# DSC520_8.2 Exercise

Ruth Maina

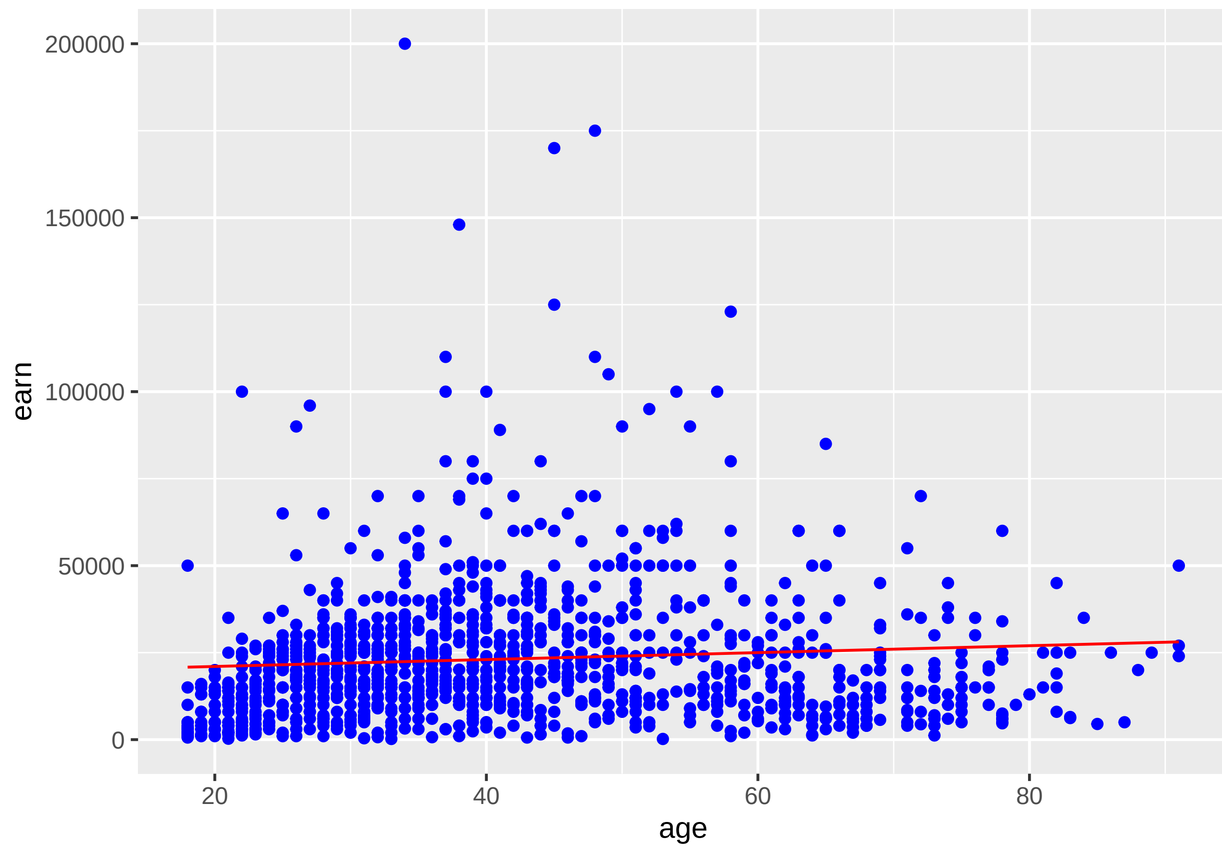
2023-02-11

```r
# Assignment: ASSIGNMENT 6
# Name: Maina, Ruth
# Date: 2023-02-11

## Set the working directory to the root of your DSC 520 directory

setwd("C:/Users/KiluWorksEnterprise/OneDrive/Desktop/Ruth Bellevue/DSC520/DSC520_RuthMaina/dsc520")

## Load the `data/r4ds/heights.csv` to
heights_df <- read.csv("data/r4ds/heights.csv")

## Load the ggplot2 library
library(ggplot2)

## Fit a linear model using the `age` variable as the predictor and `earn` as the outcome
age_lm <- lm(formula = earn ~ age, data=heights_df)

## View the summary of your model using `summary()`
summary(age_lm)
```

```
##
## Call:
## lm(formula = earn ~ age, data = heights_df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -25098 -12622  -3667   6883 177579
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19041.53    1571.26  12.119  < 2e-16 ***
## age            99.41      35.46   2.804  0.00514 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19420 on 1190 degrees of freedom
## Multiple R-squared:  0.006561,   Adjusted R-squared:  0.005727
## F-statistic:  7.86 on 1 and 1190 DF,  p-value: 0.005137
```

```r
## Creating predictions using `predict()`
age_predict_df <- data.frame(earn = predict(age_lm, heights_df), age=heights_df$age)
```

```r
## Plot the predictions against the original data
ggplot(data = heights_df, aes(y = earn, x = age)) +
  geom_point(color='blue') +
  geom_line(color='red',data = age_predict_df, aes(y=earn, x=age))
```




```r
mean_earn <- mean(heights_df$earn)
## Corrected Sum of Squares Total
sst <- sum((mean_earn - heights_df$earn)^2)
## Corrected Sum of Squares for Model
ssm <- sum((mean_earn - age_predict_df$earn)^2)
## Residuals
residuals <- heights_df$earn - age_predict_df$earn
## Sum of Squares for Error
sse <- sum(residuals^2)
## R Squared R^2 = SSM\SST
r_squared <- ssm / sst
r_squared
```

```
## [1] 0.006561482
```

```r
## Number of observations
n <- nrow(heights_df)
n
```

```
## [1] 1192
```

```r
## Number of regression parameters
p <- 2
## Corrected Degrees of Freedom for Model (p-1)
dfm <- p-1
dfm
```

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

```r
## Degrees of Freedom for Error (n-p)
dfe <- n-p
dfe
```

```
## [1] 1190
```

```r
## Corrected Degrees of Freedom Total:   DFT = n - 1
dft <- n-1
dft
```

```
## [1] 1191
```

```r
## Mean of Squares for Model:   MSM = SSM / DFM
msm <- ssm/dfm
msm
```

```
## [1] 2963111900
```

```r
## Mean of Squares for Error:   MSE = SSE / DFE
mse <- sse/dfe
mse
```

```
## [1] 376998968
```

```r
## Mean of Squares Total:   MST = SST / DFT
mst <- sst/dft
mst
```

```
## [1] 379170348
```

```r
## F Statistic F = MSM/MSE
f_score <- msm/mse
f_score
```

```
## [1] 7.859735
```

```r
## Adjusted R Squared R2 = 1 - (1 - R2)(n - 1) / (n - p)
adjusted_r_squared <-  1 - (1 - r_squared) * (n - 1) / (n - p)
adjusted_r_squared
```

```
## [1] 0.005726659
```

```r
## Calculate the p-value from the F distribution
p_value <- pf(f_score, dfm, dft, lower.tail=F)
p_value
```

```
## [1] 0.005136826
```

```r
# Assignment: ASSIGNMENT 7
# Name: Maina, Ruth
# Date: 2023-02-11


# Fit a linear model
earn_lm <-  lm(earn ~ age + sex + height + ed + race, data=heights_df)

# View the summary of your model
summary(earn_lm)
```

```
##
## Call:
## lm(formula = earn ~ age + sex + height + ed + race, data = heights_df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -39423  -9827  -2208   6157 158723
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -41478.4    12409.4  -3.342 0.000856 ***
## age             178.3       32.2   5.537 3.78e-08 ***
## sexmale       10325.6     1424.5   7.249 7.57e-13 ***
## height          202.5      185.6   1.091 0.275420
## ed             2768.4      209.9  13.190  < 2e-16 ***
## racehispanic  -1414.3     2685.2  -0.527 0.598507
## raceother       371.0     3837.0   0.097 0.922983
## racewhite      2432.5     1723.9   1.411 0.158489
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17250 on 1184 degrees of freedom
## Multiple R-squared:  0.2199, Adjusted R-squared:  0.2153
## F-statistic: 47.68 on 7 and 1184 DF,  p-value: < 2.2e-16
```

```r
predicted_df <- data.frame(
  earn = predict(earn_lm, heights_df),
  ed=heights_df$ed, race=heights_df$race, height=heights_df$height,
  age=heights_df$age, sex=heights_df$sex
  )

head(predicted_df)
```

```
##       earn ed  race   height age    sex
## 1 38666.11 16 white 74.42444  45   male
```

```
## 2 28859.09 16 white 65.53754  58 female
## 3 23301.90 16 white 63.62920  29 female
## 4 32189.84 16 other 63.10856  91 female
## 5 27807.39 17 white 63.40248  39 female
## 6 20154.60 15 white 64.39951  26 female
```

```r
## Compute deviation (i.e. residuals)
mean_earn <- mean(heights_df$earn)
## Corrected Sum of Squares Total
sst <- sum((mean_earn - heights_df$earn)^2)
## Corrected Sum of Squares for Model
ssm <- sum((mean_earn - predicted_df$earn)^2)
## Residuals
residuals <- heights_df$earn - predicted_df$earn
## Sum of Squares for Error
sse <- sum(residuals^2)
## R Squared
r_squared <- ssm / sst

## Number of observations
n <- nobs(earn_lm)
n
```

```
## [1] 1192
```

```r
## Number of regression paramaters
p <- 8
p
```

```
## [1] 8
```

```r
## Corrected Degrees of Freedom for Model
dfm <- p-1
dfm
```

```
## [1] 7
```

```r
## Degrees of Freedom for Error
dfe <- n-p
dfe
```

```
## [1] 1184
```

```r
## Corrected Degrees of Freedom Total:   DFT = n - 1
dft <- n-1
dft
```

```
## [1] 1191
```

```r
## Mean of Squares for Model:   MSM = SSM / DFM
msm <- ssm / dfm
msm
```

```
## [1] 14186131237
```

```r
## Mean of Squares for Error:   MSE = SSE / DFE
mse <- sse/dfe
mse
```

```
## [1] 297541356
```

```r
## Mean of Squares Total:   MST = SST / DFT
mst <- sst/dft
mst
```

```
## [1] 379170348
```

```r
## F Statistic
f_score <- msm/mse
f_score
```

```
## [1] 47.67785
```

```r
## Adjusted R Squared R2 = 1 - (1 - R2)(n - 1) / (n - p)
adjusted_r_squared <- 1 - (1 - r_squared)*(n - 1) / (n - p)
adjusted_r_squared
```

```
## [1] 0.2152832
```

## 3. HOUSING DATA

**a. Work individually on this assignment. You are encouraged to collaborate on ideas and strategies pertinent to this assignment. Data for this assignment is focused on real estate transactions recorded from 1964 to 2016 and can be found in Housing.xlsx. Using your skills in statistical correlation, multiple regression, and R programming, you are interested in the following variables: Sale Price and several other possible predictors.**

**i. If you worked with the Housing dataset in previous week – you are in luck, you likely have already found any issues in the dataset and made the necessary transformations. If not, you will want to take some time looking at the data with all your new skills and identifying if you have any clean up that needs to happen.**

**b. i. Explain any transformations or modifications you made to the dataset**

My rationale for below: I used mutate to combine bath__full__count and bath__half__count and bath__3qtr__count. This could be good to know for the purpose of correlation to Sale Price. i learned that three-quarter bathrooms have a toilet, sink, and either a separate shower or a separate bathtub.

```
setwd("C:/Users/KiluWorksEnterprise/OneDrive/Desktop/Ruth Bellevue/DSC520/DSC520_RuthMaina/dsc520")

library(readxl); library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## v purrr   0.3.5
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
housing_df <- read_excel('data/week-6-housing.xlsx', sheet = 'Sheet2')
head(housing_df)
```

```
## # A tibble: 6 x 24
##   `Sale Date`         `Sale Price` sale_~1 sale_~2 sale_~3 sitet~4 addr_~5  zip5
##   <dttm>                     <dbl>   <dbl>   <dbl> <chr>   <chr>   <chr>   <dbl>
```

```
## 1 2006-01-03 00:00:00       698000       1       3 <NA>    R1      17021 ~ 98052
## 2 2006-01-03 00:00:00       649990       1       3 <NA>    R1      11927 ~ 98052
## 3 2006-01-03 00:00:00       572500       1       3 <NA>    R1      13315 ~ 98052
## 4 2006-01-03 00:00:00       420000       1       3 <NA>    R1      3303 1~ 98052
## 5 2006-01-03 00:00:00       369900       1       3 15      R1      16126 ~ 98052
## 6 2006-01-03 00:00:00       184667       1      15 18 51   R1      8101 2~ 98053
## # ... 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
```

```
summary(housing_df)
```

```
##    Sale Date                        Sale Price       sale_reason
##  Min.   :2006-01-03 00:00:00.00   Min.   :    698   Min.   : 0.00
##  1st Qu.:2008-07-07 00:00:00.00   1st Qu.: 460000   1st Qu.: 1.00
##  Median :2011-11-17 00:00:00.00   Median : 593000   Median : 1.00
##  Mean   :2011-07-28 15:07:32.48   Mean   : 660738   Mean   : 1.55
##  3rd Qu.:2014-06-05 00:00:00.00   3rd Qu.: 750000   3rd Qu.: 1.00
##  Max.   :2016-12-16 00:00:00.00   Max.   :4400000   Max.   :19.00
##  sale_instrument  sale_warning         sitetype          addr_full
##  Min.   : 0.000   Length:12865       Length:12865       Length:12865
##  1st Qu.: 3.000   Class :character   Class :character   Class :character
##  Median : 3.000   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 3.678
##  3rd Qu.: 3.000
##  Max.   :27.000
##       zip5         ctyname           postalctyn             lon
##  Min.   :98052   Length:12865       Length:12865       Min.   :-122.2
##  1st Qu.:98052   Class :character   Class :character   1st Qu.:-122.1
##  Median :98052   Mode  :character   Mode  :character   Median :-122.1
##  Mean   :98053                                          Mean   :-122.1
##  3rd Qu.:98053                                          3rd Qu.:-122.0
##  Max.   :98074                                          Max.   :-121.9
##       lat        building_grade  square_feet_total_living    bedrooms
##  Min.   :47.46   Min.   : 2.00   Min.   :  240            Min.   : 0.000
##  1st Qu.:47.67   1st Qu.: 8.00   1st Qu.: 1820            1st Qu.: 3.000
##  Median :47.69   Median : 8.00   Median : 2420            Median : 4.000
##  Mean   :47.68   Mean   : 8.24   Mean   : 2540            Mean   : 3.479
##  3rd Qu.:47.70   3rd Qu.: 9.00   3rd Qu.: 3110            3rd Qu.: 4.000
##  Max.   :47.73   Max.   :13.00   Max.   :13540            Max.   :11.000
##  bath_full_count  bath_half_count  bath_3qtr_count   year_built
##  Min.   : 0.000   Min.   :0.0000   Min.   :0.000   Min.   :1900
##  1st Qu.: 1.000   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:1979
##  Median : 2.000   Median :1.0000   Median :0.000   Median :1998
##  Mean   : 1.798   Mean   :0.6134   Mean   :0.494   Mean   :1993
##  3rd Qu.: 2.000   3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.:2007
##  Max.   :23.000   Max.   :8.0000   Max.   :8.000   Max.   :2016
##  year_renovated    current_zoning       sq_ft_lot        prop_type
##  Min.   :   0.00   Length:12865       Min.   :    785   Length:12865
##  1st Qu.:   0.00   Class :character   1st Qu.:   5355   Class :character
```

```
##  Median :   0.00   Mode  :character   Median :   7965   Mode  :character
##  Mean   :  26.24                      Mean   :  22229
##  3rd Qu.:   0.00                      3rd Qu.:  12632
##  Max.   :2016.00                      Max.   :1631322
##   present_use
##  Min.   :  0.000
##  1st Qu.:  2.000
##  Median :  2.000
##  Mean   :  6.598
##  3rd Qu.:  2.000
##  Max.   :300.000
```

```r
## Mutate
housing_df_mutate <- housing_df %>%
    select(`Sale Date`, `Sale Price`, sq_ft_lot, bedrooms, year_built, year_renovated) %>%
    mutate (Total_Bath = housing_df$bath_full_count + housing_df$bath_half_count + housing_df$bath_3qtr_
head(housing_df_mutate)
```

```
## # A tibble: 6 x 7
##   `Sale Date`         `Sale Price` sq_ft_lot bedrooms year_built year_~1 Total~2
##   <dttm>                     <dbl>     <dbl>    <dbl>      <dbl>   <dbl>   <dbl>
## 1 2006-01-03 00:00:00       698000      6635        4       2003       0       3
## 2 2006-01-03 00:00:00       649990      5570        4       2006       0       3
## 3 2006-01-03 00:00:00       572500      8444        4       1987       0       3
## 4 2006-01-03 00:00:00       420000      9600        3       1968       0       2
## 5 2006-01-03 00:00:00       369900      7526        3       1980       0       2
## 6 2006-01-03 00:00:00       184667      7280        4       2005       0       4
## # ... with abbreviated variable names 1: year_renovated, 2: Total_Bath
```

### ii. 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.

**My rationale for below: I am using the mutated data from item i above. Also included a view of several other data points of interest, below are my general assumptions:**

**housing_df_mutate - contains 1 new calculated columns for total baths since these do influence house prices in general.**

**bedrooms - the more the bedrooms the higher the price**

**year_built - new built houses seem to have high prices**

**year_renovated - renovations increase prices**

**Sale Date - more recent sales are pricey compared to older**

### iii. 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?

**My rationale for summary function executed above: R-squared value goes from 0.01435 to 0.127 to 0.13 for lot size, count of bathrooms and bedrooms respectively - This improvement in a positive direction is an indication of correlation, though bathrooms and bedrooms was just .01 increase so not a huge increase between these two.**

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

```
library(lm.beta)
lm.beta(sale_lm_tFootage)
```

```
##
## Call:
## lm(formula = housing_df$`Sale Price` ~ housing_df$sq_ft_lot)
```

```
##
## Standardized Coefficients::
##         (Intercept) housing_df$sq_ft_lot
##                  NA            0.1198122
```

```
lm.beta(sale_lm_ftBath)
```

```
##
## Call:
## lm(formula = housing_df$`Sale Price` ~ housing_df$sq_ft_lot +
##     housing_df_mutate$Total_Bath)
##
## Standardized Coefficients::
##                  (Intercept)         housing_df$sq_ft_lot
##                           NA                   0.08941303
## housing_df_mutate$Total_Bath
##                   0.33704755
```

```
lm.beta(sale_lm_ftBathBed)
```

```
##
## Call:
## lm(formula = housing_df$`Sale Price` ~ housing_df$sq_ft_lot +
##     housing_df_mutate$Total_Bath + housing_df$bedrooms)
##
## Standardized Coefficients::
##                  (Intercept)         housing_df$sq_ft_lot
##                           NA                   0.08938620
## housing_df_mutate$Total_Bath          housing_df$bedrooms
##                   0.30406744                   0.06362467
```

**My rationale for betas values calculated above: 0.01, 0.3 and 0.06 are all very low but positive thus indicating the variables would increase as the predictor (Sale) increases. This is aexpected**

## v. Calculate the confidence intervals for the parameters in your model and explain what the results indicate.

```
confint(sale_lm_tFootage)
```

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

```
confint(sale_lm_ftBath)
```

```
##                                     2.5 %       97.5 %
## (Intercept)                  1.615995e+05 2.077640e+05
## housing_df$sq_ft_lot         5.199074e-01 7.502436e-01
## housing_df_mutate$Total_Bath 1.513241e+05 1.666197e+05
```

```r
confint(sale_lm_ftBathBed)
```

```
##                                   2.5 %       97.5 %
## (Intercept)                 9.916918e+04 1.562934e+05
## housing_df$sq_ft_lot         5.199078e-01 7.498621e-01
## housing_df_mutate$Total_Bath 1.344980e+05 1.523351e+05
## housing_df$bedrooms          2.066472e+04 3.806788e+04
```

**My rationale for confidence interval value of 97.5% indicates a high probability of samples containing true values of the population**

### vi. 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.

```r
anova(sale_lm_tFootage)
```

```
## Analysis of Variance Table
##
## Response: housing_df$`Sale Price`
##                        Df     Sum Sq    Mean Sq F value    Pr(>F)
## housing_df$sq_ft_lot     1 3.0197e+13 3.0197e+13  187.34 < 2.2e-16 ***
## Residuals            12863 2.0734e+15 1.6119e+11
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(sale_lm_ftBath)
```

```
## Analysis of Variance Table
##
## Response: housing_df$`Sale Price`
##                                 Df     Sum Sq    Mean Sq F value    Pr(>F)
## housing_df$sq_ft_lot             1 3.0197e+13 3.0197e+13   211.5 < 2.2e-16 ***
## housing_df_mutate$Total_Bath     1 2.3702e+14 2.3702e+14  1660.1 < 2.2e-16 ***
## Residuals                    12862 1.8364e+15 1.4277e+11
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(sale_lm_ftBathBed)
```

```
## Analysis of Variance Table
##
## Response: housing_df$`Sale Price`
##                                 Df     Sum Sq    Mean Sq F value    Pr(>F)
## housing_df$sq_ft_lot             1 3.0197e+13 3.0197e+13  212.20 < 2.2e-16 ***
## housing_df_mutate$Total_Bath     1 2.3702e+14 2.3702e+14 1665.66 < 2.2e-16 ***
## housing_df$bedrooms              1 6.2271e+12 6.2271e+12   43.76 3.859e-11 ***
## Residuals                    12861 1.8301e+15 1.4230e+11
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

**My rationale for anova results: The observable change here is the F value which significantly increased from the first model to the second model (~212 to 1666), then it dropped on the third model (~44). This shows the addition of the second variable (# of bathrooms) was significant which indicate a statistical significance, that an increase in the number of bathrooms would mean an increase in house sale price**

### vii. Perform casewise diagnostics to identify outliers and/or influential cases, storing each function's output in a dataframe assigned to a unique variable name.

```
#calculation of outliers and influential cases
residuals <- c(resid(sale_lm_ftBathBed))
standardized.residuals <- c(rstandard(sale_lm_ftBathBed))
studentized.residuals <- c(rstudent(sale_lm_ftBathBed))
cooks.distance <- c(cooks.distance(sale_lm_ftBathBed))
dfbeta <- c(dfbeta(sale_lm_ftBathBed))
dffit <- c(dffits(sale_lm_ftBathBed))
leverage <- c(hatvalues(sale_lm_ftBathBed))
covariance.ratios <- c(covratio(sale_lm_ftBathBed))
#store above calculations in a data frame
sale_lm <- data.frame (residuals, standardized.residuals, studentized.residuals,
cooks.distance, dfbeta, dffit, leverage, covariance.ratios)
```

```
## Warning in data.frame(residuals, standardized.residuals,
## studentized.residuals, : row names were found from a short variable and have
## been discarded
```

```
head(sale_lm)
```

```
##    residuals standardized.residuals studentized.residuals cooks.distance
## 1   18341.49             0.04862468            0.04862280   6.844055e-08
## 2  -28992.35            -0.07686095           -0.07685798   1.722771e-07
## 3 -108307.01            -0.28712987           -0.28711962   2.358933e-06
## 4  -88758.12            -0.23531034           -0.23530170   2.294850e-06
## 5 -137541.36            -0.36464179           -0.36462950   5.538162e-06
## 6 -638817.53            -1.69363834           -1.69376138   1.549259e-04
##       dfbeta         dffit     leverage covariance.ratios
## 1  -1.393682  0.0005232026 0.0001157736         1.0004262
## 2   2.199443 -0.0008300936 0.0001166341         1.0004259
## 3   8.252236 -0.0030716544 0.0001144377         1.0003999
## 4 -31.655911 -0.0030296413 0.0001657526         1.0004597
## 5 -49.087465 -0.0047065015 0.0001665791         1.0004364
## 6 153.598875 -0.0248956586 0.0002159976         0.9996349
```

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

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

```
sum(sale_lm$large.residual)
```

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

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

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

**Please find more calculations in item vii above. Cooks distance values were all under 1**

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

```
library(Rmisc)
```

```
## Loading required package: lattice

## Loading required package: plyr

## ------------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##
## Attaching package: 'plyr'
```

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

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

```
library(Hmisc)
```

```
## Loading required package: survival

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:plyr':
##
##     is.discrete, summarize

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

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

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

```
durbinWatsonTest(sale_lm_tFootage)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.6309692     0.7380424       0
##  Alternative hypothesis: rho != 0
```

```r
durbinWatsonTest(sale_lm_ftBathBed)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.6757566     0.6484822       0
##  Alternative hypothesis: rho != 0
```

**The durbin test above raises an alarm because it is less than 1 (should ideally be close to 2). However, the p-value of zero is consistent with earlier values which were less than 0, indicating statistical significance**

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

```r
mean(vif(sale_lm_ftBathBed))
```

```
## [1] 1.25044
```

```r
vif(sale_lm_ftBathBed)
```

```
##         housing_df$sq_ft_lot housing_df_mutate$Total_Bath
##                     1.008202                     1.375632
##          housing_df$bedrooms
##                     1.367485
```

```r
1/vif(sale_lm_ftBathBed)
```

```
##         housing_df$sq_ft_lot housing_df_mutate$Total_Bath
##                    0.9918651                    0.7269385
##          housing_df$bedrooms
##                    0.7312695
```

For above results, VIF values are below 10 and average VIFs slightly greater than 1 which is an indication of lack of bias which is good. Tolerance values (1/vif) are not below 0.2 which is good

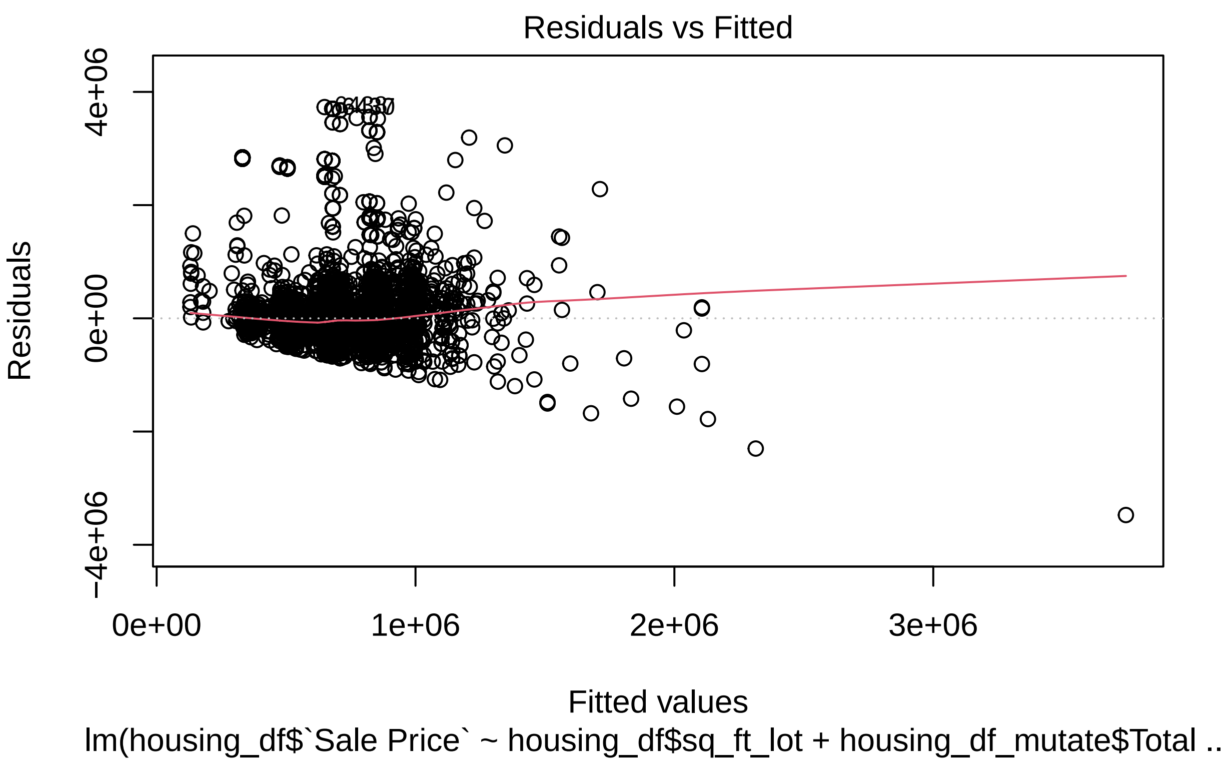
## xiv. Visually check the assumptions related to the residuals using the plot() and hist() functions. Summarize what each graph is informing you of and if any anomalies are present.

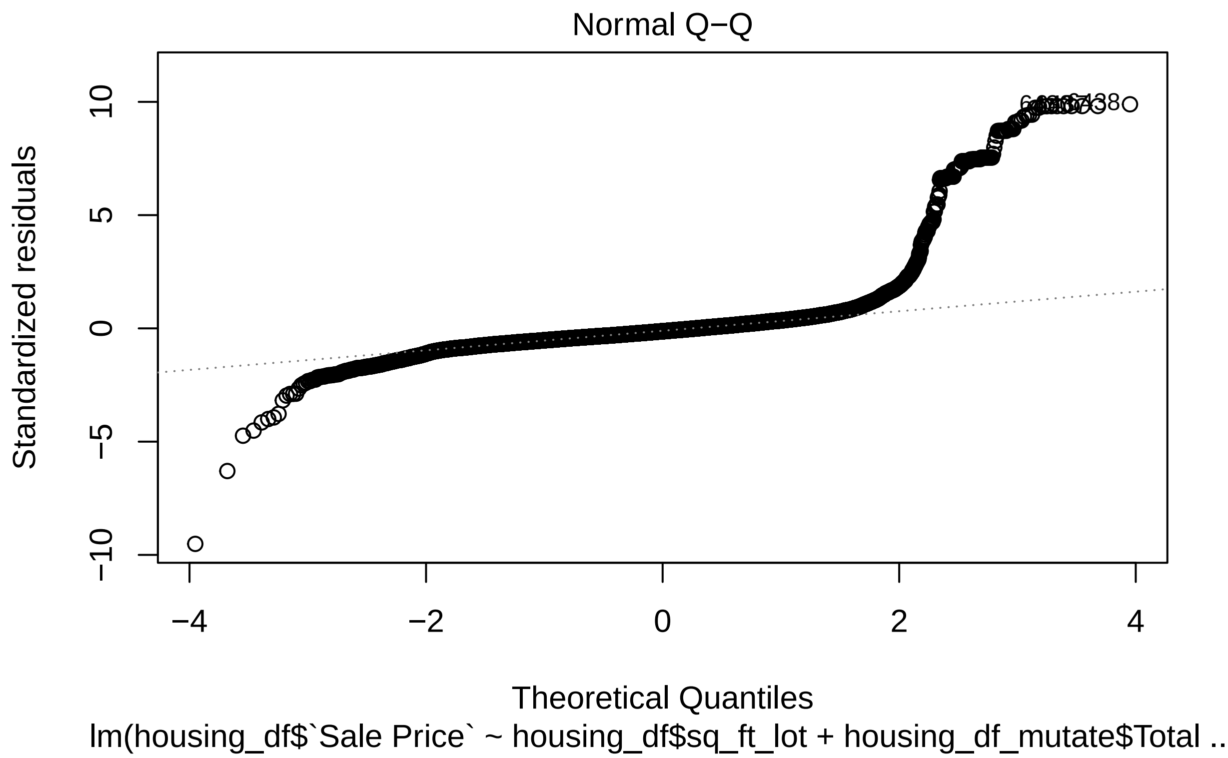
Residual vs Fitted Graph: the data points are distributed randomly and not so distant from the line

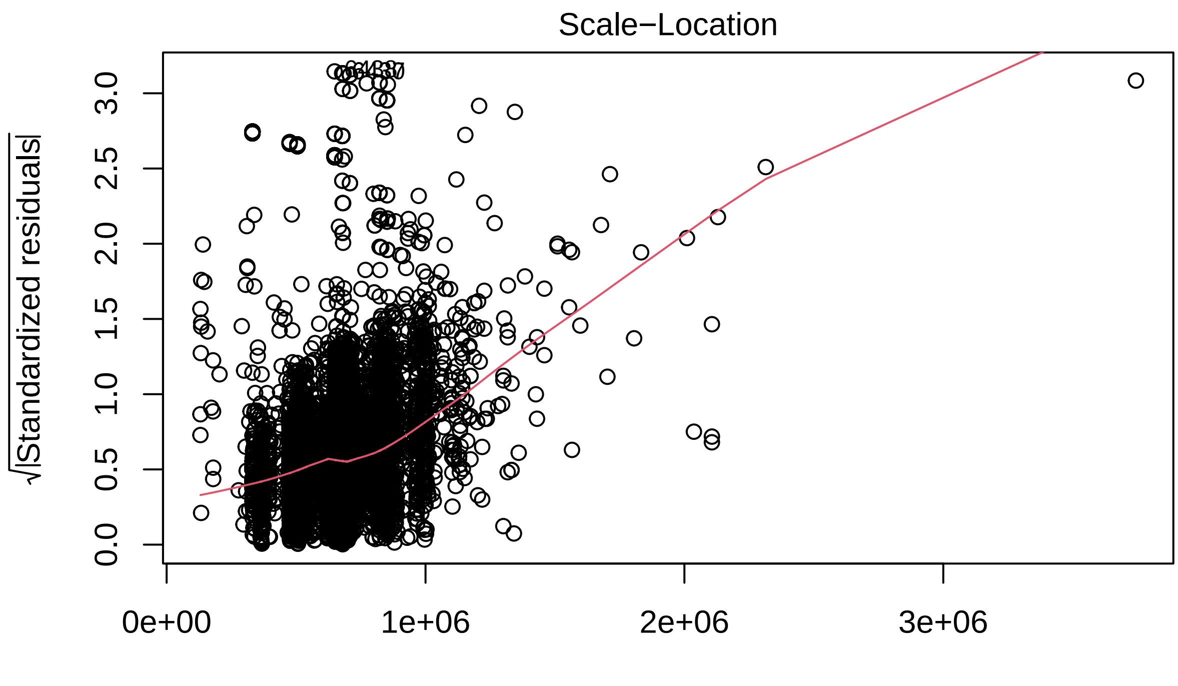
The QQ plot shows residuals from the line

The histogram is not showing a normal distribution

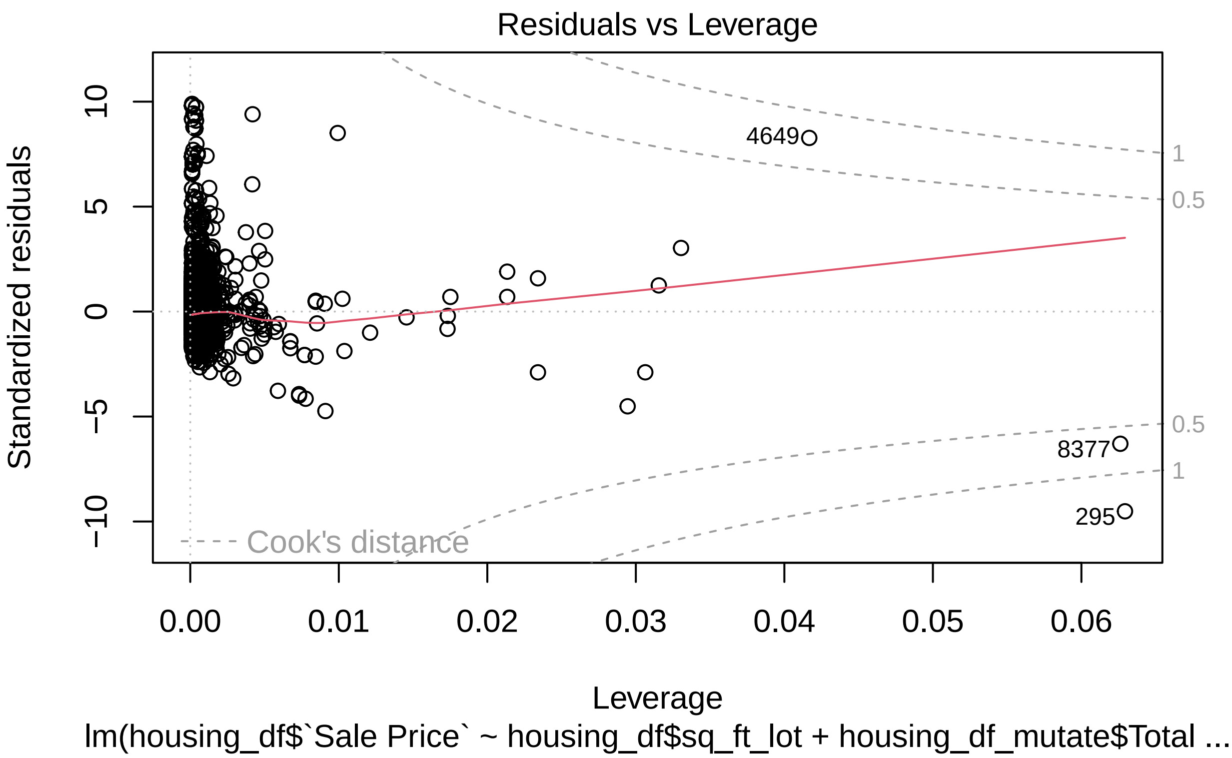
```
library(ggplot2)
plot(sale_lm_ftBathBed)
```



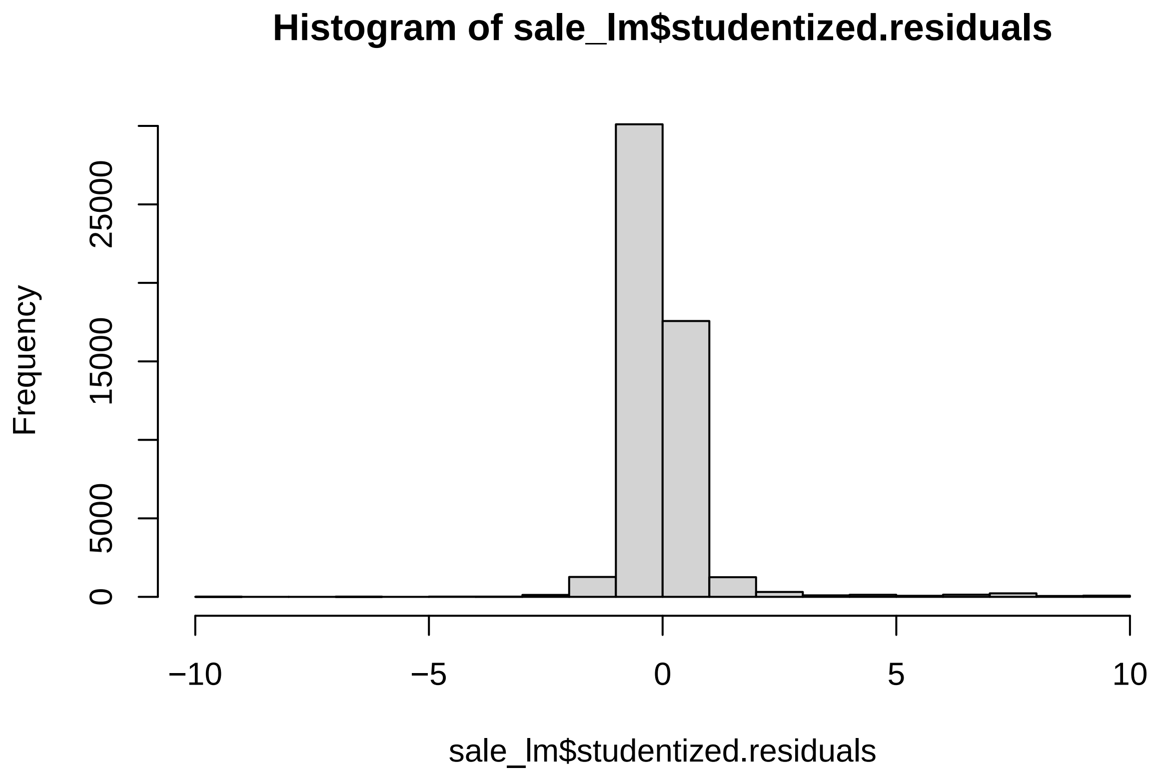

Normal Q−Q
Standardized residuals
10
5
0
−5
−10
−4
−2
0
2
4
Theoretical Quantiles
lm(housing_df$`Sale Price` ~ housing_df$sq_ft_lot + housing_df_mutate$Total ...



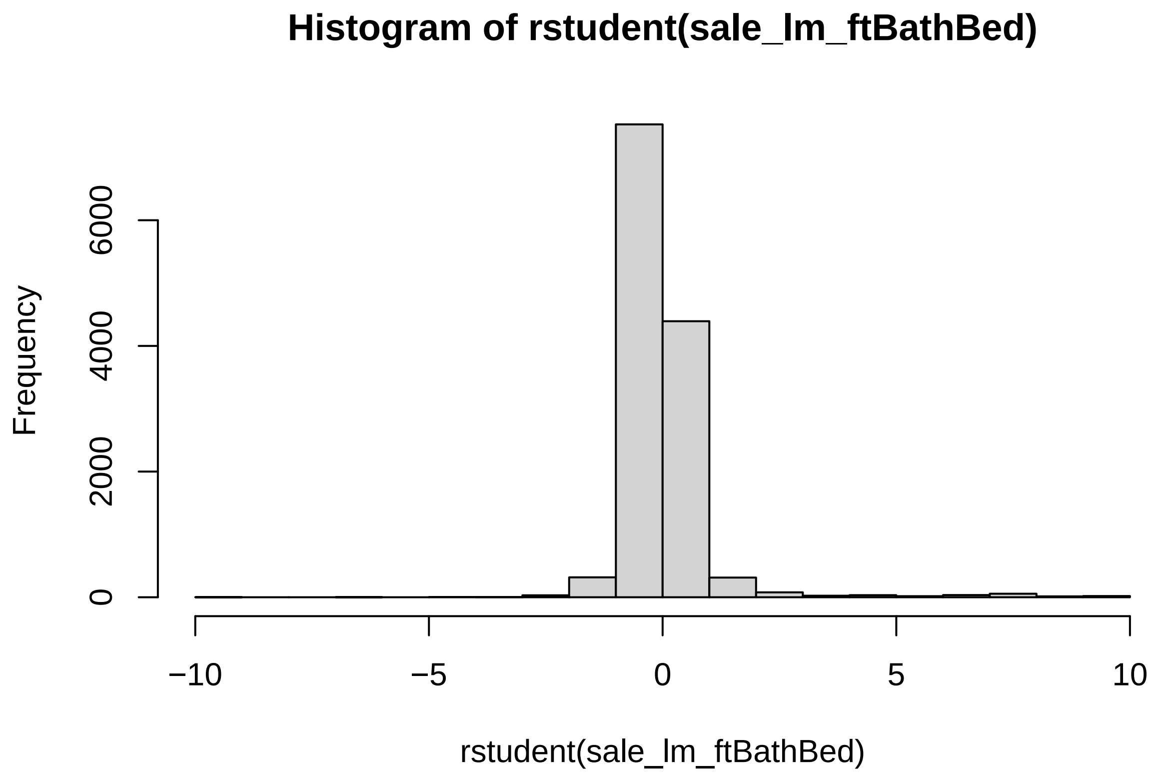
lm(housing_df$`Sale Price` ~ housing_df$sq_ft_lot + housing_df_mutate$Total ...




```
hist(sale_lm$studentized.residuals)
```




```
hist(rstudent(sale_lm_ftBathBed))
```




# xv. Overall, is this regression model unbiased? If an unbiased regression model, what does this tell us about the sample vs. the entire population model? ## This model is unbiased, especially based on multicollinearity results slightly greater than 1 which is an indication of lack of bias

### Citations

- R for Everyone (Lander 2014)
- Discovering Statistics Using R(Field, Miles, and Field 2012)

### References

Field, A., J. Miles, and Z. Field. 2012. *Discovering Statistics Using r*. SAGE Publications. https://books.google.com/books?id=wd2K2zC3swIC.

Lander, J. P. 2014. *R for Everyone: Advanced Analytics and Graphics*. Addison-Wesley Data and Analytics Series. Addison-Wesley. https://books.google.com/books?id=3eBVAgAAQBAJ.