Untitled

Reenie Christudass

2022-07-31

Load Libraries

```
library(readxl)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.1
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.1
```

```
##
## 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(caTools)
```

```
## Warning: package 'caTools' was built under R version 4.2.1
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.1
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
```

```
## v tibble  3.1.7     v purrr   0.3.4
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## Warning: package 'tidyr' was built under R version 4.2.1

## Warning: package 'readr' was built under R version 4.2.1

## Warning: package 'forcats' was built under R version 4.2.1

## -- Conflicts --------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.2.1

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.2.1

##
## Attaching package: 'car'
##
## The following object is masked from 'package:purrr':
##
##     some
##
## The following object is masked from 'package:dplyr':
##
##     recode
```

Read the data

```
## Set the working directory to the root of your DSC 520 directory
setwd("C:/Users/chris/dsc520/data")
```

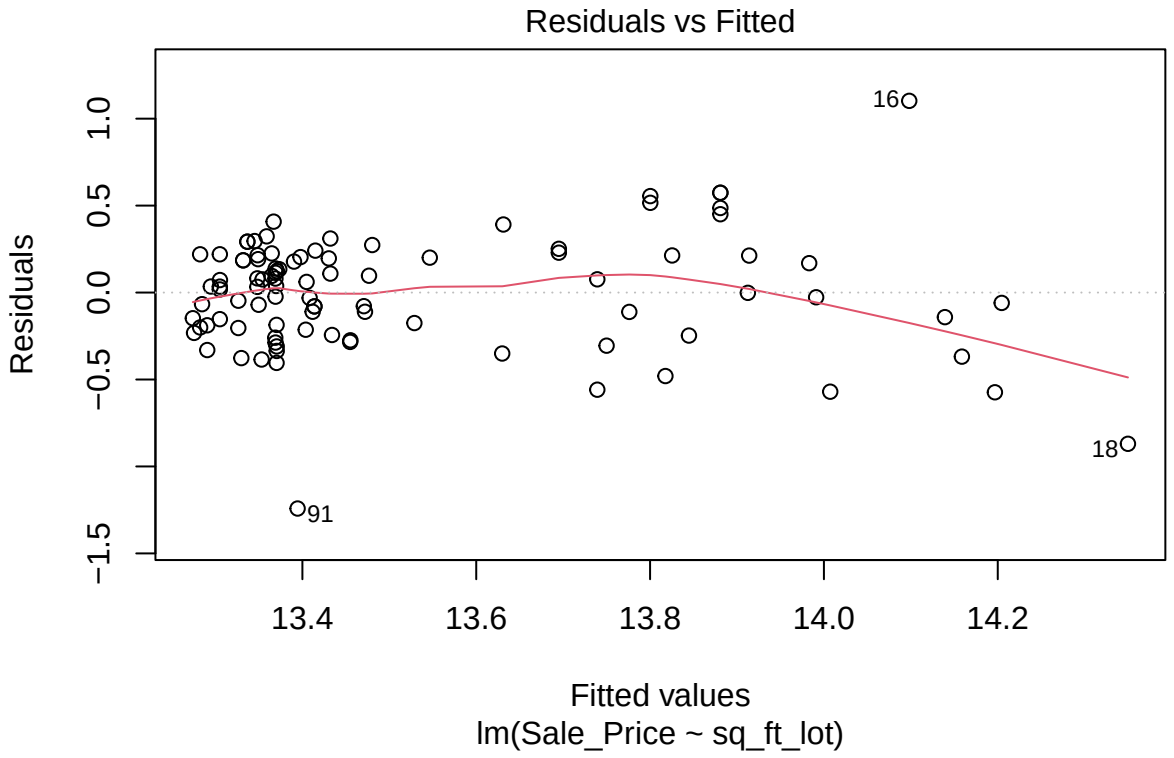
```
df <- read_excel("C:/Users/chris/dsc520/data/week-7-housing.xlsx")
```

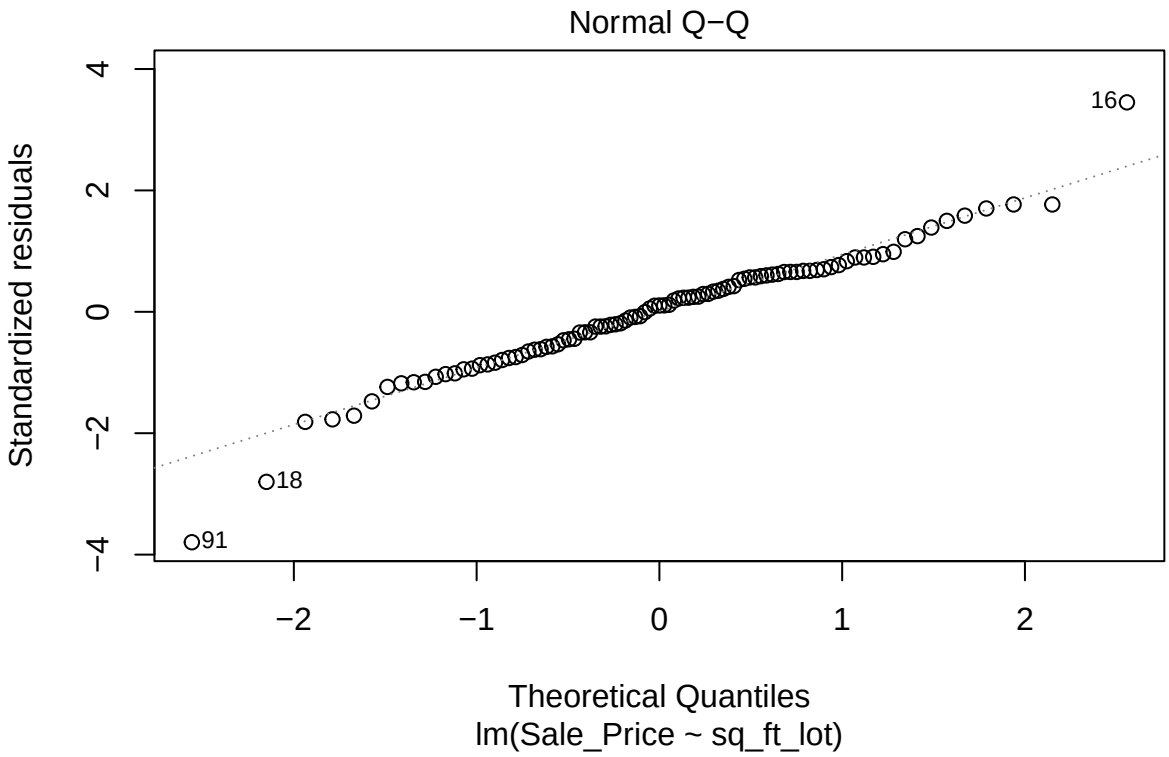
```
## Transformations ( Filter the data for a particular year)
names(df) <- sub(" ", "_", names(df))
df <- subset(df, year_built == "2000")
head(df)
```

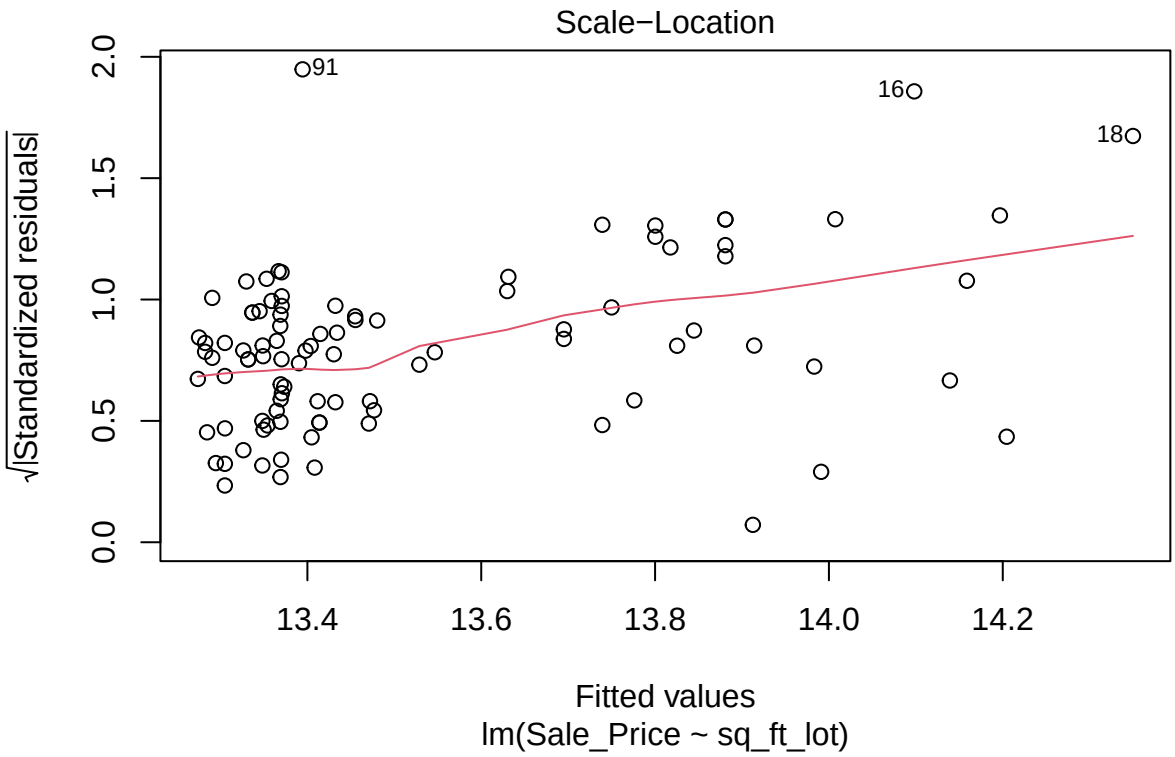
```
## # A tibble: 6 x 24
##   Sale_Date           Sale_Price sale_re~1 sale_~2 sale_~3 sitet~4 addr_~5  zip5
##   <dttm>                   <dbl>     <dbl>   <dbl> <chr>   <chr>   <chr>   <dbl>
## 1 2006-02-09 00:00:00     647500        12       3 <NA>    R1      9206 1~ 98052
## 2 2006-02-15 00:00:00    1390000         1       3 <NA>    R1      19656 ~ 98053
## 3 2006-02-24 00:00:00     532000         1       3 <NA>    R1      10119 ~ 98053
## 4 2006-06-06 00:00:00    1650000         1       3 <NA>    R1      2005 2~ 98074
## 5 2006-07-19 00:00:00     804000        14       3 22      R1      4521 2~ 98053
```

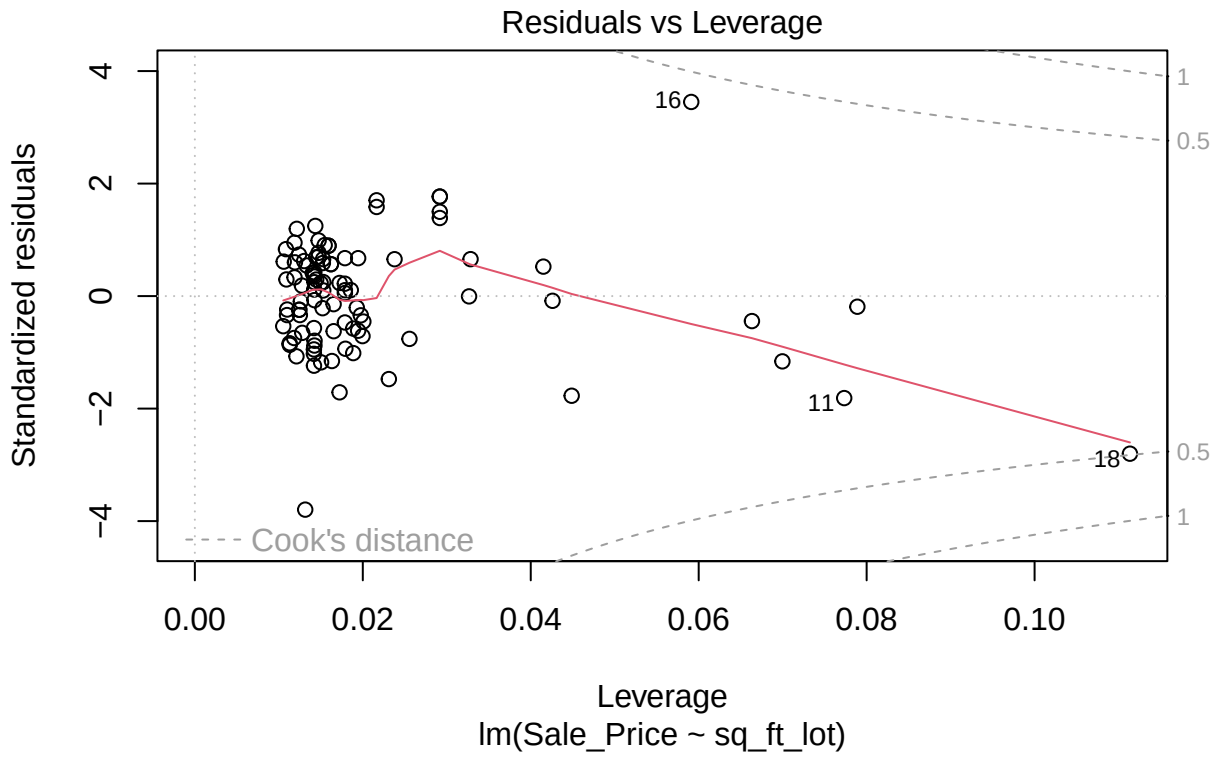
```
## 6 2006-09-12 00:00:00     777000         1       3 <NA>    R1      7228 1~ 98052
## # ... with 16 more variables: ctyname <chr>, postalctyn <chr>, lon <dbl>,
## #   lat <dbl>, building_grade <dbl>, square_feet_total_living <dbl>,
## #   bedrooms <dbl>, bath_full_count <dbl>, bath_half_count <dbl>,
## #   bath_3qtr_count <dbl>, year_built <dbl>, year_renovated <dbl>,
## #   current_zoning <chr>, sq_ft_lot <dbl>, prop_type <chr>, present_use <dbl>,
## #   and abbreviated variable names 1: sale_reason, 2: sale_instrument,
## #   3: sale_warning, 4: sitetype, 5: addr_full
## # i Use `colnames()` to see all variable names
```

```r
# Select out data of interest
d <- df %>% select(Sale_Price, sq_ft_lot)

d <- log(d)

# Fit the model
model1 <- lm(Sale_Price ~ sq_ft_lot, data = d)
plot(model1)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Sale_Price ~ sq_ft_lot)
16
18
91
4
2
0
−2
−4
−2
−1
0
1
2


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
91
16
18
13.4
13.6
13.8
14.0
14.2
Fitted values
lm(Sale_Price ~ sq_ft_lot)




```
summary(model1)
```

```
##
## Call:
## lm(formula = Sale_Price ~ sq_ft_lot, data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.24192 -0.20235  0.03406  0.20824  1.10238
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.48253    0.26206  43.816  < 2e-16 ***
## sq_ft_lot    0.22082    0.02807   7.867 6.41e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3293 on 93 degrees of freedom
## Multiple R-squared:  0.3996, Adjusted R-squared:  0.3931
## F-statistic: 61.89 on 1 and 93 DF,  p-value: 6.41e-12
```

```
# Check the assumption of independence is using the Durbin Watson test
# The Durbin Watson (DW) statistic is used as a test for checking auto correlation in the residuals of
durbinWatsonTest(model1)
```

```
##  lag Autocorrelation D-W Statistic p-value
```

```
##    1    -0.003792149      2.007405   0.986
##  Alternative hypothesis: rho != 0
```

```
#examine correlation between variable
correlation <- cor(d$Sale_Price, d$sq_ft_lot)
print(paste("correlation between variable:",correlation))
```

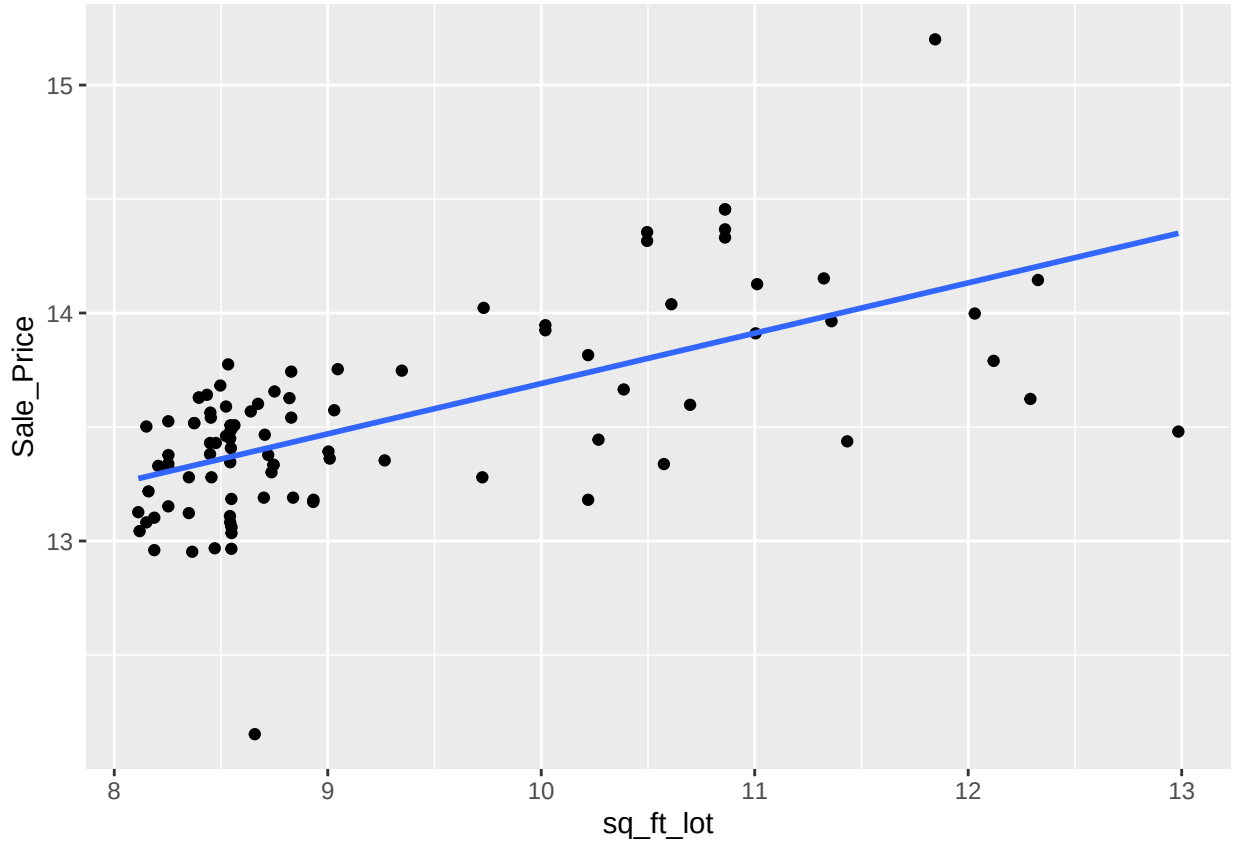
```
## [1] "correlation between variable: 0.632132632626453"
```

```
# Obtain predicted and residual values
d$predicted_LM <- predict(model1)
d$residuals_LM <- residuals(model1)
d
```

```
## # A tibble: 95 x 4
##    Sale_Price sq_ft_lot predicted_LM residuals_LM
##         <dbl>     <dbl>        <dbl>        <dbl>
##  1       13.4      8.45         13.3       0.0327
##  2       14.1     12.3          14.2      -0.0597
##  3       13.2      8.55         13.4      -0.186
##  4       14.3     10.5          13.8       0.516
##  5       13.6     10.7          13.8      -0.247
##  6       13.6      8.45         13.3       0.215
##  7       14.0      9.73         13.6       0.391
##  8       13.6      8.64         13.4       0.178
##  9       13.8     10.2          13.7       0.0761
## 10       13.3      8.25         13.3       0.0341
## # ... with 85 more rows
## # i Use `print(n = ...)` to see more rows
```

```
ggplot(d, aes(x=sq_ft_lot, y=Sale_Price)) + geom_point() + geom_smooth(method = lm, se = FALSE)
```

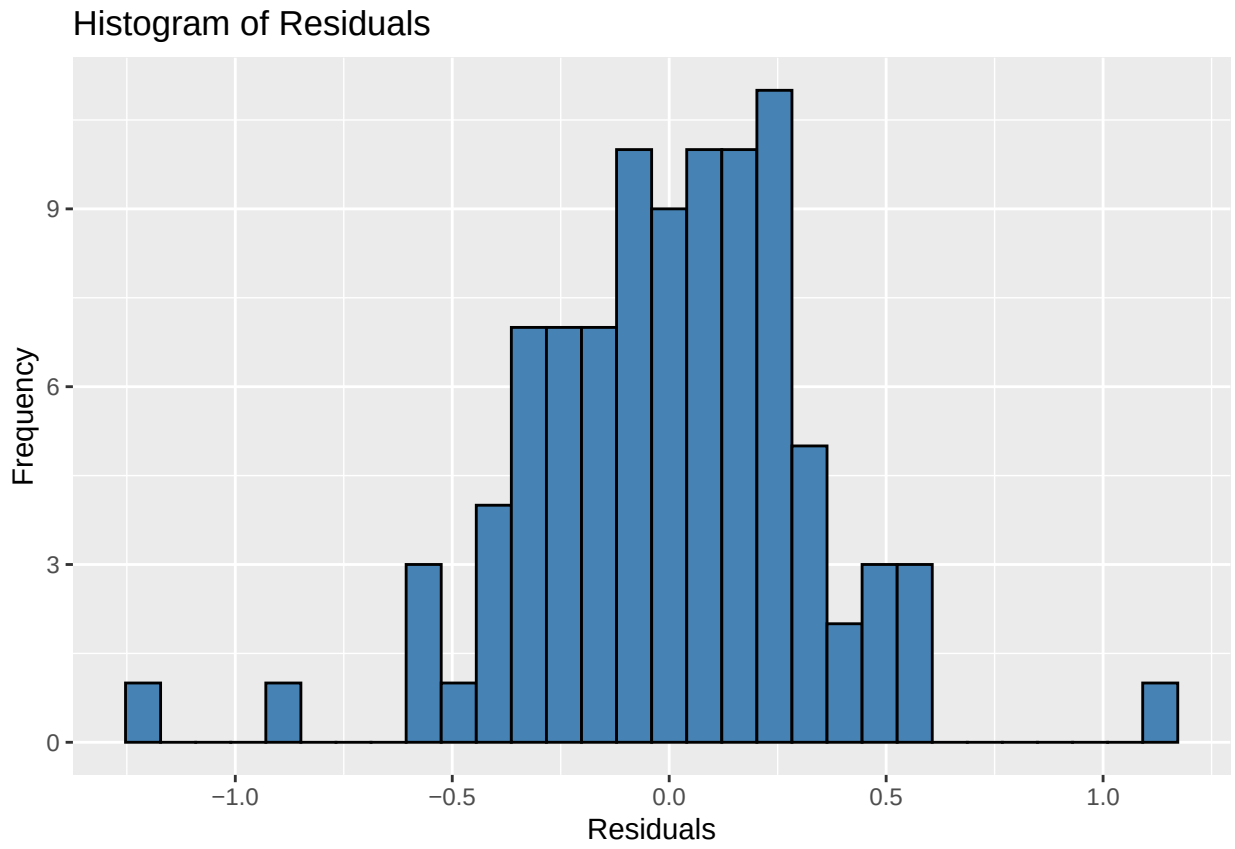
```
## `geom_smooth()` using formula 'y ~ x'
```




```
#create histogram of residuals
ggplot(data = d, aes(x = d$residuals_LM )) +
    geom_histogram(fill = 'steelblue', color = 'black') +
    labs(title = 'Histogram of Residuals', x = 'Residuals', y = 'Frequency')
```

```
## Warning: Use of ‘d$residuals_LM‘ is discouraged. Use ‘residuals_LM‘ instead.
```

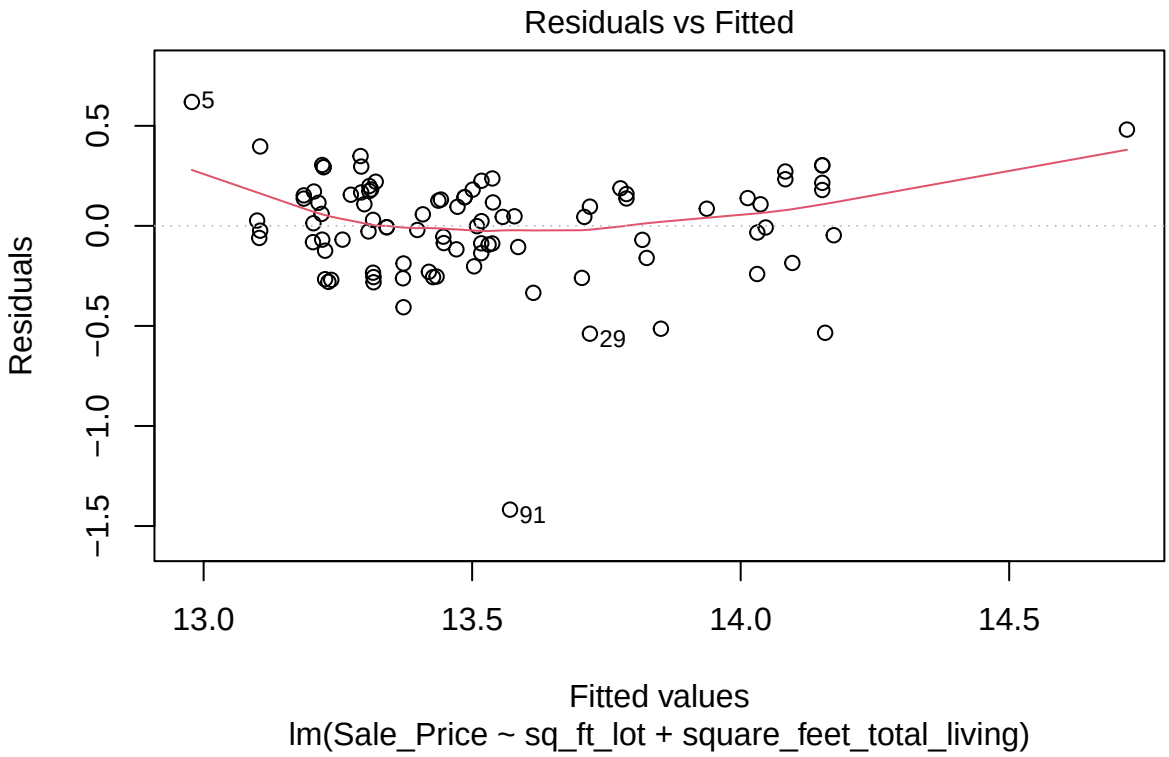
```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```

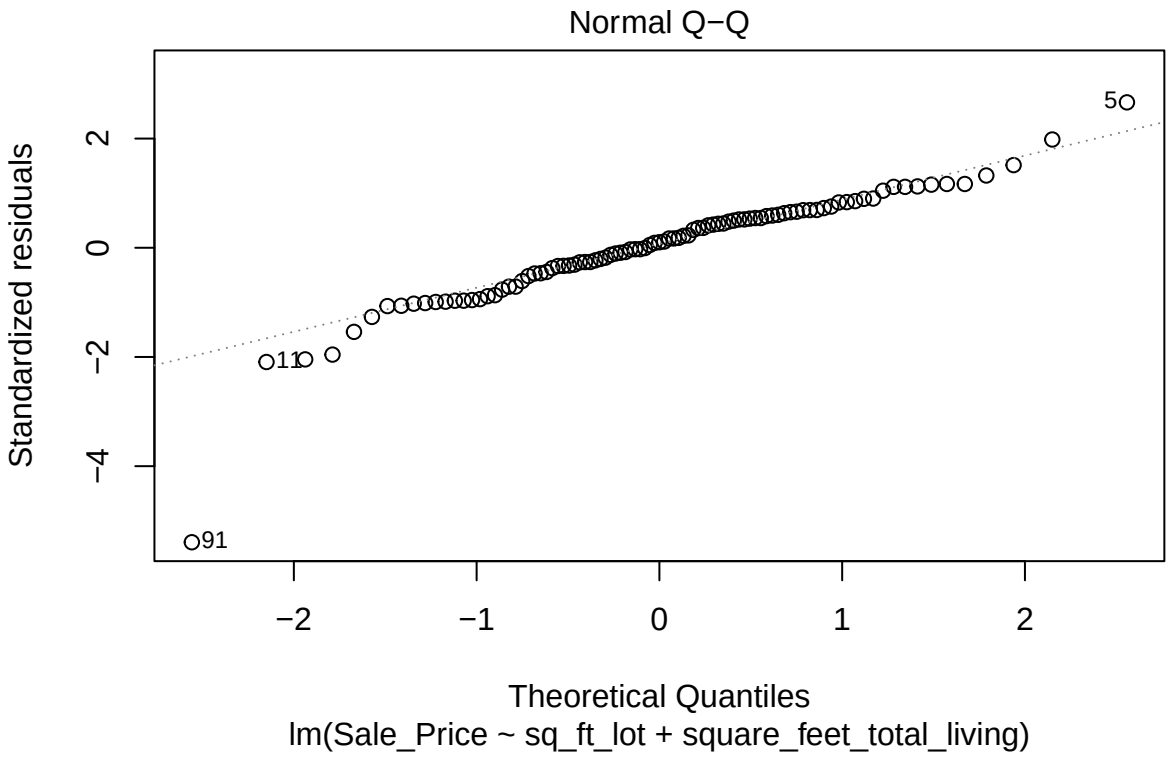


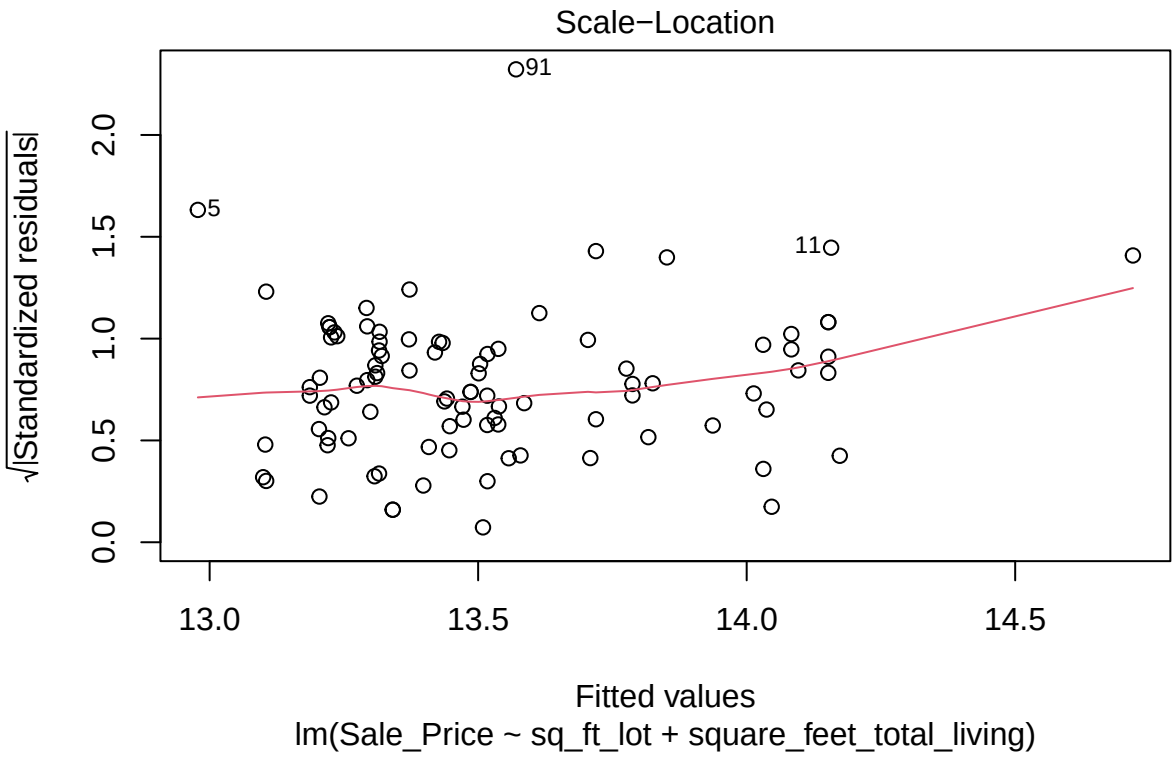

```
# Select out data of interest
d1 <- df %>% select(Sale_Price, sq_ft_lot, square_feet_total_living )

d1 <- log(d1)

# check for multicollinearity
cor(d1, method="pearson")
```

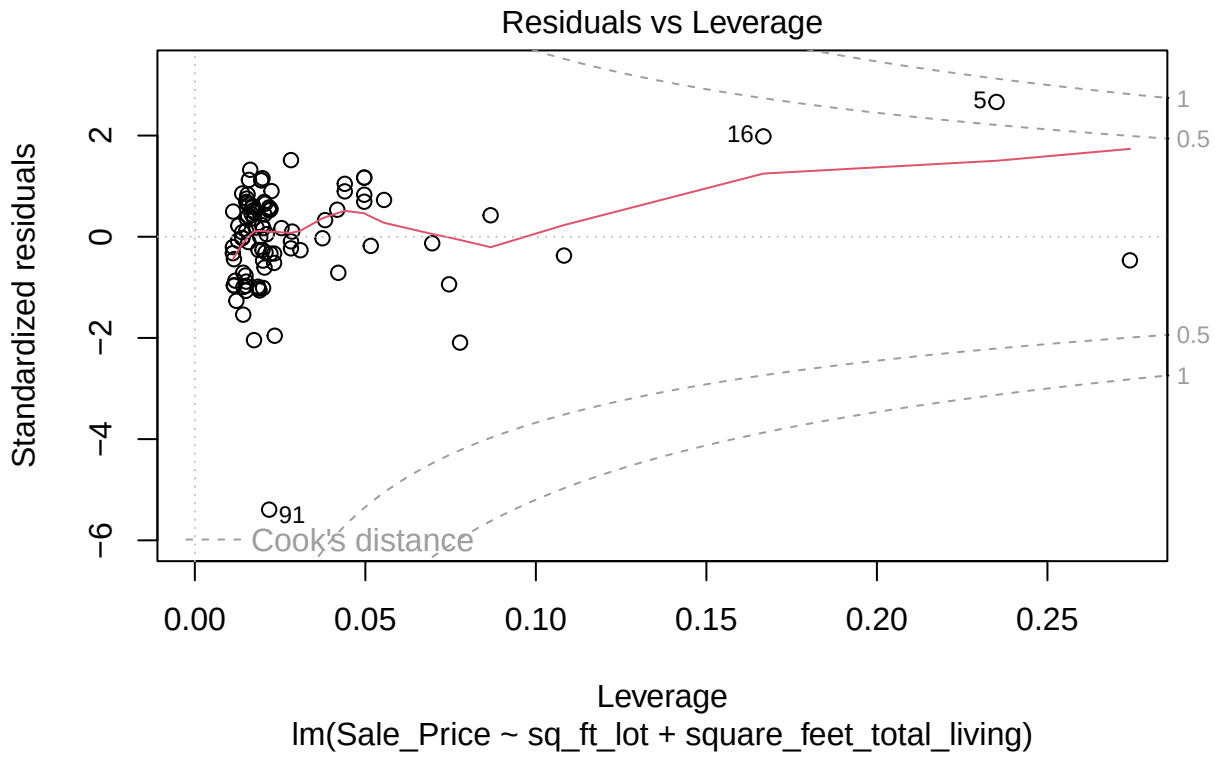
```
##                          Sale_Price sq_ft_lot square_feet_total_living
## Sale_Price                1.0000000 0.6321326                0.6789385
## sq_ft_lot                 0.6321326 1.0000000                0.4062814
## square_feet_total_living  0.6789385 0.4062814                1.0000000
```

```
# Fit the model
model2 <- lm(Sale_Price ~ sq_ft_lot + square_feet_total_living, data = d1)
plot(model2)
```


Residuals vs Fitted
Residuals
0.5
0.0
−0.5
−1.0
−1.5
5
29
91
13.0
13.5
14.0
14.5
Fitted values
lm(Sale_Price ~ sq_ft_lot + square_feet_total_living)


Normal Q−Q
Standardized residuals
2
0
−2
−4
5
11
91
−2
−1
0
1
2
Theoretical Quantiles
lm(Sale_Price ~ sq_ft_lot + square_feet_total_living)


Scale−Location
91
5
11
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
13.0
13.5
14.0
14.5
Fitted values
lm(Sale_Price ~ sq_ft_lot + square_feet_total_living)



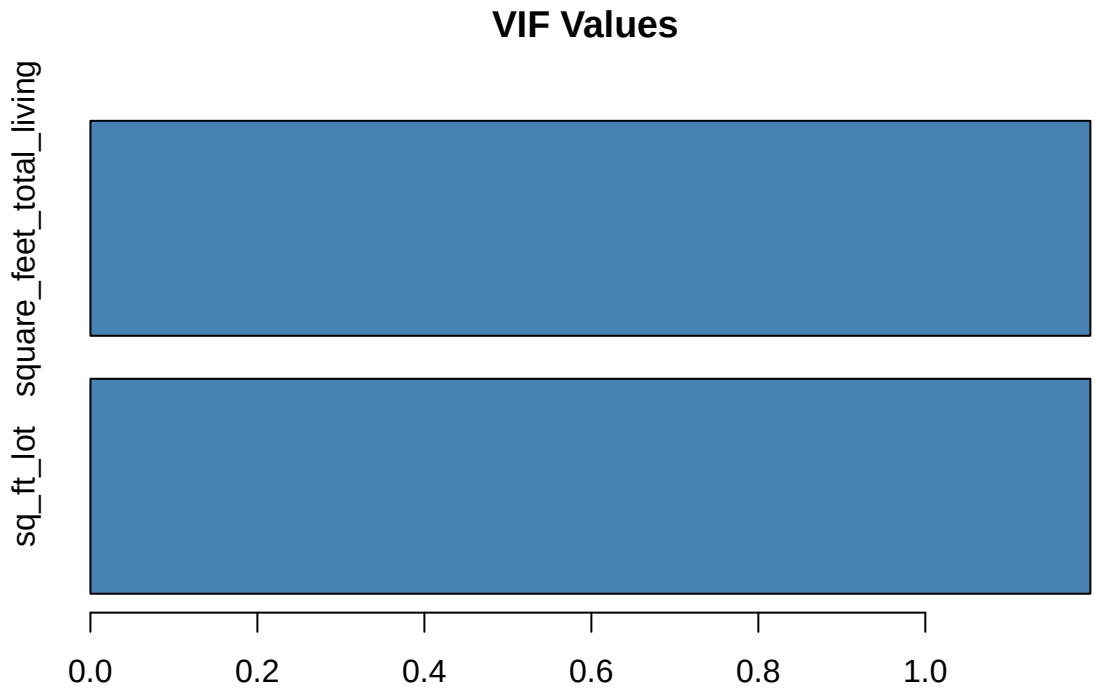

```
summary(model2)
```

```
##
## Call:
## lm(formula = Sale_Price ~ sq_ft_lot + square_feet_total_living,
##     data = d1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.41796 -0.12059  0.02671  0.16316  0.61930
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)               7.43406    0.60648  12.258  < 2e-16 ***
## sq_ft_lot                 0.14907    0.02480   6.012 3.65e-08 ***
## square_feet_total_living  0.58864    0.08264   7.123 2.30e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2658 on 92 degrees of freedom
## Multiple R-squared:  0.613,  Adjusted R-squared:  0.6046
## F-statistic: 72.86 on 2 and 92 DF,  p-value: < 2.2e-16
```

```
#create vector of VIF values
vif_values <- vif(model2)
print(vif_values)
```

```
##                sq_ft_lot square_feet_total_living
##                 1.197697                 1.197697
```

```
#create horizontal bar chart to display each VIF value
barplot(vif_values, main = "VIF Values", horiz = TRUE, col = "steelblue")

#add vertical line at 5
abline(v = 5, lwd = 3, lty = 2)
```




```
# Check the assumption of independence is using the Durbin Watson test
# The Durbin Watson (DW) statistic is used as a test for checking auto correlation in the residuals of
durbinWatsonTest(model2)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1      0.06219242      1.867343   0.498
##  Alternative hypothesis: rho != 0
```

```
#  ANOVA test to estimate the effect of each feature on the variances with the anova() function
anova(model2)
```

```
## Analysis of Variance Table
##
```

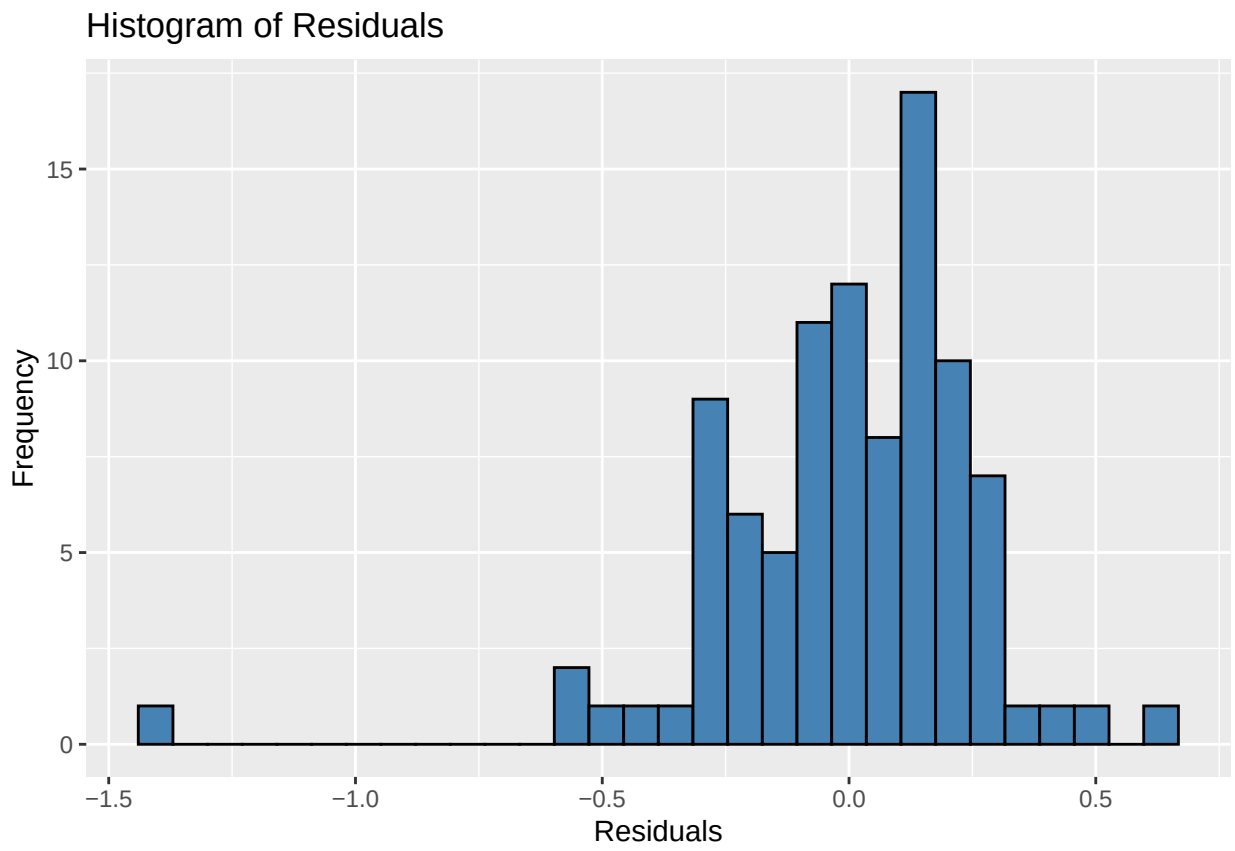
```
## Response: Sale_Price
##                           Df Sum Sq Mean Sq F value    Pr(>F)
## sq_ft_lot                  1 6.7131  6.7131  94.993 7.795e-16 ***
## square_feet_total_living  1 3.5852  3.5852  50.732 2.300e-10 ***
## Residuals                 92 6.5016  0.0707
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Obtain predicted and residual values
d1$predicted_MLM <- predict(model2)
d1$residuals_MLM <- residuals(model2)
head(d1)
```

```
## # A tibble: 6 x 5
##   Sale_Price sq_ft_lot square_feet_total_living predicted_MLM residuals_MLM
##        <dbl>     <dbl>                    <dbl>         <dbl>         <dbl>
## 1       13.4      8.45                     8.19          13.5        -0.136
## 2       14.1     12.3                      8.10          14.0         0.108
## 3       13.2      8.55                     7.92          13.4        -0.188
## 4       14.3     10.5                      8.64          14.1         0.233
## 5       13.6     10.7                      6.71          13.0         0.619
## 6       13.6      8.45                     8.06          13.4         0.126
```

```
#create histogram of residuals
ggplot(data = d, aes(x = d1$residuals_MLM )) +
    geom_histogram(fill = 'steelblue', color = 'black') +
    labs(title = 'Histogram of Residuals', x = 'Residuals', y = 'Frequency')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#calculate residual sum of squares for both models
deviance(model1)
```

```
## [1] 10.0868
```

```
deviance(model2)
```

```
## [1] 6.501586
```

```
#calculate R-squared for both models
summary(model1)$r.squared
```

```
## [1] 0.3995917
```

```
summary(model2)$r.squared
```

```
## [1] 0.6129985
```

```
#Influence Measures for multiple regressions
##inf_measures <- influence.measures(model2)
##head(inf_measures)
```

```
#Cook's D Bar
install.packages("olsrr", repos="http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/chris/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
## package 'olsrr' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'olsrr'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying C:
## \Users\chris\AppData\Local\R\win-library\4.2\00LOCK\olsrr\libs\x64\olsrr.dll
## to C:\Users\chris\AppData\Local\R\win-library\4.2\olsrr\libs\x64\olsrr.dll:
## Permission denied
```

```
## Warning: restored 'olsrr'
```

```
##
## The downloaded binary packages are in
##  C:\Users\chris\AppData\Local\Temp\RtmpO8Ulnp\downloaded_packages
```

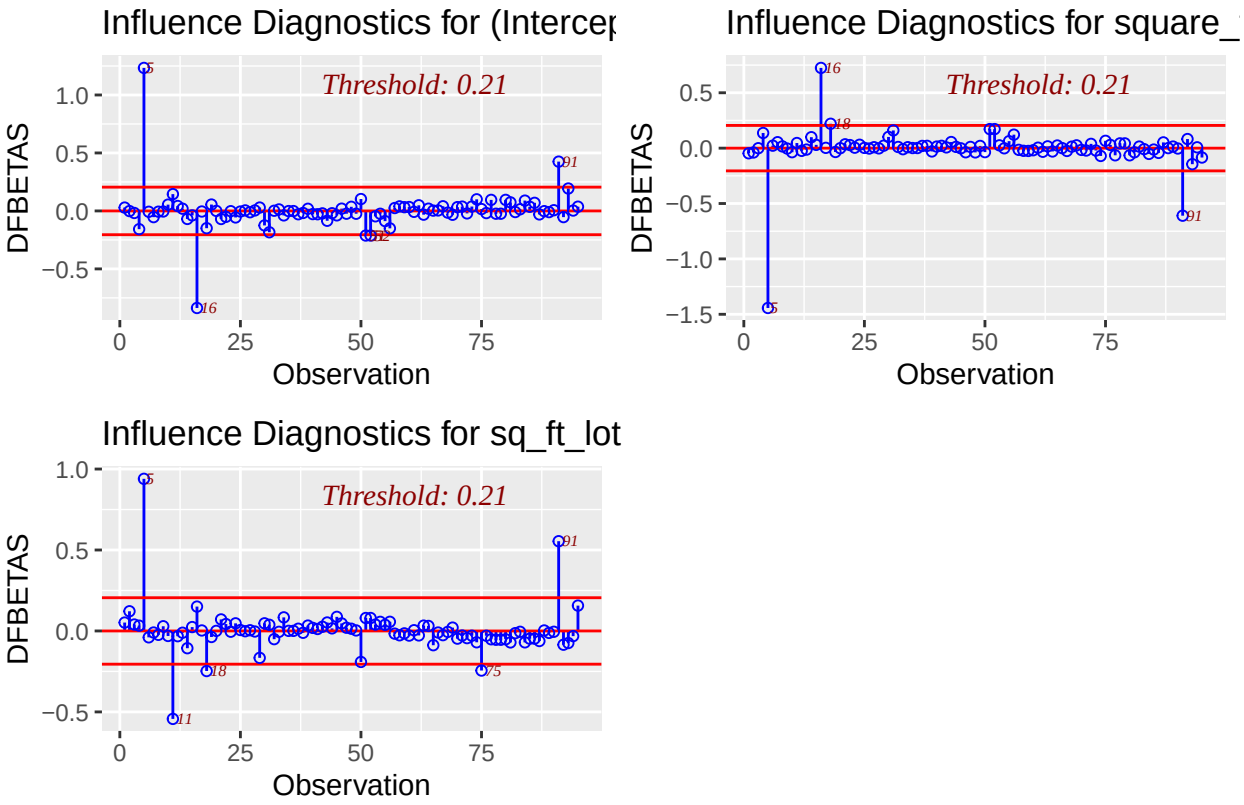
```
library(olsrr)
```

```
## Warning: package 'olsrr' was built under R version 4.2.1
```

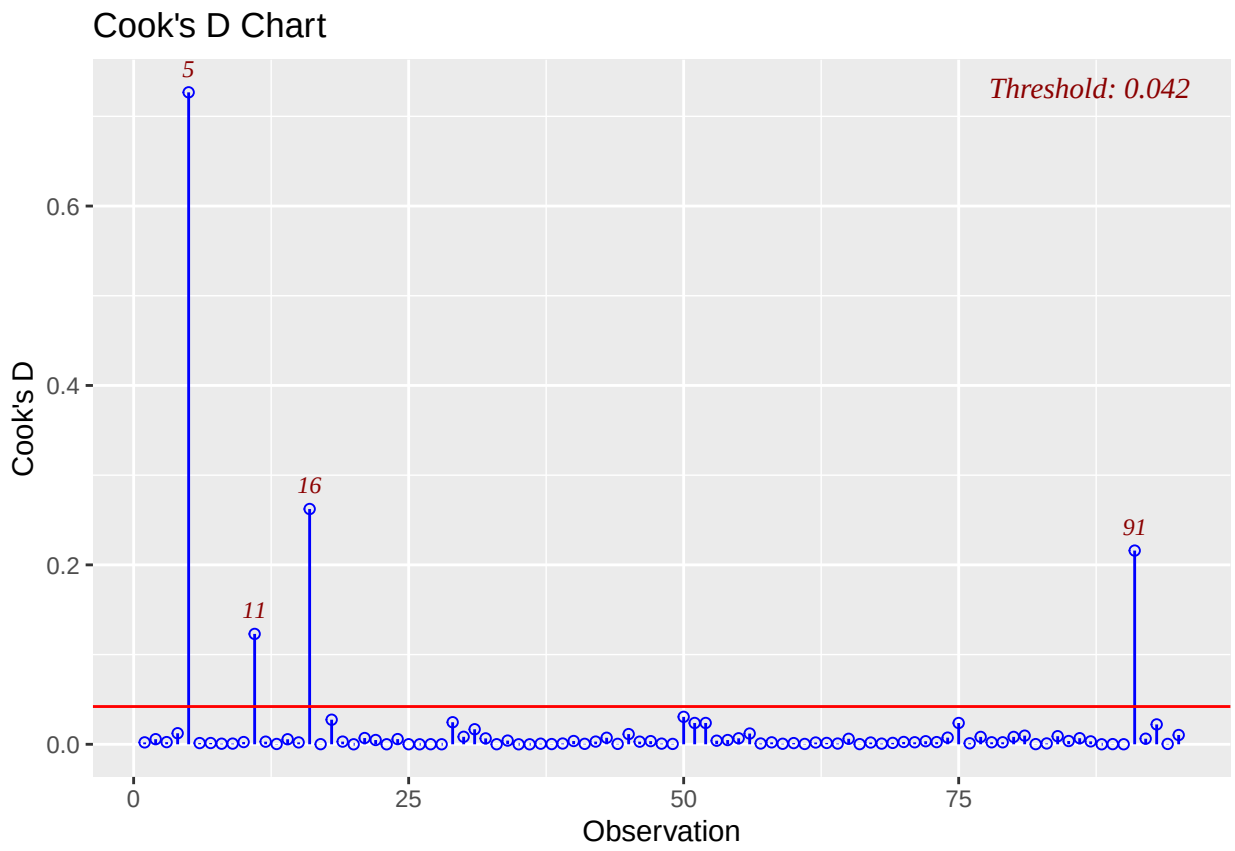
```
##
## Attaching package: 'olsrr'
```

```
## The following object is masked from 'package:datasets':
##
##     rivers
```

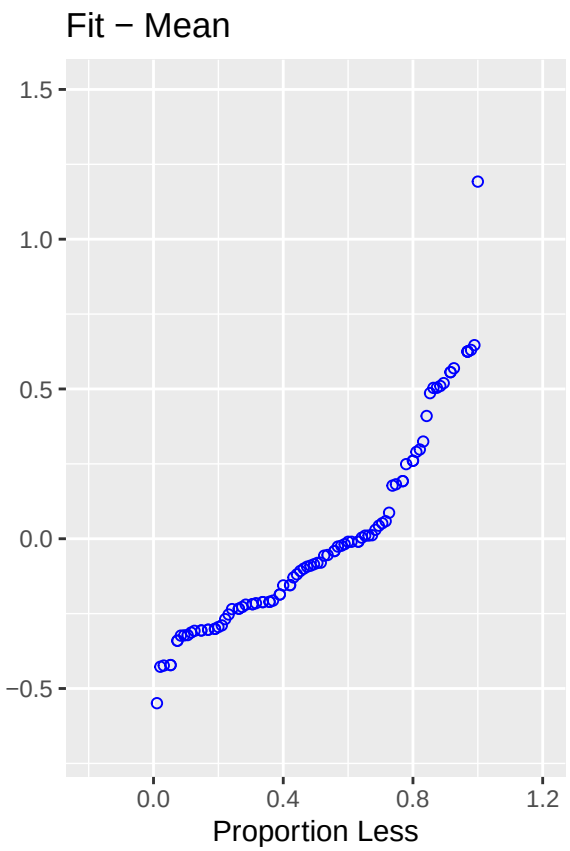
```
ols_plot_dfbetas(model2)
```

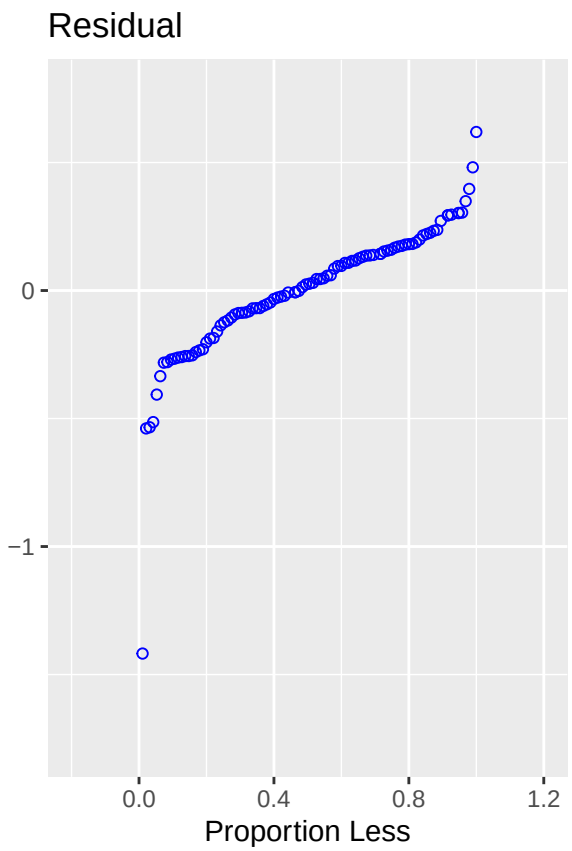



```
ols_plot_cooksd_chart(model2)
```

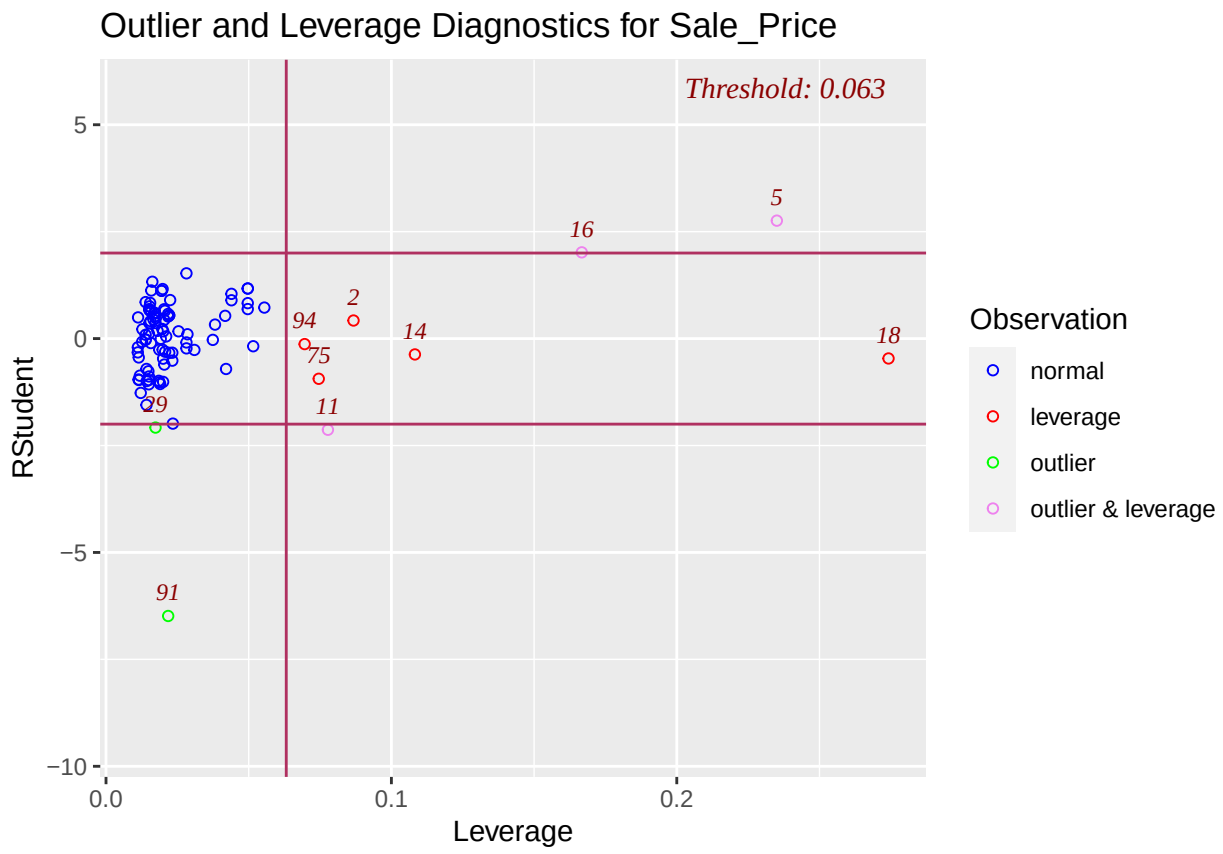



```
ols_plot_resid_fit_spread(model2)
```

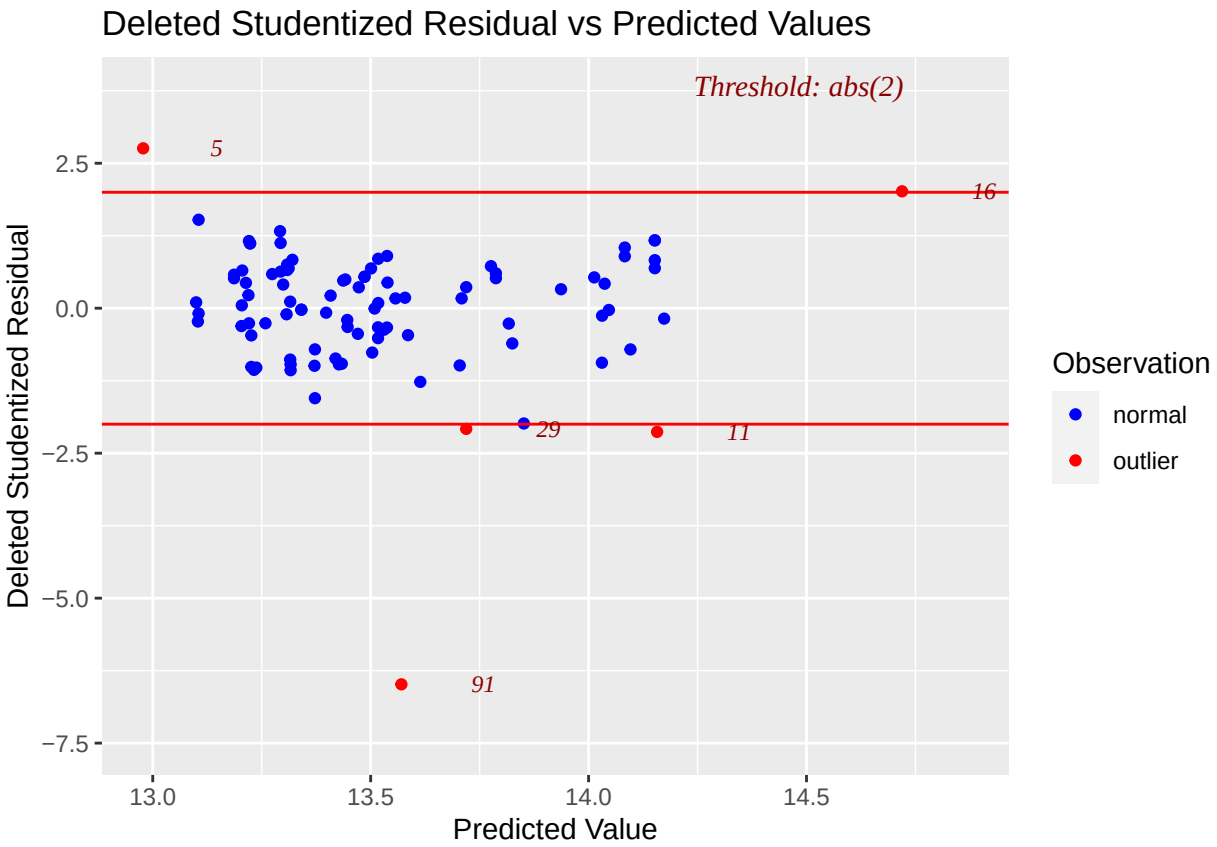





```
ols_plot_resid_lev(model2)
```




```
ols_plot_resid_stud_fit(model2)
```



##Explain any transformations or modifications you made to the dataset The column name was standardized by adding "-" in between. Also i transformed the selected subset to log() function in R Language returns the natural logarithm (base-e logarithm) of the argument passed in the parameter. Also i have filtered my data for houses built in a particular year "2000".

##Create two variables; one that will contain the variables Sale Price and Square Foot of Lot (same variables used from previous assignment on simple regression) and one that will contain Sale Price and several additional predictors of your choice. Explain the basis for your additional predictor selections. I created two models.

1. Linear regression

Relationship between two variable (Sales prices and Square foot of lot). Dependent variable is Sales price and independent variable is "Square foot of lot".

2. Multiple Linear Regression(Sales prices and Square feet total living + Square foot of lot)

Multiple regression is a statistical technique that can be used to analyze the relationship between a single dependent variable and several independent variables. So for independent variable or predictor i have chosen "Square feet total living" and "Square foot of lot".

##Execute a summary() function on two variables defined in the previous step to compare the model results. What are the R2 and Adjusted R2 statistics? Explain what these results tell you about the overall model. Did the inclusion of the additional predictors help explain any large variations found in Sale Price?

For multiple regression, R2 must be adjusted and the value found is 0.6046 vs for Linear regression the R2 is 0.3996. Adding one more predictor value has improved the R2 values.

Multiple linear regression is a more specific calculation than simple linear regression. For straight-forward relationships, simple linear regression may easily capture the relationship between the two variables. For more complex relationships requiring more consideration, multiple linear regression is often better.

A multiple regression formula has multiple slopes (one for each variable) and one y-intercept. It is interpreted the same as a simple linear regression formula except there are multiple variables that all impact the slope of the relationship.

The R-squared for model 2 turns out to be higher, which indicates that it's able to explain more of the variance in the response values compared to model 1. (Model 1 = 0.3995917, Model 2 = 0.6129985)

##Considering the parameters of the multiple regression model you have created. What are the standardized betas for each parameter and what do the values indicate?

```
                Estimate Std. Error t value Pr(>|t|)
```

(Intercept) 7.43406 0.60648 12.258 < 2e-16 ***sq__ft__lot 0.14907 0.02480 6.012 3.65e-08*** square_feet_total_living 0.58864 0.08264 7.123 2.30e-10 ***

p value is 0. The null hypothesis is rejected and your test is statistically significant

##Calculate the confidence intervals for the parameters in your model and explain what the results indicate. Assess the improvement of the new model compared to your original model (simple regression model) by testing whether this change is significant by performing an analysis of variance. Perform casewise diagnostics to identify outliers and/or influential cases, storing each function's output in a dataframe assigned to a unique variable name.

##Calculate the standardized residuals using the appropriate command, specifying those that are +-2, storing the results of large residuals in a variable you create.

We can see that the residual sum of squares for mode2 1 is lower, which indicates that it fits the data better than model 1.

Model 1: 10.0868 Model 2: 6.501586

##Use the appropriate function to show the sum of large residuals.Which specific variables have large residuals (only cases that evaluate as TRUE)?Investigate further by calculating the leverage, cooks distance, and covariance rations. Comment on all cases that are problematics.

deviance function was used to check the large residual between two models. ols_plot_resid_lev shows the leverage, outlier in the model. ols_plot_resid_stud_fit - Graph shows the detecting outliers.

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

Durbin watson autocorrelation test was performed to investigate where the residuals from the linear or multiple regression model are independent.

The Durban Watson statistic will always assume a value between 0 and 4. A value of DW = 2 indicates that there is no autocorrelation. When the value is below 2, it indicates a positive autocorrelation, and a value higher than 2 indicates a negative serial correlation

For linear regression, the D-W Statistic observed is 2.007405 and for multiple regression the D-W Statistic observed is 1.867343. The conditions are met.

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

Performed multicollinearity calculations only for the multiple regressions. If the independent variable is >-0.85 or >0.85, those variable should not included part of the model. R2 will be large but none of the individual beta weights are statistically significant.

To visualize the VIF values for each predictor variable, we can create a simple horizontal bar chart and add a vertical line at 5 so we can clearly see which VIF values exceed 5. In the multiple regression none of the values were above 5

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

Linear regression plot() and Hist() both show that there are couple of anomalies present in the chart. Anyhow, the charts shows they have roughly normally distributed.

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

No. The model is not unbiased.