8.2 Exercise

Harlan Wittlieff

10/31/2021

Assignment 06

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(earn ~ age, data = heights_df)
```

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

```
summary(age_lm)
```

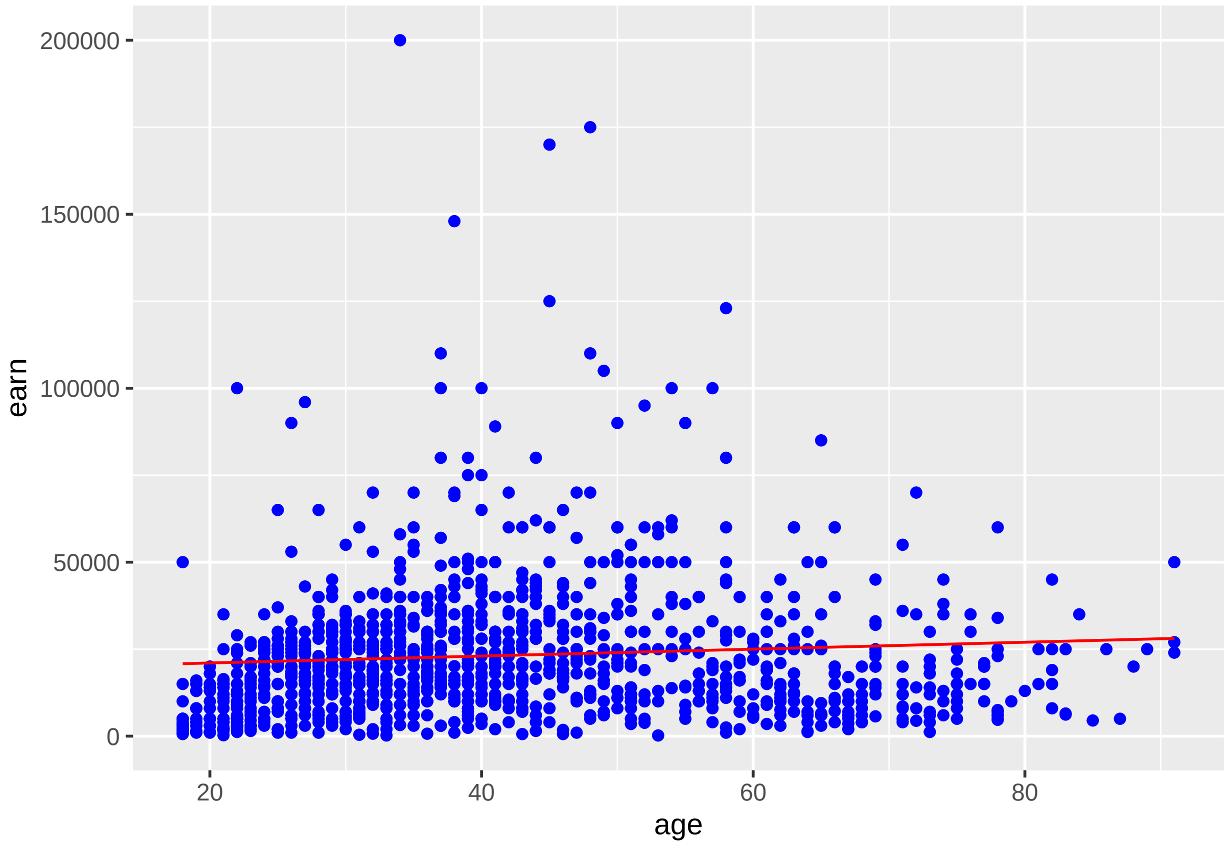
```
##
## Call:
## lm(formula = earn ~ age, data = heights_df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -25098 -12622  -3667   6883 177579
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19041.53    1571.26  12.119  < 2e-16 ***
## age            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
```

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



## Calculated values

```
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
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/dfe
mst
```

```
## [1] 379488978
```

```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
```

Assignment 07

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

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

Fit a linear model

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

View the summary of your model

```r
summary(earn_lm)
```

```
##
## Call:
## lm(formula = earn ~ height + sex + ed + age + 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 ***
## height          202.5      185.6   1.091 0.275420
## sexmale       10325.6     1424.5   7.249 7.57e-13 ***
## ed             2768.4      209.9  13.190  < 2e-16 ***
## age             178.3       32.2   5.537 3.78e-08 ***
## 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
```

Build a df of predicted values

```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
  )
```

Calculated Values

```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
r_squared
```

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

```r
## Number of observations
n <- nrow(heights_df)
## Number of regression paramaters
p <- 8
## Corrected Degrees of Freedom for Model
dfm <- p-1
## Degrees of Freedom for Error
dfe <- n-p
```

```
## Corrected Degrees of Freedom Total:   DFT = n - 1
dft <- n-1
## Mean of Squares for Model:   MSM = SSM / DFM
msm <- ssm/dfm
## Mean of Squares for Error:   MSE = SSE / DFE
mse <- sse/dfe
## Mean of Squares Total:   MST = SST / DFT
mst <- sst/dfe
## 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
```

Housing Data

Explain any transformations or modifications you made to the dataset

- Replaced spaces in column names with "_"
- Replaced any values of 0 in year_renovated with NA
- Created a new bathroom_count column based on the number of full and partial bathrooms

Create a model to predict sale price based on square foot of lot

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

Create a model that predicts sale price based on square foot of lot and several other additional predictors

Additional predictors used * building_grade * square_feet_total_living * bedrooms * custom "bathroom_count" field These fields were chosen due to their historical correlation with sales price in other markets

```
salesprice2_lm <-  lm(Sale_Price ~ sq_ft_lot + building_grade + square_feet_total_living + bedrooms + ba
```

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(salesprice_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(salesprice2_lm)
```

```
##
## Call:
## lm(formula = Sale_Price ~ sq_ft_lot + building_grade + square_feet_total_living +
##     bedrooms + bathroom_count, data = housing_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1962684  -113545   -41956    40076  3754813
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -3.968e+04  3.170e+04  -1.252 0.210641
## sq_ft_lot                 1.279e-01  5.794e-02   2.207 0.027309 *
## building_grade            4.005e+04  4.422e+03   9.056  < 2e-16 ***
## square_feet_total_living  1.487e+02  6.465e+00  22.993  < 2e-16 ***
## bedrooms                 -2.000e+04  4.564e+03  -4.382 1.19e-05 ***
## bathroom_count            2.408e+04  6.975e+03   3.452 0.000558 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 358400 on 12859 degrees of freedom
## Multiple R-squared:  0.2148, Adjusted R-squared:  0.2145
## F-statistic: 703.6 on 5 and 12859 DF,  p-value: < 2.2e-16
```

From the original model, the R2 values is 0.01435 and the adjusted R2 value is 0.01428. This means that not much predictive power is lost due to shrinkage, however the model only can account for 1.4% of the variation in sale price.

The second model has an R2 value of .2148 and a adjusted R2 value of 0.2145. This model also does not lose much predictive power due to shrinkage. The inclusion of the additional predictors increased the predictability of the model when compared to the original.

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('QuantPsyc')
```

```
## Loading required package: boot

## Loading required package: MASS

##
## Attaching package: 'QuantPsyc'

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

```
lm.beta(salesprice2_lm)
```

```
##                sq_ft_lot           building_grade square_feet_total_living
##               0.01800524               0.10821336               0.36386458
##                 bedrooms           bathroom_count
##              -0.04332910               0.04139706
```

- sq_ft_lot
  - as sq_ft_lot increases by 1 standard deviation, sales prices increases by 0.018 standard deviations
- building_grade
  - as building_grade increases by 1 standard deviation, sales prices increases by 0.108 standard deviations
- square_feet_total_living
  - as square_feet_total_living increases by 1 standard deviation, sales prices increases by 0.364 standard deviations
- bedrooms
  - as bedrooms increases by 1 standard deviation, sales prices decreases by 0.043 standard deviations
- bathroom_count
  - as bathroom_count increases by 1 standard deviation, sales prices increases by 0.041 standard deviations

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

```
confint(salesprice2_lm)
```

```
##                                  2.5 %        97.5 %
## (Intercept)               -1.018186e+05  2.245201e+04
## sq_ft_lot                  1.432041e-02  2.414518e-01
## building_grade             3.138161e+04  4.871813e+04
## square_feet_total_living   1.359808e+02  1.613265e+02
## bedrooms                  -2.894548e+04 -1.105206e+04
## bathroom_count             1.040719e+04  3.775283e+04
```

square_feet_total_living has a tight confidence interval indicating that the true value is likely represented in the model. None of the other variables cross 0, indicating that they may be less representative, but still significant.

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(salesprice_lm, salesprice2_lm)
```

```
## Analysis of Variance Table
##
## Model 1: Sale_Price ~ sq_ft_lot
## Model 2: Sale_Price ~ sq_ft_lot + building_grade + square_feet_total_living +
##     bedrooms + bathroom_count
##   Res.Df        RSS Df  Sum of Sq     F    Pr(>F)
## 1  12863 2.0734e+15
## 2  12859 1.6517e+15  4 4.2167e+14 820.7 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The value of Pr(>F) is equal to 2.2e-16. salesprice2_lm significantly improved the fit of the model to the data compared to salesprice_lm.

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.

```
housing_df$standardized.residuals <- rstandard(salesprice2_lm)
housing_df$large.residual <- housing_df$standardized.residuals>2 | housing_df$standardized.residuals< -
```

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

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

```
## [1] 319
```

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: 319 x 8
##    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 309 more rows, and 2 more variables: bathroom_count <dbl>,
## #   standardized.residuals <dbl>
```

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

```
housing_df$leverage <- hatvalues(salesprice2_lm)
housing_df$cooks.distance <- cooks.distance(salesprice2_lm)
housing_df$covariance.ratios <- covratio(salesprice2_lm)
potential_problems_df <- housing_df[housing_df$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", "cova
```

```
## # A tibble: 275 x 4
##    addr_full                          cooks.distance leverage covariance.ratios
##    <chr>                                       <dbl>    <dbl>             <dbl>
##  1 19656 NE REDMOND RD                      0.00358  0.00255              0.999
##  2 28527 NE 47TH PL                         0.00421  0.00597              1.00
##  3 13414 WOODINVILLE REDMOND RD NE          0.000994 0.000966             0.999
##  4 3068 W LAKE SAMMAMISH PKWY NE            0.000981 0.000591             0.996
##  5 28218 NE 40TH ST                         0.000830 0.000831             0.999
##  6 5806 249TH CT NE                     0.509    0.149                 1.17
##  7 8015 228TH AVE NE                    0.00620  0.00315               0.998
##  8 3131 269TH AVE NE                    0.00386  0.00265               0.999
##  9 20424 NE 64TH PL                     0.00297  0.00174               0.997
## 10 8024 255TH AVE NE                    0.00169  0.00149               0.999
## # ... with 265 more rows
```

The listed values fail the tests for either cooks distance, leverage, or covariance ratios.

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

```
library('car')
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

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

```
dwt(salesprice2_lm)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.7409137     0.5181686       0
##  Alternative hypothesis: rho != 0
```

The D-W Statistic is less than 1, meaning the model fails the test of independence.

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

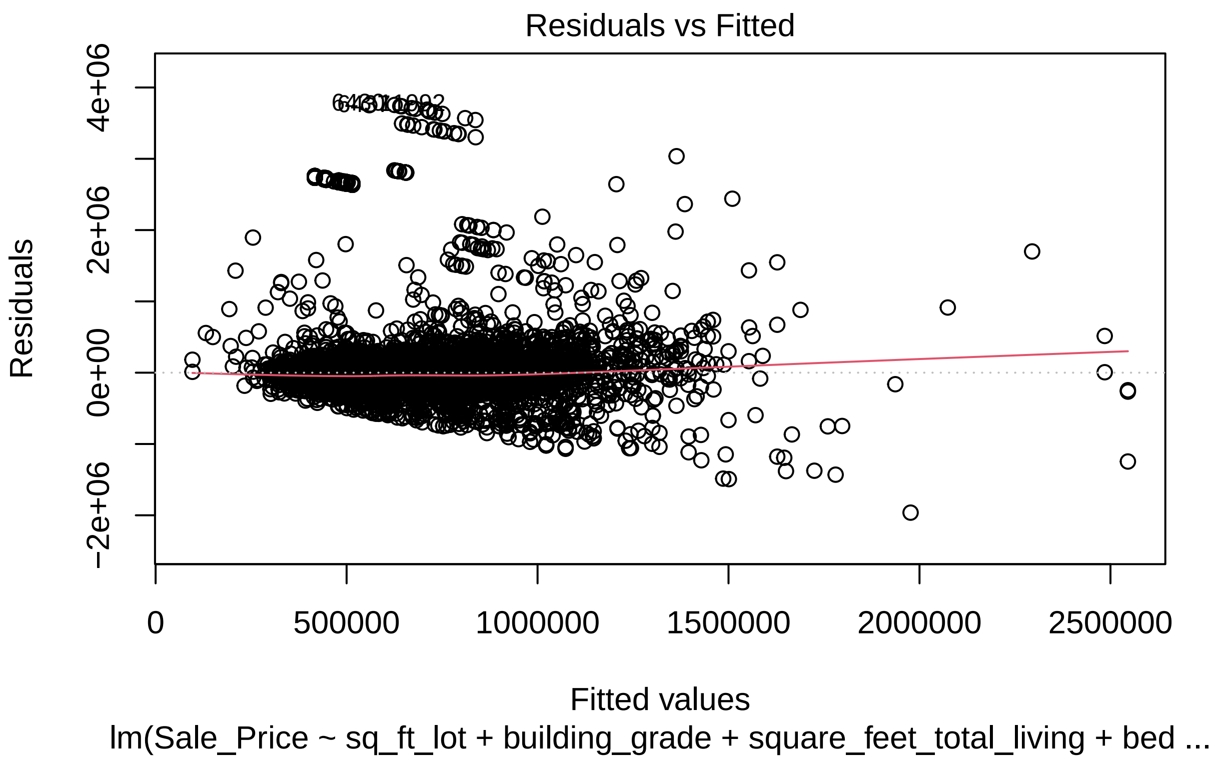
```
mean(vif(salesprice2_lm))
```

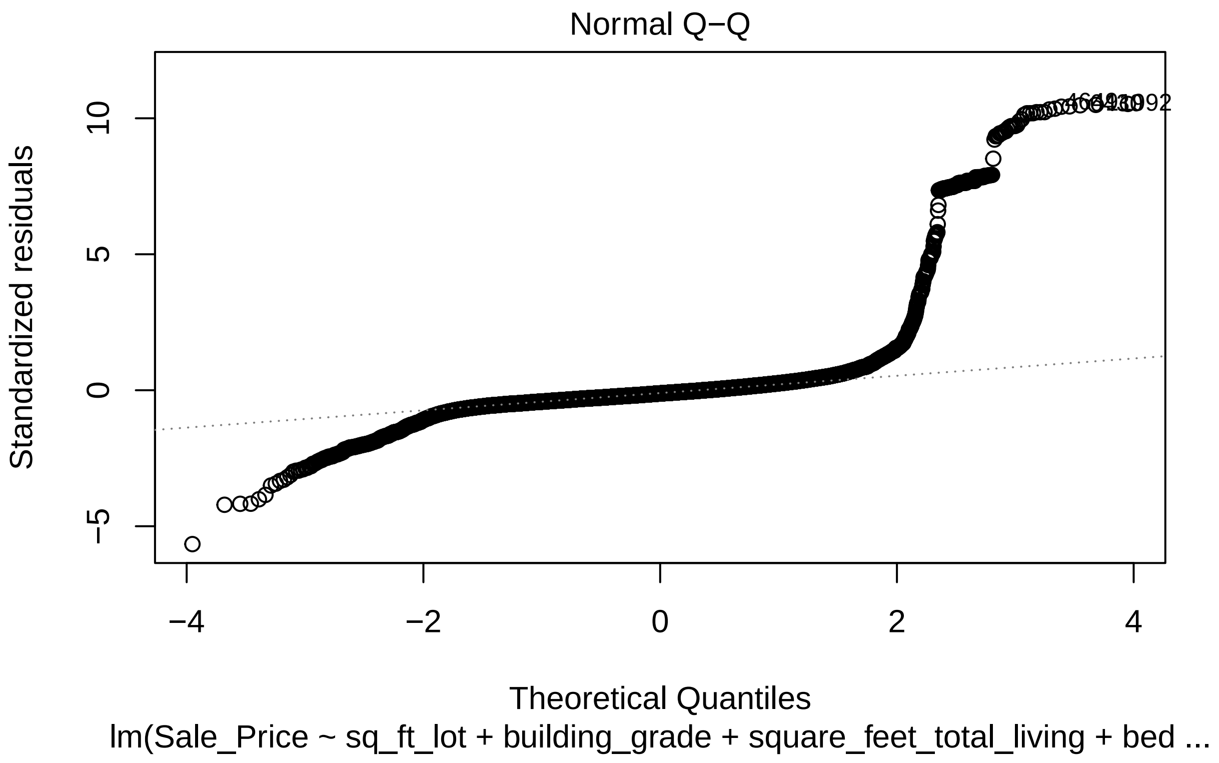
```
## [1] 2.297156
```

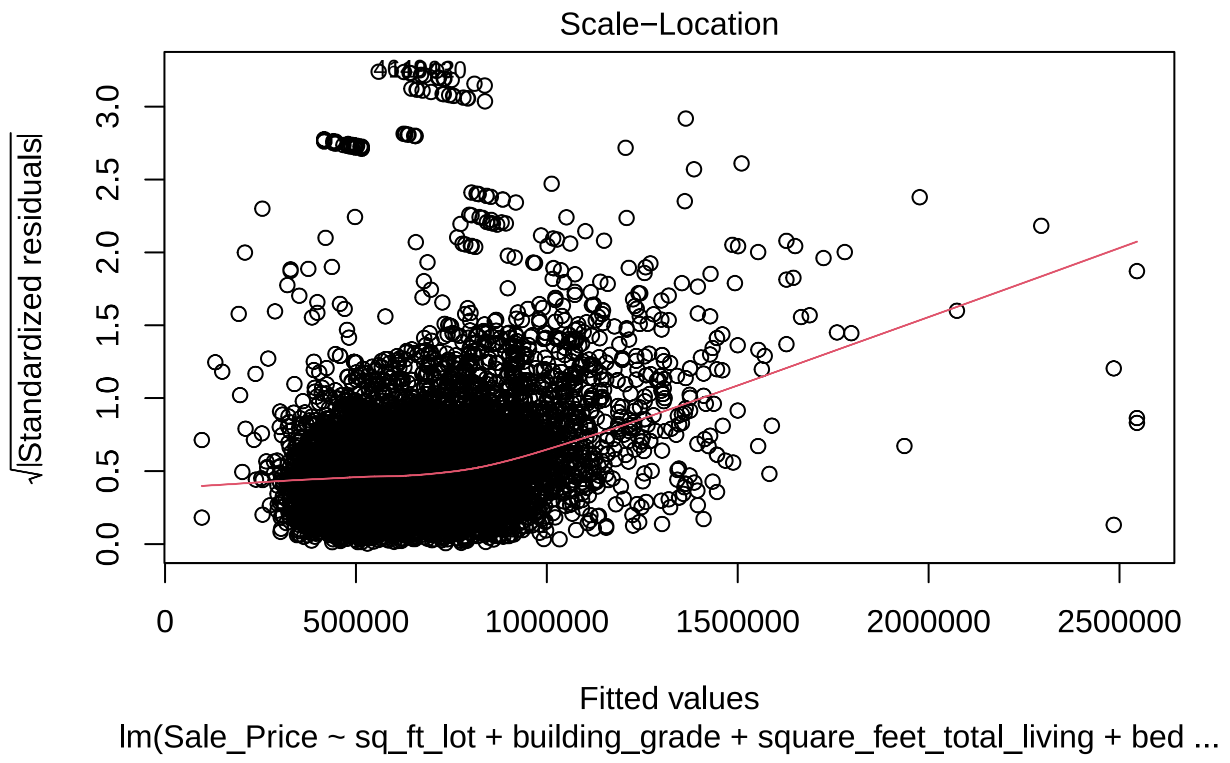
The result of 2.3 indicates that the regression may be biased.

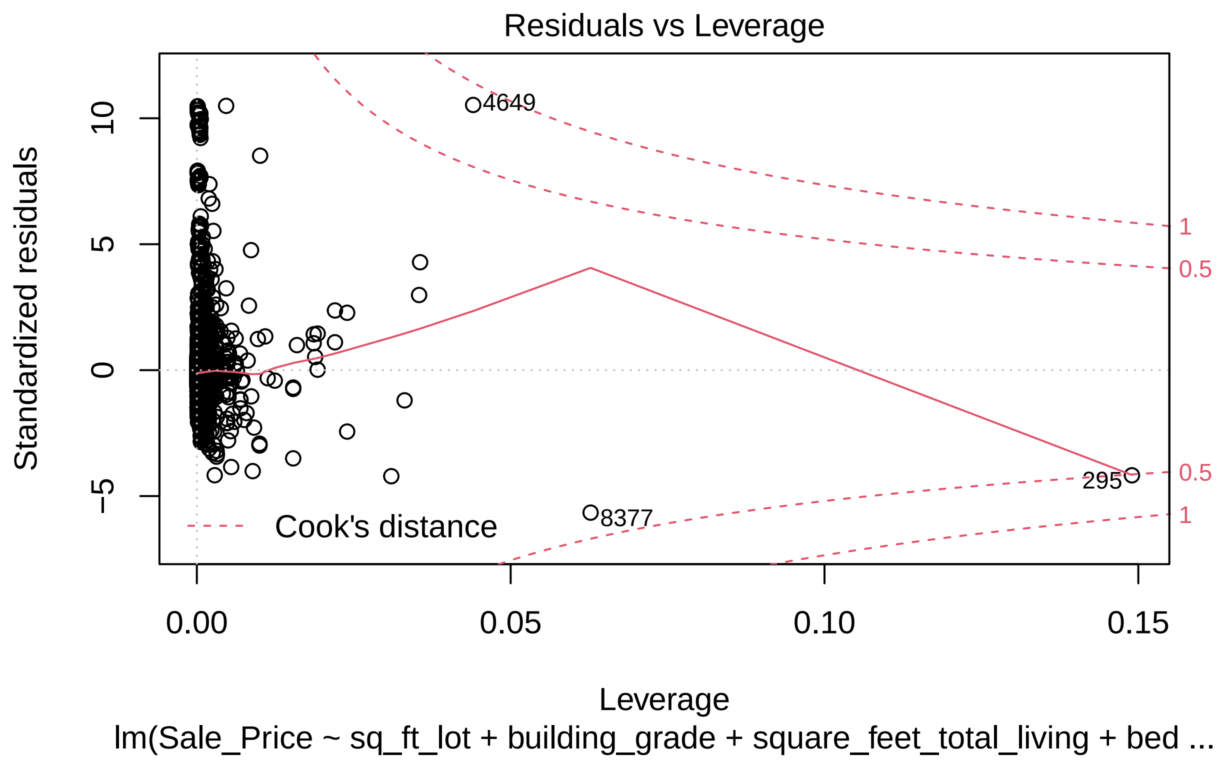
Visually check the assumptions related to the residuals using the plot() and hist() functions. Summarize what each graph is informing you of and if any anomalies are present.

```
plot(salesprice2_lm)
```

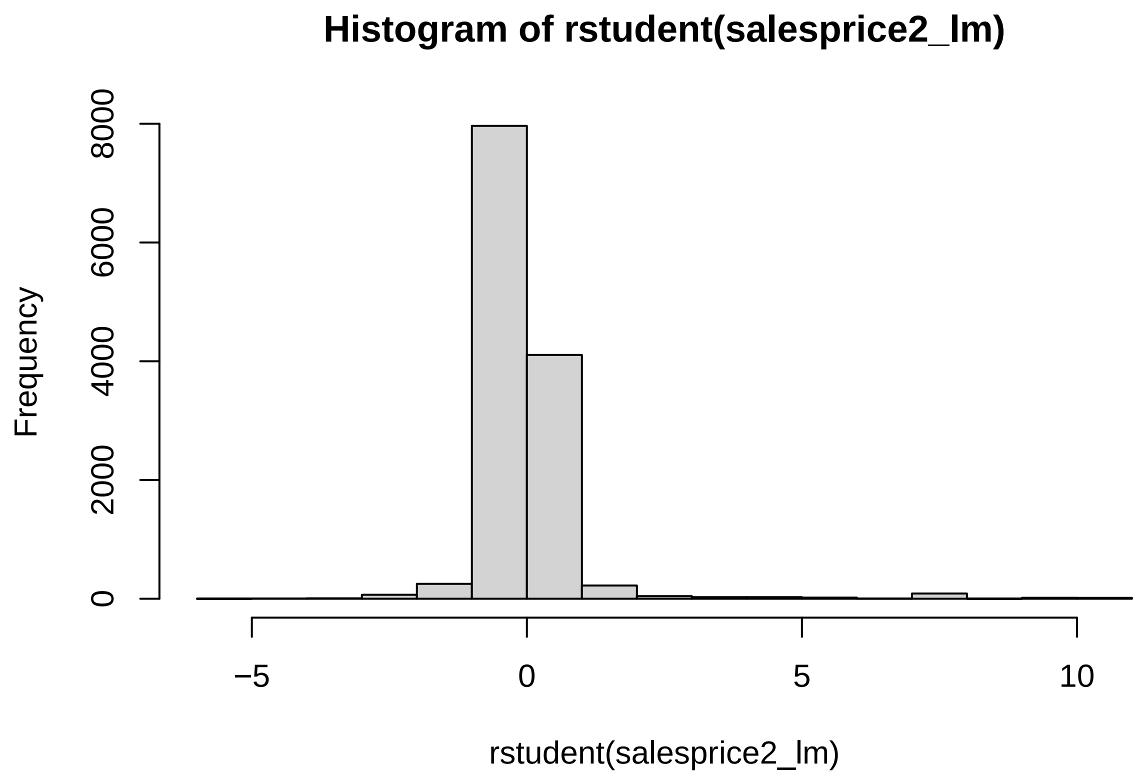

Residuals vs Fitted
Residuals
4e+06
2e+06
0e+00
−2e+06
0
500000
1000000
1500000
2000000
2500000
Fitted values
lm(Sale_Price ~ sq_ft_lot + building_grade + square_feet_total_living + bed ...


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


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
500000
1000000
1500000
2000000
2500000
Fitted values
lm(Sale_Price ~ sq_ft_lot + building_grade + square_feet_total_living + bed ...



```
hist(rstudent(salesprice2_lm))
```



The plots show the potential for non-linearity and heteroscedasticity in the model.

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

Overall, this regression model is biased.