Week 8-9 - Exercise 8.2

Ganesh Kale

May 14, 2021

Load the required packages

```
library(readxl)
library(dplyr)
library(QuantPsyc)
library(car)
```

load the data which was transformed by removing unwanted columns and na/null values

```
## # A tibble: 6 x 9
##   `Sale Price` building_grade square_feet_total_living bedrooms bath_full_count
##          <dbl>          <dbl>                    <dbl>    <dbl>           <dbl>
## 1       698000              9                     2810        4               2
## 2       649990              9                     2880        4               2
## 3       572500              8                     2770        4               1
## 4       420000              8                     1620        3               1
## 5       369900              7                     1440        3               1
## 6       184667              7                     4160        4               2
## # ... with 4 more variables: year_built <dbl>, sq_ft_lot <dbl>, zip5 <dbl>,
## #   total_area <dbl>
```

Explain any transformations or modifications you made to the dataset

1. Checked and removed outliers from the features
2. Created new column - sq feet total lot size - total_area = square_feet_total_living + sq_ft_lot
3. Removed NA or null value rows from the data sets
4. Changed Sale Date type to Date format
5. Removed unwanted columns such as - sale date, sale_reason, sale_instrument, addr_full, cityname, postalctyn,lon,lat, present use etc.

b.ii] Create Two variables. . . . .

Created house.1 dataframe with sale price and sq_ft_lot for simple regression

Created house.2 dataframe with sale price and other variables for multiple regression, selected predictors: - sq_ft_lot,building_grade,square_feet_total_living,bedrooms,bath_full_count,year_built

The additional predictors selected based on

1. created scatter plot of each predictor and calculated correlation coefficient to see how strong they are related.
2. Checked multicollinearity of all the predictors pairs and removed the ones which have strong correlations
3. Checked the variance of predictors and they do not have 0 variance included in dataset in addition to above steps.

```
house.1 <- house1 %>% select(`Sale Price` ,sq_ft_lot)
print("Simple Regression variable")
```

```
## [1] "Simple Regression variable"
```

```
head(house.1)
```

```
## # A tibble: 6 x 2
##   ‘Sale Price‘ sq_ft_lot
##          <dbl>     <dbl>
## 1       698000      6635
## 2       649990      5570
## 3       572500      8444
## 4       420000      9600
## 5       369900      7526
## 6       184667      7280
```

```
house.2 <- house1 %>% select(`Sale Price` ,sq_ft_lot,building_grade,square_feet_total_living,bedrooms,ba
print("Multiple Regression variable")
```

```
## [1] "Multiple Regression variable"
```

```
head(house.2)
```

```
## # A tibble: 6 x 7
##   ‘Sale Price‘ sq_ft_lot building_grade square_feet_total_living bedrooms
##          <dbl>     <dbl>          <dbl>                    <dbl>    <dbl>
## 1       698000      6635              9                     2810        4
## 2       649990      5570              9                     2880        4
## 3       572500      8444              8                     2770        4
## 4       420000      9600              8                     1620        3
## 5       369900      7526              7                     1440        3
## 6       184667      7280              7                     4160        4
## # ... with 2 more variables: bath_full_count <dbl>, year_built <dbl>
```

```
simple_result <- lm(`Sale Price` ~ sq_ft_lot, data = house.1)

formula = `Sale Price` ~ sq_ft_lot + building_grade + square_feet_total_living + bedrooms + bath_full_c
multi_result <- lm(formula , data = house.2)
```

b iii] Execute a summary() function on two variables. . . .

```
summary(simple_result)
```

Summary result of Simple Linear regression and Multiple Linear Regression

```
##
## Call:
## lm(formula = ‘Sale Price‘ ~ sq_ft_lot, data = house.1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -781186 -146630  -23612  117672 1480253
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.674e+05  2.845e+03  199.45   <2e-16 ***
## sq_ft_lot   4.206e+00  1.626e-01   25.87   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 219900 on 11311 degrees of freedom
## Multiple R-squared:  0.05585,    Adjusted R-squared:  0.05577
## F-statistic: 669.1 on 1 and 11311 DF,  p-value: < 2.2e-16
```

```
summary(multi_result)
```

```
##
## Call:
## lm(formula = formula, data = house.2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1046471   -78345    -8858    67138  1160980
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -1.825e+06  2.455e+05  -7.434 1.13e-13 ***
## sq_ft_lot                -4.011e-01  1.305e-01  -3.074  0.00212 **
## building_grade            6.172e+04  2.085e+03  29.608  < 2e-16 ***
## square_feet_total_living  1.396e+02  3.114e+00  44.822  < 2e-16 ***
## bedrooms                 -6.033e+03  2.261e+03  -2.668  0.00764 **
## bath_full_count           4.010e+03  3.054e+03   1.313  0.18921
## year_built                8.036e+02  1.240e+02   6.483 9.37e-11 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 146500 on 11306 degrees of freedom
## Multiple R-squared:  0.5814, Adjusted R-squared:  0.5812
## F-statistic:  2618 on 6 and 11306 DF,  p-value: < 2.2e-16
```

R-squared - It is a measure of how much of the variability in the outcome is accounted for by the predictors. Adjusted R-squared - It gives some idea of how well model generalizes, that means when we implement the same model on the population, what would be value of R-squared on population data.

Simple Regression Result - Based on this result, the R2 values is 0.05585, which tells that sq_ft_lot variable accounts for 5.6% variation in the sale price of house, which indicates that there are other variables that have high impact the house sale price. The Adjusted R2 is same as R2, which tells that variation in sales price when same model was run on entire population.

Multiple Regression Result - Based on the result, R2 value is 0.5814, which tells that all the predictors accounts for 58% variation in house sale price. The adjusted R2 is 0.5812 and the difference between R2 and Adjusted R2 is 0.0002, This explains that if the models would have run on the population there could have shrinkage of 0.02% in the R2 value and which is very trivial or minimal difference, so we can say that model will predict similar result when run on population.

Overall, addition of more predictor variables improved the R2 value, which explains that large variation in house sales price when we include more predictors. Additional predictors account for more variation in the sales price of house.

b iV] What are the standardized betas for each parameter and what do the values indicate?

```
lm.beta(multi_result)
```

```
##                sq_ft_lot           building_grade square_feet_total_living
##              -0.02253801               0.27314689               0.53382312
##                 bedrooms          bath_full_count               year_built
##              -0.02185546               0.01034856               0.05244055
```

```
sd(house.2$`Sale Price`)
```

```
## [1] 226312.4
```

The Standardized beta values indicate the number of standard deviations by which the outcome will change as a result of one standard deviation change in the predictor. Since unit of all these values is Standard Deviation so easy to compare.

Based on the result of Standardized beta values , The Std Dev of House Sale Price is = 244743.4

1. square_feet_total_living - The value is 0.53, this value indicates that as living area of house increases by one std dev the house price increases by (0.53 * 226312.4) = 119945.6 dollars considering effect of all other predictors are held constant.
2. building_grade - The value is 0.27 which indicates that a building grade of house increased by 1 std dev the house price increases by 61104.35 dollars considering effect of all other predictors are held constant.
3. sq_ft_lot - The value is -0.023 (negatively related) which indicates that a increasing the lot size by 1 std dev and the house price will drop by 5205.185 dollars considering effect of all other predictors are held constant.

4. bedrooms - The value is -0.022 (negatively related) which indicates that a increasing the bedroom by 1 std dev and the house price will drop by 4978.873 dollars considering effect of all other predictors are held constant.

5. bath_full_count - The value is 0.010 which indicates that a bath room count of house increased by 1 std dev the house price increases by 2263.124 dollars considering effect of all other predictors are held constant.

6. year_built - The value is 0.052 which indicates that a build of year of house increased by 1 std dev the house price increases by 11768.24 dollars considering effect of all other predictors are held constant.

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

```
confint(multi_result)
```

```
##                                 2.5 %        97.5 %
## (Intercept)              -2.305892e+06 -1.343588e+06
## sq_ft_lot                -6.568615e-01 -1.453605e-01
## building_grade            5.763810e+04  6.581084e+04
## square_feet_total_living  1.334582e+02  1.456650e+02
## bedrooms                 -1.046477e+04 -1.600757e+03
## bath_full_count          -1.976711e+03  9.997401e+03
## year_built                5.606293e+02  1.046578e+03
```

The Cnfidence Interval of the standardized beta values are boundaries constructed such that 95% of these samples these boundaris will contain the true value of coefficients or betas. With values from CI for each predictor we can say that the betas value will be there in population if values lies between these confidence intervals.

Based on above result we can say that -

1. square_feet_total_living - It is positively correlated with Sale Price and the CI is range is positive and the interval is very small, this means the beta value of this predictor is very close to actual beta value in population.

2. building_grade - It is positively correlated with Sale Price and the CI is range is positive and the interval is very small, this means the beta value of this predictor is very close to actual beta value in population.

3. sq_ft_lot - It is negatively correlated with Sale Price and the CI is range is negative and the interval is very small and will not cross zero, this means the beta value of this predictor is very close to actual beta value in population.

4. bedrooms - It is negatively correlated with Sale Price and the CI is range is negative and the interval is very small and will not cross zero, this means the beta value of this predictor is very close to actual beta value in population.

5. bath_full_count - It is positively correlated with house Sale Price and CI range falls in positive and negative values and interval is large, this means the beta value of this predictor may cross zero and not stay close the actual bets value in population.

6. year_built - It is positively correlated with Sale Price and the CI is range is positive and the interval is very small, this means the beta value of this predictor is very close to actual beta value in population.

b vi] (simple regression model) by testing whether this change is significant by performing an analysis of variance.

```
anova(simple_result,multi_result)
```

```
## Analysis of Variance Table
##
## Model 1: ‘Sale Price‘ ~ sq_ft_lot
## Model 2: ‘Sale Price‘ ~ sq_ft_lot + building_grade + square_feet_total_living +
##     bedrooms + bath_full_count + year_built
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
## 1  11311 5.4701e+14
## 2  11306 2.4250e+14  5 3.0451e+14 2839.5 < 2.2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

Based on above result the p-value of second model is 2.2e-16 means very very small number this indicates the second model(multi_result) is significantly improved the fit of the model of data compared with first(simple_result) having F- score is 2839.5

b vii] Perform casewise diagnostics to identify outliers and/or influential cases. . . .

```
house_case <- data.frame(house.2)
house_case$residuals <- resid(multi_result)
house_case$stdz.residuals <- rstandard(multi_result)
house_case$stud.residuals <- rstudent(multi_result)
house_case$cooks.dist <- cooks.distance(multi_result)
house_case$dfbeta <- dfbeta(multi_result)
house_case$dfiit <- dffits(multi_result)
house_case$leverage <- hatvalues(multi_result)
house_case$cov.ratio <- covratio(multi_result)
select <- dplyr::select
head(house_case %>% select(residuals,stdz.residuals,stud.residuals,cooks.dist,dfbeta,dfiit,leverage,cov
```

Created new dataframe variable called house_case and stored all the casewise diagnostic result in it.

```
##    residuals stdz.residuals stud.residuals   cooks.dist dfbeta.(Intercept)
## 1  -15794.54    -0.10786000    -0.10785528 3.941455e-07       1.025606e+02
## 2  -76411.85    -0.52181597    -0.52179918 9.812222e-06       7.137722e+02
## 3  -56393.99    -0.38515526    -0.38514075 9.867775e-06      -6.235855e+01
## 4  -38698.76    -0.26431115    -0.26430028 5.359879e-06      -9.359212e+02
## 5  -12428.34    -0.08487576    -0.08487203 3.249033e-07      -7.530359e+01
## 6 -595435.26    -4.06979707    -4.07260135 4.759076e-03      -1.921174e+03
##   dfbeta.sq_ft_lot dfbeta.building_grade dfbeta.square_feet_total_living
## 1     5.764190e-05         -1.834468e+00                    1.789255e-03
## 2     2.971821e-04         -7.913746e+00                    6.881558e-03
## 3     2.640822e-04          2.981295e+00                   -1.093670e-02
## 4     2.413633e-04         -4.802518e+00                    2.119585e-03
## 5     1.347590e-05          6.855240e-01                    1.402696e-04
## 6     3.274988e-03          3.111689e+02                   -4.916126e-01
##   dfbeta.bedrooms dfbeta.bath_full_count dfbeta.year_built        dfiit
## 1   -1.744659e+00           3.162716e-01     -4.436362e-02 -0.001660957
## 2   -8.191811e+00           3.500824e+00     -3.277826e-01 -0.008287408
## 3   -2.613085e+00           1.863061e+01      1.617295e-02 -0.008310789
## 4    1.962138e+00           2.940314e+00      4.771241e-01 -0.006125036
## 5    7.059119e-02           1.530226e+00      3.259384e-02 -0.001508020
## 6    1.209965e+02           7.444876e+01     -3.302702e-02 -0.182645706
##       leverage cov.ratio
## 1 0.0002370996  1.000849
## 2 0.0002521861  1.000703
## 3 0.0004654183  1.000993
## 4 0.0005367708  1.001113
## 5 0.0003156078  1.000931
## 6 0.0020072549  0.992395
```

b viii] Calculate the standardized residuals using the appropriate command. . . .

```
house_case$large.residual <- house_case$stdz.residuals > 2 | house_case$stdz.residuals < -2
```

created new column called - large.residual to store in house__cases dataframe

b vix] Use the appropriate function to show the sum of large residuals.

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

The Sum of large residuals is

```
## [1] 532
```

b vx] Which specific variables have large residuals (only cases that evaluate as TRUE)?

The predictors those are having large residuals We have 4% cases which are outside the limit

```
head(house_case %>% filter(large.residual) %>%  select(Sale.Price, sq_ft_lot , building_grade , square_
```

```
##   Sale.Price sq_ft_lot building_grade square_feet_total_living bedrooms
## 1     184667      7280              7                     4160        4
## 2    1392000     17291              9                     3740        4
## 3    1053649      8517              9                     2680        2
## 4     148000      3430              9                     1930        3
## 5    1900000     37017             11                     6610        4
## 6    1080135      7694              9                     2700        3
##   bath_full_count year_built
## 1               2       2005
## 2               3       1998
## 3               2       2005
## 4               2       2003
## 5               3       1990
## 6               2       2006
```

b vxi] calculating the leverage, cooks distance, and covariance ratios.

Cook's distance - it is a measure of the overall influence of a case on the model, value greater than 1 cause for concern leverage = (k+1)/n , where k = number of predictor and n = number of participants and hat value should be between 0- 1

```
head(house_case %>% filter(large.residual) %>%  select(cooks.dist,leverage,cov.ratio))
```

```
##     cooks.dist     leverage cov.ratio
## 1 0.0047590756 0.0020072549 0.9923950
## 2 0.0012398837 0.0006086606 0.9924280
## 3 0.0005144412 0.0006477815 0.9978290
## 4 0.0006602211 0.0004883802 0.9952604
## 5 0.0076641828 0.0037610773 0.9955931
## 6 0.0002200434 0.0002364899 0.9968279
```

```
head(house_case %>% filter(large.residual) %>%  select(cooks.dist,leverage,cov.ratio) %>% filter(cooks.
```

From the above result, calculated the cooks distance > 1 and found none of the case having cooks distance > 1 that means non of the cases having an undue influence on the model.

```
## [1] cooks.dist leverage   cov.ratio
## <0 rows> (or 0-length row.names)
```

```
k <- 7
n <- nrow(house_case)
leverage_val = round((k+1)/n,7)
house_case %>% filter(large.residual) %>%  select(cooks.dist,leverage,cov.ratio) %>% filter(leverage >=
```

The average leverage can be calculated as (k+1)/n

```
## [1] 64
```

based on above average leverage we can see there are 64 values above three time average levarage value which is 0.5% of the total cases.

```
k <- 7
n <- nrow(house_case)
cov_grter_1 <- 1 + (3*(k+1)/n)
cov_less_1 <- 1 - (3*(k+1)/n)
# covariance ratio greater than 1 limit
print(cov_grter_1)
```

Calculated covariance ratio as

```
## [1] 1.002121
```

```
# covariance ratio less than 1 limit
print(cov_less_1)
```

```
## [1] 0.9978785
```

```
house_case %>% filter(large.residual) %>%  select(cooks.dist,leverage,cov.ratio) %>% filter(cov.ratio >=
```

```
## [1] 164
```

Based on above result of covariance ratio we can say that 1.44% of total cases are outside these limits. But most of them are close to that but not too big or less than these limits. We can say that we do have fairly reliable model that has not been unduly influenced by any large subsets of cases.

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

```
durbinWatsonTest(multi_result)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.3637946      1.272275       0
##  Alternative hypothesis: rho != 0
```

Based on this result the value of D-W Statistic is 1.272275 which between 1 and 3. We can say that the condition of assumptions of independence is met.

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

```
vif(multi_result)
```

to test this assumption we are calcualting Variance Inflation Factor (VIF), VIF tolerance and average of VIF

```
##                 sq_ft_lot            building_grade square_feet_total_living
##                  1.451796                  2.298916                 3.831560
##                  bedrooms           bath_full_count               year_built
##                  1.812421                  1.678008                 1.767429
```

```
1/vif(multi_result)
```

```
##                 sq_ft_lot            building_grade square_feet_total_living
##                 0.6888022                 0.4349877                0.2609903
##                  bedrooms           bath_full_count               year_built
##                 0.5517482                 0.5959446                0.5657935
```

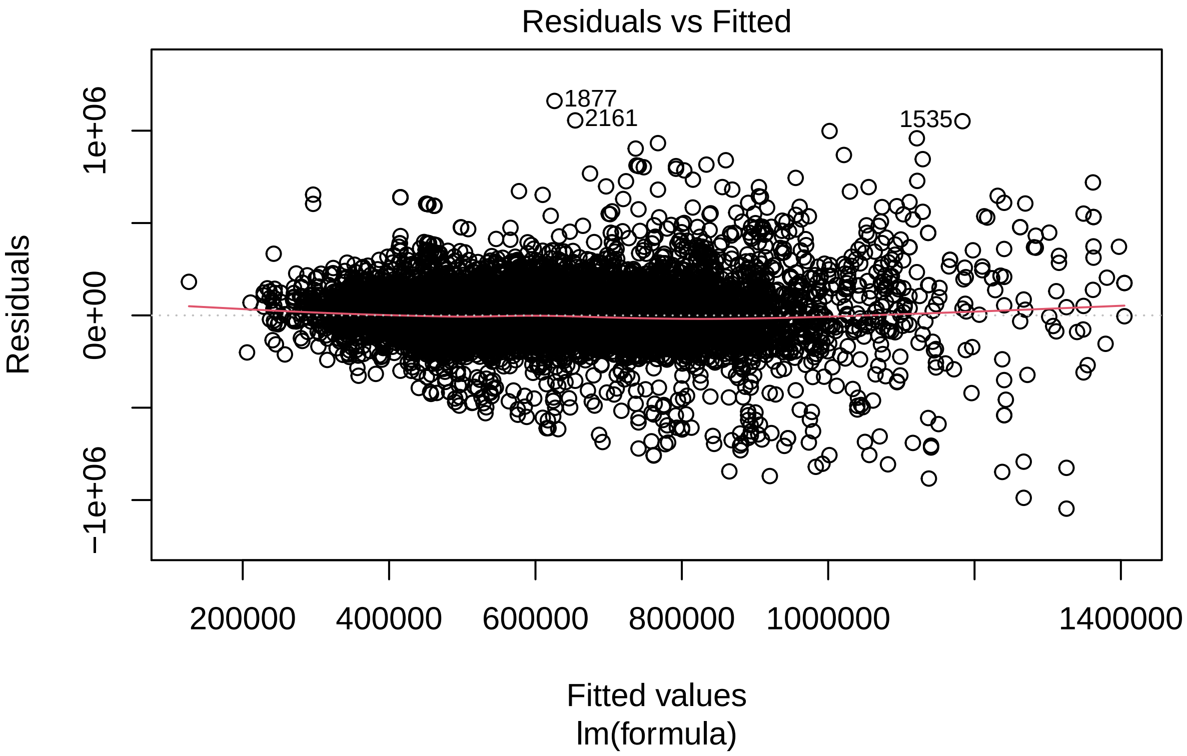
```
mean(vif(multi_result))
```

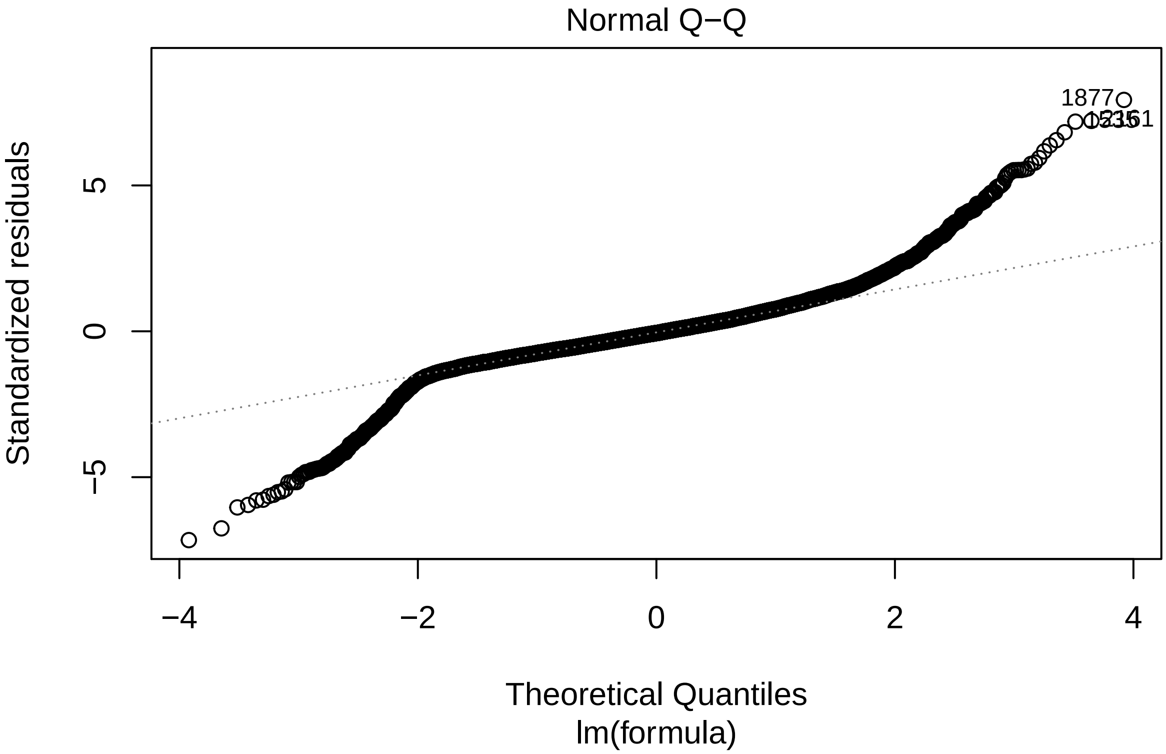
```
## [1] 2.140022
```

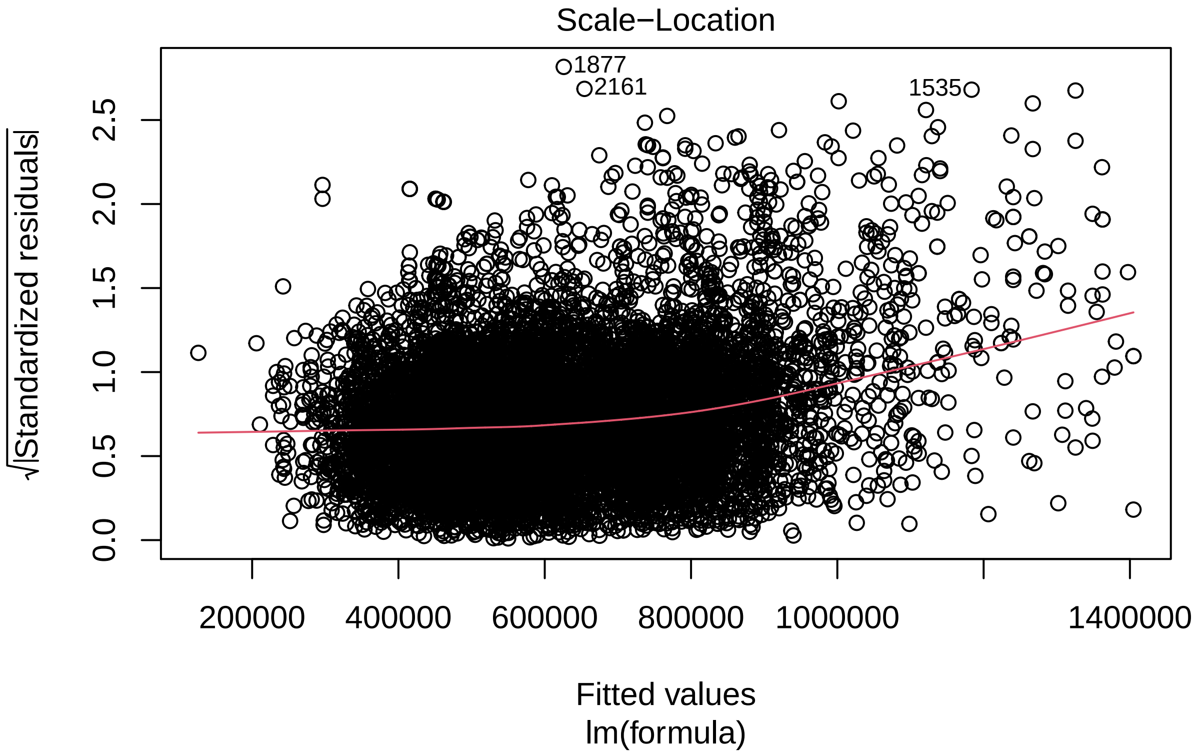
Based on above result - 1. The VIF value for the predictors are less than 10 2. Tolerance VIF for all the predictors are above 0.2 3. Average VIF is greater than 1 which is 2.14, which tells that regression may be biased.

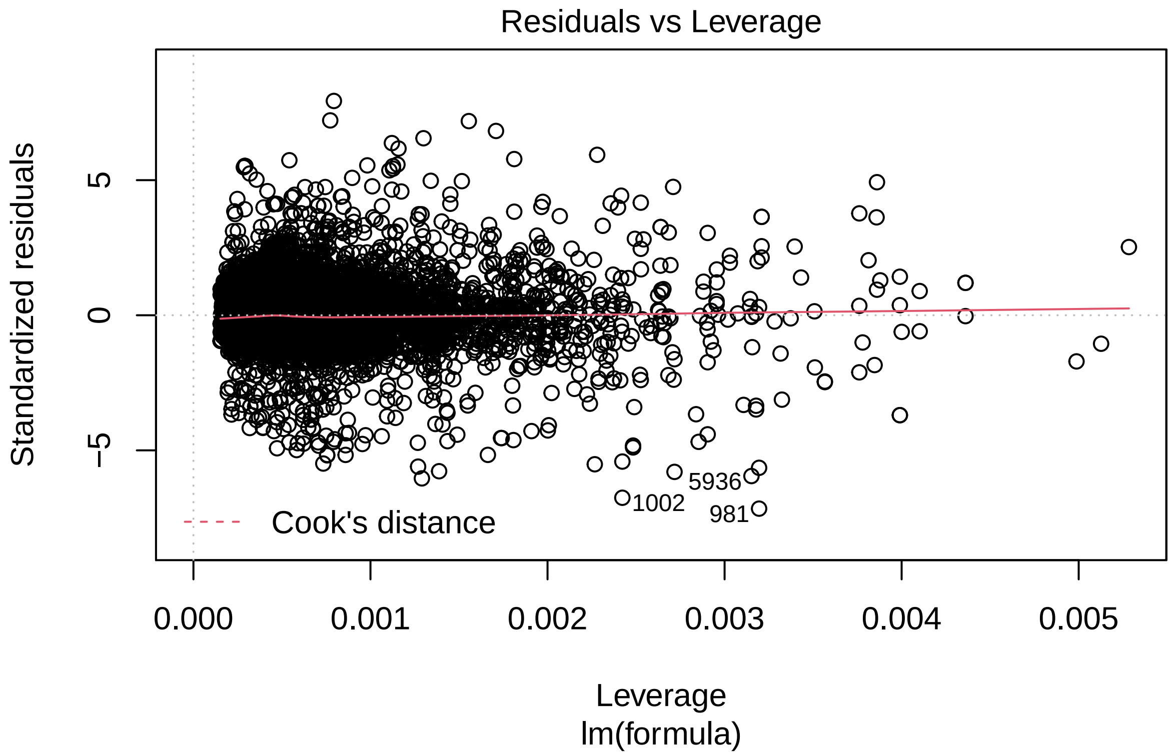
b vxiv] Visually check the assumptions related to the residuals using the plot() and hist() functions. Summarize what each graph is informing you of and if any anomalies are present.

```
plot(multi_result)
```

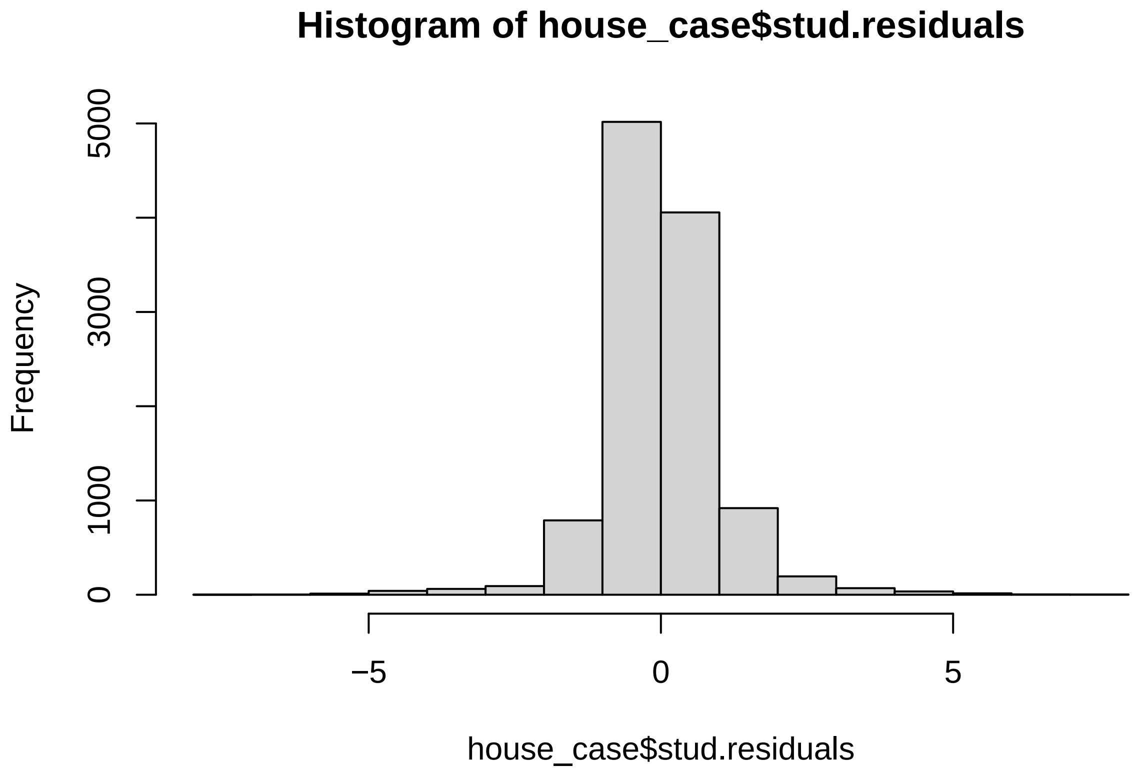

Residuals vs Fitted
Residuals
1e+06
0e+00
−1e+06
200000
400000
600000
800000
1000000
1400000
Fitted values
lm(formula)
1877
2161
1535


Normal Q−Q
Standardized residuals
5
0
−5
1877
15351
−4
−2
0
2
4
Theoretical Quantiles
lm(formula)


Scale−Location
1877
2161
1535
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
200000
400000
600000
800000
1000000
1400000
Fitted values
lm(formula)



```
hist(house_case$stud.residuals)
```



Based on the above Plots -

1. Residuals vs Fitted - This plot shows the randomly distributed points evenly dispersed around the zero residual line. The residuals in the housing data model shows a fairly random pattern which indicates that assumptions of linearity, randomness and homoscedasticity have been met.
2. Normal Q-Q - This plot shows the deviation from normality, the most of the points in middle of the graph lies on the straight line indicating normality, the data points at the end shows deviation from line mean skewness in data.
3. Scale Location - The dots are dispersed randomly around the standardized residual line, which indicates that assumptions of linearity, randomness and homoscedasticity have been met.
4. Residuals vs Leverage - This plot is also shows tha randomly distributed points, which indicates assumptions of linearity, randomness and homoscedasticity have been met.
5. Histogram of Residuals - Histograms of residual shows the normally distributed residuals, which is good.

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

Based on the above all the analysis done so far we can say that the regression model used to prdict the house price based on predictors is unbiased. The assumptions made about these models are validates and are met which tells that the an average, regression model we ran on the sample can be accurately applied to the population. This does mean that even all the assumptions were met , it is possible that a model obtained from this sample may or may not be the same as the population model but likelihood of them being the same is increased.