Assignment 6.1 housing data

Rachel Nelson

7/11/2020

Data for this assignment is focused on real estate transactions recorded from 1964 to 2016 and will use statistical correlation, multiple regression and R programming with focus on following variables: Sale Price and several other possible predictors.

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

```
setwd("C:/Users/Rachel/Desktop/College/DSC520/dsc520-master")
housing_df <- read_excel("data/week-6-housing.xlsx")
```

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.

```
model_1 <- lm(housing_df$‘Sale Price‘ ~ sq_ft_lot ,data = housing_df)
model_2 <- lm(housing_df$‘Sale Price‘ ~ sq_ft_lot + zip5 + bedrooms + square_feet_total_living + bath_fu
```

The additional predictors are baesd on which variables are avaialble as numerical values excluding lon and lat since I have zip code as a factor and it woukld be duplicative.

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?

```
summary(model_1)
```

```
##
## Call:
## lm(formula = housing_df$‘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(model_2)
```

```
##
## Call:
## lm(formula = housing_df$‘Sale Price‘ ~ sq_ft_lot + zip5 + bedrooms +
##     square_feet_total_living + bath_full_count + bath_half_count +
##     housing_df$bath_3qtr_count + year_built + building_grade +
##     sale_reason + sale_instrument + housing_df$year_renovated,
##     data = housing_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2101121  -120052   -44063    42308  3724591
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -3.931e+07  1.843e+08  -0.213    0.831
## sq_ft_lot                   2.927e-01  5.941e-02   4.927 8.44e-07 ***
## zip5                        3.507e+02  1.880e+03   0.187    0.852
## bedrooms                   -3.718e+03  4.729e+03  -0.786    0.432
## square_feet_total_living    1.443e+02  6.488e+00  22.247  < 2e-16 ***
## bath_full_count            -6.489e+01  7.610e+03  -0.009    0.993
## bath_half_count            -9.841e+02  7.133e+03  -0.138    0.890
## housing_df$bath_3qtr_count -1.568e+04  6.945e+03  -2.258    0.024 *
## year_built                  2.507e+03  2.288e+02  10.955  < 2e-16 ***
## building_grade              3.077e+04  4.449e+03   6.917 4.84e-12 ***
## sale_reason                -1.179e+04  1.284e+03  -9.183  < 2e-16 ***
## sale_instrument             2.430e+02  1.040e+03   0.234    0.815
## housing_df$year_renovated   7.629e+01  1.435e+01   5.318 1.07e-07 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 354700 on 12852 degrees of freedom
## Multiple R-squared:  0.2315, Adjusted R-squared:  0.2308
## F-statistic: 322.7 on 12 and 12852 DF,  p-value: < 2.2e-16
```

The R Squared for model_1 is 0.014 whle the R squared for model_2 is 0.2315. The R squared will tell you how well the model fits your data. In the first model, it only fits a little more then 1% of my data, while in the second model, it fits 20%.

The R Squared adjusted is the same meaning, but adjusted for the number of predictors in the model. The first model has one predictor, while the second model has 8 predictors. In this case, the first model has an R squared adjusted of 0.1428 while the second model has an adjusted r-squared of 0.2308. In both cases, model_2 is a better model to fit the data then model_1.

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)
model_2.beta <- lm.beta(model_2)
coef(model_2.beta)
```

```
##                (Intercept)                  sq_ft_lot
##                0.000000000                0.041214299
##                       zip5                   bedrooms
##                0.001470073               -0.008056089
##   square_feet_total_living            bath_full_count
##                0.353302510               -0.000104433
##            bath_half_count housing_df$bath_3qtr_count
##               -0.001280755               -0.025206690
##                 year_built             building_grade
##                0.106744000                0.083140469
##                sale_reason            sale_instrument
##               -0.077996300                0.001989165
##  housing_df$year_renovated
##                0.042918307
```

The standardized betaS for each parameter are listed above. Standardized betas compare strength of each predictor to the Sale Price with the higher the absolute value of the beta coefficient, the stronger the effect.

The strongest standardized beta in this data set is the square_feet_total_living followed by the year built.

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

```
confint(model_2)
```

```
##                                    2.5 %        97.5 %
## (Intercept)                -4.006125e+08  3.219891e+08
## sq_ft_lot                   1.762814e-01  4.091856e-01
## zip5                       -3.334524e+03  4.035919e+03
## bedrooms                   -1.298818e+04  5.551522e+03
## square_feet_total_living    1.316209e+02  1.570563e+02
## bath_full_count            -1.498144e+04  1.485166e+04
## bath_half_count            -1.496524e+04  1.299702e+04
## housing_df$bath_3qtr_count -2.929383e+04 -2.067434e+03
## year_built                  2.058172e+03  2.955165e+03
## building_grade              2.205019e+04  3.949055e+04
## sale_reason                -1.430655e+04 -9.273119e+03
## sale_instrument            -1.796427e+03  2.282353e+03
## housing_df$year_renovated   4.817277e+01  1.044134e+02
```

The results indicate that 95% of these confidence intervals would contain the true value b. A good model would have a small confidence interval. However, note that in this model, we have a confidence interval that cross zero, which indicates that in some samples, the predictor has a negative relationship to the outcome whereas in others, it has a positive relationship.

This tells us that the best predictors which have a small confidence interval include square_feet_total_living, year built, year built and bath_3qtr_count. These predictors have smaller confidence intervals, which indicates the estimates for the current model are likely to be representative of the true population values.

The predictors that cross zero, such as zip5, bedrooms, bath_full_count, bath_half_count, sale_instrament all cross zero, so are not the best predictors since sometimes the predictor has a negative relationship, and other samples it has a positive relationship.

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_3 <- lm(housing_df$‘Sale Price‘ ~ sq_ft_lot + square_feet_total_living + year_built + building_gra
anova(model_1, model_3)
```

```
## Analysis of Variance Table
##
## Model 1: housing_df$‘Sale Price‘ ~ sq_ft_lot
## Model 2: housing_df$‘Sale Price‘ ~ sq_ft_lot + square_feet_total_living +
##     year_built + building_grade
##   Res.Df        RSS Df  Sum of Sq    F    Pr(>F)
## 1  12863 2.0734e+15
## 2  12860 1.6339e+15  3 4.3946e+14 1153 < 2.2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

The value of F is 1153 and Pr(>F) is 2.2e-16; we can say that model 2 significantly improved the fit of the model to the data compared to model 1.

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_df$residuals<-resid(model_3)
housing_df$standardized.residuals<- rstandard(model_3)
housing_df$studentized.residuals<-rstudent(model_3)
housing_df$cooks.distance<-cooks.distance(model_3)
housing_df$dfbeta<-dfbeta(model_3)
housing_df$dffit<-dffits(model_3)
housing_df$leverage<-hatvalues(model_3)
housing_df$covariance.ratios<-covratio(model_3)
```

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_df$large.residual <- housing_df$standardized.residuals > 2 | housing_df$standardized.residuals
```

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

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

```
## [1] 334
```

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

```
housing_df[housing_df$large.residual,c("Sale Price", "sq_ft_lot", "square_feet_total_living", "year_buil
```

```
## # A tibble: 334 x 6
##    `Sale Price` sq_ft_lot square_feet_tot~ year_built building_grade
##           <dbl>     <dbl>            <dbl>      <dbl>          <dbl>
##  1       265000    112650             4920       2007             10
##  2      1390000    225640              660       1955              6
##  3       229000    236966             3840       2008             10
##  4       390000     63162             5800       2008             11
##  5      1588359      8752             3360       2005              9
##  6      1450000     14043             3480       1972              8
##  7      1450000     14043              900       1918              6
##  8       163000     18498             4710       2014              9
##  9       270000     89734             5060       2016             11
## 10       200000    288367             6880       2008             10
## # ... with 324 more rows, and 1 more variable: standardized.residuals <dbl>
```

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

```
housing_df[housing_df$large.residual , c("cooks.distance", "leverage", "covariance.ratios")]
```

```
## # A tibble: 334 x 3
##    cooks.distance leverage covariance.ratios
##             <dbl>    <dbl>             <dbl>
##  1       0.000719 0.000628             0.999
##  2       0.00340  0.00176              0.998
##  3       0.00132  0.00140              1.00
##  4       0.00111  0.000925             0.999
##  5       0.000143 0.000157             0.999
##  6       0.000393 0.000480             0.999
##  7       0.00423  0.00161              0.997
##  8       0.000842 0.000686             0.999
##  9       0.000937 0.000708             0.999
## 10       0.00771  0.00312              0.999
## # ... with 324 more rows
```

For cooks, you are looking for a cook's distance of > 1. For leveragwe, you are looking for values either twice or three times as large as the average For covariance ratios, we are looking for any cases that deviate substantially the covariance boundaries

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

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: ’zoo’
```

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

```
dwtest(model_3)
```

```
##
##  Durbin-Watson test
##
## data:  model_3
## DW = 0.54148, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

The closer to 2 that the value is, the better. In this case, the value is 0.54148, which is not close to 2 and may be a cause of concern. The p-value of < 2.2e-16 means it is significant and confirms this conclusion. The condition is not met.

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
```

```
mean(vif(model_3))
```

```
## [1] 1.761031
```

```
vif(model_3)
```

```
##                sq_ft_lot square_feet_total_living               year_built
##                 1.113594                 2.365744                 1.211716
##           building_grade
##                 2.353068
```

```
1/vif(model_3)
```

```
##                sq_ft_lot square_feet_total_living               year_built 
##                0.8979930                0.4226999                0.8252756 
##           building_grade 
##                0.4249772
```

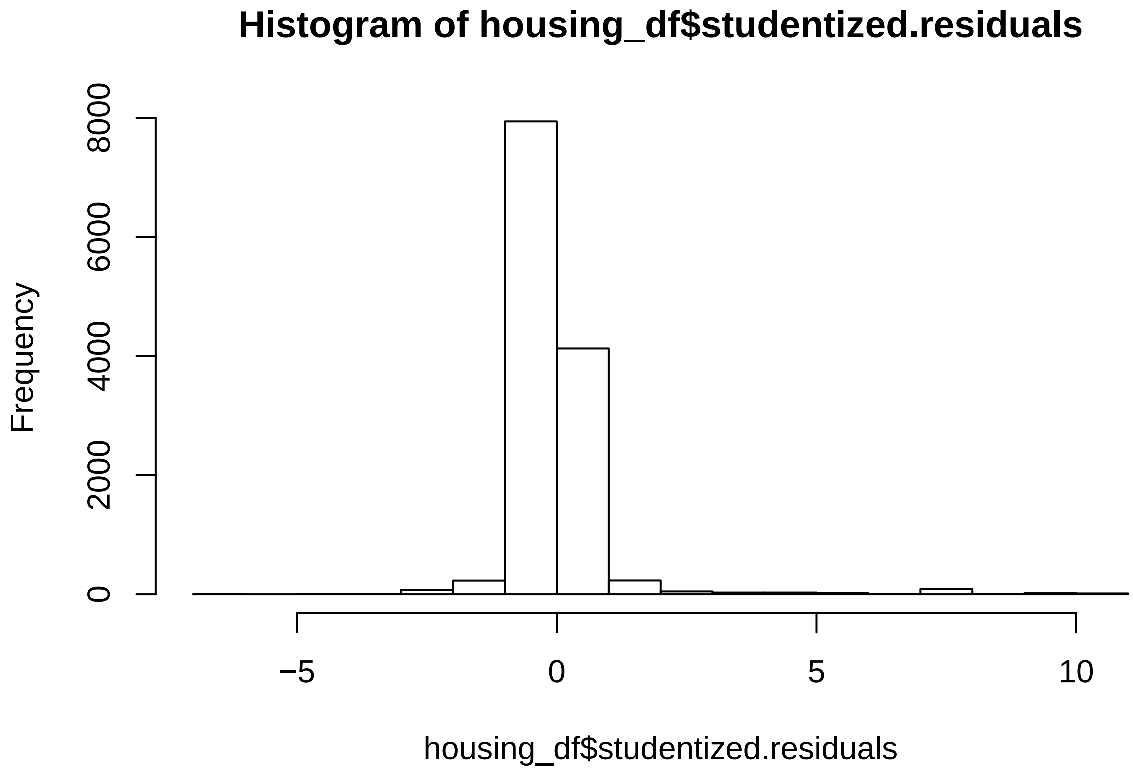
```
mean(vif(model_3))
```

```
## [1] 1.761031
```

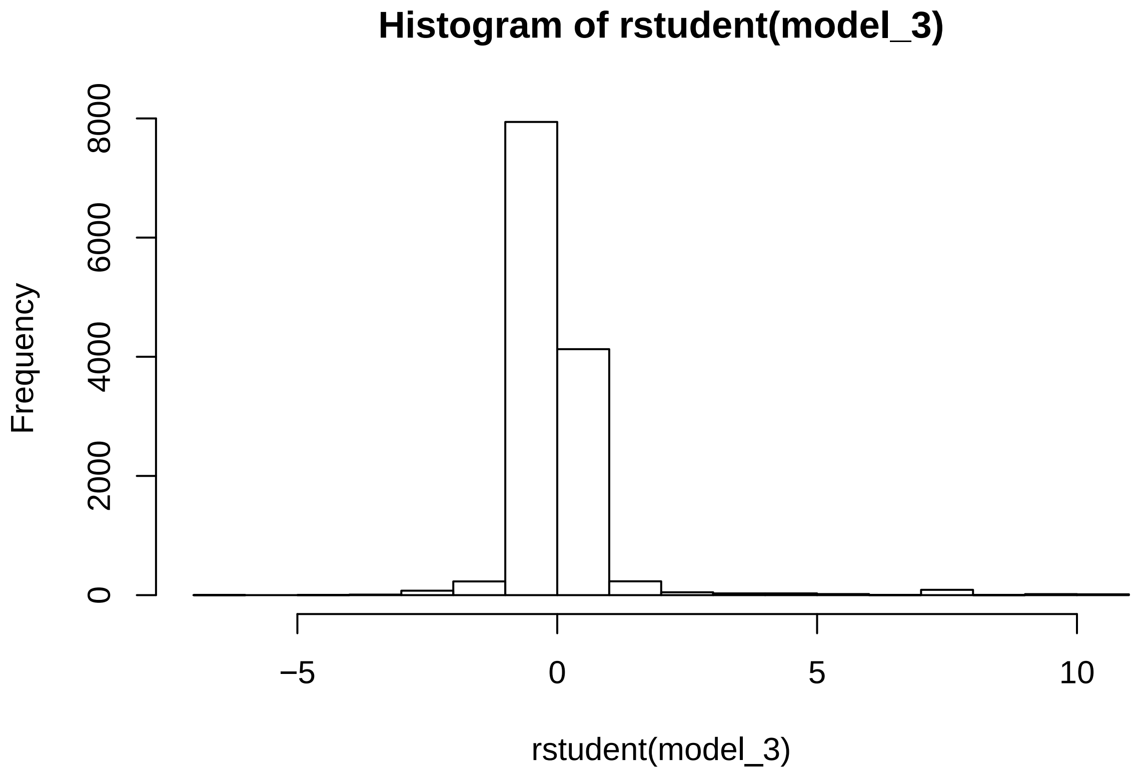
The state of the condition is met. Based on these measures we can safely conclude that there is no collinearity within our data.

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.

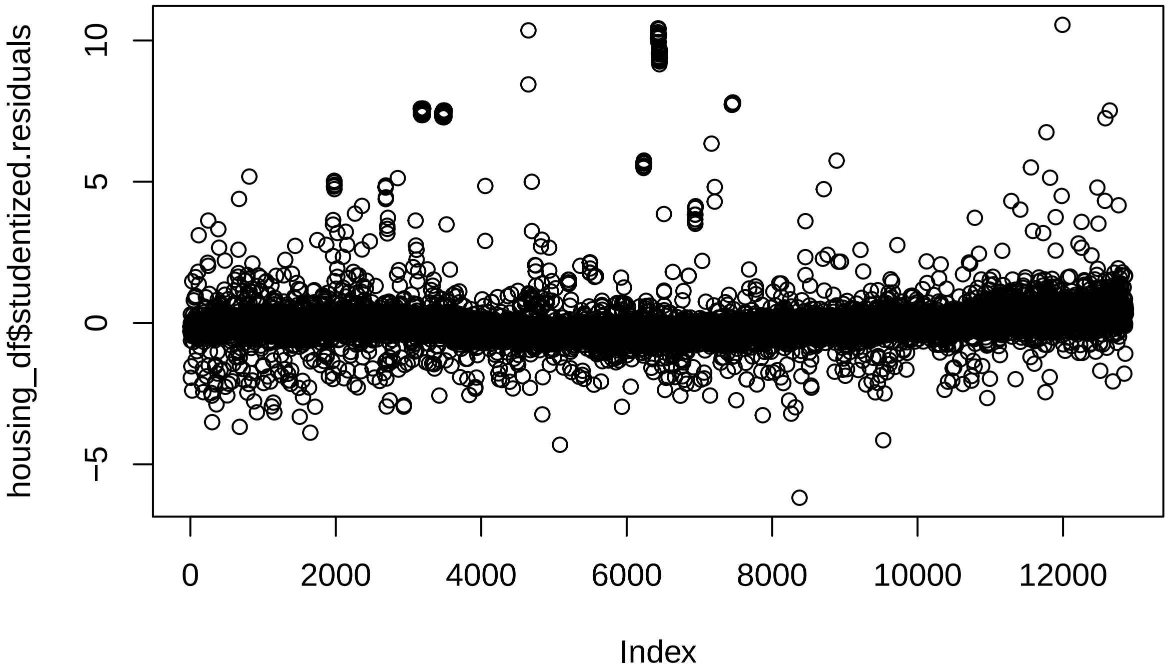
```
hist(housing_df$studentized.residuals)
```



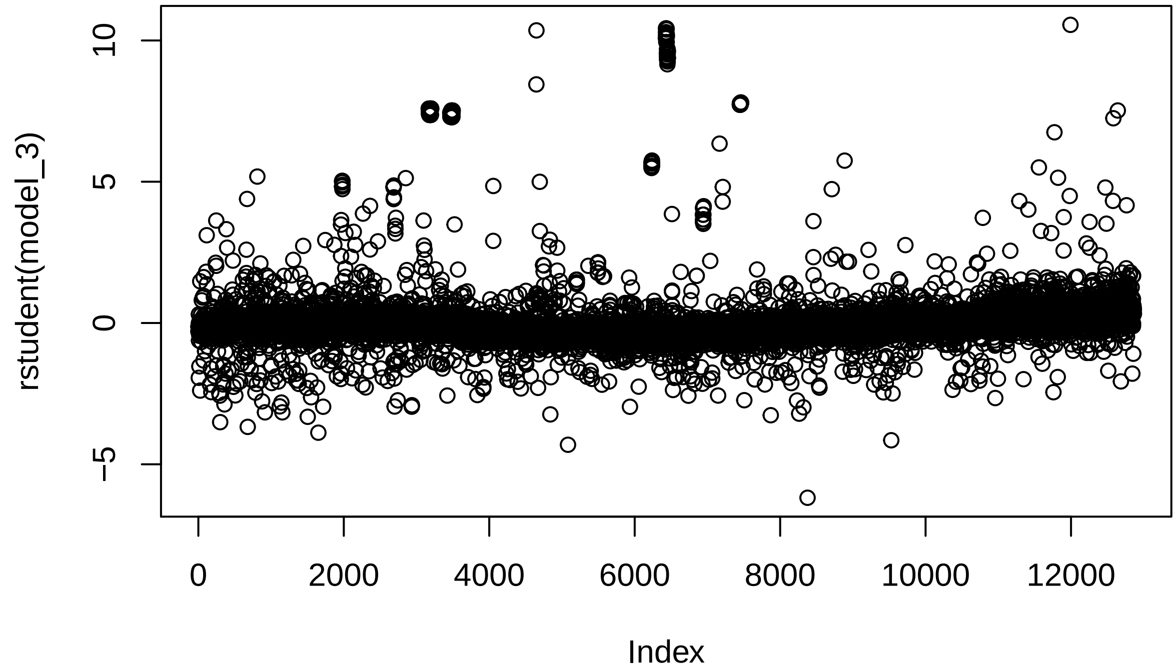
```
hist(rstudent(model_3))
```



```r
plot(housing_df$studentized.residuals)
```



```
plot(rstudent(model_3))
```



data looks normal. It does not look like the data have violated the assumption of linearity .

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

The model does not appear to be biased. The sample is a good represention of the entire population.