Week 8 Assignment 8.2.3

Koppula Veera

July 30 2021

Housing Data

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

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

#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.

```
## # A tibble: 6 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
## # ... with 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>
```

```
## tibble [12,865 x 24] (S3: tbl_df/tbl/data.frame)
##  $ Sale Date               : POSIXct[1:12865], format: "2006-01-03" "2006-01-03" ...
##  $ Sale Price              : num [1:12865] 698000 649990 572500 420000 369900 ...
##  $ sale_reason             : num [1:12865] 1 1 1 1 1 1 1 1 1 1 ...
##  $ sale_instrument         : num [1:12865] 3 3 3 3 3 15 3 3 3 3 ...
##  $ sale_warning            : chr [1:12865] NA NA NA NA ...
##  $ sitetype                : chr [1:12865] "R1" "R1" "R1" "R1" ...
##  $ addr_full               : chr [1:12865] "17021 NE 113TH CT" "11927 178TH PL NE" "13315 174TH AVE
##  $ zip5                    : num [1:12865] 98052 98052 98052 98052 98052 ...
##  $ ctyname                 : chr [1:12865] "REDMOND" "REDMOND" NA "REDMOND" ...
##  $ postalctyn              : chr [1:12865] "REDMOND" "REDMOND" "REDMOND" "REDMOND" ...
```

```
##  $ lon                     : num [1:12865] -122 -122 -122 -122 -122 ...
##  $ lat                     : num [1:12865] 47.7 47.7 47.7 47.6 47.7 ...
##  $ building_grade          : num [1:12865] 9 9 8 8 7 7 10 10 9 8 ...
##  $ square_feet_total_living: num [1:12865] 2810 2880 2770 1620 1440 4160 3960 3720 4160 2760 ...
##  $ bedrooms                : num [1:12865] 4 4 4 3 3 4 5 4 4 4 ...
##  $ bath_full_count         : num [1:12865] 2 2 1 1 1 2 3 2 2 1 ...
##  $ bath_half_count         : num [1:12865] 1 0 1 0 0 1 0 1 1 0 ...
##  $ bath_3qtr_count         : num [1:12865] 0 1 1 1 1 1 1 0 1 1 ...
##  $ year_built              : num [1:12865] 2003 2006 1987 1968 1980 ...
##  $ year_renovated          : num [1:12865] 0 0 0 0 0 0 0 0 0 0 ...
##  $ current_zoning          : chr [1:12865] "R4" "R4" "R6" "R4" ...
##  $ sq_ft_lot               : num [1:12865] 6635 5570 8444 9600 7526 ...
##  $ prop_type               : chr [1:12865] "R" "R" "R" "R" ...
##  $ present_use             : num [1:12865] 2 2 2 2 2 2 2 2 2 2 ...

## tibble [12,865 x 14] (S3: tbl_df/tbl/data.frame)
##  $ Sale Date               : POSIXct[1:12865], format: "2006-01-03" "2006-01-03" ...
##  $ Sale Price              : num [1:12865] 698000 649990 572500 420000 369900 ...
##  $ zip5                    : num [1:12865] 98052 98052 98052 98052 98052 ...
##  $ postalctyn              : chr [1:12865] "REDMOND" "REDMOND" "REDMOND" "REDMOND" ...
##  $ square_feet_total_living: num [1:12865] 2810 2880 2770 1620 1440 4160 3960 3720 4160 2760 ...
##  $ bedrooms                : num [1:12865] 4 4 4 3 3 4 5 4 4 4 ...
##  $ bath_full_count         : num [1:12865] 2 2 1 1 1 2 3 2 2 1 ...
##  $ bath_half_count         : num [1:12865] 1 0 1 0 0 1 0 1 1 0 ...
##  $ bath_3qtr_count         : num [1:12865] 0 1 1 1 1 1 1 0 1 1 ...
##  $ year_built              : num [1:12865] 2003 2006 1987 1968 1980 ...
##  $ current_zoning          : chr [1:12865] "R4" "R4" "R6" "R4" ...
##  $ sq_ft_lot               : num [1:12865] 6635 5570 8444 9600 7526 ...
##  $ prop_type               : chr [1:12865] "R" "R" "R" "R" ...
##  $ present_use             : num [1:12865] 2 2 2 2 2 2 2 2 2 2 ...

## # A tibble: 6 x 14
##   `Sale Date`         `Sale Price`  zip5 postalctyn square_feet_total_~ bedrooms
##   <dttm>                     <dbl> <dbl> <chr>                    <dbl>    <dbl>
## 1 2006-01-03 00:00:00       698000 98052 REDMOND                   2810        4
## 2 2006-01-03 00:00:00       649990 98052 REDMOND                   2880        4
## 3 2006-01-03 00:00:00       572500 98052 REDMOND                   2770        4
## 4 2006-01-03 00:00:00       420000 98052 REDMOND                   1620        3
## 5 2006-01-03 00:00:00       369900 98052 REDMOND                   1440        3
## 6 2006-01-03 00:00:00       184667 98053 REDMOND                   4160        4
## # ... with 8 more variables: bath_full_count <dbl>, bath_half_count <dbl>,
## #   bath_3qtr_count <dbl>, year_built <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

I did not want to do a lot of data removal, I am not sure which data would be useful for the assignment and I wanted to leave some options. I did remove ten columns. I chose this by reviewing the structure. I could see that even though the city was left blank on some rows the zip & postalcity was not.Considering that we use zips/postalcity for tracking our locations I felt only one of these columns might be needed and the one with the best data is the zip &postal city. In addition since we were not provided translation reference for

what the sales reason, instrument and warning are - they did not seem to provide significant information for the analysis. I also ran a command to clean up data that still had missing elements after the removal of those ten columns. Though it appears that the the removal of the na's was not needed after removing the columns. After all the data cleanup I have 12865 lines of data left in the data set (this should be signifant for rest of the analysis)

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

I used lm command to create the code for the variables, because I did it in this fashion it already created the summaries for me. We can see a jump in the r2 and adjusted r2 between these two. f the second variable I chose bathrooms, bedrooms count along with squarefootage as predictors. Based on my experience,among houses with same squarefootage, houses with higher bedrooms and bathrooms have better potential for higher price.

#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?

```
##
## Call:
## lm(formula = housing_Updated$`Sale Price` ~ housing_Updated$square_feet_total_living,
##     data = housing_Updated)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1800136  -120257   -41547    44028  3811745
##
## Coefficients:
##                                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             1.891e+05  8.745e+03   21.62   <2e-16
## housing_Updated$square_feet_total_living 1.857e+02  3.208e+00   57.88   <2e-16
##
## (Intercept)                             ***
## housing_Updated$square_feet_total_living ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 360200 on 12863 degrees of freedom
## Multiple R-squared:  0.2066, Adjusted R-squared:  0.2066
## F-statistic:  3351 on 1 and 12863 DF,  p-value: < 2.2e-16
```

```
##
## Call:
## lm(formula = housing_Updated$`Sale Price` ~ housing_Updated$square_feet_total_living +
##     bath_full_count + bath_half_count + bath_3qtr_count + bedrooms +
##     sq_ft_lot, data = housing_Updated)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1917422  -118558   -40743    43781  3775867
##
## Coefficients:
##                                           Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)                                2.055e+05  1.457e+04  14.105  < 2e-16
## housing_Updated$square_feet_total_living  1.829e+02  5.288e+00  34.580  < 2e-16
## bath_full_count                            3.640e+04  7.248e+03   5.023 5.17e-07
## bath_half_count                            1.081e+04  7.167e+03   1.508   0.1315
## bath_3qtr_count                           -9.397e+03  6.998e+03  -1.343   0.1793
## bedrooms                                  -2.268e+04  4.550e+03  -4.984 6.31e-07
## sq_ft_lot                                  1.062e-01  5.794e-02   1.832   0.0669
##
## (Intercept)                               ***
## housing_Updated$square_feet_total_living ***
## bath_full_count                           ***
## bath_half_count
## bath_3qtr_count
## bedrooms                                  ***
## sq_ft_lot                                 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 358900 on 12858 degrees of freedom
## Multiple R-squared:  0.2127, Adjusted R-squared:  0.2123
## F-statistic: 578.8 on 6 and 12858 DF,  p-value: < 2.2e-16
```

I used the r lm to create the code for the variables, because I did it in this fashion it already created the summaries for me. We can see a jump in the r2 and adjusted r2 between these two.

Residual standard error: 360200 on 12863 degrees of freedom Multiple R-squared: 0.2066, Adjusted R-squared: 0.2066

Residual standard error: 358900 on 12858 degrees of freedom Multiple R-squared: 0.2127, Adjusted R-squared: 0.2123 Based on my experience,among houses with same squarefootage, houses with higher bedrooms and bathrooms have better potential for higher price,also the results indicate same directionality.

#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?

I used the compareCoefs and the anova to help me look at the information. from this I was able to see that the lot size, and bathrooms have the most significance when it comes to the difference in the comparisons. So with that if i had to do a full analysis I would used price per square foot, square foot of the lot and number of bedrooms.

```
## Calls:
## 1: lm(formula = housing_Updated$`Sale Price` ~
##   housing_Updated$square_feet_total_living, data = housing_Updated)
## 2: lm(formula = housing_Updated$`Sale Price` ~
##   housing_Updated$square_feet_total_living + bath_full_count +
##   bath_half_count + bath_3qtr_count + bedrooms + sq_ft_lot, data =
##   housing_Updated)
##
##                                          Model 1 Model 2
## (Intercept)                               189107  205466
## SE                                          8745   14567
##
## housing_Updated$square_feet_total_living  185.72  182.85
## SE                                          3.21    5.29
##
## bath_full_count                                    36403
```

```
## SE                                                 7248
##
## bath_half_count                                  10809
## SE                                                 7167
##
## bath_3qtr_count                                  -9397
## SE                                                 6998
##
## bedrooms                                        -22680
## SE                                                 4550
##
## sq_ft_lot                                       0.1062
## SE                                              0.0579
##
```

```
## Analysis of Variance Table
##
## Model 1: housing_Updated$‘Sale Price‘ ~ housing_Updated$square_feet_total_living
## Model 2: housing_Updated$‘Sale Price‘ ~ housing_Updated$square_feet_total_living +
##     bath_full_count + bath_half_count + bath_3qtr_count + bedrooms +
##     sq_ft_lot
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
## 1  12863 1.6689e+15
## 2  12858 1.6562e+15  5 1.2626e+13 19.604 < 2.2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

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

Our confidence level is defaulted to 95%. I was able to locate the means pair testing in the statistics option, this allowed me to see the confidence intervals as well as the means of the Price and square footage.

```
##
##  Paired t-test
##
## data:  housing_Updated$‘Sale Price‘ and housing_Updated$square_feet_total_living
## t = 184.82, df = 12864, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  651217.6 665178.8
## sample estimates:
## mean of the differences
##                658198.2
```

#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.

```
## Calls:
## 1: lm(formula = housing_Updated$‘Sale Price‘ ~
##   housing_Updated$square_feet_total_living, data = housing_Updated)
## 2: lm(formula = housing_Updated$‘Sale Price‘ ~
##   housing_Updated$square_feet_total_living + bath_full_count +
##   bath_half_count + bath_3qtr_count + bedrooms + sq_ft_lot, data =
```

```
##   housing_Updated)
##
##                                          Model 1 Model 2
## (Intercept)                               189107  205466
## SE                                          8745   14567
##
## housing_Updated$square_feet_total_living  185.72  182.85
## SE                                          3.21    5.29
##
## bath_full_count                                    36403
## SE                                                  7248
##
## bath_half_count                                    10809
## SE                                                  7167
##
## bath_3qtr_count                                    -9397
## SE                                                  6998
##
## bedrooms                                          -22680
## SE                                                  4550
##
## sq_ft_lot                                         0.1062
## SE                                                0.0579
##
```

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

```
##
## Call:
## lm(formula = housing_orginial$`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_orginial)
##
## 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
```

```
##       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
```

```
##       rstudent unadjusted p-value Bonferroni p
## 11992 10.62908         2.8052e-26   3.6089e-22
## 4649  10.49256         1.1864e-25   1.5263e-21
## 6438  10.42218         2.4775e-25   3.1873e-21
## 6430  10.41696         2.6161e-25   3.3656e-21
## 6437  10.36474         4.5024e-25   5.7923e-21
## 6431  10.26036         1.3225e-24   1.7014e-20
## 6436  10.22905         1.8232e-24   2.3455e-20
## 6441  10.15082         4.0508e-24   5.2113e-20
## 6432  10.12475         5.2784e-24   6.7907e-20
## 6442  10.07261         8.9440e-24   1.1506e-19
```

```
##       rstudent unadjusted p-value Bonferroni p
## 11992 10.59113         4.1963e-26   5.3986e-22
## 4649  10.45584         1.7432e-25   2.2427e-21
## 6430  10.44787         1.8947e-25   2.4375e-21
## 6437  10.39635         3.2426e-25   4.1716e-21
## 6438  10.38900         3.5001e-25   4.5029e-21
## 6431  10.29271         9.4816e-25   1.2198e-20
## 6436  10.25951         1.3342e-24   1.7164e-20
## 6432  10.21506         2.1042e-24   2.7071e-20
## 6442  10.16464         3.5196e-24   4.5280e-20
## 6441  10.14209         4.4268e-24   5.6951e-20
```

with above evaluations I decided to remove the 11 rows of data that are outliers. The original data had line 6433 and with the adjusted models row 4649 is listed. I created updated data frames as well as models with out the outlier rows.

```
## tibble [12,854 x 24] (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 ...
##  $ sale_warning            : chr [1:12854] NA NA NA NA ...
##  $ 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" NA "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 ...

## tibble [12,854 x 14] (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 ...
##  $ zip5                    : num [1:12854] 98052 98052 98052 98052 98052 ...
##  $ postalctyn              : chr [1:12854] "REDMOND" "REDMOND" "REDMOND" "REDMOND" ...
##  $ 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 ...
##  $ 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 ...

##
## Call:
## lm(formula = Out_housing_updated$`Sale Price` ~ square_feet_total_living,
##     data = Out_housing_updated)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1794523  -117206   -38480    46773  3607972
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.870e+05  8.347e+03   22.40   <2e-16 ***
## square_feet_total_living 1.853e+02  3.062e+00   60.52   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 343800 on 12852 degrees of freedom
## Multiple R-squared:  0.2218, Adjusted R-squared:  0.2217
## F-statistic:  3662 on 1 and 12852 DF,  p-value: < 2.2e-16
```

```
##
## Call:
## lm(formula = Out_housing_updated$`Sale Price` ~ square_feet_total_living +
##     bath_full_count + bath_half_count + bath_3qtr_count + bedrooms +
##     sq_ft_lot, data = Out_housing_updated)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1680220  -115787   -38535    46497  3622874
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               2.094e+05  1.390e+04  15.065  < 2e-16 ***
## square_feet_total_living  1.866e+02  5.052e+00  36.931  < 2e-16 ***
## bath_full_count           3.510e+04  6.918e+03   5.074 3.95e-07 ***
## bath_half_count           1.038e+04  6.839e+03   1.518    0.129
## bath_3qtr_count          -8.414e+03  6.678e+03  -1.260    0.208
## bedrooms                 -2.582e+04  4.343e+03  -5.945 2.84e-09 ***
## sq_ft_lot                -5.166e-02  5.666e-02  -0.912    0.362
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 342400 on 12847 degrees of freedom
## Multiple R-squared:  0.2282, Adjusted R-squared:  0.2278
## F-statistic:   633 on 6 and 12847 DF,  p-value: < 2.2e-16
```

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

```
## tibble [12,854 x 20] (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 ...
##  $ zip5                    : num [1:12854] 98052 98052 98052 98052 98052 ...
##  $ postalctyn              : chr [1:12854] "REDMOND" "REDMOND" "REDMOND" "REDMOND" ...
##  $ 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 ...
##  $ 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] -0.0369 -0.1605 -0.2543 -0.1179 -0.1665 ...
##   ..- attr(*, "names")= chr [1:12854] "1" "2" "3" "4" ...
##  $ studentized.residuals   : Named num [1:12854] -0.0369 -0.1605 -0.2543 -0.1179 -0.1665 ...
```

```
##   ..- attr(*, "names")= chr [1:12854] "1" "2" "3" "4" ...
##  $ cooks.distance          : Named num [1:12854] 3.89e-08 9.83e-07 3.35e-06 5.31e-07 1.12e-06 ...
##   ..- attr(*, "names")= chr [1:12854] "1" "2" "3" "4" ...
##  $ dfbeta                  : num [1:12854, 1:7] 0.852 6.013 -7.22 -15.475 -20.518 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:12854] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:7] "(Intercept)" "square_feet_total_living" "bath_full_count" "bath_half_count" ..
##  $ leverage                : Named num [1:12854] 0.0002 0.000267 0.000363 0.000267 0.000282 ...
##   ..- 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" ...
```

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

```
## tibble [12,854 x 21] (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 ...
##  $ zip5                    : num [1:12854] 98052 98052 98052 98052 98052 ...
##  $ postalctyn              : chr [1:12854] "REDMOND" "REDMOND" "REDMOND" "REDMOND" ...
##  $ 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 ...
##  $ 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] -0.0369 -0.1605 -0.2543 -0.1179 -0.1665 ...
##   ..- attr(*, "names")= chr [1:12854] "1" "2" "3" "4" ...
##  $ studentized.residuals   : Named num [1:12854] -0.0369 -0.1605 -0.2543 -0.1179 -0.1665 ...
##   ..- attr(*, "names")= chr [1:12854] "1" "2" "3" "4" ...
##  $ cooks.distance          : Named num [1:12854] 3.89e-08 9.83e-07 3.35e-06 5.31e-07 1.12e-06 ...
##   ..- attr(*, "names")= chr [1:12854] "1" "2" "3" "4" ...
##  $ dfbeta                  : num [1:12854, 1:7] 0.852 6.013 -7.22 -15.475 -20.518 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:12854] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:7] "(Intercept)" "square_feet_total_living" "bath_full_count" "bath_half_count" ..
##  $ leverage                : Named num [1:12854] 0.0002 0.000267 0.000363 0.000267 0.000282 ...
##   ..- 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 TRUE ...
##   ..- attr(*, "names")= chr [1:12854] "1" "2" "3" "4" ...
```

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

```
## [1] 331
```

```
## # A tibble: 331 x 7
##    `Sale Price` square_feet_tot~ bath_full_count bath_half_count bath_3qtr_count
##           <dbl>            <dbl>           <dbl>           <dbl>           <dbl>
```

```
##  1       184667             4160              2               1               1
##  2       265000             4920              4               1               0
##  3      1390000              660              1               0               0
##  4       229000             3840              0               0               0
##  5       390000             5800              4               1               0
##  6      1588359             3360              2               1               0
##  7      1450000              900              1               0               0
##  8      1490000             3540              2               0               1
##  9       163000             4710              2               1               2
## 10       270000             5060             23               1               0
## # ... with 321 more rows, and 2 more variables: bedrooms <dbl>, sq_ft_lot <dbl>
```

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

```
## # A tibble: 331 x 3
##    leverage cooks.distance covariance.ratios
##       <dbl>          <dbl>             <dbl>
##  1 0.000391       0.000282             0.998
##  2 0.00122        0.00122              0.998
##  3 0.00250        0.00328              0.998
##  4 0.00598        0.00346              1.00
##  5 0.00127        0.00131              0.998
##  6 0.000766       0.000489             0.999
##  7 0.000497       0.000719             0.996
##  8 0.0192         0.0121               1.02
##  9 0.000969       0.000926             0.998
## 10 0.163          0.720                1.18
## # ... with 321 more rows
```

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

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.7112925     0.5774113       0
##  Alternative hypothesis: rho != 0
```

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

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

## The following object is masked from 'package:MASS':
##
##     cement

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

## Tolerance and Variance Inflation Factor
## ---------------------------------------
```

```
##                  Variables Tolerance      VIF
## 1 square_feet_total_living 0.3645947 2.742772
## 2          bath_full_count 0.4499431 2.222503
## 3          bath_half_count 0.7040653 1.420323
## 4          bath_3qtr_count 0.4838620 2.066705
## 5                 bedrooms 0.6298003 1.587805
## 6                sq_ft_lot 0.9171007 1.090393
##
##
## Eigenvalue and Condition Index
## ------------------------------
##   Eigenvalue Condition Index    intercept square_feet_total_living
## 1 5.05272828        1.000000 1.741253e-03             1.804270e-03
## 2 0.82429420        2.475836 6.298138e-04             7.443847e-05
## 3 0.74089228        2.611472 1.164028e-05             5.110552e-08
## 4 0.25683599        4.435423 9.123483e-03             1.153436e-03
## 5 0.06873119        8.574051 2.883965e-01             3.106294e-01
## 6 0.03452421       12.097652 2.560654e-02             1.750905e-01
## 7 0.02199384       15.156973 6.744907e-01             5.112479e-01
##   bath_full_count bath_half_count bath_3qtr_count     bedrooms    sq_ft_lot
## 1     0.001832821     0.007750751     0.005020372 0.0014095684 0.0067518989
## 2     0.001037857     0.013301868     0.004138862 0.0004741795 0.8774691509
## 3     0.001871553     0.081290744     0.301929111 0.0001182788 0.0425711363
## 4     0.037604356     0.709765621     0.142552323 0.0041984957 0.0007305889
## 5     0.052879262     0.007828104     0.007113851 0.0332157786 0.0288471504
## 6     0.591908601     0.112891060     0.421164384 0.4384248506 0.0065457880
## 7     0.312865551     0.067171853     0.118081097 0.5221588484 0.0370842867
```

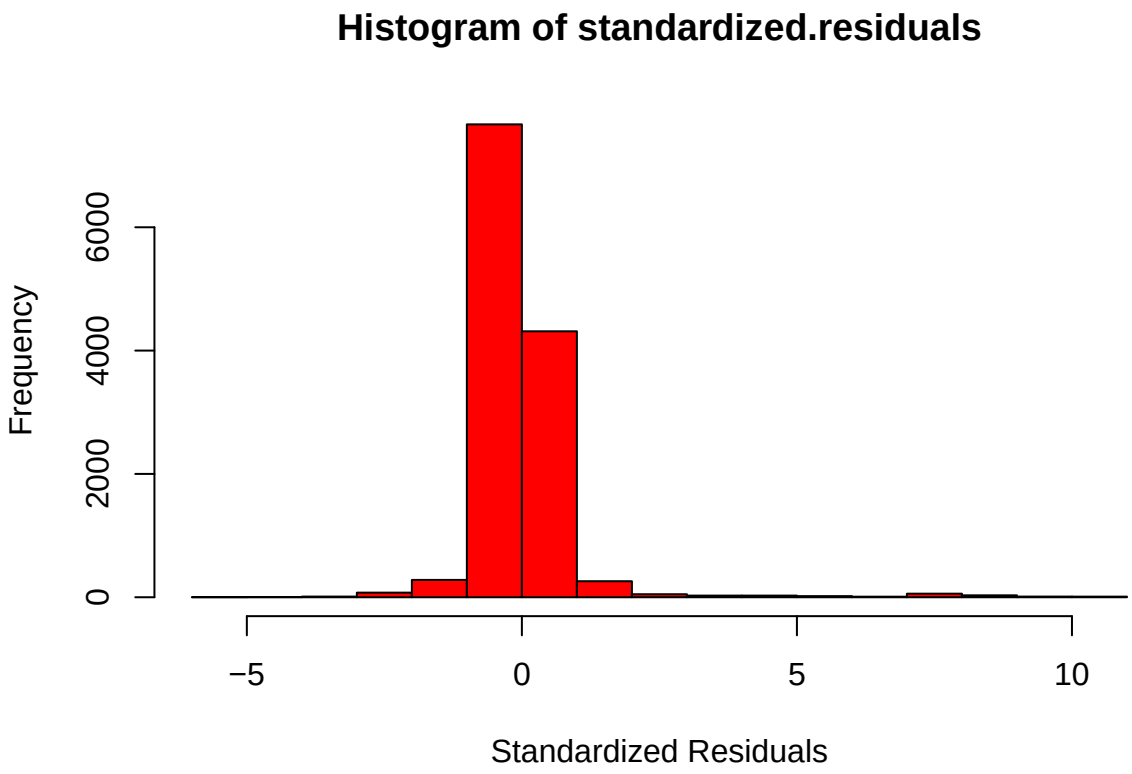
#Visually check the assumptions related to the residuals using the plot() and hist() functions. Summarize what each graph is informing you of and if any anomalies are present.

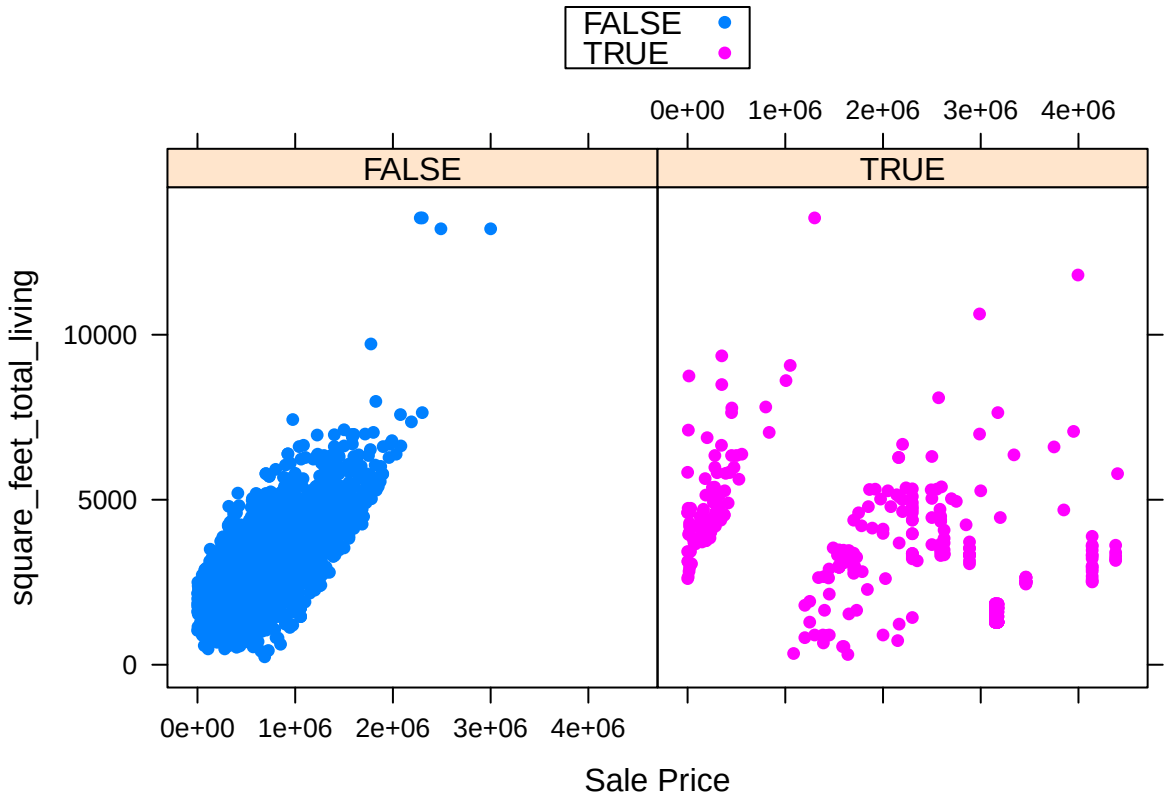
```
## Warning in plot.window(xlim, ylim, "", ...): "scale" is not a graphical
## parameter
```

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

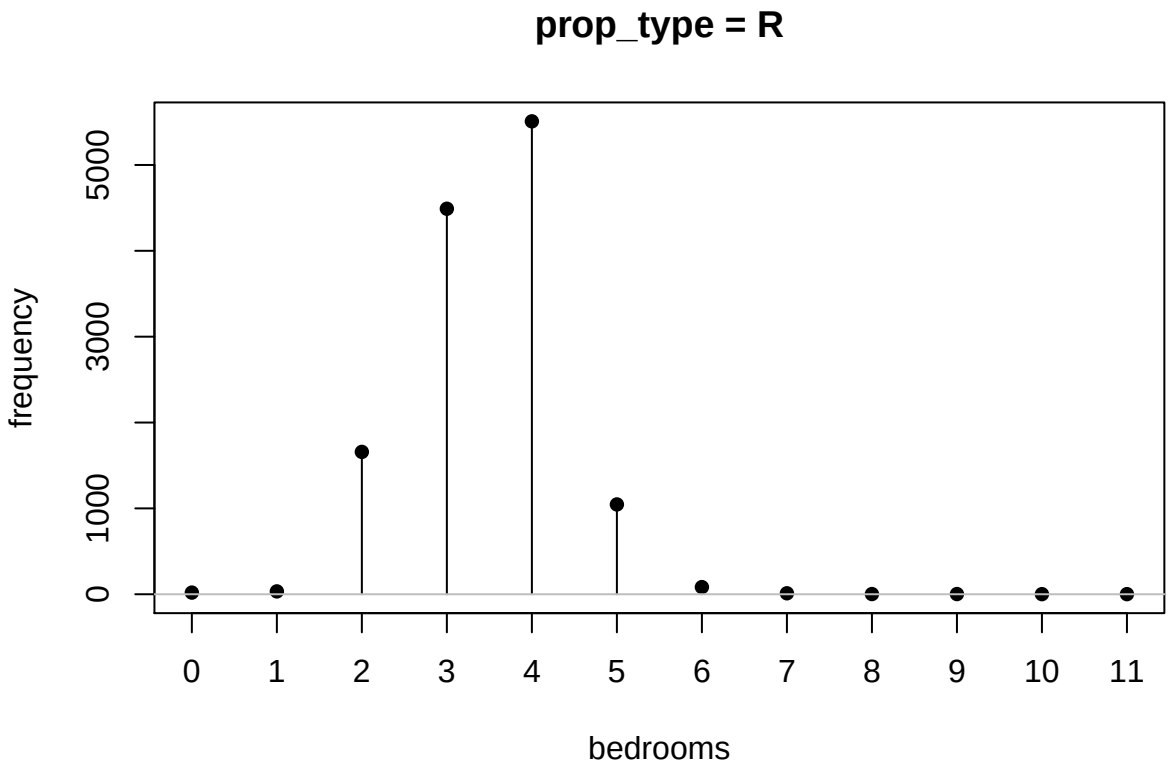
```
## Warning in axis(1, ...): "scale" is not a graphical parameter
```

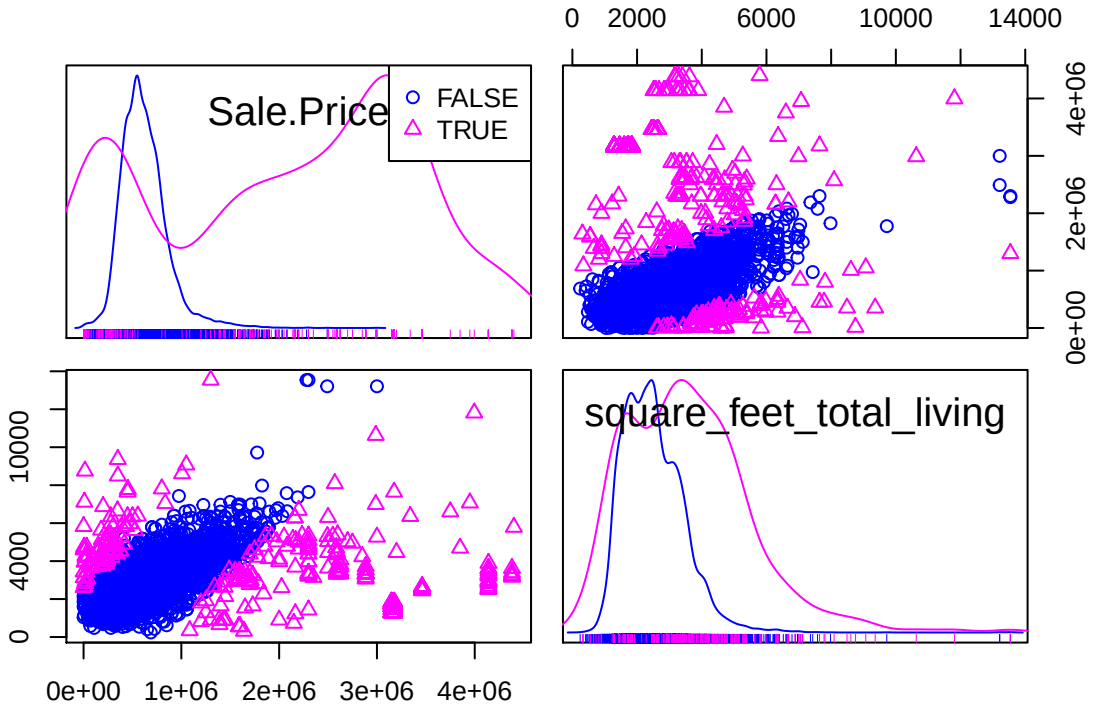
```
## Warning in axis(2, ...): "scale" is not a graphical parameter
```


Histogram of standardized.residuals
Frequency
0
2000
4000
6000
−5
0
5
10
Standardized Residuals




```
## Loading required package: sandwich
```


prop_type = R
frequency
0
1000
3000
5000
0
1
2
3
4
5
6
7
8
9
10
11
bedrooms



High frequency of availability of houses are 4 bedrooms, followed by 3 bedrooms. They are the most sold and their average price per square foot living space is concentrated. There are some anamolies with houses that have either very large or small square foot living space to price per square foot.

#Overall, is this regression model unbiased? If an unbiased regression model, what does this tell us about the sample vs. the entire population model? This regression model is fairly unbiased. When a model is unbiased, it has Samples selected for building the model truly represents the distribution of entire population in the data set.