Location, Location, Location: Linear Regression on King County House Sales

Capstone IDV for PH125.9x

KateBWilliams

4/30/2020

1 Introduction

One major aspect of the American Dream is house ownership. Purchasing a house represents one of the few big-ticket purchases that most people will make during their lifetimes. Therefore, there is considerable interest in real estate valuation- namely, what makes a house worth a certain price, and what features of a house predict its sales price. Financing a home through a bank or other financial institution often requires an assessment of the property by an appraiser, who through experience and calculations of similar properties determines an estimate, or prediction, of the property's value.

The House Sales in King County, USA (https://www.kaggle.com/harlfoxem/housesalesprediction), publically available through Kaggle, is a data set containing many of the property features that could be useful predictors for estimating a sale price.

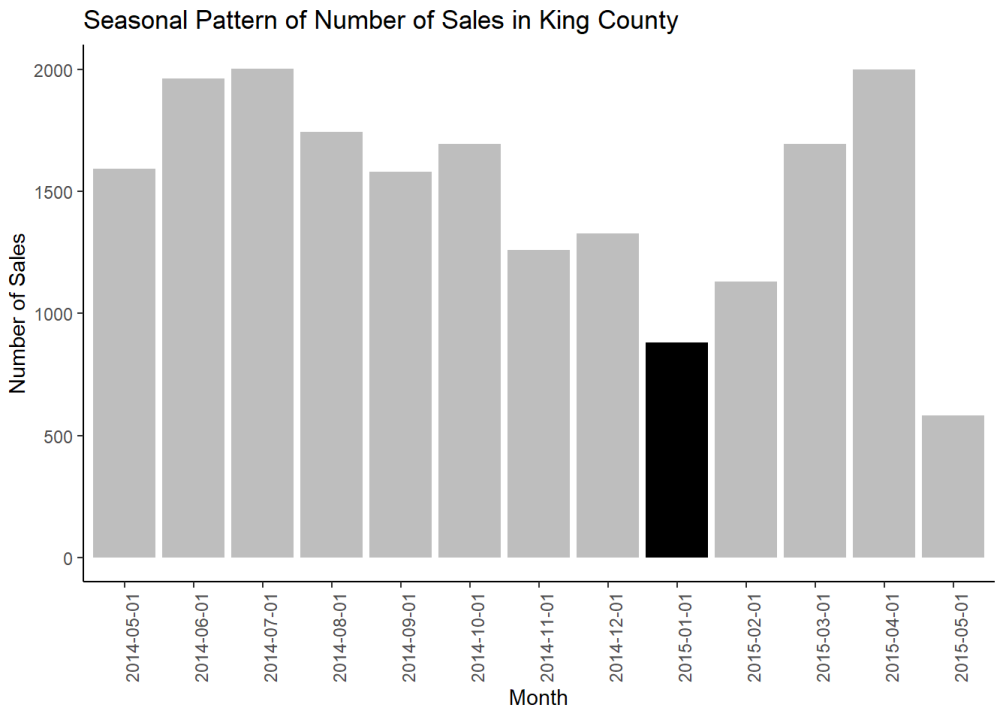
Here we can see an overview of the partitioned training data set `dl`, representing the other 90% of the randomly sampled observations:

```
glimpse(dl)
```

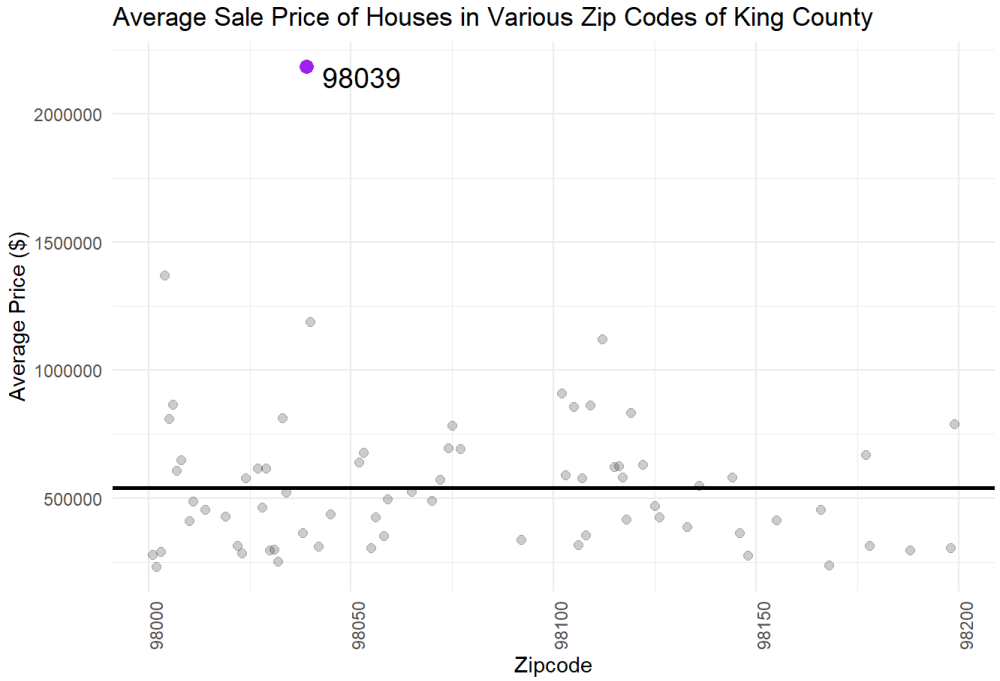
```
## Rows: 19,450
## Columns: 21
## $ id            <chr> "7129300520", "6414100192", "5631500400", "2487200875...
## $ date          <dttm> 2014-10-13, 2014-12-09, 2015-02-25, 2014-12-09, 2015...
## $ price         <dbl> 221900, 538000, 180000, 604000, 510000, 1225000, 2575...
## $ bedrooms      <dbl> 3, 3, 2, 4, 3, 4, 3, 3, 3, 3, 3, 2, 3, 3, 5, 4, 3, 4,...
## $ bathrooms     <dbl> 1.00, 2.25, 1.00, 3.00, 2.00, 4.50, 2.25, 1.50, 1.00,...
## $ sqft_living   <dbl> 1180, 2570, 770, 1960, 1680, 5420, 1715, 1060, 1780, ...
## $ sqft_lot      <dbl> 5650, 7242, 10000, 5000, 8080, 101930, 6819, 9711, 74...
## $ floors        <dbl> 1.0, 2.0, 1.0, 1.0, 1.0, 1.0, 2.0, 1.0, 1.0, 2.0, 1.0...
## $ waterfront    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ view          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0,...
## $ condition     <dbl> 3, 3, 3, 5, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 3, 3, 3, 4,...
## $ grade         <dbl> 7, 7, 6, 7, 8, 11, 7, 7, 7, 7, 8, 7, 7, 7, 7, 9, 7, 7...
## $ sqft_above    <dbl> 1180, 2170, 770, 1050, 1680, 3890, 1715, 1060, 1050, ...
## $ sqft_basement <dbl> 0, 400, 0, 910, 0, 1530, 0, 0, 730, 0, 1700, 300, 0, ...
## $ yr_built      <dbl> 1955, 1951, 1933, 1965, 1987, 2001, 1995, 1963, 1960,...
## $ yr_renovated  <dbl> 0, 1991, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...
## $ zipcode       <dbl> 98178, 98125, 98028, 98136, 98074, 98053, 98003, 9819...
## $ lat           <dbl> 47.5112, 47.7210, 47.7379, 47.5208, 47.6168, 47.6561,...
## $ long          <dbl> -122.257, -122.319, -122.233, -122.393, -122.045, -12...
## $ sqft_living15 <dbl> 1340, 1690, 2720, 1360, 1800, 4760, 2238, 1650, 1780,...
## $ sqft_lot15    <dbl> 5650, 7639, 8062, 5000, 7503, 101930, 6819, 9711, 811...
```

There are a variety of predictors, randing from square feet of living space, number of bedrooms, view, condition, and grade of a property (these last three are attempting to quantify more qualitative predictors). `sqft_living15` and `sqft_lot15` are not intuitively named: they represent the averages of the nearest 15 properties to a given house.

Since there is a `date` column, we can attempt to investigate what is happening in sales over the course of the year. Note how there is a dip in the number of sales per month coinciding with the winter months (January is lowlighted in black). This seasonal phenomenon is well known in real estate, and is attributed to buying behavior by individuals. For instance, families with children who may move tend to purchase houses and move during the summer vacation from school so as to be less disruptive. Homes often have greater "curb appeal" during the better weather of spring and summer.

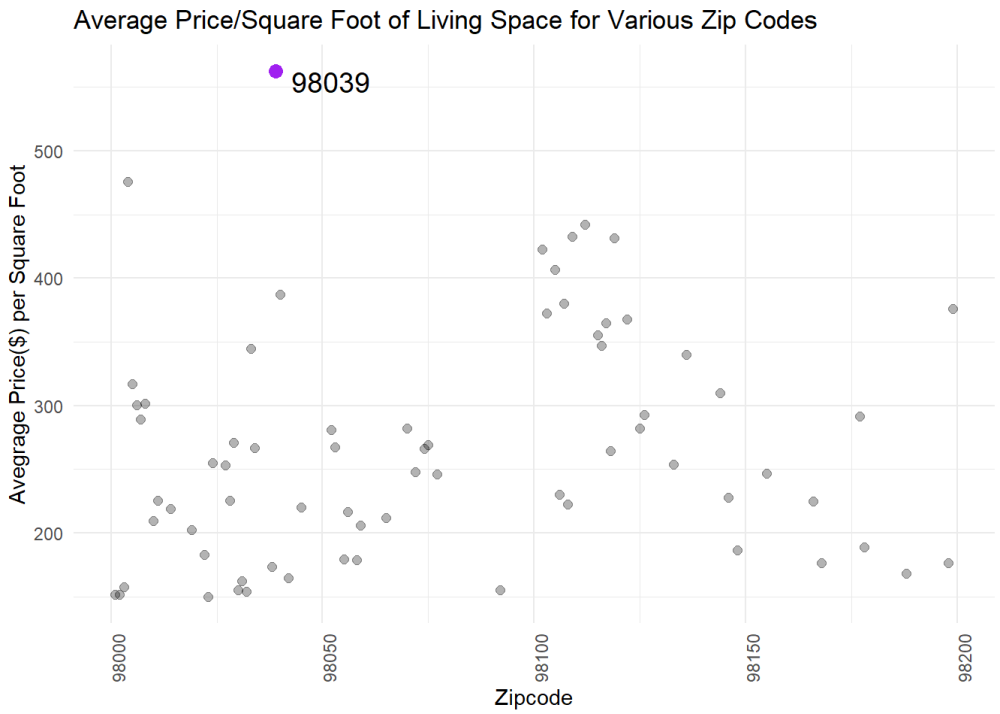


In addition, one of the most common rules of thumb in real estate is “location, location, location”. So, if you break down average sale price by zipcode, you can clearly see that there is considerable variation in average sale price due to this variable.



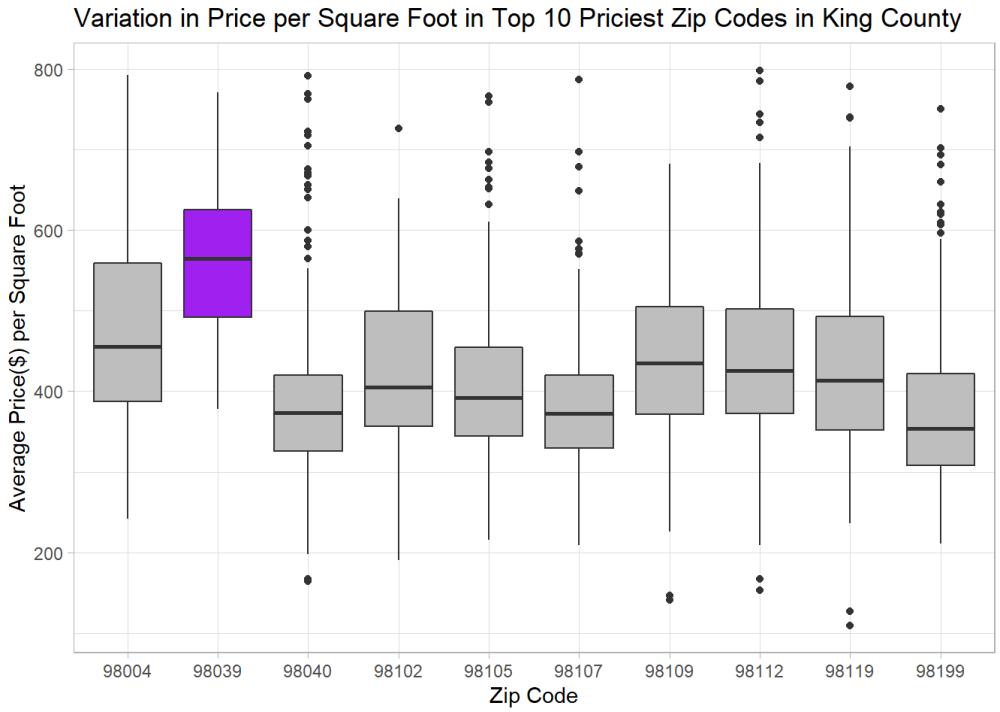
Horizontal line indicates county average sale price. Sale price in zipcode 98039 is highlighted in purple.

The zipcode represented in purple encompasses the area around Bellvue, just to the west of Seattle, and is the zipcode where Microsoft and its well-paid employees live. However, it is possible that the building codes and general house size in Bellvue is larger, resulting in higher sale prices. If we mutate the data in order to add in a column of, for example, price per square foot of living space, we might be able to normalize the sale prices some.



We can see here that normalizing for size (at least with living space), does account for some of the variation. However, the Bellvue zip code pricing is still quite high, suggesting other aspects driving the high prices for these properties.

Even within the relatively small areas encompassed by zip codes, there is considerable varibility on price per square foot. If we examine the top ten zipcodes with the highest average sale price, we can see the ranges in price represented in those areas.



Therefore, it is useful to examine other available factors that could be used as predictors for price.

2 Methods and Analysis

2.1 Partitioning the data into training and test sets

The most critical component for machine learning algorithms is to partition and set aside a test set of the data for later validation of the model. This validation set will not be used for any of the training algorithms. We can partition the data in the following way:

```
set.seed(1,sample.kind = "Rounding") # this is for R 3.6.3

test_index <- createDataPartition(dl$price, times = 1, p = 0.1, list = FALSE)
dl <- dl %>% slice(-test_index)
validation <- dl %>% slice(test_index)
```

We can partition the data set using a 90/10 split (represented by `p = 0.1`), due to the fact that there are 20,000+ observations. The 10% partition for the `validation` set still has nearly 2000 observations, which are plenty of points for effective testing of a final algorithm.

2.2 Linear regression

Linear regression analysis is a simple and intuitive way to begin modeling relationships within the data, especially for continous data, such as pricing (as opposed to categorical data, such as was a property sold, yes or no).

The most basic form of linear regression is of course, the intercept and slope for a straight line.

$$Y_p = \beta_0 + \beta_1 x$$

The goal of determining β_0 and β_1 is to minimize the distance between the regression line and the actual data points. There are a few different ways to quantify this fit. One is the R^2 value, which has a value between -1 to 1. An R^2 value equal to 1 would effectively be a perfect fits- all values would fall exactly on the regression line. A value of -1 is a similar idea, but with a negative correlation. The closer the values are to 0, the poorer the correlation and therefore the worse the predictive power of the regression.

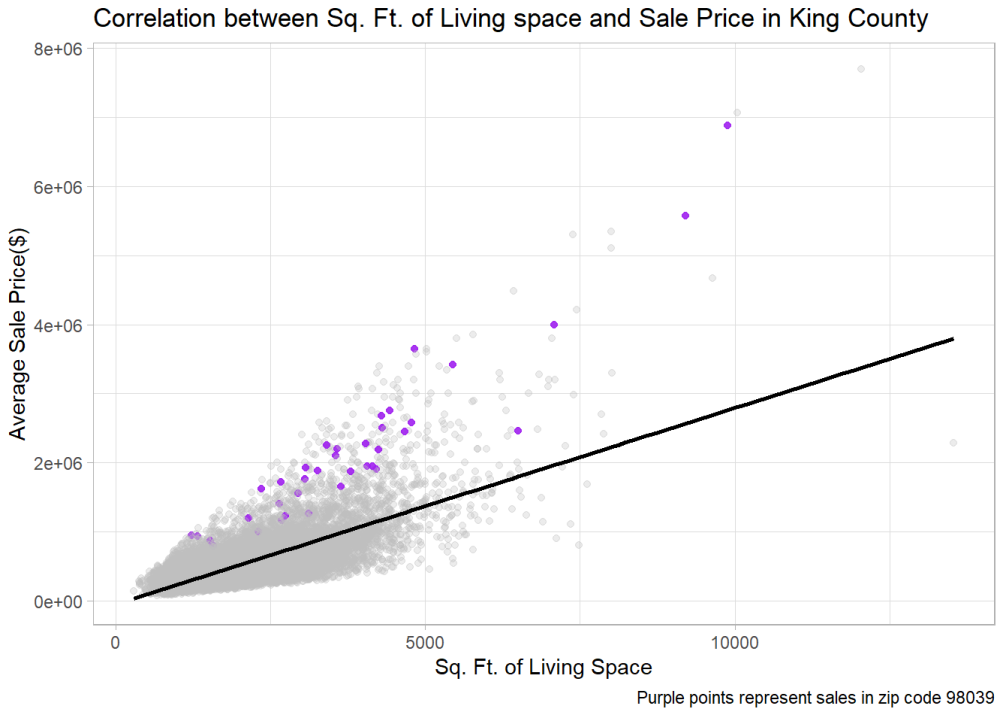
The Root Mean Squared Error (RMSE) is a related evaluation - an RMSE that equals 0 indicates the same thing as the R^2 having a value of 1 or -1, that the fit of the regression is perfect. The values of RMSE can vary quite a bit more, however.

Overall, a reduction in RMSE and an improvement in R^2 away from 0 are both heloful indicators for comparing regression algorithms and whether they represent improvements for prediction.

The simplest linear regression assumes a bivariate model, with one independent variable (let us assume `sqft_living`) acting on the dependent variable we are trying to predict: `price` .

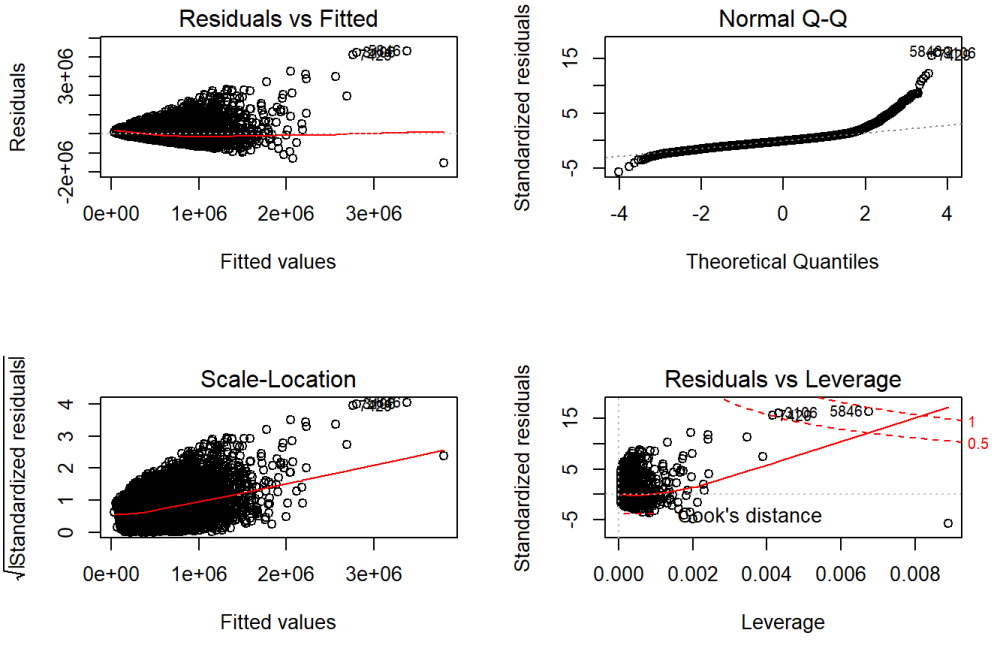
```
##               Estimate  Std. Error   t value     Pr(>|t|)
## (Intercept) -49597.9396 4953.544815 -10.01262 1.552266e-23
## sqft_living    284.1022    2.177505 130.47139 0.000000e+00
```

You can see this regression line and how it fits with the data. The R^2 value for this regression line is 0.4930741. There is clearly a positive correlation between `sqft_living` and sales `price` , but the R^2 value indicates that there is still a considerable amount of variation in price that is not accounted for by this simple regression.

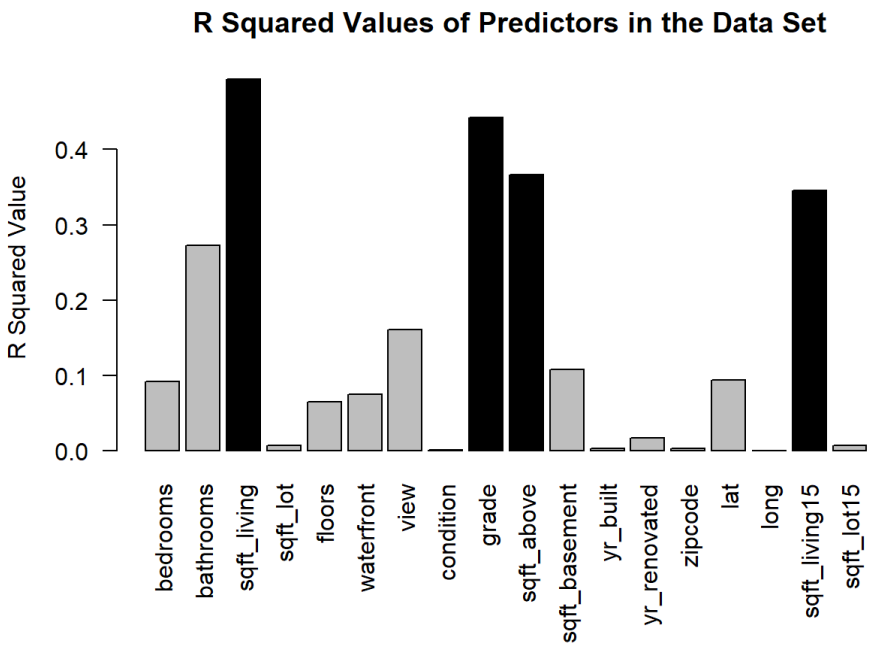


Note how the data appears to have the typical “fan shape” associated indicative of **heteroscedasticity**. Heteroscedasticity implies that there is some additional factor acting on the data in a predictable fashion.

If we examine the residuals plots from the regession algorithm, the plots still show distinctive predictive patterns (in other words, the residuals are not randomly scattered), suggesting that this bivariate model is not accounting for the variation in price, and is not the best predictor. Here, the residuals in the `Residuals vs. Fitted` plot also follow the typical *fan* pattern of heteroscedasticity.



In order to determine whether any of the other predictors are more effective in a bivariate model, we can examine the individual R^2 values for each predictor.



We can see that `sqft_living` is still the best individual predictor, followed by `grade`, `sqft_above`, `sqft_living15`.

2.3 Multivariate Linear Regression

The fact that at least three other predictors have R^2 values nearly as high (>0.3) as `sqft_living` predictor begs the question whether grouping these variable together will reduce residual error significantly.

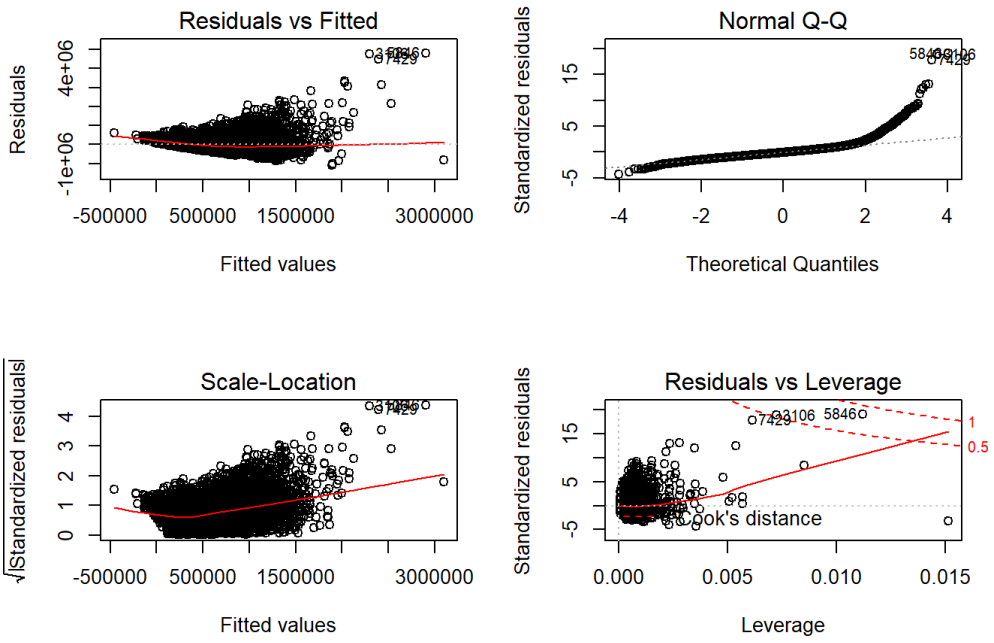
Multivariate linear regression is similar to the simple univariate model shown above, with additional variables added into the equation.

$$Y_p = \beta_0 + \beta_1 x_2 + \beta_2 x_2 + \beta_3 x_3 \ldots \beta_n x_n$$

Based upon the predictors above that show `sqft_living`, `grade`, `sqft_above`, `sqft_living15` all have the strongest individual R^2 values, we can incorporate these predictors into a multivariate algorithm.

```
fit_mv <- lm(price~ sqft_living+
               grade+
               sqft_above+
               sqft_living15, data = dl_train )
summary(fit_mv)$coefficients
```

```
##                   Estimate   Std. Error   t value      Pr(>|t|)
## (Intercept)   -652044.59315 15286.973938 -42.65361  0.000000e+00
## sqft_living       231.84945     4.706466  49.26190  0.000000e+00
## grade          105147.46952  2723.583949  38.60629 5.425898e-313
## sqft_above        -83.40575     5.048895 -16.51960  7.659040e-61
## sqft_living15      27.71278     4.558129   6.07986  1.227835e-09
```



We can see that there is some improvement in the R^2 value from the summary, and there is some improvement in the residuals plots. However, there is still heteroschedasticity, and so another undetermined factor is influencing the data.

2.4 Stepwise Regression Analysis

Would adding more predictors make the algorithm better? Intuitively, the answer would be yes, as the more information you have, the better your ability to predict. However, some predictors are not very useful, and their contribution to prediction is minimal.

WIth many predictors, it is possible to perform a stepwise analysis of the various predictors to determine their "quality" in their ability to contribute to algorithm performance (i.e. minimizing residual error).

In order to perform this analysis, we istall two additional libraries:

```
library(leaps)
library(MASS)
```

We can use a cross-validation model within this algorithm development method, which requires sampling, and we set the 10-fold cross-validation in `train_ctrl` . Additionally, we set the number of predictors to 18 through the `tuneGrid` . This is the number of independent variables we intend to assess for predictive quality. *Note that we have removed* `id` *and* `date` *as predictors. The assumption is that* `id` *is random, and that* `date` *does not have much impact on sales price.* There are a few options for the `method` parameter, but here we will use the `leapBackward` method, which starts with all predictors and iteratively trims them from the algorithm until it determines an optimal predictor set.

```
set.seed(123, sample.kind = "Rounding") # for repeatability, R 3.6.3

train_ctrl <- trainControl(method = "cv", number = 10) # set-up for 10-fold cross-validation

step_model <- train(price~., data = dl_numerics, method = "leapBackward", tuneGrid = data.frame(nvmax=1:18), trControl
= train_ctrl)
```

```
step_model$results
```

nvmax	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
1	262211.7	0.4927776	174093.7	16655.738	0.0171705	4604.691
2	252779.3	0.5288309	166275.3	17817.437	0.0110047	4120.050

nvmax	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
3	250638.5	0.5366464	164735.3	16912.473	0.0113864	4033.117
4	243028.7	0.5648083	158514.5	18089.750	0.0164466	4817.258
5	234419.5	0.5945706	155410.9	12110.865	0.0202631	3354.837
6	231148.3	0.6059241	152495.3	11867.074	0.0189908	3258.032
7	227331.3	0.6172286	148202.1	8133.108	0.0338571	6445.679
8	213578.0	0.6638463	134180.5	13070.665	0.0110908	3780.703
9	212814.7	0.6661657	133825.3	12795.023	0.0128781	3615.554
10	212776.5	0.6662809	133737.9	12809.282	0.0130046	3645.811
11	209872.5	0.6754184	131041.2	12799.751	0.0121167	3181.981
12	209781.2	0.6757126	131079.7	12762.418	0.0120491	3191.765
13	209686.8	0.6759926	131159.3	12749.509	0.0121186	3169.609
14	208949.8	0.6781956	131000.4	12452.182	0.0114187	3035.365
15	208913.9	0.6782706	130989.8	12418.440	0.0115273	3040.135
16	208805.0	0.6786240	130967.1	12359.412	0.0114713	2993.382
17	208740.5	0.6788489	131120.4	12277.994	0.0119136	2969.675
18	208740.5	0.6788489	131120.4	12277.994	0.0119136	2969.675

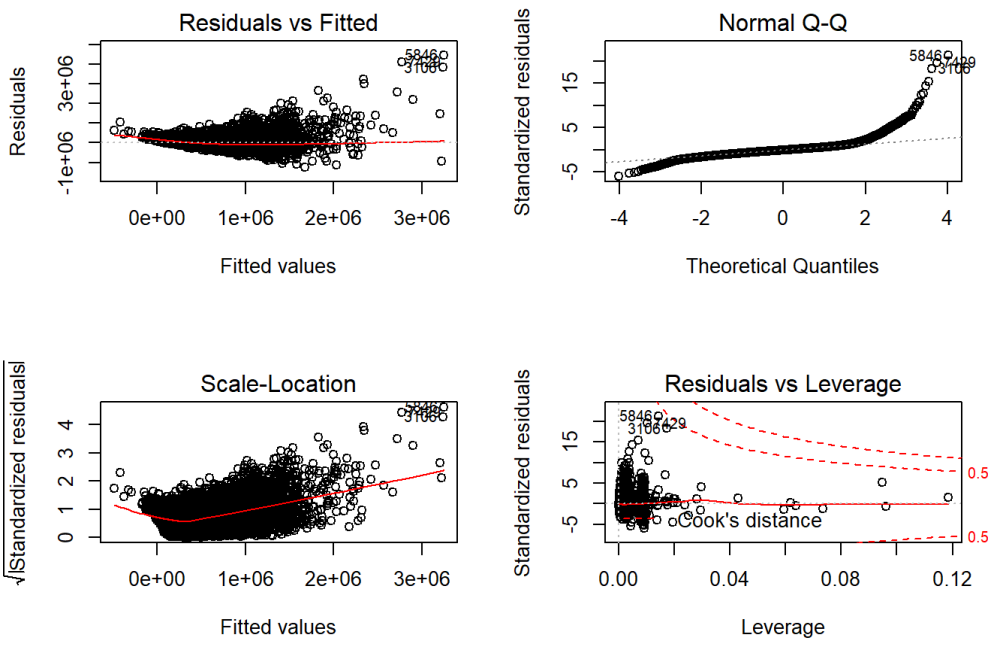
```
coef(step_model$finalModel, step_model$bestTune[,1])
```

```
##   (Intercept)      bedrooms     bathrooms   sqft_living      sqft_lot
## -2.962892e+07 -3.241614e+04  1.385079e+03  1.708465e+02  2.252902e-01
##        floors    waterfront          view     condition         grade
## -1.929954e+04  5.823870e+05  5.896572e+04  5.294752e+04  7.746647e+04
##    sqft_above  yr_renovated       zipcode           lat          long
##  3.090140e+01  6.211098e+01 -4.399854e+02  6.762437e+05 -3.275863e+05
## sqft_living15    sqft_lot15 sqft_basement
##  2.735194e+01 -4.469734e-01  0.000000e+00
```

The ouptut reveals that a majority of predictors were incorporated into the best algorithm, although the RMSE is still quite high.

Incorporating the recommended predictors from the stepwise backward regression, and adding them into a multivariate regression model gives the following outcome.

```
##                   Estimate   Std. Error    t value      Pr(>|t|)
## (Intercept)   -2.937464e+07 3.193732e+06  -9.197591  4.063413e-20
## bedrooms      -3.266987e+04 2.179338e+03 -14.990728  1.738311e-50
## bathrooms      1.723286e+03 3.573888e+03   0.482188  6.296784e-01
## sqft_living    1.735211e+02 5.042932e+00  34.408772 4.116870e-251
## sqft_lot       2.458043e-01 5.666528e-02   4.337828  1.447035e-05
## floors        -2.008680e+04 4.084173e+03  -4.918206  8.813409e-07
## waterfront     5.832713e+05 2.009628e+04  29.023837 6.269176e-181
## view           5.987801e+04 2.487632e+03  24.070282 5.697936e-126
## condition      5.401013e+04 2.591284e+03  20.843001  2.539127e-95
## grade          7.665502e+04 2.441131e+03  31.401430 1.311692e-210
## sqft_above     3.174287e+01 5.075131e+00   6.254592  4.078112e-10
## yr_renovated   6.253955e+01 4.009386e+00  15.598287  1.741561e-54
## zipcode       -4.519959e+02 3.814960e+01 -11.847984  2.928890e-32
## lat            6.775054e+05 1.232340e+04  54.977164  0.000000e+00
## long          -3.346280e+05 1.481967e+04 -22.579987 2.649747e-111
## sqft_living15  2.737917e+01 4.028114e+00   6.797019  1.102537e-11
## sqft_lot15    -4.784153e-01 8.516808e-02  -5.617307  1.969091e-08
```



2.4.1 Comparison of the different RMSE values using the `dl_test` set.

Analysis Method	**RMSE**	**R^2**
Bivariate	2.349861710^{5}	0.4930741
Multivariate	2.205960110^{5}	0.5409679
Stepwise	1.82966710^{5}	0.6799968

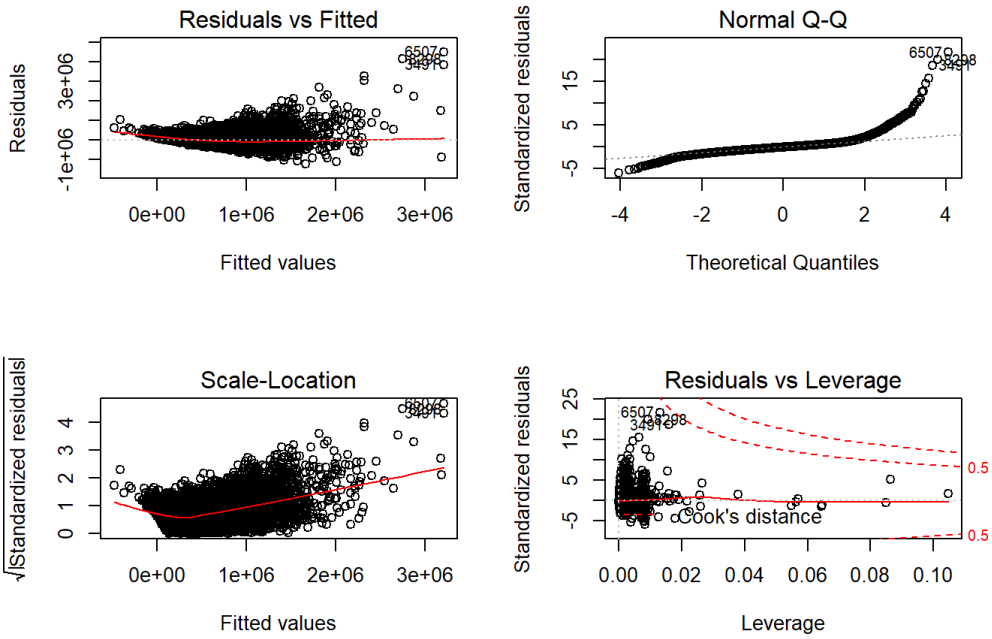
This clearly shows that using most of the predictors available gives the smallest RMSE and best R^2 value.

2.5 Results

Now that we have a potential algorithm, we need to return to the orginal training `dl` and test `validation` data sets that were partitioned in the beginning.

We retrain the *stepwise* algorithm on the full `dl` set, and use this new trained algorithm to predict against the `validation` test set that was partitioned at the beginning of the analysis.

```
##                   Estimate   Std. Error     t value      Pr(>|t|)
## (Intercept)   -2.962892e+07 2.990711e+06  -9.9069818  4.403321e-23
## bedrooms      -3.241614e+04 2.054623e+03 -15.7771671  9.911025e-56
## bathrooms      1.385079e+03 3.353536e+03   0.4130206  6.795961e-01
## sqft_living    1.708465e+02 4.756902e+00  35.9155031 1.450063e-273
## sqft_lot       2.252902e-01 5.258625e-02   4.2842029  1.842769e-05
## floors        -1.929954e+04 3.838036e+03  -5.0284934  4.987471e-07
## waterfront     5.823870e+05 1.898401e+04  30.6777629 7.187442e-202
## view           5.896572e+04 2.331401e+03  25.2919689 6.870697e-139
## condition      5.294752e+04 2.420905e+03  21.8709583 9.000336e-105
## grade          7.746647e+04 2.291316e+03  33.8087259 1.601664e-243
## sqft_above     3.090140e+01 4.770229e+00   6.4779697  9.519361e-11
## yr_renovated   6.211098e+01 3.788514e+00  16.3945477  5.290143e-60
## zipcode       -4.399854e+02 3.566656e+01 -12.3360768  7.819060e-35
## lat            6.762437e+05 1.154411e+04  58.5791038  0.000000e+00
## long          -3.275863e+05 1.384012e+04 -23.6693179 3.980258e-122
## sqft_living15  2.735194e+01 3.757500e+00   7.2792933  3.483635e-13
## sqft_lot15    -4.469734e-01 8.023387e-02  -5.5708824  2.568112e-08
```



Final RMSE with the validation set is **1.952778810^{5}**.

3 Conclusion and Future Directions

The final RMSE is consistent with RMSE from the Stepwise regression predictors in the training set, however, an RMSE in the hundreds of thousands leaves room for improvement. Whether more advanced regression analysis could improve the outcome is unclear; another possibility is that the predictor variables captured in the data set are insufficient for more effective predictive algorithms. This seems especially possible given the continued heteroscedasticity in the residual plots, even when most of the predictors have been used.

Indeed, one set of predictors that would be useful, but likely will never be available, centers around the concept of user bias. User bias is relatively easy to capture when users actively rate or categorize, for instance on Yelp or Amazon reviews. Even final purchase behavior over time can give insight into a user's preference. Unfortunaly, although real estate presumably suffers from user bias (the emotional or "gut instinct" involved for most people in purchasing a home), the limited frequency of purchase behavior prevents capture of these predictive variables.

What user variables might be indicative of how much a “user” or potential buyer is willing to pay? One can imagine that income, family/marital status, reason for purchase (i.e. moving to a new area? need to purchase quickly for a 1031 exchange?) might be information with predictive value. Since house purchasing is behavioural, other purchase decsions may also be predictive. For instance, automobiles, much like houses, require some form of negotiations on price before purchase. Behavior and decisions in the automobile purchase may indicate what type of property a person may buy, and how much that person may be willing to offer for the property.