assignment_07_MunjewarSheetal-03

Sheetal M

2023-02-12

Contents

Install and Load required packages:

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8     v purrr   1.0.0
## v tidyr   1.2.1     v stringr 1.5.0
## v readr   2.1.3     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::extract()   masks magrittr::extract()
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x purrr::set_names() masks magrittr::set_names()
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
##
##
## Attaching package: 'scales'
##
##
## The following object is masked from 'package:purrr':
##
##     discard
##
##
## The following object is masked from 'package:readr':
##
##     col_factor
##
##
```

```
##
## Attaching package: 'reshape'
##
##
## The following objects are masked from 'package:tidyr':
##
##     expand, smiths
##
##
## The following object is masked from 'package:dplyr':
##
##     rename
```

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

setwd("E:\Data_Science_DSC510\DSC520-Statistics\dsc520")

```r
library("readxl")
# xls files
housing.data <- read_excel("week-7-housing.xlsx")
str(housing.data)
```

```
## 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 NE" "3303
##  $ 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 ...
```

Explain any transformations or modifications you made to the dataset.

```r
# Calculate number for NA values.
sum(is.na(housing.data))
```

```
## [1] 16646
```

```
colnames(housing.data)[1] <- "sale_date"
colnames(housing.data)[2] <- "sale_price"

str(housing.data)
```

```
## 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 NE" "3303
##  $ 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 ...
```

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.

```
# Find co-relation between variables using cor()
cor(housing.data$sale_price,housing.data$square_feet_total_living)
```

```
## [1] 0.4545876
```

```
cor(housing.data$sale_price,housing.data$bedrooms)
```

```
## [1] 0.2254675
```

```
cor(housing.data$sale_price,housing.data$bath_full_count)
```

```
## [1] 0.284849
```

```r
cor(housing.data$sale_price,housing.data$building_grade)
```

```
## [1] 0.3912291
```

```r
cor(housing.data$sale_price,housing.data$sq_ft_lot)
```

```
## [1] 0.1198122
```

```r
cor(housing.data$sale_price,housing.data$year_built)
```

```
## [1] 0.2426713
```

```r
#Model_01 <- lm(sale_price ~ bath_full_count, data = housing.data)
#Model_02 <- lm(sale_price ~ building_grade + square_feet_total_living + year_built , data = housing.data)

Model_01 <- lm(sale_price ~ bath_full_count, data = housing.data)
Model_02 <- lm(sale_price ~ bath_full_count + building_grade + square_feet_total_living , data = housing.data)

# Check for NULL/NA and observation row count.
# is.null(housing.data$sale_price)
# is.na(housing.data$square_feet_total_living)
# nrow(housing_df)
```

IExecute 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?

```r
#Execute a summary() function on two variables defined in the previous step to compare the model results. What a

summary(Model_01)
```

```
##
## Call:
## lm(formula = sale_price ~ bath_full_count, data = housing.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4143301  -166512   -53732    70583  3880583
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       342422      10044   34.09   <2e-16 ***
## bath_full_count   176995       5252   33.70   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 387600 on 12863 degrees of freedom
## Multiple R-squared:  0.08114,    Adjusted R-squared:  0.08107
## F-statistic:  1136 on 1 and 12863 DF,  p-value: < 2.2e-16
```

```r
# plot(Model_01)
summary(Model_02)
```

```
##
## Call:
## lm(formula = sale_price ~ bath_full_count + building_grade +
##     square_feet_total_living, data = housing.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1678543  -116217   -43217    39534  3875671
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -91100.790  28115.007  -3.240   0.0012 **
## bath_full_count           35214.013   5721.583   6.155 7.75e-10 ***
## building_grade            40205.127   4372.001   9.196  < 2e-16 ***
## square_feet_total_living    140.658      5.011  28.072  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 358300 on 12861 degrees of freedom
## Multiple R-squared:  0.2152, Adjusted R-squared:  0.215
## F-statistic:  1175 on 3 and 12861 DF,  p-value: < 2.2e-16
```

```r
# plot(Model_02)

#  Reference:-
#  Interpret the R and R2 square result after watching video :
#  https://www.youtube.com/watch?v=bMccdk8EdGo
#  R2 and P-Values
#  https://www.youtube.com/watch?v=xxFYro8QuXA
```

R2 and Adjusted R2 for Model_01 are: 0.08114 and 0.08107

R2 and Adjusted R2 for Model_02 are: 0.2152 and 0.215

- We are seeing R2 variance improvement with multiple predictors in model.
- Model_01 explain 8% of variances sales price.
- Model_02 explain 22% of variance of data.
- Consider R2 variance of data, Model-02 must be a good fit.

Standardized Betas - 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?

(Intercept) -4.713e+06 - Y-intercept. building_grade 3.243e+04 - Beta 1 square_feet_total_living 1.464e+02 - Beta 2 year_built 23706e+03 - Beta 3

- Beta 1 : 35214.013 - bath_full_count

- Beta 2 : 40205.127 - building_grade

- Beta 3 : 140.658 - square_feet_total_living
- Beta 1 indicates change in unit of bath_full_count will cost $35214.013 in sale price.
- Beta 2 indicates change in unit of building_grade will lift sale price by $40205.127.
- Beta 3 indicates change in unit of square_feet_total_living will change sale price by $140.658.

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

```r
# reference -  https://www.statology.org/confint-r/
confint(Model_01, level = 0.95)
```

```
##                    2.5 %   97.5 %
## (Intercept)     322734.3 362110.6
## bath_full_count 166700.6 187288.8
```

```r
confint(Model_02, level = 0.95)
```

```
##                                2.5 %      97.5 %
## (Intercept)             -146210.3779 -35991.2023
## bath_full_count           23998.8596  46429.1657
## building_grade            31635.3556  48774.8984
## square_feet_total_living    130.8361    150.4792
```

confidence interval - Model_01

- 95% C.I. for year_built = [166700.6,187288.8]
- For model-1 with confidence level 95%, Sale price mean for bath_full_count variable lies 166700.6 and 187288.8.

confidence interval - Model_02

- 95% C.I. for bath_full_count = [23998.8596 ,46429.1657]
- 95% C.I. for building_grade = [31635.3556 ,48774.8984]
- 95% C.I. for square_feet_total_living = [130.8361 ,150.4792]
- For model_2 with confidence level 95%, Sale price mean for bath_full_count variable lies between [23998.8596,46429.1657]
- For model_2 with confidence level 95%, Sale price mean for building_grade variable lies between [31635.3556,48774.8984]
- For model_2 with confidence level 95%, Sale price mean for square_feet_total_living variable lies between [130.8361,150.4792]

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.

- Model_01 explain 8% of variances of the data.
- Model_02 explain 22% of variance of data.
- Model_02 is much improved in comparison with Model_01.

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

```
housing.data$residuals <- resid(Model_02)
housing.data$std_residuals <- rstandard(Model_02)
housing.data$stu_residuals <- rstudent(Model_02)
housing.data$cooks_distance <- cooks.distance(Model_02)
housing.data$dfbeta <- dfbeta(Model_02)
housing.data$dffit <- dffits(Model_02)
housing.data$leverage <- hatvalues(Model_02)
housing.data$covariance_ratios <- covratio(Model_02)

head(housing.data$residuals)
```

```
##          1          2          3          4          5          6
##  -38421.34  -96277.37  -82875.89  -73619.61  -58196.11 -661231.89
```

```
housing.data
```

```
## # A tibble: 12,865 x 32
##    sale_date           sale_price sale_r~1 sale_~2 sale_~3 sitet~4 addr_~5  zip5
##    <dttm>                   <dbl>    <dbl>   <dbl> <chr>   <chr>   <chr>   <dbl>
##  1 2006-01-03 00:00:00     698000        1       3 <NA>    R1      17021 ~ 98052
##  2 2006-01-03 00:00:00     649990        1       3 <NA>    R1      11927 ~ 98052
##  3 2006-01-03 00:00:00     572500        1       3 <NA>    R1      13315 ~ 98052
##  4 2006-01-03 00:00:00     420000        1       3 <NA>    R1      3303 1~ 98052
##  5 2006-01-03 00:00:00     369900        1       3 15      R1      16126 ~ 98052
##  6 2006-01-03 00:00:00     184667        1      15 18 51   R1      8101 2~ 98053
##  7 2006-01-04 00:00:00    1050000        1       3 <NA>    R1      21634 ~ 98053
##  8 2006-01-04 00:00:00     875000        1       3 <NA>    R1      21404 ~ 98053
##  9 2006-01-04 00:00:00     660000        1       3 <NA>    R1      7525 2~ 98053
## 10 2006-01-04 00:00:00     650000        1       3 <NA>    R1      17703 ~ 98052
## # ... with 12,855 more rows, 24 more variables: 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>, residuals <dbl>, std_residuals <dbl>,
## #   stu_residuals <dbl>, cooks_distance <dbl>, dfbeta <dbl[,4]>, ...
```

```
write.table(housing.data, "Housing_updated_data.dat", sep = "\t", row.names = FALSE)
```

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

```
# housing.data$std_residuals > 2 | housing.data$std_residuals < -2
housing.data$large_residual <- housing.data$std_residuals > 2 | housing.data$std_residuals < -2
```

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

```r
# round(housing.data, digits = 10)
sum(housing.data$large_residual)
```

```
## [1] 314
```

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

```r
housing.data[ housing.data$large_residual,c("bath_full_count","building_grade","square_feet_total_living","year_
```

```
## # A tibble: 314 x 5
##    bath_full_count building_grade square_feet_total_living year_built std_resi~1
##              <dbl>          <dbl>                    <dbl>      <dbl>      <dbl>
##  1               4             10                     4920       2007      -2.45
##  2               1              6                      660       1955       3.10
##  3               4             11                     5800       2008      -2.56
##  4               2              9                     3360       2005       2.16
##  5               1              6                      900       1918       3.18
##  6               2              9                     4710       2014      -2.35
##  7              23             11                     5060       2016      -4.73
##  8               1             10                     6880       2008      -3.11
##  9               2             11                     4490       2008      -2.10
## 10               2             11                     5140       2008      -2.67
## # ... with 304 more rows, and abbreviated variable name 1: std_residuals
```

- Total observations : 12865
- Total Larger residuals reported : 314
- Percent Residual out of limits : 314/12865*100 = 2.44 (well within expected +/- 2.5 limits)

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

```r
housing.data[ housing.data$large_residual,c("cooks_distance","leverage","covariance_ratios") ]
```

```
## # A tibble: 314 x 3
##    cooks_distance leverage covariance_ratios
##             <dbl>    <dbl>             <dbl>
##  1       0.00157  0.00104              0.999
##  2       0.00104  0.000431             0.998
##  3       0.00201  0.00122              0.999
##  4       0.000157 0.000134             0.999
##  5       0.00104  0.000413             0.998
##  6       0.000902 0.000654             0.999
##  7       0.660    0.105                1.11
##  8       0.00772  0.00317              1.00
##  9       0.000730 0.000660             1.00
## 10       0.00145  0.000812             0.999
## # ... with 304 more rows
```

- Total observations : 12865

- Average leverage : (k + 1/n) = (3+1)/12865 = 0.000310 (k is number of predictors in a model.)
- Twice/Thrice of 0.000310 = 0.00093 (0.000310*3) - Leverage finding - Most of the cases are within the boundaries (< 0.00093), three times of the average.
- Even covariance ration (+1/-1) for all large residuals are upper side of 1 and there are few above one on the border.

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

durbinWatsonTest(Model_02)

lag Autocorrelation D-W Statistic p-value

1 0.7365797 0.5268369 0

Alternative hypothesis: rho != 0

- Conservative rule suggest values below 1 and above 3, will raise alarm. In our case D-W stats reported 0.52 which is less than 1 raise the alarm and p-value = 0 is again a concern of no co-relationship.

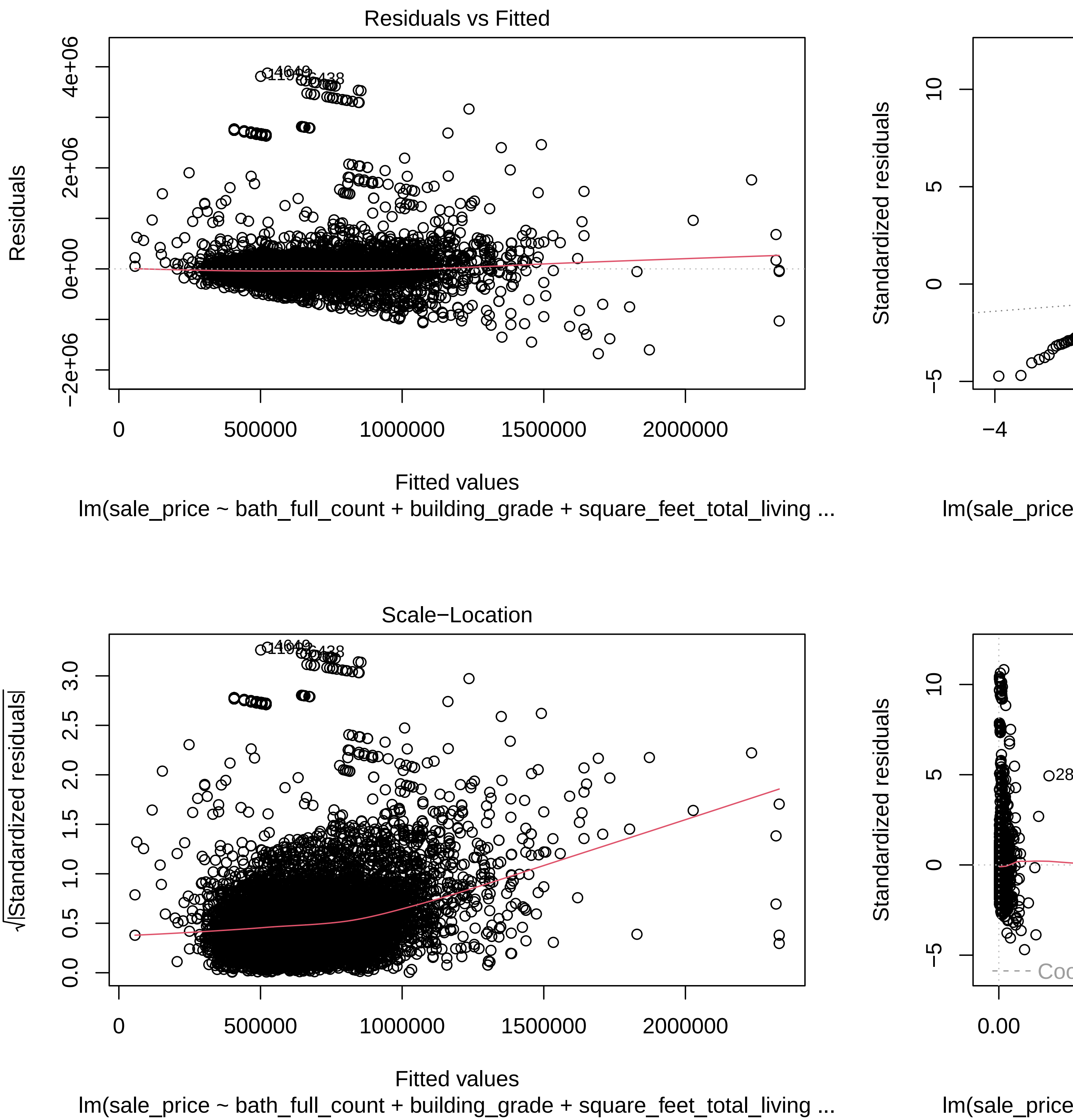
Perform the necessary calculations to assess the assumption of no multicollinearity and state if the condition is met or not.

```
# vif(Model_02)
# 1/vif(Model_02)
# mean(vif(Model_02))

#> vif(Model_02)
#          bath_full_count           building_grade square_feet_total_living
#                 1.389415                 2.286707                 2.464910
#> 1/vif(Model_02)
#          bath_full_count           building_grade square_feet_total_living
#                0.7197276                0.4373100                0.4056944
#> mean(vif(Model_02))
#[1] 2.04701
```

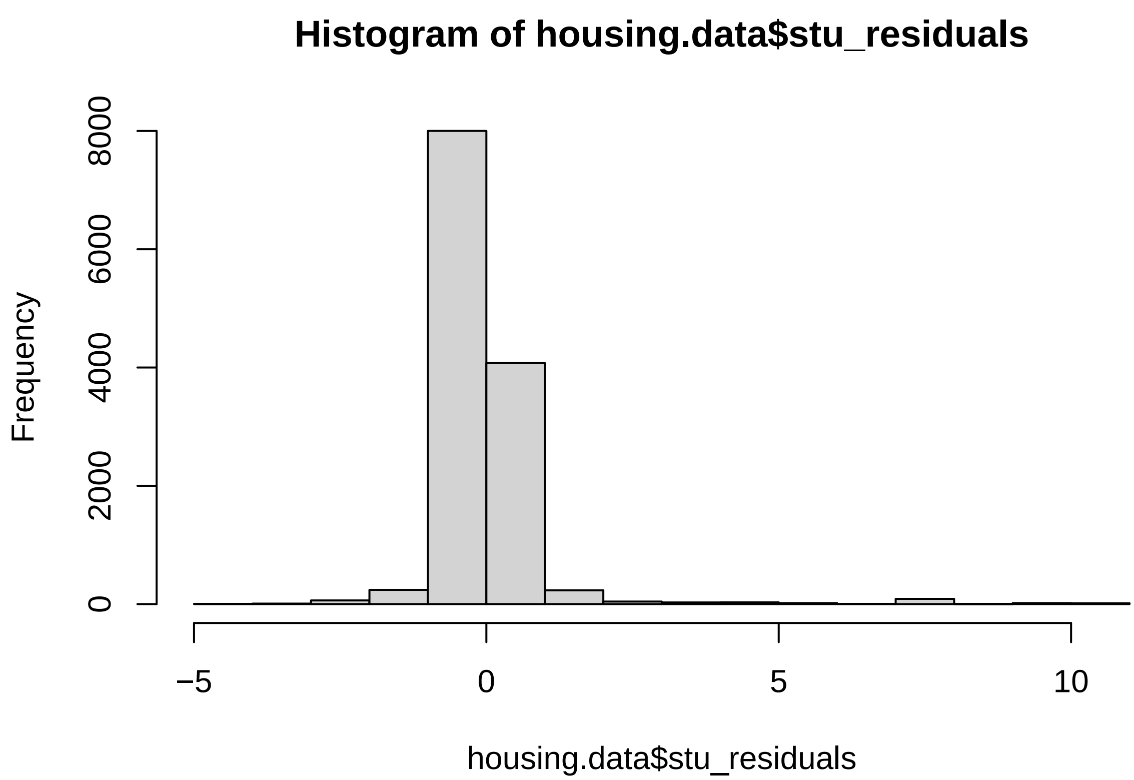
- multicollinearity - VIF values are all below 10, and tolerance stats all well above 0.2 and Average VIF value i.e 2 is above 1 is a concern to conclude no collinearity within the data

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.

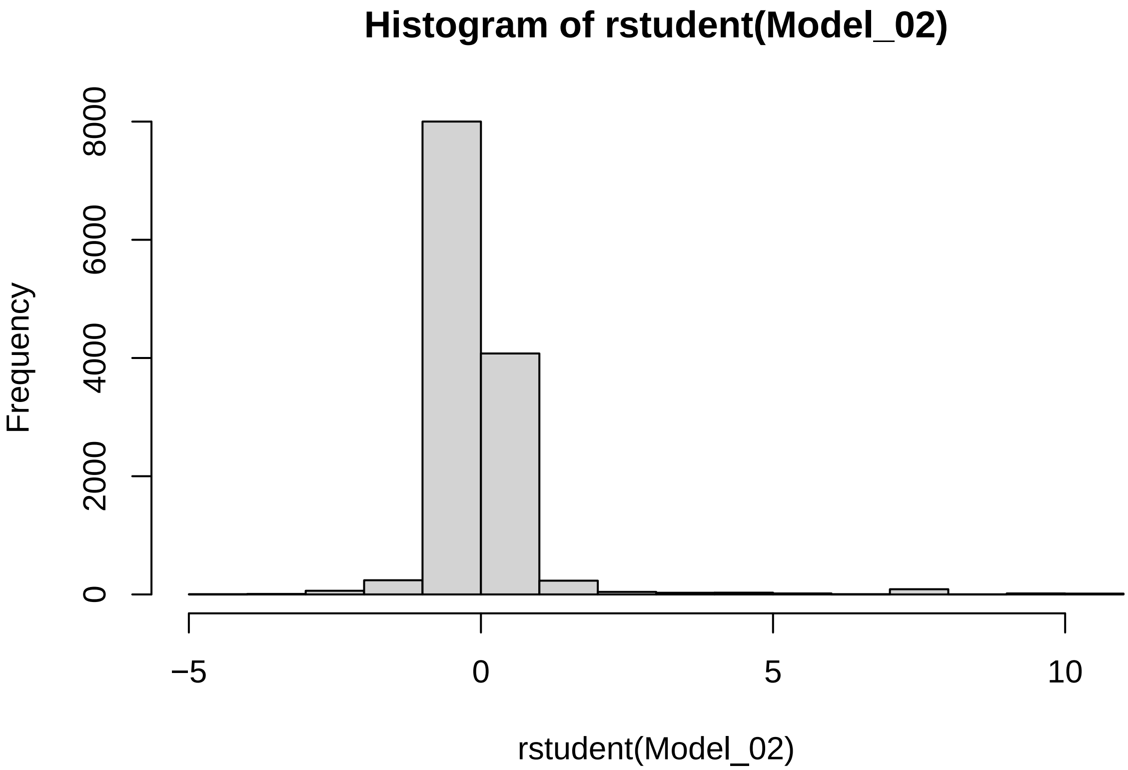
```
plot(Model_02)
```


Residuals vs Fitted
Residuals
Fitted values
lm(sale_price ~ bath_full_count + building_grade + square_feet_total_living ...
Standardized residuals
lm(sale_price
Scale−Location
√|Standardized residuals|
Fitted values
lm(sale_price ~ bath_full_count + building_grade + square_feet_total_living ...
Standardized residuals
lm(sale_price

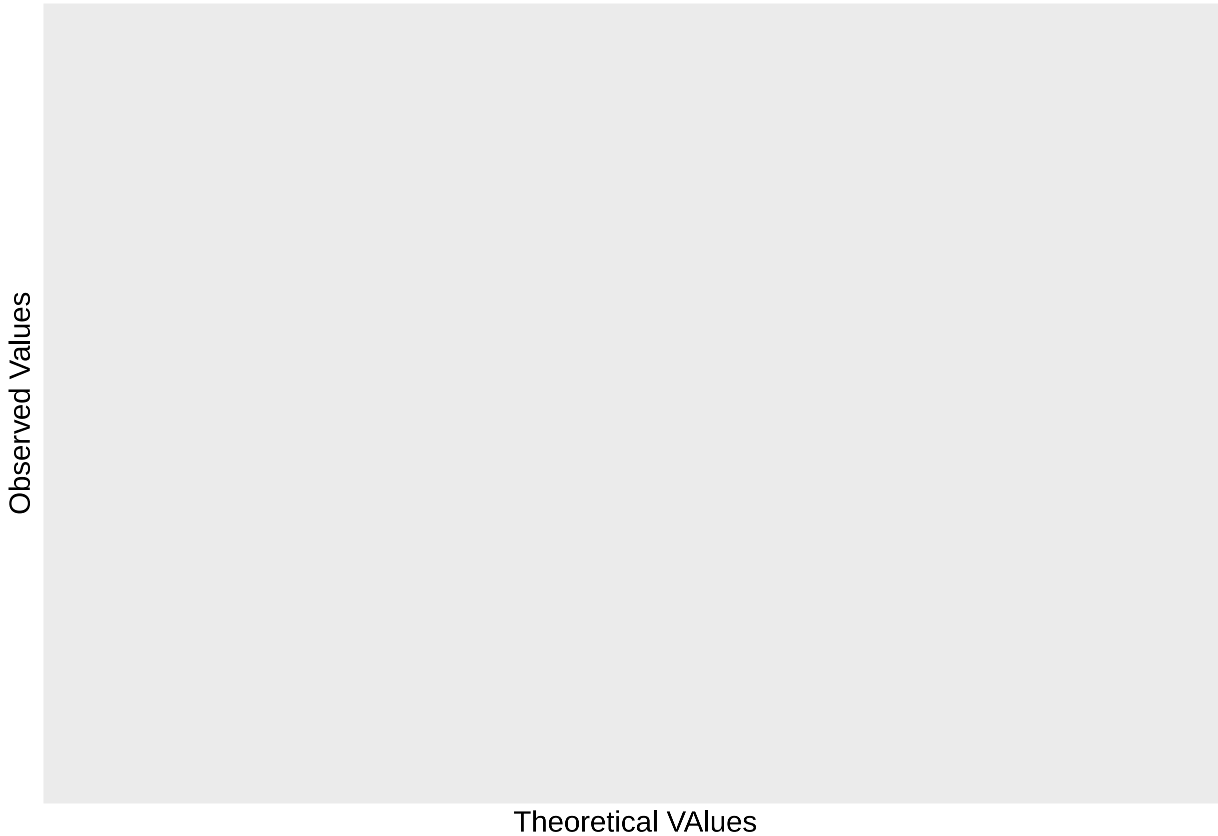
```r
hist(housing.data$stu_residuals)
```




```r
hist(rstudent(Model_02))
```



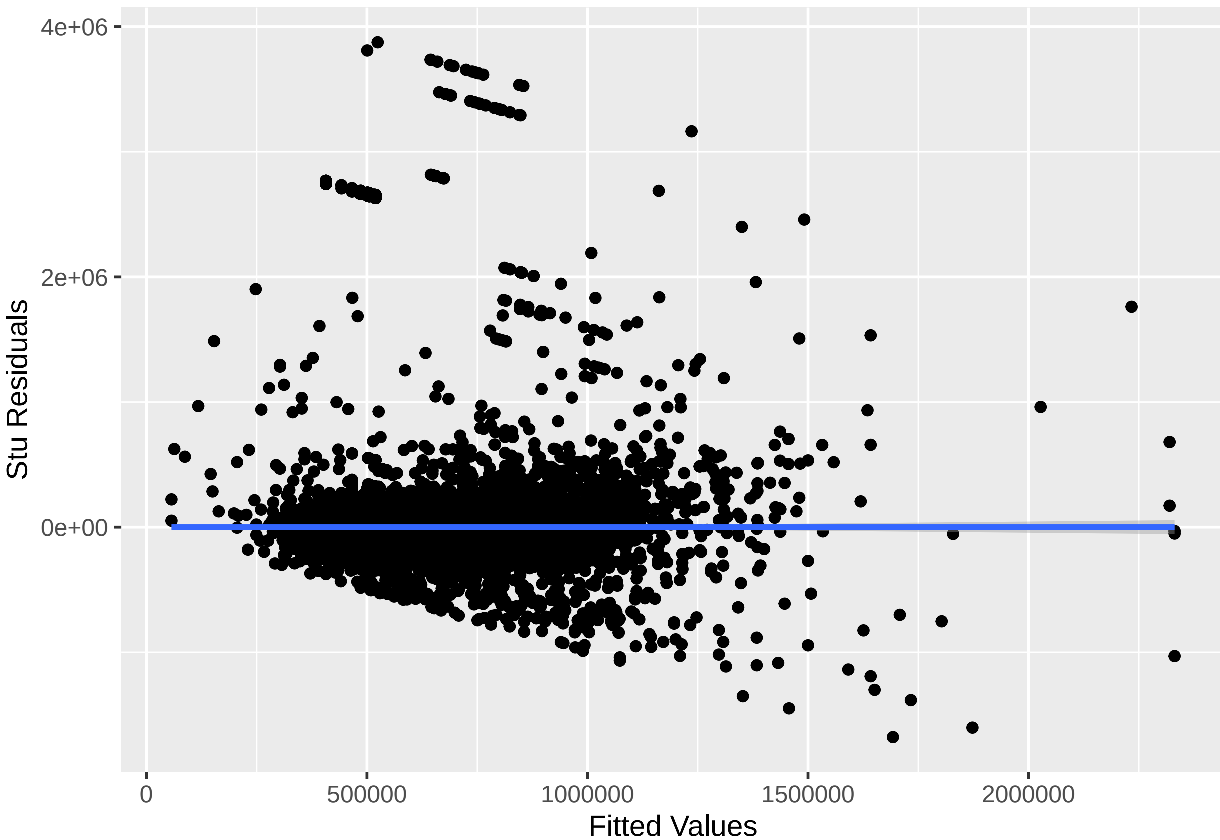

```
Model_02$fitted <- Model_02$fitted.values
#Model_02$df.residual

ggplot(sample = housing.data$std_residuals, stat = "qq") + labs(x = "Theoretical VAlues", y = "Observed Values")
```




```
ggplot(Model_02, aes(Model_02$fitted.values, Model_02$residuals)) + geom_point() + geom_smooth(method = "lm", co
```

```
## Warning in geom_smooth(method = "lm", colours = "Blue"): Ignoring unknown
## parameters: ‘colours‘
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



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 VIF mean value[2], Model can be consider biased.