Housing Project

Taylor Anderson

7/14/2019

a.

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

```
housing_clean <- na.omit(subset(housing, select =c(Sale.Price,
square_feet_total_living, bedrooms, bath_full_count, bath_half_count,
bath_3qtr_count, year_built, year_renovated)))
```

I removed many non deterministic values such as address and sale type. I have 8 variables remaining: price, squarefeet, year built, year renovated, bedrooms, and several bathroom variables.

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.

```
sqfeet_lm <- lm(Sale.Price ~ square_feet_total_living, housing_clean)
other_lm <- lm(Sale.Price ~ square_feet_total_living + year_built + bedrooms
+ bath_full_count, housing_clean)
```

I decided to add year built, bathrooms and bedrooms to the second model because they seem important to the price of a house. I am worried that bathrooms and bedrooms will have multicolinearity with square feet though.

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?

```
summary(sqfeet_lm)

##
## Call:
## lm(formula = Sale.Price ~ square_feet_total_living, data = housing_clean)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1800136  -120257   -41547    44028  3811745
##
```

```
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                 1.891e+05  8.745e+03   21.62   <2e-16 ***
## square_feet_total_living 1.857e+02  3.208e+00   57.88   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 360200 on 12863 degrees of freedom
## Multiple R-squared:  0.2066, Adjusted R-squared:  0.2066 
## F-statistic:  3351 on 1 and 12863 DF,  p-value: < 2.2e-16
```

```r
summary(other_lm)
```

```
## 
## Call:
## lm(formula = Sale.Price ~ square_feet_total_living + year_built + 
##     bedrooms + bath_full_count, data = housing_clean)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1719151  -120511   -42398    45744  3904824 
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                 -4.430e+06  4.195e+05 -10.559  < 2e-16 ***
## square_feet_total_living  1.744e+02  4.423e+00  39.424  < 2e-16 ***
## year_built                  2.340e+03  2.117e+02  11.053  < 2e-16 ***
## bedrooms                    -1.375e+04  4.517e+03  -3.045  0.00234 ** 
## bath_full_count             1.730e+04  6.095e+03   2.838  0.00454 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 357300 on 12860 degrees of freedom
## Multiple R-squared:  0.2194, Adjusted R-squared:  0.2192 
## F-statistic: 903.7 on 4 and 12860 DF,  p-value: < 2.2e-16
```

With only square feet the model was able to account for 20.7% of the variability in the data. By adding the other two variables we increased to 22%.

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?

```r
library(lm.beta)
lm.beta(other_lm)
```

```
## 
## Call:
## lm(formula = Sale.Price ~ square_feet_total_living + year_built + 
##     bedrooms + bath_full_count, data = housing_clean)
```

```
##
## Standardized Coefficients::
##              (Intercept) square_feet_total_living               year_built
##               0.00000000               0.42677620               0.09966661
##                 bedrooms          bath_full_count
##              -0.02979645               0.02783809
```

The values of standardized betas tells us that the number of square feet a house has the highest importance in determaining the cost. Year built is somewhat important and bedrooms and bathrooms are far behind that.

e.

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

```
confint(other_lm)

##                                  2.5 %        97.5 %
## (Intercept)              -5252187.0182 -3607553.8714
## square_feet_total_living      165.6868      183.0244
## year_built                   1925.4259     2755.5146
## bedrooms                   -22607.0742    -4898.3341
## bath_full_count              5351.3294    29243.8018
```

These confidence intervals show us that we are 95% certain that the true betas of each of these variables fall between these values. It is clear that there is multicollinearity with bedrooms unfortunetly.

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(sqfeet_lm, other_lm)

## Analysis of Variance Table
##
## Model 1: Sale.Price ~ square_feet_total_living
## Model 2: Sale.Price ~ square_feet_total_living + year_built + bedrooms +
##     bath_full_count
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
## 1  12863 1.6689e+15
## 2  12860 1.6420e+15  3 2.6831e+13 70.045 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The anova table shows us that the multiple regression is significantly better than the simple regression model.

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.

```
other_lm_diag <- subset(housing_clean, select = c(Sale.Price,
square_feet_total_living, year_built, bedrooms, bath_full_count))
other_lm_diag$residuals <- resid(other_lm)
other_lm_diag$standardized.residuals <- rstandard(other_lm)
other_lm_diag$studentized.residuals <- rstudent(other_lm)
other_lm_diag$cooks.distance <- cooks.distance(other_lm)
other_lm_diag$dfbeta <- dfbeta(other_lm)
other_lm_diag$dffit <- dffits(other_lm)
other_lm_diag$leverage <- hatvalues(other_lm)
other_lm_diag$covariance.ratios <- covratio(other_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.

```
other_lm_diag$large.residuals <- other_lm_diag$standardized.residuals >
2|other_lm_diag$standardized.residuals < -2
```

i.

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

```
sum(other_lm_diag$large.residuals)

## [1] 329
```

j.

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

```
head(other_lm_diag[other_lm_diag$large.residuals, c("Sale.Price",
"square_feet_total_living", "year_built", "bedrooms", "bath_full_count",
"standardized.residuals")])

##     Sale.Price square_feet_total_living year_built bedrooms
## 6       184667                     4160       2005        4
## 25      265000                     4920       2007        4
## 115    1390000                      660       1955        0
## 178     390000                     5800       2008        5
## 239    1588359                     3360       2005        2
## 246    1450000                      900       1918        2
##     bath_full_count standardized.residuals
## 6                 2              -2.191642
## 25                4              -2.448659
## 115               1               3.114499
## 178               4              -2.496458
## 239               2               2.051065
## 246               1               3.485119
```

Only showing the first 6 examples as the full data frame is 329 rows.

k.

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

```
high_res <- other_lm_diag[other_lm_diag$large.residuals, c("cooks.distance",
"leverage", "covariance.ratios")]

sum(high_res$cooks.distance > 0.5)

## [1] 0
```

There are no cooks distances greater than 1, or even 0.5. This means that there are no points that would greatly alter results if removed.

```
average_leverage = (4 + 1)/12865
sum(high_res$leverage > average_leverage * 2)

## [1] 98

sum(high_res$leverage > average_leverage * 3)

## [1] 65
```

There are 98 observations more than double the average leverage and 65 over triple the average leverage.

```
sum((high_res$covariance.ratios > 1 + (3*(4 +
1)/12865))|(high_res$covariance.ratios < 1 - (3*(4 + 1)/12865)))

## [1] 262
```

There are 262 observations outside of the standard range of covariance ratios.

l.

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

```
library(car)

## Warning: package 'car' was built under R version 3.6.1

## Loading required package: carData

dwt(other_lm)

##  lag Autocorrelation D-W Statistic p-value
##    1       0.7210338     0.5579232       0
##  Alternative hypothesis: rho != 0
```

This model does have positive autocorrelation.

m.

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

```
vif(other_lm)
```

```
## square_feet_total_living                 year_built                   bedrooms
##                 1.930570                   1.339428                   1.577994
##           bath_full_count
##                 1.584923
```

```
1/vif(other_lm)
```

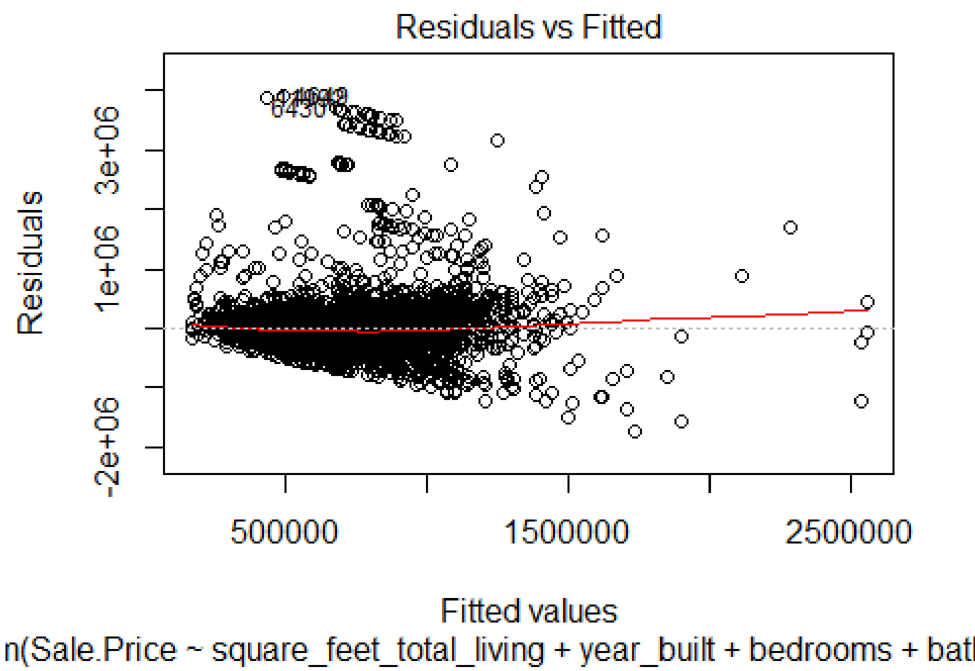
```
## square_feet_total_living                 year_built                   bedrooms
##                0.5179818                  0.7465875                  0.6337161
##           bath_full_count
##                0.6309454
```

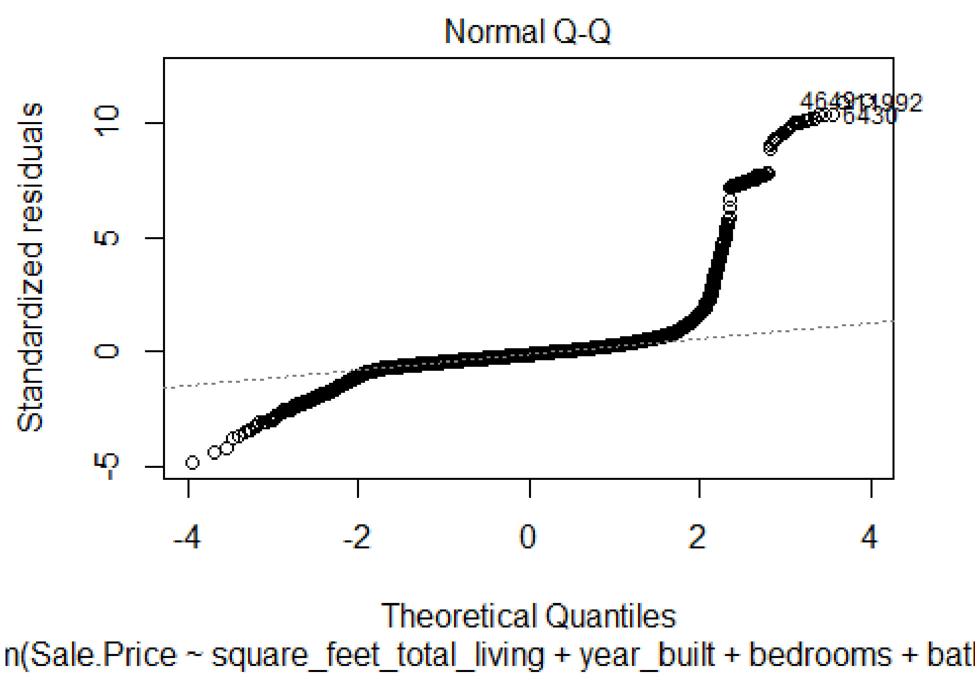
There does not appear to be an issue with multicollinearity.

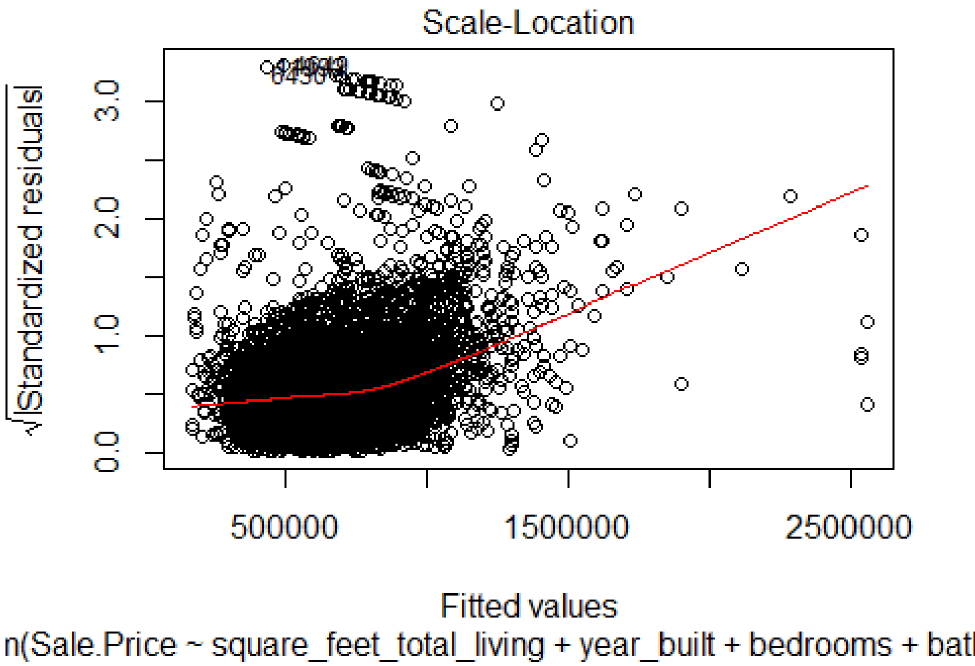
m.

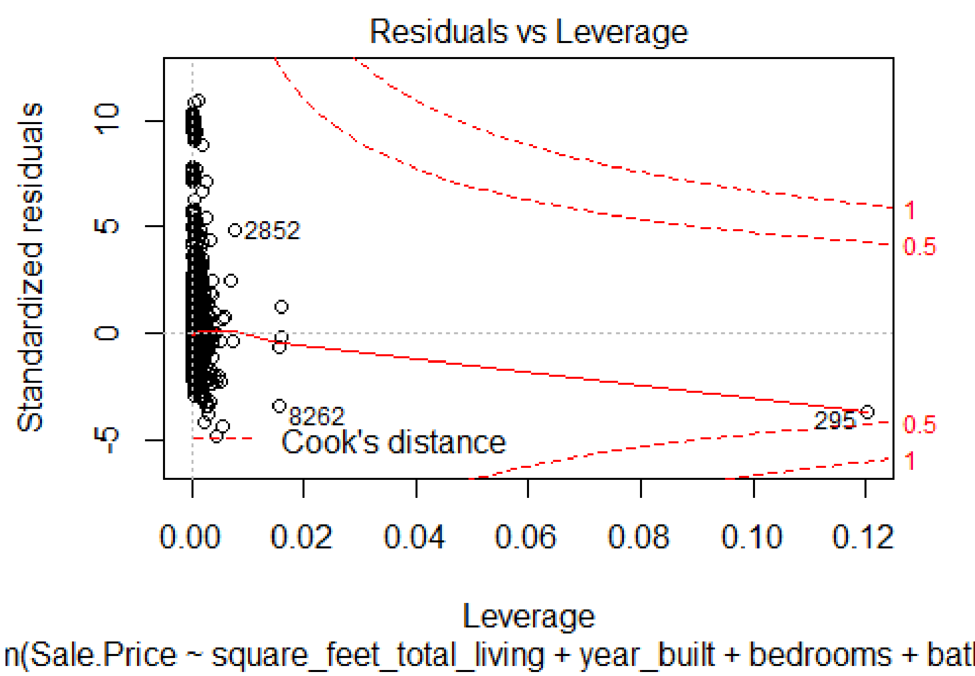
Visually check the assumptions related to the residuals using the plot() and hist() functions. Summarize what each graph is informing you of and if any anomalies are present.

```
plot(other_lm)
```


Residuals vs Fitted
Residuals
-2e+06
1e+06
3e+06
6430
500000
1500000
2500000
Fitted values
n(Sale.Price ~ square_feet_total_living + year_built + bedrooms + batl


Normal Q-Q
Standardized residuals
-5
0
5
10
6430
-4
-2
0
2
4
Theoretical Quantiles
n(Sale.Price ~ square_feet_total_living + year_built + bedrooms + batl



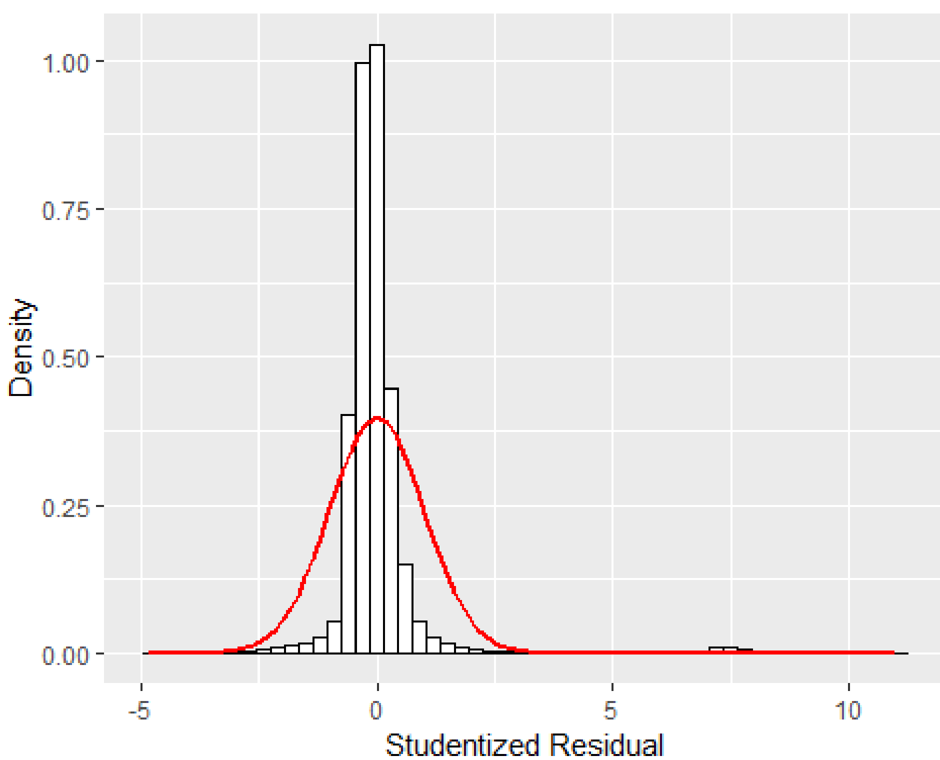


Residuals vs Fitted shows that the model may be heteroscedastic. Normal Q-Q shows that the data has heavy tails and has more values at extremes than we would expect with a normal distribution Scale-location shows that residuals are not spread evenly, especially in the more expensive homes. Residual vs leverage shows that there is no influencing case though case 295 is close.

```
library(ggplot2)

## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang

histogram <- ggplot(other_lm_diag, aes(studentized.residuals)) +
geom_histogram(aes(y = ..density..), color = "black", fill = "white",
binwidth = .3) +
  labs(x = "Studentized Residual", y = "Density")
histogram + stat_function(fun=dnorm, args = list(mean =
mean(other_lm_diag$studentized.residuals, na.rm = TRUE),
                                                 sd =
sd(other_lm_diag$studentized.residuals, na.rm = TRUE)), color = "red", size =
1)
```



The residuals are not very normal. This histogram has some residuals further in the tail than we would expect and it is very narrow otherwise.

o.

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

This model does seem to have bias and may not be a good model to use against another population.