Housing Data

Burden Joshua

require(devtools)

## Loading required package: devtools

## Loading required package: usethis

install\_version("QuantPsyc", version="1.5", repos="http://cran.us.r-project.org")

## Downloading package from url: http://cran.us.r-project.org/src/contrib/Archive/QuantPsyc/QuantPsyc\_1.5.tar.gz

#install.packages("https://cran.r-project.org/src/contrib/Archive/fArma/fArma\_3042.81.tar.gz", repos = , type = "source")

## Set workding directory to read source datasets.  
setwd("/Users/joshua/Documents/PERSONAL\_GITHUB\_REPOS/dsc520/Housing")  
## Read housing dataset  
housing\_data\_df <- read\_excel("../data/week-7-housing.xlsx")

# Explain any transformations or modifications you made to the dataset

#### Sales Date and Sales Price should conform to the xxx\_xxx style like the other column names

colnames(housing\_data\_df)[c(1,2)]<-c("sales\_date", "sale\_price")  
colnames(housing\_data\_df)

## [1] "sales\_date" "sale\_price"   
## [3] "sale\_reason" "sale\_instrument"   
## [5] "sale\_warning" "sitetype"   
## [7] "addr\_full" "zip5"   
## [9] "ctyname" "postalctyn"   
## [11] "lon" "lat"   
## [13] "building\_grade" "square\_feet\_total\_living"  
## [15] "bedrooms" "bath\_full\_count"   
## [17] "bath\_half\_count" "bath\_3qtr\_count"   
## [19] "year\_built" "year\_renovated"   
## [21] "current\_zoning" "sq\_ft\_lot"   
## [23] "prop\_type" "present\_use"

# 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.

housing\_data\_df\_lm1 <- lm(formula = sale\_price ~ sq\_ft\_lot, data = housing\_data\_df)  
housing\_data\_df\_lm2 <- lm(formula = sale\_price ~ zip5 + bedrooms + year\_built + square\_feet\_total\_living, sq\_ft\_lot, data = housing\_data\_df)

#### Answer

* data includes items columns such as zip5, and bedrooms as these items often are major factors in predicting the sale price of a home
* additional parameters that were chosen were the square\_feet\_total\_living and the year built.
* Houses should be go up in price based on the amount of area it covers and potentially when it was built

# Execute a summary() function on two variables defined in the previous step to compare the model results.

##### sumamary housing\_data\_df\_lm1

summary(housing\_data\_df\_lm1)

##   
## Call:  
## lm(formula = sale\_price ~ sq\_ft\_lot, data = housing\_data\_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

##### sumamary housing\_data\_df\_lm2

summary(housing\_data\_df\_lm2)

##   
## Call:  
## lm(formula = sale\_price ~ zip5 + bedrooms + year\_built + square\_feet\_total\_living,   
## data = housing\_data\_df, subset = sq\_ft\_lot)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1466706 -115191 -42005 34341 3727315   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.315e+08 2.496e+08 -0.527 0.59831   
## zip5 1.298e+03 2.546e+03 0.510 0.61004   
## bedrooms -1.324e+04 4.997e+03 -2.650 0.00806 \*\*   
## year\_built 2.203e+03 2.168e+02 10.161 < 2e-16 \*\*\*  
## square\_feet\_total\_living 1.808e+02 4.631e+00 39.045 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 350000 on 9709 degrees of freedom  
## (3151 observations deleted due to missingness)  
## Multiple R-squared: 0.2222, Adjusted R-squared: 0.2219   
## F-statistic: 693.4 on 4 and 9709 DF, p-value: < 2.2e-16

# What are the R2 and Adjusted R2 statistics? Explain what these results tell you about the overall model.

#### answer

* R2 for housing\_data\_df\_lm1: 0.01 while adjusted: 0.01
* R2 for housing\_data\_df\_lm2: 0.11 while adjusted: 0.11
* Rsquared for first variable: 0.01435
* adjusted Rsquared for the first variable: 0.01428
* Rsquared for second variable: 0.2222
* adjusted Rsquared for the second variable: 0.02219

# Did the inclusion of the additional predictors help explain any large variations found in Sale Price?

the overall model improved when adding addtional variables

# 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?

lm.beta(housing\_data\_df\_lm1)

## sq\_ft\_lot   
## 0.1198122

lm.beta(housing\_data\_df\_lm2)

## zip5 bedrooms year\_built   
## 0.00463955 -0.02898933 0.09674136   
## square\_feet\_total\_living   
## 0.44729452

The standardized betas for the linear models indicate that the sale\_price increased by 0.11 standard deviations when there is an increase in st

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

confint(housing\_data\_df\_lm1)

## 2.5 % 97.5 %  
## (Intercept) 6.343730e+05 6.492698e+05  
## sq\_ft\_lot 7.291208e-01 9.728641e-01

confint(housing\_data\_df\_lm2)

## 2.5 % 97.5 %  
## (Intercept) -6.207217e+08 3.577402e+08  
## zip5 -3.691807e+03 6.288624e+03  
## bedrooms -2.303836e+04 -3.447748e+03  
## year\_built 1.778314e+03 2.628393e+03  
## square\_feet\_total\_living 1.717445e+02 1.899005e+02

The confidence intervals calculated for “housing\_data\_df\_lm1” have a small range. This indicates that the predictor’s *b* value is close to the real *b* value. The confidence intervals calculated for “housing\_data\_df\_lm2” have a larger range. In addition, these values cross zero and include negative values. This indicates that the sale price can increase or decrease depending on the number of bedrooms. This makes the output for sale price not consistent. However, the other variables do have better consistency and shorter range.

# 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.

model1\_aov <- aov(`sale\_price`~sq\_ft\_lot, data = housing\_data\_df)  
summary(model1\_aov)

## Df Sum Sq Mean Sq F value Pr(>F)   
## sq\_ft\_lot 1 3.020e+13 3.020e+13 187.3 <2e-16 \*\*\*  
## Residuals 12863 2.073e+15 1.612e+11   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

model2\_aov <- aov(`sale\_price` ~ `sales\_date` + sq\_ft\_lot + bedrooms + bath\_full\_count + year\_built, data = housing\_data\_df)  
summary(model2\_aov)

## Df Sum Sq Mean Sq F value Pr(>F)   
## sales\_date 1 7.263e+12 7.263e+12 51.9 6.16e-13 \*\*\*  
## sq\_ft\_lot 1 2.983e+13 2.983e+13 213.2 < 2e-16 \*\*\*  
## bedrooms 1 9.945e+13 9.945e+13 710.7 < 2e-16 \*\*\*  
## bath\_full\_count 1 1.062e+14 1.062e+14 759.0 < 2e-16 \*\*\*  
## year\_built 1 6.147e+13 6.147e+13 439.3 < 2e-16 \*\*\*  
## Residuals 12859 1.799e+15 1.399e+11   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 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.

saleprice\_var <- data.frame(std.residuals=rstandard(housing\_data\_df\_lm2),  
 stud.residuals=rstudent(housing\_data\_df\_lm2),  
 cooks.dist=cooks.distance(housing\_data\_df\_lm2),  
 dfbeta=dfbeta(housing\_data\_df\_lm2),  
 dffit=dffits(housing\_data\_df\_lm2),  
 leverage=hatvalues(housing\_data\_df\_lm2),  
 covariance.ratios=covratio(housing\_data\_df\_lm2))

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

saleprice\_var$large\_residual <- saleprice\_var$std.residuals > 2 | saleprice\_var$std.residuals < -2

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

sum(saleprice\_var$large\_residual)

## [1] 261

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

large\_res <- filter(saleprice\_var, saleprice\_var$large\_residual == TRUE)

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

## leverage cooks.dist covariance.ratios  
## 7507 0.0009497935 0.0012879288 0.9979778  
## 4695 0.0004797951 0.0026369874 0.9869160  
## 8262 0.0151048536 0.0418036288 1.0087496  
## 7210 0.0008245121 0.0033131929 0.9910311  
## 3170 0.0003744344 0.0043839059 0.9710883  
## 4834 0.0018748237 0.0034212961 0.9977020  
## 4834.1 0.0018748237 0.0034212961 0.9977020  
## 7650 0.0005575413 0.0004619717 0.9989405  
## 3491 0.0005107812 0.0057475722 0.9723710  
## 6442 0.0002784936 0.0059530605 0.9469228  
## 5491 0.0002695332 0.0002931919 0.9979856  
## 3169 0.0005460855 0.0066127306 0.9702515  
## 4740 0.0004142166 0.0004257958 0.9982843  
## 4750 0.0031762435 0.0028619454 1.0013840  
## 3200 0.0005460855 0.0066127306 0.9702515  
## 4435 0.0005274343 0.0005421220 0.9983979  
## 3496 0.0005460855 0.0064918108 0.9708080  
## 7210.1 0.0008245121 0.0033131929 0.9910311  
## 6451 0.0003419521 0.0067802188 0.9508077  
## 6451.1 0.0003419521 0.0067802188 0.9508077  
## 12816 0.0010330831 0.0013334269 0.9982290  
## 7210.2 0.0008245121 0.0033131929 0.9910311  
## 6230 0.0002853232 0.0020238956 0.9826594  
## 6456 0.0003126962 0.0060972176 0.9515925  
## 12472 0.0009997544 0.0046815067 0.9895107  
## 6527 0.0008480284 0.0009460835 0.9984935  
## 3193 0.0003264335 0.0037712047 0.9714309  
## 4740.1 0.0004142166 0.0004257958 0.9982843  
## 7210.3 0.0008245121 0.0033131929 0.9910311  
## 12255 0.0004540485 0.0012032442 0.9941611  
## 7458 0.0002744827 0.0034916196 0.9684444  
## 7458.1 0.0002744827 0.0034916196 0.9684444  
## 8946 0.0250174280 0.0294725485 1.0231563  
## 2243 0.0018075001 0.0014965509 1.0001956  
## 7457 0.0003309048 0.0041789114 0.9687318  
## 2686 0.0006211791 0.0025912398 0.9904363  
## 2708 0.0007099091 0.0018215348 0.9946330  
## 7456 0.0003569036 0.0045781132 0.9682595  
## 7871 0.0023533680 0.0043737008 0.9980965  
## 10318 0.0009989907 0.0017662242 0.9969691  
## 7210.4 0.0008245121 0.0033131929 0.9910311  
## 6739 0.0005549431 0.0006769960 0.9979318  
## 9722 0.0004142166 0.0007393603 0.9963397  
## 3479 0.0005460855 0.0064918108 0.9708080  
## 3193.1 0.0003264335 0.0037712047 0.9714309  
## 7210.5 0.0008245121 0.0033131929 0.9910311  
## 4285 0.0005157424 0.0004998261 0.9985372  
## 10125 0.0025631509 0.0028332317 1.0002417  
## 3174 0.0003264335 0.0037712047 0.9714309  
## 2742 0.0003750973 0.0005055129 0.9974234  
## 7650.1 0.0005575413 0.0004619717 0.9989405  
## 4750.1 0.0031762435 0.0028619454 1.0013840  
## 7650.2 0.0005575413 0.0004619717 0.9989405  
## 5491.1 0.0002695332 0.0002931919 0.9979856  
## 7459 0.0003757137 0.0048663461 0.9679648  
## 6739.1 0.0005549431 0.0006769960 0.9979318  
## 6443 0.0006699317 0.0128721728 0.9526546  
## 4571 0.0011598703 0.0009549665 0.9995576  
## 6527.1 0.0008480284 0.0009460835 0.9984935  
## 6234 0.0005746350 0.0041162149 0.9827718  
## 6451.2 0.0003419521 0.0067802188 0.9508077  
## 3170.1 0.0003744344 0.0043839059 0.9710883  
## 7457.1 0.0003309048 0.0041789114 0.9687318  
## 8541 0.0006439560 0.0006011150 0.9987573  
## 4750.2 0.0031762435 0.0028619454 1.0013840  
## 3188 0.0005460855 0.0066127306 0.9702515  
## 3918 0.0020388084 0.0023455178 0.9995989  
## 4056 0.0019915599 0.0101338947 0.9894710  
## 3476 0.0002953809 0.0032934511 0.9724138  
## 7507.1 0.0009497935 0.0012879288 0.9979778  
## 6447 0.0005594540 0.0101105325 0.9553748  
## 7650.3 0.0005575413 0.0004619717 0.9989405  
## 7650.4 0.0005575413 0.0004619717 0.9989405  
## 3480 0.0005460855 0.0064918108 0.9708080  
## 7446 0.0003289659 0.0041490066 0.9687712  
## 6233 0.0002919319 0.0021279091 0.9821692  
## 7210.6 0.0008245121 0.0033131929 0.9910311  
## 3170.2 0.0003744344 0.0043839059 0.9710883  
## 3476.1 0.0002953809 0.0032934511 0.9724138  
## 4740.2 0.0004142166 0.0004257958 0.9982843  
## 4740.3 0.0004142166 0.0004257958 0.9982843  
## 3918.1 0.0020388084 0.0023455178 0.9995989  
## 8698 0.0003151700 0.0004018864 0.9975498  
## 5496 0.0002695332 0.0002931919 0.9979856  
## 6447.1 0.0005594540 0.0101105325 0.9553748  
## 4056.1 0.0019915599 0.0101338947 0.9894710  
## 7210.7 0.0008245121 0.0033131929 0.9910311  
## 7507.2 0.0009497935 0.0012879288 0.9979778  
## 3199 0.0005460855 0.0066127306 0.9702515  
## 6434 0.0005976025 0.0127610581 0.9472966  
## 2686.1 0.0006211791 0.0025912398 0.9904363  
## 3182 0.0005460855 0.0066127306 0.9702515  
## 7507.3 0.0009497935 0.0012879288 0.9979778  
## 4435.1 0.0005274343 0.0005421220 0.9983979  
## 5496.1 0.0002695332 0.0002931919 0.9979856  
## 2710 0.0007983429 0.0018903783 0.9952291  
## 3465 0.0003744344 0.0043024004 0.9716358  
## 6435 0.0005492916 0.0112405840 0.9494440  
## 6451.3 0.0003419521 0.0067802188 0.9508077  
## 4834.2 0.0018748237 0.0034212961 0.9977020  
## 3464 0.0004073230 0.0047782602 0.9710639  
## 4750.3 0.0031762435 0.0028619454 1.0013840  
## 4750.4 0.0031762435 0.0028619454 1.0013840  
## 5497 0.0006397662 0.0007223676 0.9982501  
## 7507.4 0.0009497935 0.0012879288 0.9979778  
## 7462 0.0003717941 0.0048062666 0.9680237  
## 7210.8 0.0008245121 0.0033131929 0.9910311  
## 4056.2 0.0019915599 0.0101338947 0.9894710  
## 3172 0.0003264335 0.0037712047 0.9714309  
## 4740.4 0.0004142166 0.0004257958 0.9982843  
## 10125.1 0.0025631509 0.0028332317 1.0002417  
## 7210.9 0.0008245121 0.0033131929 0.9910311  
## 3497 0.0005460855 0.0064918108 0.9708080  
## 4750.5 0.0031762435 0.0028619454 1.0013840  
## 6456.1 0.0003126962 0.0060972176 0.9515925  
## 3168 0.0005460855 0.0066127306 0.9702515  
## 1504 0.0033587542 0.0071028568 0.9984507  
## 7210.10 0.0008245121 0.0033131929 0.9910311  
## 6440 0.0002919873 0.0060620113 0.9484524  
## 4740.5 0.0004142166 0.0004257958 0.9982843  
## 1305 0.0017172374 0.0017630590 0.9995940  
## 7447 0.0003569036 0.0045781132 0.9682595  
## 3497.1 0.0005460855 0.0064918108 0.9708080  
## 11558 0.0009398288 0.0056838341 0.9859725  
## 7211 0.0005112589 0.0024726628 0.9886290  
## 4696 0.0011386801 0.0024673012 0.9960859  
## 7871.1 0.0023533680 0.0043737008 0.9980965  
## 10707 0.0001703323 0.0001875476 0.9978522  
## 7210.11 0.0008245121 0.0033131929 0.9910311  
## 6739.2 0.0005549431 0.0006769960 0.9979318  
## 4671 0.0033671688 0.0035341571 1.0011943  
## 3175 0.0004073230 0.0048678465 0.9705109  
## 5549 0.0006929716 0.0005931409 0.9990058  
## 8911 0.0250174280 0.0294725485 1.0231563  
## 4740.6 0.0004142166 0.0004257958 0.9982843  
## 6448 0.0003144393 0.0062034447 0.9510260  
## 6055 0.0002921534 0.0002388270 0.9987033  
## 3424 0.0004950986 0.0007268516 0.9972343  
## 6452 0.0002956756 0.0055723336 0.9531917  
## 4740.7 0.0004142166 0.0004257958 0.9982843  
## 2742.1 0.0003750973 0.0005055129 0.9974234  
## 6237 0.0005922830 0.0042485105 0.9827643  
## 9453 0.0006373956 0.0005752472 0.9988304  
## 7459.1 0.0003757137 0.0048663461 0.9679648  
## 6239 0.0003311129 0.0022543369 0.9834290  
## 7210.12 0.0008245121 0.0033131929 0.9910311  
## 7210.13 0.0008245121 0.0033131929 0.9910311  
## 7210.14 0.0008245121 0.0033131929 0.9910311  
## 5496.2 0.0002695332 0.0002931919 0.9979856  
## 5497.1 0.0006397662 0.0007223676 0.9982501  
## 6931 0.0003427994 0.0003220679 0.9984401  
## 8458 0.0009937996 0.0028267819 0.9942039  
## 6440.1 0.0002919873 0.0060620113 0.9484524  
## 4740.8 0.0004142166 0.0004257958 0.9982843  
## 5496.3 0.0002695332 0.0002931919 0.9979856  
## 5935 0.0019132099 0.0034927277 0.9977387  
## 7453 0.0003176648 0.0040231856 0.9686278  
## 6438 0.0002716811 0.0061663046 0.9436648  
## 4740.9 0.0004142166 0.0004257958 0.9982843  
## 7455 0.0003388163 0.0043471699 0.9682335  
## 6739.3 0.0005549431 0.0006769960 0.9979318  
## 6945 0.0004134549 0.0012847611 0.9929491  
## 6940 0.0003006235 0.0010893270 0.9915156  
## 2689 0.0005797006 0.0025063953 0.9900062  
## 3476.2 0.0002953809 0.0032934511 0.9724138  
## 4056.3 0.0019915599 0.0101338947 0.9894710  
## 10707.1 0.0001703323 0.0001875476 0.9978522  
## 10418 0.0011273335 0.0009918170 0.9993798  
## 11289 0.0002664642 0.0010683726 0.9904950  
## 4750.6 0.0031762435 0.0028619454 1.0013840  
## 4695.1 0.0004797951 0.0026369874 0.9869160  
## 6527.2 0.0008480284 0.0009460835 0.9984935  
## 6527.3 0.0008480284 0.0009460835 0.9984935  
## 3169.1 0.0005460855 0.0066127306 0.9702515  
## 3918.2 0.0020388084 0.0023455178 0.9995989  
## 4696.1 0.0011386801 0.0024673012 0.9960859  
## 7650.5 0.0005575413 0.0004619717 0.9989405  
## 7460 0.0002743605 0.0034878019 0.9684650  
## 7210.15 0.0008245121 0.0033131929 0.9910311  
## 3497.2 0.0005460855 0.0064918108 0.9708080  
## 9528 0.0030258731 0.0106931683 0.9944805  
## 11165 0.0004372946 0.0006833687 0.9969334  
## 3480.1 0.0005460855 0.0064918108 0.9708080  
## 7650.6 0.0005575413 0.0004619717 0.9989405  
## 10787 0.0016140370 0.0045058090 0.9949613  
## 7447.1 0.0003569036 0.0045781132 0.9682595  
## 7210.16 0.0008245121 0.0033131929 0.9910311  
## 11165.1 0.0004372946 0.0006833687 0.9969334  
## 6796 0.0004320913 0.0003480525 0.9988742  
## 4740.10 0.0004142166 0.0004257958 0.9982843  
## 7210.17 0.0008245121 0.0033131929 0.9910311  
## 4934 0.0011667382 0.0016676916 0.9980069  
## 6438.1 0.0002716811 0.0061663046 0.9436648  
## 6447.2 0.0005594540 0.0101105325 0.9553748  
## 7210.18 0.0008245121 0.0033131929 0.9910311  
## 4056.4 0.0019915599 0.0101338947 0.9894710  
## 6237.1 0.0005922830 0.0042485105 0.9827643  
## 1305.1 0.0017172374 0.0017630590 0.9995940  
## 7210.19 0.0008245121 0.0033131929 0.9910311  
## 4648 0.0022767962 0.0374682120 0.9611129  
## 6232 0.0003514536 0.0023181324 0.9839893  
## 5496.4 0.0002695332 0.0002931919 0.9979856  
## 3481 0.0005460855 0.0064918108 0.9708080  
## 7650.7 0.0005575413 0.0004619717 0.9989405  
## 7210.20 0.0008245121 0.0033131929 0.9910311  
## 7210.21 0.0008245121 0.0033131929 0.9910311  
## 7507.5 0.0009497935 0.0012879288 0.9979778  
## 6943 0.0006282374 0.0017843303 0.9938483  
## 8946.1 0.0250174280 0.0294725485 1.0231563  
## 7210.22 0.0008245121 0.0033131929 0.9910311  
## 6230.1 0.0002853232 0.0020238956 0.9826594  
## 5497.2 0.0006397662 0.0007223676 0.9982501  
## 6055.1 0.0002921534 0.0002388270 0.9987033  
## 5494 0.0002673891 0.0002844487 0.9980451  
## 3188.1 0.0005460855 0.0066127306 0.9702515  
## 3172.1 0.0003264335 0.0037712047 0.9714309  
## 10958 0.0004852004 0.0005907249 0.9978682  
## 11899 0.0009488745 0.0046416588 0.9889263  
## 6436 0.0002955270 0.0065082846 0.9453470  
## 8710 0.0008057584 0.0038193419 0.9891698  
## 3479.1 0.0005460855 0.0064918108 0.9708080  
## 10125.2 0.0025631509 0.0028332317 1.0002417  
## 6435.1 0.0005492916 0.0112405840 0.9494440  
## 7650.8 0.0005575413 0.0004619717 0.9989405  
## 1142 0.0016543639 0.0026747370 0.9980148  
## 5497.3 0.0006397662 0.0007223676 0.9982501  
## 6451.4 0.0003419521 0.0067802188 0.9508077  
## 6447.3 0.0005594540 0.0101105325 0.9553748  
## 3486 0.0003593284 0.0041355647 0.9715735  
## 6931.1 0.0003427994 0.0003220679 0.9984401  
## 7210.23 0.0008245121 0.0033131929 0.9910311  
## 6440.2 0.0002919873 0.0060620113 0.9484524  
## 7871.2 0.0023533680 0.0043737008 0.9980965  
## 4056.5 0.0019915599 0.0101338947 0.9894710  
## 2689.1 0.0005797006 0.0025063953 0.9900062  
## 3193.2 0.0003264335 0.0037712047 0.9714309  
## 7210.24 0.0008245121 0.0033131929 0.9910311  
## 6230.2 0.0002853232 0.0020238956 0.9826594  
## 6451.5 0.0003419521 0.0067802188 0.9508077  
## 10125.3 0.0025631509 0.0028332317 1.0002417  
## 4571.1 0.0011598703 0.0009549665 0.9995576  
## 5935.1 0.0019132099 0.0034927277 0.9977387  
## 7210.25 0.0008245121 0.0033131929 0.9910311  
## 6444 0.0005939012 0.0110276572 0.9541789  
## 9722.1 0.0004142166 0.0007393603 0.9963397  
## 7451 0.0002744827 0.0034916196 0.9684444  
## 3465.1 0.0003744344 0.0043024004 0.9716358  
## 3464.1 0.0004073230 0.0047782602 0.9710639  
## 7455.1 0.0003388163 0.0043471699 0.9682335  
## 8458.1 0.0009937996 0.0028267819 0.9942039  
## 4571.2 0.0011598703 0.0009549665 0.9995576  
## 3424.1 0.0004950986 0.0007268516 0.9972343  
## 7210.26 0.0008245121 0.0033131929 0.9910311  
## 7507.6 0.0009497935 0.0012879288 0.9979778  
## 7507.7 0.0009497935 0.0012879288 0.9979778  
## 7211.1 0.0005112589 0.0024726628 0.9886290  
## 7446.1 0.0003289659 0.0041490066 0.9687712  
## 9293 0.0013173417 0.0014548737 0.9989929  
## 3480.2 0.0005460855 0.0064918108 0.9708080  
## 8458.2 0.0009937996 0.0028267819 0.9942039  
## 7210.27 0.0008245121 0.0033131929 0.9910311

### This is not easy to observe. Taking a look at sample again by filtering residuals that are over a +/- 2/3 threshold

#Percentage of sample with residuals over (+/-)2  
nrow(housing\_data\_df)

## [1] 12865

nrow(large\_res)

## [1] 261

322/12865\*100

## [1] 2.502915

#calculate average leverage for comparison with 4 parameters  
avg\_leverage = (4+1)/12865  
avg\_leverage

## [1] 0.0003886514

#calculate limit(s) leverage should not exceed  
leverage\_limit= avg\_leverage\*2  
leverage\_limit

## [1] 0.0007773028

leverage\_limit3 = avg\_leverage\*3  
leverage\_limit3

## [1] 0.001165954

#get count of samples over leverage limit  
large\_res%>%  
 filter(leverage > leverage\_limit)%>%  
 nrow()

## [1] 99

large\_res%>%  
 filter(leverage > leverage\_limit3)%>%  
 nrow()

## [1] 43

this produced a result of 99 cases outside of a threshold that go past a boundary for average leverage. Once tripled the results were 43.

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

dwt(housing\_data\_df\_lm1)

## lag Autocorrelation D-W Statistic p-value  
## 1 0.6309692 0.7380424 0  
## Alternative hypothesis: rho != 0

dwt(housing\_data\_df\_lm2)

## lag Autocorrelation D-W Statistic p-value  
## 1 0.003213724 1.993554 0.732  
## Alternative hypothesis: rho != 0

The D-W Statistic for both housing\_data\_df\_lm1 and housing\_data\_df\_lm1 contain values that are less than 1 and remotely close to a value of 2. Since the values aren’t close to 2 we can safely assume taht eh independence of our two models are not met.

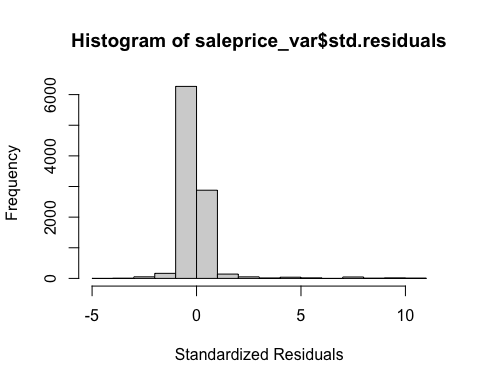
# Perform the necessary calculations to assess the assumption of no multicollinearity and state if the condition is met or not.

vif(housing\_data\_df\_lm2)

## zip5 bedrooms year\_built   
## 1.032914 1.493593 1.131387   
## square\_feet\_total\_living   
## 1.638171

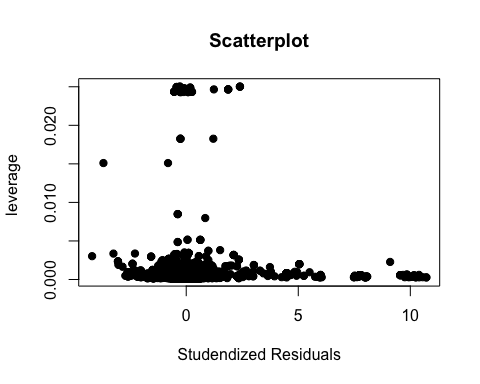
# Visually check the assumptions related to the residuals using the plot() and hist() functions. Summarize what each graph is informing you of and if any anomalies are present.

## Histogram Chart



I am not observing any annomolies from this historgram chart

## Plot



Again I see no annomolies

# 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 regression model appears to be unbiased.