Multiple Linear Regression Stat Project 2

2023-12-06

TODO WYATT NOTES:

#1. Check for outliers and high impact variables (one house has 33 bedrooms) #2. Review why logging variables #3. use automated search procedures for model from existing vars

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.4.4     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.0
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
library(MASS)
```

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

```
library(leaps)
library(faraway)

#Disable scientific notation
options(scipen = 100)


setwd('C:/Users/wyatt/OneDrive/Documents/repos/stat6021-project2')

# Read in Data
data <- read.csv('kc_house_data.csv')


# Remove Data with no bedrooms or bathrooms
data <- subset(data, data$bathroom != 0 | data$bedrooms != 0)

# View headers
head(data, 3)
```

```
##           id            date  price bedrooms bathrooms sqft_living sqft_lot
## 1 7129300520 20141013T000000 221900        3      1.00        1180     5650
## 2 6414100192 20141209T000000 538000        3      2.25        2570     7242
## 3 5631500400 20150225T000000 180000        2      1.00         770    10000
##   floors waterfront view condition grade sqft_above sqft_basement yr_built
## 1      1          0    0         3     7       1180             0     1955
## 2      2          0    0         3     7       2170           400     1951
## 3      1          0    0         3     6        770             0     1933
##   yr_renovated zipcode     lat     long sqft_living15 sqft_lot15
## 1            0   98178 47.5112 -122.257          1340       5650
## 2         1991   98125 47.7210 -122.319          1690       7639
## 3            0   98028 47.7379 -122.233          2720       8062
```

```
colnames(data)
```

```
##  [1] "id"            "date"          "price"         "bedrooms"
##  [5] "bathrooms"     "sqft_living"   "sqft_lot"      "floors"
##  [9] "waterfront"    "view"          "condition"     "grade"
## [13] "sqft_above"    "sqft_basement" "yr_built"      "yr_renovated"
## [17] "zipcode"       "lat"           "long"          "sqft_living15"
## [21] "sqft_lot15"
```

Data Definitions AND Summary Stats

DON'T WANT FOR ANALYSIS

Only have 2 years - should not have a big impact on price of houses within the market with this small of a time frame

```
years_sold <- data$date %>%
  substr(1, 4)

table(years_sold)
```

```
## years_sold
##  2014  2015
## 14629  6977
```

RESPONSE VARIABLE ^ NEED :)

Potentiall take price in ’000s

```
summary(data$price/1000)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    75.0   322.0   450.0   540.1   645.0  7700.0
```

KEEP IN REGRESSION

```
table(data$bedrooms)
```

```
##
##    0    1    2    3    4    5    6    7    8    9   10   11   33
##    6  199 2760 9824 6882 1601  272   38   13    6    3    1    1
```

KEEP IN REGRESSION

```
table(data$bathrooms)
```

```
##
##    0  0.5 0.75    1 1.25  1.5 1.75    2 2.25  2.5 2.75    3 3.25  3.5 3.75    4
##    3    4   72 3852    9 1446 3048 1930 2047 5380 1185  753  589  731  155  136
## 4.25  4.5 4.75    5 5.25  5.5 5.75    6 6.25  6.5 6.75  7.5 7.75    8
##   79  100   23   21   13   10    4    6    2    2    2    1    1    2
```

POTENTIALLY KEEP IN REGRESSION

```
summary(data$sqft_living)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     370    1428    1910    2080    2550   13540
```

POTENTIALLY KEEP IN REGRESSION

```
summary(data$sqft_lot)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     520    5040    7619   15108   10688 1651359
```

```
summary(data$sqft_above)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     370    1190    1560    1788    2210    9410
```

```
summary(data$sqft_basement)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0     0.0   291.6   560.0  4820.0
```

#Potentially keep in regression - probably not though

```
table(data$floors)
```

```
##
##     1   1.5     2   2.5     3   3.5
## 10678  1910  8238   161   612     7
```

proabbly drop,not big sample size difference - just keep view

```
table(data$waterfront)
```

```
##
##     0     1
## 21443   163
```

Maybe just keep view though

```
table(data$view)
```

```
##
##     0     1     2     3     4
## 19484   332   961   510   319
```

definately keep condition

```
table(data$condition)
```

```
##
##     1     2     3     4     5
##    29   172 14026  5678  1701
```

potentially keep grade

CREATEINTO BUCKETS

```
table(data$grade)
```

```
##
##    3    4    5    6    7    8    9   10   11   12   13
##    3   29  242 2038 8978 6066 2615 1134  399   89   13
```

Potentially Keep

```
table(data$yr_built)
```

```
##
## 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915
##   87   29   27   46   45   74   92   65   86   94  134   73   79   58   54   64
## 1916 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930 1931
##   79   56  120   88   98   76   95   84  139  165  180  115  126  114   90   61
## 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946 1947
##   38   30   21   24   40   68   52  106  156  161  223  170  140   95  126  263
## 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963
##  235  195  250  229  220  223  305  271  198  198  224  334  248  224  312  255
## 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979
##  172  187  250  350  381  280  132  104  149  149  162  189  253  417  387  343
## 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995
##  240  199  105  212  229  228  215  294  270  290  317  224  198  202  249  169
## 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011
##  194  177  239  265  218  305  222  422  433  450  453  417  367  230  143  130
## 2012 2013 2014 2015
##  170  201  559   38
```

Probably don’t keep year renovated OR convert to dummy variable for Renovated or Not

```
table(data$yr_renovated)
```

```
##
##     0  1934  1940  1944  1945  1946  1948  1950  1951  1953  1954  1955  1956
## 20692     1     2     1     3     2     1     2     1     3     1     3     3
##  1957  1958  1959  1960  1962  1963  1964  1965  1967  1968  1969  1970  1971
##     3     5     1     4     2     4     5     5     2     8     4     9     2
##  1972  1973  1974  1975  1976  1977  1978  1979  1980  1981  1982  1983  1984
##     4     5     3     6     3     8     6    10    11     5    11    18    18
##  1985  1986  1987  1988  1989  1990  1991  1992  1993  1994  1995  1996  1997
##    17    17    18    15    22    25    20    17    19    19    16    15    15
##  1998  1999  2000  2001  2002  2003  2004  2005  2006  2007  2008  2009  2010
##    19    17    35    19    22    36    26    35    24    35    18    22    18
##  2011  2012  2013  2014  2015
##    13    11    37    91    16
```

NEIGHBORHOOD VARIABLES

Geo spatial regression too complicated for this analysis

```
table(data$zipcode)
```

```
##
## 98001 98002 98003 98004 98005 98006 98007 98008 98010 98011 98014 98019 98022
##   361   199   280   317   168   498   141   283   100   195   124   190   234
## 98023 98024 98027 98028 98029 98030 98031 98032 98033 98034 98038 98039 98040
##   499    80   412   283   321   256   273   125   432   545   590    50   282
## 98042 98045 98052 98053 98055 98056 98058 98059 98065 98070 98072 98074 98075
##   548   221   574   404   268   406   455   468   309   118   273   441   359
## 98077 98092 98102 98103 98105 98106 98107 98108 98109 98112 98115 98116 98117
##   198   351   104   602   229   335   266   186   109   269   583   330   553
## 98118 98119 98122 98125 98126 98133 98136 98144 98146 98148 98155 98166 98168
##   508   184   290   410   354   493   263   343   288    57   446   254   269
## 98177 98178 98188 98198 98199
##   255   262   136   280   317
```

DONT NEED LAT OR LONG

DONT NEED LAT OR LONG

```
summary(data$sqft_living15)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     399    1490    1840    1986    2360    6210
```

```
summary(data$sqft_lot15)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     651    5100    7620   12768   10083  871200
```

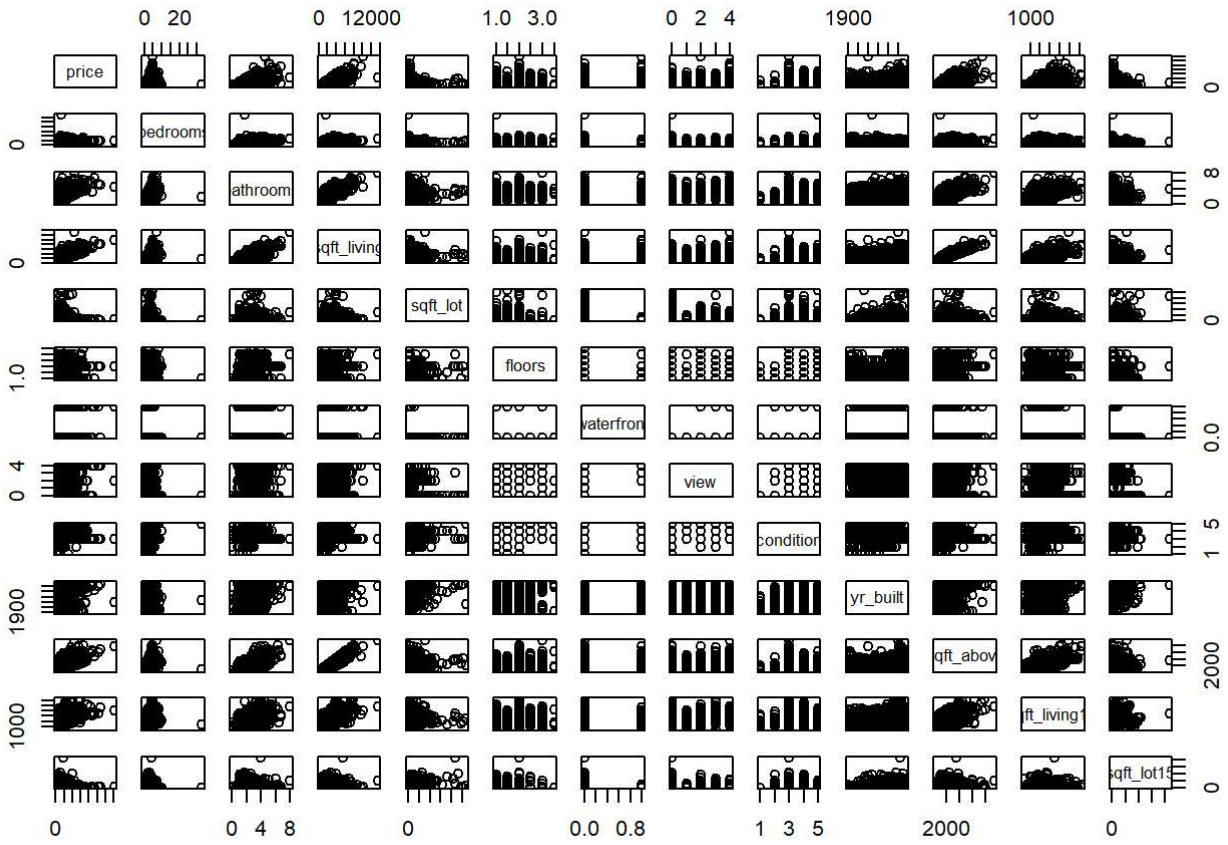
SUBSECT INITIAL DATA REVIEW

Create new column for renovated boolean and grade category

```
# Add renovated boolean
data$renovated <- as.factor(ifelse(data$yr_renovated == 0, 0,1))

# Create Quality buckets FROM GRADE ##NEW VARIABLE
data$grade <- cut(as.numeric(data$grade), breaks = c(0, 3, 10, 14), right=FALSE, labels = c('sub
par', 'average', 'high quality'))
data$grade <- relevel(data$grade, "average")


#Drop ID - ID only differentiates rows so not needed
# Date - there are only two years of data here so not much variation for time series
# Zipcode - omit regional data to simplify regression
# lat, long -omit specific location to simplify regression
# sqft_basement - omit since NA - linearly related to other variables
#TRANSFORM GRADE, yr_built, #TRANSFORM yr Renovated,
data <- data %>%
             dplyr::select(price,
                           bedrooms,
                           bathrooms,
                           sqft_living,
                           sqft_lot,
                           floors,
                           waterfront,
                           view,
                           condition,
                           grade,
                           yr_built,
                           sqft_above,
                           renovated,
                           sqft_living15,
                           sqft_lot15
                           )

# Create TEST/TRAIN Split
set.seed(6021)
sample.data<-sample.int(nrow(data), floor(.50*nrow(data)), replace = F)


train<-data[sample.data, ]
rownames(train) <- NULL
test<-data[-sample.data, ]
rownames(test) <- NULL

# Get correlation coefficients
round(cor(train[,unlist(lapply(train, is.numeric), use.names = FALSE)]), 3)
```

```
##               price bedrooms bathrooms sqft_living sqft_lot floors waterfront
## price         1.000    0.306     0.529       0.699    0.093  0.261      0.242
## bedrooms      0.306    1.000     0.505       0.566    0.035  0.180     -0.014
## bathrooms     0.529    0.505     1.000       0.760    0.099  0.503      0.050
## sqft_living   0.699    0.566     0.760       1.000    0.192  0.354      0.085
## sqft_lot      0.093    0.035     0.099       0.192    1.000  0.004      0.012
## floors        0.261    0.180     0.503       0.354    0.004  1.000      0.018
## waterfront    0.242   -0.014     0.050       0.085    0.012  0.018      1.000
## view          0.386    0.087     0.199       0.281    0.064  0.037      0.400
## condition     0.030    0.038    -0.116      -0.056   -0.005 -0.256      0.025
## yr_built      0.058    0.142     0.501       0.315    0.061  0.486     -0.037
## sqft_above    0.610    0.474     0.691       0.878    0.199  0.524      0.061
## sqft_living15 0.581    0.388     0.575       0.750    0.156  0.284      0.078
## sqft_lot15    0.078    0.028     0.086       0.186    0.715 -0.006      0.023
##                 view condition yr_built sqft_above sqft_living15 sqft_lot15
## price          0.386     0.030    0.058      0.610         0.581      0.078
## bedrooms       0.087     0.038    0.142      0.474         0.388      0.028
## bathrooms      0.199    -0.116    0.501      0.691         0.575      0.086
## sqft_living    0.281    -0.056    0.315      0.878         0.750      0.186
## sqft_lot       0.064    -0.005    0.061      0.199         0.156      0.715
## floors         0.037    -0.256    0.486      0.524         0.284     -0.006
## waterfront     0.400     0.025   -0.037      0.061         0.078      0.023
## view           1.000     0.041   -0.059      0.167         0.277      0.072
## condition      0.041     1.000   -0.355     -0.151        -0.094      0.002
## yr_built      -0.059    -0.355    1.000      0.423         0.329      0.073
## sqft_above     0.167    -0.151    0.423      1.000         0.729      0.193
## sqft_living15  0.277    -0.094    0.329      0.729         1.000      0.193
## sqft_lot15     0.072     0.002    0.073      0.193         0.193      1.000
```

```
# start visualization
pairs(train[,unlist(lapply(train, is.numeric), use.names = FALSE)])
```




```
# Floors, waterfront, views, and conditions are categorical variables
```

Convert Categorical Variables to factors

```
train$floors <- as.factor(train$floors)
train$view <- as.factor(train$view)
train$condition <- as.factor(train$condition)
train$waterfront <- as.factor(train$waterfront)
```

```
# Run initial regression
# Looking at price in '000s
results <- lm(price/1000~., data=train)
summary(results)
```

```
##
## Call:
## lm(formula = price/1000 ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1315.7  -118.5   -14.1    96.4  3892.9
##
## Coefficients:
##                       Estimate    Std. Error t value             Pr(>|t|)
## (Intercept)       5456.35658239  227.38635620  23.996 < 0.0000000000000002 ***
## bedrooms           -37.93340984    2.89658644 -13.096 < 0.0000000000000002 ***
## bathrooms           57.83597890    5.12991734  11.274 < 0.0000000000000002 ***
## sqft_living          0.18494236    0.00671316  27.549 < 0.0000000000000002 ***
## sqft_lot            -0.00008343    0.00007467  -1.117             0.263884
## floors1.5           10.47679798    8.45315412   1.239             0.215226
## floors2             16.10183497    6.81635065   2.362             0.018183 *
## floors2.5          125.47974160   24.51701389   5.118    0.000000314000935 ***
## floors3            196.00222166   14.45397970  13.560 < 0.0000000000000002 ***
## floors3.5          354.78209090   99.87069522   3.552             0.000383 ***
## waterfront1        459.17050239   31.41412382  14.617 < 0.0000000000000002 ***
## view1              130.13706486   17.80733120   7.308    0.000000000000290 ***
## view2               76.97703427   10.82776662   7.109    0.000000000001241 ***
## view3              104.27249924   14.44750982   7.217    0.000000000000566 ***
## view4              241.50294774   22.51538887  10.726 < 0.0000000000000002 ***
## condition2         -55.36930002   71.59798136  -0.773             0.439340
## condition3         -23.41352863   67.38049085  -0.347             0.728236
## condition4         -14.94596373   67.38739341  -0.222             0.824480
## condition5          30.60490182   67.71515380   0.452             0.651303
## gradehigh quality  278.21218642   10.17956302  27.330 < 0.0000000000000002 ***
## yr_built            -2.77799655    0.11331934 -24.515 < 0.0000000000000002 ***
## sqft_above           0.01272591    0.00692684   1.837             0.066210 .
## renovated1          24.89703240   11.32868654   2.198             0.027992 *
## sqft_living15        0.06765579    0.00510730  13.247 < 0.0000000000000002 ***
## sqft_lot15          -0.00068113    0.00011198  -6.083    0.000000001222777 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 222.8 on 10778 degrees of freedom
## Multiple R-squared:  0.6294, Adjusted R-squared:  0.6286
## F-statistic: 762.7 on 24 and 10778 DF,  p-value: < 0.00000000000000022
```

MULTICOLLinearity

sqft living and sqft above both indicate multicollinearity. drop sqft_above as sqft_living

is more relevant to home price. condition has high multi-collinearity, drop from model

```
faraway::vif(results)
```

```
##          bedrooms         bathrooms       sqft_living          sqft_lot
##          1.629912          3.427313          8.318360          2.087336
##         floors1.5           floors2         floors2.5           floors3
##          1.275943          2.382799          1.069060          1.243718
##         floors3.5       waterfront1             view1             view2
##          1.004419          1.598586          1.025775          1.068476
##             view3             view4        condition2        condition3
##          1.083215          1.649791          8.405742        224.364565
##        condition4        condition5 gradehigh quality          yr_built
##        192.067524         69.933548          1.596345          2.416605
##        sqft_above        renovated1     sqft_living15        sqft_lot15
##          7.176229          1.176700          2.633165          2.099453
```

Re-run regression w/o sqft_above and condition

```
train <- train %>%
                     dplyr::select(price,
                                   bedrooms,
                                   bathrooms,
                                   sqft_living,
                                   sqft_lot,
                                   floors,
                                   waterfront,
                                   view,
                                   grade,
                                   yr_built,
                                   renovated,
                                   sqft_living15,
                                   sqft_lot15
                                   )

# Run initial regression
results <- lm(price/1000~., data=train)
summary(results)
```

```
##
## Call:
## lm(formula = price/1000 ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1341.0  -118.3   -13.4    96.1  3885.4
##
## Coefficients:
##                        Estimate    Std. Error t value             Pr(>|t|)
## (Intercept)       5717.95510047  208.62490804  27.408 < 0.0000000000000002 ***
## bedrooms           -37.38054082    2.89764233 -12.900 < 0.0000000000000002 ***
## bathrooms           59.60313078    5.03571839  11.836 < 0.0000000000000002 ***
## sqft_living          0.19340543    0.00502972  38.453 < 0.0000000000000002 ***
## sqft_lot            -0.00008110    0.00007465  -1.086             0.277303
## floors1.5           13.73469397    8.25703623   1.663             0.096263 .
## floors2             19.31083849    5.98507013   3.227             0.001257 **
## floors2.5          131.15422516   24.22919293   5.413    0.000000063282552 ***
## floors3            198.23003598   14.20572391  13.954 < 0.0000000000000002 ***
## floors3.5          363.84045808   99.97170709   3.639             0.000275 ***
## waterfront1        464.48611615   31.44174198  14.773 < 0.0000000000000002 ***
## view1              129.41874763   17.80314702   7.269    0.000000000000386 ***
## view2               75.10277883   10.79371789   6.958    0.000000000003651 ***
## view3              100.77972703   14.36941325   7.013    0.000000000002464 ***
## view4              237.52571757   22.47135164  10.570 < 0.0000000000000002 ***
## gradehigh quality  279.64620898   10.01818751  27.914 < 0.0000000000000002 ***
## yr_built            -2.92118459    0.10745015 -27.186 < 0.0000000000000002 ***
## renovated1          15.69281327   11.16302441   1.406             0.159817
## sqft_living15        0.06842657    0.00502605  13.614 < 0.0000000000000002 ***
## sqft_lot15          -0.00066714    0.00011206  -5.953    0.000000002708143 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 223.2 on 10783 degrees of freedom
## Multiple R-squared:  0.6279, Adjusted R-squared:  0.6273
## F-statistic: 957.8 on 19 and 10783 DF,  p-value: < 0.00000000000000022
```

#no longer evidence of multicollinearity

```
faraway::vif(results)
```

```
##          bedrooms         bathrooms       sqft_living          sqft_lot
##          1.625388          3.291031          4.653157          2.078901
##         floors1.5           floors2         floors2.5           floors3
##          1.213160          1.830620          1.040450          1.197153
##         floors3.5       waterfront1             view1             view2
##          1.002926          1.595789          1.021702          1.058047
##             view3             view4 gradehigh quality         yr_built
##          1.067783          1.637588          1.540718          2.165149
##        renovated1     sqft_living15        sqft_lot15
##          1.138535          2.541120          2.095113
```

Automated search procedures utilizing stepwise prediction

```
reg_null <- lm(price/1000~1, data=train)

step(reg_null, scope=list(lower=reg_null, upper=results), direction='both')
```

```
## Start:  AIC=127506.3
## price/1000 ~ 1
##
##                 Df Sum of Sq        RSS    AIC
## + sqft_living    1 706056302  737325131 120252
## + sqft_living15  1 487635497  955745936 123055
## + grade          1 445559369  997822064 123520
## + bathrooms      1 403541322 1039840111 123966
## + view           4 224622119 1218759314 125687
## + bedrooms       1 135355411 1308026022 126445
## + floors         5 118003627 1325377805 126595
## + waterfront     1  84797634 1358583798 126854
## + renovated      1  19345674 1424035759 127363
## + sqft_lot       1  12483165 1430898267 127414
## + sqft_lot15     1   8702023 1434679410 127443
## + yr_built       1   4917110 1438464323 127471
## <none>                       1443381433 127506
##
## Step:  AIC=120251.8
## price/1000 ~ sqft_living
##
##                 Df Sum of Sq        RSS    AIC
## + view           4  65209063  672116068 119259
## + grade          1  58084511  679240620 119367
## + waterfront     1  48559514  688765617 119518
## + yr_built       1  42133512  695191619 119618
## + bedrooms       1  17072306  720252825 120001
## + floors         5  14817651  722507480 120042
## + sqft_living15  1  10556296  726768835 120098
## + renovated      1   9732809  727592322 120110
## + sqft_lot15     1   4144482  733180649 120193
## + sqft_lot       1   2587701  734737430 120216
## <none>                       737325131 120252
## + bathrooms      1     24357  737300774 120253
## - sqft_living    1 706056302 1443381433 127506
##
## Step:  AIC=119259.4
## price/1000 ~ sqft_living + view
##
##                 Df Sum of Sq        RSS    AIC
## + grade          1  51844006  620272063 118394
## + yr_built       1  28551566  643564502 118792
## + waterfront     1  12729063  659387006 119055
## + bedrooms       1  11842628  660273441 119069
## + floors         5  11291502  660824566 119086
## + sqft_living15  1   6331189  665784880 119159
## + renovated      1   4743846  667372222 119185
## + sqft_lot15     1   4525049  667591020 119188
## + sqft_lot       1   2616898  669499171 119219
## <none>                       672116068 119259
## + bathrooms      1      2215  672113854 119261
## - view           4  65209063  737325131 120252
```

```
## - sqft_living    1 546643246 1218759314 125687
##
## Step:  AIC=118394.2
## price/1000 ~ sqft_living + view + grade
##
##                 Df Sum of Sq       RSS    AIC
## + yr_built       1  31043482 589228581 117842
## + waterfront     1  12222991 608049072 118181
## + floors         5  11934076 608337987 118194
## + renovated      1   6094746 614177317 118290
## + bedrooms       1   4956016 615316047 118310
## + sqft_lot15     1   4922945 615349118 118310
## + sqft_lot       1   3120731 617151332 118342
## + sqft_living15  1   1903532 618368530 118363
## <none>                       620272063 118394
## + bathrooms      1     65201 620206862 118395
## - grade          1  51844006 672116068 119259
## - view           4  58968557 679240620 119367
## - sqft_living    1 255576977 875849040 122120
##
## Step:  AIC=117841.6
## price/1000 ~ sqft_living + view + grade + yr_built
##
##                 Df Sum of Sq       RSS    AIC
## + floors         5  15255020 573973560 117568
## + waterfront     1  11521773 577706808 117630
## + bathrooms      1   8500011 580728569 117687
## + bedrooms       1   6512856 582715725 117724
## + sqft_living15  1   5452383 583776198 117743
## + sqft_lot15     1   4497978 584730603 117761
## + sqft_lot       1   3104967 586123614 117786
## + renovated      1   1258658 587969923 117820
## <none>                       589228581 117842
## - yr_built       1  31043482 620272063 118394
## - view           4  45666438 634895018 118640
## - grade          1  54335921 643564502 118792
## - sqft_living    1 286025951 875254532 122114
##
## Step:  AIC=117568.2
## price/1000 ~ sqft_living + view + grade + yr_built + floors
##
##                 Df Sum of Sq       RSS    AIC
## + waterfront     1  11557983 562415578 117350
## + sqft_living15  1   7670940 566302620 117425
## + bedrooms       1   6060720 567912840 117456
## + bathrooms      1   4660195 569313366 117482
## + sqft_lot15     1   3177622 570795939 117510
## + sqft_lot       1   2224879 571748681 117528
## + renovated      1    579496 573394064 117559
## <none>                       573973560 117568
## - floors         5  15255020 589228581 117842
## - yr_built       1  34364426 608337987 118194
```

```
## - view           4  43521155 617494715 118350
## - grade          1  52207042 626180602 118507
## - sqft_living    1 253601639 827575199 121519
##
## Step:  AIC=117350.4
## price/1000 ~ sqft_living + view + grade + yr_built + floors +
##     waterfront
##
##                 Df Sum of Sq       RSS    AIC
## + sqft_living15  1   7906446 554509131 117199
## + bedrooms       1   5271608 557143970 117251
## + bathrooms      1   4847952 557567626 117259
## + sqft_lot15     1   3251739 559163838 117290
## + sqft_lot       1   2209989 560205588 117310
## + renovated      1    246860 562168717 117348
## <none>                       562415578 117350
## - waterfront     1  11557983 573973560 117568
## - floors         5  15291231 577706808 117630
## - view           4  16706980 579122558 117659
## - yr_built       1  33529212 595944790 117974
## - grade          1  51697669 614113247 118298
## - sqft_living    1 256215544 818631122 121404
##
## Step:  AIC=117199.5
## price/1000 ~ sqft_living + view + grade + yr_built + floors +
##     waterfront + sqft_living15
##
##                 Df Sum of Sq       RSS    AIC
## + bathrooms      1   5595116 548914015 117092
## + bedrooms       1   4965044 549544087 117104
## + sqft_lot15     1   4104686 550404445 117121
## + sqft_lot       1   2341400 552167731 117156
## + renovated      1    387488 554121643 117194
## <none>                       554509131 117199
## - sqft_living15  1   7906446 562415578 117350
## - waterfront     1  11793489 566302620 117425
## - view           4  13547843 568056974 117452
## - floors         5  17611867 572120998 117527
## - yr_built       1  37257817 591766948 117900
## - grade          1  44190258 598699389 118026
## - sqft_living    1 131037333 685546464 119489
##
## Step:  AIC=117091.9
## price/1000 ~ sqft_living + view + grade + yr_built + floors +
##     waterfront + sqft_living15 + bathrooms
##
##                 Df Sum of Sq       RSS    AIC
## + bedrooms       1   7377630 541536385 116948
## + sqft_lot15     1   3416667 545497348 117026
## + sqft_lot       1   1932182 546981833 117056
## + renovated      1    127830 548786186 117091
## <none>                       548914015 117092
```

```
## - bathrooms      1   5595116 554509131 117199
## - sqft_living15  1   8653611 557567626 117259
## - waterfront     1  12009024 560923039 117324
## - view           4  12396447 561310462 117325
## - floors         5  13695692 562609707 117348
## - yr_built       1  42448417 591362432 117895
## - grade          1  45631137 594545152 117953
## - sqft_living    1  63253327 612167342 118268
##
## Step:  AIC=116947.8
## price/1000 ~ sqft_living + view + grade + yr_built + floors +
##     waterfront + sqft_living15 + bathrooms + bedrooms
##
##                 Df Sum of Sq       RSS    AIC
## + sqft_lot15     1   4355969 537180416 116863
## + sqft_lot       1   2659913 538876472 116897
## <none>                       541536385 116948
## + renovated      1     67997 541468388 116948
## - bedrooms       1   7377630 548914015 117092
## - bathrooms      1   8007702 549544087 117104
## - sqft_living15  1   8430243 549966628 117113
## - view           4  11133546 552669931 117160
## - waterfront     1  11090189 552626574 117165
## - floors         5  12514517 554050902 117185
## - grade          1  39002981 580539366 117697
## - yr_built       1  44225512 585761896 117794
## - sqft_living    1  70626515 612162900 118270
##
## Step:  AIC=116862.5
## price/1000 ~ sqft_living + view + grade + yr_built + floors +
##     waterfront + sqft_living15 + bathrooms + bedrooms + sqft_lot15
##
##                 Df Sum of Sq       RSS    AIC
## <none>                       537180416 116863
## + renovated      1     97586 537082829 116863
## + sqft_lot       1     57951 537122465 116863
## - sqft_lot15     1   4355969 541536385 116948
## - bathrooms      1   7280282 544460697 117006
## - bedrooms       1   8316932 545497348 117026
## - sqft_living15  1   9278083 546458499 117046
## - view           4  11105588 548286004 117076
## - waterfront     1  11116832 548297248 117082
## - floors         5  11785917 548966332 117087
## - grade          1  38671860 575852275 117612
## - yr_built       1  41763783 578944199 117669
## - sqft_living    1  73915491 611095906 118253
```

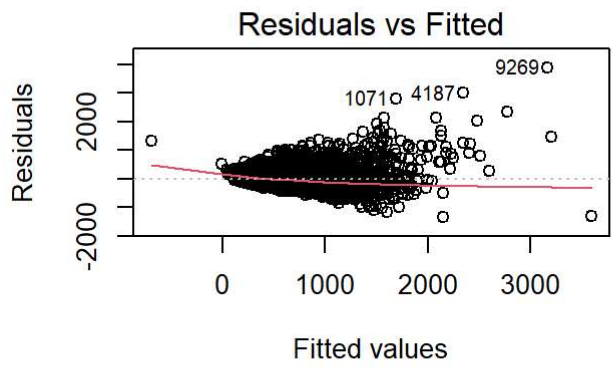
```
##
## Call:
## lm(formula = price/1000 ~ sqft_living + view + grade + yr_built +
##     floors + waterfront + sqft_living15 + bathrooms + bedrooms +
##     sqft_lot15, data = train)
##
## Coefficients:
##       (Intercept)        sqft_living              view1              view2
##      5809.9814218          0.1930497        130.1500693         75.3347891
##             view3              view4  gradehigh quality           yr_built
##       101.4175028        237.9800031        278.9416158         -2.9683239
##         floors1.5            floors2          floors2.5            floors3
##        12.6926791         20.4060739        132.1767814        199.1892131
##         floors3.5        waterfront1      sqft_living15          bathrooms
##       363.9708137        468.2950581          0.0684417         60.4526751
##          bedrooms         sqft_lot15
##       -37.3981705         -0.0007494
```

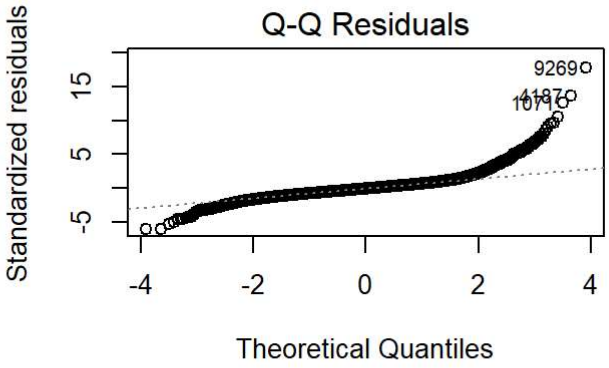
```
# Build model based on results of step-wise automated search procedures
automated_results <- lm(formula = price/1000 ~ sqft_living + view + grade + yr_built +
    floors + waterfront + sqft_living15 + bathrooms + bedrooms +
    sqft_lot15, data = train)

summary(automated_results)
```

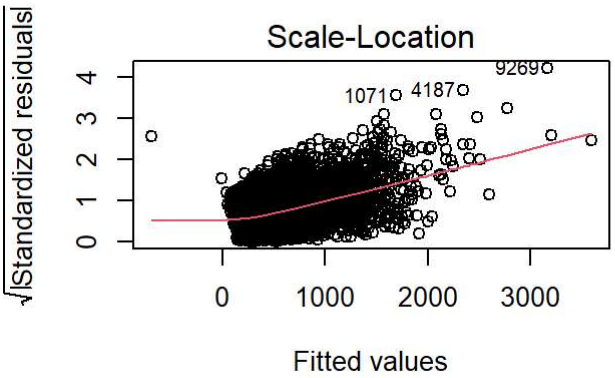
```
##
## Call:
## lm(formula = price/1000 ~ sqft_living + view + grade + yr_built +
##     floors + waterfront + sqft_living15 + bathrooms + bedrooms +
##     sqft_lot15, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1345.3  -118.1   -13.5    96.2  3894.9
##
## Coefficients:
##                        Estimate    Std. Error t value             Pr(>|t|)
## (Intercept)       5809.98142176  198.88665703  29.213 < 0.0000000000000002 ***
## sqft_living          0.19304974    0.00501131  38.523 < 0.0000000000000002 ***
## view1              130.15006932   17.79765729   7.313     0.00000000000028 ***
## view2               75.33478912   10.79235805   6.980     0.00000000000312 ***
## view3              101.41750278   14.36250375   7.061     0.00000000000175 ***
## view4              237.98000312   22.47095789  10.591 < 0.0000000000000002 ***
## gradehigh quality  278.94161578   10.01073108  27.864 < 0.0000000000000002 ***
## yr_built            -2.96832392    0.10250886 -28.957 < 0.0000000000000002 ***
## floors1.5           12.69267910    8.23628426   1.541              0.123329
## floors2             20.40607386    5.94745130   3.431              0.000603 ***
## floors2.5          132.17678140   24.22302449   5.457     0.00000004959065 ***
## floors3            199.18921310   14.19297155  14.034 < 0.0000000000000002 ***
## floors3.5          363.97081367   99.97574201   3.641              0.000273 ***
## waterfront1        468.29505811   31.34577857  14.940 < 0.0000000000000002 ***
## sqft_living15        0.06844173    0.00501466  13.648 < 0.0000000000000002 ***
## bathrooms           60.45267511    5.00024681  12.090 < 0.0000000000000002 ***
## bedrooms           -37.39817046    2.89413498 -12.922 < 0.0000000000000002 ***
## sqft_lot15          -0.00074945    0.00008014  -9.352 < 0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 223.2 on 10785 degrees of freedom
## Multiple R-squared:  0.6278, Adjusted R-squared:  0.6272
## F-statistic:  1070 on 17 and 10785 DF,  p-value: < 0.00000000000000022
```

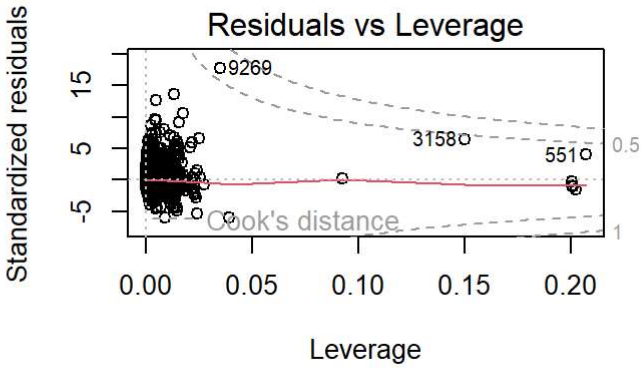
Assess assumptions

```
#assess regression assumptions
par(mfrow=c(2,2))
plot(automated_results)
```



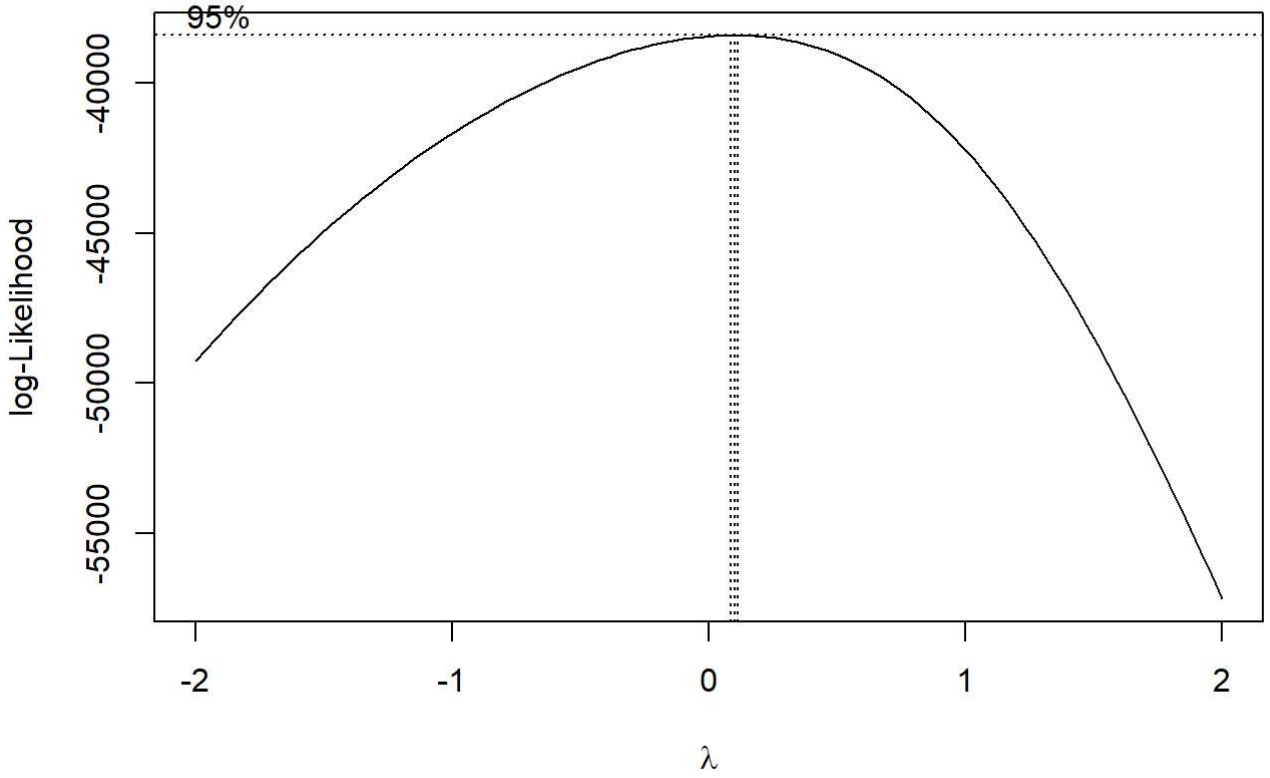







```
# ASSUMPTION 2 are not met so we need to transform the response variable (price)
```

```
# 1 does not lie in the CI AND there is increasing variance from left to right therefore a trans
formation of Y may be needed - we will use log so we can still interpret the estimated coefficie
nts
boxcox(automated_results)
```



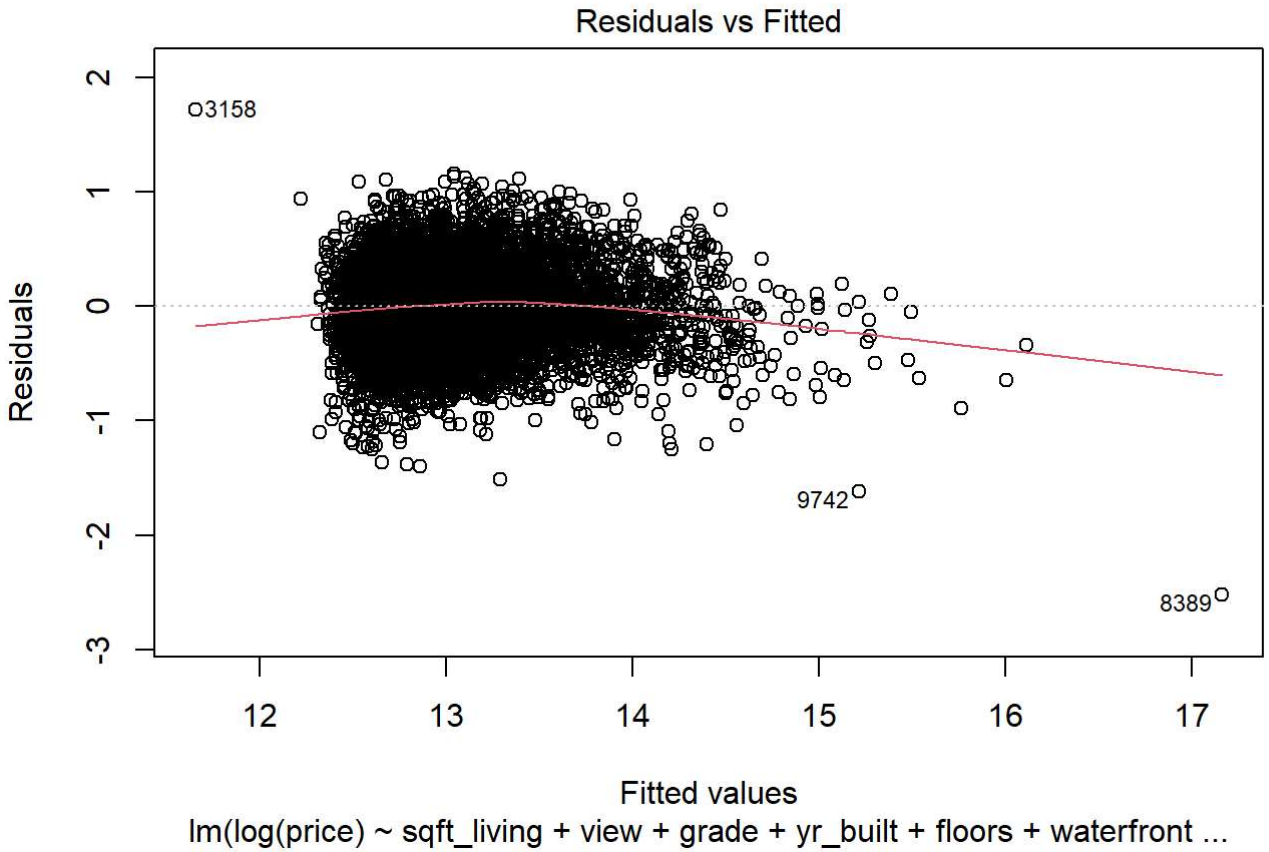
Re-run regression with transformed Y

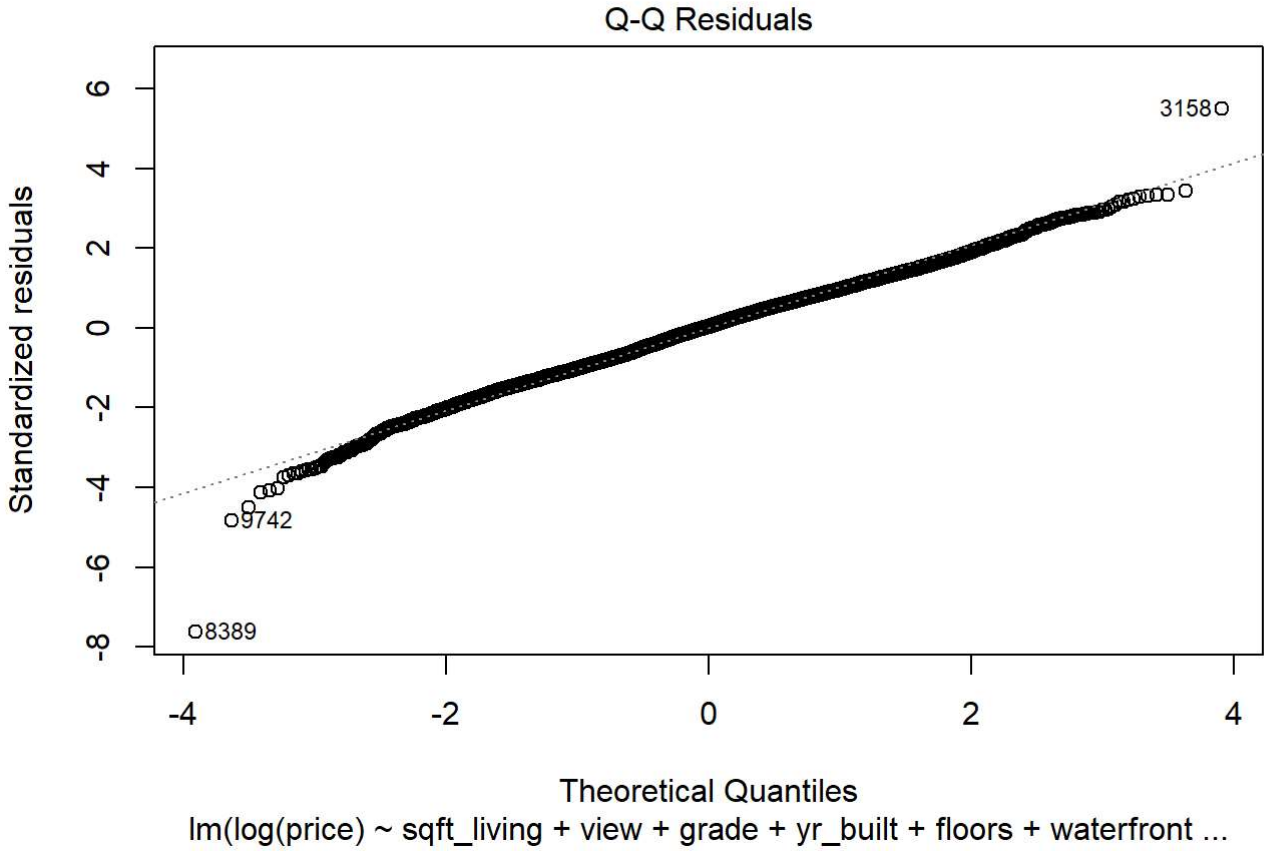
```
transformed_results <- lm(formula = log(price) ~ sqft_living + view + grade + yr_built +
    floors + waterfront + sqft_living15 + bathrooms + bedrooms +
    sqft_lot15, data = train)



# Example interpretation:
#1.For a 1 unit increase in sqft living, there is a 0.022% increase in price accounting for all other variables
#2. For a 1 unit increase in year built, there is a 0.44% decrease in price accounting for all other variables
summary(transformed_results)
```

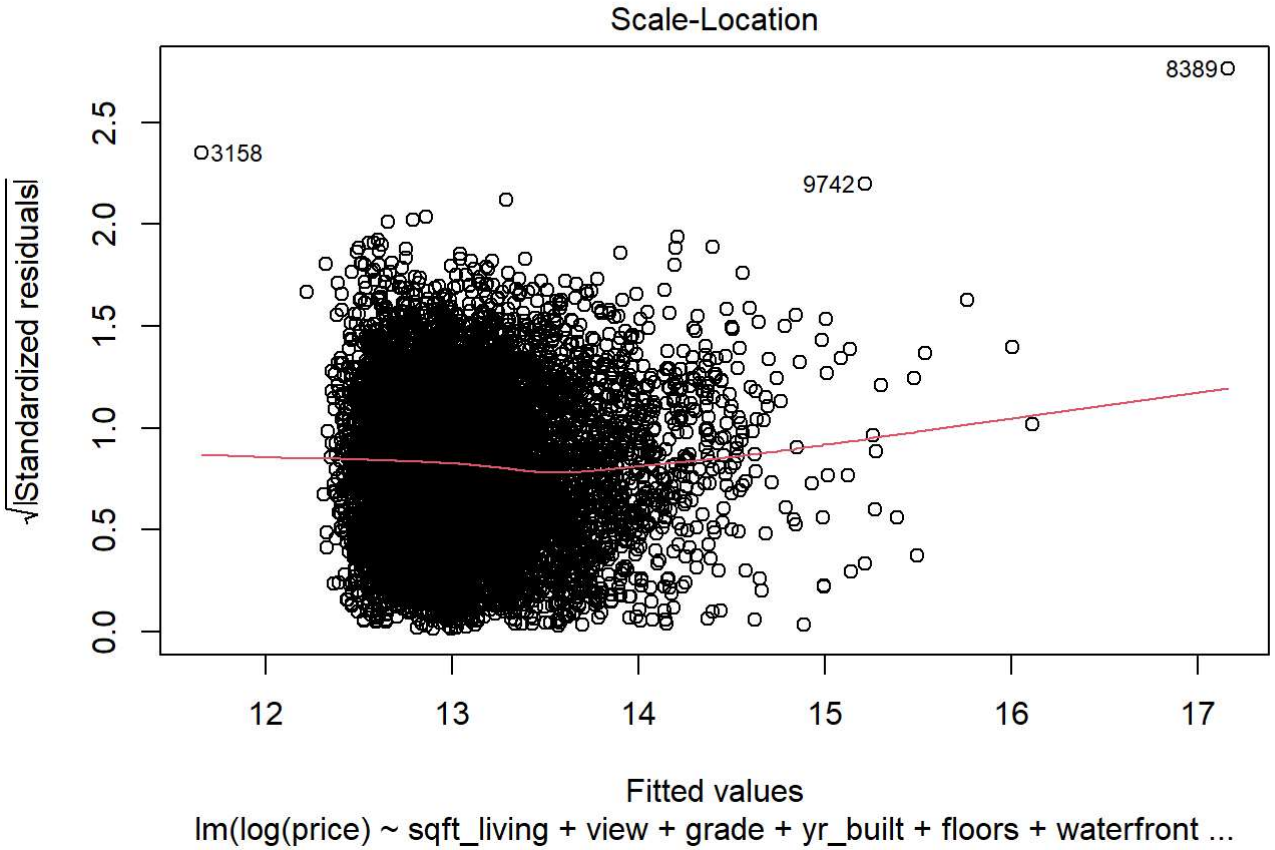
```
##
## Call:
## lm(formula = log(price) ~ sqft_living + view + grade + yr_built +
##     floors + waterfront + sqft_living15 + bathrooms + bedrooms +
##     sqft_lot15, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.52354 -0.23910  0.01264  0.23294  1.71645
##
## Coefficients:
##                         Estimate    Std. Error t value             Pr(>|t|)
## (Intercept)       21.2681033702  0.3009581597  70.668 < 0.0000000000000002 ***
## sqft_living        0.0002215410  0.0000075832  29.215 < 0.0000000000000002 ***
## view1              0.1989103620  0.0269316719   7.386    0.000000000000163 ***
## view2              0.1205371713  0.0163311520   7.381    0.000000000000169 ***
## view3              0.1270493548  0.0217335479   5.846    0.000000005188424 ***
## view4              0.2303374429  0.0340033778   6.774    0.000000000013183 ***
## gradehigh quality  0.1740232038  0.0151483827  11.488 < 0.0000000000000002 ***
## yr_built          -0.0047021965  0.0001551179 -30.314 < 0.0000000000000002 ***
## floors1.5          0.0825551311  0.0124632642   6.624    0.000000000036656 ***
## floors2            0.0942727571  0.0089997691  10.475 < 0.0000000000000002 ***
## floors2.5          0.1901596584  0.0366546302   5.188    0.000000216568279 ***
## floors3            0.4031531624  0.0214770094  18.771 < 0.0000000000000002 ***
## floors3.5          0.2915873833  0.1512847356   1.927                0.054 .
## waterfront1        0.2541151880  0.0474328845   5.357    0.000000086183255 ***
## sqft_living15      0.0001917973  0.0000075883  25.276 < 0.0000000000000002 ***
## bathrooms          0.1248989013  0.0075664456  16.507 < 0.0000000000000002 ***
## bedrooms          -0.0390629224  0.0043794468  -8.920 < 0.0000000000000002 ***
## sqft_lot15        -0.0000007729  0.0000001213  -6.373    0.000000000192400 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3377 on 10785 degrees of freedom
## Multiple R-squared:  0.5884, Adjusted R-squared:  0.5878
## F-statistic:   907 on 17 and 10785 DF,  p-value: < 0.00000000000000022
```

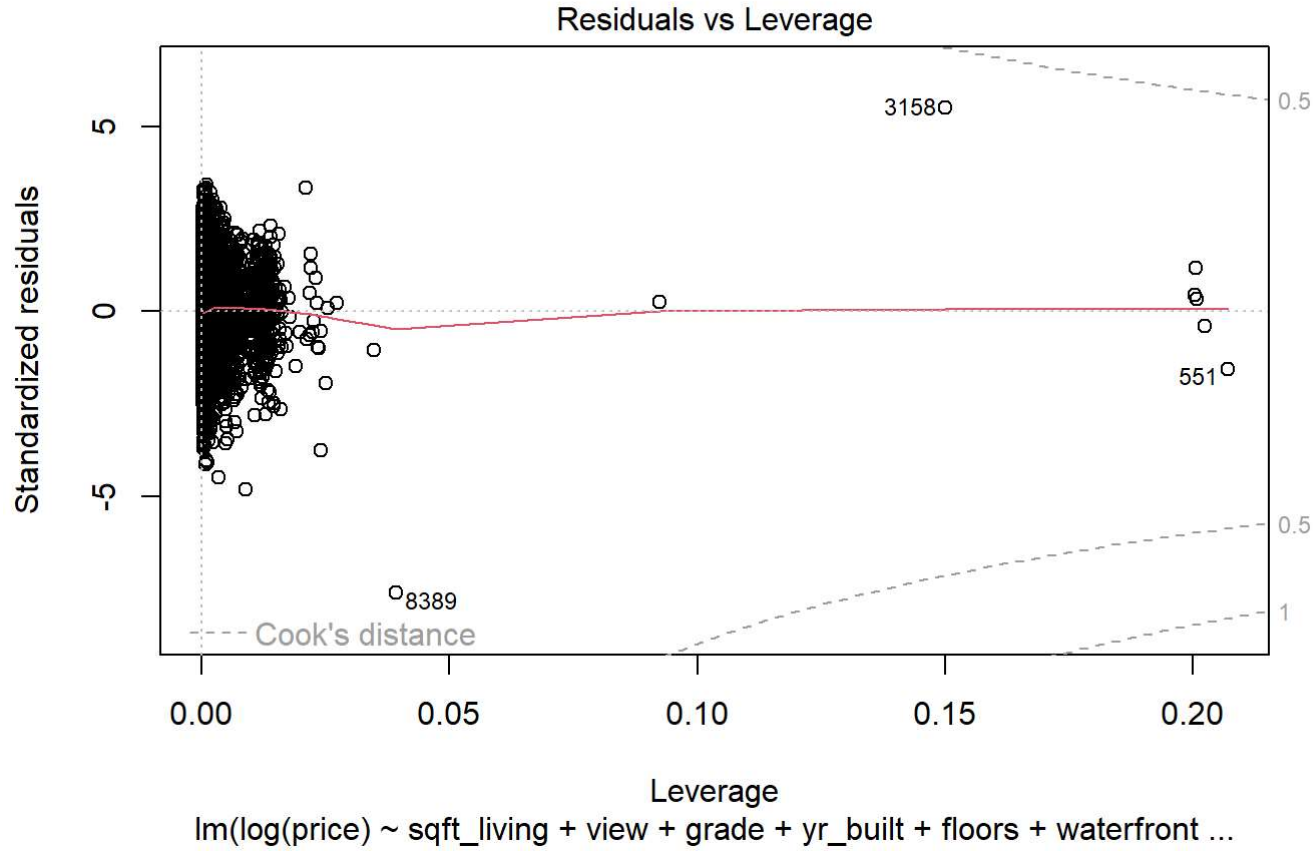
Assumptions look much better - go with this model

```
plot(transformed_results)
```


Residuals vs Fitted
Residuals
Fitted values
lm(log(price) ~ sqft_living + view + grade + yr_built + floors + waterfront ...
3158
9742
8389
2
1
0
-1
-2
-3
12
13
14
15
16
17


Q-Q Residuals
Standardized residuals
Theoretical Quantiles
lm(log(price) ~ sqft_living + view + grade + yr_built + floors + waterfront ...
3158
9742
8389
6
4
2
0
-2
-4
-6
-8
-4
-2
0
2
4


Scale-Location
8389
3158
9742
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
12
13
14
15
16
17
Fitted values
lm(log(price) ~ sqft_living + view + grade + yr_built + floors + waterfront ...


Residuals vs Leverage
3158
551
8389
Cook's distance
0.5
0.5
1
Standardized residuals
5
0
-5
0.00
0.05
0.10
0.15
0.20
Leverage
lm(log(price) ~ sqft_living + view + grade + yr_built + floors + waterfront ...

```r
#confidence intervals for coefficients
confint(transformed_results, level=0.95)
```

```
##                              2.5 %           97.5 %
## (Intercept)       20.678170010126 21.8580367302194
## sqft_living        0.000206676600  0.0002364054969
## view1              0.146119330589  0.2517013934925
## view2              0.088525108982  0.1525492335229
## view3              0.084447602558  0.1696511071127
## view4              0.163684566904  0.2969903188971
## gradehigh quality  0.144329586843  0.2037168208517
## yr_built          -0.005006256061 -0.0043981368961
## floors1.5          0.058124840368  0.1069854217620
## floors2            0.076631553920  0.1119139603029
## floors2.5          0.118309839797  0.2620094769801
## floors3            0.361054272895  0.4452520518601
## floors3.5         -0.004958530304  0.5881332968146
## waterfront1        0.161138008209  0.3470923678536
## sqft_living15      0.000176922885  0.0002066716661
## bathrooms          0.110067275843  0.1397305267418
## bedrooms          -0.047647443890 -0.0304784009713
## sqft_lot15        -0.000001010609 -0.0000005351898
```