Housing3_Week8_TatreauGillian

Gillian Tatreau

2022-10-23

```
# import packages
library(readxl)
library(ggplot2)
library(plyr)
library(pastecs)
library(ggplot2)
library(dplyr)
```

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

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

## 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(QuantPsyc)
```

```
## Loading required package: boot

## Loading required package: purrr

##
## Attaching package: 'purrr'

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

```
## Loading required package: MASS

##
## Attaching package: 'MASS'

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

##
## Attaching package: 'QuantPsyc'

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

```
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

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

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

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

```
# file source, read file
file_source <- "/Users/gillian/Documents/Bellevue Grad Program/Fall 2022/DSC520/DSC520 Repo/week-6-housi
housing <- read_excel(file_source)

# remove spaces in column names
names(housing) <- gsub(" ", "_", names(housing))
colnames(housing)
```

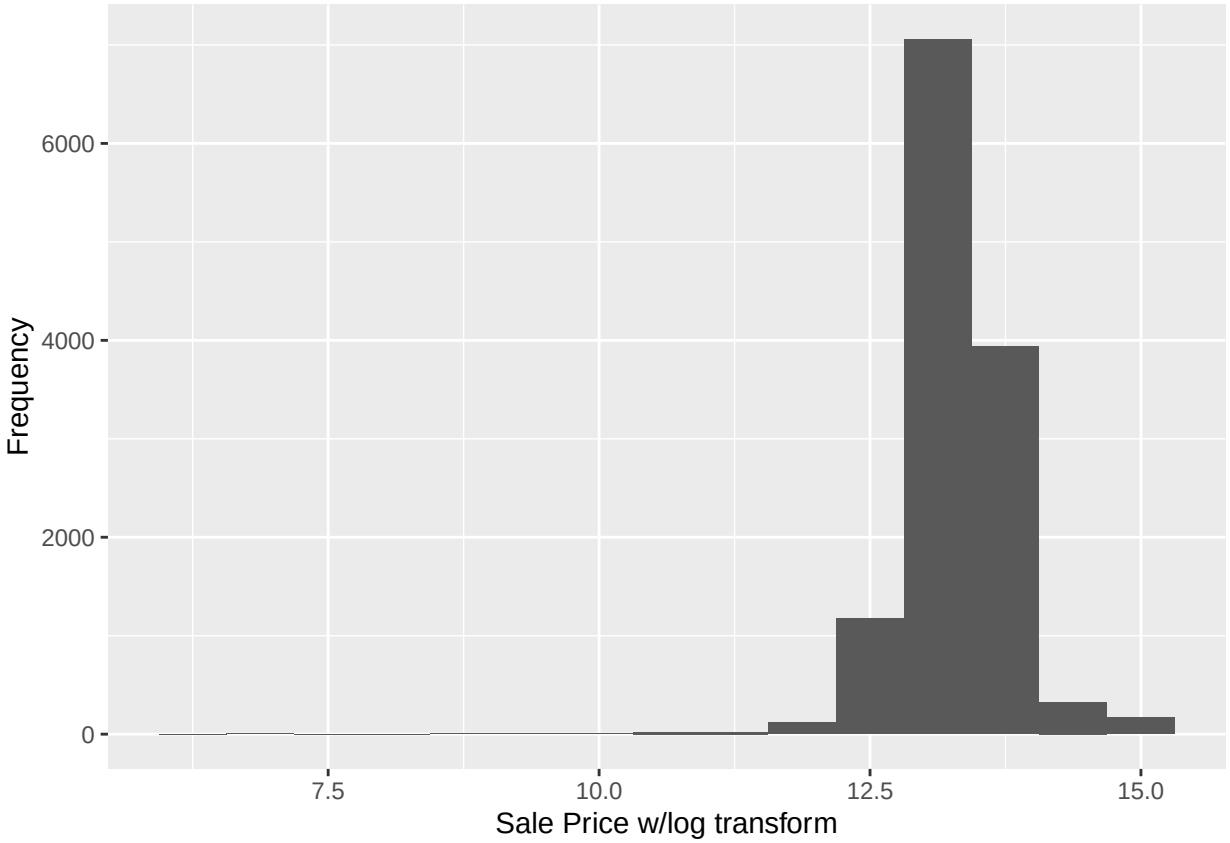
```
##  [1] "Sale_Date"                "Sale_Price"
##  [3] "sale_reason"              "sale_instrument"
##  [5] "sale_warning"             "sitetype"
##  [7] "addr_full"                "zip5"
##  [9] "ctyname"                  "postalctyn"
## [11] "lon"                      "lat"
## [13] "building_grade"           "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"
```

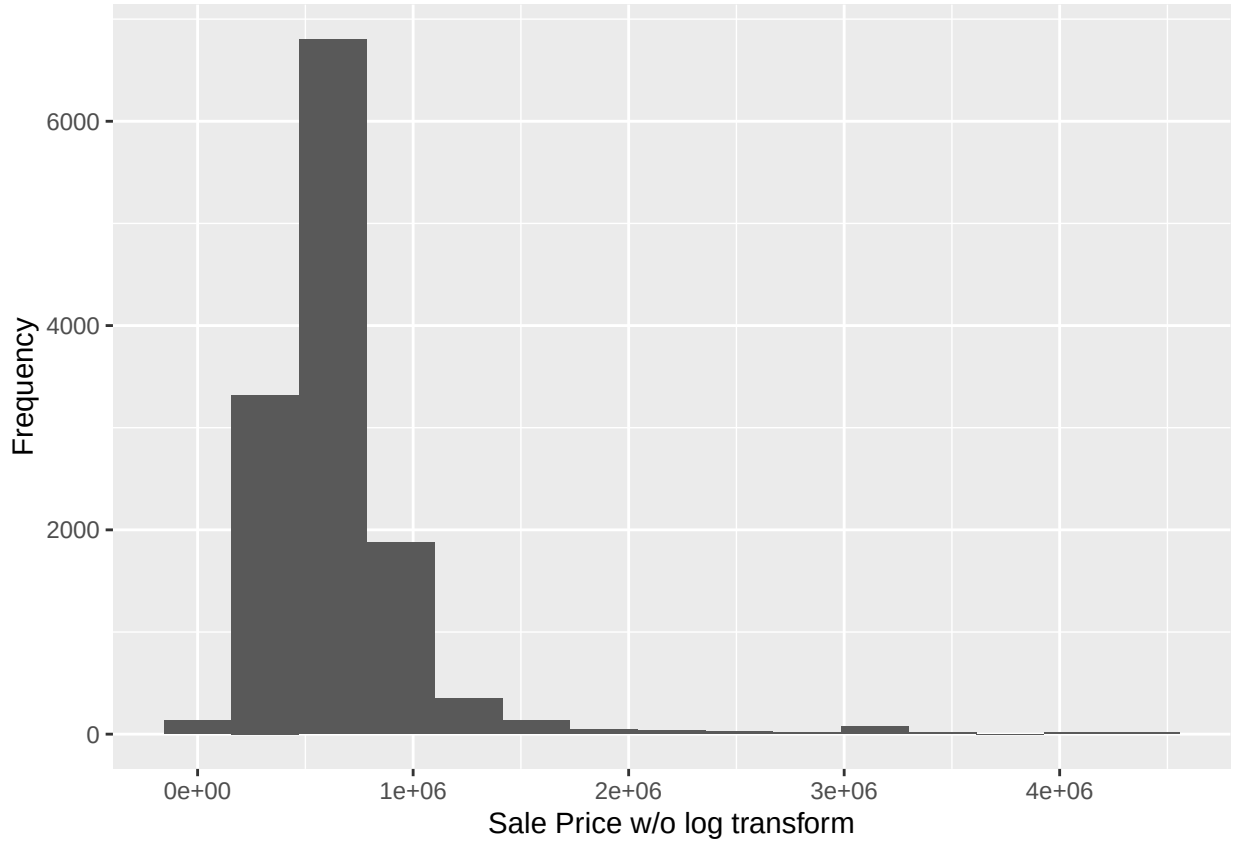
```
# Bathrooms
housing$bathrooms <- with(housing, bath_full_count + (bath_half_count * 0.5) + (bath_3qtr_count * 0.75))

# Sale Year
housing$Sale_Year <- format(housing$Sale_Date, format="%Y")

# Outside space
housing$sq_ft_outside <- with(housing, sq_ft_lot - square_feet_total_living)

# log transform of sale price
housing$log_price <- log(housing$Sale_Price)
stat.desc(housing$log_price, options(digits = 2))
```

```
##      nbr.val     nbr.null       nbr.na          min          max        range
##      1.3e+04      0.0e+00      0.0e+00      6.5e+00      1.5e+01      8.7e+00
##          sum       median         mean      SE.mean CI.mean.0.95          var
##      1.7e+05      1.3e+01      1.3e+01      4.6e-03      8.9e-03      2.7e-01
##      std.dev     coef.var
##      5.2e-01      3.9e-02
```

```
ggplot(data = housing) + geom_histogram(aes(x = log_price), bins = 15) + xlab("Sale Price w/log transfo
```



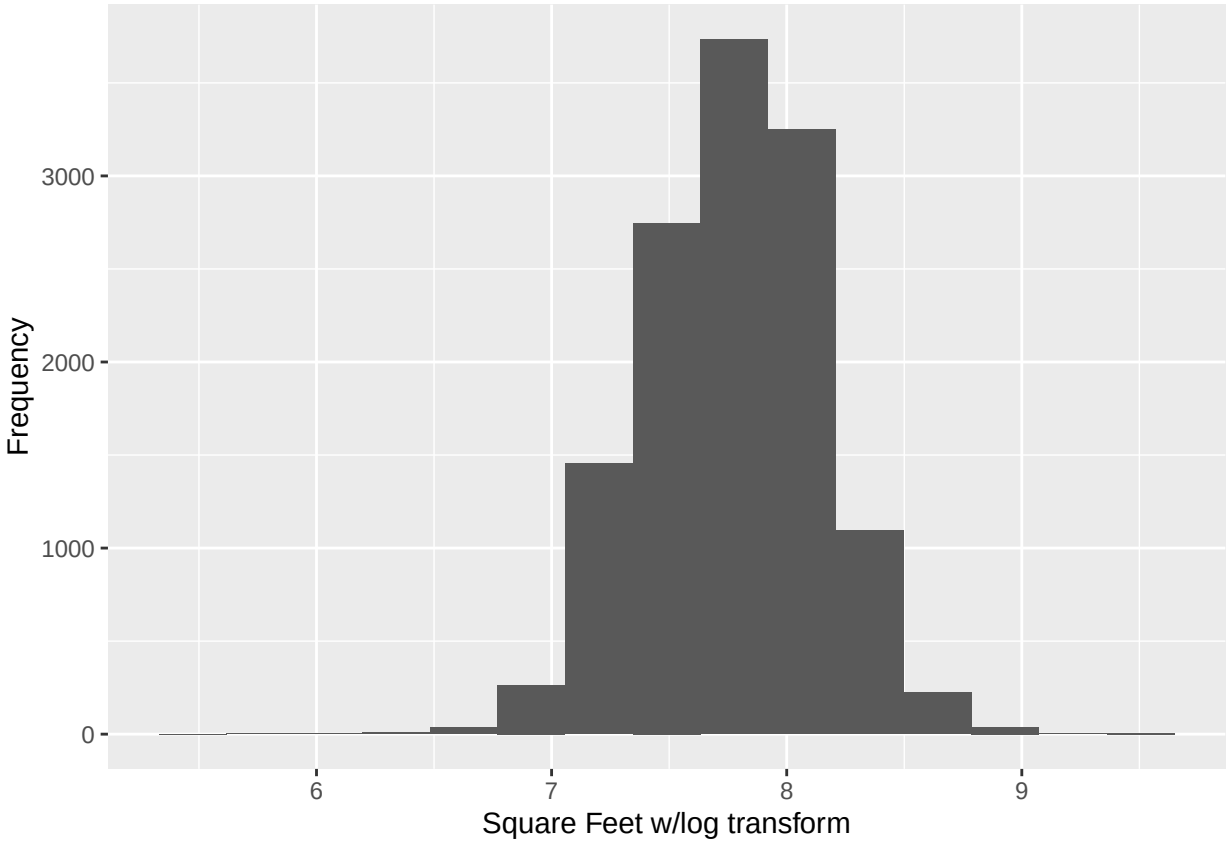

```
ggplot(data = housing) + geom_histogram(aes(x = Sale_Price), bins = 15) + xlab("Sale Price w/o log trans
```



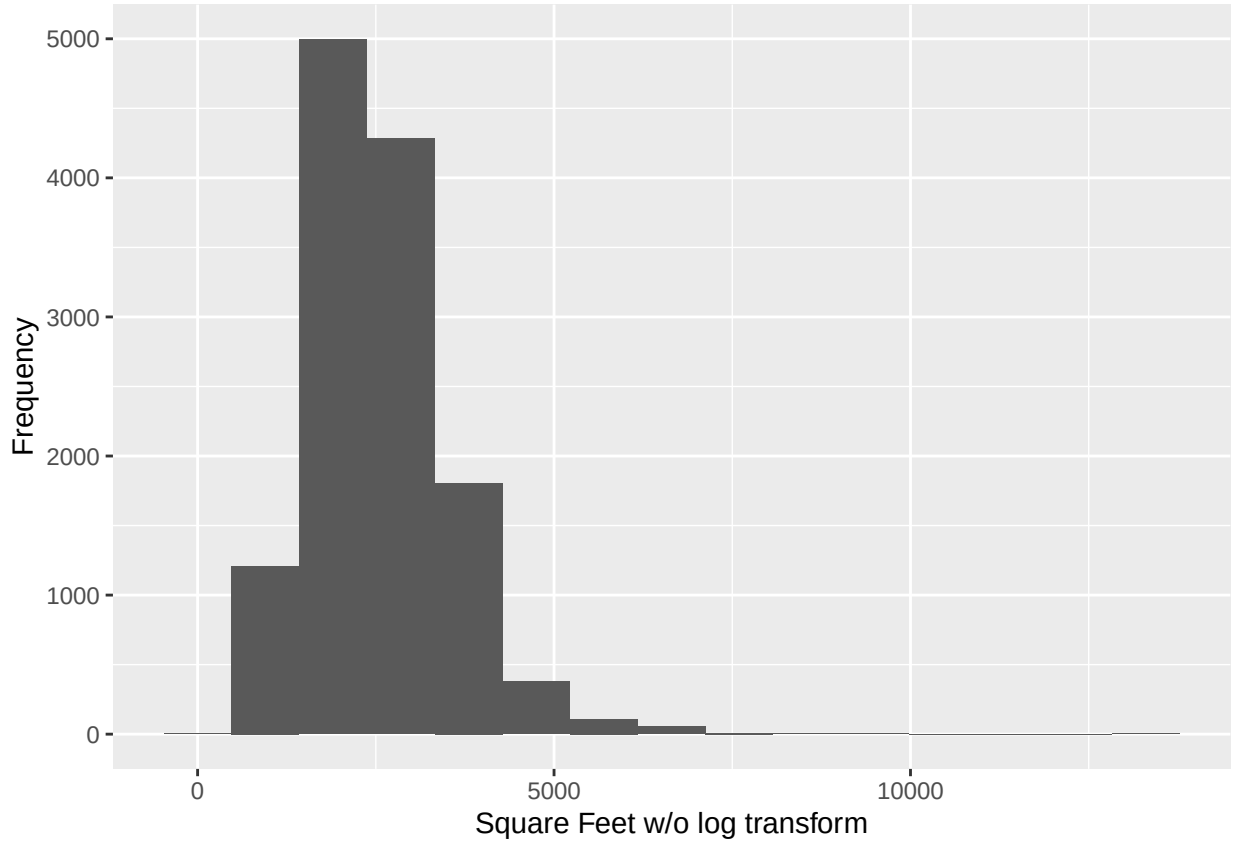

```
# log transform of square feet
housing$log_sqft <- log(housing$square_feet_total_living)
stat.desc(housing$log_sqft, options(digits = 2))
```

```
##      nbr.val     nbr.null       nbr.na          min          max        range
##      1.3e+04      0.0e+00      0.0e+00      5.5e+00      9.5e+00      4.0e+00
##          sum       median         mean      SE.mean CI.mean.0.95          var
##      1.0e+05      7.8e+00      7.8e+00      3.3e-03      6.5e-03      1.4e-01
##      std.dev     coef.var
##      3.8e-01      4.9e-02
```

```
ggplot(data = housing) + geom_histogram(aes(x = log_sqft), bins = 15) + xlab("Square Feet w/log transfo
```




```
ggplot(data = housing) + geom_histogram(aes(x = square_feet_total_living), bins = 15) + xlab("Square Fe
```

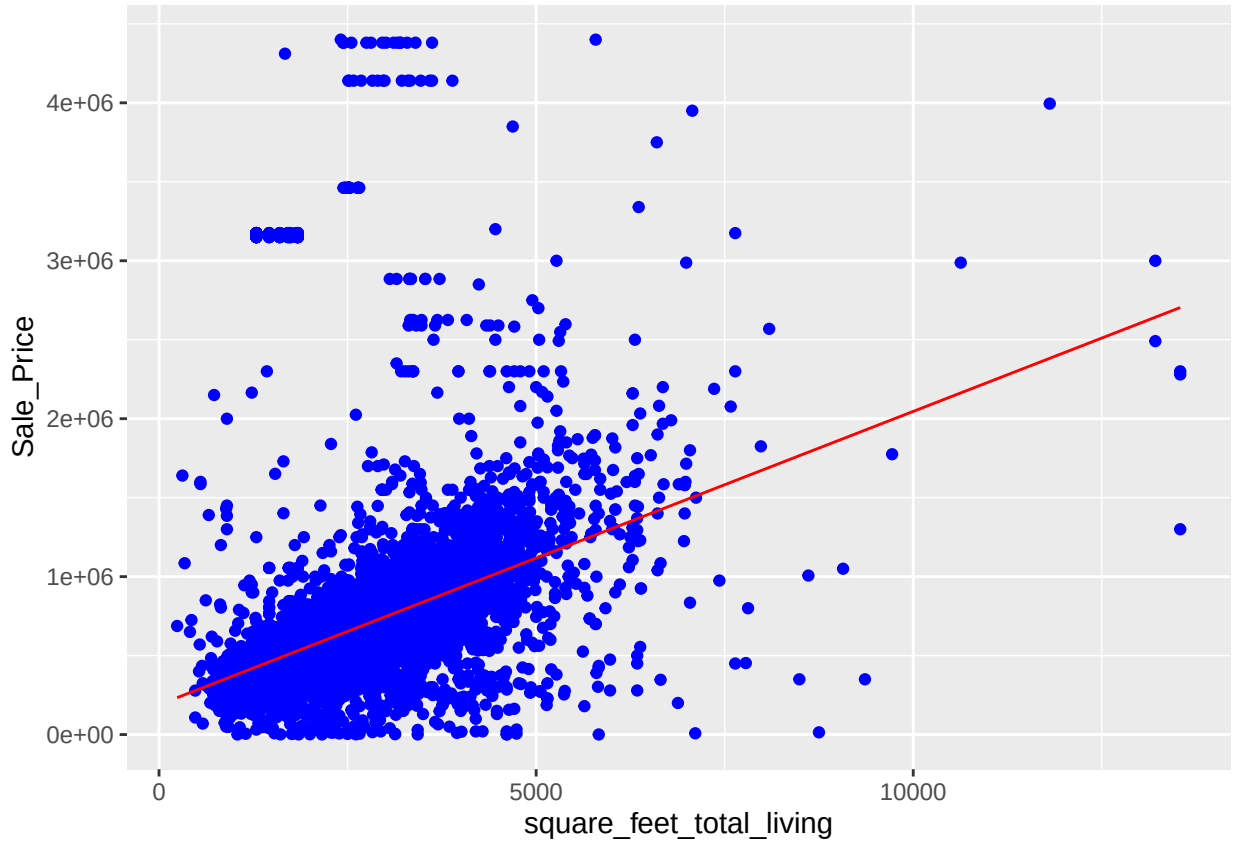



```
# variables
price <- housing$log_price
year1 <- housing$Sale_Year
sqft <- housing$log_sqft
bed <- housing$bedrooms
bath <- housing$bathrooms
year2 <- housing$year_built

# simple regression
simple_lm <- lm(Sale_Price ~ square_feet_total_living, data = housing)
summary(simple_lm)
```

```
##
## Call:
## lm(formula = Sale_Price ~ square_feet_total_living, data = housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1800136  -120257   -41547    44028  3811745
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.89e+05   8.74e+03    21.6   <2e-16 ***
## square_feet_total_living 1.86e+02   3.21e+00    57.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 360000 on 12863 degrees of freedom
## Multiple R-squared:  0.207,  Adjusted R-squared:  0.207
## F-statistic: 3.35e+03 on 1 and 12863 DF,  p-value: <2e-16
```

```
price_predict_df <- data.frame(sale_price = predict(simple_lm), square_feet=housing$square_feet_total_li
ggplot(data = housing, aes(y = Sale_Price, x = square_feet_total_living)) +
  geom_point(color='blue') +
  geom_line(color='red',data = price_predict_df, aes(y=sale_price, x=square_feet))
```

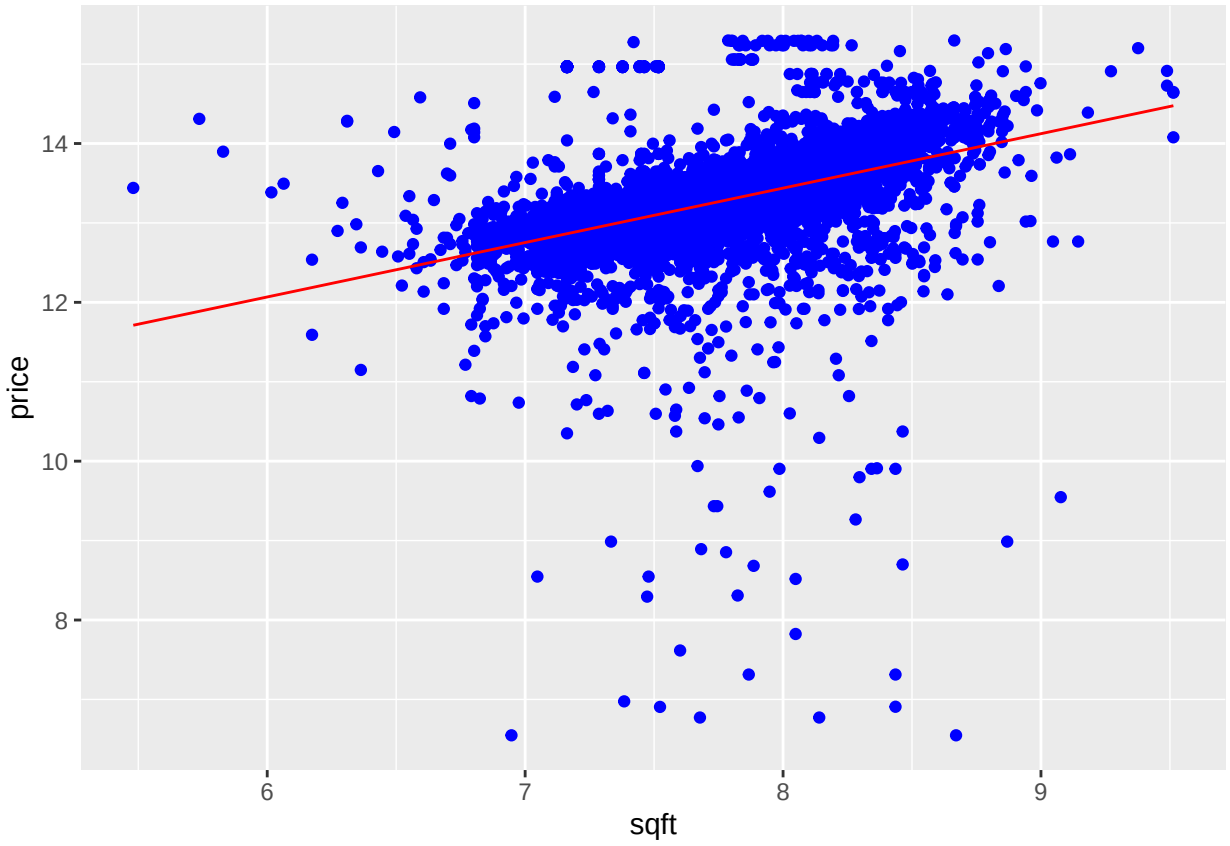



```
# with log
simplelog_lm <- lm(price ~ sqft, data = housing)
summary(simplelog_lm)
```

```
##
## Call:
## lm(formula = price ~ sqft, data = housing)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.350 -0.129  0.016  0.152  2.423
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.9548     0.0816    97.5   <2e-16 ***
```

```
## sqft          0.6854     0.0105    65.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.45 on 12863 degrees of freedom
## Multiple R-squared:  0.249,  Adjusted R-squared:  0.249
## F-statistic: 4.27e+03 on 1 and 12863 DF,  p-value: <2e-16
```

```
pricelog_predict_df <- data.frame(logsale_price = predict(simplelog_lm), logsquare_feet=sqft)
ggplot(data = housing, aes(y = price, x = sqft)) +
  geom_point(color='blue') +
  geom_line(color='red',data = pricelog_predict_df, aes(y=logsale_price, x=logsquare_feet))
```




```
# correlation to see which variables to use as predictors
cor(price, sqft)
```

```
## [1] 0.5
```

```
cor(price, bed)
```

```
## [1] 0.27
```

```r
cor(price, bath)
```

```
## [1] 0.39
```

```r
cor(price, as.numeric(year1))
```

```
## [1] 0.095
```

```r
cor(price, year2)
```

```
## [1] 0.28
```

```r
cor(price, as.numeric(housing$Sale_Date))
```

```
## [1] 0.099
```

```r
cor(price, housing$sq_ft_outside)
```

```
## [1] -0.0098
```

```r
# multiple regression
multiple_lm <- lm(price ~ sqft + bed + bath + year2 + as.numeric(year1), data = housing)
summary(multiple_lm)
```

```
##
## Call:
## lm(formula = price ~ sqft + bed + bath + year2 + as.numeric(year1),
##     data = housing)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.231 -0.124  0.017  0.148  2.447
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -2.53e+01   2.47e+00  -10.24   <2e-16 ***
## sqft               6.32e-01   1.70e-02   37.12   <2e-16 ***
## bed               -1.67e-02   5.96e-03   -2.79   0.0052 **
## bath               1.50e-02   8.68e-03    1.73   0.0832 .
## year2              3.41e-03   2.59e-04   13.15   <2e-16 ***
## as.numeric(year1)  1.34e-02   1.20e-03   11.16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.44 on 12859 degrees of freedom
## Multiple R-squared:  0.269,  Adjusted R-squared:  0.269
## F-statistic:  949 on 5 and 12859 DF,  p-value: <2e-16
```

```r
mult_price_predict <- data.frame(
  sale_price = predict(multiple_lm),
  square_feet=sqft, bed=bed, bath=bath,
  year2=year2, year1 = as.numeric(year1))


# standardized betas
lm.beta(multiple_lm)
```

```
##              sqft               bed              bath             year2
##             0.460            -0.028             0.020             0.113
## as.numeric(year1)
##             0.084
```

```r
# confidence intervals
confint(multiple_lm)
```

```
##                       2.5 %   97.5 %
## (Intercept)       -30.0965 -20.4265
## sqft                0.5986   0.6654
## bed                -0.0284  -0.0050
## bath               -0.0020   0.0320
## year2               0.0029   0.0039
## as.numeric(year1)   0.0110   0.0157
```

```r
# analysis of variance
anova(simplelog_lm, multiple_lm)
```

```
## Analysis of Variance Table
##
## Model 1: price ~ sqft
## Model 2: price ~ sqft + bed + bath + year2 + as.numeric(year1)
##   Res.Df  RSS Df Sum of Sq    F Pr(>F)
## 1  12863 2587
## 2  12859 2517  4      69.9 89.3 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# case-wise diagnostics for outliers/influential cases
housing$residuals <- resid(multiple_lm)
housing$stan_resid <- rstandard(multiple_lm)
housing$cooks <- cooks.distance(multiple_lm)
housing$leverage <- hatvalues(multiple_lm)
housing$covariance <- covratio(multiple_lm)
head(housing[c("residuals", "stan_resid", "cooks", "leverage", "covariance")], n=10)
```

```
## # A tibble: 10 x 5
##    residuals stan_resid        cooks leverage covariance
##        <dbl>      <dbl>        <dbl>    <dbl>      <dbl>
##  1   0.109       0.247  0.00000364   0.000359      1.00
##  2   0.00826     0.0187 0.0000000200 0.000345      1.00
```

```
##  3  -0.0218     -0.0492 0.000000154  0.000381      1.00
##  4   0.0631      0.143  0.00000155   0.000456      1.00
##  5  -0.0304     -0.0687 0.000000321  0.000408      1.00
##  6  -1.49       -3.36   0.000864     0.000459      0.996
##  7   0.333       0.752  0.0000608    0.000645      1.00
##  8   0.209       0.472  0.0000197    0.000529      1.00
##  9  -0.121      -0.273  0.00000868   0.000697      1.00
## 10   0.153       0.345  0.0000131    0.000662      1.00
```

```
# large residuals
housing$large_resid <- housing$stan_resid > 2 | housing$stan_resid < -2
head(housing$large_resid, n = 30)
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12    13
## FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    14    15    16    17    18    19    20    21    22    23    24    25    26
##  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
##    27    28    29    30
## FALSE FALSE FALSE FALSE
```

```
# sum of large residuals
sum(housing$large_resid)
```

```
## [1] 475
```

```
# which variables have large residuals
housing[housing$large_resid, c("log_price", "log_sqft", "Sale_Year", "bedrooms", "bathrooms", "year_buil
```

```
## # A tibble: 475 x 7
##    log_price log_sqft Sale_Year bedrooms bathrooms year_built stan_resid
##        <dbl>    <dbl> <chr>        <dbl>     <dbl>      <dbl>      <dbl>
##  1      12.1     8.33 2006             4      3.25       2005      -3.36
##  2      12.0     7.52 2006             3      2          2011      -2.50
##  3      12.5     8.50 2006             4      4.5        2007      -2.84
##  4      11.9     7.57 2006             3      2.5        2003      -2.76
##  5      14.1     6.49 2006             0      1          1955       4.15
##  6      12.3     8.25 2006             0      0          2008      -2.83
##  7      12.9     8.67 2006             5      4.5        2008      -2.17
##  8      12.7     8.35 2006             4      3.25       2015      -2.25
##  9      14.2     6.80 2006             2      1          1918       4.16
## 10      12.0     8.46 2006             4      4          2014      -3.92
## # ... with 465 more rows
```

```
# leverage, Cook's distance, covariance ratios, k = 5
housing[housing$large_resid, c("leverage", "cooks", "covariance")]
```

```
## # A tibble: 475 x 3
##    leverage    cooks covariance
##       <dbl>    <dbl>      <dbl>
##  1 0.000459 0.000864      0.996
##  2 0.000463 0.000482      0.998
```

```
##  3 0.00106  0.00143       0.998
##  4 0.000348 0.000442      0.997
##  5 0.00198  0.00568       0.994
##  6 0.00591  0.00795       1.00
##  7 0.000962 0.000759      0.999
##  8 0.000509 0.000431      0.999
##  9 0.00199  0.00575       0.994
## 10 0.000706 0.00180       0.994
## # ... with 465 more rows
```

```
housing$cooksprob <- housing$cooks > 1
avelev <- 6 / 12865
housing$badlev <- housing$leverage > (3 * avelev)
highcov <- 1 + ((3 * 6) / 12865)
lowcov <- 1 - ((3 * 6) / 12865)
housing$covissue <- housing$covariance > highcov | housing$covariance < lowcov
sum(housing$covissue)
```

```
## [1] 723
```

```
sum(housing$badlev)
```

```
## [1] 240
```

```
sum(housing$cooksprob)
```

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

```
housing[housing$large_resid, c("badlev", "cooksprob", "covissue")]
```

```
## # A tibble: 475 x 3
##    badlev cooksprob covissue
##    <lgl>  <lgl>     <lgl>
##  1 FALSE  FALSE     TRUE
##  2 FALSE  FALSE     TRUE
##  3 FALSE  FALSE     TRUE
##  4 FALSE  FALSE     TRUE
##  5 TRUE   FALSE     TRUE
##  6 TRUE   FALSE     TRUE
##  7 FALSE  FALSE     FALSE
##  8 FALSE  FALSE     FALSE
##  9 TRUE   FALSE     TRUE
## 10 FALSE  FALSE     TRUE
## # ... with 465 more rows
```

```
# assumption of independence
durbinWatsonTest(multiple_lm)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1            0.33           1.3       0
##  Alternative hypothesis: rho != 0
```

```r
# assumption of no multicollinearity
vif(multiple_lm)
```

```
##              sqft               bed              bath             year2
##               2.7               1.8               2.4               1.3
## as.numeric(year1)
##               1.0
```

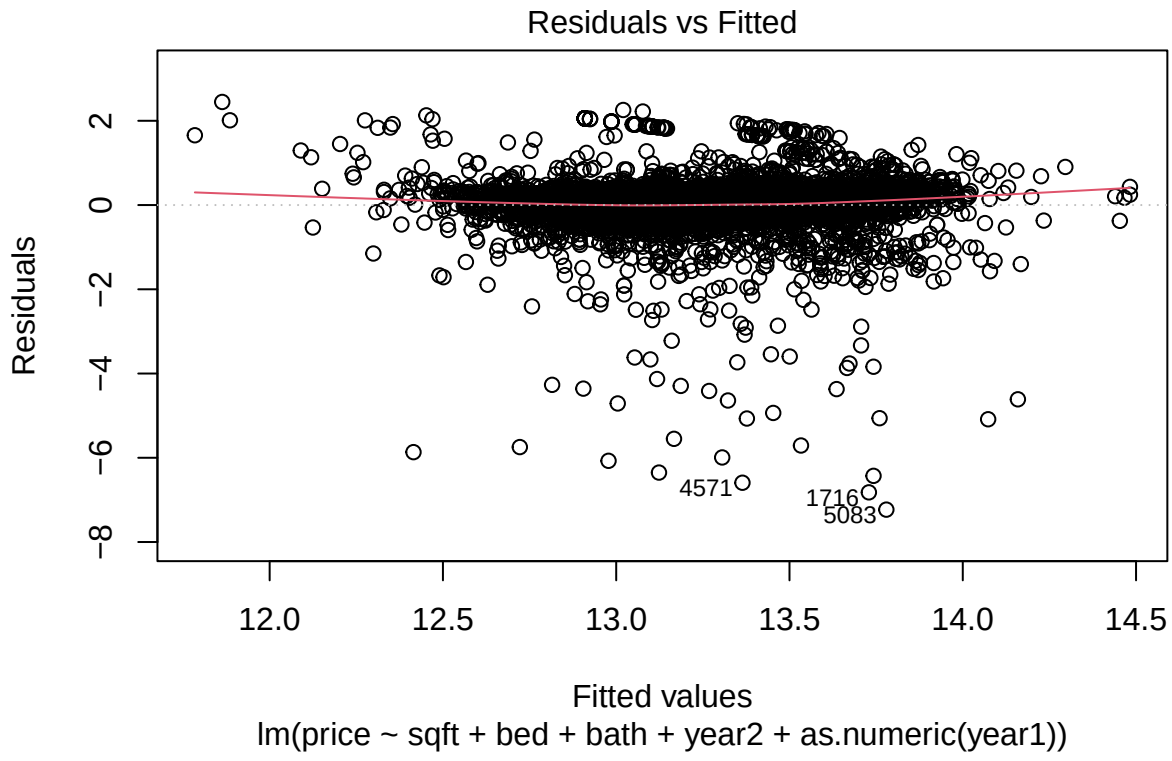
```r
mean(vif(multiple_lm))
```

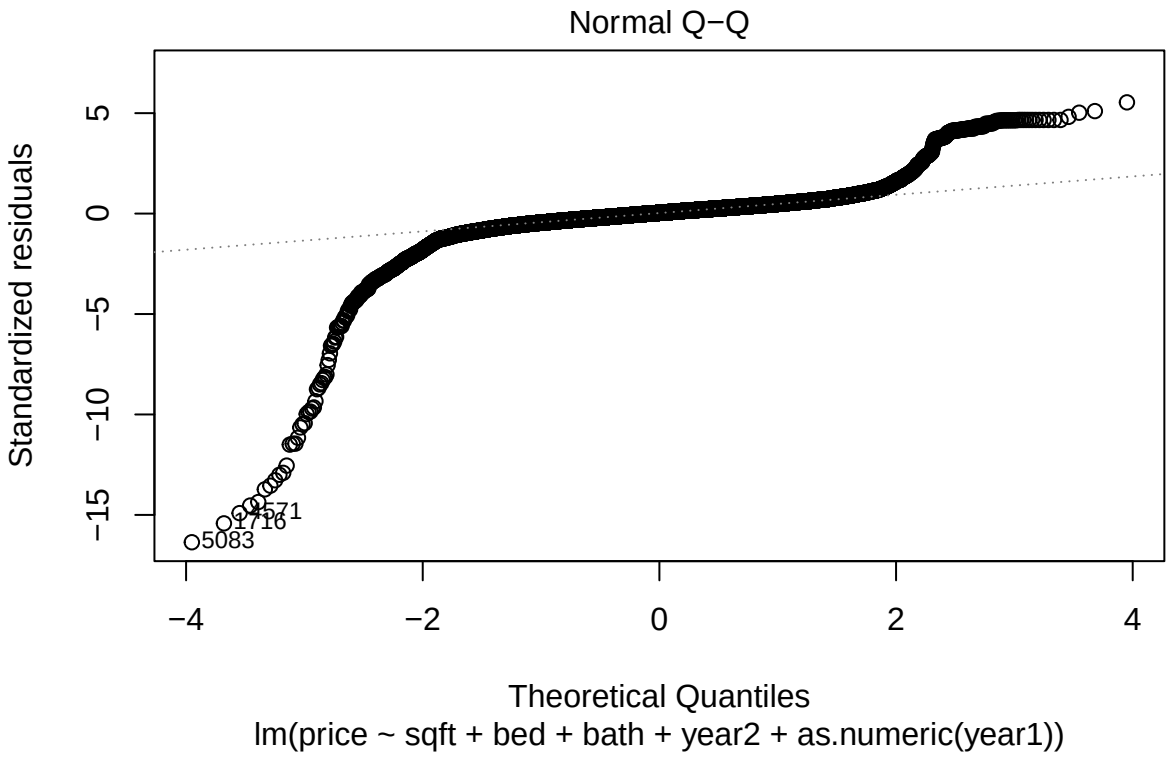
```
## [1] 1.8
```

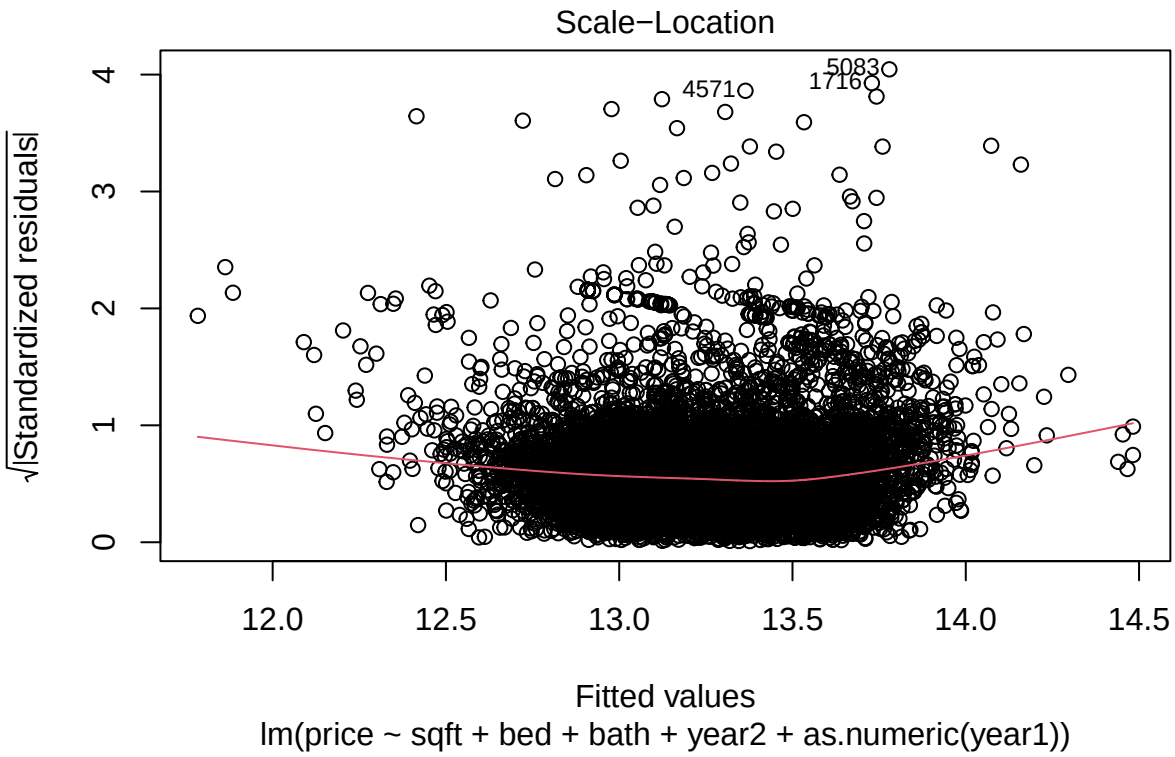
```r
1/vif(multiple_lm)
```

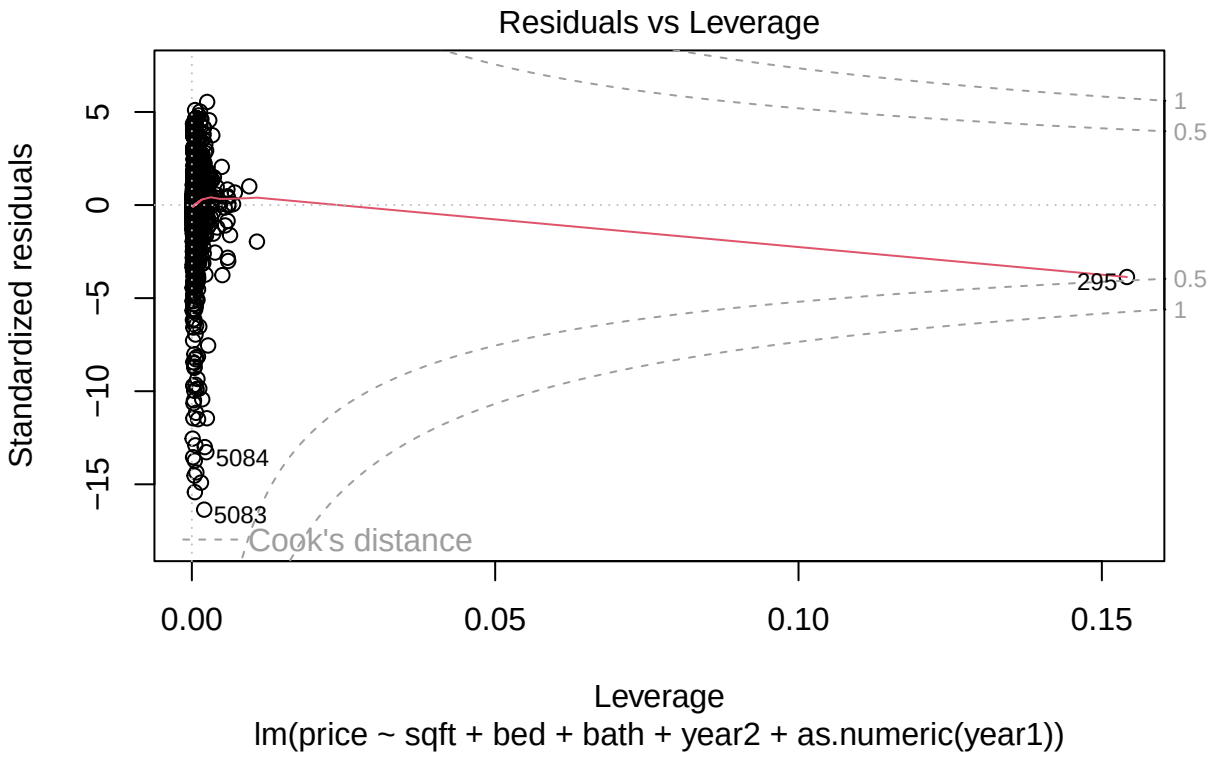
```
##              sqft               bed              bath             year2
##              0.37              0.56              0.42              0.76
## as.numeric(year1)
##              1.00
```

```r
# assumptions related to residuals
housing$fitted <- multiple_lm$fitted.values

plot(multiple_lm)
```

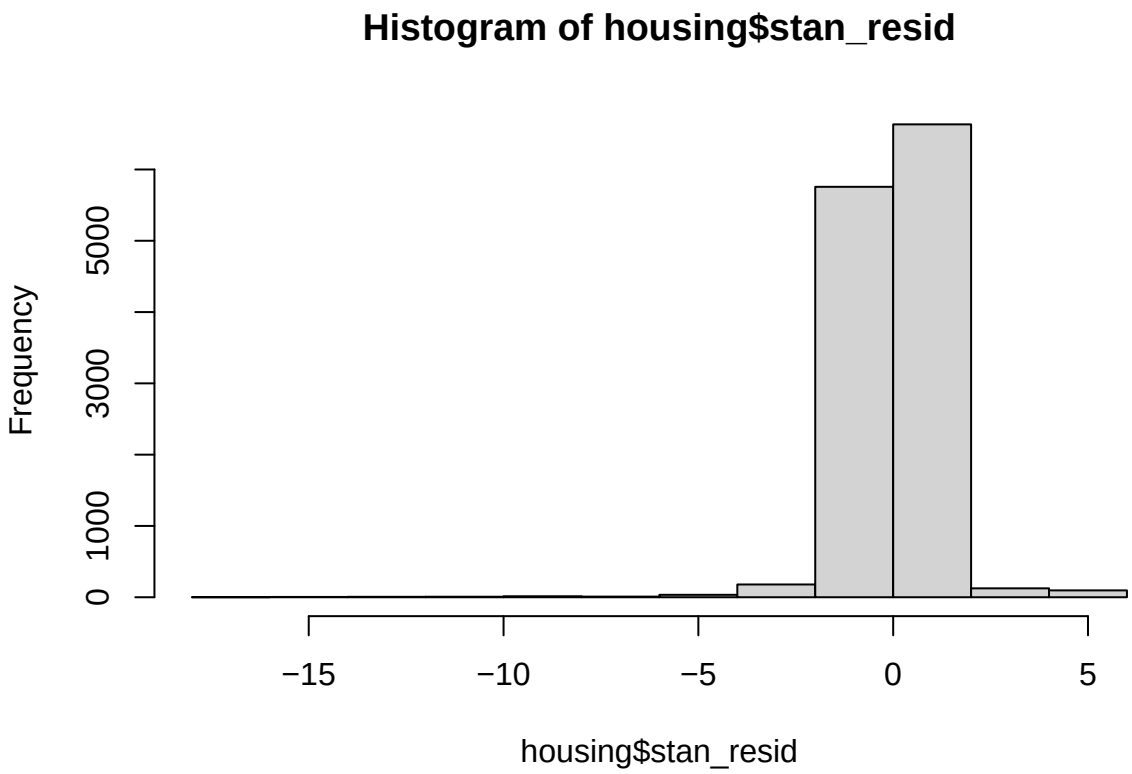



Normal Q−Q
Standardized residuals
5
0
−5
−10
−15
−4
−2
0
2
4
5083
1716
571
Theoretical Quantiles
lm(price ~ sqft + bed + bath + year2 + as.numeric(year1))


Scale−Location
√|Standardized residuals|
0
1
2
3
4
5083
1716
4571
12.0
12.5
13.0
13.5
14.0
14.5
Fitted values
lm(price ~ sqft + bed + bath + year2 + as.numeric(year1))

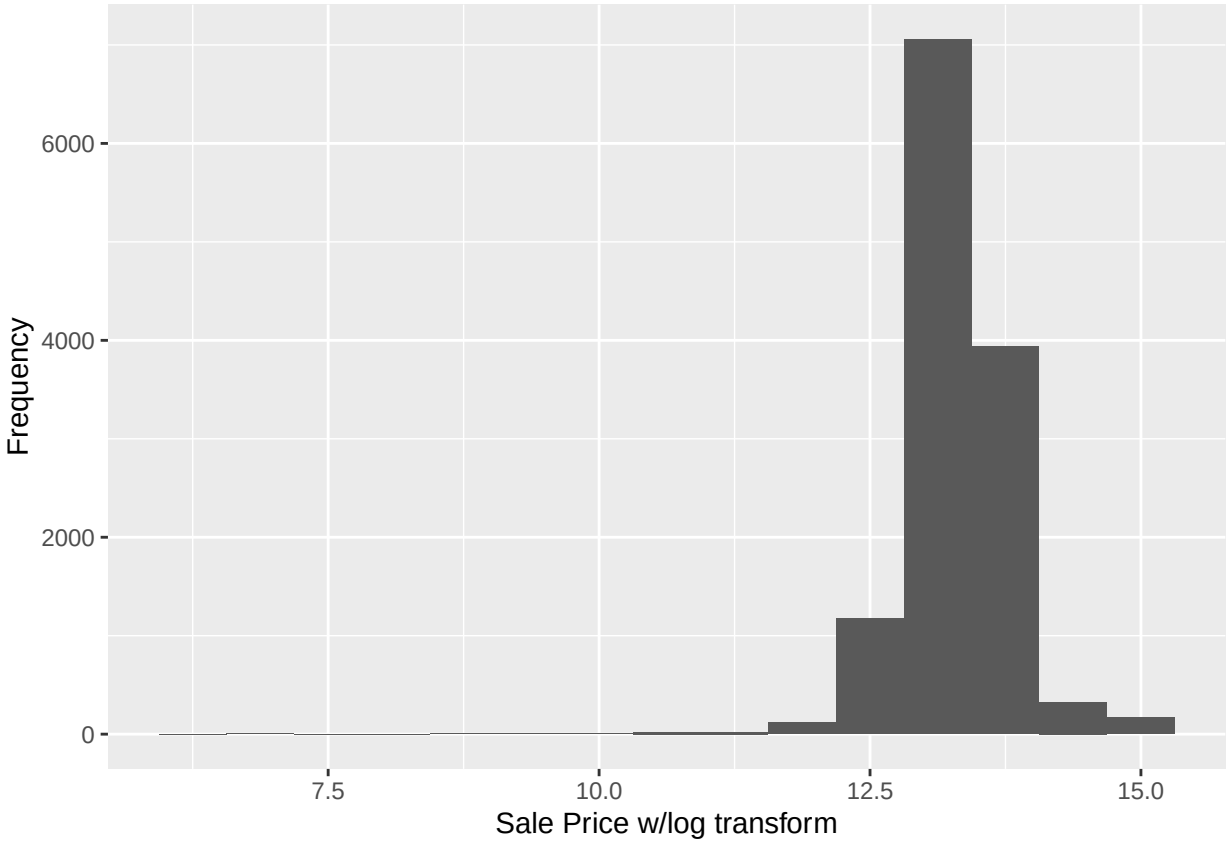


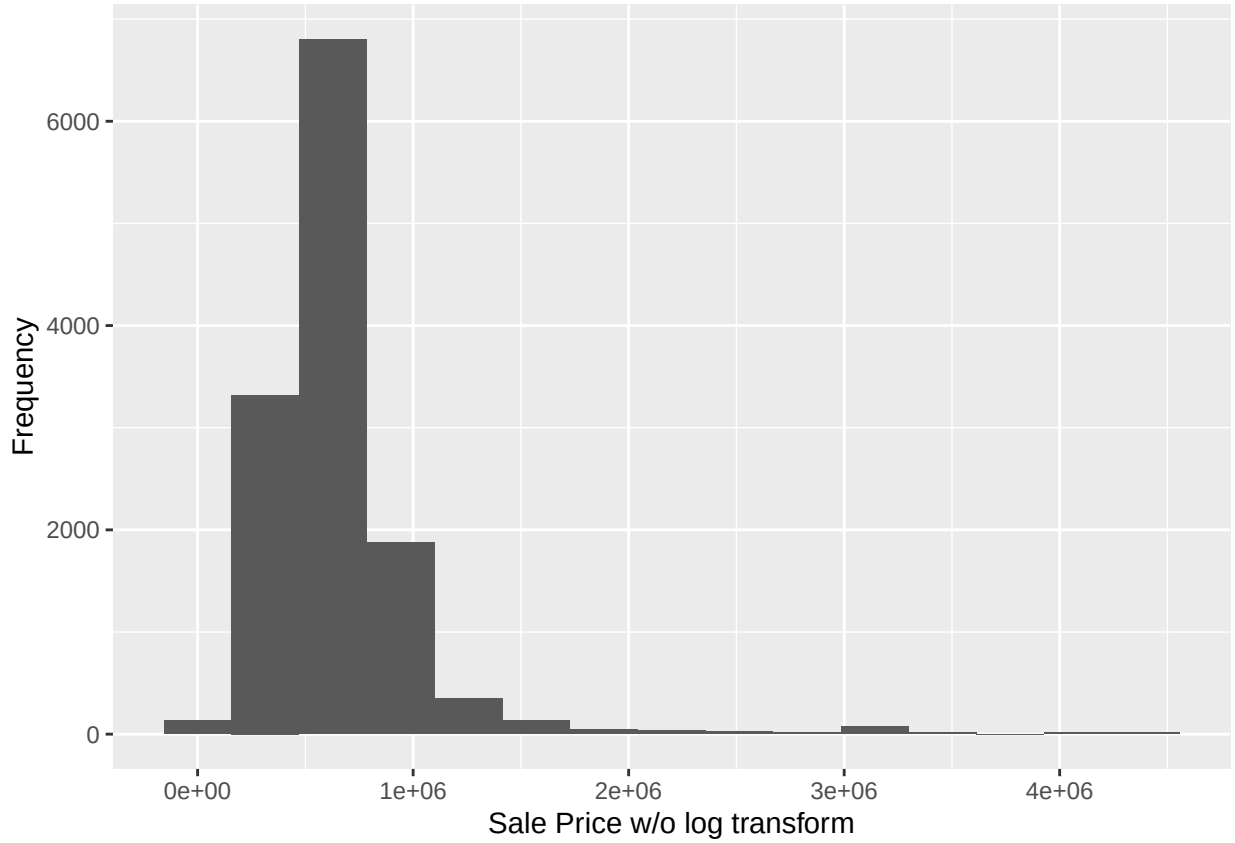

```
hist(housing$stan_resid)
```



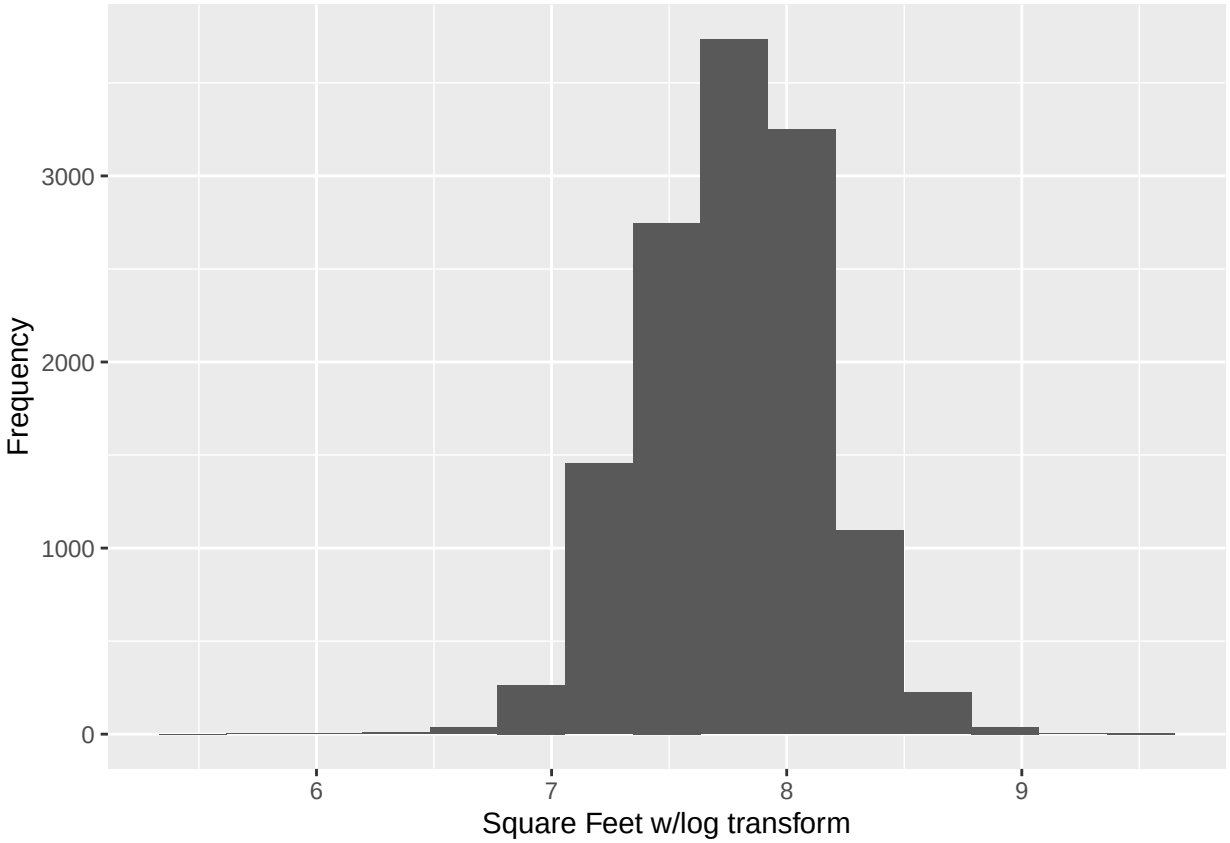
i. I used a log transform on both the Sale_Price and square_feet_total_living columns in order to make the distributions of that data more normal as well as more linear. I also added the variables bathrooms (the total number of bathrooms including half and three quarter bathrooms), Sale_Year (the year the house was sold), and sq_ft_outside which was meant to calculate the square footage of any outside space of the property.

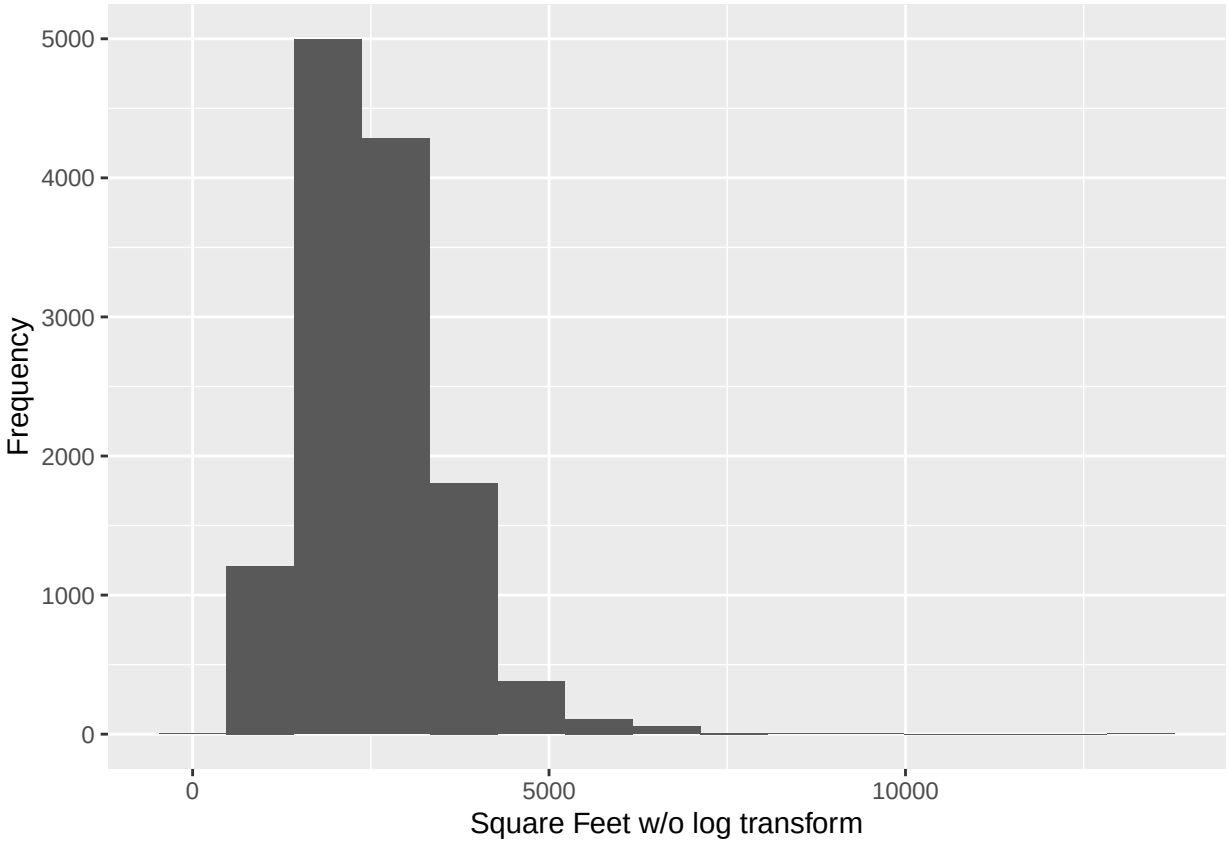
```
##      nbr.val     nbr.null       nbr.na          min          max        range
##      1.3e+04      0.0e+00      0.0e+00      6.5e+00      1.5e+01      8.7e+00
##          sum       median         mean      SE.mean CI.mean.0.95          var
##      1.7e+05      1.3e+01      1.3e+01      4.6e-03      8.9e-03      2.7e-01
##      std.dev     coef.var
##      5.2e-01      3.9e-02
```


6000
4000
2000
0
Frequency
7.5
10.0
12.5
15.0
Sale Price w/log transform

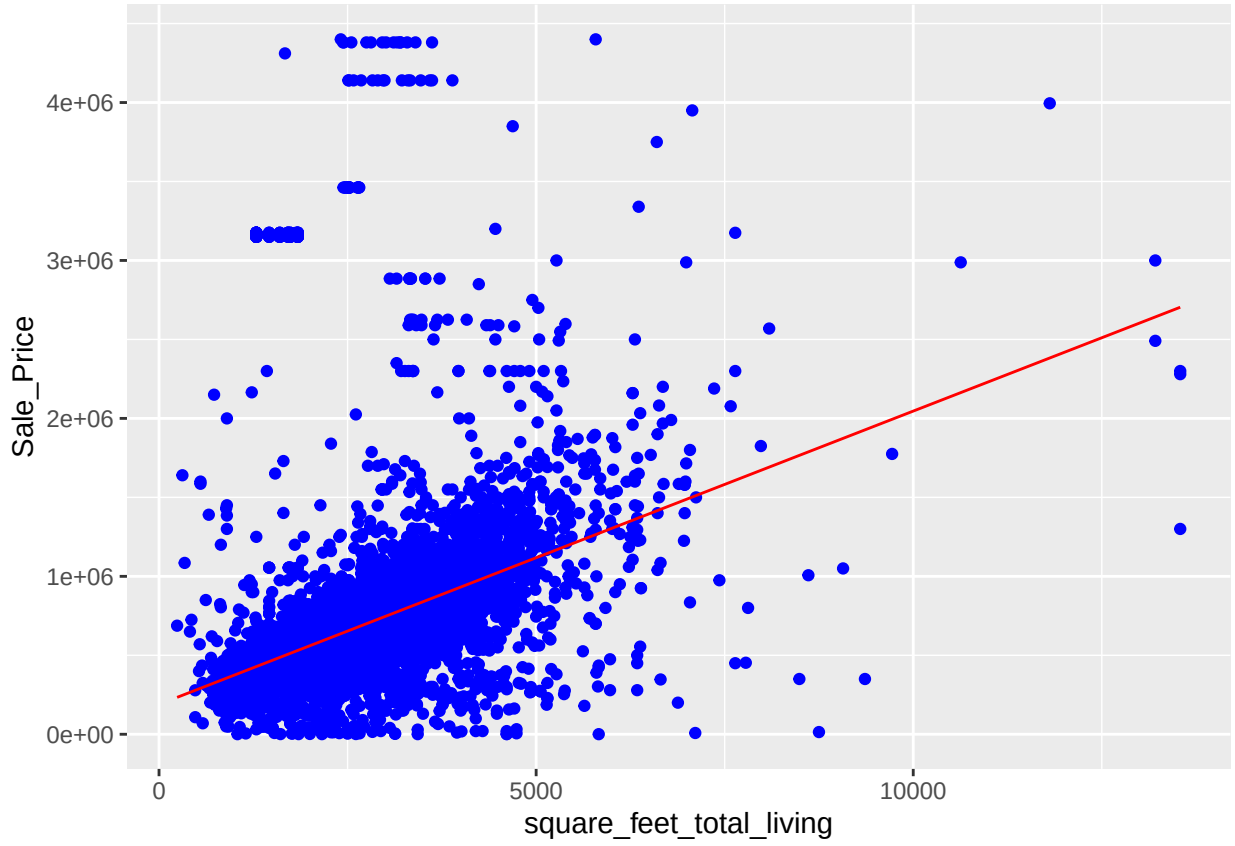



```
##      nbr.val     nbr.null       nbr.na          min          max        range
##      1.3e+04      0.0e+00      0.0e+00      5.5e+00      9.5e+00      4.0e+00
##          sum       median         mean      SE.mean CI.mean.0.95          var
##      1.0e+05      7.8e+00      7.8e+00      3.3e-03      6.5e-03      1.4e-01
##      std.dev     coef.var
##      3.8e-01      4.9e-02
```

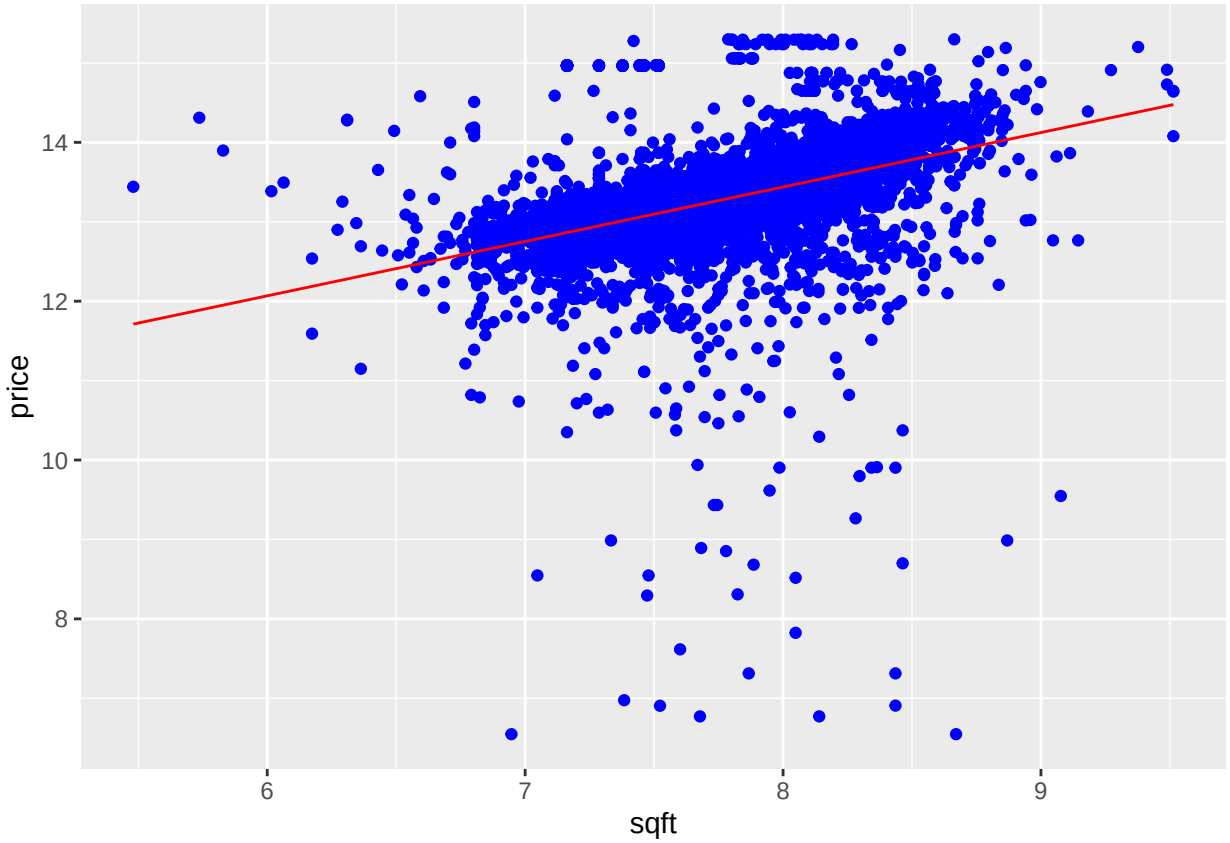

Frequency
3000
2000
1000
0
6
7
8
9
Square Feet w/log transform




```
##
## Call:
## lm(formula = Sale_Price ~ square_feet_total_living, data = housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1800136  -120257   -41547    44028  3811745
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               1.89e+05   8.74e+03    21.6   <2e-16 ***
## square_feet_total_living 1.86e+02   3.21e+00    57.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 360000 on 12863 degrees of freedom
## Multiple R-squared:  0.207,  Adjusted R-squared:  0.207
## F-statistic: 3.35e+03 on 1 and 12863 DF,  p-value: <2e-16
```

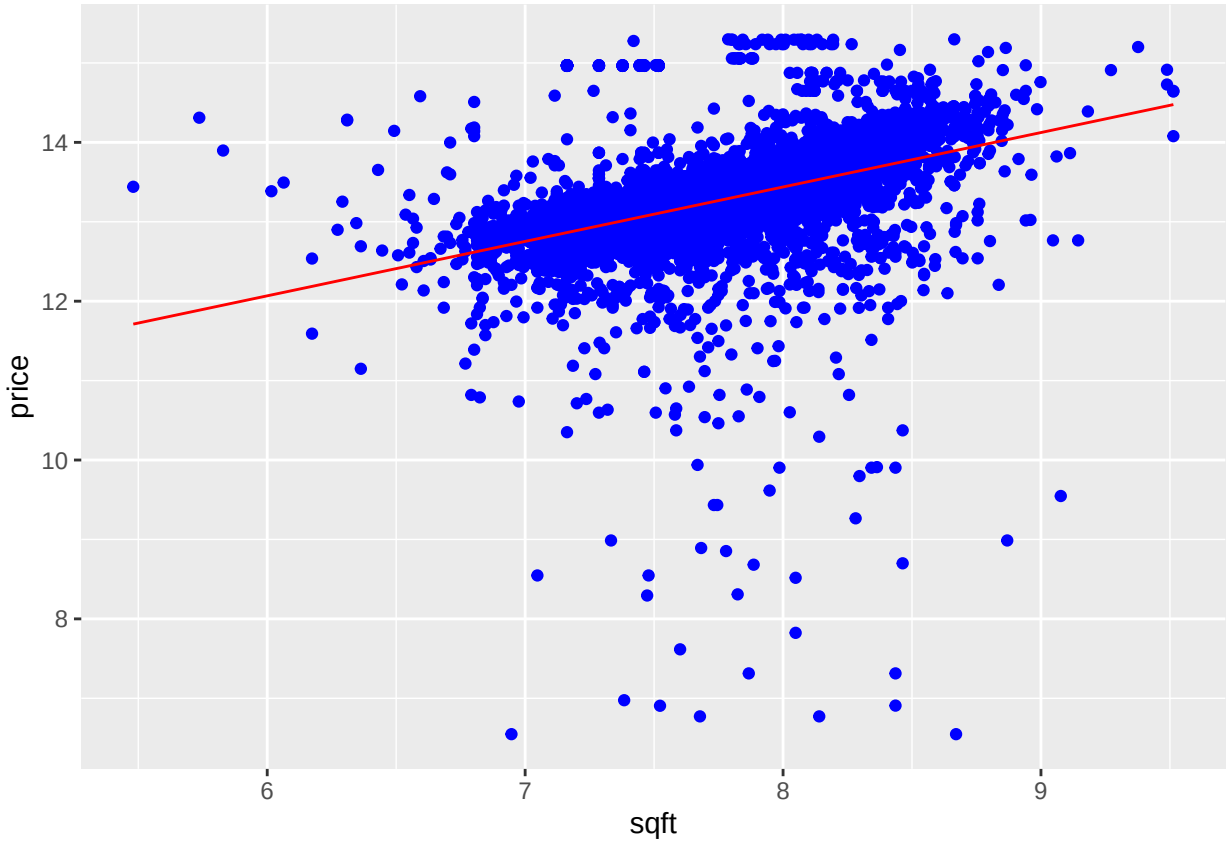



```
##
## Call:
## lm(formula = price ~ sqft, data = housing)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.350 -0.129  0.016  0.152  2.423
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.9548     0.0816    97.5   <2e-16 ***
## sqft          0.6854     0.0105    65.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.45 on 12863 degrees of freedom
## Multiple R-squared:  0.249,  Adjusted R-squared:  0.249
## F-statistic: 4.27e+03 on 1 and 12863 DF,  p-value: <2e-16
```



ii. For the predictors I chose, I started with what I thought would make sense intuitively and then checked the correlations for the variables that made the most sense.

```
##
## Call:
## lm(formula = price ~ sqft, data = housing)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.350 -0.129  0.016  0.152  2.423
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.9548     0.0816    97.5   <2e-16 ***
## sqft          0.6854     0.0105    65.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.45 on 12863 degrees of freedom
## Multiple R-squared:  0.249,  Adjusted R-squared:  0.249
## F-statistic: 4.27e+03 on 1 and 12863 DF,  p-value: <2e-16
```




```
# correlation to see which variables to use as predictors
cor(price, sqft)
```

```
## [1] 0.5
```

```
cor(price, bed)
```

```
## [1] 0.27
```

```
cor(price, bath)
```

```
## [1] 0.39
```

```
cor(price, as.numeric(year1))
```

```
## [1] 0.095
```

```
cor(price, year2)
```

```
## [1] 0.28
```

```
cor(price, as.numeric(housing$Sale_Date))
```

```
## [1] 0.099
```

```
cor(price, housing$sq_ft_outside)
```

```
## [1] -0.0098
```

```
##
## Call:
## lm(formula = price ~ sqft + bed + bath + year2 + as.numeric(year1),
##     data = housing)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.231 -0.124  0.017  0.148  2.447
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -2.53e+01   2.47e+00  -10.24   <2e-16 ***
## sqft               6.32e-01   1.70e-02   37.12   <2e-16 ***
## bed               -1.67e-02   5.96e-03   -2.79   0.0052 **
## bath               1.50e-02   8.68e-03    1.73   0.0832 .
## year2              3.41e-03   2.59e-04   13.15   <2e-16 ***
## as.numeric(year1)  1.34e-02   1.20e-03   11.16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.44 on 12859 degrees of freedom
## Multiple R-squared:  0.269,  Adjusted R-squared:  0.269
## F-statistic:  949 on 5 and 12859 DF,  p-value: <2e-16
```

iii. The r^2 for the simplelog_lm is 0.249. The r^2 for the multiple_lm is 0.269. The adjusted r^2 is also 0.269, which means the cross-validity of this model is good. The model with more variables is slightly better at explaining the variations in the data.

```
##
## Call:
## lm(formula = price ~ sqft, data = housing)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.350 -0.129  0.016  0.152  2.423
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.9548     0.0816    97.5   <2e-16 ***
## sqft          0.6854     0.0105    65.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.45 on 12863 degrees of freedom
## Multiple R-squared:  0.249,  Adjusted R-squared:  0.249
## F-statistic: 4.27e+03 on 1 and 12863 DF,  p-value: <2e-16
```

```
##
## Call:
## lm(formula = price ~ sqft + bed + bath + year2 + as.numeric(year1),
##     data = housing)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.231 -0.124  0.017  0.148  2.447
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -2.53e+01   2.47e+00  -10.24   <2e-16 ***
## sqft               6.32e-01   1.70e-02   37.12   <2e-16 ***
## bed               -1.67e-02   5.96e-03   -2.79   0.0052 **
## bath               1.50e-02   8.68e-03    1.73   0.0832 .
## year2              3.41e-03   2.59e-04   13.15   <2e-16 ***
## as.numeric(year1)  1.34e-02   1.20e-03   11.16   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.44 on 12859 degrees of freedom
## Multiple R-squared:  0.269,  Adjusted R-squared:  0.269
## F-statistic:  949 on 5 and 12859 DF,  p-value: <2e-16
```

iv. In this model, the most important predicter is the square_feet_total_living, followed by the year the house was built, then year the house was sold. The number of bathrooms and number of bedrooms have a similar degree of importance below the rest of the others.

```
##              sqft               bed              bath             year2
##             0.460            -0.028             0.020             0.113
## as.numeric(year1)
##             0.084
```

v. The confidence intervals for square feet, year built, and year sold are all very tight, indicating that these parameters are both representative of the true population and also significant. The confidence interval for bedrooms is more braod but is still significant, while less representative of the population. The confidence interval for bathrooms switches signs, which means that it is very unrepresentative of the population as well as not being a prticularly good predictor of the sale price.

```
##                      2.5 %   97.5 %
## (Intercept)       -30.0965 -20.4265
## sqft                0.5986   0.6654
## bed                -0.0284  -0.0050
```

```
## bath               -0.0020   0.0320
## year2               0.0029   0.0039
## as.numeric(year1)   0.0110   0.0157
```

vi. We can say that the multiple_lm model significantly improved the fit of the model as compared to the simplelog_lm, F(4, 12859) = 89.3, p < 0.001.

```
## Analysis of Variance Table
##
## Model 1: price ~ sqft
## Model 2: price ~ sqft + bed + bath + year2 + as.numeric(year1)
##   Res.Df  RSS Df Sum of Sq    F Pr(>F)
## 1  12863 2587
## 2  12859 2517  4      69.9 89.3 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

vii.

```
# case-wise diagnostics for outliers/influential cases
housing$residuals <- resid(multiple_lm)
housing$stan_resid <- rstandard(multiple_lm)
housing$cooks <- cooks.distance(multiple_lm)
housing$leverage <- hatvalues(multiple_lm)
housing$covariance <- covratio(multiple_lm)
head(housing[c("residuals", "stan_resid", "cooks", "leverage", "covariance")], n=10)
```

```
## # A tibble: 10 x 5
##    residuals stan_resid        cooks leverage covariance
##        <dbl>      <dbl>        <dbl>    <dbl>      <dbl>
##  1   0.109       0.247  0.00000364   0.000359      1.00
##  2   0.00826     0.0187 0.0000000200 0.000345      1.00
##  3  -0.0218     -0.0492 0.000000154  0.000381      1.00
##  4   0.0631      0.143  0.00000155   0.000456      1.00
##  5  -0.0304     -0.0687 0.000000321  0.000408      1.00
##  6  -1.49       -3.36   0.000864     0.000459      0.996
##  7   0.333       0.752  0.0000608    0.000645      1.00
##  8   0.209       0.472  0.0000197    0.000529      1.00
##  9  -0.121      -0.273  0.00000868   0.000697      1.00
## 10   0.153       0.345  0.0000131    0.000662      1.00
```

viii.

```
# large residuals
housing$large_resid <- housing$stan_resid > 2 | housing$stan_resid < -2
head(housing$large_resid, n = 30)
```

```
##     1     2     3     4     5     6     7     8     9    10    11    12    13
```

```
## FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##    14    15    16    17    18    19    20    21    22    23    24    25    26
##  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
##    27    28    29    30
## FALSE FALSE FALSE FALSE
```

ix.

```
# sum of large residuals
sum(housing$large_resid)
```

```
## [1] 475
```

x.

```
# which variables have large residuals
housing[housing$large_resid, c("log_price", "log_sqft", "Sale_Year", "bedrooms", "bathrooms", "year_bui
```

```
## # A tibble: 475 x 7
##    log_price log_sqft Sale_Year bedrooms bathrooms year_built stan_resid
##        <dbl>    <dbl> <chr>        <dbl>     <dbl>      <dbl>      <dbl>
##  1      12.1     8.33 2006             4      3.25       2005      -3.36
##  2      12.0     7.52 2006             3      2          2011      -2.50
##  3      12.5     8.50 2006             4      4.5        2007      -2.84
##  4      11.9     7.57 2006             3      2.5        2003      -2.76
##  5      14.1     6.49 2006             0      1          1955       4.15
##  6      12.3     8.25 2006             0      0          2008      -2.83
##  7      12.9     8.67 2006             5      4.5        2008      -2.17
##  8      12.7     8.35 2006             4      3.25       2015      -2.25
##  9      14.2     6.80 2006             2      1          1918       4.16
## 10      12.0     8.46 2006             4      4          2014      -3.92
## # ... with 465 more rows
```

xi. None of the cases with large standard residuals have a Cook's distance greater than 1, so none of the observations have an undue influence on the model. Some of the observations have large leverage values, which means that these observations have a large influence on the outcome variables. There are also a great many observations with large covariance ratios. However, since none of these observations have a Cook's distance greater than 1, there is no need to delete them or exclude them from the model, but they can be studied in order to try to understand why they did not fit the model.

```
# leverage, Cook's distance, covariance ratios, k = 5
housing[housing$large_resid, c("leverage", "cooks", "covariance")]
```

```
## # A tibble: 475 x 3
##    leverage    cooks covariance
##       <dbl>    <dbl>      <dbl>
##  1 0.000459 0.000864      0.996
##  2 0.000463 0.000482      0.998
##  3 0.00106  0.00143       0.998
##  4 0.000348 0.000442      0.997
##  5 0.00198  0.00568       0.994
##  6 0.00591  0.00795       1.00
##  7 0.000962 0.000759      0.999
##  8 0.000509 0.000431      0.999
##  9 0.00199  0.00575       0.994
## 10 0.000706 0.00180       0.994
## # ... with 465 more rows
```

```
housing$cooksprob <- housing$cooks > 1
avelev <- 6 / 12865
housing$badlev <- housing$leverage > (3 * avelev)
highcov <- 1 + ((3 * 6) / 12865)
lowcov <- 1 - ((3 * 6) / 12865)
housing$covissue <- housing$covariance > highcov | housing$covariance < lowcov
sum(housing$covissue)
```

```
## [1] 723
```

```
sum(housing$badlev)
```

```
## [1] 240
```

```
sum(housing$cooksprob)
```

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

```
housing[housing$large_resid, c("badlev", "cooksprob", "covissue")]
```

```
## # A tibble: 475 x 3
##    badlev cooksprob covissue
##    <lgl>  <lgl>     <lgl>
##  1 FALSE  FALSE     TRUE
##  2 FALSE  FALSE     TRUE
##  3 FALSE  FALSE     TRUE
##  4 FALSE  FALSE     TRUE
##  5 TRUE   FALSE     TRUE
##  6 TRUE   FALSE     TRUE
##  7 FALSE  FALSE     FALSE
##  8 FALSE  FALSE     FALSE
##  9 TRUE   FALSE     TRUE
## 10 FALSE  FALSE     TRUE
## # ... with 465 more rows
```

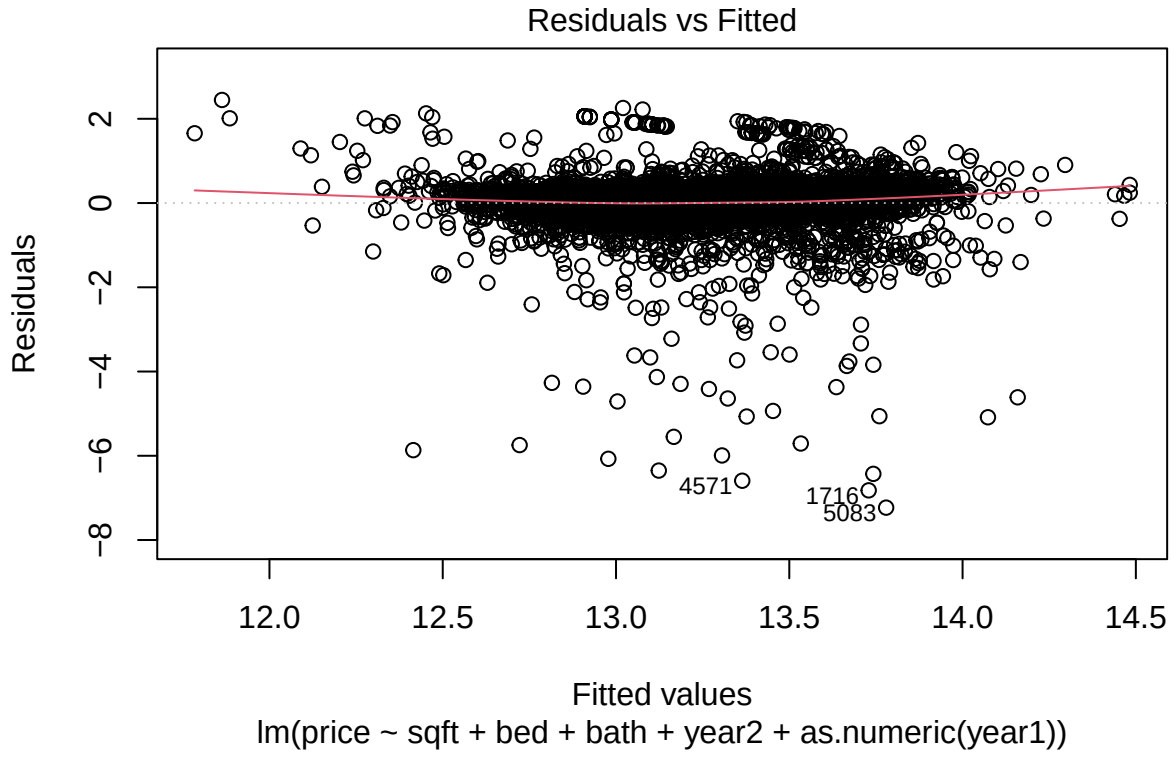
xii. The model does not meet the assumption of independence.

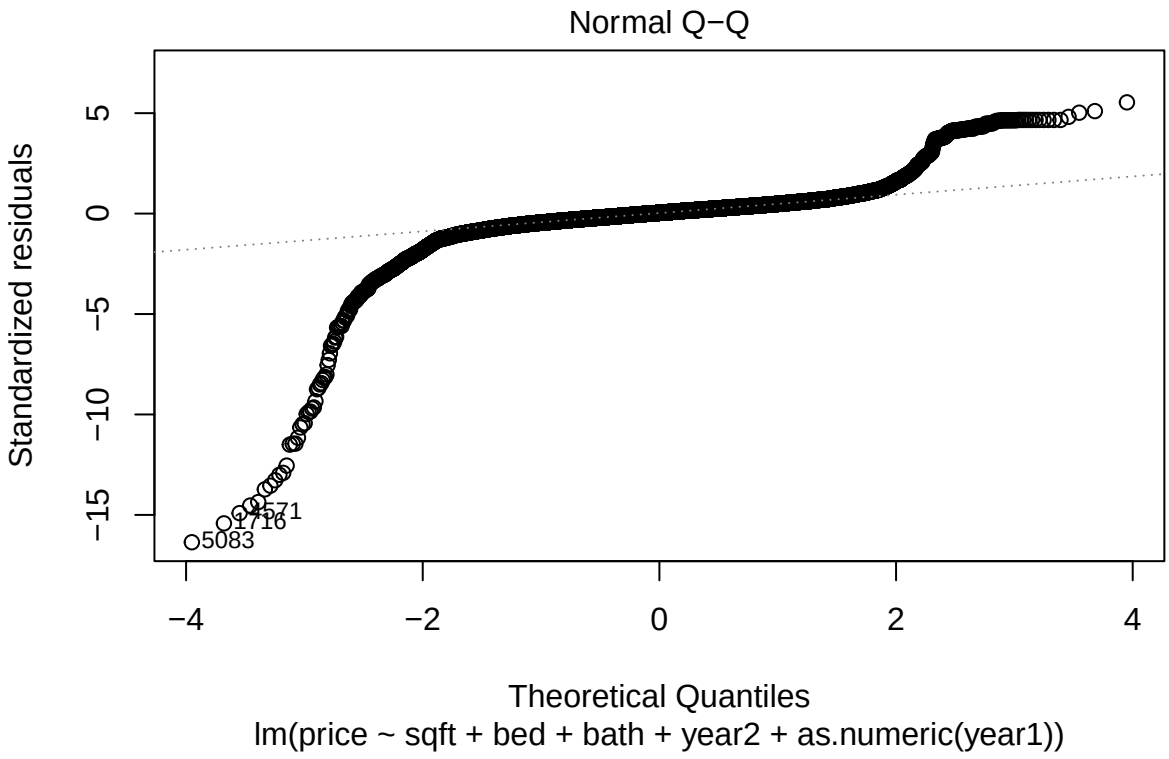
```
##  lag Autocorrelation D-W Statistic p-value
##    1            0.33           1.3       0
##  Alternative hypothesis: rho != 0
```

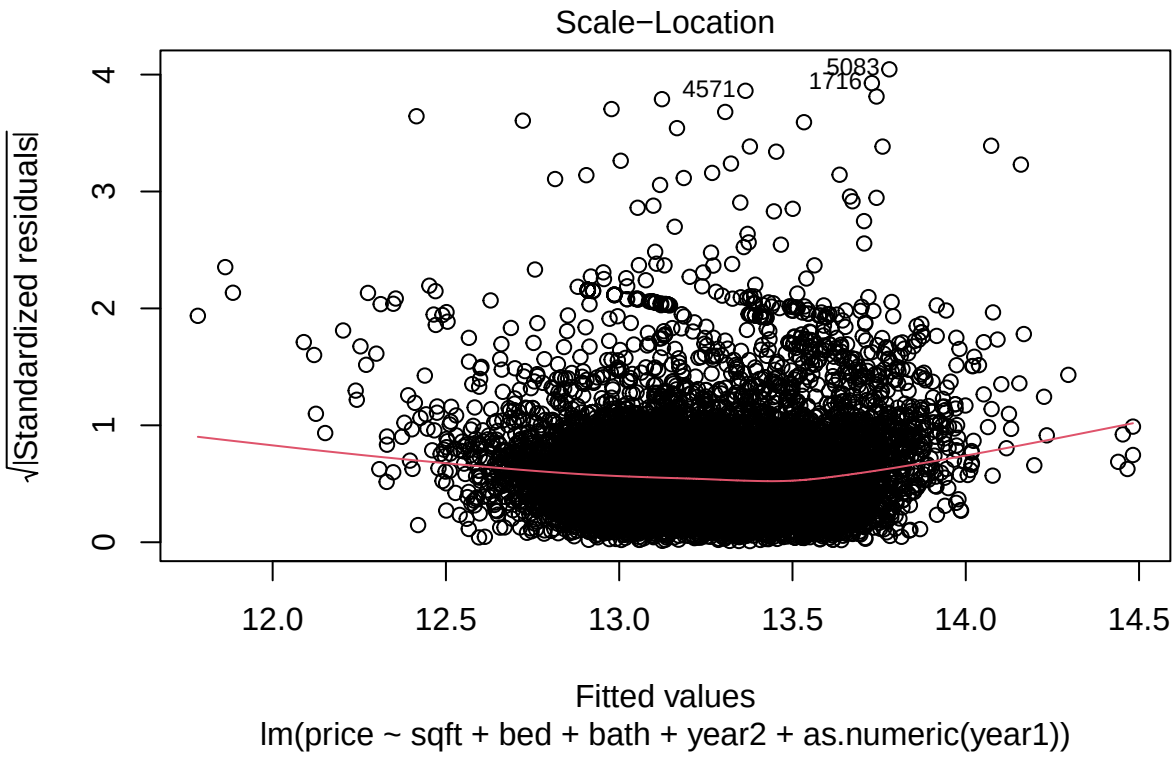
xiii. The VIF, then the mean of the VIF, then the tolerance is listed below. The average VIF is larger than 1 (at 1.8) which could suggest some bias in the model but since none of the VIFs are larger than 10 and the tolerances are all greater than 0.2, there is likely not too much collinearity within the data.

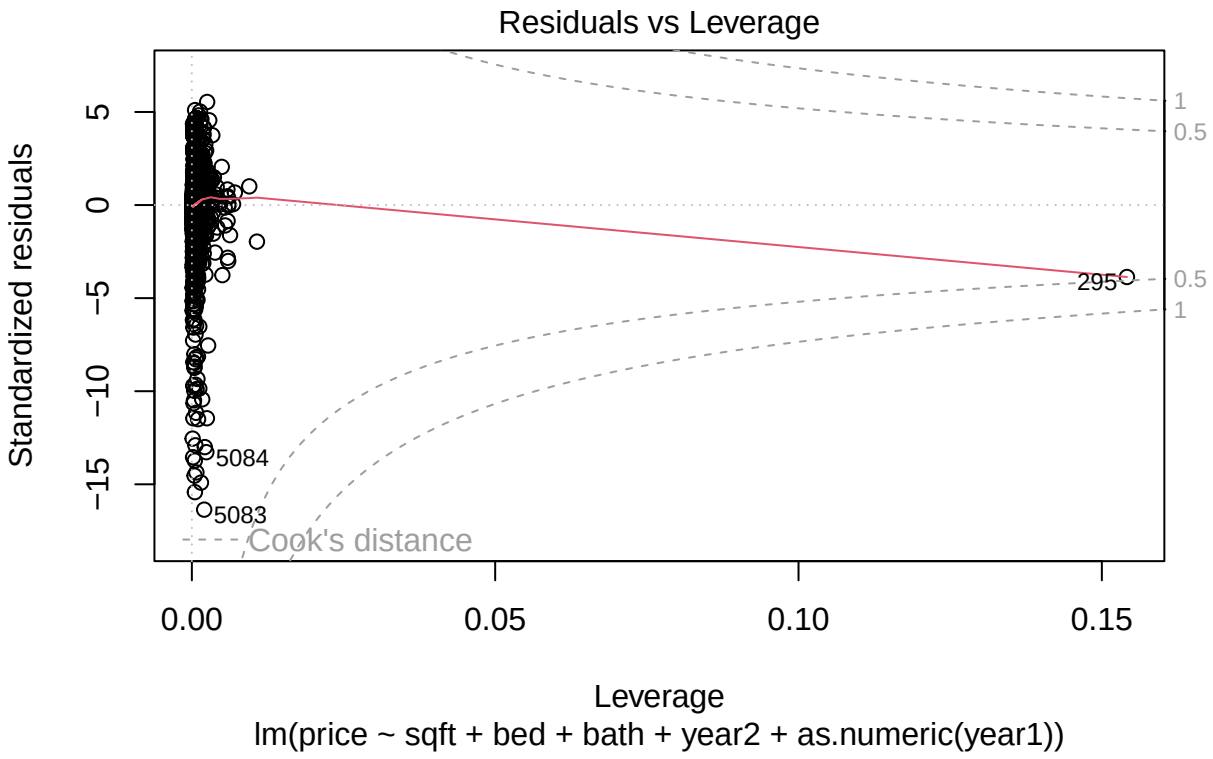
```
##              sqft               bed              bath             year2
##               2.7               1.8               2.4               1.3
## as.numeric(year1)
##               1.0

## [1] 1.8

##              sqft               bed              bath             year2
##              0.37              0.56              0.42              0.76
## as.numeric(year1)
##              1.00
```

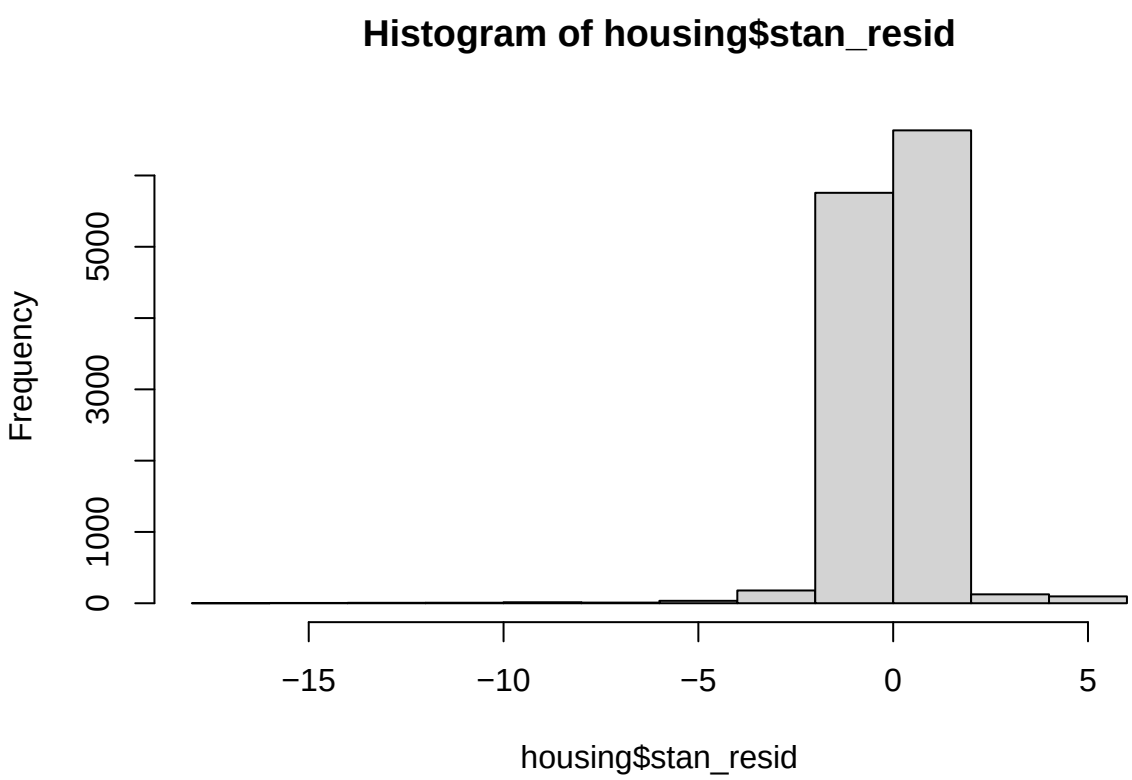
xiv. There is some evidence of heteroscedasticity in the data, and the residuals are not normally distributed, with an over-dispersed distribution. The histogram proves there is an obvious left-skew of the standardized residuals.




Normal Q−Q
Standardized residuals
5
0
−5
−10
−15
−4
−2
0
2
4
5083
1716
571
Theoretical Quantiles
lm(price ~ sqft + bed + bath + year2 + as.numeric(year1))


Scale−Location
√|Standardized residuals|
0
1
2
3
4
5083
1716
4571
12.0
12.5
13.0
13.5
14.0
14.5
Fitted values
lm(price ~ sqft + bed + bath + year2 + as.numeric(year1))


Residuals vs Leverage
Standardized residuals
5
0
−5
−10
−15
1
0.5
0.5
1
295
5084
5083
Cook's distance
0.00
0.05
0.10
0.15
Leverage
lm(price ~ sqft + bed + bath + year2 + as.numeric(year1))



xv. The regression model is not entirely unbiased- there is some evidence of that the residuals are not random, homoscedasticity is not met. The model also does not meet the assumption of independence. Therefore, we are unable to make accurate assumptions or predictions about the population based off the model. Our predictions would have a propensity of being erroneous or innacurate when applied to the overall population.