DSC520 HousingData Exercise 8.2

Anjani Bonda

February 12th 2022

```
library(readxl)
library(dplyr)

##
## 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

library(purrr)
library(ggplot2)
library(lmtest)

## Loading required package: zoo

##
## Attaching package: 'zoo'

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

library(lm.beta)
library(car)

## Loading required package: carData

##
## Attaching package: 'car'

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

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

```r
## Set workding directory to read source datasets.
setwd("/Users/anjanibonda/DSC520/dsc520")

## Read housing dataset
housingdata <- read_excel("data/week-6-housing.xlsx")
glimpse(housingdata)
```

```
## 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, …
```

```
## Check for nulls in all rows
apply(housingdata, 2, function(i) any(is.na(i)))
```

```
##                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
```

```
## Looking at the data, there is missing data for sale_warning and ctyname

# I. Explain any transformations or modifications you made to the dataset
colnames(housingdata)[1] <- "Sale_Date"
colnames(housingdata)[2] <- "Sale_Price"

## I have Changed the column names of Sale Date and Sale Price to avoid any p
ossible issues.

# II. Create two variables;
#    one that will contain the variables Sale Price and Square Foot of Lot (s
ame variables used from previous assignment on simple regression)
#    and one that will contain Sale Price and several additional predictors o
f your choice.
#    Explain the basis for your additional predictor selections.
housingdata_lm1 <- lm(formula = Sale_Price ~ sq_ft_lot, data = housingdata)
housingdata_lm2 <- lm(formula = Sale_Price ~ zip5 + bedrooms + year_built, da
ta = housingdata)

## I have inlcuded other predictors like zip5, bedroomms and year built as th
ose are often key factors in home price predictions.
```

```
# III. 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 t
ell you about the overall model.
#    Did the inclusion of the additional predictors help explain any large va
riations found in Sale Price?
summary(housingdata_lm1)

##
## Call:
## lm(formula = Sale_Price ~ sq_ft_lot, data = housingdata)
##
## 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(housingdata_lm2)

##
## Call:
## lm(formula = Sale_Price ~ zip5 + bedrooms + year_built, data = housingdata
)
##
## 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
```

```
## R2 for housingdata_lm1: 0.01  adjusted: 0.01
## R2 for housingdata_lm2: 0.11 adjusted: 0.11
## RSquared is a statistical measure of fit for the model.
## These low RSquared values mean that the model is not a great fit.
## The multiple regression seems OK, but not ideal.

# IV. Considering the parameters of the multiple regression model you have cr
eated,
#     What are the standardized betas for each parameter and what do the valu
es indicate?
coef_lmbeta <- lm.beta(housingdata_lm2)
coef_lmbeta

##
## Call:
## lm(formula = Sale_Price ~ zip5 + bedrooms + year_built, data = housingdata
)
##
## 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 standard deviatio
n.
## bedrooms (standardized β = 0.22417183) -This value indicates that as bedro
oms
## 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 yea
r_# built
## increase by 1 standard deviation, sales price increase by 0.23537926 stand
ard deviation.


# V. Calculate the confidence intervals for the parameters in your model and
#     explain what the results indicate.
confint(housingdata_lm2)

##                     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 predictor (year_built) have very tight confidence inter
vals,
## indicating that the estimates for the current model are likely
## to be representative of the true population.
```

```
## The confidence interval for (zip5 and bedrooms) is wider but still does no
t cross zero,
## indicating that the parameter for this variable is less representative, bu
t still 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 performing an analysis
of variance.
anova(housingdata_lm1,housingdata_lm2)

## 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 p value is very small value indeed,
## we can say that housingdata_lm2 significantly improved
## the fit of the model to the data compared to housingdata_lm1

# 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.
housingdata$residuals<-resid(housingdata_lm2)
housingdata$standardized.residuals<- rstandard(housingdata_lm2)
housingdata$studentized.residuals<-rstudent(housingdata_lm2)
housingdata$cooks.distance<-cooks.distance(housingdata_lm2)
housingdata$leverage<-hatvalues(housingdata_lm2)
housingdata$covariance.ratios<-covratio(housingdata_lm2)

head(housingdata)

## # A tibble: 6 × 30
##   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 25 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>, residuals <dbl>,
## #   standardized.residuals <dbl>, studentized.residuals <dbl>, …
```

```
# VIII. Calculate the standardized residuals using the appropriate command,
#     specifying those that are +-2, storing the results of large residuals i
n a variable you create.
housingdata$large.residual <- housingdata$standardized.residuals > 2 | housin
gdata$standardized.residuals < -2
head(housingdata$large.residual)
```

```
##     1     2     3     4     5     6
## FALSE FALSE FALSE FALSE FALSE FALSE
```

```
# IX. Use the appropriate function to show the sum of large residuals.
sum(housingdata$large.residual)
```

```
## [1] 346
```

```
# X. Which specific variables have large residuals (only cases that evaluate
as TRUE)?
housingdata[housingdata$large.residual,c("Sale_Price", "zip5", "bedrooms", "y
ear_built","standardized.residuals")]
```

```
## # A tibble: 346 × 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 covariance ratios.
# Comment on all cases that are problematics.
housingdata[housingdata$large.residual , c("cooks.distance", "leverage", "cov
ariance.ratios")]
```

```
## # A tibble: 346 × 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
```

```
## None of the values has a Cook's distance greater than 1 ,
## The leverage values also seem miniscule.
```

```
# XII. Perform the necessary calculations to assess the assumption of indepen
dence
#    and state if the condition is met or not.
durbinWatsonTest(housingdata_lm2)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.6278972     0.7442029       0
##  Alternative hypothesis: rho != 0
```

```
## The test statistic is 0.7442029 and the corresponding p-value is 0.
## Since this p-value is less than 0.05, we can reject the null hypothesis an
d
## conclude that the residuals in this regression model are autocorrelated.
## Value less than 1 suggests that the assumption might not been met.
```

```
# XIII. Perform the necessary calculations to assess the assumption of no mul
ticollinearity
#    and state if the condition is met or not.
vif(housingdata_lm2)
```

```
##       zip5   bedrooms year_built
##   1.011771   1.001607   1.010570
```

```
## tolerance statistics
1/vif(housingdata_lm2)
```

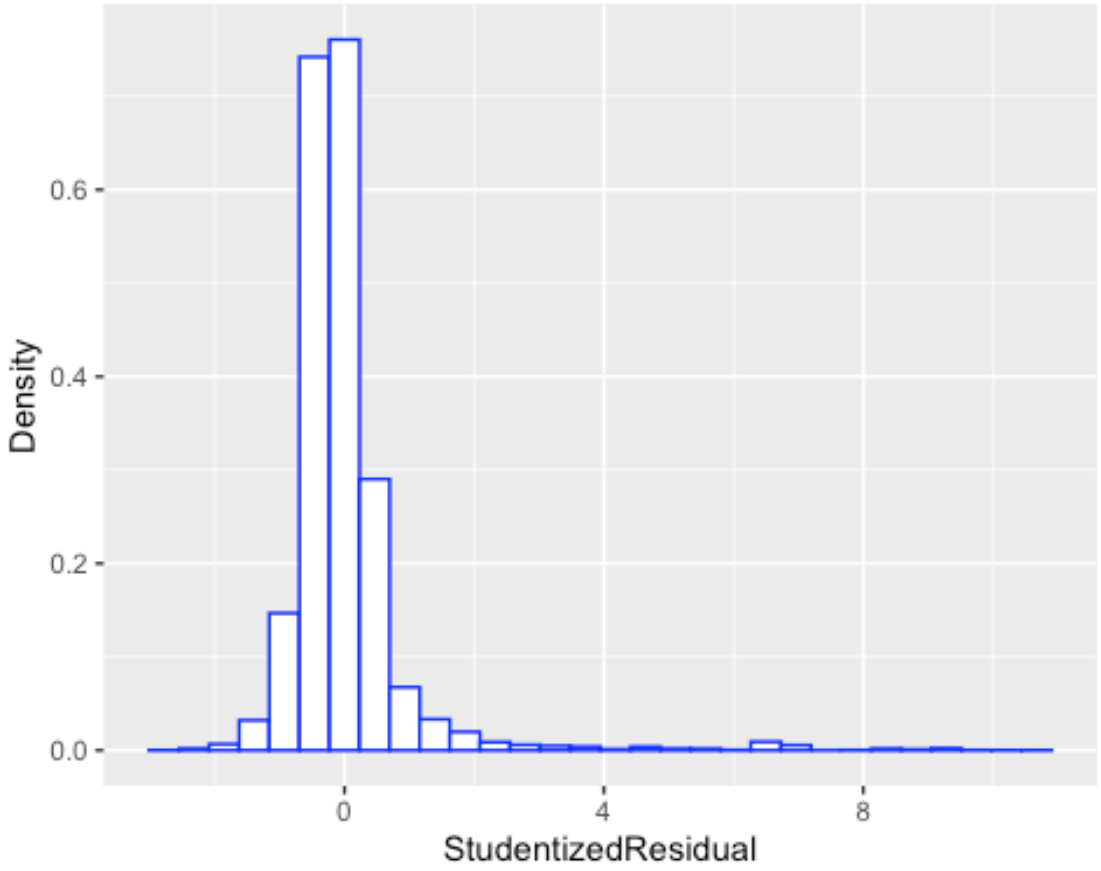
```
##       zip5   bedrooms year_built
##  0.9883661  0.9983956  0.9895403
```

```
mean(vif(housingdata_lm2))
```

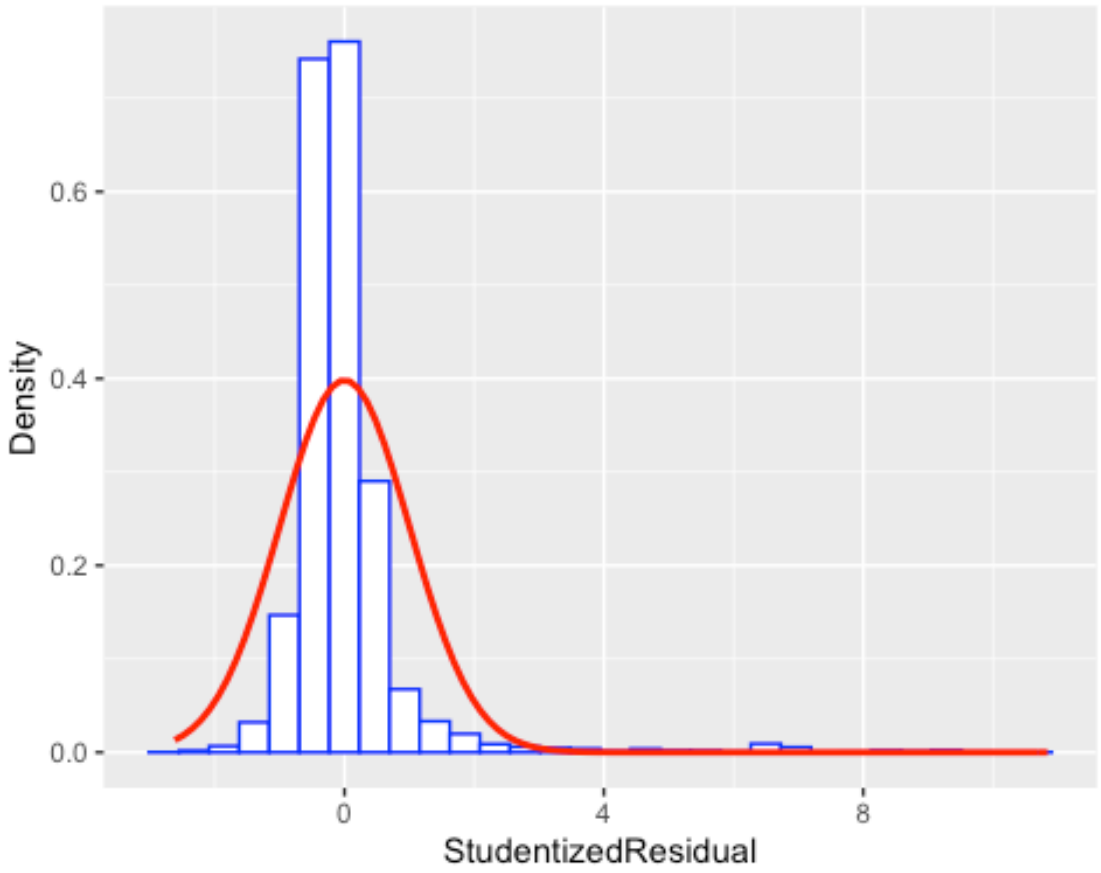
```
## [1] 1.007983
```

```
## VIF values are all below 10 and the tolerance statistics above 0.2.
## Also, the mean VIF is ~ 1.
## Based on these results we can conclude that there is no collinearity in da
ta.
```

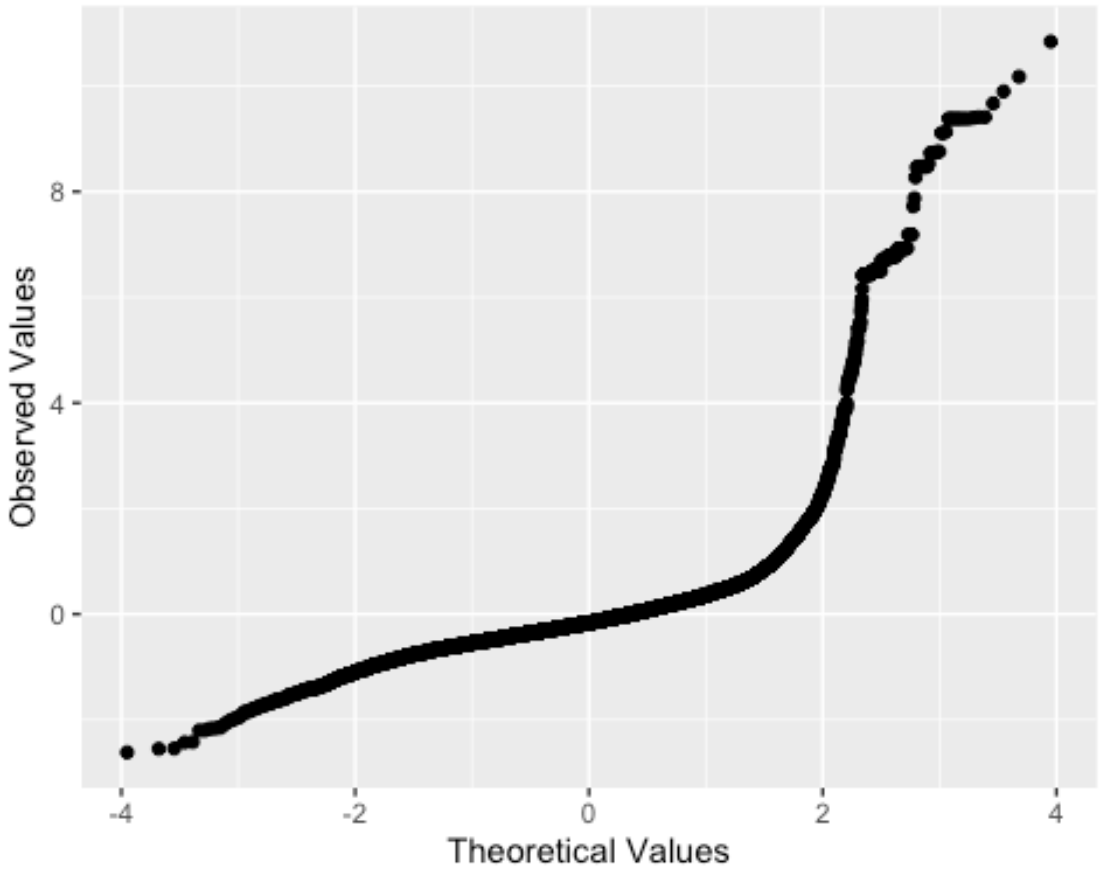
```
# XIV. Visually check the assumptions related to the residuals using the plot
() and hist() functions.
#     Summarize what each graph is informing you of and if any anomalies are
present.
housingdata$fitted <- housingdata_lm2$fitted.values

histogram<-ggplot(housingdata, aes(studentized.residuals)) +
  geom_histogram(aes(y = ..density..), colour = "blue", fill = "white") +
  labs(x = "StudentizedResidual", y = "Density")
histogram

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




```
histogram + stat_function(fun = dnorm, args = list(mean = mean(housingdata$st
udentized.residuals, na.rm = TRUE),
  sd = sd(housingdata$studentized.residuals,na.rm = TRUE)), colour= "red", si
ze = 1)

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




```
qplot(sample = housingdata$studentized.residuals, stat="qq") + labs(x ="Theor
etical Values", y = "Observed Values")

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




```
## The distribution is roughly normal.
## To summarize, the model appears to be accurate for the sample and can be g
eneralized to the population.

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

## Based on vif score/values calculated above, since the values are not close
to 5, the predictors doesn't have
## any significant multi collinearity.
## Mean vif is also just above 1 but no where near 5.
## Hence, Model does not appear to be biased.
```