Multiple Regression with Housing Data

Demond Love

Using statistical correlation, multiple regression and R programming, I will investigate the following variables: Sale Price and several other possible predictors.

```
library(readxl)
library(ggplot2)
library(pastecs)
library(ggm)
library(lm.beta)
library(lmtest)
library(QuantPsyc)
library(car)
setwd('/Users/Love/Documents/Projects')
file = read_excel("./housing-data.xlsx")
```

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

```
data = data.frame(file$`Sale Price`, file$sq_ft_lot)
names(data) = c("sale_price", "sq_footage")
simpledataset = data[!(data$sale_price <= 14000 & data$sq_footage > 50000),]
str(simpledataset)
```

```
## 'data.frame':    12850 obs. of  2 variables:
##  $ sale_price: num  698000 649990 572500 420000 369900 ...
##  $ sq_footage: num  6635 5570 8444 9600 7526 ...
```

I identified these outliers, in which the price of the home was less than $14,001, but the home had more than 50,000 sqare feet. These cases were removed from the dataset, since there must be a data entry error (+50,000 square foot properties will almost never sell for this small amount), there are a few of them so don't they drastically impact the underlying integrity of the sample, and I beleive these not to be from the population that we intended for this sample.

Creating two variables; one that will contain the variables Sale Price and Square Foot of Lot and another with several additional predictors of your choice.

```
multipledataset = data.frame(file$`Sale Price`, file$sq_ft_lot, file$bedrooms, file$bath_full_count)
names(multipledataset) = c("sale_price", "sq_footage", "bedroom", "fullbath")
multipledataset = multipledataset[!(data$sale_price <= 14000 & data$sq_footage > 50000),]
str(multipledataset)
```

```
## 'data.frame':    12850 obs. of  4 variables:
##  $ sale_price: num  698000 649990 572500 420000 369900 ...
##  $ sq_footage: num  6635 5570 8444 9600 7526 ...
##  $ bedroom   : num  4 4 4 3 3 4 5 4 4 4 ...
##  $ fullbath  : num  2 2 1 1 1 2 3 2 2 1 ...
```

I have chosen to include number of bedrooms and the number of full bathrooms to my dataset for comparison analysis.

Executing the summary() function on the two models and analyzing the result.

The value of R-squared is a measure of how much of the variability in the outcome is accounted for by the predictors.

The adjusted R-squared gives us some idea of how well our model generalizes, and ideally we would like its value to be the same, or very close to, the value of R-squared.

```
housingsimplemodel = lm(sale_price ~ sq_footage, simpledataset)
summary(housingsimplemodel)
```

```
##
## Call:
## lm(formula = sale_price ~ sq_footage, data = simpledataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1351982  -193383   -63070    91469  3738065
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.379e+05  3.813e+03  167.27   <2e-16 ***
## sq_footage  1.085e+00  6.649e-02   16.32   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 399900 on 12848 degrees of freedom
## Multiple R-squared:  0.0203, Adjusted R-squared:  0.02023
## F-statistic: 266.3 on 1 and 12848 DF,  p-value: < 2.2e-16
```

```
housingmultimodel = lm(sale_price ~ sq_footage + bedroom + fullbath, multipledataset)
summary(housingmultimodel)
```

```
##
## Call:
## lm(formula = sale_price ~ sq_footage + bedroom + fullbath, data = multipledataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3604036  -152415   -50420    69658  3739026
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.381e+05  1.477e+04   9.348   <2e-16 ***
## sq_footage  9.564e-01  6.314e-02  15.146   <2e-16 ***
## bedroom     6.862e+04  4.010e+03  17.112   <2e-16 ***
## fullbath    1.468e+05  5.405e+03  27.155   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 379200 on 12846 degrees of freedom
## Multiple R-squared:  0.1192, Adjusted R-squared:  0.119
## F-statistic: 579.6 on 3 and 12846 DF,  p-value: < 2.2e-16
```

Whereas the R-squared for the simple linear regression model is 0.02 and the adjusted R-squared is also 0.02. Therefore, only 2% of the variability is explained with the by the single predictor of this model.

On the other hand, the R-squared for the multiple regression model is 0.12 and the adjusted R-squared is also 0.12. Therefore, about 6x of the variability is explained with by the predictors of this model.

Considering the parameters of the multiple regression model, exploring the standardized betas for each parameter.

```
lm.beta(housingmultimodel)
```

```
## sq_footage    bedroom   fullbath
##  0.1256045  0.1487875  0.2360901
```

These are the standardized regression coefficients, which are not dependent on the units of measurement of the variables. These estimates tell us the number of standard deviations by which the outcome will change as a result of one standard devisation change in the predictor. The standardized beta values are all measured in standard deviation units and so are directly comparable: therefore, they provide a better insight into the 'importance' of a predictor in the model.

Here, you can clearly see a deligniation of variables that are chosen. The number of full bathrooms are by far the most important, followed by the number of bedrooms, and the square footage falling well behind. This is a little surprising, since I would have thought the number of bedrooms would have lead the path in order of importance.

Calculating the confidence intervals for the parameters in my model.

```
confidenceintervals = confint(housingmultimodel)
confidenceintervals
```

```
##                    2.5 %       97.5 %
## (Intercept) 1.091043e+05 1.670014e+05
## sq_footage  8.325929e-01 1.080128e+00
## bedroom     6.076186e+04 7.648345e+04
## fullbath    1.361740e+05 1.573629e+05
```

The number one thing we are looking for here is if the confidence interval crosses 0. If one is positive and another is negative, then that means that we have a bad model. However, all four predictors have a positive relationship, which doesn't cross zero, which means that these are all statistically significant.

Assessing the improvement of the models by testing whether the second model is significant by performing an analysis of variance.

```
comparingmodels = anova(housingsimplemodel, housingmultimodel)
comparingmodels
```

```
## Analysis of Variance Table
##
## Model 1: sale_price ~ sq_footage
## Model 2: sale_price ~ sq_footage + bedroom + fullbath
##   Res.Df        RSS Df  Sum of Sq      F    Pr(>F)
## 1  12848 2.0545e+15
## 2  12846 1.8471e+15  2 2.0742e+14 721.29 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here, you can see that the RSS for the second model is lower than the first, meaning it is a better model. However, the problem with RSS is that it always improves when an additional variable is added to the model.

Performing casewise diagnostics to identify outliers and/or influential cases, storing each function's output in a dataframe assigned to a unique variable name and calculating the standardized residuals using the appropriate command, specifying those that are +-2, then storing the results of large residuals in a variable.

```
multipledataset$standardized.residuals = rstandard(housingmultimodel)
multipledataset$large.residuals = multipledataset$standardized.residuals > 2 | multipledataset$standardi
```

Showing the sum of large residuals.

```
sum(multipledataset$large.residuals)
```

```
## [1] 327
```

Grabbing the specific variables that have large residuals?

```
LARGES = subset(multipledataset, large.residuals == TRUE)
```

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

Here we are looking for Cook's distances above 1, leverage values that are twice the average leverage, and cases falling outside of the upper and lower CVR limits (defined as 1 plus three times the average leverage, whereas the lower limit is 1 minus three times the average leverage).

```
multipledataset$cooks.distance = cooks.distance(housingmultimodel)
cooksdistanceoutliers = which(multipledataset$cooks.distance > 1)
multipledataset[cooksdistanceoutliers,]
```

```
##     sale_price sq_footage bedroom fullbath standardized.residuals
## 295     270000      89734       4       23              -9.965287
##     large.residuals cooks.distance
## 295            TRUE        2.46568
```

This is a property with 23 full bathrooms and only 4 bedrooms. This clearly is not a residential property, and therefore is not a part of our desired population.

```
multipledataset$leverage = hatvalues(housingmultimodel)
```

```
multipledataset$covariance.ratios = covratio(housingmultimodel)
```

Performing the necessary calculations to assess the assumption of independence.

```
durbinWatsonTest(housingmultimodel)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.6528159     0.6943671       0
##  Alternative hypothesis: rho != 0
```

No, it did not meet the condition. Values less than 1 or greater than 3 should definitely raise alarm bells. THe closer to 2 that the value is, the better, but my value is 0.69.

Performing the necessary calculations to assess the assumption of no multicollinearity.

```
vif(housingmultimodel)
```

```
## sq_footage    bedroom   fullbath
##   1.002999   1.102681   1.102468
```

The largest VIF is not greater than 10, so there is no cause for concern.

```
1/vif(housingmultimodel)
```

```
## sq_footage    bedroom   fullbath
##  0.9970096  0.9068807  0.9070560
```

Neither of the tolerances are below 0.2 or 0.1, so there is no cause for concern.

```
mean(vif(housingmultimodel))
```

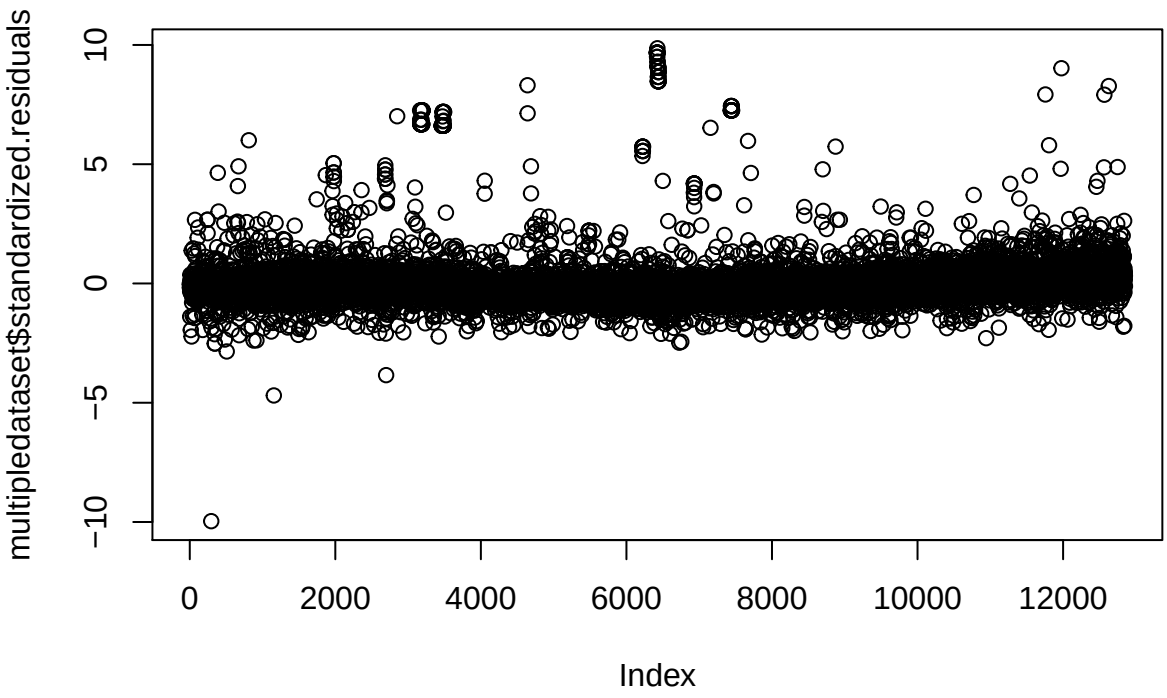
```
## [1] 1.069383
```

Lastly, the average VIF is greater than 1, but isn't substantially greater than 1.

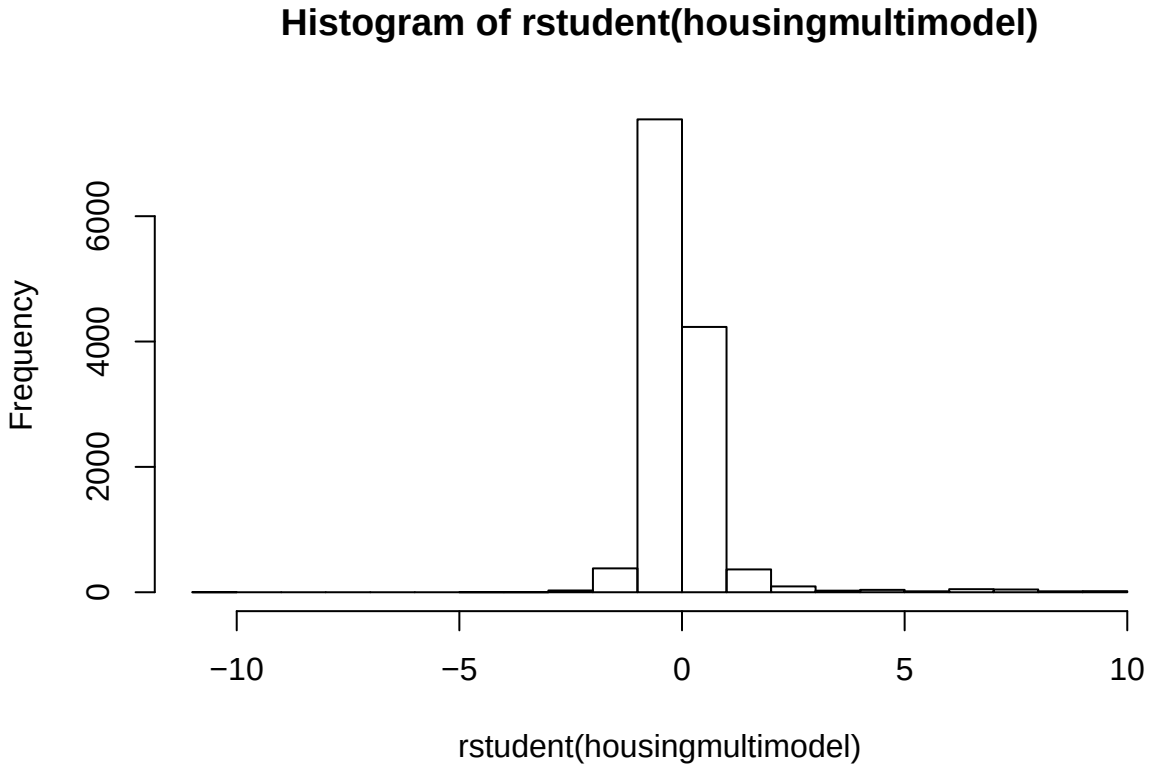
Based on these measures we can safely conclude that there is no collinearity within our data.

Visually checking the assumptions related to the residuals using the plot() and hist() functions instead of ggplot.

```
plot(multipledataset$standardized.residuals)
```




```
hist(rstudent(housingmultimodel))
```



This is a non-normal distribution of studentized residuals.

Based on the above analysis, this model is bias and doesn't do an effective job of accuratly reresent the sample or the population as a whole.