RMarkdown Week 8 & 9

Dipika Sharma

May 16, 2021

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Add Citations

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

Assignment 06

Set the working directory to the root of your DSC 520 directory Load the `data/r4ds/heights.csv` to

```
## Set the working directory to the root of your DSC 520 directory
setwd("/Users/dipikasharma/R_Projects/DSC520")

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

Fit a linear model using the `age` variable as the predictor and `earn` as the outcome. View the summary of your model using `summary()`

```
## 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)
##
```

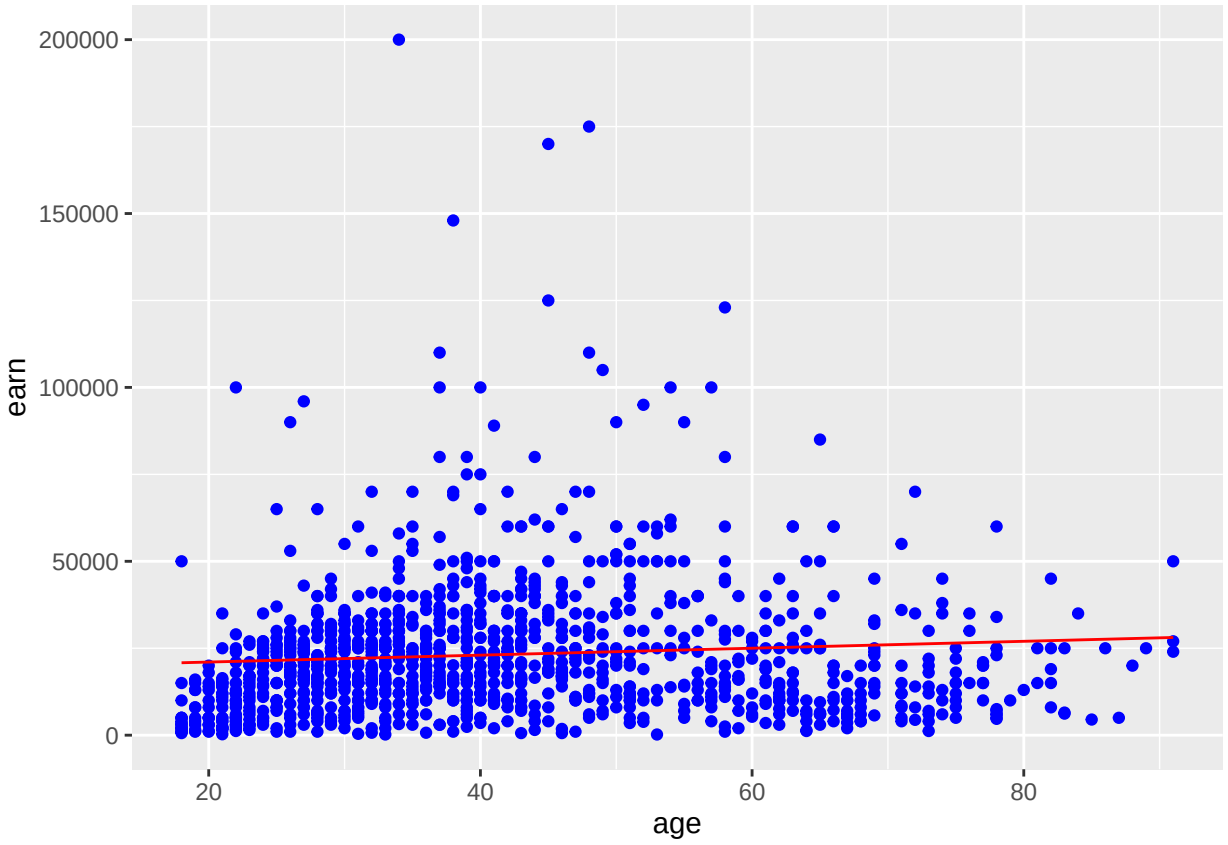
```
## Residuals:
##    Min     1Q Median     3Q    Max
## -25098 -12622  -3667   6883 177579
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19041.53    1571.26  12.119  < 2e-16 ***
## age            99.41      35.46   2.804  0.00514 **
## ---
## Signif. codes:  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()`

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

Plot the predictions against the original data

```
##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))
```



Compute deviation (i.e. residuals)

```
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] 2963111900
```

Residuals

```
residuals <- heights_df$earn - age_predict_df$earn
residuals
```

```
##    [1]  26485.214165  35192.939138   8075.706507  21912.548684  28081.648793
##    [6] -12626.076179   5087.591080   8385.808394 -19129.047323   5373.923821
##   [11] -18972.901261   7578.677650  -9725.481951 -12111.220463   -520.728121
##   [16]  -7807.060862  18075.706507  20584.619937 -17508.843548  30479.271879
##   [21] -19111.220463  -7129.047323 -15728.453094  16882.837251  10485.214165
##   [26] -12520.728121   1994.127596  27181.054565  18678.083421  -6526.670408
##   [31]   2678.083421  52081.648793   4876.894964  -9626.076179 -19295.176288
##   [36]   7876.894964  -3707.655091   5373.923821 -22502.901261   2976.300735
##   [41]    882.837251  10075.706507 -12023.699265 -10129.047323  -3522.510807
##   [46]  -1924.293493 -17330.830009 -11608.249319   -502.901261  -2117.162749
##   [51]  10087.591080 -19502.901261  -4824.887722  12777.489193 -16830.830009
##   [56] -10508.843548 -14707.655091   3075.706507  -5725.481951  -5824.887722
##   [61]  -6815.974292 -21801.118575 -20370.626234   -918.351207 -24110.032005
##   [66]   4274.518049 -18629.047323  -5918.351207 -13228.453094 -13801.118575
##   [71] -13713.597377 -15701.712804  -7918.351207 -20518.351207   7777.489193
##   [76]   7678.083421 -12818.945435 -16626.076179 -13304.089718  -5105.278176
##   [81]  -2620.133893  -1327.858865 -19829.641551 -21522.510807  -1725.481951
##   [86] -11228.453094   3701.852568  15391.750681  -2888.639773 -16894.582060
##   [91]  34695.910282  -7602.307032  -6281.916579  -7023.699265 -11327.858865
##   [96]   9684.025708  -4023.699265  -7719.539664   -228.453094  -2315.974292
##  [101]  14280.460336  14584.619937  -6315.974292   9976.300735   2692.939138
##  [106]  -6915.380063   2479.271879   3684.025708 -16824.887722  47181.054565
##  [111] -18017.756978 -16427.264637  -1626.076179 -17129.047323  37479.271879
##  [116] -17123.105036   7181.054565  47479.271879  26882.837251  21081.648793
##  [121]   6684.025708 -13017.756978   -216.568521  19795.316053  -7123.105036
```

```
##  [126] -22099.335889  -5123.105036   7777.489193   5476.300735 -20212.524346
##  [131]  -6608.249319  19584.619937   8584.619937 -14608.249319  36485.214165
##  [136]  -3011.814691 -18828.453094 -20824.887722   5274.518049 -15224.887722
##  [141]  -7315.974292  68373.923821 -15216.568521  -2222.510807  -5830.830009
##  [146] -13005.872404    385.808394  -2918.351207  -4614.191606 -13912.408920
##  [151]  17578.677650   -403.495490   3572.735363   2777.489193  -6105.278176
##  [156]  20186.996851  -8912.408920  -5620.133893   -608.249319    280.460336
##  [161] -13397.553203  41385.808394 -18304.089718  14695.910282  -8315.974292
##  [166]  -1526.670408  -4011.814691  -3216.568521   1087.591080  26280.460336
##  [171]   -502.901261 -13707.655091  -6818.945435  11684.025708 101485.214165
##  [176]   -713.597377  -4824.887722   3882.837251  47777.489193  11286.402623
##  [181] -15204.683947  12081.648793  -7818.945435 -10918.351207 -17801.118575
##  [186] -16011.814691  10988.185309  20888.779537  -6228.453094 -11695.770517
##  [191]   -111.220463   2777.489193    590.562223  -1824.887722   2572.735363
##  [196]  20385.808394  46181.054565  38584.619937   9777.489193  -2321.916579
##  [201]   9479.271879    988.185309 146485.214165  12678.083421  17181.054565
##  [206]  10976.300735  -3228.453094   6684.025708   3876.894964 -16129.047323
##  [211]  -6298.147432  -5023.699265  10175.112278  -6228.453094  25491.156452
##  [216] -14918.351207  15888.779537  16882.837251  10328.677650  -5900.524346
##  [221]  13081.648793 -19403.495490 -10421.322350  34695.910282  16584.619937
##  [226]  20181.054565  22578.677650 -16918.351207 -19192.799374  -2918.351207
##  [231]  -4327.858865 -20222.510807  43175.112278  26882.837251 -11818.945435
##  [236]  12976.300735   4678.083421 -21605.278176  19280.460336  -1924.293493
##  [241]  -6924.293493 -11129.047323 -20795.176288  -4327.858865   9497.098739
##  [246]   6175.112278  -7321.916579  -1427.264637  -1924.293493 -12716.568521
##  [251]  -8228.453094 -14210.626234 -19719.539664   2075.706507  -5520.728121
##  [256] -12813.003149   8982.243022  -4626.076179 -19520.728121 -19029.047323
##  [261] -19229.322350  -4626.076179   8175.112278  -7222.510807    783.431479
##  [266] -15701.712804  25689.967995  25689.967995  -3912.408920   5590.562223
##  [271]  -1514.785835   5572.735363 -18818.945435  -4126.076179  -5324.887722
##  [276]   5081.648793  27988.185309  -6924.293493  -3222.510807   5672.141135
##  [281]  -9807.060862  -7321.916579 -10008.843548   1976.300735  -4620.133893
##  [286] -21801.118575 -20707.655091 -21920.133893  -4087.451316   4081.648793
##  [291] -10520.728121  -1713.597377   7379.866107   9894.721824  -3216.568521
##  [296]   9777.489193 -16123.105036 -10918.351207 -12123.105036 -19830.830009
##  [301] -14099.335889  -6620.133893   2678.083421   3274.518049   9976.300735
##  [306]  11684.025708 -15824.887722  56584.619937  -1526.670408 -21818.945435
##  [311]   5181.054565   4672.141135   5572.735363  -5117.162749 -22906.466633
##  [316]  -5602.307032 -17813.003149   -304.089718  -9924.293493 -12707.655091
##  [321]   3403.635254 -17228.453094   3280.460336    479.271879  -1321.916579
##  [326]   -204.683947  -8496.958974 -19415.380063   2671.546906  11982.243022
##  [331]   3976.300735   2379.866107   1774.518049  -9228.453094  -6029.641551
##  [336] -18327.858865 -19701.712804 -11900.524346  -7123.105036 151186.996851
##  [341] -20526.670408 -15304.089718  17807.200626 -12093.393602  -3117.162749
##  [346]  12976.300735  18175.112278   8801.258340  25590.562223  75292.344909
##  [351]  13175.112278   1479.271879  11684.025708  11602.446797   5789.373766
##  [356] -17918.351207 125181.054565   7777.489193 -15623.105036 -19719.539664
##  [361]  -2900.524346  16882.837251   7081.648793  -8719.539664  -7421.322350
##  [366]   3379.866107 -13526.670408   2274.518049 -20577.712804 -11526.670408
##  [371] -10222.510807  27578.677650   1373.923821  17578.677650 -15626.076179
##  [376] -17023.699265  87280.460336  18777.489193  -3795.176288   -327.858865
##  [381] -17924.293493   2379.866107   6385.808394 -10111.220463 -16520.728121
##  [386]  -9614.191606  21274.518049   2678.083421  14397.692968  41982.243022
##  [391]  -6222.510807 -18099.335889  -2918.351207 -12198.741660  18602.446797
```

```
##  [396] -18198.741660  -9824.887722 -14123.105036  -6222.510807 -20298.147432
##  [401]  -3222.510807  -1918.351207  19385.808394  13870.952677 -18496.958974
##  [406]  -2117.162749 -16223.699265  -6614.191606     87.591080 -21099.335889
##  [411] -10327.858865 -11725.481951  17379.866107  16286.402623   2075.706507
##  [416]   2373.923821  -2321.916579  35988.185309 -12713.597377 -18017.756978
##  [421]   4976.300735  65882.837251  18175.112278  -6626.076179 -20029.641551
##  [426]  -3005.872404 -11813.003149    175.112278  15391.750681  16584.619937
##  [431]  10689.967995 -14824.887722  -8719.539664  -9608.249319 -16421.322350
##  [436] -12117.162749 -12123.105036  -3117.162749 -19298.147432  14385.808394
##  [441]   4192.939138  -3415.380063 -16707.655091  11888.779537 -21915.380063
##  [446]  20075.706507   3771.546906  -8029.641551   5888.779537 -14327.858865
##  [451]   7186.996851  -8216.568521  26087.591080 -16029.641551 -20230.830009
##  [456] -10111.220463  17479.271879   1379.866107    572.735363  -2620.133893
##  [461]  -2706.945435 -13701.712804  17982.243022 -11011.814691  37876.894964
##  [466]   2473.329592  15379.866107 -18725.481951  -9023.699265 -16129.047323
##  [471]  -2719.539664    175.112278   3491.156452  -1117.162749   6584.619937
##  [476]   7777.489193   7876.894964   6099.475654   2684.025708   -421.322350
##  [481]   2678.083421 -10427.264637 -11310.032005 -13216.568521 -11514.785835
##  [486] -18403.495490  34280.460336  13280.460336   7075.706507 -12930.235780
##  [491] -17830.830009  -3117.162749   8292.344909 -10801.118575  -6321.916579
##  [496] -21795.176288 -15315.974292   9578.677650  -6927.264637  -8924.293493
##  [501] -14526.670408  24497.098739 -10520.728121  -6105.278176 -15508.843548
##  [506]  -2520.728121  32479.271879 -13017.756978  -7930.235780  -2123.105036
##  [511]   8373.923821 -10421.322350   3678.083421  -9225.481951 -13924.293493
##  [516]  -7222.510807  43473.329592   3870.952677 -13129.047323  -6129.047323
##  [521] -11017.756978  12280.460336  17876.894964   7771.546906  -7123.105036
##  [526]  -2620.133893 -22204.683947 -19930.235780 -13327.858865 -15830.830009
##  [531]   3274.518049  -4707.655091   7578.677650 -10520.728121 -12222.510807
##  [536]  -4930.235780  23075.706507  15590.562223  -2093.393602  -5210.626234
##  [541]  -4423.699265 -18701.712804   7777.489193  -5713.597377   1286.402623
##  [546] -19204.683947  46783.431479  -7421.322350   2678.083421  13175.112278
##  [551]  -3427.264637   5976.300735  -6129.047323    175.112278  12976.300735
##  [556]  -5614.191606  12280.460336  -4729.047323   -719.539664   5192.939138
##  [561]  -4725.481951   2379.866107 -14111.220463  17280.460336 -12804.089718
##  [566]  11485.214165  81087.591080  76982.243022  -3111.220463 -14608.249319
##  [571]  11373.923821   4473.329592  33204.823712  -5514.785835 -10105.278176
##  [576] -16029.641551  36783.431479 -13900.524346   5181.054565  18684.025708
##  [581] -20321.916579  -2321.916579   4982.243022 -15204.683947   8876.894964
##  [586]  -5719.539664  -6918.351207  12181.054565 -18626.076179  -6918.351207
##  [591]  11783.431479   4678.083421   3075.706507  16882.837251  21684.025708
##  [596]  -7327.858865   2280.460336  16783.431479  11479.271879  17578.677650
##  [601]  13175.112278 -12111.220463   6684.025708    -17.756978 -20023.699265
##  [606]  -1526.670408 -15204.683947  36684.025708 -11117.162749   -105.278176
##  [611]  -4298.147432 -13526.670408 -19129.047323  -3026.076179  -1029.641551
##  [616]  77280.460336  26982.243022   3783.431479 -23807.060862  -3029.641551
##  [621]  -4327.858865 -16397.553203  -8520.728121 -18402.307032  10081.648793
##  [626]    578.677650 -13117.162749    572.735363  -7930.235780  10876.894964
##  [631]  13379.866107 -12222.510807 -19005.872404 -11315.974292 -13713.597377
##  [636]  -2514.785835   3777.489193  -2011.814691  -1409.437777  -1427.264637
##  [641] -12818.945435 -11099.335889 -14427.264637  -7818.945435   3578.677650
##  [646] -12123.105036 -18029.641551   3882.837251   6473.329592  -6725.481951
##  [651]  32976.300735  -1626.076179 -14915.380063   7976.300735  -2123.105036
##  [656]  -4496.958974   3373.923821 -18228.453094  -7912.408920 -12918.351207
##  [661]  -6824.887722   4385.808394   8979.271879  70789.373766  14982.243022
```

```
##  [666]   6882.837251  12683.431479  -9228.453094 -11526.670408  21584.619937
##  [671]  15391.750681   -225.481951 -11204.683947  13590.562223 -19830.830009
##  [676] -12298.147432  -7123.105036    783.431479 -16827.858865  -7304.089718
##  [681]  -7029.641551 -11824.887722 -12722.510807 -14602.307032  -8701.712804
##  [686] -15824.887722 -22307.060862  16882.837251   2081.648793   2994.127596
##  [691]   9485.214165 -21403.495490 -17923.699265  -2192.799374  10175.112278
##  [696] -11807.060862 -13725.481951  -5818.945435 -11695.770517  -4222.510807
##  [701]  -9427.264637 -16912.408920 -20028.453094  -7129.047323  -6017.756978
##  [706]   6783.431479    689.967995   -321.916579  -6129.047323  -1087.451316
##  [711]  -1626.076179 -13421.322350  -5023.699265 -21814.191606  -5614.191606
##  [716]   1783.431479  -3117.162749  51982.243022 -19517.756978   9900.664111
##  [721]   8473.329592  -1795.176288 -20626.076179  34397.692968   8403.635254
##  [726] -13321.916579   7608.389084 -13807.060862 -12210.626234   6684.025708
##  [731]  -5707.655091 -16105.278176   8075.706507 -13222.510807  -7023.699265
##  [736] -14123.105036 -16496.958974  10280.460336  -9824.887722  -4719.539664
##  [741]   4894.721824  -9722.510807 -12620.133893 -17599.335889  -2614.191606
##  [746]  -5695.770517   -526.670408 -19210.626234   -620.133893   8373.923821
##  [751]  98192.939138 -15514.785835 -19526.670408 -17604.683947 -11526.670408
##  [756]  11186.996851  -9707.655091 -17228.453094  -5427.264637 -22121.916579
##  [761]   1473.329592  -5725.481951 -17912.408920 -23014.191606   5578.677650
##  [766] -21602.307032  -7520.728121  -6526.670408   2976.300735  -6626.076179
##  [771]  16584.619937   5888.779537 -16924.293493  12578.677650  -6123.105036
##  [776]   4982.243022  -3216.568521  -1023.699265  -6216.568521    -11.814691
##  [781]  -1496.958974  16385.808394 -12930.235780  -4514.785835  19099.475654
##  [786] -17321.916579    982.243022 -15930.235780  10578.677650   2572.735363
##  [791] -15327.858865  65491.156452  33286.402623 -10609.437777   7280.460336
##  [796]  -8017.756978 -11496.958974   9075.706507   9982.243022   4894.721824
##  [801] -10818.945435 -16228.453094  -3906.466633   4385.808394 -10123.105036
##  [806]  -4614.191606 -16520.728121 -19705.872404  -6824.887722  -6614.191606
##  [811] -22095.176288 -16002.901261 -19528.453094 -16723.105036 -12123.105036
##  [816]  -2292.205145  11684.025708   4584.619937  18982.243022  20192.939138
##  [821]  30888.779537  13976.300735   2876.894964  -2824.887722  13988.185309
##  [826]  -7719.539664 -22991.016688  74274.518049 -12713.597377 -13421.322350
##  [831] -16427.264637  25888.779537 -17415.380063  78771.546906  -7114.191606
##  [836]   8982.243022   6882.837251 -13216.568521  -5930.235780 -20792.205145
##  [841]    695.910282  -7906.466633  -2906.466633 -13117.162749  25192.939138
##  [846]  -2520.728121 -21723.105036  -8906.466633 -15602.307032 -10807.060862
##  [851] -16029.641551  17578.677650 -19508.843548   1286.402623 -12111.220463
##  [856]  35578.677650  -3626.076179 -19830.830009  -2713.597377 -14204.683947
##  [861]  -9620.133893  -7789.234002   2479.271879 -22715.974292   3373.923821
##  [866]  14175.112278  -7321.916579 -20611.220463  -3416.322350  37590.562223
##  [871]   1783.431479  25789.373766  14894.721824   4373.923821  -1813.003149
##  [876]  -1918.351207 -19394.901261   4596.504510 -20795.176288   5075.706507
##  [881] -19216.568521 -21782.741660 -22689.828230  21982.243022  -5514.785835
##  [886] -15228.453094  -7427.264637  13572.735363  -6924.293493  -9924.293493
##  [891] -22713.597377   9678.083421 -19430.235780 -18327.858865  11783.431479
##  [896]  46186.996851   -824.887722  29169.169991  -6129.047323 -13520.728121
##  [901] -11526.670408   6286.402623  -5129.047323  -4111.220463  75590.562223
##  [906] -13315.974292   9695.910282 -15123.105036 -13725.481951 -13011.814691
##  [911]   1286.402623   1485.214165   6186.996851  25578.677650  -6614.191606
##  [916]   4572.735363 -15105.278176  35689.967995     75.706507 -21092.205145
##  [921]   1777.489193  -3017.756978 -15023.699265   6186.996851   1175.112278
##  [926]  -3017.756978 -18924.293493   6882.837251  -9327.858865  17181.054565
##  [931]  -6327.858865  12485.214165 -14310.032005 -10813.003149 -11725.481951
```

```
##  [936]   7280.460336    777.489193  -5801.118575  -8192.799374  -5602.307032
##  [941]   2695.910282 -13993.987831 -14496.958974  -1327.858865   2976.300735
##  [946]    789.373766  -5017.756978 -19801.118575  -5327.858865  22181.054565
##  [951]   -900.524346 -10818.945435 -18830.830009 -10830.830009  -8620.133893
##  [956]  -7123.105036 -12918.351207 -24203.495490 -10204.683947 -15403.495490
##  [961] -18830.830009 -18321.916579 -12222.510807 -19827.858865  -3626.076179
##  [966] -17427.264637   6373.923821 -11228.453094 -18228.453094  -9129.047323
##  [971]  -4514.785835  13081.648793 -17906.466633  12578.677650  16684.025708
##  [976]  -6526.670408 -15315.974292    491.156452   5181.054565   -520.728121
##  [981]     72.735363 -18327.858865   1175.112278  -8298.147432  -6099.335889
##  [986]  -7614.191606  -6626.076179  30777.489193   4274.518049 -17824.887722
##  [991] -14906.466633  18888.779537  -5321.916579 -18818.945435  26186.996851
##  [996]  30888.779537   4882.837251 -18813.003149  -5912.408920  -4204.683947
## [1001] -10321.916579   7795.316053  24596.504510 -20397.553203  -8906.466633
## [1006]  10988.185309   1684.025708  -1626.076179  -4620.133893   7204.823712
## [1011]   3473.329592  -6695.770517   7099.475654 -22298.147432  -6695.770517
## [1016]  17678.083421  -9017.756978 -11813.003149   2876.894964  43801.258340
## [1021]  13491.156452  -7824.887722  -7719.539664  -5123.105036  34397.692968
## [1026] -10900.524346  -2590.422459  -7304.089718  46286.402623   5093.533367
## [1031] -16111.220463   7578.677650   4777.489193  23684.025708  -9228.453094
## [1036]   1578.677650  -9520.728121  57280.460336 -15830.830009     -5.872404
## [1041]   4584.619937 -13029.641551 -14111.220463 -14117.162749  -2719.539664
## [1046] -14099.335889 -10602.307032  -2620.133893   5379.866107 -20864.047323
## [1051]   8602.446797   3777.489193   4578.677650  -8023.699265   9678.083421
## [1056] -12222.510807 -15023.699265   4385.808394  35590.562223 -13204.683947
## [1061] -11496.958974 -10304.089718   3777.489193   4473.329592  -3900.133893
## [1066] -14298.147432  57081.648793  25081.648793  -3725.481951  59497.098739
## [1071]  11286.402623 -16228.453094 -12228.453094   8274.518049  55192.939138
## [1076]  12777.489193 -10421.322350 177578.677650 -19514.785835 -18830.830009
## [1081]  -1900.524346 -19327.858865  -7023.699265 -19502.901261 -21496.958974
## [1086] -19629.047323   4186.996851 -13520.728121  35789.373766 -19730.235780
## [1091] -24003.495490 -16315.974292  -1123.105036  19192.939138   6379.866107
## [1096] -17830.830009  -9228.453094  -1824.887722 -25098.147432  15473.329592
## [1101] -17005.872404  12976.300735   2280.460336  -3924.293493   7379.866107
## [1106]  -5900.524346 -11017.756978 -15801.118575  -5813.003149  -1626.076179
## [1111]   7684.025708 -16620.133893 -11017.756978  -9321.916579   3578.677650
## [1116]  -7123.105036   8175.112278 -21117.162749 -16228.453094  31373.923821
## [1121] -18830.830009   3075.706507 -15830.830009   -725.481951  -4725.481951
## [1126] -10719.539664  18584.619937  -1321.916579 -19502.901261   3572.735363
## [1131] -17930.235780 -22101.712804 -16930.235780  -4807.060862  27988.185309
## [1136]  36684.025708  -8117.162749  65988.185309    497.098739 -19129.047323
## [1141]  -9105.278176   2274.518049  -6824.887722  18684.025708 -13315.974292
## [1146]  -5620.133893   7578.677650   9280.460336  13578.677650 -11029.641551
## [1151]  -1924.293493   2075.706507  -5222.510807   -813.003149   2678.083421
## [1156]  25988.185309  -2023.699265   7678.083421 -15918.351207 -19795.176288
## [1161]   7777.489193 -18427.264637  -3315.974292  15093.533367 -16695.770517
## [1166]  -6719.539664 -13111.220463  -7315.974292  -3725.481951  -9626.076179
## [1171] -17129.047323  36485.214165  19982.243022   8081.648793  27081.648793
## [1176]   5075.706507   5391.750681  -9129.047323  -2023.699265 -11596.364746
## [1181]   2280.460336 -23701.712804 -18029.641551  86186.996851  28900.664111
## [1186]  33689.967995 -12620.133893  -2924.293493 -12192.799374 -14321.916579
## [1191]  35988.185309 -15725.481951
```

Sum of Squares for Error

```
sse <- sum(residuals^2)
sse
```

```
## [1] 448628772037
```

R Squared

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

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

Number of observations

```
n <- nrow(heights_df)
n
```

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

Number of regression parameters

```
p <- 2
p
```

```
## [1] 2
```

Corrected Degrees of Freedom for Model

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

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

Degrees of Freedom for Error

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

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

Corrected Degrees of Freedom Total

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

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

Mean of Squares for Model

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

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

Mean of Squares for Error

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

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

Mean of Squares Total

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

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

F Statistic

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

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

Adjusted R Squared R2

```
#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
```

Calculate the pvalue from the F distribution

```
p_value <- pf(f_score, dfm, dft, lower.tail=F)
p_value
```

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

Assignment 07

Set the working directory to the root of your DSC 520 directory Load the `data/r4ds/heights.csv` to

```
setwd("/Users/dipikasharma/R_Projects/DSC520")

## 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 Model

```
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
)
#predicted_df
```

Compute deviation (i.e. residuals)

```
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 - predicted_df$earn)^2)
ssm
```

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

Residuals

```
residuals <- (heights_df$earn - predicted_df$earn)
residuals
```

```
##    [1]  11333.890941  31140.911188   6698.099079  17810.164851  23192.609973
##    [6] -11154.599443  13604.930235  -9263.321847 -25288.836877   3238.413948
##   [11] -34707.558926   8431.415457 -10992.711766  -4515.869768  -2783.485318
##   [16] -11560.980665  21278.601782  14792.164289  -7544.315052  27127.790515
##   [21] -13665.656619  -2216.202925 -22071.699892  25242.760836   6882.092950
##   [26]   -756.855658  -1969.892050  13145.215569   3993.937070  -2105.566159
##   [31]  12023.211638  35208.801814 -10540.644416    151.524576 -18505.781671
##   [36]  13631.453485   3437.419145   9780.867692 -24556.200216  12333.550089
##   [41]   9396.930903   2491.492079 -17168.532548   4688.275674   2932.577058
##   [46]   8138.885154 -13164.241773  -1608.481840  -5242.623862 -18777.168420
##   [51]   3530.028252 -11951.073208  -6869.465800  10384.864604 -21204.898309
##   [56]  -9314.031509  -7777.883713  -9762.076217  -1336.643042  -7700.985598
##   [61]  -3881.294636  -6920.147978  -4861.751428  -3920.901310 -26567.929098
##   [66]   2615.296998 -17179.884312  -2745.360279  -7127.812156 -10415.049466
##   [71]  -5079.059595 -17745.449056 -10997.799427   2617.000856   9810.940390
##   [76]   5437.653742  -3464.926732  -9324.607282  -9502.945994   6232.131975
##   [81]   5561.946112  -8297.251702  -9938.545623 -24114.190946 -14541.345088
```

```
##   [86]   3368.442767   9980.022048  17315.770961  -8999.217483 -24387.262083
##   [91]  24043.827014  -1646.421969   2922.664553  -1155.523997 -26362.143368
##   [96]  14810.732583   4680.317692   1403.081485  -1876.620163  -5206.744098
##  [101]  14804.424337   8468.180326    186.099706  -4472.590442   9396.903318
##  [106]   1896.157339  -5564.141052  -3811.065444  -7150.198013  45515.404627
##  [111]  -8972.549391 -28654.110131     -3.881121 -27203.209519  35523.420586
##  [116]  -9887.229293   3667.206244  39506.929523  12718.966843  30197.936829
##  [121]   9652.365699 -14847.034004  -6775.360166   1190.096369 -15027.580600
##  [126] -22590.183672 -12186.746557  17210.410045  14393.215537  -2204.407779
##  [131]   2307.689974  16491.322829  14405.740607 -22189.889273  29726.679828
##  [136]   6292.518431 -19392.328343 -19883.861488   3897.640181  -6358.926987
##  [141]   1125.332677  66350.743919  -7099.263831   1490.648589   3755.459040
##  [146]  -6800.007713  -2428.836828    151.254975   4343.760774 -17840.445775
##  [151]  15425.619379 -10202.574919   1957.452621  -5192.159347 -21850.923020
##  [156]   6067.345323  -5619.371308   3693.697496  -4268.954084   9259.740283
##  [161] -18810.082613  27411.694962 -10734.910921  -6194.017146 -14046.936173
##  [166] -14163.282694  -7792.991188   5147.208452   8732.104620  20963.152390
##  [171]   6707.461748 -12259.747084  -5616.919424   2806.472821  80814.141624
##  [176]   6965.225880  -6625.987683  12968.719173  39730.969576  11068.086031
##  [181]  -7417.904352  13030.230217    603.047386  -2025.530157 -23059.184763
##  [186]  -8301.762623  -9807.038184  20277.060148  -1669.370514  -5266.874689
##  [191]  -9401.148295   9832.703834   5416.167417   4466.091263  11774.641361
##  [196]   5633.487632  43529.298561  18901.180355   5032.119736   9526.455320
##  [201]   3613.667778  -5261.208831 126488.502286     53.850617   4782.832136
##  [206]   3653.152007   6671.499215  12886.994823  13792.561570  -5555.474955
##  [211] -10798.003560   5198.754987  14171.088005 -13592.345671  15644.822158
##  [216]  -6040.190791   6537.363821   5223.379132   8806.591687  -5174.730159
##  [221]  15933.657766 -12087.103976  -4671.912521  25083.497154  22249.041553
##  [226]  29248.418158  20023.996496 -25507.061331  -6126.668811   5497.557098
##  [231]  -2620.666663 -10993.418736  42440.129722  23695.271892  -3788.674776
##  [236]  -6530.054264   3816.055352 -14572.885984  17103.878588  -9621.339964
##  [241]   2025.019128 -13022.288326 -15606.846547  -1639.766246  -2194.087351
##  [246]  -2555.062694   1235.312859  -4843.068874   7653.393988 -11252.805968
##  [251] -20649.481287  -5954.503821 -10543.962114   6496.888274    -92.507946
##  [256]  -4711.146726  11179.144701  -1992.557102  -8762.663651  -5958.159879
##  [261] -17383.062470  -8471.587669  14994.212961  -9338.465895  -2473.083447
##  [266] -14808.150356  33108.456012  10421.409940  -9615.078888    743.012989
##  [271]   -611.347947   3016.116315 -10064.794753  -8566.711681  -1646.581418
##  [276]   2839.321495  19023.247647  -8432.895485   3667.208536   3707.559217
##  [281]  -3302.207564 -15398.405033  -2471.365770   8703.696682 -14951.327586
##  [286] -14908.899720 -16976.358141 -12064.577206    443.367133    164.312850
##  [291]  -3059.550404  -9670.758759  14162.596931   5338.129297  -6988.260841
##  [296]   7789.983357  -4470.182412  -2663.626562  -2511.548097  -9493.149022
##  [301] -16004.636575   2261.431152   1063.282225  11934.482241   8387.103912
##  [306]   3821.849964  -6244.433513  48194.467565   2046.069389 -27117.745856
##  [311]   3345.721926  -2738.135109   3333.736591   1197.463545 -15897.738249
##  [316] -10314.748262 -21187.218229  -1434.419379 -12486.262006 -18088.753900
##  [321]  -7664.905252 -12700.375996  12180.583893    503.310543  -3772.437683
##  [326]   6564.747383 -25196.310916 -13594.036217  12860.605042   4058.252634
##  [331]  -2896.986471 -10454.954118   7970.978764    696.508697   3668.609214
##  [336] -34210.400221 -12993.945709 -17269.237401  -8588.439567 136575.449371
##  [341] -39423.460319 -14042.053807   9763.058215  -6949.548796   3759.759459
##  [346]    -93.721424   4923.633903  14988.554140  21747.419002  54620.816888
##  [351]   8856.888870   9873.310125  -5957.725937  -5455.106202   2613.315836
```

```
##  [356]  -5237.059139 106601.160960   5667.015766 -14415.995073 -21309.261105
##  [361] -24443.964818    482.706789  -1591.027833    450.963593    849.934309
##  [366]  -4163.111779  -5009.601461   3443.689542 -13632.554068  -9282.917128
##  [371] -12672.202119  11734.858672   8502.735765  13029.107387  -5685.706765
##  [376] -16390.590672  68936.414159  17285.961093   -519.269279  -1861.508060
##  [381]  -7963.645544   -235.502113 -13541.490778    568.332569  -7267.573565
##  [386]  -1454.859944  19408.670257   4143.754872   9041.794515  25841.395537
##  [391] -17929.566866 -12111.731641  -2438.042180 -10760.913235  13509.489644
##  [396]  -8583.633968 -10860.305787  -9971.173541  -4874.725895 -34741.037174
##  [401]  -5399.532798   3273.769933   9937.445751   4536.984867 -18609.364954
##  [406]   1267.607399  -6910.024497  -9741.969417  -8449.536194 -11687.973666
##  [411]   -674.324404 -15334.799323   9414.638478  12853.264929   8237.027140
##  [416] -10076.900032 -15253.382123  15853.308519  -7394.082288 -17649.210850
##  [421]   3181.690693  50619.201983  10792.689708  -7934.373669 -15271.629244
##  [426]   4093.623924   7446.634207   -908.368889  26630.187127  14348.031542
##  [431]  16636.072688  -5051.628098    628.006405  -2120.369073 -10345.569542
##  [436]  -2082.690742  -5497.550541 -11662.783902 -12761.984186  19774.192728
##  [441]   8734.861368  11008.880887  -8994.924004  -2583.213952 -14094.896455
##  [446]  13026.144602   2325.951216  -3491.057050   7586.773890 -15485.028710
##  [451]  18582.326831   2320.704414  12198.374323 -24784.966722  -8572.881158
##  [456]  11180.777239  12893.572416 -17350.967578    925.539695   5604.411399
##  [461]   6253.960101 -20849.311359  23310.695214  -4146.893302  24171.784127
##  [466] -10262.079289   1821.616701 -15927.640642    120.334978 -18389.983584
##  [471]   1489.913932  -2163.751120   9288.176903   8406.240711   5219.183401
##  [476]  19473.047363  -4689.215871  13160.820475  -3711.085223   3183.997902
##  [481]    373.687994   4389.113373  -3216.065580 -12883.606617  -3881.546897
##  [486]  -8677.656511  25658.293013   -915.759846   3586.017549  -6157.705736
##  [491] -13812.447546  -5948.802135   4897.853649  -1281.305124  -8617.179894
##  [496] -15599.275922 -12691.848864  10361.196570  -9367.201806    170.726942
##  [501]   3990.934835  16955.038845      5.903462 -16384.463062  -7514.587734
##  [506]   8908.174379  16976.440355  -4448.931515   2650.757751   4576.197797
##  [511]   6369.201774  -1113.307518   8986.249032    355.705815  -4367.056738
##  [516]  -5107.485842  36945.203820   7474.526042 -16621.592718  -9780.217216
##  [521] -13324.355484  20310.118284   5412.115461   5615.379814   -315.647346
##  [526]  -5838.952401  -8477.252286 -17976.633733   2364.715466  -5028.869351
##  [531]   1364.529467  -8551.650990   4723.036084  -1144.823488  -9027.816495
##  [536]  -5937.571415  20869.863749   5739.365220 -13695.960053  -3002.637851
##  [541]  -5981.725883 -23196.557538   4741.178003   5696.968136  -1814.970902
##  [546]   8008.717215  27352.078240  -6549.300974  -2047.325463  11355.190901
##  [551] -16008.858646  -2211.898810  10757.141175  -3625.570515  10291.250507
##  [556]  -3214.986231   4629.753605   7904.177075 -11030.182610    936.568124
##  [561]   8875.015538   8321.704123  -5865.534343  14943.000850  -5712.148871
##  [566]   -486.111726  76662.610295  57679.850359  -6631.956204 -12658.017651
##  [571]   9298.816055 -13632.057010  44682.785022 -19995.897715   9007.368397
##  [576] -19666.410028  18568.570897 -13867.265555 -11559.978331   -804.991510
##  [581] -10761.429292 -19300.726649  -3264.701571 -31581.162399   8781.526421
##  [586]   1924.953815   1270.255333  12448.324879  -9446.160894 -15412.031456
##  [591]  14388.990580   3063.035646 -12561.280286   3021.944609  18707.281413
##  [596]  -7935.138958   5248.375888  13464.684073   9309.140800  16123.854186
##  [601]   5909.076608    398.777426   1243.260872   9223.511015 -11105.422084
##  [606]  -8103.012928   1772.519837  36231.714573 -23483.774543  -9918.689175
##  [611]  -1028.353841 -28589.439829 -20436.099033  -4545.129299   4261.385539
##  [616]  57587.096491  27857.149469    875.776825 -12882.876680  -3909.511698
##  [621]   -448.872535 -10294.287857 -16523.717832 -29283.272524   5235.236464
```

```
##  [626]   -979.105366  -4622.007015    953.574914   6970.699612  19691.624891
##  [631]  10277.144164  -7263.036780 -11931.705356  -3335.945094  -2354.164406
##  [636]   5154.319662   1885.158868   7966.349873    -76.093656 -10537.639841
##  [641]  -6982.243249  -4805.381433 -19485.564254   1063.585637   1579.304470
##  [646] -11878.330103 -24979.676190   2522.284624   2636.922462   2473.556960
##  [651]  13318.393327  -2141.939214  -7114.395885  -5213.653468   6458.226324
##  [656]   6475.028617  -8674.866226  -8625.202119    400.044275  -1055.649190
##  [661]   5132.098761   1682.534272   2074.275219  51317.640034  12900.839059
##  [666]   3162.497415  -2105.319522  -2286.019298 -13662.129240  12987.153116
##  [671]   -615.186365  -2469.158301 -14840.343243  -1378.566252  -9490.708418
##  [676] -16954.408450   2233.989737   5836.043141 -17837.800409 -11510.787842
##  [681]   3435.316627  -3132.805245  -4034.979170 -15994.316125 -12982.931949
##  [686]  -8048.351110 -11772.395082   2448.630097 -11012.985548   3811.184939
##  [691]  19792.290830  -2574.144955 -30051.311255   2318.103413   7997.983676
##  [696]   6611.794858    548.617457   3042.529945  -6428.676476  -8336.277749
##  [701] -11574.849995  -7343.182823 -15561.282411  -7024.221298   2850.124892
##  [706]   4577.862824  -3106.933959  12451.642348   4235.677418  -8522.014565
##  [711]  -2643.046121  -2423.763153  -7659.812251 -14302.875236   2740.979396
##  [716]  -5943.620254 -14646.003184  52366.044716 -11121.098914  16843.993822
##  [721]   6452.688792   1079.576442 -16355.439089  29958.968107   -780.601759
##  [726] -17644.009357   9681.700580  -1366.154988 -16217.546426   3480.975521
##  [731]   6139.794459 -13139.161427  -5868.096744 -15768.816300  -5084.216359
##  [736]  -2334.920670  -5473.449146   7384.053307   4512.301872  -4029.551371
##  [741]   9056.423841   -482.670884   -709.064246 -22823.203450  -9685.378982
##  [746]    150.553305  -5284.586315 -11830.043851  -6856.507963    607.154861
##  [751] 100509.491540   1301.124226  -1359.776403   5406.803084  -1418.730682
##  [756]   7220.076509 -13789.782723  -8220.765786   4547.988080 -13396.217155
##  [761] -12035.374227  -6939.876394  -9101.647110  -5510.161632  -6678.615221
##  [766]  -4222.743117  -8733.278328  -8386.713108  12695.185804   2590.275487
##  [771]  16537.163717   4932.091600  -7787.239779   4001.470796  -2356.581335
##  [776]   2755.736774 -13988.596481  -4900.181261  -6493.719128   6007.420976
##  [781]   5143.116701  12105.578646  -5275.696335   -361.564796  13838.366826
##  [786] -29772.245151   6745.494075 -11679.121642  -1974.323130   6803.773577
##  [791]  -7727.221446  67030.580256  24856.998529 -20188.582625 -12133.518688
##  [796] -11151.467622 -10164.401378   7199.219675  -7456.456555 -15665.634927
##  [801] -14327.244375 -18609.690245  -7820.843515  -3658.666664 -12613.380918
##  [806]   3565.254329 -12628.317891 -15685.103119    738.933629 -10507.238418
##  [811] -14220.193195  -8832.546610 -19789.513857  -6926.770300 -13819.397669
##  [816]  -4200.410263   8231.580515  -3342.532629  27986.190618  16425.195771
##  [821]  38741.858188  11208.494004   6281.129461  -5108.889537  13402.458717
##  [826]  -4078.908544   2810.332751  78694.548550  -4250.905839 -24561.807430
##  [831] -13044.866539  22268.774675 -11928.078904  74328.889755  -1181.129266
##  [836]    332.422444 -13152.536590 -16486.938021   4047.764100  -3529.541180
##  [841]  -8228.357717   -830.086290  -9959.471780  -4430.064250  15346.307677
##  [846]  -4342.602246 -12038.004036  -7408.162603  -9392.078435  -6028.073252
##  [851] -14246.802466   3423.987859 -15353.395804  -5072.675137  -4480.573081
##  [856]  33791.805537  -5618.480247 -19969.118144 -20252.480710 -15424.467137
##  [861]   -483.030017 -10951.584239   2176.264300 -14101.610366   -862.951866
##  [866]  12770.323232   1265.047347 -13054.899831  -5387.587119  28088.282364
##  [871]  -1189.420315  22857.086364  11327.501656 -14664.757355   6666.208380
##  [876]  -7248.377675 -12278.938308  -2866.938211 -14922.244732  -8106.497771
##  [881] -22056.264792   3618.855161 -11934.120911   3279.026518  -2037.629776
##  [886] -13846.924291  -9438.419201  16972.817873  -8406.955175   2432.131251
##  [891] -13629.433726   1601.371259 -21127.804433 -17051.667427  20322.758562
```

```
##  [896]  34138.174344  -2832.836212  31066.823272    705.141866   8729.096396
##  [901]  -4535.726626   2668.585786 -12375.167703  -7349.403865  57462.397811
##  [906]  -5136.384144    271.569469 -17022.647308 -18866.892444 -11800.639217
##  [911]   6209.214776   4238.876412  -2613.722772  18363.238196    861.117943
##  [916]  -8559.685360  -7735.020616  15033.343843  -7104.952074 -18546.277292
##  [921]    -63.116791 -11483.593349 -12155.506251   4918.034295  -6159.234883
##  [926]   5654.222419 -20757.669553  -7003.488486  -4728.407121  14428.202674
##  [931]  -1737.334485   9888.944719  -7035.799541  -9115.005148    986.823891
##  [936]   7459.220612  -3026.459665 -24560.603693  -5799.517470  -3367.685477
##  [941]  -3794.396376  -2977.958577 -18859.214629    135.672304    241.401965
##  [946]   4978.262778  -7864.033428  -7275.420498  -9996.597568   7777.219631
##  [951] -11486.661949  -7347.793938  -9195.115007   -121.923195 -11237.935185
##  [956]   2392.130877  -3349.787527   4445.090431  -7696.376411 -11130.375743
##  [961] -19113.594511 -12077.860738  -8454.682569 -10092.946296 -14393.113111
##  [966] -27844.193463  -6878.039480  -8682.948916 -19261.677398 -10541.480469
##  [971]    250.025650   -799.798297 -21770.041875  25641.423897  -2525.770185
##  [976] -13248.429962 -10544.958451   1423.253063 -15124.964139  -8275.057247
##  [981]  -2202.414986  -9111.805064  -3745.760103 -14166.141448 -21858.959274
##  [986]   4671.197553  -8150.748695  30393.979577   2045.918123  -6354.693501
##  [991] -30810.329845  15410.782012 -10789.511531  -5554.775891  13816.851878
##  [996]  21583.950499   2670.886278 -15855.293195   -920.921637 -13503.184687
## [1001]  -1662.971804   7282.692199  20502.517686  -2456.476552  -7152.268690
## [1006]   7774.402661   9331.169594 -10138.056070   4728.373938  -5007.886551
## [1011]  -8895.629546 -26665.840883   2516.240629 -16260.129959  11000.116452
## [1016]   3890.546056  -5746.296969  -4503.442467  12917.535367  54940.967967
## [1021]   6570.110495   2067.436067   -738.112397 -18995.746739  24261.245277
## [1026] -13297.010592 -19051.654661  -6482.058576  25967.585812  -4379.835419
## [1031]  -7894.615548  -5996.148025  -3190.854673   4034.020235  -6051.557244
## [1036]  -2585.651904   3697.340539  61377.924006 -17249.642945  13005.498535
## [1041]   2325.014440 -11029.547216  -9487.350772 -11842.764346   7169.504534
## [1046] -18358.650978  -9592.617714   1451.001048   8930.239432 -19318.440755
## [1051]  -2273.326848  11066.882495   -571.834111    631.629877  -2722.798042
## [1056]  -9407.101615 -27958.925684 -12921.969606  28145.193911   4714.513457
## [1061] -10532.795294 -12512.930794   4777.933393  -4698.871594  13389.098903
## [1066]  -5677.298056  40444.456901  30656.124037 -12980.533856  46486.522597
## [1071]  21992.252606 -28406.689708  -1014.094073   6344.801602  45110.102938
## [1076]  10872.200075  -6607.797748 158722.627132 -22272.090112 -19487.669033
## [1081] -12441.762262 -24486.654715 -22801.179121 -12368.336203 -26219.045218
## [1086]  -6034.560824  -4208.679078  -5571.283458  25678.237225 -19949.904572
## [1091]  12095.050119 -11060.836284   8690.132860  15177.217231  -4267.328561
## [1096] -20942.719653  -9782.022130   -319.648276   3323.628121   2033.879999
## [1101] -13021.353079   4177.791607  -2114.836789   9780.279905   8106.762469
## [1106] -27433.276590   6408.342884  -9648.464849 -15152.999300  -1304.981306
## [1111]   4408.819744  -8393.840827 -16542.622542 -11567.975326  -1545.995460
## [1116]   6717.608144   4078.794433 -18122.608844 -18294.604732  34222.884681
## [1121] -10679.971870 -12859.289946 -17754.574627   6243.239624  -4415.700325
## [1126]  -4498.646580  13867.231463  -4299.307426  -8616.946779  -4208.005985
## [1131] -13480.150354 -24065.681551 -18247.000200 -14252.691452  18070.824369
## [1136]  45198.013569  -2174.317660  50953.548369   2264.799898 -20152.599796
## [1141]  -2639.002641  18553.040999 -15540.424265    922.120282  -5041.404527
## [1146]  -7155.720934  11179.175994   3443.877073  12731.567289 -11545.971463
## [1151]   -315.614577  11517.054394  -7714.229289   1538.785180  11926.463748
## [1156]  22232.784675   7970.362142   5450.182840  -1564.313071 -19397.412070
## [1161]   8778.920060  -8766.736572   -744.479954  10916.458959 -21744.030134
```

```
## [1166]  -8489.030718   3220.526685  -2300.959284  -5574.409401 -10563.622002
## [1171] -21519.375066  32934.801399   2386.691896  11788.623366  21219.122597
## [1176]   8339.849794 -12599.332936 -11438.446284  -5946.458318  -2396.118951
## [1181]   2225.724639   4147.823467 -24358.716589  79291.863728   7260.430309
## [1186]  13197.058759 -15851.346398  -5282.484795 -22783.845322  -4948.753772
## [1191]  32021.719023 -17157.030785
```

Sum of Squares for Error

```
sse <- sum(residuals^2)
sse
```

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

R Squared

```
r_squared <- (ssm/sst)
r_squared
```

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

Number of observations

```
n <- nrow(heights_df)
n
```

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

Number of regression paramaters

```
p <- 8
p
```

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

Corrected Degrees of Freedom for Model

```
dfm <- (p-1)
dfm
```

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

Degrees of Freedom for Error

```
dfe <- (n-p)
dfe
```

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

Corrected Degrees of Freedom Total

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

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

Mean of Squares for Model

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

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

Mean of Squares for Error

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

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

Mean of Squares Total

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

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

F Statistic

```r
f_score <- (msm/mse)
f_score
```

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

Adjusted R Squared R2

```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
```

Housing Data

```
## Set the working directory to the root of your DSC 520 directory
setwd("/Users/dipikasharma/R_Projects/DSC520")
library(readxl)
housing_df <- read_excel("data/week-7-housing.xlsx")
housing_df
```

```
## # A tibble: 12,865 x 24
##    `Sale Date`         `Sale Price` sale_reason sale_instrument sale_warning
##    <dttm>                     <dbl>       <dbl>           <dbl> <chr>
##  1 2006-01-03 00:00:00       698000           1               3 <NA>
##  2 2006-01-03 00:00:00       649990           1               3 <NA>
##  3 2006-01-03 00:00:00       572500           1               3 <NA>
##  4 2006-01-03 00:00:00       420000           1               3 <NA>
##  5 2006-01-03 00:00:00       369900           1               3 15
##  6 2006-01-03 00:00:00       184667           1              15 18 51
##  7 2006-01-04 00:00:00      1050000           1               3 <NA>
##  8 2006-01-04 00:00:00       875000           1               3 <NA>
##  9 2006-01-04 00:00:00       660000           1               3 <NA>
## 10 2006-01-04 00:00:00       650000           1               3 <NA>
## # ... with 12,855 more rows, and 19 more variables: sitetype <chr>,
## #   addr_full <chr>, zip5 <dbl>, 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>
```

```
#unique(housing_df$ctyname)
```

3a. 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.

```
housing_df$ctyname <- ifelse(is.na(housing_df$ctyname), housing_df$postalctyn, housing_df$ctyname)
#housing_df
#unique(housing_df$ctyname)

housing_df <- subset(housing_df, select = -sale_warning)
housing_df
```

```
## # A tibble: 12,865 x 23
##    `Sale Date`         `Sale Price` sale_reason sale_instrument sitetype
##    <dttm>                     <dbl>       <dbl>           <dbl> <chr>
##  1 2006-01-03 00:00:00       698000           1               3 R1
##  2 2006-01-03 00:00:00       649990           1               3 R1
##  3 2006-01-03 00:00:00       572500           1               3 R1
##  4 2006-01-03 00:00:00       420000           1               3 R1
##  5 2006-01-03 00:00:00       369900           1               3 R1
```

```
##  6 2006-01-03 00:00:00       184667           1              15 R1
##  7 2006-01-04 00:00:00      1050000           1               3 R1
##  8 2006-01-04 00:00:00       875000           1               3 R1
##  9 2006-01-04 00:00:00       660000           1               3 R1
## 10 2006-01-04 00:00:00       650000           1               3 R1
## # ... with 12,855 more rows, and 18 more variables: addr_full <chr>,
## #   zip5 <dbl>, 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>
```

Complete the following:

Explain any transformations or modifications you made to the dataset

Answer: After reading the data I found that sale_warning and ctyname has null values. sale_warning has only 18% of the rows with actual data in housing dataset. rest of the 82% of rows are showing NULL. Adding zero instead of null value will not solve the problem and can lead to wrong prediction. Better way is to remove this column from dataset in order to avoid miscalculation. ctyname has 50% of rows with actual values and rest 50% with NULL values, but we have another column postalctyn, we can either remove this column or can replace NULL values in ctyname column with postalctyn value. I thought of removing ctyname first but then realized that we have different values in ctyname Column and have same value for all address in postalctyn. so i think it is better idea to just relace null values in ctyname with postalctyn.

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.

```
SP_FL_lm <- lm(`Sale Price`~sq_ft_lot, data = housing_df)
SP_FL_lm
```

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

```
SP_other_lm <- lm(`Sale Price`~building_grade+square_feet_total_living , data = housing_df)
SP_other_lm
```

```
##
## Call:
## lm(formula = `Sale Price` ~ building_grade + square_feet_total_living,
##     data = housing_df)
##
```

```
## Coefficients:
##              (Intercept)           building_grade  square_feet_total_living
##                 -79560.6                 43675.2                    149.8
```

Explain the basis for your additional predictor selections.

Answer: I am using the building_grade and square_feet_total_living as the predictor variable because i think that the change in building grade or square_feet_total_living will affect the sale price of the house.

iii Execute a summary() function on two variables defined in the previous step to compare the model results.

```
summary(SP_FL_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(SP_other_lm)
```

```
##
## Call:
## lm(formula = `Sale Price` ~ building_grade + square_feet_total_living,
##     data = housing_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1741217  -116774   -43474    38722  3856512
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -79560.628  28092.600  -2.832  0.00463 **
## building_grade            43675.220   4341.704  10.059  < 2e-16 ***
## square_feet_total_living    149.791      4.793  31.254  < 2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 358800 on 12862 degrees of freedom
## Multiple R-squared:  0.2128, Adjusted R-squared:  0.2127
## F-statistic:  1739 on 2 and 12862 DF,  p-value: < 2.2e-16
```

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?

Answer: R2 is used to determine to what extent the variance of one variable explain the variance of the other variable. Adjusted R2 is the modified version of R2, it is adjusted for number of predictor variable. if the new term improves the model from what it is expected then adjusted R2 increases and decreases when predictor variable improves the model less then what it is expected.

By compary the R2 and adjusted R2 of both the variable I found that the relationship between model and multiple independent variables (building grade and square_feet_total_living) is better than the relationship between model and independent variable square ft lot. Looking at the R2 and adjusted R2 of sale price and predictor variables building grade and square_feet_total_living we can understand that the increase in these predictor variable will show the change in dependent variable 'sale price'

yes, by adding the predictor variable building grade and square_feet_total_living, we found 20% variation in 'sale price'

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)
lmbeta_SP <- lm.beta(SP_FL_lm)
lmbeta_sp_other <- lm.beta(SP_other_lm)
lmbeta_SP
```

```
##
## Call:
## lm(formula = `Sale Price` ~ sq_ft_lot, data = housing_df)
##
## Standardized Coefficients::
## (Intercept)   sq_ft_lot
##   0.0000000   0.1198122
```

```
lmbeta_sp_other
```

```
##
## Call:
## lm(formula = `Sale Price` ~ building_grade + square_feet_total_living,
##     data = housing_df)
##
## Standardized Coefficients::
##              (Intercept)           building_grade square_feet_total_living
##                0.0000000               0.1180089               0.3666496
```

Answer: Standard coefficient is uded to find out which of the independent variable in multiple regression model have greater effect on the dependent variables. By looking at standard coeffcient of all variable we can see that square_feet_total_living has most effect on dependent variable 'Sale Price' compare to the others independent variables.

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

```
summary(SP_FL_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(SP_other_lm)
```

```
##
## Call:
## lm(formula = `Sale Price` ~ building_grade + square_feet_total_living,
##     data = housing_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1741217  -116774   -43474    38722  3856512
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -79560.628  28092.600  -2.832  0.00463 **
## building_grade            43675.220   4341.704  10.059  < 2e-16 ***
## square_feet_total_living    149.791      4.793  31.254  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 358800 on 12862 degrees of freedom
## Multiple R-squared:  0.2128, Adjusted R-squared:  0.2127
## F-statistic:  1739 on 2 and 12862 DF,  p-value: < 2.2e-16
```

```
confint(SP_other_lm, 'sq_ft_lot', level=0.95)
```

```
##           2.5 % 97.5 %
## sq_ft_lot    NA     NA
```

```
confint(SP_other_lm, level=0.95)
```

```
##                                2.5 %      97.5 %
## (Intercept)              -134626.2948 -24494.9615
## building_grade             35164.8358  52185.6048
## square_feet_total_living     140.3971    159.1857
```

Answer: Looking at the parameter values we can say that the confidence interval of building grade (35164.8 to 52185.6) signifies the range in which the true population parameter lies at a 95% level of confidence And the confidence interval of square_feet_total_living (140 to 159) signifies the range in which the true population parameter lies at a 95% level of confidence.

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.

```
summary(SP_FL_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(SP_other_lm)
```

```
##
## Call:
## lm(formula = `Sale Price` ~ building_grade + square_feet_total_living,
##     data = housing_df)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1741217  -116774   -43474    38722  3856512
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -79560.628  28092.600  -2.832  0.00463 **
## building_grade            43675.220   4341.704  10.059  < 2e-16 ***
## square_feet_total_living    149.791      4.793  31.254  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 358800 on 12862 degrees of freedom
## Multiple R-squared:  0.2128, Adjusted R-squared:  0.2127
## F-statistic:  1739 on 2 and 12862 DF,  p-value: < 2.2e-16
```

```
library(car)
```

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

```
compareCoefs(SP_FL_lm, SP_other_lm)
```

```
## Calls:
## 1: lm(formula = `Sale Price` ~ sq_ft_lot, data = housing_df)
## 2: lm(formula = `Sale Price` ~ building_grade + square_feet_total_living,
##   data = housing_df)
##
##                           Model 1 Model 2
## (Intercept)                641821  -79561
## SE                           3800   28093
##
## sq_ft_lot                  0.8510
## SE                         0.0622
##
## building_grade                      43675
## SE                                   4342
##
## square_feet_total_living           149.79
## SE                                   4.79
##
```

Answer: When I compared the R2 and Adjusted R2 of both model simple regression model and multiple regression model where I am using two independent variables building grades and square feet living total, I found that the values of multiple regression model is higher and it is expected to always choose the model which has higher Adjusted R2. Adjusted R2 increases only if new term improves the model more than would be expected by chance.

```
anova(SP_FL_lm, SP_other_lm)
```

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

```
## Model 1: `Sale Price` ~ sq_ft_lot
## Model 2: `Sale Price` ~ building_grade + square_feet_total_living
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
## 1  12863 2.0734e+15
## 2  12862 1.6558e+15  1 4.1753e+14 3243.3 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As we can see from the p-value, both models are slightly different. but since the RSS df is less in model 2 so we can say model 2 is better.

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.

```
HousingOrg <-
  lm(`Sale Price`~square_feet_total_living+bath_3qtr_count+bath_full_count+bath_half_count+bedrooms+bui
     data=housing_df)
summary(HousingOrg)
```

```
##
## Call:
## lm(formula = `Sale Price` ~ square_feet_total_living + bath_3qtr_count +
##     bath_full_count + bath_half_count + bedrooms + building_grade +
##     lat + lon + present_use + sale_instrument + sale_reason +
##     sq_ft_lot + year_built + year_renovated + zip5, data = housing_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2261467  -120306   -43998    41921  3690837
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -3.757e+08  1.997e+08  -1.881   0.0599 .
## square_feet_total_living  1.477e+02  6.530e+00  22.624  < 2e-16 ***
## bath_3qtr_count          -1.590e+04  6.948e+03  -2.288   0.0222 *
## bath_full_count          -1.088e+03  7.597e+03  -0.143   0.8862
## bath_half_count          -1.932e+03  7.161e+03  -0.270   0.7873
## bedrooms                 -1.042e+04  4.909e+03  -2.122   0.0338 *
## building_grade            2.755e+04  4.499e+03   6.124 9.37e-10 ***
## lat                      -2.941e+04  1.397e+05  -0.210   0.8333
## lon                      -3.376e+05  7.570e+04  -4.459 8.30e-06 ***
## present_use              -7.498e+02  1.049e+02  -7.150 9.15e-13 ***
## sale_instrument           1.311e+02  1.038e+03   0.126   0.8995
## sale_reason              -1.164e+04  1.281e+03  -9.087  < 2e-16 ***
## sq_ft_lot                 3.933e-01  6.121e-02   6.426 1.35e-10 ***
## year_built                3.116e+03  2.677e+02  11.638  < 2e-16 ***
## year_renovated            8.101e+01  1.433e+01   5.654 1.60e-08 ***
## zip5                      3.364e+03  1.998e+03   1.683   0.0923 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 353700 on 12849 degrees of freedom
## Multiple R-squared:  0.236,  Adjusted R-squared:  0.2352
## F-statistic: 264.7 on 15 and 12849 DF,  p-value: < 2.2e-16
```

```
library(car)
outlierTest(HousingOrg)
```

```
##       rstudent unadjusted p-value Bonferroni p
## 11992 10.50924         9.9591e-26   1.2812e-21
## 6430  10.44866         1.8795e-25   2.4180e-21
## 6438  10.41399         2.6990e-25   3.4723e-21
## 6437  10.40667         2.9127e-25   3.7472e-21
## 6431  10.30124         8.6846e-25   1.1173e-20
## 6436  10.26956         1.2034e-24   1.5482e-20
## 6441  10.25805         1.3546e-24   1.7427e-20
## 6432  10.22762         1.8507e-24   2.3809e-20
## 6442  10.19008         2.7164e-24   3.4946e-20
## 6433  10.16111         3.6491e-24   4.6945e-20
```

```
outlierTest(SP_FL_lm)
```

```
##      rstudent unadjusted p-value Bonferroni p
## 6438 9.334760         1.1763e-20   1.5134e-16
## 6437 9.334494         1.1793e-20   1.5171e-16
## 6441 9.334316         1.1813e-20   1.5197e-16
## 6433 9.334031         1.1844e-20   1.5237e-16
## 6434 9.333823         1.1867e-20   1.5267e-16
## 6430 9.333677         1.1884e-20   1.5288e-16
## 6442 9.332473         1.2018e-20   1.5462e-16
## 6439 9.331469         1.2132e-20   1.5608e-16
## 6431 9.331388         1.2141e-20   1.5620e-16
## 6429 9.329466         1.2362e-20   1.5904e-16
```

```
outlierTest(SP_other_lm)
```

```
##       rstudent unadjusted p-value Bonferroni p
## 4649  10.80053         4.4713e-27   5.7523e-23
## 11992 10.61238         3.3497e-26   4.3094e-22
## 6438  10.48261         1.3170e-25   1.6943e-21
## 6430  10.47838         1.3766e-25   1.7711e-21
## 6437  10.43612         2.1420e-25   2.7557e-21
## 6431  10.35167         5.1557e-25   6.6329e-21
## 6436  10.32636         6.6999e-25   8.6194e-21
## 6441  10.26310         1.2858e-24   1.6542e-20
## 6432  10.24202         1.5963e-24   2.0537e-20
## 6442  10.19989         2.4569e-24   3.1608e-20
```

The original data had line 6430 and with the adjusted model we have row 4649 is listed. so the updated data frames with out the outlier rows look like below:

```
HousingOrg_out <- housing_df[-c(11992,6430,6438,6437,6431,6436,6441,6432,6442,6433,4649),]
str(HousingOrg_out)
```

```
## tibble [12,854 x 23] (S3: tbl_df/tbl/data.frame)
##  $ Sale Date               : POSIXct[1:12854], format: "2006-01-03" "2006-01-03" ...
##  $ Sale Price              : num [1:12854] 698000 649990 572500 420000 369900 ...
##  $ sale_reason             : num [1:12854] 1 1 1 1 1 1 1 1 1 1 ...
##  $ sale_instrument         : num [1:12854] 3 3 3 3 3 15 3 3 3 3 ...
##  $ sitetype                : chr [1:12854] "R1" "R1" "R1" "R1" ...
##  $ addr_full               : chr [1:12854] "17021 NE 113TH CT" "11927 178TH PL NE" "13315 174TH AVE N
##  $ zip5                    : num [1:12854] 98052 98052 98052 98052 98052 ...
##  $ ctyname                 : chr [1:12854] "REDMOND" "REDMOND" "REDMOND" "REDMOND" ...
##  $ postalctyn              : chr [1:12854] "REDMOND" "REDMOND" "REDMOND" "REDMOND" ...
##  $ lon                     : num [1:12854] -122 -122 -122 -122 -122 ...
##  $ lat                     : num [1:12854] 47.7 47.7 47.7 47.6 47.7 ...
##  $ building_grade          : num [1:12854] 9 9 8 8 7 7 10 10 9 8 ...
##  $ square_feet_total_living: num [1:12854] 2810 2880 2770 1620 1440 4160 3960 3720 4160 2760 ...
##  $ bedrooms                : num [1:12854] 4 4 4 3 3 4 5 4 4 4 ...
##  $ bath_full_count         : num [1:12854] 2 2 1 1 1 2 3 2 2 1 ...
##  $ bath_half_count         : num [1:12854] 1 0 1 0 0 1 0 1 1 0 ...
##  $ bath_3qtr_count         : num [1:12854] 0 1 1 1 1 1 1 0 1 1 ...
##  $ year_built              : num [1:12854] 2003 2006 1987 1968 1980 ...
##  $ year_renovated          : num [1:12854] 0 0 0 0 0 0 0 0 0 0 ...
##  $ current_zoning          : chr [1:12854] "R4" "R4" "R6" "R4" ...
##  $ sq_ft_lot               : num [1:12854] 6635 5570 8444 9600 7526 ...
##  $ prop_type               : chr [1:12854] "R" "R" "R" "R" ...
##  $ present_use             : num [1:12854] 2 2 2 2 2 2 2 2 2 2 ...
```

Creating above 2 models with Housing data set without the outliers

```
model3 <- lm(`Sale Price`~sq_ft_lot, data = HousingOrg_out)
summary(model3)
```

```
##
## Call:
## lm(formula = `Sale Price` ~ sq_ft_lot, data = HousingOrg_out)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1842138  -193138   -61116    93160  3735963
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.411e+05  3.677e+03  174.34   <2e-16 ***
## sq_ft_lot   7.448e-01  6.140e-02   12.13   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 387500 on 12852 degrees of freedom
## Multiple R-squared:  0.01132,    Adjusted R-squared:  0.01125
## F-statistic: 147.2 on 1 and 12852 DF,  p-value: < 2.2e-16
```

```
model4 <- lm(`Sale Price`~building_grade+square_feet_total_living, data = HousingOrg_out)
summary(model4)
```

```
##
## Call:
## lm(formula = `Sale Price` ~ building_grade + square_feet_total_living,
##     data = HousingOrg_out)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1731248  -113245   -40662    41244  3643179
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -1.013e+05  2.680e+04   -3.78 0.000157 ***
## building_grade            4.686e+04  4.142e+03   11.31  < 2e-16 ***
## square_feet_total_living  1.468e+02  4.571e+00   32.11  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 342100 on 12851 degrees of freedom
## Multiple R-squared:  0.2294, Adjusted R-squared:  0.2293
## F-statistic:  1913 on 2 and 12851 DF,  p-value: < 2.2e-16
```

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.

```
HousingOrg_out$standardized.residuals <- rstandard(model4)
HousingOrg_out$studentized.residuals <- rstudent(model4)
HousingOrg_out$cooks.distance <- cooks.distance(model4)
HousingOrg_out$dfbeta <- dfbeta(model4)
HousingOrg_out$leverage <- hatvalues(model4)
HousingOrg_out$covariance.ratios <- covratio(model4)
str(HousingOrg_out)
```

```
## tibble [12,854 x 29] (S3: tbl_df/tbl/data.frame)
##  $ Sale Date               : POSIXct[1:12854], format: "2006-01-03" "2006-01-03" ...
##  $ Sale Price              : num [1:12854] 698000 649990 572500 420000 369900 ...
##  $ sale_reason             : num [1:12854] 1 1 1 1 1 1 1 1 1 1 ...
##  $ sale_instrument         : num [1:12854] 3 3 3 3 3 15 3 3 3 3 ...
##  $ sitetype                : chr [1:12854] "R1" "R1" "R1" "R1" ...
##  $ addr_full               : chr [1:12854] "17021 NE 113TH CT" "11927 178TH PL NE" "13315 174TH AVE N
##  $ zip5                    : num [1:12854] 98052 98052 98052 98052 98052 ...
##  $ ctyname                 : chr [1:12854] "REDMOND" "REDMOND" "REDMOND" "REDMOND" ...
##  $ postalctyn              : chr [1:12854] "REDMOND" "REDMOND" "REDMOND" "REDMOND" ...
##  $ lon                     : num [1:12854] -122 -122 -122 -122 -122 ...
##  $ lat                     : num [1:12854] 47.7 47.7 47.7 47.6 47.7 ...
##  $ building_grade          : num [1:12854] 9 9 8 8 7 7 10 10 9 8 ...
##  $ square_feet_total_living: num [1:12854] 2810 2880 2770 1620 1440 4160 3960 3720 4160 2760 ...
```

```
##  $ bedrooms                : num [1:12854] 4 4 4 3 3 4 5 4 4 4 ...
##  $ bath_full_count         : num [1:12854] 2 2 1 1 1 2 3 2 2 1 ...
##  $ bath_half_count         : num [1:12854] 1 0 1 0 0 1 0 1 1 0 ...
##  $ bath_3qtr_count         : num [1:12854] 0 1 1 1 1 1 1 0 1 1 ...
##  $ year_built              : num [1:12854] 2003 2006 1987 1968 1980 ...
##  $ year_renovated          : num [1:12854] 0 0 0 0 0 0 0 0 0 0 ...
##  $ current_zoning          : chr [1:12854] "R4" "R4" "R6" "R4" ...
##  $ sq_ft_lot               : num [1:12854] 6635 5570 8444 9600 7526 ...
##  $ prop_type               : chr [1:12854] "R" "R" "R" "R" ...
##  $ present_use             : num [1:12854] 2 2 2 2 2 2 2 2 2 2 ...
##  $ standardized.residuals  : Named num [1:12854] -0.102 -0.272 -0.315 -0.267 -0.199 ...
##   ..- attr(*, "names")= chr [1:12854] "1" "2" "3" "4" ...
##  $ studentized.residuals   : Named num [1:12854] -0.102 -0.272 -0.315 -0.267 -0.199 ...
##   ..- attr(*, "names")= chr [1:12854] "1" "2" "3" "4" ...
##  $ cooks.distance          : Named num [1:12854] 4.35e-07 2.98e-06 3.60e-06 4.37e-06 2.52e-06 ...
##   ..- attr(*, "names")= chr [1:12854] "1" "2" "3" "4" ...
##  $ dfbeta                  : num [1:12854, 1:3] 16.1 39.4 -45.2 18.4 -41.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:12854] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:3] "(Intercept)" "building_grade" "square_feet_total_living"
##  $ leverage                : Named num [1:12854] 0.000126 0.000121 0.000109 0.000184 0.00019 ...
##   ..- attr(*, "names")= chr [1:12854] "1" "2" "3" "4" ...
##  $ covariance.ratios       : Named num [1:12854] 1 1 1 1 1 ...
##   ..- attr(*, "names")= chr [1:12854] "1" "2" "3" "4" ...
```

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

```
HousingOrg_out$large.residual <- HousingOrg_out$standardized.residuals > 2 | HousingOrg_out$studentized
str(HousingOrg_out)
```

```
## tibble [12,854 x 30] (S3: tbl_df/tbl/data.frame)
##  $ Sale Date               : POSIXct[1:12854], format: "2006-01-03" "2006-01-03" ...
##  $ Sale Price              : num [1:12854] 698000 649990 572500 420000 369900 ...
##  $ sale_reason             : num [1:12854] 1 1 1 1 1 1 1 1 1 1 ...
##  $ sale_instrument         : num [1:12854] 3 3 3 3 3 15 3 3 3 3 ...
##  $ sitetype                : chr [1:12854] "R1" "R1" "R1" "R1" ...
##  $ addr_full               : chr [1:12854] "17021 NE 113TH CT" "11927 178TH PL NE" "13315 174TH AVE N
##  $ zip5                    : num [1:12854] 98052 98052 98052 98052 98052 ...
##  $ ctyname                 : chr [1:12854] "REDMOND" "REDMOND" "REDMOND" "REDMOND" ...
##  $ postalctyn              : chr [1:12854] "REDMOND" "REDMOND" "REDMOND" "REDMOND" ...
##  $ lon                     : num [1:12854] -122 -122 -122 -122 -122 ...
##  $ lat                     : num [1:12854] 47.7 47.7 47.7 47.6 47.7 ...
##  $ building_grade          : num [1:12854] 9 9 8 8 7 7 10 10 9 8 ...
##  $ square_feet_total_living: num [1:12854] 2810 2880 2770 1620 1440 4160 3960 3720 4160 2760 ...
##  $ bedrooms                : num [1:12854] 4 4 4 3 3 4 5 4 4 4 ...
##  $ bath_full_count         : num [1:12854] 2 2 1 1 1 2 3 2 2 1 ...
##  $ bath_half_count         : num [1:12854] 1 0 1 0 0 1 0 1 1 0 ...
##  $ bath_3qtr_count         : num [1:12854] 0 1 1 1 1 1 1 0 1 1 ...
##  $ year_built              : num [1:12854] 2003 2006 1987 1968 1980 ...
##  $ year_renovated          : num [1:12854] 0 0 0 0 0 0 0 0 0 0 ...
##  $ current_zoning          : chr [1:12854] "R4" "R4" "R6" "R4" ...
##  $ sq_ft_lot               : num [1:12854] 6635 5570 8444 9600 7526 ...
```

```
##  $ prop_type               : chr [1:12854] "R" "R" "R" "R" ...
##  $ present_use             : num [1:12854] 2 2 2 2 2 2 2 2 2 2 ...
##  $ standardized.residuals  : Named num [1:12854] -0.102 -0.272 -0.315 -0.267 -0.199 ...
##   ..- attr(*, "names")= chr [1:12854] "1" "2" "3" "4" ...
##  $ studentized.residuals   : Named num [1:12854] -0.102 -0.272 -0.315 -0.267 -0.199 ...
##   ..- attr(*, "names")= chr [1:12854] "1" "2" "3" "4" ...
##  $ cooks.distance          : Named num [1:12854] 4.35e-07 2.98e-06 3.60e-06 4.37e-06 2.52e-06 ...
##   ..- attr(*, "names")= chr [1:12854] "1" "2" "3" "4" ...
##  $ dfbeta                  : num [1:12854, 1:3] 16.1 39.4 -45.2 18.4 -41.1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:12854] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:3] "(Intercept)" "building_grade" "square_feet_total_living"
##  $ leverage                : Named num [1:12854] 0.000126 0.000121 0.000109 0.000184 0.00019 ...
##   ..- attr(*, "names")= chr [1:12854] "1" "2" "3" "4" ...
##  $ covariance.ratios       : Named num [1:12854] 1 1 1 1 1 ...
##   ..- attr(*, "names")= chr [1:12854] "1" "2" "3" "4" ...
##  $ large.residual          : Named logi [1:12854] FALSE FALSE FALSE FALSE FALSE FALSE ...
##   ..- attr(*, "names")= chr [1:12854] "1" "2" "3" "4" ...
```

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

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

```
## [1] 323
```

```
HousingOrg_out[HousingOrg_out$large.residual , c("Sale Price", "square_feet_total_living", "bath_full_c
```

```
## # A tibble: 323 x 7
##    `Sale Price` square_feet_tot~ bath_full_count bath_half_count bath_3qtr_count
##           <dbl>            <dbl>           <dbl>           <dbl>           <dbl>
##  1       265000             4920               4               1               0
##  2      1390000              660               1               0               0
##  3       229000             3840               0               0               0
##  4       390000             5800               4               1               0
##  5      1588359             3360               2               1               0
##  6      1450000              900               1               0               0
##  7       163000             4710               2               1               2
##  8       270000             5060              23               1               0
##  9       200000             6880               1               1               4
## 10       300000             4490               2               1               1
## # ... with 313 more rows, and 2 more variables: bedrooms <dbl>, sq_ft_lot <dbl>
```

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

```
HousingOrg_out[HousingOrg_out$large.residual , c("leverage" , "cooks.distance","covariance.ratios") ]
```

```
## # A tibble: 323 x 3
##    leverage cooks.distance covariance.ratios
##       <dbl>          <dbl>             <dbl>
##  1 0.000533       0.00103              0.999
##  2 0.000429       0.00152              0.998
##  3 0.000282       0.000395             1.00
##  4 0.000922       0.00202              1.00
##  5 0.000132       0.000226             0.999
##  6 0.000408       0.00151              0.998
##  7 0.000606       0.00125              0.999
##  8 0.000651       0.00146              0.999
##  9 0.00205        0.00814              1.00
## 10 0.000575       0.000980             1.00
## # ... with 313 more rows
```

As we can none of the values in cook's distance is greater than 1 or even closer to 1, so we can say none of the cases is having an undue influence on the model. Lets calculate the average of top 4 leverage which will be equal to 4/21.66=5.415e-04 and we can see all the cases are within boundary of the 4 times the average of 5.415e-04 and many cases are close to 3 times the average. We know the covariance ration should be between [1 + 4(leverage average)] and [1 - 4(leverage average] wg=hich will give us [1 + 4(4/21.66)] = 1.00216 and [1-3(4/12865)] = 0.9978 i.e. the range is 0.978 to 1.00216. Most of the cases lies between these boundaries. From above theries we can conclude that the Cook's distance can raise no or little cause for alarm.

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

```
dwt(model4)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.7245632     0.5508705       0
##  Alternative hypothesis: rho != 0
```

We can see the DW value is 0.55087 from which we ca conclude that the value is within the limits.

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

```
vif(model4)
```

```
##           building_grade square_feet_total_living
##                 2.250209                 2.250209
```

```
1/vif(model4)
```

```
##           building_grade square_feet_total_living
##                0.4444033                0.4444033
```

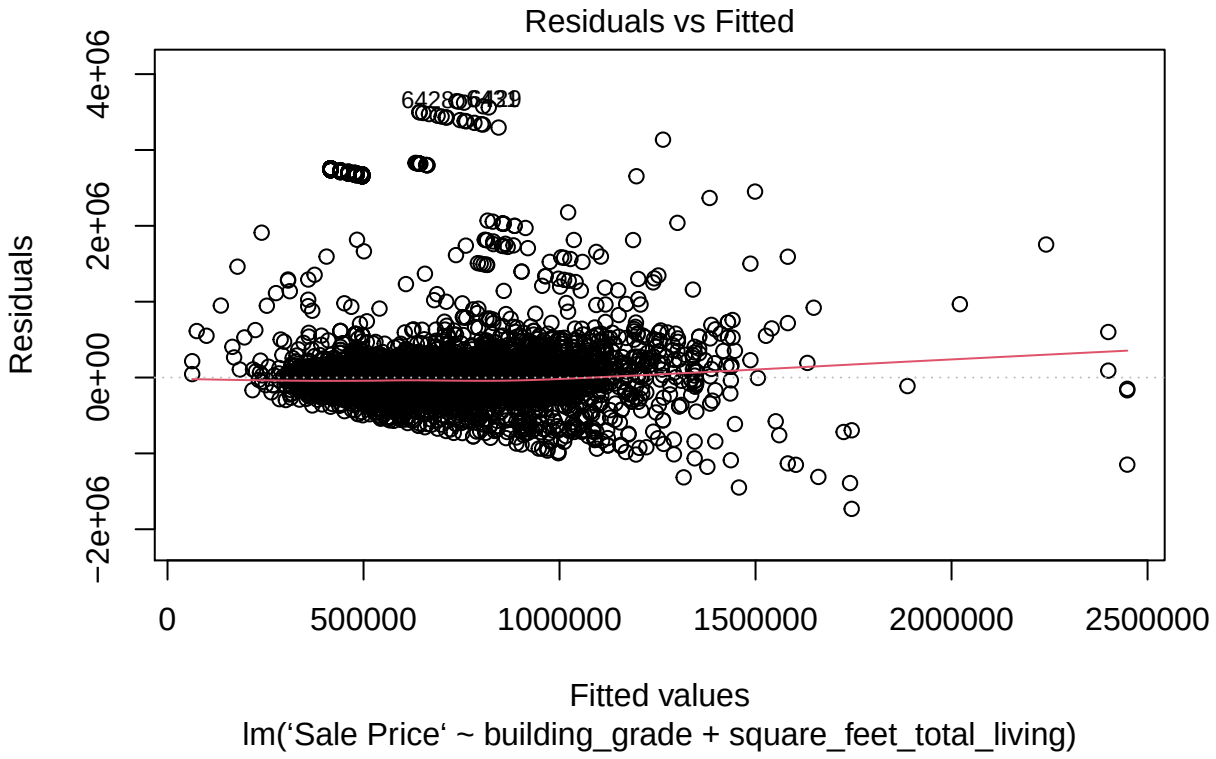
```
mean(vif(model4))
```

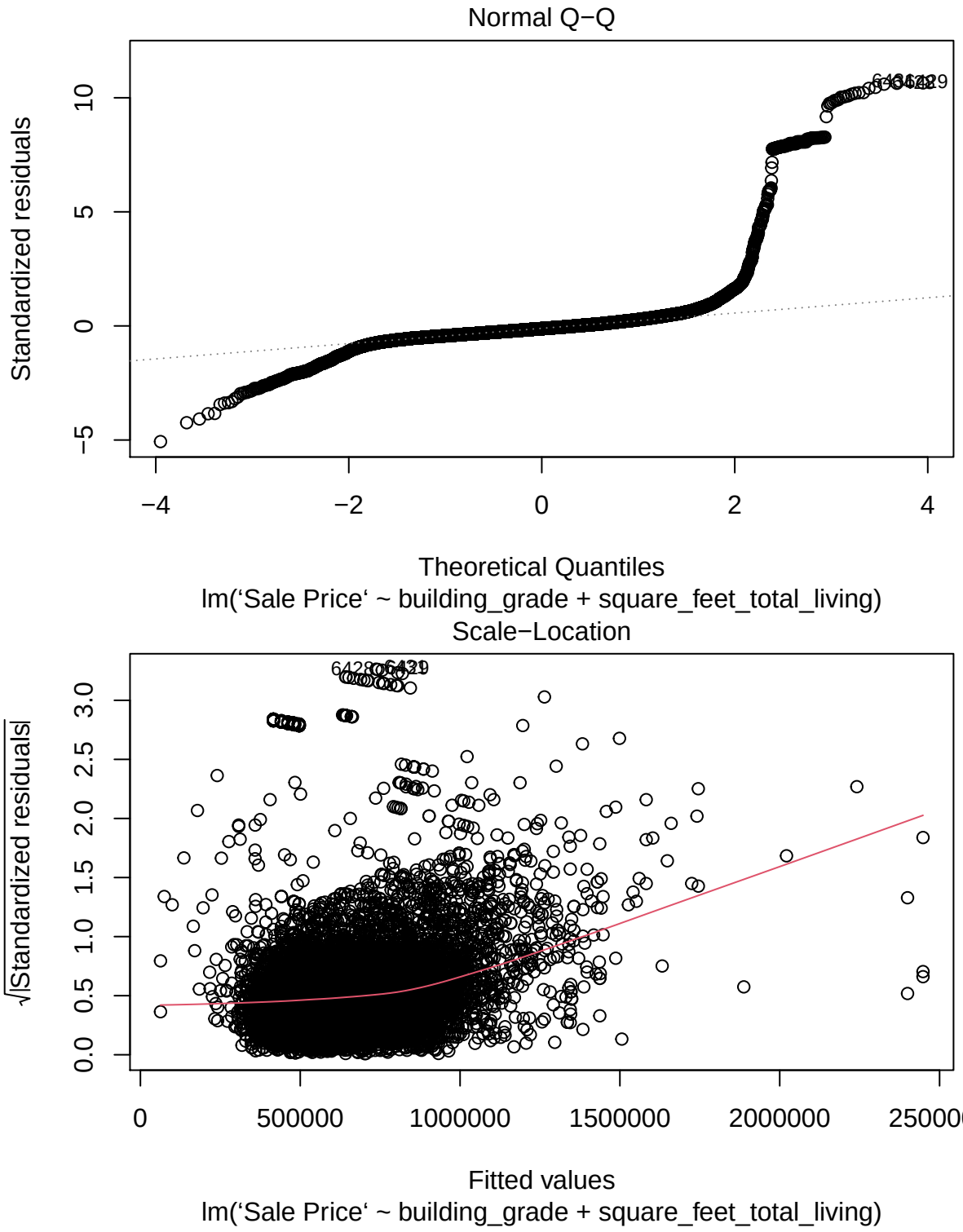
```
## [1] 2.250209
```

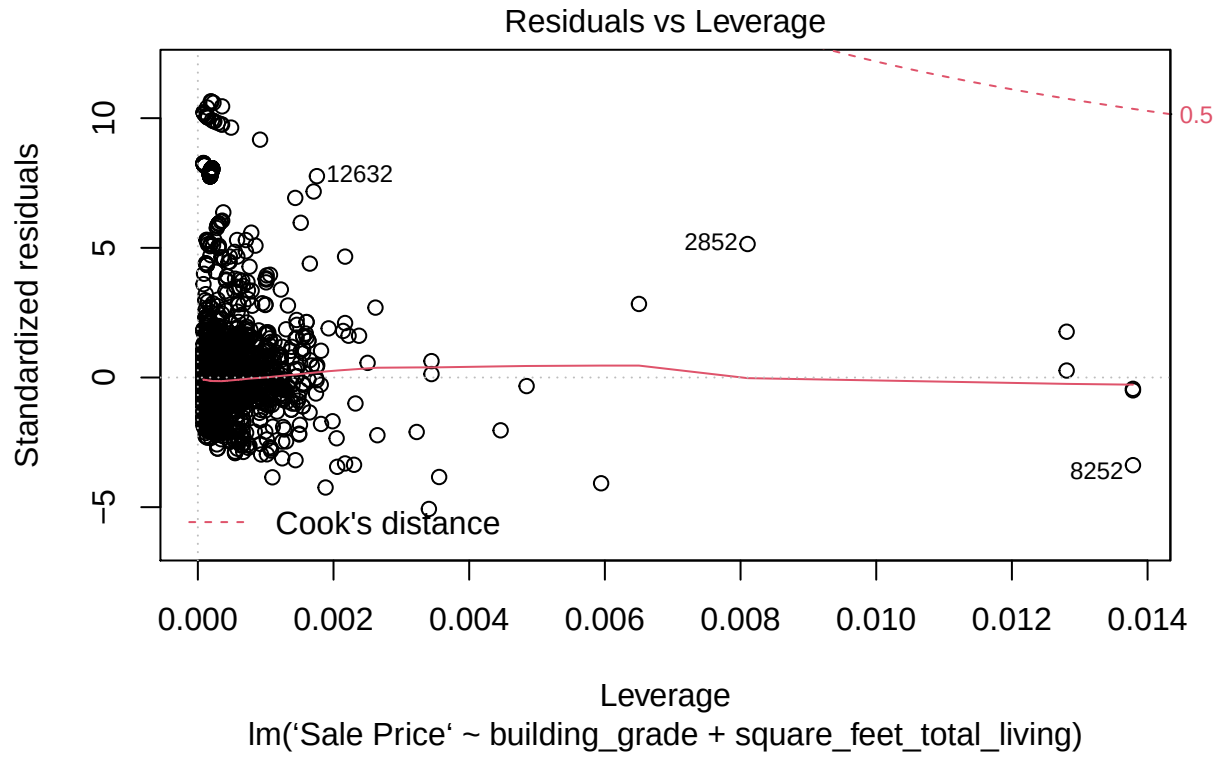
We can stat after seeing the result of above function that the largest vif(2.25) which is not greater than 10. we got 0.44 as tolerance values and the mean of the vif is 2.25 We can conclude from these observation that there is no collinearity within the data.

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(model4)
```

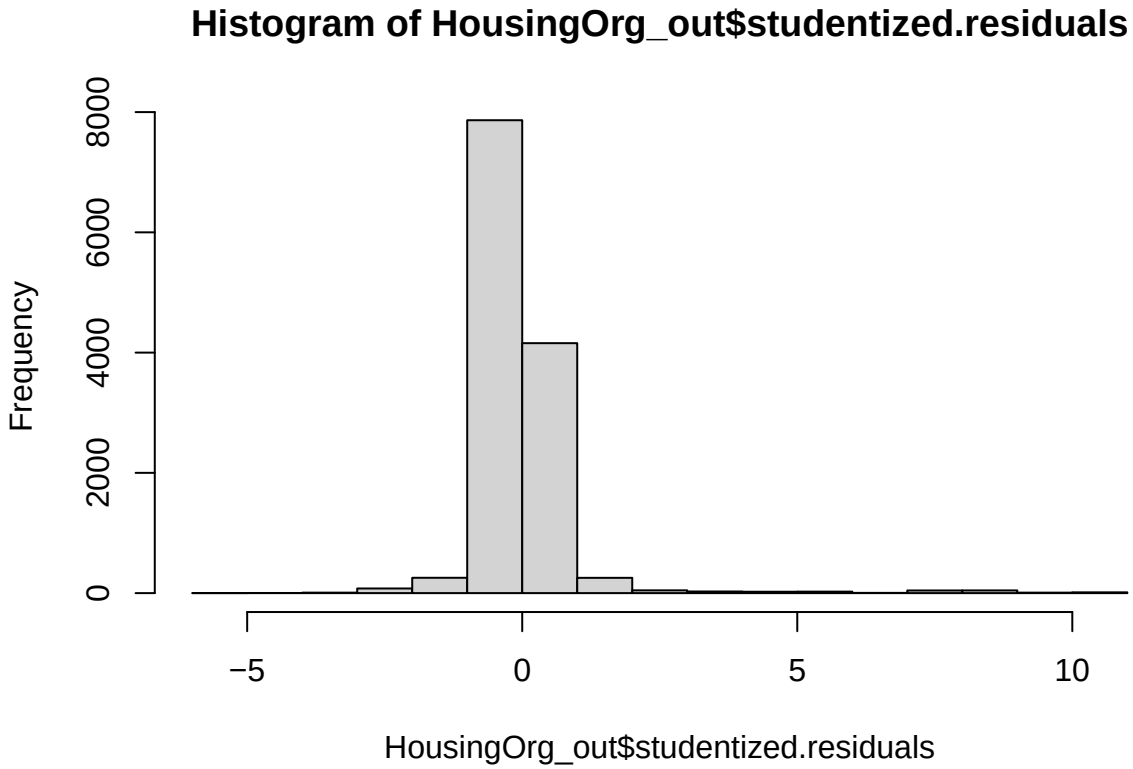



Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(‘Sale Price‘ ~ building_grade + square_feet_total_living)
Scale−Location
√|Standardized residuals|
Fitted values
lm(‘Sale Price‘ ~ building_grade + square_feet_total_living)




```
library(ggplot2)

hist(HousingOrg_out$studentized.residuals)
```



Lets look at the fitted values against residual plot we understand that the values are evenly distributed

around 0. It is save to assume that this is linear graph. we do not see any funnel type data as random variables do not show finite variances so we can say this model does not show homoscedasticity. By looking at the histogram of final model it looks similar to bell shape. so the data is not skewed and it is not biased.

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?

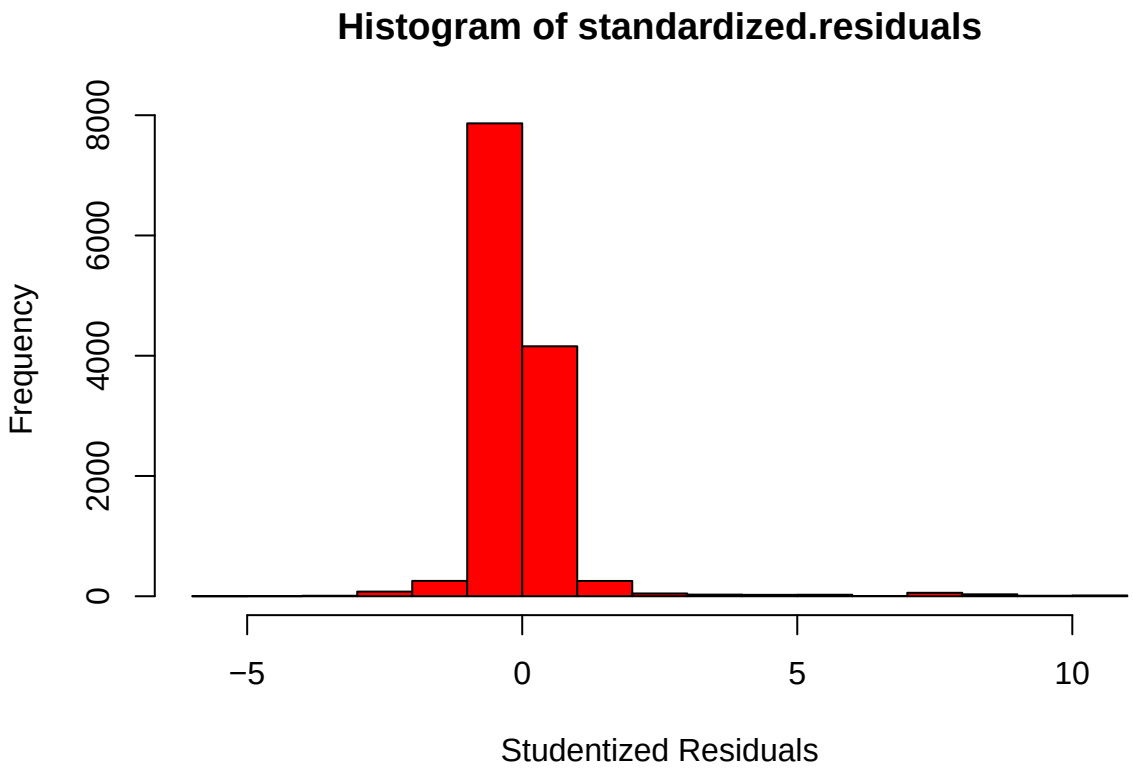
```
with(HousingOrg_out, hist(standardized.residuals, scale="frequency", breaks="Sturges", col="red",
                          xlab="Studentized Residuals"))
```

```
## Warning in plot.window(xlim, ylim, "", ...): "scale" is not a graphical
## parameter
```

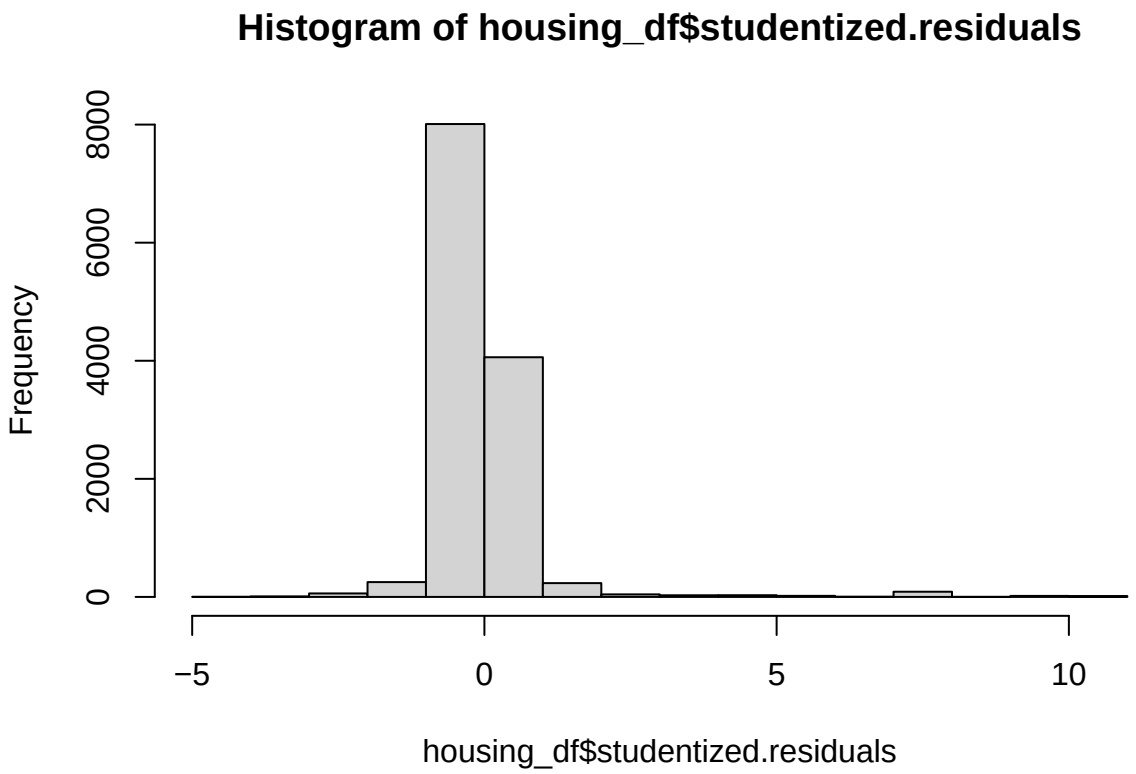
```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...): "scale"
## is not a graphical parameter
```

```
## Warning in axis(1, ...): "scale" is not a graphical parameter
```

```
## Warning in axis(2, ...): "scale" is not a graphical parameter
```




```
housing_df$studentized.residuals <- rstudent(SP_other_lm)
hist(housing_df$studentized.residuals)
```



Looking at the histgram we can see it is bell shaped plot so the designed model is not skewed or not biased. In above plots I used the housing original data and the housing data without the outliers and both show the bell shape plot, i think it is save to believe that both sample and population model are not biased model.

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.