Real Estate Regression

Frank Neugebauer

January 27, 2019

# Introduction

Uses statistical correlation, multiple regression and R programming Sale Price and several other possible predictors are analyzed and used.

# Import Data and Extract Sale Price and Sq Ft Lot

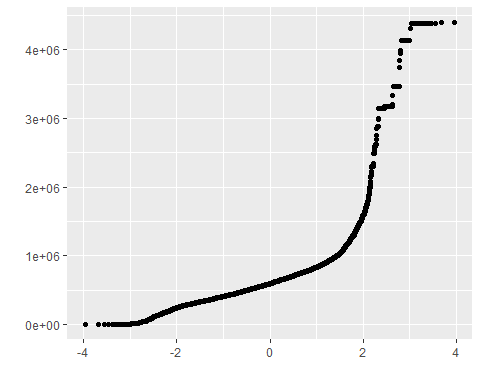
housing\_data <- read\_excel("housing.xlsx")  
housing\_subset <- housing\_data[c('Sale Price', 'sq\_ft\_lot', 'square\_feet\_total\_living', 'bedrooms', 'bath\_full\_count', 'year\_built')]  
count\_all <- count(housing\_subset)  
  
# rename column  
setnames(housing\_subset, old = c('Sale Price', 'sq\_ft\_lot', 'square\_feet\_total\_living', 'bedrooms', 'bath\_full\_count', 'year\_built'),   
 new = c('sale\_price', 'sq\_ft\_lot', 'square\_feet\_total\_living', 'bedrooms','bath\_full\_count', 'year\_built'))

# How the data is cleansed.

Shifted from the raw data to log(size) and log(price). This created a more normalized distribution. The core transformation and resulting plot follow. Omitted the analysis that shows that neither price nor size follow a normal distribution, except for the qplot of each.

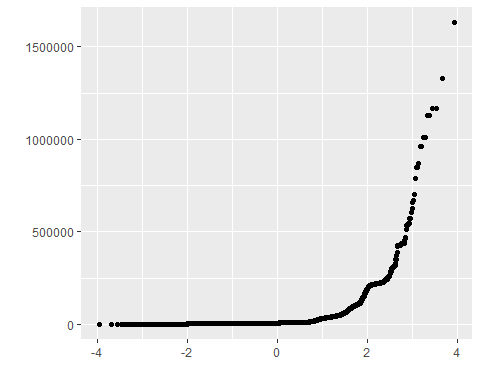
qplot(sample = housing\_subset$sale\_price, stat='qq')

## Warning: `stat` is deprecated



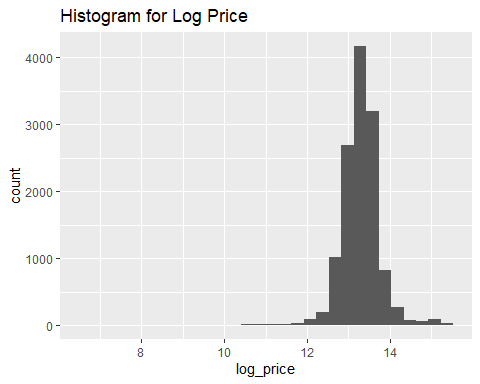
qplot(sample = housing\_subset$sq\_ft\_lot, stat='qq')

## Warning: `stat` is deprecated



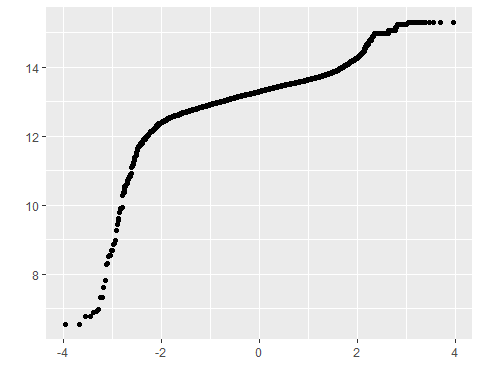
# Convert to log for price then size.  
housing\_subset$log\_price <- log(housing\_subset$sale\_price)  
ggplot(housing\_subset, aes(x = log\_price)) +  
 geom\_histogram() +  
 labs(title = 'Histogram for Log Price')

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



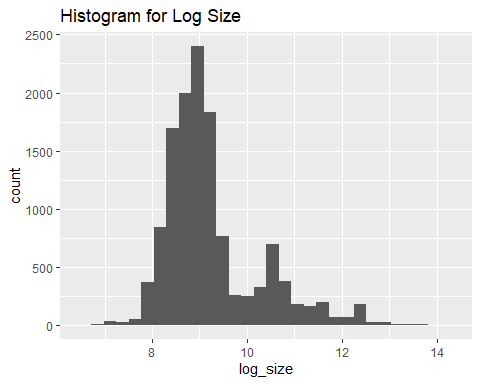
qplot(sample = housing\_subset$log\_price, stat='qq')

## Warning: `stat` is deprecated



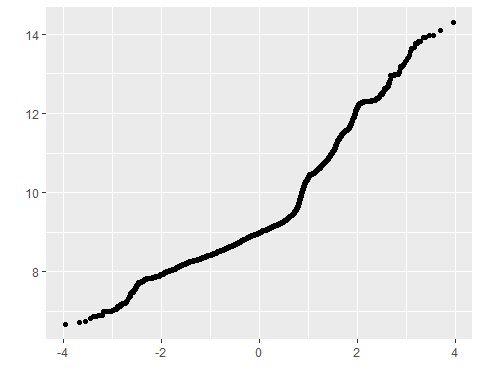
housing\_subset$log\_size <- log(housing\_subset$sq\_ft\_lot)  
ggplot(housing\_subset, aes(x = log\_size)) +  
 geom\_histogram() +  
 labs(title = 'Histogram for Log Size')

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



qplot(sample = housing\_subset$log\_size, stat='qq')

## Warning: `stat` is deprecated



# Do the data science thing

First, setup the linear regression between price and size, and then setup all the ratios.

price\_size\_lm <- lm(log\_price ~ log\_size, data = housing\_subset)  
housing\_subset$resid <- resid(price\_size\_lm)  
housing\_subset$stand\_resid <- rstandard(price\_size\_lm)  
housing\_subset$stud\_resid <- rstudent(price\_size\_lm)  
housing\_subset$cooks\_dist <- cooks.distance(price\_size\_lm)  
housing\_subset$leverage <- hatvalues(price\_size\_lm)  
housing\_subset$covariance\_ratios <- covratio(price\_size\_lm)

Look for large residuals > |2|. Store these because we won’t be removing anything that’s not in this set.

housing\_subset$large\_residual <- housing\_subset$stand\_resid > 2 | housing\_subset$stand\_resid < -2  
large\_resid\_sum <- sum(housing\_subset$large\_residual)  
housing\_subset\_large\_resid <- housing\_subset[housing\_subset$large\_residual, c('sale\_price', 'log\_price',  
 'sq\_ft\_lot', 'log\_price', 'resid', 'stand\_resid', 'cooks\_dist', 'leverage', 'covariance\_ratios')]

There are 468 observations with a standard residual > |2|, which represents 0.0363778 of the entire population as expected.

Next, very large (> |2.5|) deviations from the standard residual are considered. 99% of the cases should fall within |2.5| with 1% falling outside those limits.

housing\_subset$far\_outside\_resid <- housing\_subset$stand\_resid > 2.5 | housing\_subset$stand\_resid < -2.5  
sum\_far\_outside <- sum(housing\_subset$far\_outside\_resid)

In this case, 335 fall far outside, which is 0.0260396, which is not expected, since it’s more than double the expected 1%. More investigation is required.

The next metric to be considered is Cook’s Distance. This measures influence when the value is above 1.

housing\_subset\_large\_resid$large\_cooks <- housing\_subset\_large\_resid$cooks\_dist >= 1  
large\_cooks\_sum <- sum(housing\_subset\_large\_resid$large\_cooks)

There are 0 observations that meet the criteria for influence, but as will be shown shortly, there are two cases that come fairly close.

Next, leverage is considered, which take 2x and 3x of the average leverage. The calculation for “average” leverage (which is not the mean for all leverage values) is .02 (k + 1)/n = (1 + 1)/12865 or 2/12865 or simply `0.000155460552’.

avg\_leverage <- 0.000155460552  
two\_x <- avg\_leverage \* 2  
three\_x <- avg\_leverage \* 3  
two\_x\_leverage\_sum <- sum(housing\_subset\_large\_resid$leverage >= two\_x &   
 housing\_subset\_large\_resid$leverage < three\_x)  
three\_x\_leverage\_sum <- sum(housing\_subset\_large\_resid$leverage >= three\_x)  
  
# I need this later on.  
housing\_subset$two\_x\_leverage <- housing\_subset$leverage >= two\_x &   
 housing\_subset$leverage < three\_x

The average leverage for the sample is 1.554605510^{-4}. There are 20 cases at two times this, and 85 cases at three times.

Covariance boundaries are defined as 1 +/- [3(k + 1)/n] = 1 +/- [3(1 + 1)/12865 = 1 +/- 6/12865 = 1 +/- 0.000466381656 or 1.000466381656 and .999533618344.

bound\_low <- .999533618344  
bound\_high <- 1.000466381656  
covariance\_deviation\_sum <- sum(housing\_subset\_large\_resid$covariance\_ratios   
 <= bound\_high & housing\_subset\_large\_resid$covariance\_ratios >= bound\_low)

There are 105 outlier cases.

Some cases clearly require removal. The |2| standard residual outliers are accounted for as the superset of possible observations for removal. Cook’s did not further reduce this number, but leverage did significantly. Covariance reduced the number from the standard residual list, but only by the same number as the sum of both 2x and 3x leverage.

**As a very conservative data remover, I chose the 2x leverage observations and removed them.**

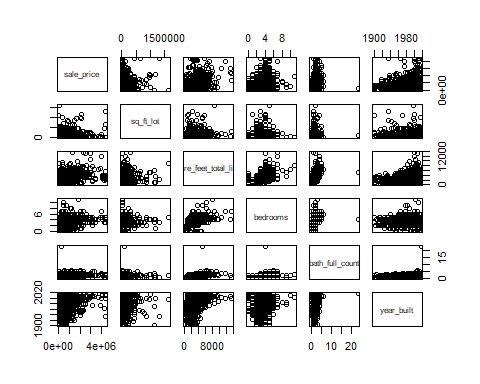
final\_data <- housing\_subset[(housing\_subset$two\_x\_leverage == FALSE) |   
 (housing\_subset$large\_residual == FALSE),   
 c('sale\_price', 'sq\_ft\_lot',   
 'square\_feet\_total\_living',  
 'bedrooms','bath\_full\_count',   
 'year\_built')]  
final\_data\_count <- count(final\_data)

Created two variables: one that contains the variables Sale Price and Square Foot of Lot and one that contains Sale Price and several additional predictors. The basis for the additional predictor selections is explained.

lm\_price\_size <- lm(sale\_price ~ sq\_ft\_lot, data = final\_data)  
lm\_price\_size\_br\_living <- lm(sale\_price ~ sq\_ft\_lot + bedrooms + square\_feet\_total\_living, data = final\_data)

Chose bedrooms and square\_feet\_total\_living as additional predictors becasue in looking at the pairs() graphic, it visually seemed to have the clearest correlation.

pairs(final\_data)



xecuted a summary() function on two variables defined in the previous step to compare the model results. Included the R2 and Adjusted R2 statistics? I then explain what these results say about the overall model.

summary(lm\_price\_size\_br\_living)

##   
## Call:  
## lm(formula = sale\_price ~ sq\_ft\_lot + bedrooms + square\_feet\_total\_living,   
## data = final\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1943892 -118473 -40376 43697 3779738   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.445e+05 1.300e+04 18.810 < 2e-16 \*\*\*  
## sq\_ft\_lot 7.275e-02 5.743e-02 1.267 0.205   
## bedrooms -2.373e+04 4.441e+03 -5.342 9.33e-08 \*\*\*  
## square\_feet\_total\_living 1.958e+02 4.051e+00 48.336 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 358300 on 12841 degrees of freedom  
## Multiple R-squared: 0.207, Adjusted R-squared: 0.2068   
## F-statistic: 1117 on 3 and 12841 DF, p-value: < 2.2e-16

The original and Adjusted are 0.01434 and 0.01426 respectively. With the addition of the two extra predictors, those values are 0.207 and 0.2068. This is a significant change, which may indicate that price is more affected by factors outside of lot size. This is not a surprise, given that generally, the correlation between price and lot size was pretty small. In other words, it statistically supports the prior model.

However, this change may also simply be because I added more predictors. To understand this, I used the AIC function and a higher AIC (between two lm models) means means the fit is worse, and a lower AIC means the fit is better.

AIC\_delta <- AIC(lm\_price\_size\_br\_living, k = 2) - AIC(lm\_price\_size, k = 2)

The difference between the original model and the new model is -2789.3068297, which indicates a better fit, and possibly that the additional factors are a better predictor of price.

I used the QuantPsyc library’s lm.beta to get standardized betas for each parameter.

lm.beta(lm\_price\_size\_br\_living)

## sq\_ft\_lot bedrooms square\_feet\_total\_living   
## 0.01029776 -0.05166265 0.48011660

These values mean that every 1 change in standard deviation, the lot size has a minimal impact on price, the number of bedrooms appears to reduce the price, and square footage (internal) has a significant impact. The bedrooms anomoly may be because it’s taken in the context of the external lot size.

I assessed the improvement of the new model compared to my original model (simple regression model) by testing whether this change is significant by performing an analysis of variance.

This is a job for anova, or One-way Analysis of Variance.

anova <- anova(lm\_price\_size, lm\_price\_size\_br\_living)

The output is a little challenging to interpret. While Pr(>F) is very small in the second model, there’s no comparable value to compare with the first.

I then performed casewise diagnostics to identify outliers and/or influential cases, storing each functions output in a dataframe assigned to a unique variable name.

I’ll get right into the r code, which really gets to the heart of g-k. Instead of creating new data frames, which is wasteful, I’m going to use the final\_data data frame and add new columns to it.

final\_data$resid <- resid(lm\_price\_size\_br\_living)  
final\_data$stand\_resid <- rstandard(lm\_price\_size\_br\_living)  
final\_data$stud\_resid <- rstudent(lm\_price\_size\_br\_living)  
final\_data$cooks\_dist <- cooks.distance(lm\_price\_size\_br\_living)  
final\_data$leverage <- hatvalues(lm\_price\_size\_br\_living)  
final\_data$covariance\_ratios <- covratio(lm\_price\_size\_br\_living)

Now that the data is all calcuated, it can be used. First, show some of the data.

final\_data[c('resid', 'stand\_resid', 'stud\_resid', 'cooks\_dist', 'leverage', 'covariance\_ratios')]

## # A tibble: 12,845 x 6  
## resid stand\_resid stud\_resid cooks\_dist leverage covariance\_ratios  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 -2313. -0.00646 -0.00646 0.00000000117 0.000112 1.00   
## 2 -63952. -0.179 -0.179 0.000000908 0.000114 1.00   
## 3 -120112. -0.335 -0.335 0.00000313 0.000111 1.00   
## 4 -71241. -0.199 -0.199 0.00000144 0.000145 1.00   
## 5 -85945. -0.240 -0.240 0.00000252 0.000175 1.00   
## 6 -780036. -2.18 -2.18 0.000406 0.000343 0.999  
## 7 141616. 0.395 0.395 0.0000171 0.000438 1.00   
## 8 -5247. -0.0146 -0.0146 0.0000000103 0.000193 1.00   
## 9 -307279. -0.858 -0.858 0.0000558 0.000303 1.00   
## 10 -46943. -0.131 -0.131 0.00000105 0.000245 1.00   
## # ... with 12,835 more rows

Here are the standardized residuals.

final\_data$large\_residual <- final\_data$stand\_resid > 2 | final\_data$stand\_resid < -2

And the sum of large residuals. The sum() function on the large\_residuals variable yields:

sum\_large\_resids <- sum(final\_data$large\_residual)  
final\_data[final\_data$large\_residual, c('sale\_price', 'sq\_ft\_lot', 'resid')]

## # A tibble: 317 x 3  
## sale\_price sq\_ft\_lot resid  
## <dbl> <dbl> <dbl>  
## 1 184667 7280 -780036.  
## 2 265000 112650 -856184.  
## 3 1390000 225640 999841.  
## 4 229000 236966 -784658.  
## 5 390000 63162 -876170.  
## 6 1588359 8752 732744.  
## 7 1450000 14043 1075692.  
## 8 163000 18498 -910215.  
## 9 270000 89734 -876930.  
## 10 200000 288367 -1294028.  
## # ... with 307 more rows

Calculated the leverage, cooks distance, and covariance rations.

cooks <- final\_data[final\_data$cooks\_dist, c('sale\_price', 'sq\_ft\_lot', 'resid')]

There is only 1 observation with a Cook’s Distance >= 1:

cooks

## # A tibble: 1 x 3  
## sale\_price sq\_ft\_lot resid  
## <dbl> <dbl> <dbl>  
## 1 698000 6635 -2313.

Leverage is first calculated and then 2x and 3x. Finally, the number of observations that are within 2x and 3x.

avg\_leverage <- 3 / count(final\_data)  
final\_data$two\_x\_leverage <- final\_data$leverage >= 0.0004671078 &   
 final\_data$leverage < 0.0007006617  
final\_data$three\_x\_leverage <- final\_data$leverage >= 0.0007006617  
  
two\_x\_lev <- sum(final\_data$two\_x\_leverage)  
three\_x\_lev <- sum(final\_data$three\_x\_leverage)

We expect to see 5% of the observations outside of 2x. The data is 0.0292721, which is good. Only 1% of observations should lie outside of 3x (1%). The data is 0.0496691. This is much higher than expected, indicating the potential for outliers.

Covariance boundaries are defined as 1 +/- [3(k + 1)/n] = 1 +/- [3(3 + 1)/12845 = 1 +/- 12/12845 = 1 +/- 0.000934215648 or 1.000934215648 and .999065784352.

bound\_low <- 1 - 12/12845  
bound\_high <- 1 + 12/12845  
covariance\_deviation\_sum <- sum(final\_data$covariance\_ratios <= bound\_high &   
 final\_data$covariance\_ratios >= bound\_low)

There are 12116 outlier cases. This is pretty extraordinary, and hopefully, given there’s only 1 outlier using Cook’s this is OK.

Calculations to assess the assumption of independence.

I (arbitrarily) chose the durbinWatsonTest() version of the Durbin Watson Test to check for independence.

dwt <- durbinWatsonTest(lm\_price\_size\_br\_living)  
dwt

## lag Autocorrelation D-W Statistic p-value  
## 1 0.7371108 0.5257765 0  
## Alternative hypothesis: rho != 0

The value of for the D-W Statistic is only 0.526, which is not very close to 2, implying that the test for independence has not been met. Similarly, the p-value is not bigger than 0.05.

Calculations to assess the assumption of no multicollinearity.

vif <- vif(lm\_price\_size\_br\_living)  
vif\_tol <- 1 / vif  
mean\_vif <- mean(vif)  
  
vif

## sq\_ft\_lot bedrooms square\_feet\_total\_living   
## 1.070255 1.514212 1.597586

vif\_tol

## sq\_ft\_lot bedrooms square\_feet\_total\_living   
## 0.9343571 0.6604095 0.6259442

mean\_vif

## [1] 1.394018

The multicollinearity guidelines: 1. There are no VIF values greater than 10, so no cause for concern there. 2. The mean is greater than 1, but not substantially, so it is unlikely the regression is biased. 3. No tolerances are below 0.1 or 0.2, which is good - no serious problems.

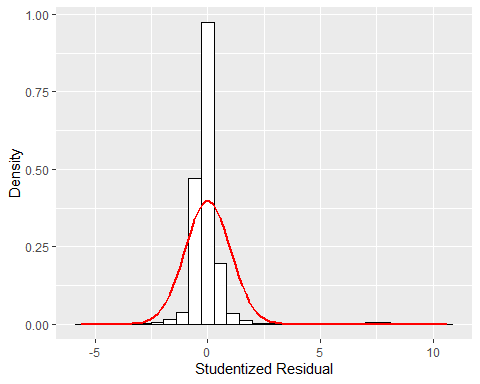
This all implies there is no colliniarity within the model.

Checked the assumptions related to the residuals using the plot() and hist() functions.

Here we go…histogram, ggplot, and scatter.

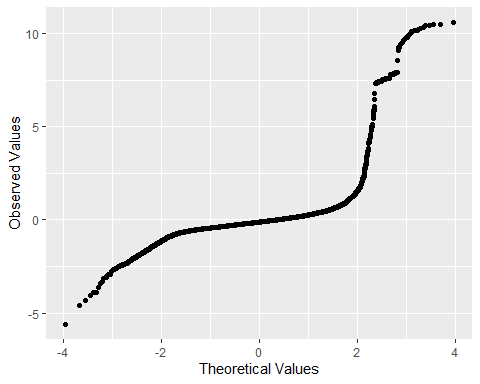
final\_data$fitted <- lm\_price\_size\_br\_living$fitted.values  
histogram <- ggplot(final\_data, aes(stud\_resid)) +   
 geom\_histogram(aes(y = ..density..), color = 'black', fill = 'white') +  
 labs(x = 'Studentized Residual', y = 'Density')  
  
histogram + stat\_function(fun = dnorm, args = list(mean = mean(final\_data$stud\_resid, na.rm = TRUE), sd = sd(final\_data$stud\_resid, na.rm = TRUE)), color = 'red', size = 1)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

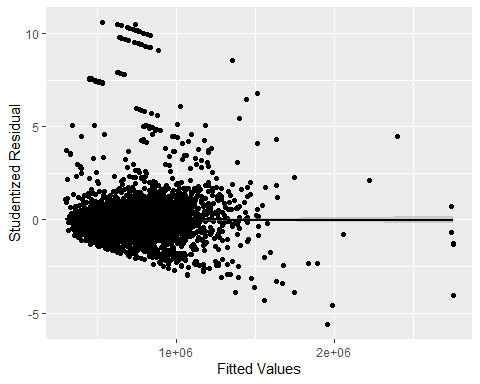


qplot(sample = final\_data$stud\_resid, stat = 'qq') + labs(x = 'Theoretical Values', y = 'Observed Values')

## Warning: `stat` is deprecated



scatter <- ggplot(final\_data, aes(fitted, stud\_resid))  
scatter + geom\_point() + geom\_smooth(method = 'lm', color = 'black') + labs(x = 'Fitted Values', y = 'Studentized Residual')



Is this regression model unbiased?

I believe the model is unbiased - but I’m not entirely sure becasue there are mixed results from the tests performed. If the model is unbiased, this implies that the sample is a valid representation of the population and statistically accurate for regression modeling.