8.2 Exercise

Alexa Wittlieff

February 6th 2022

1. ASSIGNMENT 6

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

Load the `data/r4ds/heights.csv` to

Load the ggplot2 library

Fit a linear model using the `age` variable as the predictor and `earn` as the outcome

```
age_lm <- lm(age ~ earn, data=heights_df)
age_lm
```

```
##
## Call:
## lm(formula = age ~ earn, data = heights_df)
##
## Coefficients:
## (Intercept)         earn
##   3.985e+01    6.601e-05
```

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

```
summary(age_lm)
```

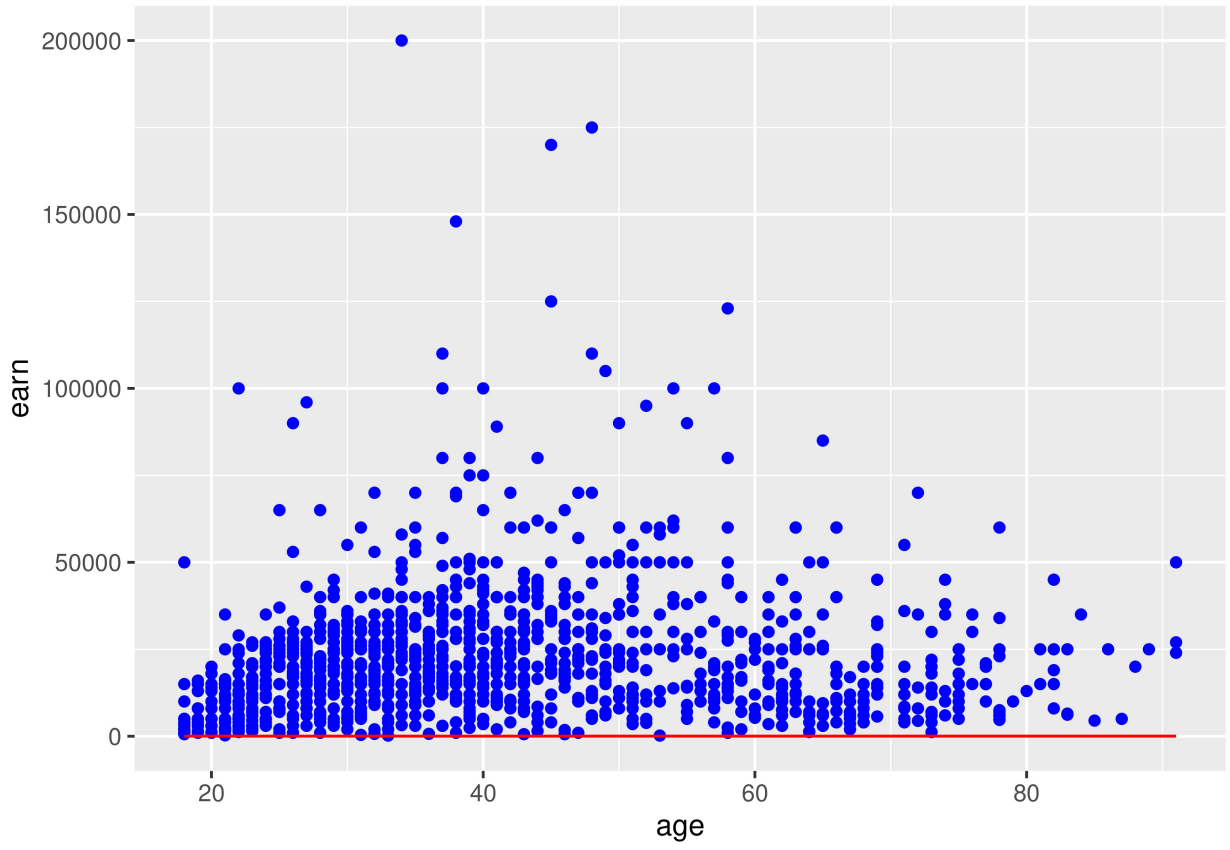
```
##
## Call:
## lm(formula = age ~ earn, data = heights_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -25.150 -12.160  -3.840   8.712  49.566
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.985e+01  7.122e-01  55.954  < 2e-16 ***
```

```
## earn        6.601e-05  2.354e-05   2.804  0.00514 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.82 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
```

Creating predictions using `predict()`

```
age_predict_df <- data.frame(earn = predict(age_lm, heights_df), age=heights_df$age)
```

Plot the predictions against the original data




```
mean_earn <- mean(heights_df$earn)
mean_earn
```

```
## [1] 23154.77
```

```
## Corrected Sum of Squares Total
sst <- sum((mean_earn - heights_df$earn)^2)
sst
```

```
## [1] 451591883937
```

```
## Corrected Sum of Squares for Model
ssm <- sum((mean_earn - age_predict_df$earn)^2)
ssm
```

```
## [1] 6.36801e+11
```

```
## Residuals
residuals <- heights_df$earn - age_predict_df$earn
## Sum of Squares for Error
sse <- sum(residuals^2)
sse
```

```
## [1] 1.088333e+12
```

```
## R Squared R^2 = SSM\SST
r_squared <- ssm/sst
r_squared
```

```
## [1] 1.410125
```

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

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

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

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

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

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

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

```
## [1] 6.36801e+11
```

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

```
## [1] 914565780
```

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

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

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

```
## [1] 696.2878
```

```
## 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] 1.41047
```

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

```
## [1] 3.339188e-121
```

2. ASSIGNMENT 7

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

Load the `data/r4ds/heights.csv` to

```
heights_df <- read.csv("data/r4ds/heights.csv")
```

Fit a linear model

```
earn_lm <- lm(earn ~ ed + race + height + age + sex, data=heights_df)
earn_lm
```

```
##
## Call:
## lm(formula = earn ~ ed + race + height + age + sex, data = heights_df)
##
## Coefficients:
##  (Intercept)            ed  racehispanic     raceother     racewhite
##     -41478.5        2768.4       -1414.3         371.0        2432.5
##       height           age       sexmale
##        202.5         178.3       10325.6
```

View the summary of your model

```
summary(earn_lm)
```

```
##
## Call:
## lm(formula = earn ~ ed + race + height + age + sex, 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 ***
## 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
## height          202.5      185.6   1.091 0.275420
## age             178.3       32.2   5.537 3.78e-08 ***
## sexmale       10325.6     1424.5   7.249 7.57e-13 ***
## ---
## 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
```

```
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
  )
```

```
## Compute deviation (i.e. residuals)
mean_earn <- mean(heights_df$earn)
mean_earn
```

```
## [1] 23154.77
```

```r
## Corrected Sum of Squares Total
sst <- sum((mean_earn - heights_df$earn)^2)
sst
```

```
## [1] 451591883937
```

```r
## Corrected Sum of Squares for Model
ssm <- sum((mean_earn - predicted_df$earn)^2)
ssm
```

```
## [1] 99302918657
```

```r
## Residuals
residuals <- heights_df$earn - predicted_df$earn
## Sum of Squares for Error
sse <- sum(residuals^2)
sse
```

```
## [1] 3.52289e+11
```

```r
## R Squared
r_squared <- ssm/sst
r_squared
```

```
## [1] 0.2198953
```

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

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

```r
## Number of regression paramaters
p <- 8
## Corrected Degrees of Freedom for Model
dfm <- p - 1
## 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
```

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

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

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

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

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

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

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

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

```
## 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

Necessary Libraries

a. Setup the Data

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

Load the `data/week-6-housing.xlsx` file

i. Data Transformations from Previous Week

Mutate for Sale Price/Sq. Ft. Total Living (Calculation for price per sq.ft.)

```
## # A tibble: 12,865 x 4
##    Sale_Date           Sale_Price square_feet_total_living   psf
##    <dttm>                   <dbl>                     <dbl> <dbl>
##  1 2006-01-03 00:00:00     698000                      2810 248.
```

```
##  2 2006-01-03 00:00:00     649990                    2880 226.
##  3 2006-01-03 00:00:00     572500                    2770 207.
##  4 2006-01-03 00:00:00     420000                    1620 259.
##  5 2006-01-03 00:00:00     369900                    1440 257.
##  6 2006-01-03 00:00:00     184667                    4160  44.4
##  7 2006-01-04 00:00:00    1050000                    3960 265.
##  8 2006-01-04 00:00:00     875000                    3720 235.
##  9 2006-01-04 00:00:00     660000                    4160 159.
## 10 2006-01-04 00:00:00     650000                    2760 236.
## # ... with 12,855 more rows
```

Add psf column

```
housing_df$psf <- housing_df$Sale_Price/housing_df$square_feet_total_living
```

Remove zero as value for year renovated, replace with "N/A" value

```
housing_df$year_renovated[housing_df$year_renovated == 0] <- NA
```

b. Data Actions

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

- Formatted the headings/column names to remove spaces and add "_"
- Remove zeros from year renovated and replace with "N/A"
- Added a column to feature the calculated price per sq. ft. (psf)

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.

#Create Two Variables

```
housing_df$psf_lot <- housing_df$Sale_Price/housing_df$sq_ft_lot

saleprice_lm <- lm(Sale_Price ~ sq_ft_lot, data=housing_df)
saleprice_lm
```

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

```
saleprice2_lm <- lm(Sale_Price ~ building_grade + square_feet_total_living + bedrooms + year_built, data
saleprice2_lm
```

```
##
## Call:
## lm(formula = Sale_Price ~ building_grade + square_feet_total_living +
##     bedrooms + year_built, data = housing_df)
##
## Coefficients:
##              (Intercept)            building_grade  square_feet_total_living
##                -4547195.3                   31404.0                      152.1
##                 bedrooms                year_built
##                  -8751.1                    2304.8
```

Additional Predictors

- Building Grade
- Square Feet Total Living
- Bedrooms
- Year Built

Reason for Selections

Based on my experience buying and selling houses, I've seen these variables cause a sales price to be either higher or lower in the housing market. These predictors are also sometimes compared during the property value assessment process.

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?

```
summary(saleprice_lm)
```

```
##
## Call:
## lm(formula = Sale_Price ~ sq_ft_lot, data = housing_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2016064  -194842   -63293    91565  3735109
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.418e+05  3.800e+03  168.90   <2e-16 ***
## sq_ft_lot   8.510e-01  6.217e-02   13.69   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 401500 on 12863 degrees of freedom
## Multiple R-squared:  0.01435,    Adjusted R-squared:  0.01428
## F-statistic: 187.3 on 1 and 12863 DF,  p-value: < 2.2e-16
```

```
summary(saleprice2_lm)
```

```
##
## Call:
## lm(formula = Sale_Price ~ building_grade + square_feet_total_living +
##     bedrooms + year_built, data = housing_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1702737  -119049   -43715    40658  3958807
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -4.547e+06  3.933e+05 -11.562  < 2e-16 ***
## building_grade            3.140e+04  4.447e+03   7.061 1.74e-12 ***
## square_feet_total_living  1.521e+02  5.603e+00  27.138  < 2e-16 ***
## bedrooms                 -8.751e+03  4.522e+03  -1.935    0.053 .
## year_built                2.305e+03  1.997e+02  11.542  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 356800 on 12860 degrees of freedom
## Multiple R-squared:  0.2219, Adjusted R-squared:  0.2217
## F-statistic:   917 on 4 and 12860 DF,  p-value: < 2.2e-16
```

R2 and Adjusted R2 Statistics

For saleprice_lm containing sale price and square foot of the lot, the R2 value is 0.01435, and the adjusted R2 value is 0.01428.

For saleprice2_lm containing sale price and additional predictors, the R2 value is 0.2219, and the adjusted R2 value is 0.2217.

The value of R2 tells us that square foot of the lot can account for 1.4% of the variation in sale price. In the second instance, the R2 value tells us that the additional predictors can account for 22% of the variation in sale price. This is substantially higher than the original model. This would suggest that the additional predictors increased the predictability of the second model.

Both models also does not lose much predictive power due to shrinkage, as demonstrated by the minimal differences between the R2 values and the adjusted R2 values. Because the adjusted R2 values are very similar to the observed values of R2, the cross-validity of these models are each very good.

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?

```
lm.beta(saleprice2_lm)
```

```
##           building_grade square_feet_total_living                 bedrooms
##               0.08485238               0.37219094              -0.01896013
##               year_built
##               0.09814711
```

What the standardized beta values indicate

- building_grade
  - As building_grade increases by 1 standard deviation, sale price increases by 0.0849 standard deviations
- square_feet_total_living
  - As square_feet_total_living increases by 1 standard deviation, sale prices increases by 0.3722 standard deviations
- bedrooms
  - As bedrooms increases by 1 standard deviation, sale prices decreases by 0.019 standard deviations
- year_built
  - As year_built increases by 1 standard deviation, sale prices increases by 0.09815 standard deviations

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

```
confint(saleprice2_lm)
```

```
##                                 2.5 %        97.5 %
## (Intercept)              -5318107.6997 -3776282.9625
## building_grade              22686.4470    40121.4567
## square_feet_total_living      141.0726      163.0379
## bedrooms                   -17615.1284      112.8368
## year_built                   1913.3863     2696.1895
```

Building_grade and year_built have very tight confidence intervals, indicating that the estimates for the model are likely to be representative of the true population values. The interval for square_feet_total_living is only slightly wider, which suggests that the variable is less representative. However, square_feet_total_living is still significant. Bedrooms appears to cross zero, indicating that in some samples the predictor has a negative relationship to the outcome whereas in others it has a positive relationship.

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.

```
anova(saleprice_lm, saleprice2_lm)
```

```
## Analysis of Variance Table
##
## Model 1: Sale_Price ~ sq_ft_lot
## Model 2: Sale_Price ~ building_grade + square_feet_total_living + bedrooms +
##     year_built
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
## 1  12863 2.0734e+15
## 2  12860 1.6367e+15  3 4.3665e+14 1143.6 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Pr(>F) equals 2.2e-16. Therefore, saleprice2_lm significantly improved the fit when compared to saleprice_lm.

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.

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.

```
housing_df$residuals<-resid(saleprice2_lm)
```

```
housing_df$standardized.residuals<- rstandard(saleprice2_lm)
housing_df$studentized.residuals<-rstudent(saleprice2_lm)
housing_df$cooks.distance<-cooks.distance(saleprice2_lm)
housing_df$dfbeta<-dfbeta(saleprice2_lm)
housing_df$dffit<-dffits(saleprice2_lm)
housing_df$leverage<-hatvalues(saleprice2_lm)
housing_df$covariance.ratios<-covratio(saleprice2_lm)
```

```
housing_df$large.residual <- housing_df$standardized.residuals>2 | housing_df$standardized.residuals< -
```

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

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

```
## [1] 327
```

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

```
housing_df[housing_df$large.residual, c("addr_full", "Sale_Price", "sq_ft_lot", "building_grade", "squa
```

```
## # A tibble: 327 x 7
##    addr_full      Sale_Price sq_ft_lot building_grade square_feet_tota~ bedrooms
##    <chr>               <dbl>     <dbl>          <dbl>             <dbl>    <dbl>
##  1 25149 NE PATT~     265000    112650             10              4920        4
##  2 19656 NE REDM~    1390000    225640              6               660        0
##  3 28527 NE 47TH~     229000    236966             10              3840        0
##  4 13414 WOODINV~     390000     63162             11              5800        5
##  5 24103 NE 122N~    1588359      8752              9              3360        2
##  6 3068 W LAKE S~    1450000     14043              6               900        2
##  7 28218 NE 40TH~     163000     18498              9              4710        4
##  8 5806 249TH CT~     270000     89734             11              5060        4
##  9 8015 228TH AV~     200000    288367             10              6880        5
## 10 11650 154TH P~     300000     55303             11              4490        4
## # ... with 317 more rows, and 1 more variable: standardized.residuals <dbl>
```

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

```
housing_df$leverage <- hatvalues(saleprice2_lm)
housing_df$cooks.distance <- cooks.distance(saleprice2_lm)
housing_df$covariance.ratios <- covratio(saleprice2_lm)
potential_problems_df <- housing_df[housing_df$large.residual, c("addr_full", "cooks.distance", "levera
average_leverage <- (5 + 1)/nrow(housing_df)
problem_leverage <- average_leverage*3
cvr_high <- 1+(3*(5+1)/nrow(housing_df))
cvr_low <- 1-(3*(5+1)/nrow(housing_df))
potential_problems_df$problems <- potential_problems_df$cooks.distance>1 | potential_problems_df$levera
potential_problems_df[potential_problems_df$problems, c("addr_full", "cooks.distance", "leverage", "cov
```

```
## # A tibble: 261 x 4
##    addr_full                    cooks.distance leverage covariance.ratios
##    <chr>                                 <dbl>    <dbl>             <dbl>
##  1 19656 NE REDMOND RD                0.00391  0.00190             0.998
##  2 28527 NE 47TH PL                   0.00253  0.00285             1.00
##  3 3068 W LAKE SAMMAMISH PKWY NE      0.00467  0.00184             0.997
##  4 5806 249TH CT NE                   0.000938 0.000722            0.999
##  5 8015 228TH AVE NE                  0.00498  0.00220             0.998
##  6 3131 269TH AVE NE                  0.00116  0.000755            0.998
##  7 20424 NE 64TH PL                   0.00337  0.00160             0.998
##  8 8024 255TH AVE NE                  0.00164  0.00108             0.999
##  9 7717 252ND AVE NE                  0.000251 0.000232            0.999
## 10 2829 W LAKE SAMMAMISH PKWY NE      0.00152  0.00156             1.00
## # ... with 251 more rows
```

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

```
dwt(saleprice2_lm)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.7299016     0.5401875       0
##  Alternative hypothesis: rho != 0
```

The dwt value is approximately 0.54, which means that due to the value being less than one, the author of "Discovering Statistics Using R" suggests that this should "definitely raise alarm bells." Therefore, the condition is not met relative to the assumption of independence.

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

```
mean(vif(saleprice2_lm))
```

```
## [1] 2.069255
```

```
(vif(saleprice2_lm))
```

```
##           building_grade square_feet_total_living                 bedrooms
##                 2.386656                 3.108761                 1.586560
##               year_built
##                 1.195045
```

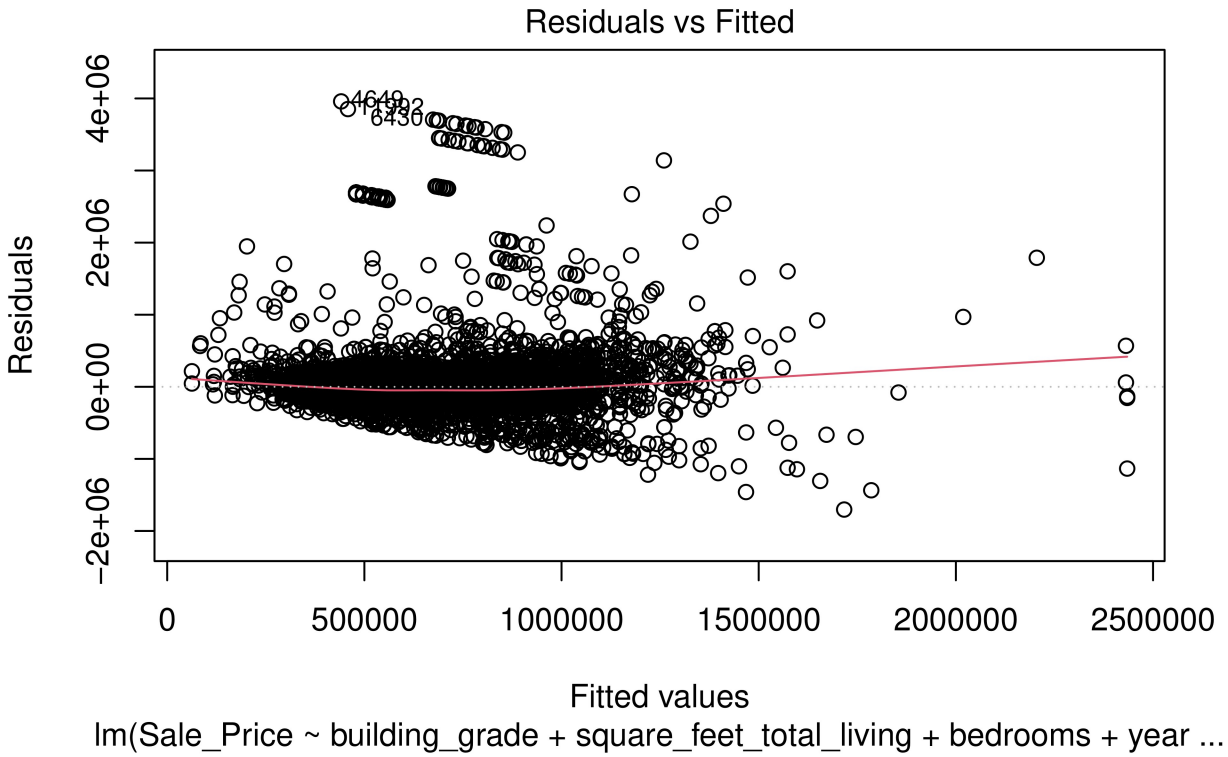
```
1/(vif(saleprice2_lm))
```

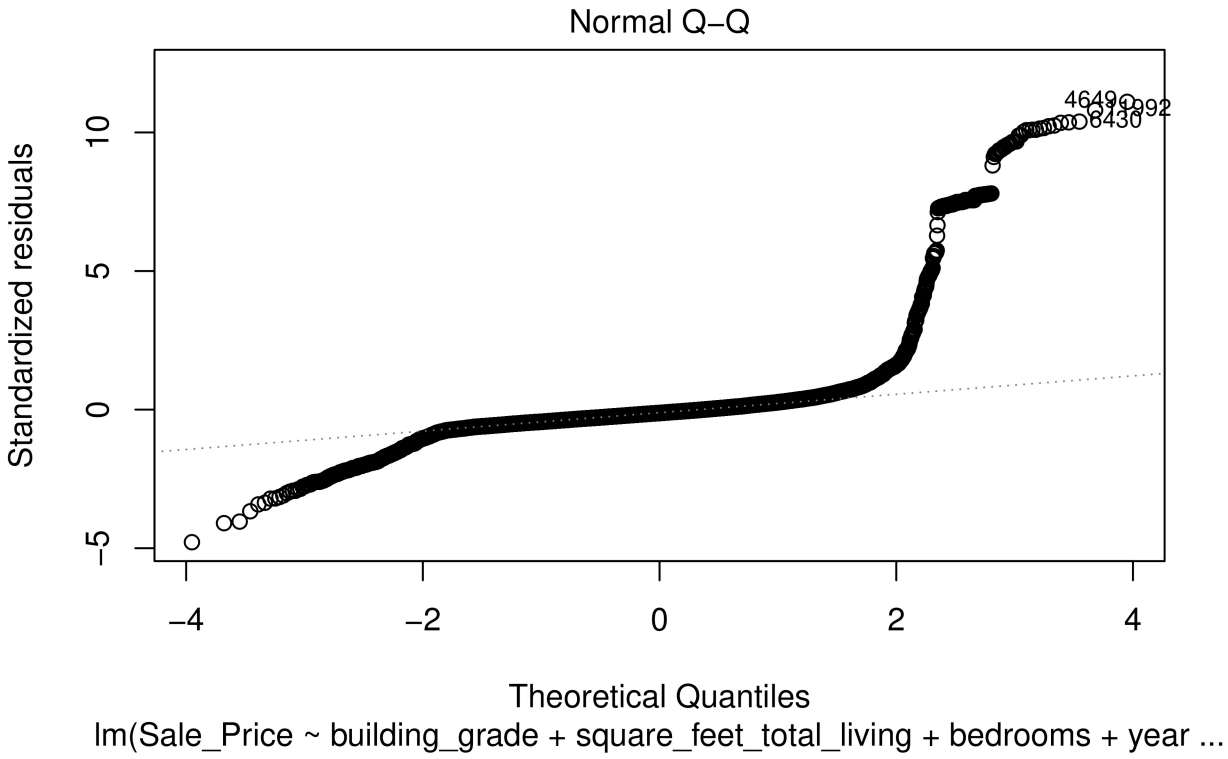
```
##           building_grade square_feet_total_living                 bedrooms
##                0.4189964                0.3216716                0.6302947
##               year_built
##                0.8367887
```

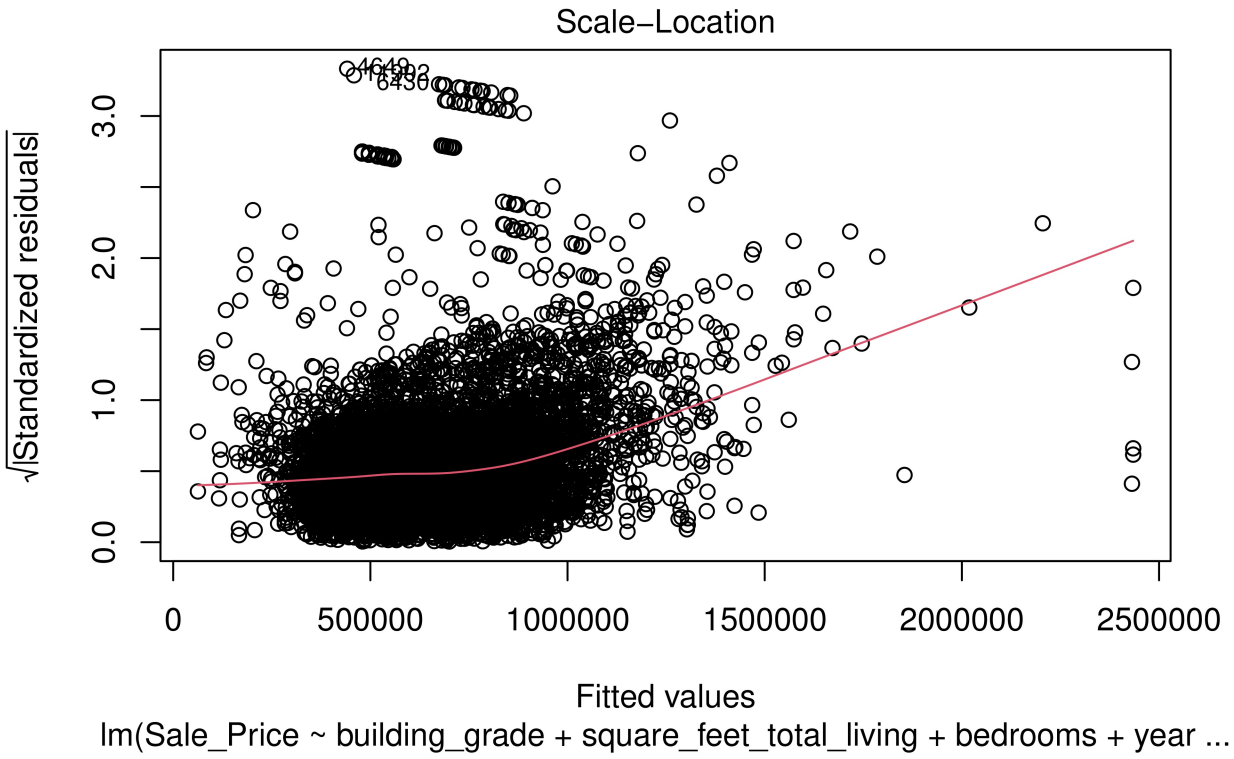
For the saleprice2 model, the VIF values are all less than 10 and the tolerance statistics are all above 0.2. The mean value is approximately 2. Based on these calculations, I would conclude that there is no collinearity within the data.

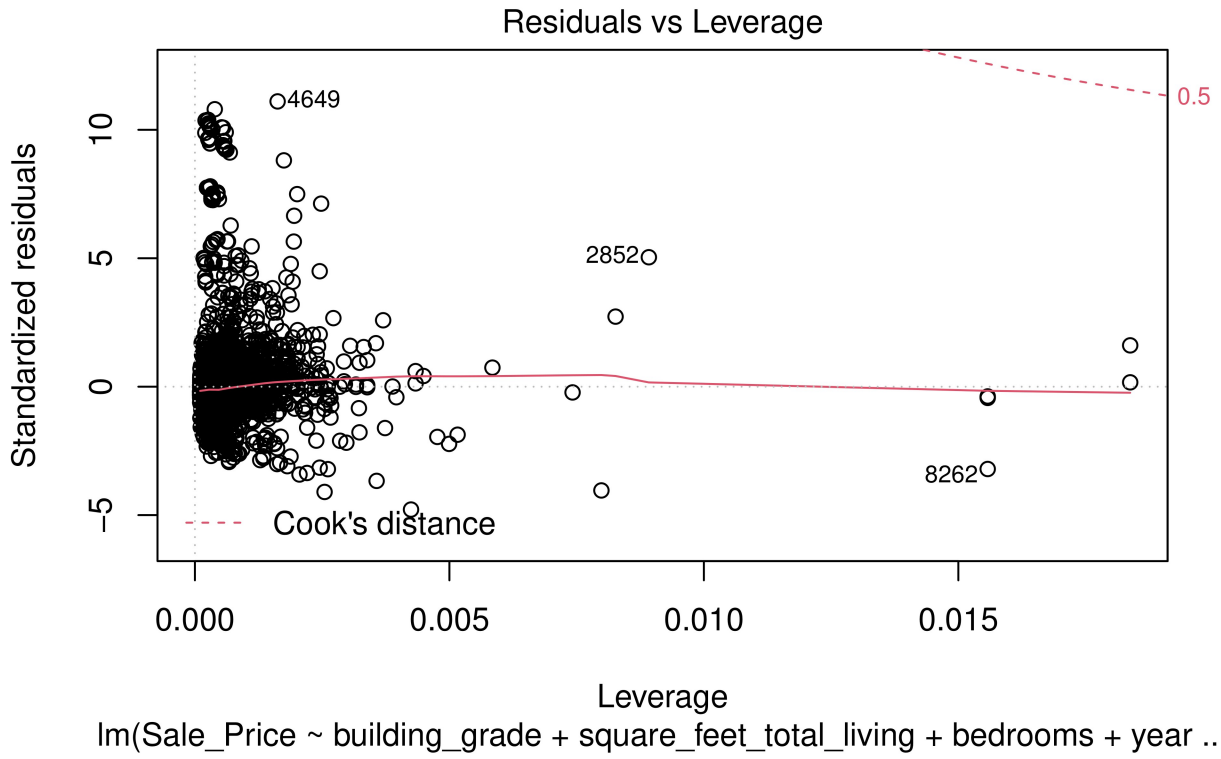
xiv. Visually check the assumptions related to the residuals using the plot() and hist() functions. Summarize what each graph is informing you of and if any anomalies are present.

```
plot(saleprice2_lm)
```

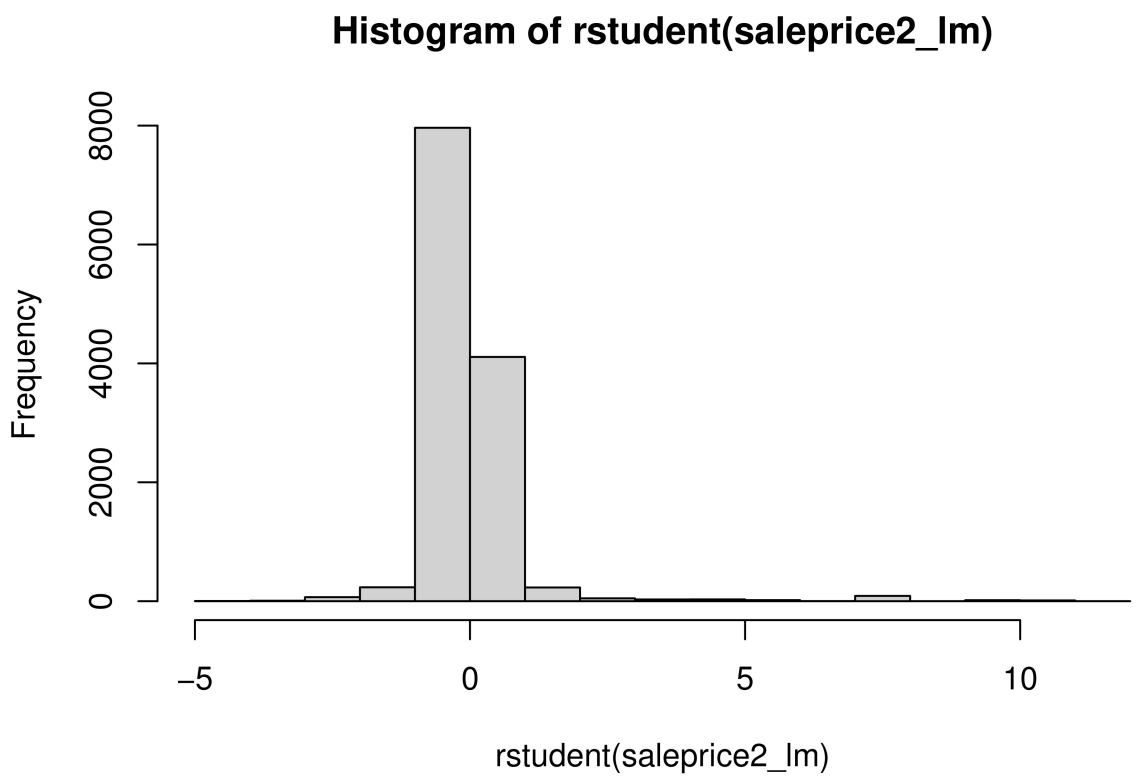

Residuals vs Fitted
Residuals
Fitted values
lm(Sale_Price ~ building_grade + square_feet_total_living + bedrooms + year ...
4e+06
2e+06
0e+00
−2e+06
0
500000
1000000
1500000
2000000
2500000


Normal Q−Q
Standardized residuals
10
5
0
−5
−4
−2
0
2
4
Theoretical Quantiles
lm(Sale_Price ~ building_grade + square_feet_total_living + bedrooms + year ...


Scale−Location
√|Standardized residuals|
0.0
1.0
2.0
3.0
0
500000
1000000
1500000
2000000
2500000
Fitted values
lm(Sale_Price ~ building_grade + square_feet_total_living + bedrooms + year ...




```
hist(rstudent(saleprice2_lm))
```



The plot demonstrates non-linearity and heteroscedasticity. The histogram and Q-Q plot resemble examples of non-normally distributed residuals.

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?

The regression model is biased.