##              (Intercept)               sq_ft_lot          bath_full_count
##               0.00000000              0.01475150              0.06957036
##                 bedrooms square_feet_total_living
##              -0.05269668              0.44543067
```

```
#Observation: The standardized betas are the coefficient printed above. The values indicate that for ev

# 5. Calculate the confidence intervals for the parameters in your model and explain what the results i

confint(multisqftprice_lm)
```

```
##                                 2.5 %        97.5 %
## (Intercept)              -4.873156e+05 -4.568098e+05
## sq_ft_lot                -2.016695e-01 -8.880732e-02
## bath_full_count           1.949407e+04  3.064176e+04
## bedrooms                  1.146838e+04  2.017985e+04
## square_feet_total_living  2.237352e+02  2.325962e+02
## price_per_sq_ft           1.653205e+03  1.686351e+03
```

```
min(housing_df$sale_price)
```

```
## [1] 698
```

```
max(housing_df$sale_price)
```

```
## [1] 4400000
```

```
# Observation: Square feet total living has the best margin of error, indicating that it is the best pr

# 6.Assess the improvement of the new model compared to your original model (simple regression model) b

anova(price_lm, multisqftprice_lm)
```

```
## Analysis of Variance Table
##
## Model 1: sale_price ~ sq_ft_lot
## Model 2: sale_price ~ sq_ft_lot + bath_full_count + bedrooms + square_feet_total_living +
##     price_per_sq_ft
##   Res.Df        RSS Df  Sum of Sq     F    Pr(>F)
## 1  12863 2.0734e+15
## 2  12859 4.1089e+14  4 1.6625e+15 13007 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#Observation: With an F score of 13007 and p-value of 2.2e-16 (much less than 0), the changes made by t

# 7. Perform case wise diagnostics to identify outliers and/or influential cases, storing each function

residuals <- resid(multisqftprice_lm)
stand_res <- rstandard(multisqftprice_lm)
student_res <- rstudent(multisqftprice_lm)
cooks_distance <- cooks.distance(multisqftprice_lm)
dfbeta <- dfbeta(multisqftprice_lm)
dffit <- dffits(multisqftprice_lm)
leverage <- hatvalues(multisqftprice_lm)
covariance_ratios <- covratio(multisqftprice_lm)

diag_df <- data.frame(residuals, stand_res, student_res, cooks_distance, dfbeta, dffit, leverage, covari

# 8. Calculate the standardized residuals using the appropriate command, specifying those that are +-2,

diag_df$large.residual <- diag_df$stand_res > 2 | diag_df$stand_res < -2

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

sum(diag_df$large.residual)
```

```
## [1] 356
```

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

colnames(diag_df)
```

```
##  [1] "residuals"                "stand_res"
##  [3] "student_res"              "cooks_distance"
##  [5] "X.Intercept."             "sq_ft_lot"
##  [7] "bath_full_count"          "bedrooms"
##  [9] "square_feet_total_living" "price_per_sq_ft"
## [11] "dffit"                    "leverage"
## [13] "covariance_ratios"        "large.residual"
```

```
lrg_residuals <- diag_df[diag_df$large.residual,]
print(nrow(lrg_residuals))
```

```
## [1] 356
```

```
#Observation: Only 356 records returned as True

# 11.Investigate further by calculating the leverage, cooks distance, and covariance rations. Comment o

nrow(lrg_residuals)/nrow(housing_df)
```

```
## [1] 0.02767198
```

```r
#Only 2.7% cases have absolute value
problem <- lrg_residuals[, c("cooks_distance","leverage","covariance_ratios")]
print(nrow(problem))
```

```
## [1] 356
```

```r
print(nrow(problem))
```

```
## [1] 356
```

```r
cooks_check <- problem %>% filter(cooks_distance>1)
nrow(cooks_check)
```

```
## [1] 3
```

```r
# Out of 356 rows,3 rows are having cooks distance greater than 1 which indicates influencial values. T

#Next we will check for leverage
k = 5
n = nrow(housing_df)
avg_leverage = (k+1)/n


threshold_1 <- avg_leverage*2
threshold_2 <- avg_leverage*3

r_threshold_1 <- problem[problem$leverage > threshold_1,]
r_threshold_2 <- problem[problem$leverage > threshold_2,]

print(nrow(r_threshold_1))
```

```
## [1] 246
```

```r
print(nrow(r_threshold_2))
```

```
## [1] 181
```

```r
#Observation,
#Of 356 rows, 246 are greater than double the average leverage and 181 are greater than 3 times of aver

#Covariance ratios
u_cvr <- 1 + threshold_2
l_cvr <- 1 - threshold_2

r_u_cvr <- problem[problem$covariance_ratios > u_cvr,]
r_l_cvr <- problem[problem$covariance_ratios > l_cvr,]

print(nrow((r_u_cvr)))
```

```
## [1] 31
```

```
print(nrow((r_l_cvr)))
```

```
## [1] 117
```

```
#Observation: There are 31 cases greater than upper covaraince and 117 rows greater than lower limits

# 12. Perform the necessary calculations to assess the assumption of independence and state if the cond

library('car')
```

```
## Loading required package: carData

##
## Attaching package: 'car'

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

```
dwt(multisqftprice_lm)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.3066934      1.386602       0
##  Alternative hypothesis: rho != 0
```

```
#The condition is somewhat met since statistic value is 1.3 which is more or less greater than 1 and cl

# 13. Perform the necessary calculations to assess the assumption of no multicollinearity and state if

print("VIF")
```

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

```
vif(multisqftprice_lm)
```

```
##                sq_ft_lot          bath_full_count                 bedrooms
##                 1.081545                 1.378717                 1.525921
## square_feet_total_living          price_per_sq_ft
##                 2.015071                 1.045117
```

```
print("Tolerance")
```

```
## [1] "Tolerance"
```

```
1/vif(multisqftprice_lm)
```

```
##                sq_ft_lot          bath_full_count                 bedrooms
##                0.9246032                0.7253122                0.6553418
## square_feet_total_living          price_per_sq_ft
##                0.4962604                0.9568304
```

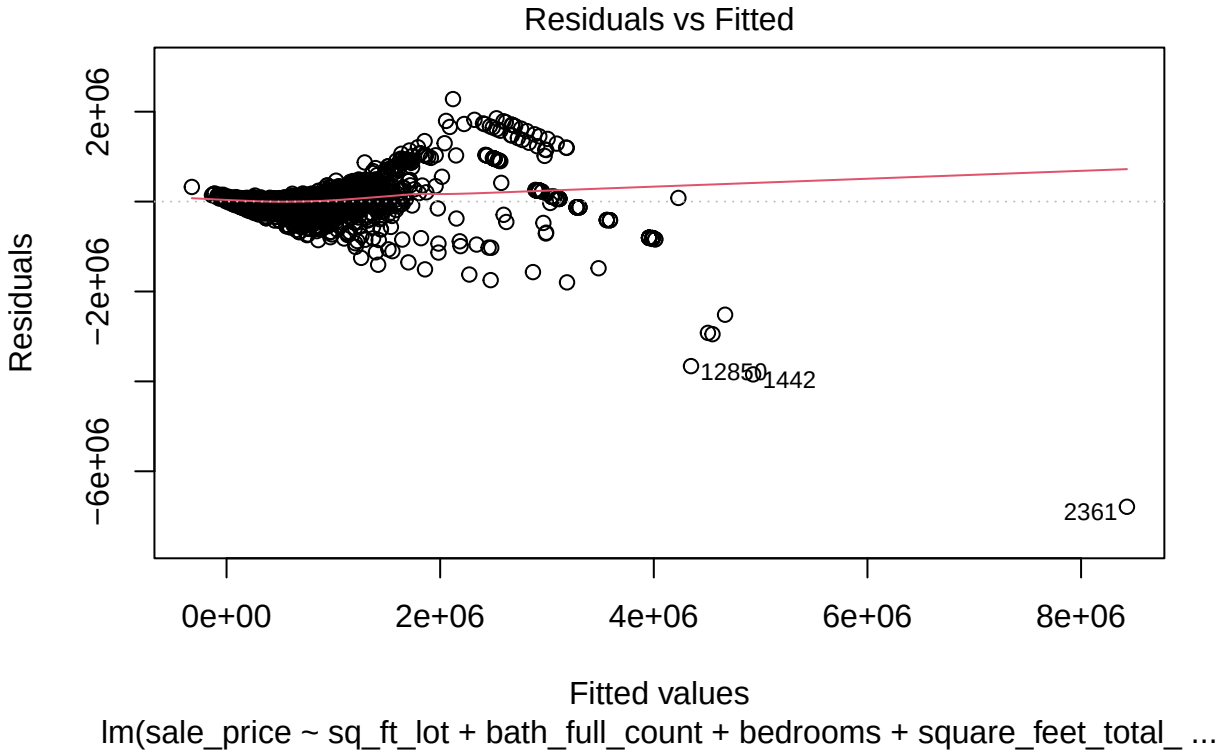
```
print("Average VIF")
```

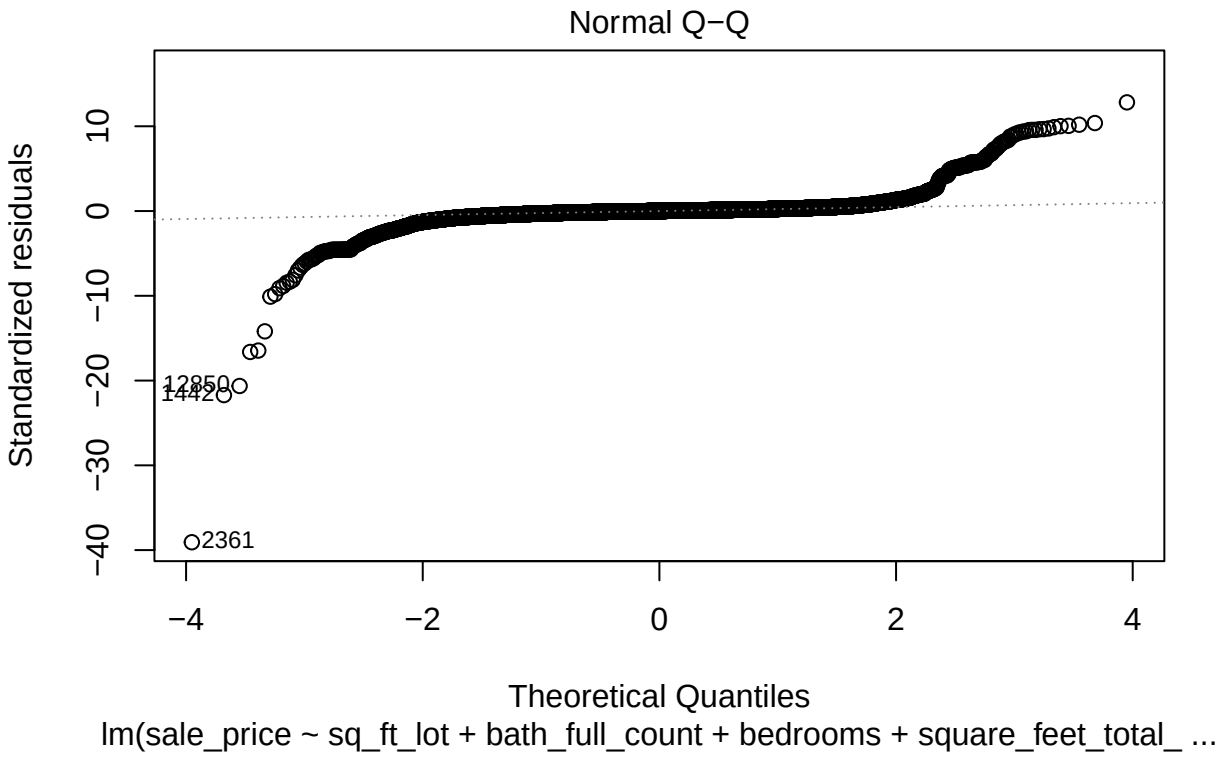
```
## [1] "Average VIF"
```

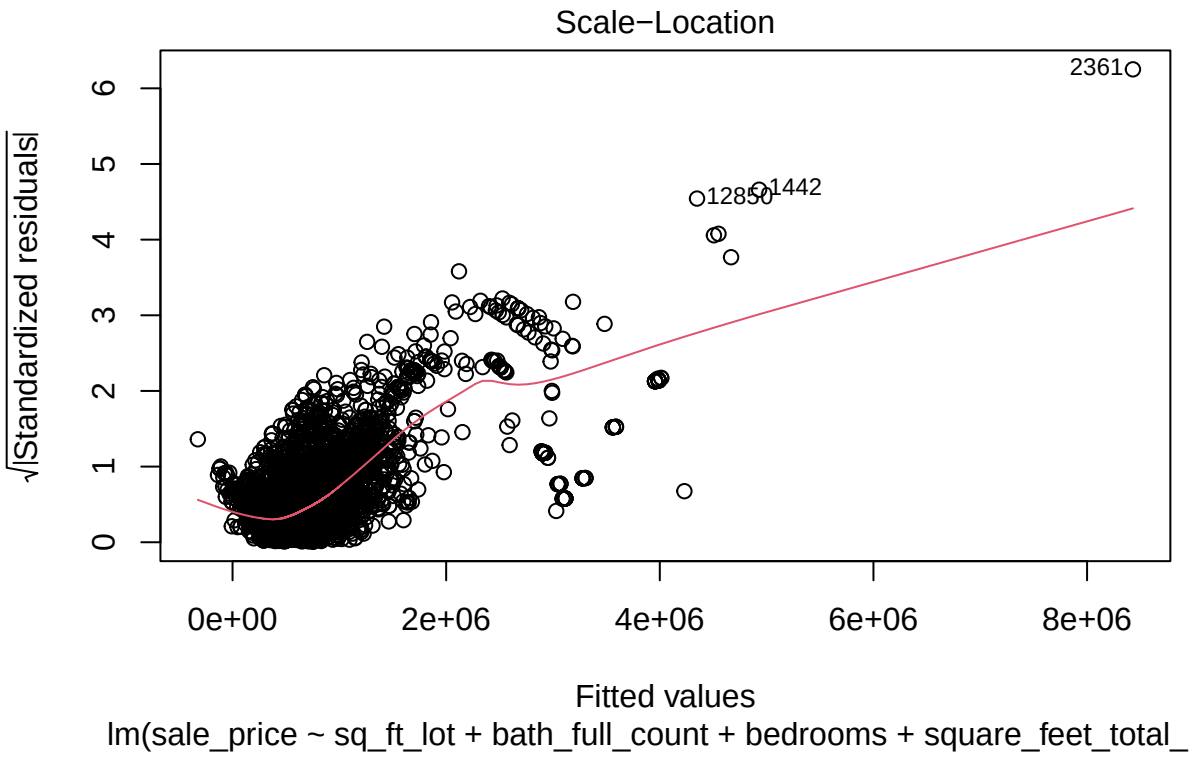
```
mean(vif(multisqftprice_lm))
```

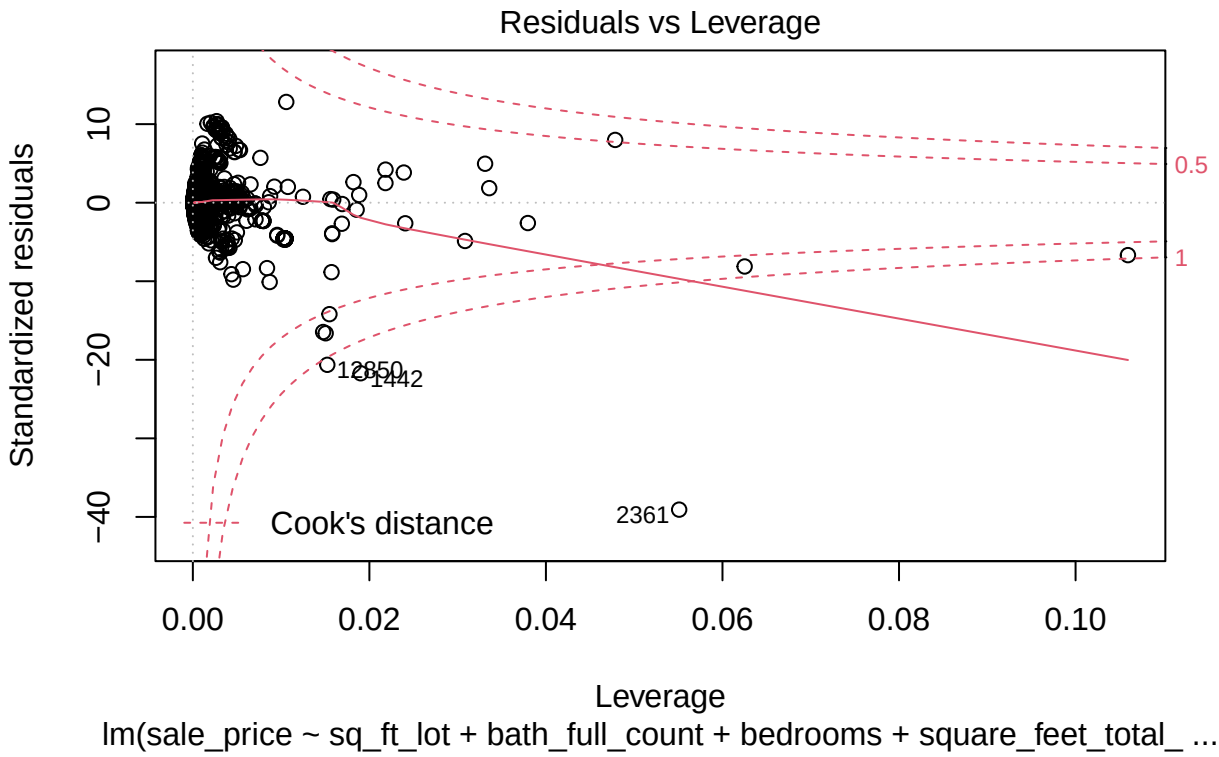
```
## [1] 1.409274
```

```
#Oberservation: The VIF values are all below 10. The tolerance statistic are above .2 and average is sl

# 14. Visually check the assumptions related to the residuals using the plot() and hist() functions. Su

plot(multisqftprice_lm)
```

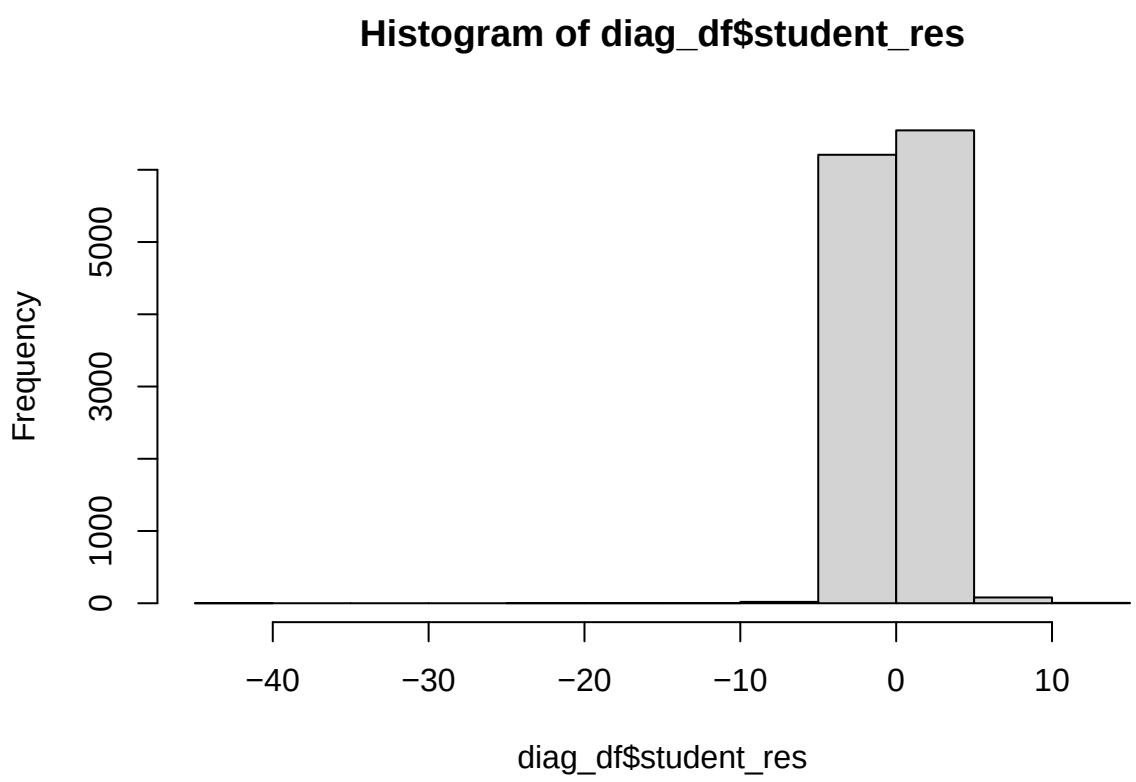



Normal Q−Q
Standardized residuals
10
0
−10
−20
−30
−40
12850
14420
2361
−4
−2
0
2
4
Theoretical Quantiles
lm(sale_price ~ sq_ft_lot + bath_full_count + bedrooms + square_feet_total_ ...

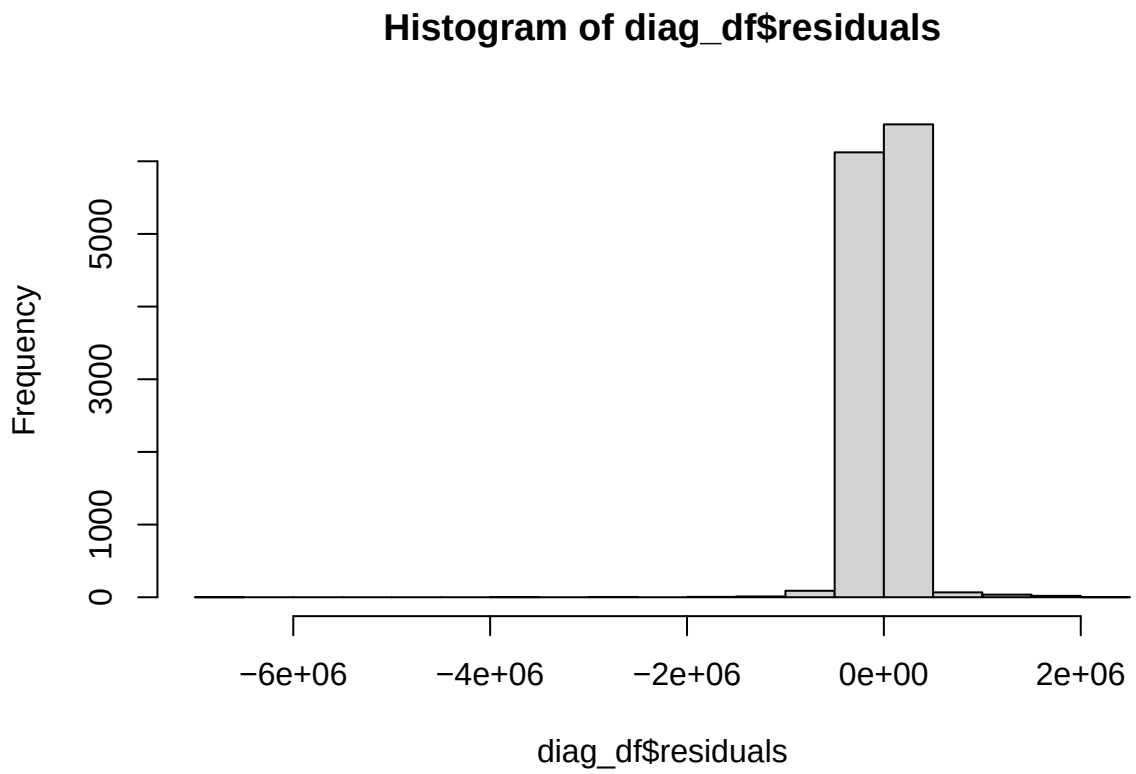

Scale−Location
√|Standardized residuals|
0
1
2
3
4
5
6
0e+00
2e+06
4e+06
6e+06
8e+06
2361
12850
1442
Fitted values
lm(sale_price ~ sq_ft_lot + bath_full_count + bedrooms + square_feet_total_ ...




```
hist(diag_df$student_res)
```




```
hist(diag_df$residuals)
```




```
# 15.Overall, is this regression model unbiased? If an unbiased regression model, what does this tell u

#Observation: Overall, the model is biased with 356 outliers and 427 influential cases.
```