```r
# Assignment: ASSIGNMENT 5.2
# Name: Anjale, Jiteshwar
# Date: 2021-05-14
#Analysis of housing data

## Load the readxl package
library(readxl)

## Warning: package 'readxl' was built under R version 4.0.5

## Load the plyr package
library(dplyr)

## Warning: package 'dplyr' was built under R version 4.0.5

##
## 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

## Load the purrr package
library(purrr)

## Warning: package 'purrr' was built under R version 4.0.5

## Load the QuantPsyc  package
library(QuantPsyc)

## Warning: package 'QuantPsyc' was built under R version 4.0.5

## Loading required package: boot

## Loading required package: MASS

##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select

##
## Attaching package: 'QuantPsyc'

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

```r
## Load the car package
library(car)

## Warning: package 'car' was built under R version 4.0.5

## Loading required package: carData

##
## Attaching package: 'car'

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

## The following object is masked from 'package:purrr':
##
##     some

## The following object is masked from 'package:dplyr':
##
##     recode

## Load the tidyverse package
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.5

## -- Attaching packages --------------------------------------- tidyverse 1.
3.1 --

## v ggplot2 3.3.3     v readr   1.4.0
## v tibble  3.1.0     v stringr 1.4.0
## v tidyr   1.1.3     v forcats 0.5.1

## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'readr' was built under R version 4.0.5

## Warning: package 'stringr' was built under R version 4.0.5

## Warning: package 'forcats' was built under R version 4.0.5

## -- Conflicts ------------------------------------------ tidyverse_conflict
s() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x car::recode()   masks dplyr::recode()
## x MASS::select()  masks dplyr::select()
## x car::some()     masks purrr::some()

## Load the ggplot2 package
library(ggplot2)
```

```
library(lmtest)

## Warning: package 'lmtest' was built under R version 4.0.5

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.0.5

##
## Attaching package: 'zoo'

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

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

# 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 necess
ary transformations. If not, you will want to take some time looking at the d
ata with all your new skills and identifying if you have any clean up that ne
eds to happen.

## Set the working directory to the root of your DSC 520 directory
setwd('C:/Users/anjal/OneDrive/Desktop/MS/DSC520/dsc520')

## Load the `data/acs-14-1yr-s0201.csv` to
housing_df <- read_excel("C:/Users/anjal/OneDrive/Desktop/MS/DSC520/dsc520/da
ta/week-6-housing.xlsx")

str(housing_df)

## 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 369
900 ...
##  $ 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 178T
H PL NE" "13315 174TH AVE NE" "3303 178TH AVE NE" ...
##  $ 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 3
960 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 ...
```

```
head(housing_df)
```

```
## # A tibble: 6 x 24
##   `Sale Date`         `Sale Price` sale_reason sale_instrument sale_warnin
g
##   <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>
```

```
glimpse(housing_df)
```

```
## Rows: 12,865
## Columns: 24
## $ `Sale Date`              <dttm> 2006-01-03, 2006-01-03, 2006-01-03, 2006
-01-~
## $ `Sale Price`             <dbl> 698000, 649990, 572500, 420000, 369900, 1
8466~
## $ sale_reason              <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, ~
## $ sale_instrument          <dbl> 3, 3, 3, 3, 3, 15, 3, 3, 3, 3, 3, 3, 3, 3
, 3,~
```

```
## $ sale_warning             <chr> NA, NA, NA, NA, "15", "18 51", NA, NA, NA
, NA~
## $ sitetype                 <chr> "R1", "R1", "R1", "R1", "R1", "R1", "R1",
"R1~
## $ addr_full                <chr> "17021 NE 113TH CT", "11927 178TH PL NE",
"13~
## $ zip5                     <dbl> 98052, 98052, 98052, 98052, 98052, 98053,
980~
## $ ctyname                  <chr> "REDMOND", "REDMOND", NA, "REDMOND", "RED
MOND~
## $ postalctyn               <chr> "REDMOND", "REDMOND", "REDMOND", "REDMOND
", "~
## $ lon                      <dbl> -122.1124, -122.1022, -122.1085, -122.103
7, -~
## $ lat                      <dbl> 47.70139, 47.70731, 47.71986, 47.63914, 4
7.69~
## $ building_grade           <dbl> 9, 9, 8, 8, 7, 7, 10, 10, 9, 8, 9, 8, 8,
9, 1~
## $ square_feet_total_living <dbl> 2810, 2880, 2770, 1620, 1440, 4160, 3960,
372~
## $ bedrooms                 <dbl> 4, 4, 4, 3, 3, 4, 5, 4, 4, 4, 3, 3, 4, 3,
3, ~
## $ bath_full_count          <dbl> 2, 2, 1, 1, 1, 2, 3, 2, 2, 1, 2, 2, 1, 2,
2, ~
## $ bath_half_count          <dbl> 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0,
1, ~
## $ bath_3qtr_count          <dbl> 0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0,
0, ~
## $ year_built               <dbl> 2003, 2006, 1987, 1968, 1980, 2005, 1993,
198~
## $ year_renovated           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, ~
## $ current_zoning           <chr> "R4", "R4", "R6", "R4", "R6", "URPSO", "R
A5",~
## $ sq_ft_lot                <dbl> 6635, 5570, 8444, 9600, 7526, 7280, 97574
, 30~
## $ prop_type                <chr> "R", "R", "R", "R", "R", "R", "R", "R", "
R", ~
## $ present_use              <dbl> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, ~
```

```
sum(is.na(housing_df$ctyname))
```

```
## [1] 6078
```

```
apply(housing_df, 2, function(x) any(is.na(x)))
```

```
##                Sale Date                Sale Price              sale_reason
##                    FALSE                     FALSE                    FALSE
##          sale_instrument              sale_warning                 sitetype
##                    FALSE                      TRUE                    FALSE
```

```
##                addr_full                      zip5                    ctyname
##                    FALSE                     FALSE                       TRUE
##               postalctyn                       lon                        lat
##                    FALSE                     FALSE                      FALSE
##           building_grade square_feet_total_living                   bedrooms
##                    FALSE                     FALSE                      FALSE
##          bath_full_count           bath_half_count            bath_3qtr_count
##                    FALSE                     FALSE                      FALSE
##               year_built            year_renovated             current_zoning
##                    FALSE                     FALSE                      FALSE
##                sq_ft_lot                 prop_type                present_use
##                    FALSE                     FALSE                      FALSE

#By looking at the data, I can see that there is missing data for sale_warnin
g and ctyname

# b. Complete the following:
# i. Explain any transformations or modifications you made to the dataset

#Rename the'Sale Date` and`Sale Price`
colnames(housing_df)[1] <- "Sale_Date"
colnames(housing_df)[2] <- "Sale_Price"

library(magrittr)

##
## Attaching package: 'magrittr'

## The following object is masked from 'package:tidyr':
##
##     extract

## The following object is masked from 'package:purrr':
##
##     set_names

housing_df %<>%
  mutate("year_of_sale"=substr(housing_df$Sale_Date,1,4))
str(housing_df)

## tibble [12,865 x 25] (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 369
900 ...
##  $ 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 178T
H PL NE" "13315 174TH AVE NE" "3303 178TH AVE NE" ...
```

```
##  $ 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 3
960 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 ...
##  $ year_of_sale            : chr [1:12865] "2006" "2006" "2006" "2006" ...

#I have change the name of Sale Date and sale price.
#I have also create new field year_of_sale that will be useful for predictor
for the sales price.

# 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 re
gression) and one that will contain Sale Price and several additional predict
ors of your choice. Explain the basis for your additional predictor selection
s.


housing_lm_1 <- lm(formula = Sale_Price ~ sq_ft_lot, data = housing_df)


housing_lm_2 <-lm(formula = Sale_Price ~ zip5 + bedrooms+ year_built, data =
housing_df)

#I think that zip codes, number of bedrooms and built year affects the sale p
rices

# iii.Execute a summary() function on two variables defined in the previous s
tep 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 inclusio
n of the additional predictors help explain any large variations found in Sal
e Price?

summary(housing_lm_1)
```

```
##
## 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(housing_lm_2)
```

```
##
## Call:
## lm(formula = Sale_Price ~ zip5 + bedrooms + year_built, data = housing_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -997873 -161449  -62624   63853 4115141
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.054e+09  1.957e+08  -5.385 7.35e-08 ***
## zip5         1.064e+04  1.996e+03   5.330 1.00e-07 ***
## bedrooms     1.035e+05  3.842e+03  26.931  < 2e-16 ***
## year_built   5.527e+03  1.963e+02  28.152  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 381500 on 12861 degrees of freedom
## Multiple R-squared:  0.1103, Adjusted R-squared:  0.1101
## F-statistic: 531.7 on 3 and 12861 DF,  p-value: < 2.2e-16
```

```
# iv.Considering the parameters of the multiple regression model you have cre
ated. What are the standardized betas for each parameter and what do the valu
es indicate?
library(lm.beta)
```

```
##
## Attaching package: 'lm.beta'
```

```
## The following object is masked from 'package:QuantPsyc':
##
##     lm.beta
```

```
coef_lmbeta <- lm.beta(housing_lm_2)
coef_lmbeta
```

```
##
## Call:
## lm(formula = Sale_Price ~ zip5 + bedrooms + year_built, data = housing_df)
##
## Standardized Coefficients::
## (Intercept)        zip5    bedrooms  year_built
##  0.00000000  0.04458759  0.22417183  0.23537926
```

```
# zip5 (standardized β = 0.04458759) - This value indicates that as zip code
increase by 1 standard deviation, sales price increase by  0.04458759 standar
d deviation.
#bedrooms (standardized β = 0.22417183) -This value indicates that as bedroom
s increase by 1 standard deviation, sales price increase by  0.22417183 stand
ard deviation.
#year_built(standardized β = 0.23537926) - This value indicates that as year_
built increase by 1 standard deviation, sales price increase by  0.23537926 s
tandard deviation.

# v. Calculate the confidence intervals for the parameters in your model and
explain what the results indicate.
confint(housing_lm_2)
```

```
##                     2.5 %        97.5 %
## (Intercept) -1.437177e+09 -6.701687e+08
## zip5         6.724735e+03  1.454870e+04
## bedrooms     9.593698e+04  1.109984e+05
## year_built   5.142553e+03  5.912266e+03
```

```
# In this model, the two best predictor (year_built) have very tight confiden
ce intervals, indicating that the estimates for the current model are likely
to be representative of the true population
# values. The interval for (zip5 and bedrooms) is wider (but still does not c
ross zero), indicating that the parameter for this variable is less represent
ative, but nevertheless significant.

# vi. Assess the improvement of the new model compared to your original model
(simple regression model) by testing whether this change is significant by pe
rforming an analysis of variance.
anova(housing_lm_1,housing_lm_2)
```

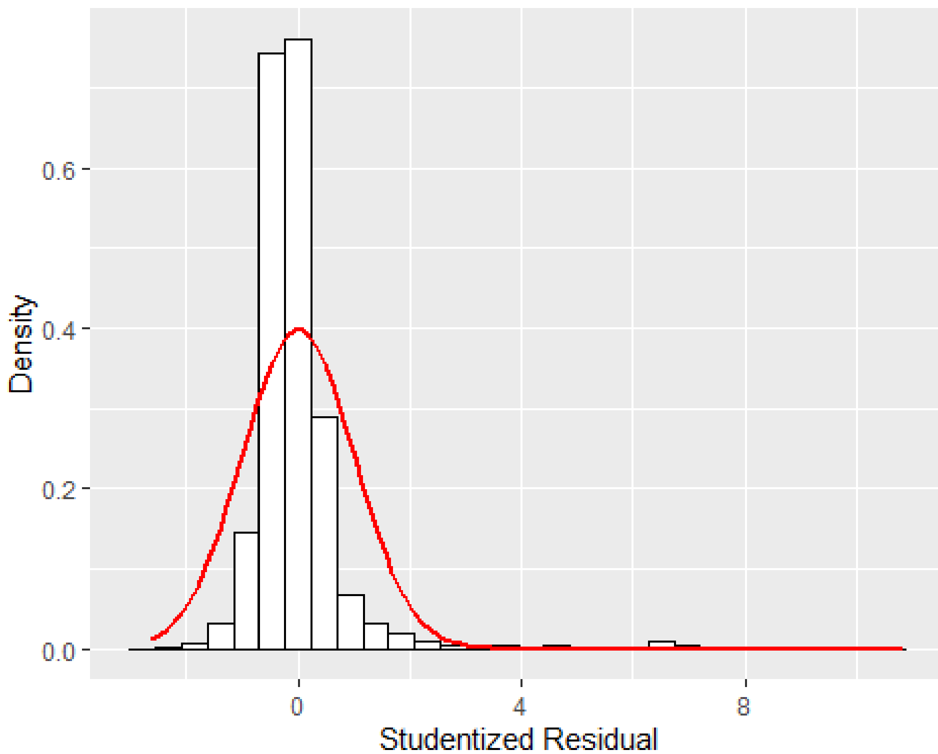
```
## Analysis of Variance Table
##
## Model 1: Sale_Price ~ sq_ft_lot
## Model 2: Sale_Price ~ zip5 + bedrooms + year_built
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
```

```
## 1  12863 2.0734e+15
## 2  12861 1.8715e+15  2 2.0192e+14 693.82 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# The value in column labelled Pr(>F) is 2.2e-16 (i.e., 2.2 with the decimal
place moved 16 places to
# the left, or a very small value indeed); we can say that housing_lm_2 signi
ficantly improved
# the fit of the model to the data compared to housing_lm_1, F(2, 12861) = 69
3.82, p < .001.

# vii. Perform casewise diagnostics to identify outliers and/or influential c
ases, storing each function's output in a dataframe assigned to a unique vari
able name.
housing_df$residuals<-resid(housing_lm_2)
housing_df$standardized.residuals<- rstandard(housing_lm_2)
housing_df$studentized.residuals<-rstudent(housing_lm_2)
housing_df$cooks.distance<-cooks.distance(housing_lm_2)
housing_df$dfbeta<-dfbeta(housing_lm_2)
housing_df$dffit<-dffits(housing_lm_2)
housing_df$leverage<-hatvalues(housing_lm_2)
housing_df$covariance.ratios<-covratio(housing_lm_2)

housing_df

## # A tibble: 12,865 x 33
##    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 28 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 <d
bl>,
## #   year_of_sale <chr>, residuals <dbl>, standardized.residuals <dbl>,
## #   studentized.residuals <dbl>, cooks.distance <dbl>, dfbeta <dbl[,4]>,
## #   dffit <dbl>, leverage <dbl>, covariance.ratios <dbl>
```

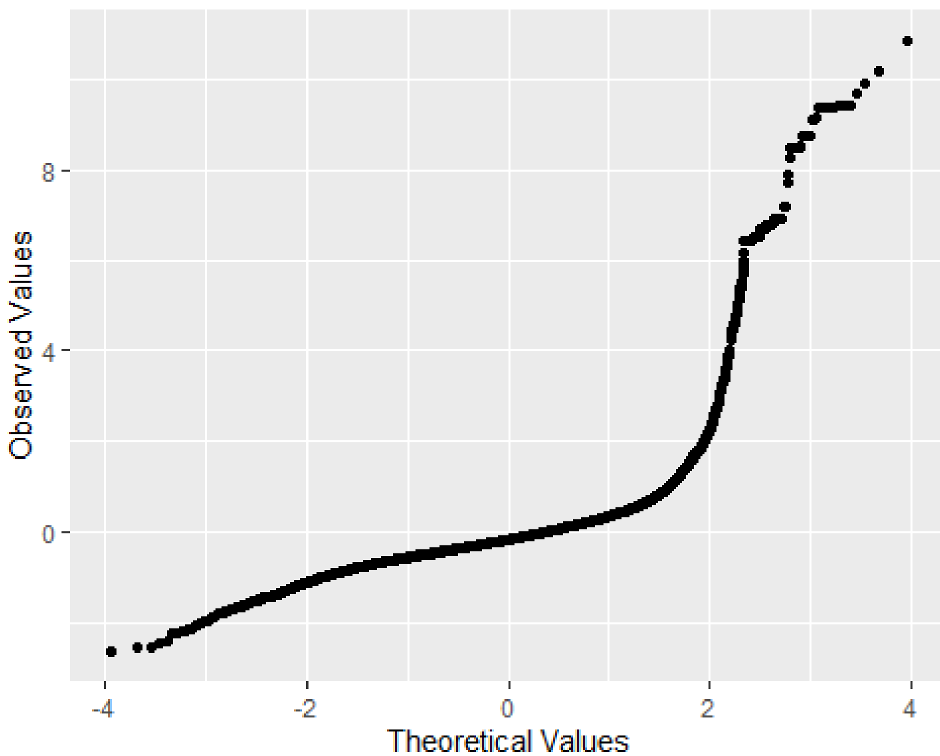
```r
# viii.Calculate the standardized residuals using the appropriate command, sp
ecifying those that are +-2, storing the results of large residuals in a vari
able you create.
housing_df$large.residual <- housing_df$standardized.residuals > 2 | housing_
df$standardized.residuals < -2

# ix.Use the appropriate function to show the sum of large residuals.
sum(housing_df$large.residual)

## [1] 346

# x.Which specific variables have large residuals (only cases that evaluate a
s TRUE)?
housing_df[housing_df$large.residual,c("Sale_Price", "zip5", "bedrooms", "yea
r_built","standardized.residuals")]

## # A tibble: 346 x 5
##    Sale_Price  zip5 bedrooms year_built standardized.residuals
##         <dbl> <dbl>    <dbl>      <dbl>                  <dbl>
##  1    1900000 98053        4       1990                   3.14
##  2    1520000 98052        5       1952                   2.45
##  3    1390000 98053        0       1955                   3.40
##  4    1588359 98053        2       2005                   2.65
##  5    1450000 98052        3       1972                   2.52
##  6    1450000 98052        2       1918                   3.58
##  7    2500000 98053        4       2005                   4.49
##  8    2169000 98053        4       2005                   3.63
##  9    1534000 98052        4       1963                   2.60
## 10    1968000 98053        4       1998                   3.20
## # ... with 336 more rows

# xi. Investigate further by calculating the leverage, cooks distance, and co
variance rations. Comment on all cases that are problematics.
housing_df[housing_df$large.residual , c("cooks.distance", "leverage", "covar
iance.ratios")]

## # A tibble: 346 x 3
##    cooks.distance leverage covariance.ratios
##             <dbl>    <dbl>             <dbl>
##  1       0.000284 0.000115             0.997
##  2       0.00114  0.000761             0.999
##  3       0.00484  0.00167              0.998
##  4       0.000597 0.000341             0.998
##  5       0.000347 0.000219             0.999
##  6       0.00563  0.00176              0.998
##  7       0.000738 0.000146             0.994
##  8       0.000480 0.000146             0.996
##  9       0.000581 0.000344             0.999
## 10       0.000300 0.000117             0.997
## # ... with 336 more rows
```

```r
# Executing this command prints the variables (or columns) labelled cooks.dis
tance, leverage, and covariance.ratios but only for cases for which large.res
idual is TRUE.
# Output shows these values; none of them has a Cook’s distance greater than
1 , so none of the cases is having an undue influence
# on the model. The average leverage can be calculated as 0.011 (k + 1/n = 4/
346) and so we are looking for values either twice as large as this (0.022) o
r three times as large
# (0.033) depending on which statistician you trust most. All cases are withi
n the boundary of three times the average and only case 1 is close to two tim
es the average.

# xii. Perform the necessary calculations to assess the assumption of indepen
dence and state if the condition is met or not.
#durbinWatsonTest(housing_lm_2)

# From the output we can see that the test statistic is 0.7442029 and the cor
responding p-value is 0. Since this p-value is less than 0.05, we can reject
the null hypothesis and conclude that the residuals in this regression model
are autocorrelated.
# As a conservative rule, D-W Statistic values less than 1 or greater than 3
should definitely raise alarm bells.
#The closer to 2 that the value is, the better, and for these data the value
is 0.744, which is less than 1 suggests that the assumption might not certain
ly been met.

# xiii.Perform the necessary calculations to assess the assumption of no mult
icollinearity and state if the condition is met or not.
vif(housing_lm_2)

##       zip5   bedrooms year_built
##   1.011771   1.001607   1.010570

#tolerance statistics
1/vif(housing_lm_2)

##       zip5   bedrooms year_built
##  0.9883661  0.9983956  0.9895403

mean(vif(housing_lm_2))

## [1] 1.007983

# For our current model the VIF values are all well below 10 and the toleranc
e statistics all  well above 0.2. Also, the average VIF is very close to 1. B
ased on these measures we can safely conclude that there is no collinearity w
ithin our 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.
housing_df$fitted <- housing_lm_2$fitted.values
```

```
library(ggplot2)
histogram<-ggplot(housing_df, aes(studentized.residuals)) + geom_histogram(ae
s(y = ..density..), colour = "black", fill = "white") + labs(x = "Studentized
Residual", y = "Density")
histogram + stat_function(fun = dnorm, args = list(mean = mean(housing_df$stu
dentized.residuals, na.rm = TRUE), sd = sd(housing_df$studentized.residuals,
na.rm = TRUE)), colour= "red", size = 1)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
qplot(sample = housing_df$studentized.residuals, stat="qq") + labs(x ="Theore
tical Values", y = "Observed Values")

## Warning: `stat` is deprecated
```




```
# The histogram should look like a normal distribution (a bell-shaped curve).
For the housing data data, the distribution is roughly normal.
#We could summarize by saying that the model appears, in most senses, to be b
oth accurate for the sample and generalizable to the population.

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

# vif values to check model bias
# When we check multi collinearity we check for vif score

vif(housing_lm_2)

##       zip5   bedrooms year_built
##   1.011771   1.001607   1.010570

# None of the vif scores are near 5 or greater and thus predictors does not
# have any significant multi collinearity. Multi collinearity problems consis
t of
# including, in the model, different variables that have a similar predictive
# relationship with the outcome.

mean(vif(housing_lm_2))

## [1] 1.007983
```

```
# Average vif is >1 but nowhere close to 5 or greater. Model does not appear
to have significant proof that model is biased.
```