Exerise12_AyachitMadhukar

Exercise #12

a. Explain why you chose to remove data points from your 'clean' dataset.

There are charater fieldds like address and city etc. Property type is the only meaningful categorical variable but it is "R" throughout the ditribution, Hence to make multiple regression keeping all the number fields to calculate sale price. After ovserving data using "head" and "str" , I have decided to create a clener dataset with all numeric variables.

```
setwd("~/MadR/Workspaces/dsc520")
library("readxl")
housing_df <- read_excel("data/week-7-housing.xlsx")

head(housing_df)
```

```
## # 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>
```

```
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 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 N
##  $ 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 ...
```

```
names(housing_df) <- make.names(names(housing_df))

Additional_Predicator<-(housing_df[c("Sale.Price",
                                     "sale_reason",
                                     "zip5",
                                     "lon",
                                     "building_grade",
                                     "square_feet_total_living",
                                     "bedrooms",
                                     "bath_3qtr_count",
                                     "year_built",
                                     "year_renovated",
                                     "sq_ft_lot")])
```

b. 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.

Keeping only significant variabls in the dataset after observing summary of multiple regression outout

```
s1<-lm(Sale.Price~sq_ft_lot, data=housing_df)
m1<-lm(Sale.Price~bath_3qtr_count+bedrooms+building_grade+lon+sale_reason+sq_ft_lot+square_feet_total_li
   data=Additional_Predicator)

summary(s1)
```

```
##
## 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(m1)
```

```
##
## Call:
## lm(formula = Sale.Price ~ bath_3qtr_count + bedrooms + building_grade +
##     lon + sale_reason + sq_ft_lot + square_feet_total_living +
##     year_built + year_renovated + zip5, data = Additional_Predicator)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2217500  -119481   -43977    41356  3697461
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -4.048e+08  1.981e+08  -2.044  0.04099 *
## bath_3qtr_count           -1.609e+04  5.107e+03  -3.151  0.00163 **
## bedrooms                  -9.959e+03  4.777e+03  -2.085  0.03710 *
## building_grade             2.705e+04  4.494e+03   6.018 1.81e-09 ***
## lon                       -3.684e+05  7.414e+04  -4.969 6.82e-07 ***
## sale_reason               -1.165e+04  1.179e+03  -9.884  < 2e-16 ***
## sq_ft_lot                  3.685e-01  6.078e-02   6.063 1.37e-09 ***
## square_feet_total_living   1.480e+02  5.954e+00  24.864  < 2e-16 ***
## year_built                 2.914e+03  2.315e+02  12.589  < 2e-16 ***
## year_renovated             7.985e+01  1.427e+01   5.595 2.25e-08 ***
## zip5                       3.611e+03  1.987e+03   1.817  0.06917 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 354300 on 12854 degrees of freedom
## Multiple R-squared:  0.233,  Adjusted R-squared:  0.2324
## F-statistic: 390.5 on 10 and 12854 DF,  p-value: < 2.2e-16
```

c. 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?

```
sales_lm<-  lm(Sale.Price~bath_3qtr_count+bedrooms+building_grade+lon+sale_reason+sq_ft_lot+square_feet_
   data=Additional_Predicator)
summary(sales_lm)
```

```
##
```

```
## Call:
## lm(formula = Sale.Price ~ bath_3qtr_count + bedrooms + building_grade +
##     lon + sale_reason + sq_ft_lot + square_feet_total_living +
##     year_built + year_renovated + zip5, data = Additional_Predicator)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2217500  -119481   -43977    41356  3697461
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -4.048e+08  1.981e+08  -2.044  0.04099 *
## bath_3qtr_count          -1.609e+04  5.107e+03  -3.151  0.00163 **
## bedrooms                 -9.959e+03  4.777e+03  -2.085  0.03710 *
## building_grade            2.705e+04  4.494e+03   6.018 1.81e-09 ***
## lon                      -3.684e+05  7.414e+04  -4.969 6.82e-07 ***
## sale_reason              -1.165e+04  1.179e+03  -9.884  < 2e-16 ***
## sq_ft_lot                 3.685e-01  6.078e-02   6.063 1.37e-09 ***
## square_feet_total_living  1.480e+02  5.954e+00  24.864  < 2e-16 ***
## year_built                2.914e+03  2.315e+02  12.589  < 2e-16 ***
## year_renovated            7.985e+01  1.427e+01   5.595 2.25e-08 ***
## zip5                      3.611e+03  1.987e+03   1.817  0.06917 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 354300 on 12854 degrees of freedom
## Multiple R-squared:  0.233,  Adjusted R-squared:  0.2324
## F-statistic: 390.5 on 10 and 12854 DF,  p-value: < 2.2e-16
```

refer anser "b" for summary output R2 and Adjusted R were 0.01435 and 0.01428 respectively for simple regression vs it got shifted significantly after adding additional predicators to R-squared: 0.233 & Adjusted R-squared: 0.2324

"R-squared (R2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model".

"With a multiple regression made up of several independent variables, the R-Squared must be adjusted. The adjusted R-squared compares the descriptive power of regression models that include diverse numbers of predictors. Every predictor added to a model increases R-squared and never decreases it"

Addional predicator bring significcatio variation 0.2324 - 0.01428 = 0.21812

d. 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?

```
library("lm.beta")
lm.beta(sales_lm)
```

```
##
## Call:
## lm(formula = Sale.Price ~ bath_3qtr_count + bedrooms + building_grade +
```

```
##     lon + sale_reason + sq_ft_lot + square_feet_total_living +
##     year_built + year_renovated + zip5, data = Additional_Predicator)
##
## Standardized Coefficients::
##              (Intercept)          bath_3qtr_count                bedrooms
##               0.00000000              -0.02586836             -0.02157809
##           building_grade                       lon             sale_reason
##               0.07308431              -0.04756314             -0.07708122
##               sq_ft_lot square_feet_total_living              year_built
##               0.05188477               0.36234732              0.12408918
##           year_renovated                      zip5
##               0.04492051               0.01513880
```

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

```
library("gmodels")
ci(sales_lm)
```

```
##                               Estimate      CI lower      CI upper   Std. Error
## (Intercept)              -4.047830e+08 -7.929967e+08 -1.656926e+07 1.980532e+08
## bath_3qtr_count          -1.609225e+04 -2.610227e+04 -6.082227e+03 5.106767e+03
## bedrooms                 -9.959480e+03 -1.932316e+04 -5.958023e+02 4.777025e+03
## building_grade            2.704858e+04  1.823894e+04  3.585821e+04 4.494370e+03
## lon                      -3.683910e+05 -5.137169e+05 -2.230650e+05 7.414028e+04
## sale_reason              -1.165151e+04 -1.396223e+04 -9.340802e+03 1.178846e+03
## sq_ft_lot                 3.685229e-01  2.493876e-01  4.876581e-01 6.077869e-02
## square_feet_total_living  1.480337e+02  1.363634e+02  1.597041e+02 5.953831e+00
## year_built                2.913985e+03  2.460252e+03  3.367719e+03 2.314791e+02
## year_renovated            7.985228e+01  5.187819e+01  1.078264e+02 1.427141e+01
## zip5                      3.611480e+03 -2.836000e+02  7.506560e+03 1.987135e+03
##                                p-value
## (Intercept)               4.099274e-02
## bath_3qtr_count           1.629956e-03
## bedrooms                  3.710070e-02
## building_grade            1.810287e-09
## lon                       6.822234e-07
## sale_reason               5.906960e-23
## sq_ft_lot                 1.370474e-09
## square_feet_total_living 2.533803e-133
## year_built                3.987167e-36
## year_renovated            2.247931e-08
## zip5                      6.917449e-02
```

f. 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(s1, sales_lm)
```

```
## Analysis of Variance Table
##
## Model 1: Sale.Price ~ sq_ft_lot
```

```
## Model 2: Sale.Price ~ bath_3qtr_count + bedrooms + building_grade + lon +
##     sale_reason + sq_ft_lot + square_feet_total_living + year_built +
##     year_renovated + zip5
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
## 1  12863 2.0734e+15
## 2  12854 1.6134e+15  9 4.5994e+14 407.14 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
Additional_Predicator$standardized.residuals<- rstandard(sales_lm)
Additional_Predicator$studentized.residuals<-rstudent(sales_lm)
Additional_Predicator$cooks.distance<-cooks.distance(sales_lm)
Additional_Predicator$dfbeta<-dfbeta(sales_lm)
Additional_Predicator$dffit<-dffits(sales_lm)
Additional_Predicator$leverage<-hatvalues(sales_lm)
Additional_Predicator$covariance.ratios<-covratio(sales_lm)
```

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

```
Additional_Predicator$standardized.residuals > 2 | Additional_Predicator$standardized.residuals < -2
```

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

```
Additional_Predicator$large.residual <- Additional_Predicator$standardized.residuals > 2 | Additional_Pr
sum(Additional_Predicator$large.residual)
```

```
## [1] 329
```

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

```
Additional_Predicator[  Additional_Predicator$large.residual ==TRUE, ]
```

```
## # A tibble: 329 x 19
##    Sale.Price sale_reason  zip5   lon building_grade square_feet_tot~ bedrooms
##         <dbl>       <dbl> <dbl> <dbl>          <dbl>            <dbl>    <dbl>
##  1     265000           1 98053 -122.             10             4920        4
##  2    1520000          18 98052 -122.              9             4640        5
##  3    1390000           1 98053 -122.              6              660        0
##  4     390000           1 98052 -122.             11             5800        5
##  5    1588359           1 98053 -122.              9             3360        2
##  6    1450000           1 98052 -122.              6              900        2
##  7     163000           1 98053 -122.              9             4710        4
##  8     270000           1 98053 -122.             11             5060        4
##  9     200000           1 98053 -122.             10             6880        5
## 10     300000           1 98052 -122.             11             4490        4
## # ... with 319 more rows, and 22 more variables: bath_3qtr_count <dbl>,
```

```
## #   year_built <dbl>, year_renovated <dbl>, sq_ft_lot <dbl>,
## #   standardized.residuals <dbl>, studentized.residuals <dbl>,
## #   cooks.distance <dbl>, dfbeta[,"(Intercept)"] <dbl>,
## #   [,"bath_3qtr_count"] <dbl>, [,"bedrooms"] <dbl>, [,"building_grade"] <dbl>,
## #   [,"lon"] <dbl>, [,"sale_reason"] <dbl>, [,"sq_ft_lot"] <dbl>,
## #   [,"square_feet_total_living"] <dbl>, [,"year_built"] <dbl>,
## #   [,"year_renovated"] <dbl>, [,"zip5"] <dbl>, dffit <dbl>, leverage <dbl>,
## #   covariance.ratios <dbl>, large.residual <lgl>
```

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

```
Additional_Predicator[Additional_Predicator$large.residual, c("cooks.distance", "leverage", "covariance
```

```
## # A tibble: 329 x 3
##    cooks.distance leverage covariance.ratios
##             <dbl>    <dbl>             <dbl>
##  1       0.000455 0.000867             0.997
##  2       0.00401  0.0101               1.01
##  3       0.00241  0.00297              0.996
##  4       0.000770 0.00130              0.997
##  5       0.000319 0.000784             0.998
##  6       0.00276  0.00231              0.992
##  7       0.000730 0.00145              0.998
##  8       0.000548 0.000908             0.996
##  9       0.00534  0.00498              0.996
## 10       0.000354 0.000785             0.997
## # ... with 319 more rows
```

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

```
library("car")
```

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

```
durbinWatsonTest(sales_lm)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.7302265     0.5395379       0
##  Alternative hypothesis: rho != 0
```

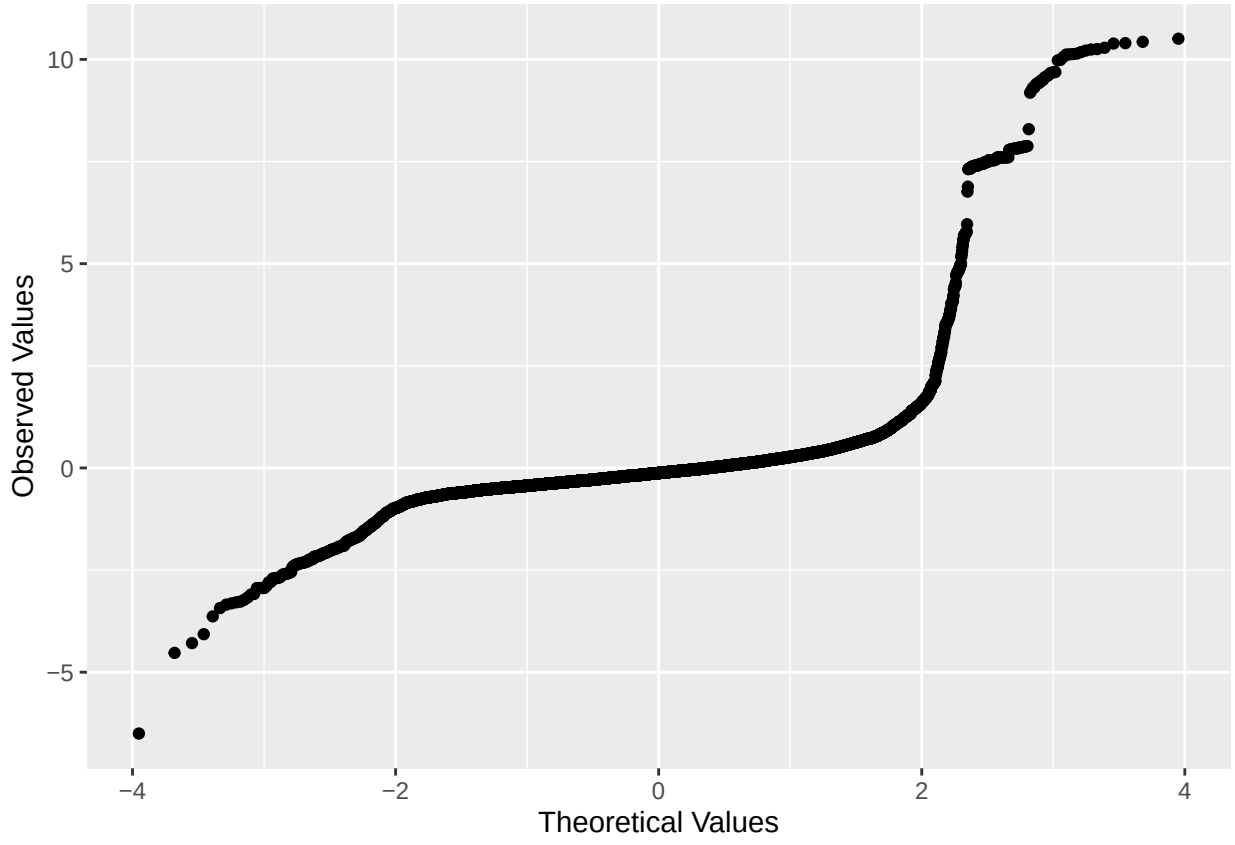
```
vif(sales_lm)
```

```
##          bath_3qtr_count                 bedrooms           building_grade
##                 1.129379                 1.795192                 2.471384
##                      lon              sale_reason               sq_ft_lot
##                 1.535588                 1.019271                 1.227148
## square_feet_total_living               year_built           year_renovated
##                 3.559302                 1.628395                 1.080169
##                     zip5
##                 1.162815
```

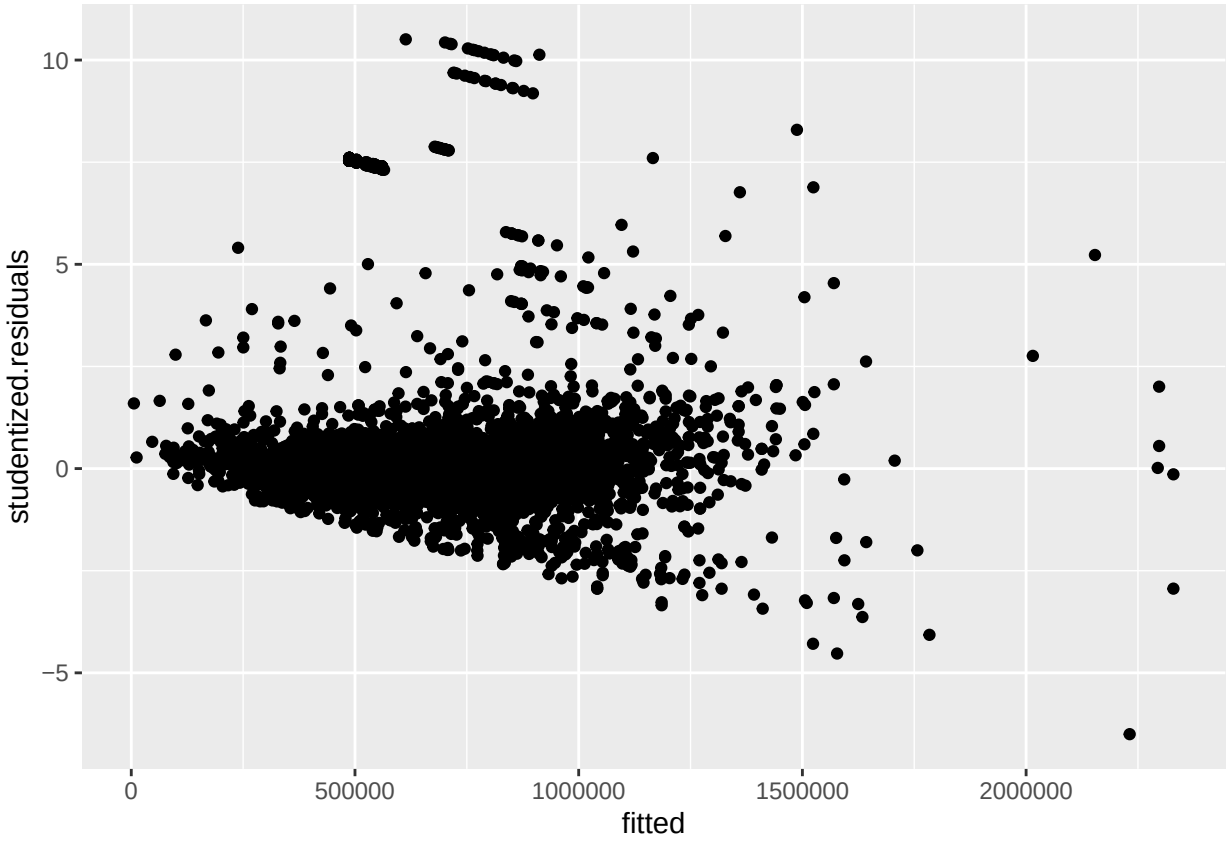
m. Visually check the assumptions related to the residuals using the plot() and hist() functions. Summarize what each graph is informing you of and if any anomalies are present.

```
Additional_Predicator$fitted <- sales_lm$fitted.values

library("ggplot2")
qqplot.resid <- qplot(sample = Additional_Predicator$studentized.residuals)
qqplot.resid                  + labs(x = "Theoretical Values", y = "Observed Values")
```

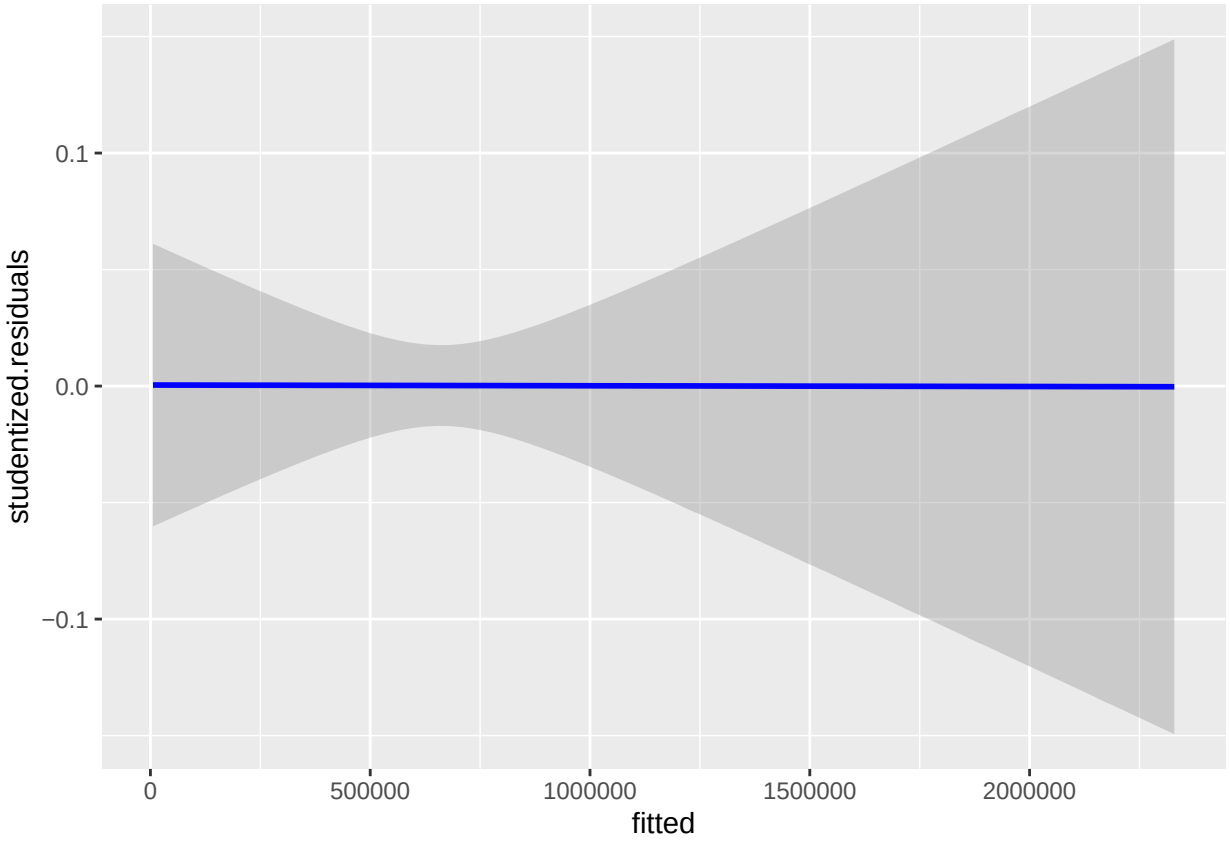



```
scatter <- ggplot(Additional_Predicator, aes(fitted, studentized.residuals))
scatter + geom_point()
```

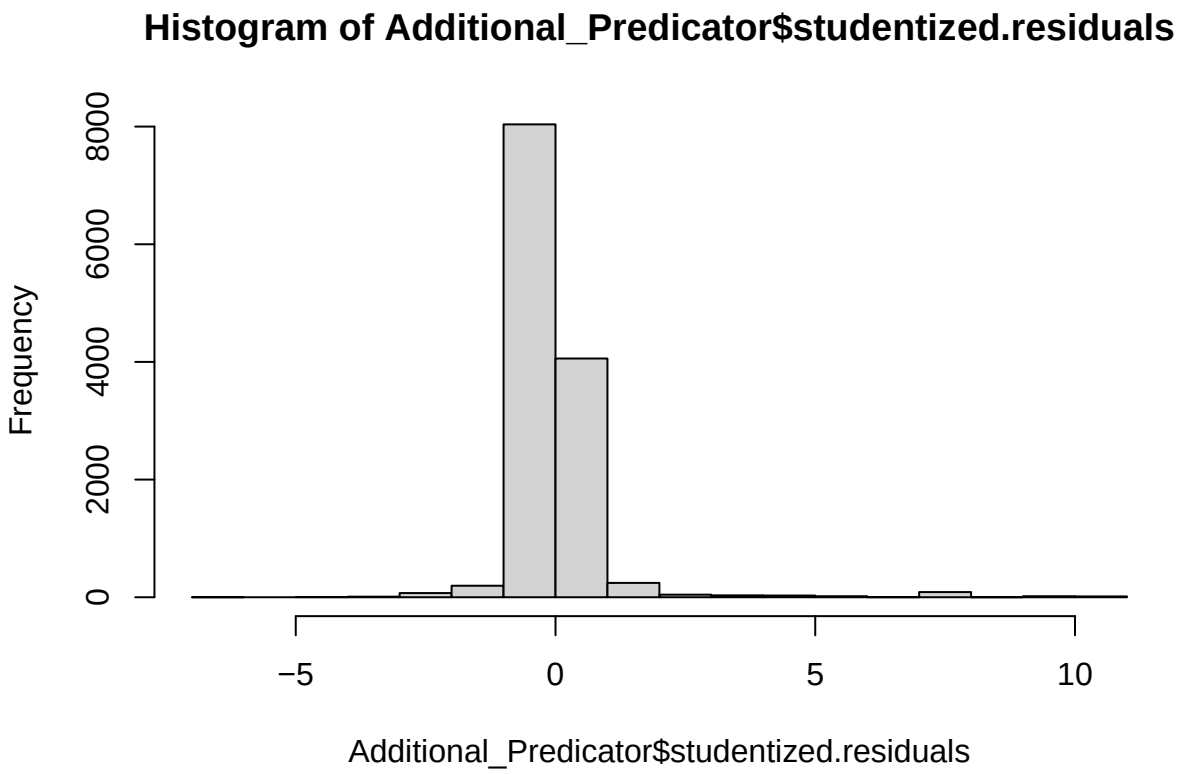



```
scatter + geom_smooth(method = "lm", colour = "Blue")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
hist(Additional_Predicator$studentized.residuals)
```



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