Multiple Linear Regression Analysis to Predict Real Estate Prices located in King County, USA

MDA 9159 Data Analysis Project Report

Western University
Master of Data Analytics

Moustafa Shaker
Huicong (Ivy) Wu

Abstract

In this project, we explored multiple linear regression models to predict home prices given data that was collected in King, County. We have shown that Least Absolute Shrinkage and Selection Operator (LASSO) regression model yields the best results using root mean square error as a performance matrix. The combination of feature engineering, the bias introduced by the penalty term of the LASSO regression, and the feature selection carried out by LASSO gave the best model to predict home prices in King, Country. We invite you to read the details in the report below.

Introduction

In this project, we will explore the power of regression analysis to predict home prices using features such as home's location, year built, condition, number of bedrooms, etc. The data we are going to use in this project was collected in King Country, which includes Seattle and Bellevue, for homes sold between May 2014 and May 2015.

We will start by defining the features we have in the original dataset. Following that we will clean and reformat our data to prepare it for regression analysis. In addition, we will explore the possibility of constructing new features that may help us make better predictions on home prices. Once we clean our data and construct any feature that we believe may improve our prediction, we will start our data exploration to gain a higher level of understanding of the data. Following that we define and discuss performance matrix to use in this project to quantify regression models performance. Afterwards, we will explore different ways to build regression models to predict home prices, discuss the advantages and limitations of each method. In closing, we will discuss our findings, inference, model limitations, and report any further questions raised by the project.

Defining Features in the Original Dataset

We will first identify all the features we have in the original dataset and explain what those features are. It's critical to understand the meaning of the features in the dataset in order to be able to make accurate inference from our regression model later when we build it. The table below contains all the features in the original dataset and provide a description of what those features mean.

Feature Name	Description	Data Class
id	A unique identification number for each home sold	Numeric
date	The date of the home sale	Numeric
price	Price paid to purchase the home	Numeric
bedrooms	Number of bedrooms	Integer
bathrooms	Number of bathrooms, where 0.5 accounts for a room with a toilet but no shower	Numeric
sqft_living	Square footage of the home interior living space	Integer

sqft_lot	Square footage of the land lot	Integer
floors	Number of floors	Numeric
waterfront	A dummy variable for whether the home overlooks a waterfront or not	Integer
view	An index from 0 to 4 of how good the view of the home. A higher score means better view	Integer
condition	An index from 1 to 5 to represent the condition of the home relative to year built and grade. A higher score means a better condition. You can find a detailed description of what each score means under Building Condition Index in Appendix A	Integer
grade	An index from 1 to 13 to represent the construction quality of improvements. A higher score means better quality of improvements. You can find a detailed description of what each score means under Building Grade Index in Appendix A	Integer
sqft_above	Square footage of the interior housing space that is above ground level	Integer
sqft_basement	Square footage of the interior housing space that is below ground level	Integer
yr_built	The year the home was initially built	Integer
yr_renovated	The year of the home's last renovation	Integer
zipcode	Zip code of the home location	Integer
lat	Latitude of the home location	Numeric
long	Longitude of the home location	Numeric
sqft_living15	Average square footage of the interior housing space for the nearest 15 homes	Numeric
sqft_lot15	Average square footage of the land lots of the nearest 15 homes	Numeric

Table 1 Descriptions of Features in the Original Data Set

Data Preprocessing and Feature Construction

Before diving in our explanatory analysis, we will first spend some time cleaning and preparing the data to enable us to extract useful insights to predict home prices. First, we will investigate if there

are any missing values for any of the features in our data. Using R programming language to carry out this task we found that luckily there are no missing values for any of the features. Summary of variables is shown in table 2 under Appendix B. Furthermore, analyzing frequency graphs, which you can find attached with the R code, of the features in the dataset we found that some home shows a frequency of zero for bedrooms and bathrooms. We think that this may be due to data entry error. Therefore, data rows that show zero for bathrooms and bedrooms were removed.

Next we will carefully examine each of the original features in the dataset and determine the best way to extract useful insights from this data to predict home prices. To extract the most useful insights from our data we will explore the possibility of constructing new features, changing our features format and dropping features that seems unrelated to home prices.

ID

This feature does not relate in any way to home prices. For that reason, we are going to drop it off from our data.

Home sale date and year of build

Since our project is focused on regression analysis rather than time series analysis, we will change the date format in the data. Our goal is to extract useful insights from the dates we have for home sale and year of build without including this information in a date format. To do this we will construct a new feature and call it home age(age). The new feature will represent the age of the home at the time of sale. Also, dropping the date format will later enable us to make better inference from our regression model.

Home sale date and year of last renovation

Following similar rational to the one we discussed above we decided to construct a new feature from the date of the home sale and the date of the last renovation and call it renovation age(renov_age). This new feature will show how many years it has been between the home sale and the last renovation of the home.

Zip Code

Zip code numbers are not assigned in a way that makes them mean anything. For example, what does it mean when a zip code goes up by 1 number or go down by one number? It doesn't really have any meaning. Treating zip codes as continuous variables wouldn't be wise because it will be impossible to make inferences from our regression model about how zip code relates to price. It makes sense to treat zip codes as categorical variables. This way we will be able to make sense of zip code effect on home prices in our regression analysis. However, if we treat zip codes as categories we run into another problem. Our database contains 70 unique zip code value and if we convert them into categorical binary form, then we would have 69 new features added to our data. Our data contains information about 1000 homes, so adding 69 features plus the other features we have in the model means that the total features we have is almost tenths of the size of our data. After splitting our data in training and testing sets the available data to train our regression model will be even less. So, our regression model won't have enough data points to make good estimates for regression coefficients.

Real estate location is an important factor in determining price. So how could we use our data to extract insights about real estate location without having too many explanatory variables that can lead to poor estimates of regression coefficients? This question leads us to our next discussion of latitude and longitude features in our dataset. However, before we move on to that part, we wanted to note to the reader that we decided to drop off the zip code column from our data and focus on extracting location insights from latitude and longitude as we explain below.

Latitude and longitude

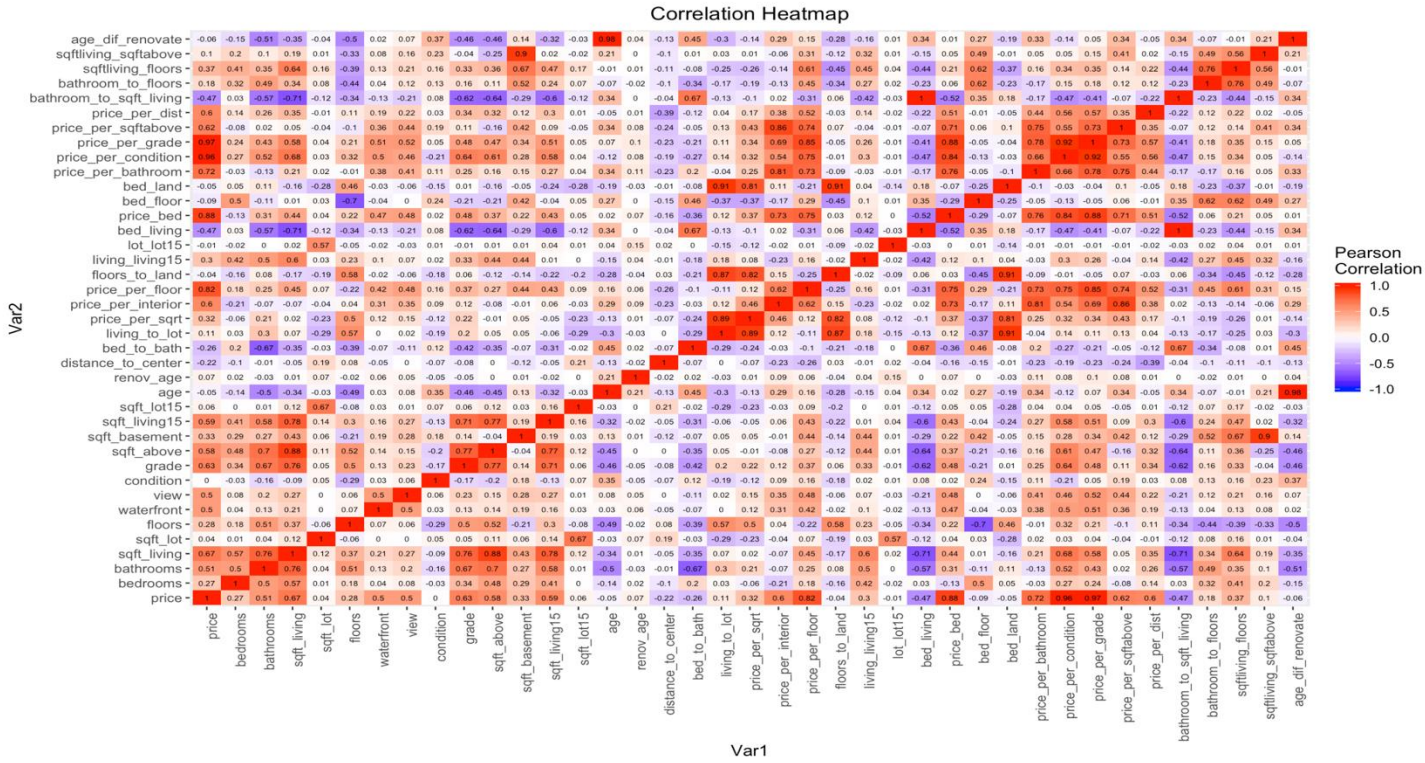
Following up on our argument that real estate location plays an important factor in its price. We will use the latitude and longitude coordinates for each home in the dataset to engineer a new feature. A feature that we believe can help our regression model make better prediction on home prices. This feature is the distance between the home and the city center. We will calculate the distance between the homes in the data set and Bellevue City Centre. Since the data we have are for homes in Bellevue city. Upon a closer look at our data we found that the closest home to Bellevue city center was only 0.648 km and the farthest one from Bellevue city center was 54.34 km.

Additional Features

We decided also to include a few more features that we believe can improve the predictions of our regression model. When we do our feature correlation analysis and build the regression model, we will be able to determine if the features we added or engineered in this section was useful in improving the prediction of home prices. For now we will assume that the ratio between number of bedrooms and bathrooms(bed_to_bath), ratio between the square footage of the interior living space and the land lot(living_to_lot), price per interior living space square footage(price_per_interior), price per floor(price_per_floor), price per land lot square footage(price_per_lot), ratio between number of floors and land lot square footage(floors_to_land), ratio between living space square footage and average living space square footage of the nearest 15 homes(living_living15), ratio between the land lot square footage and the average land lot square footage of the nearest 15 homes(lot_lot15), ratio between number of bedrooms and square footage of living space(bed_living), ratio between number of bedrooms and square footage of land lot(bed_land), price per bedroom(price_bed), number of bedrooms per floor(bed_floor), price per bathroom(price_per_bathroom), price per condition score(price_per_bathroom), price per grade score(price_per_grade), price per square footage of living space above ground(price_per_sqftabove), price per kilometer from city center(price_per_dist), ratio between number of bathrooms and square footage of living space(bathroom_to_sqft_living), ratio between number of bathrooms and number of floors(bathroom_to_floors), ratio between square footage of living space and number of floors(sqftliving_floots), ratio between square footage of living space and square footage above the ground(sqft_above), and the price per land square footage(price_per_sqft) can help the regression model in making better price predictions. Therefore, we will construct those features and add them to our data.

Data Exploration

We will start our exploratory data analysis by creating heatmap to visualize the correlation between the features in the dataset.



Graph 1 Correlation Heatmap of Variables

Our heat map shows that price per grade score, price per condition score, price per bedroom, price per floor, square footage of living space are the most positively correlated features to the price. On the other hand, features like number of bedrooms, square footage of basement, ratio of between number of bathrooms to number of floors are not strongly correlated to the price. Additionally, features like distance from the city center and bed to bath ratio show a negative correlation to the price. Note that a positive correlation means that when the feature goes up, the price goes up. A negative correlation means that as these features go up, the price go down. For example, the larger the distance from the city center gets the cheaper the price becomes.

It's worth mentioning that strong correlation between explanatory features may cause issues in regression analysis due to multicollinearity. Multicollinearity means that one predictor variable can be linearly predicted from the others with a substantial degree of accuracy. When multicollinearity exists the coefficients of multiple regression may change erratically in response to small changes in the model or the data. The prediction of a model can still be accurate when multicollinearity is present among the predictor variables. However, the estimate of the coefficients for those correlated predictors is not likely to be meaningful.

Next we will look at the frequency of the some of the features we have in our dataset through the histograms below.

Graph 2 Frequency (Count) Table of Continuous Variables

The frequency histograms show us the distribution of our numeric features and enable us to gain a higher-level understanding of our data. We can see that the most frequent condition score in our data is 3 and 4, the price ranges from $90,000 all the way to $5,110,800, the most frequent price per square footage of interior living space is around $200, majority of homes in our data does not have a scenic view, the most frequent bed to bath ratio is around 1.5, etc.

Defining Performance Metrix

For this project we will use Root Mean Square Error (RMSE) to quantify our regression models performance. RMSE is the standard deviation of the residuals, which are a measure of how far the regression line data points are. Basically, RMSE tells us how concentrated the data are around the line of best fit. RMSE is calculated by the formula below.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

where n is the number of data points. In our case, that would be the number of homes we collected information about, ŷ in the formula represents the predicted price of the home by the regression model, and y is the actual price of the home.

To get the most accurate performance of the regression models, that we are going to build in the next section, we are going to evaluate the RMSE using k-fold cross validation. Cross validation is a technique for assessing how the results of a regression model will generalize to unseen data. This technique gives us an unbiased estimate of how the model will perform in general when used to make predictions on data not used to train the model. The procedure to perform cross-validation includes

defining a parameter called k that refers to the number of groups that a given data sample is to be split into. In this project, we will set k equal to 20 which will split our data in 20 folds. Nineteen of those folds will be used to train the model, then the 20th fold will be used to evaluate the model performance using RMSE. Then the process will be repeated but in the next run the fold we used for testing will be used for training and we will pick another fold that we haven't used for testing to test the model. This process will be repeated 20 times since we set k to 20. Using cross-validation with k equal to 20 will return 20 RMSEs, one for each testing fold. We will then take the average of those RMSEs to get a final estimate of the model RMSE.

Building Regression Model

In this section we will build regression models to predict a home price based on the features we have in our original data set and the features that we constructed in the Data Preprocessing and Feature Construction section of this document. We will first start by discussing ordinary least square regression analysis, how it works, and why this approach may not be a wise choice for our data. Then we will discuss regularization, a technique that can overcome some of the short outcomes of the ordinary least square regression. Later in this section we will build two models using different approaches to the regularization technique. We will build a least absolute shrinkage and selection operator (LASSO) regression model and a Ridge regression model. At the end of this section we will discuss the results of both models and select our model of choice to predict home prices.

Ordinary Least Square

Ordinary Least Square (OLS) is a linear least squares method that estimates the parameters of a linear regression equation. OLS select the parameters of the features in a regression model by minimizing a loss function of least squares. Minimizing the least squares means minimizing the sum of squares of the difference between the predicted variable, in our case that is the price of home, and the actual observed variable, which is the actual price of the home.

Even though ordinary least square is a popular approach to regression analysis, we believe that it's not a good choice for our data. To understand why we will first discuss the concept of bias-variance tradeoff. Let's start by defining bias. Bias in this context means the difference between the average prediction of a regression model and the actual value that we are trying to predict. Regression models with high bias give little attention to the training data and oversimplifies the model. This leads to high error on training and testing data. In our case that means we will have a high RMSE for both the training data and testing data. Now we will turn our attention to define variance. Variance is the variability of the regression model prediction for a given data point. In other words, it's a value that tells us the spread of our data. Regression models with high variance pays a lot of attention to training data and that leads to poor performance on data that the model hasn't seen before. In our case for example, if the model has a high variance that will lead to low RMSE on training data but high RMSE on test data.

If a regression model is too simple meaning it has very few features, then it may have high bias and low variance. On the other hand, if a regression model has many features that leads to high variance and low bias. This is what is known as the bias-variance tradeoff. We can't have a model that is simple and complex at the same time. Building a model that gives great prediction results require finding the right balance without overfitting or underfitting the data.

We have in total 38 features to predict home prices. The 38 features include the features we constructed earlier. Fitting a regression model with ordinary least square with all the 38 features to predict home price may lead to high variance and low bias. For this reason, we need to find another approach that let us find the right balance between bias and variance to get the best possible predictions. Regularization is a technique used to solve the overfitting related to high variance by introducing bias. Regularization penalize or adjust the weights of the independence variables so that it makes a good prediction on data that the model hasn't seen before. In this coming section we are going to discuss two methods of regression analysis, the least absolute shrinkage and selection operator (LASSO) regression analysis and the Ridge regression analysis. Both of LASSO and Ridge use regularization but in different ways. We will then build two regression models using both of those techniques.

Least Absolute Shrinkage and Selection Operator (LASSO)

LASSO is a type of regression analysis that incorporate regularization to enhance the prediction accuracy of a regression model. Moreover, LASSO also performs feature selection since it can shrink less important features' coefficient to zero. The loss function of LASSO is defined as

$$Loss = Error(y, \hat{y}) + \lambda \sum_{i=1}^{N} |w_i|$$

where the 'Error' term in the equation is equal to the loss function with no regularization. The penalty that LASSO adds is the right most term in the equation. Its equal to lambda times the sum of the absolute value of magnitude of features' coefficients. When the tuning parameter lambda is small, the result is essentially the least squares estimates. As lambda increases, shrinkage occurs so that variables that are at zero can be thrown away. In this project, we will choose the tuning parameter lambda by cross validation.

Now that we have defined LASSO, how it works and its advantages over regression models that doesn't use regularization such as ordinary least square. We will now build our model using LASSO regression. Using R Studio, we fit a LASSO regression using cross-validation with 20 folds and with all the features we have in our data. Here is the summary of the of the resulting model:

```
39 x 1 sparse Matrix of class "dgCMatrix"
                                s0
(Intercept)          -5.810314e+05
bedrooms             -1.800617e+04
bathrooms             .
sqft_living           .
sqft_lot             -4.915414e-04
floors                .
waterfront            2.805159e+04
view                  8.405774e+02
condition             4.418049e+04
grade                 5.395014e+04
sqft_above            .
sqft_basement         2.388904e+01
sqft_living15        -2.485326e+00
sqft_lot15           -6.488459e-02
age                   .
renov_age            -9.450826e+01
distance_to_center    9.271963e+01
bed_to_bath          -1.086930e+03
living_to_lot        -1.399968e+04
price_per_sqrt        .
price_per_interior   -2.712136e+02
price_per_floor       7.088349e-02
floors_to_land        5.990229e+06
living_living15      -1.660493e+04
lot_lot15             3.407304e+02
bed_living            8.975378e+05
price_bed             2.185940e-01
bed_floor             .
bed_land              .
price_per_bathroom    .
price_per_condition   1.089287e+00
price_per_grade       6.478881e+00
price_per_sqftabove  -1.981111e+02
price_per_dist        7.438606e-02
bathroom_to_sqft_living  5.273208e+07
bathroom_to_floors    .
sqftliving_floors    -1.857674e+01
sqftliving_sqftabove  1.343634e+04
age_dif_renovate      5.989613e+00
```

Graph 3 R Code Output for Coefficients of Lasso Regression

Note that LASSO shrunk 10 of our feature coefficients to zero. Features such ratio between number of bathrooms to floors, total price per number of bathrooms, ratio between number of bedrooms to floors, etc. were determined by LASSO regression as not important to predict price given the other features in the model. Additionally, note that 19 of the features we constructed from the original features in the dataset were considered important features by the LASSO. Features such as distance from city center, bed to bath ratio, time since last renovation or renovation age, etc. were important to make good home price predictions by the model.

Using the defined performance matrix, we will analyze how accurate our LASSO regression in making home price estimates. The 20 folds cross validation returned an average RMSE of $35,567.33. That means that the square root of the average squared difference between our predicted price and the actual price was $35,567.33. Given the price range we have in the data set that goes from $90,000 to $5,110,800, the RMSE of the LASSO model is pretty good. However, we will see if we can do better by the Ridge regression.

Ridge Regression

Like LASSO regression, the Ridge regression is a type of regression analysis that incorporate regularization to enhance the prediction accuracy of a regression model. The key difference between LASSO and Ridge is that LASSO shrinks the less important predictors coefficient exactly to zero. That is, the LASSO can perform variable selection. In Ridge regression, even if λ is large, the final model will include all the features. The loss function of Ridge is defined as

$$Loss = Error(y, \hat{y}) + \lambda \sum_{i=1}^{N} w_i^2$$

where the ‘Error’ term in the equation is equal to the loss function with no regularization. The penalty that Ridge adds is the right most term in the equation. Its equal to lambda times the sum of the squared magnitude of features’ coefficients. When lambda is equal to zero, then there will be no difference between Ridge and ordinary least square. However, when lambda is very large that can add too much weight and it will lead to under fitting. In this project, we will choose the tuning parameter lambda by cross validation.

Using R Studio, we fit a Ridge regression using cross-validation with 20 folders and all the features we have in our data. Here is the summary of the of the resulting model:

```
39 x 1 sparse Matrix of class "dgCMatrix"
                                      s0
(Intercept)             -5.695023e+05
bedrooms                 7.391526e+03
bathrooms                2.913699e+04
sqft_living              2.554867e+01
sqft_lot                -3.653864e-02
floors                   1.993719e+04
waterfront               1.770841e+05
view                     5.181088e+03
condition                3.174058e+04
grade                    2.318776e+04
sqft_above               2.442320e+01
sqft_basement            3.462784e+01
sqft_living15            1.449346e+01
sqft_lot15               4.956853e-02
age                      4.759659e+01
renov_age                2.247144e+02
distance_to_center      -7.789690e+01
bed_to_bath             -2.265842e+03
living_to_lot           -1.185284e+04
price_per_sqrt           8.331961e+01
price_per_interior       2.235730e+01
price_per_floor          2.054653e-01
floors_to_land          -4.341182e+06
living_living15          2.558589e+04
lot_lot15                7.413861e+02
bed_living               2.196813e+07
price_bed                5.483145e-01
bed_floor               -3.527932e+03
bed_land                -5.688666e+06
price_per_bathroom       1.613078e-01
price_per_condition      8.422454e-01
price_per_grade          2.336713e+00
price_per_sqftabove      1.604988e+01
price_per_dist           1.651734e-01
bathroom_to_sqft_living  2.478789e+07
bathroom_to_floors      -5.200965e+03
sqftliving_floors       -1.753618e+01
sqftliving_sqftabove    -2.313341e+04
age_dif_renovate         7.985524e+01
```

Graph 4 R Code Output for Coefficients of Ridge Regression

Unlike the LASSO the Ridge regression still has all the features we included to predict the price of a home. Using the defined performance matrix, we will analyze the accuracy of our Ridge regression in making price estimates. The 20 folds cross validation returned an average RMSE of

$48,271.42. That means that the square root of the average squared difference between our predicted price and the actual price was $48,271.42.

The Ridge regression has an RMSE that is worse than LASSO by $12,704.09. As we mentioned above the key difference between LASSO and Ridge is that LASSO perform feature selection in addition to regularization. Eliminating some of the features has obviously led to better results as we see from the average cross validation RMSE. Simplifying the model by eliminating some of the explanatory features led to less variance and thus better RMSE.

Discussion of Findings and Inference

Introducing bias to the model by regularization and eliminating some of the features led to a low RMSE. The coefficients of the features in the regression may be interpreted to tell us something about how those features affect the price of a home. For example, the grade feature in the LASSO model has a coefficient of 53,950 suggest that for everyone increase in grade the home price will go up by $53,950. The waterfront feature has a coefficient of 28,051 which suggest that a home with a waterfront approximately costs $28,051 more than a home without a waterfront. Additionally, the bedroom to bathroom ratio has a coefficient of -1086 which suggest that when the number of bedrooms increase relative to the number of bathrooms leading to a one-point increase in the ratio of number of bedrooms to number of bathrooms, the home price will decrease by $1086.

The LASSO model gave us the best performance based on the defined performance matrix. We got an RMSE of $35,567.33 based on a 20 folds cross validation. As we mentioned earlier, we believe that this is a pretty good RMSE based on the price range we have in the data which goes from $90,000 to $5,110,800. We also believe that the feature engineering we did earlier in the project has contributed greatly to getting a good performance. To prove our point, we decided to fit a LASSO regression model using a 20-fold cross validation but this time we will fit the model with the original features in the dataset without any feature engineering. Here is a summary of the result we got.

```
19 x 1 sparse Matrix of class "dgCMatrix"
                        s0
(Intercept)   -4.042348e+05
bedrooms      -2.815724e+04
bathrooms      5.557355e+04
sqft_living    1.097536e+02
sqft_lot       8.850163e-02
floors         1.509154e+04
waterfront     1.085872e+06
view           7.485135e+04
condition      2.651096e+04
grade          8.388366e+04
sqft_above     3.254887e+01
sqft_basement  .
yr_built      -2.505543e+03
yr_renovated   1.271815e+01
zipcode       -5.015435e+02
lat            6.765360e+05
long          -1.776451e+05
sqft_living15  4.696826e+01
sqft_lot15     .
```

Graph 5 R code output for Coefficient of Lasso Regression with Original Variables

Using the original features in the dataset the LASSO regression eliminated features such as average square footage of the land lot of the nearest 15 properties and square footage of home basement. The other features were considered important by the LASSO regression. Now let’s

measure the performance of this model using our defined performance matrix. Using R studio to find the RMSE using cross validation we got an RMSE of $199,413.3. As you can see using the same technique but without feature engineering to extract additional insights from the data has led to a worse RMSE by $163,845.97. This shows the importance of data cleaning, formatting and engineering to get good results.

Limitations and Further Questions Raised by the Project

Some of the features we constructed to use in this model such as distance from Bellevue city center make sense only if we are evaluating home prices in King County, but it wouldn't be much useful if we were predicting home prices in Toronto for example. While the features we added did improve the model performance, it also made the model only useful if we are trying to predict home prices in King Country. This is one of the limitations of our model that we wanted to highlight. Additionally, it's important to understand that regression analysis provide predictions not certainties. It would not be appropriate to say that if a home has those features, then according to the model it will has a price of $450,000. It's practically impossible to predict an exact outcome using a fitted model with 100% confidence. However, it's possible to predict an outcome with certain precision using a well fitted model. It's more appropriate to expect a price with a degree of error.

In the beginning of the project we made the decision to drop the zip codes because if we turned the code into dummy variables, we will end up with an additional 69 features. In that case the model won't have enough data points to make good estimates for regression coefficients. The limited availability of data influenced us to take that decision. However, one possibility that was raised during the project to overcome the limited data availability is bootstrapping. Bootstrapping is a technique that is used to estimate statistics on a population by sampling a dataset with replacement. Such a technique can be used to increase our data sample, and thus enable us to use zip code categories as features without compromising on the size of data available to train and test the model. Bootstrapping comes with advantages and disadvantages, but this is one area that this project can expand on in the future to enable the possibility to add features such as zip codes that can help us make better predictions on home prices in King Country.

GitHub link for R code of the project

https://github.com/Waichung1015/Exploratory-Analysis-of-House-Price

Appendix A

The information below were copied from King Country residential glossary of terms, which you can visit at: https://info.kingcounty.gov/assessor/esales/Glossary.aspx?type=r

Building Condition Index:

Relative to age and grade. Coded 1-5.

1 = Poor- Worn out. Repair and overhaul needed on painted surfaces, roofing, plumbing, heating and numerous functional inadequacies. Excessive deferred maintenance and abuse, limited value-in-use, approaching abandonment or major reconstruction; reuse or change in occupancy is imminent. Effective age is near the end of the scale, regardless of the actual chronological age.

2 = Fair- Badly worn. Much repair needed. Many items need refinishing or overhauling, deferred maintenance obvious, inadequate building utility and systems all shortening the life expectancy and increasing the effective age.

3 = Average- Some evidence of deferred maintenance and normal obsolescence with age in that a few minor repairs are needed, along with some refinishing. All major components still functional and contributing toward an extended life expectancy. Effective age and utility are standard for like properties of its class and usage.

4 = Good- No obvious maintenance required but neither is everything new. Appearance and utility are above the standard and the overall effective age will be lower than the typical home.

5= Very Good- All items well maintained, many having been overhauled and repaired as they have shown signs of wear, increasing the life expectancy and lowering the effective age with little deterioration or obsolescence evident with a high degree of utility.

Building Grade Index:

Represents the construction quality of improvements. Grades run from grade 1 to 13. Generally defined as:

1-3= Falls short of minimum building standards. Normally cabin or inferior structure.

4= Generally older, low quality construction. Does not meet code.

5= Low construction costs and workmanship. Small, simple design.

6= Lowest grade currently meeting building code. Low quality materials and simple designs.

7= Average grade of construction and design. Commonly seen in plants and older subdivisions.

8= Just above average in construction and design. Usually better materials in both the exterior and interior finish work.

9= Better architectural design with extra interior and exterior design and quality.

10= Homes of this quality generally have high quality features. Finish work is better, and more design quality is seen in the floor plans. Generally, have a larger square footage.

11= Custom design and higher quality finish work with added amenities of solid woods, bathroom fixtures and more luxurious options.

12= Custom design and excellent builders. All materials are of the highest quality and all conveniences are present.

13= Generally custom designed and built. Mansion level. Large amount of highest quality cabinet work, wood trim, marble, entry ways etc.

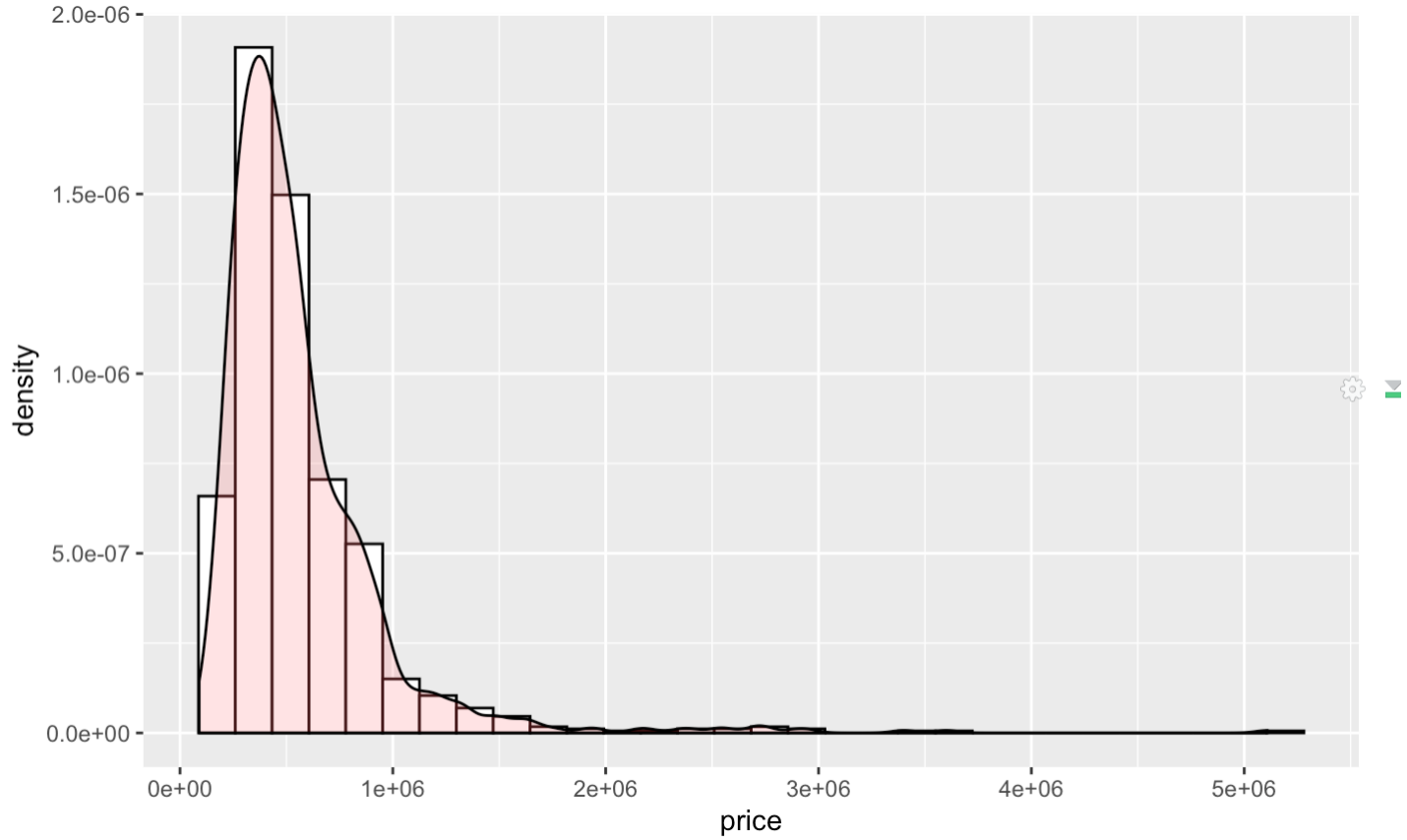
Appendix B

Table 1 Summary of variables

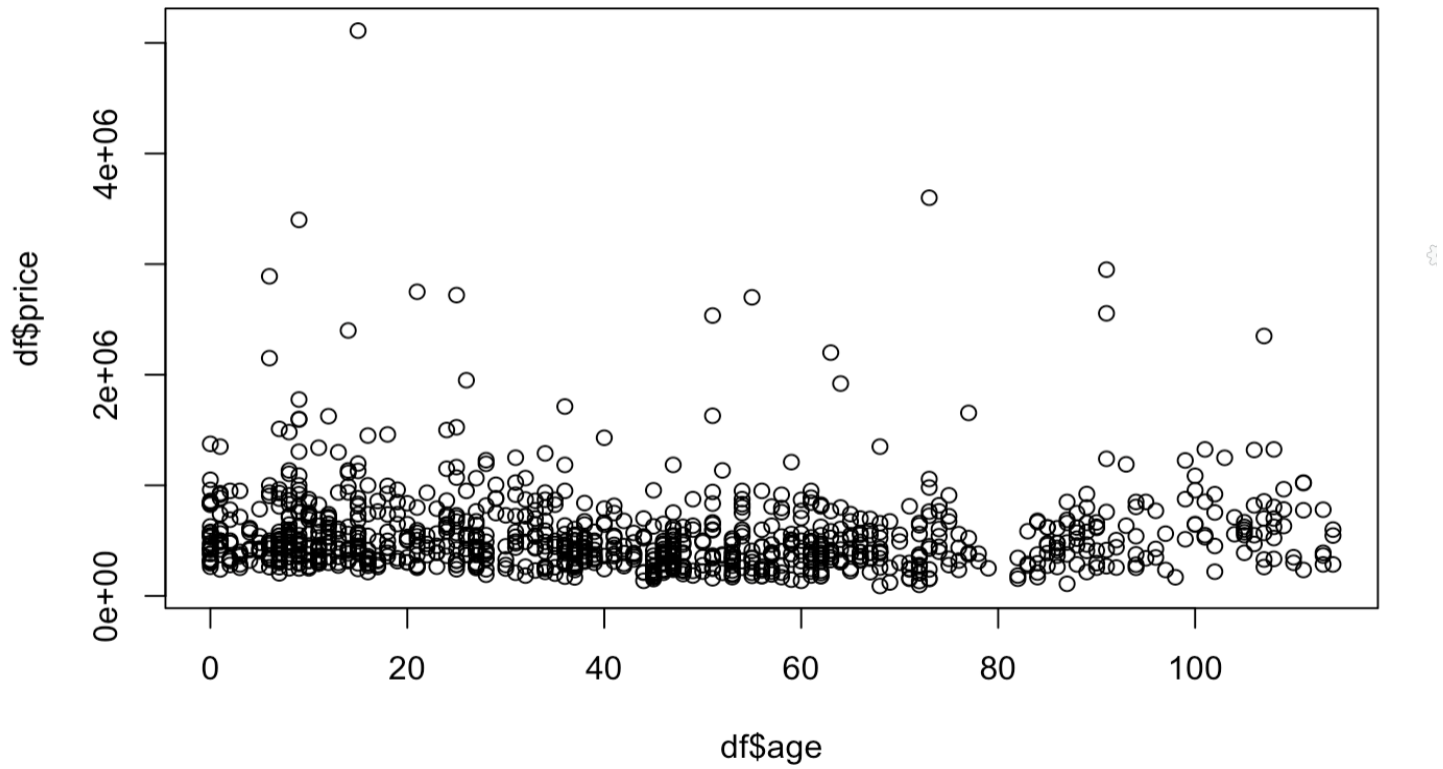
variable <chr>	q_zeros <int>	p_zeros <dbl>	q_na <int>	p_na <dbl>	q_inf <int>	p_inf <dbl>	type <fctr>	unique <int>
id	0	0.0	0	0	0	0	numeric	1000
date	0	0.0	0	0	0	0	numeric	13
price	0	0.0	0	0	0	0	numeric	614
bedrooms	1	0.1	0	0	0	0	integer	9
bathrooms	1	0.1	0	0	0	0	numeric	18
sqft_living	0	0.0	0	0	0	0	integer	351
sqft_lot	0	0.0	0	0	0	0	integer	809
floors	0	0.0	0	0	0	0	numeric	5
waterfront	986	98.6	0	0	0	0	integer	2
view	893	89.3	0	0	0	0	integer	5
condition	0	0.0	0	0	0	0	integer	4
grade	0	0.0	0	0	0	0	integer	9
sqft_above	0	0.0	0	0	0	0	integer	319
sqft_basement	620	62.0	0	0	0	0	integer	137
yr_built	0	0.0	0	0	0	0	integer	113
yr_renovated	953	95.3	0	0	0	0	integer	30
zipcode	0	0.0	0	0	0	0	integer	70
lat	0	0.0	0	0	0	0	numeric	888
long	0	0.0	0	0	0	0	numeric	396
sqft_living15	0	0.0	0	0	0	0	integer	283
sqft_lot15	0	0.0	0	0	0	0	integer	799

21 rows

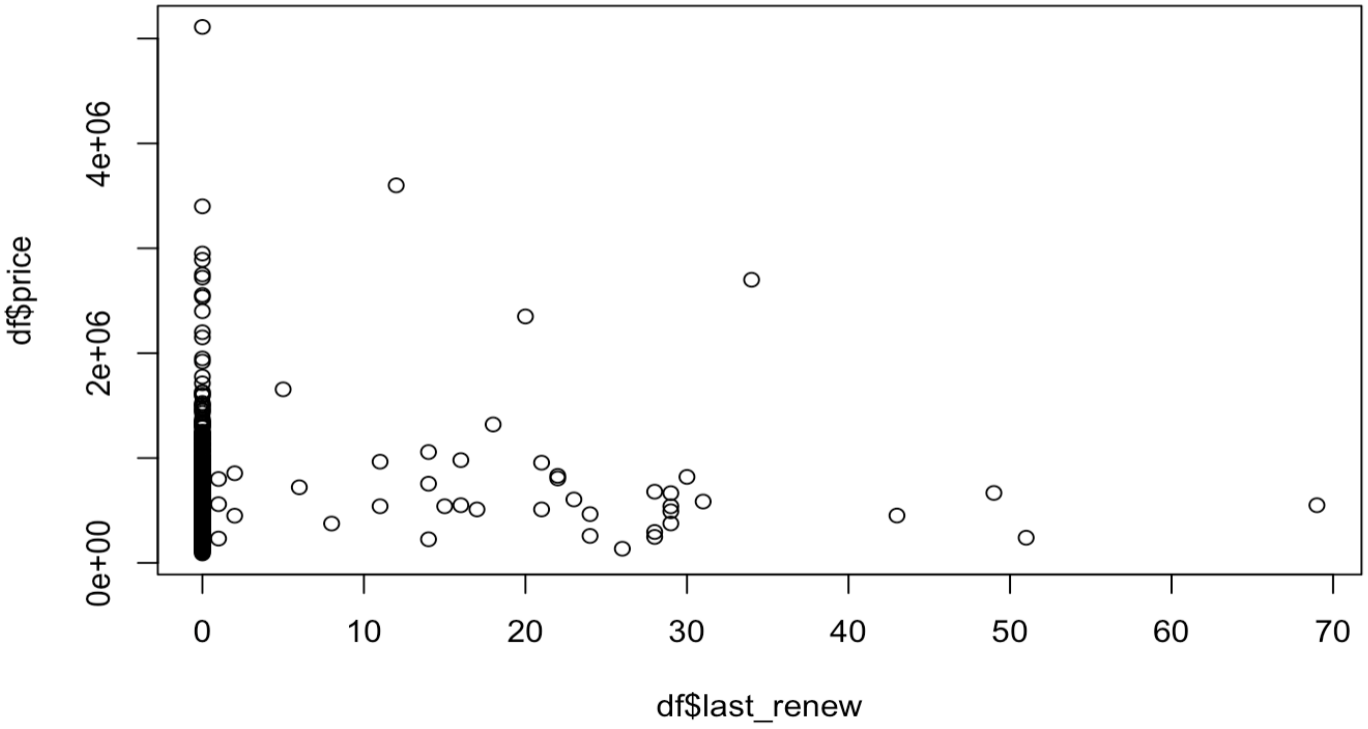
Graph 1 Distribution of Price



Graph 2 Distribution of age



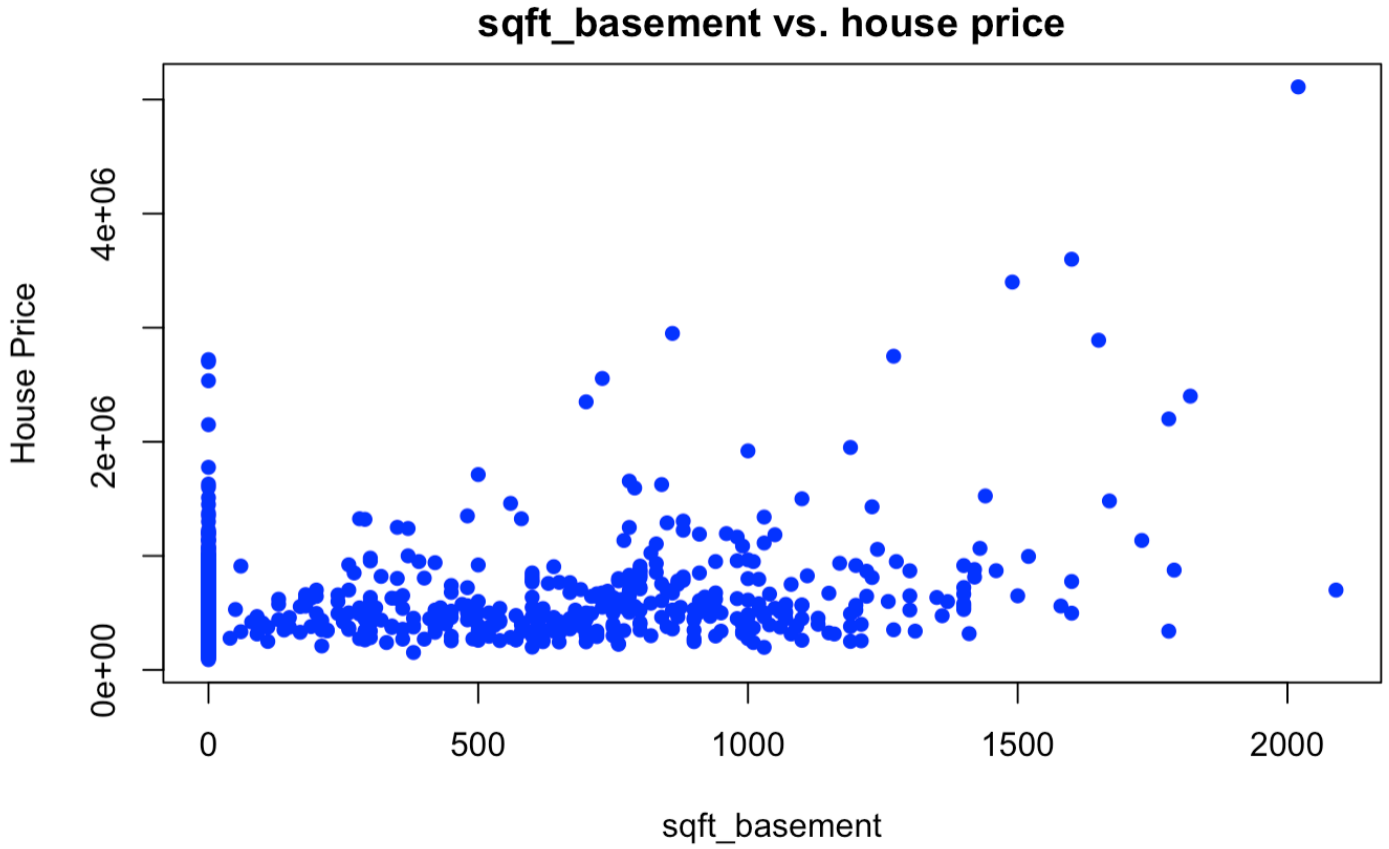
Graph 3 Distribution of last_renew



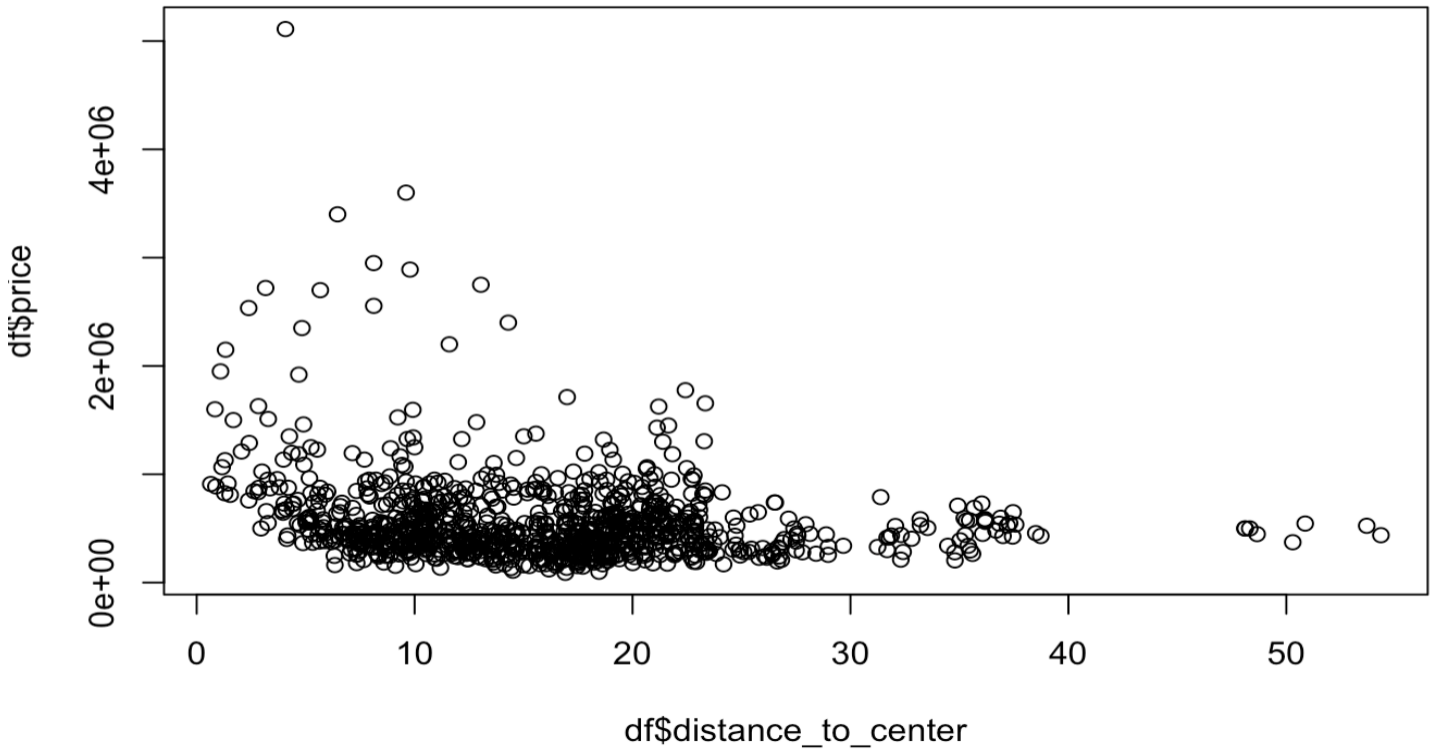
Graph 4 Distribution Plot of sqft_living

Graph 5 Distribution Plot of sqft_above

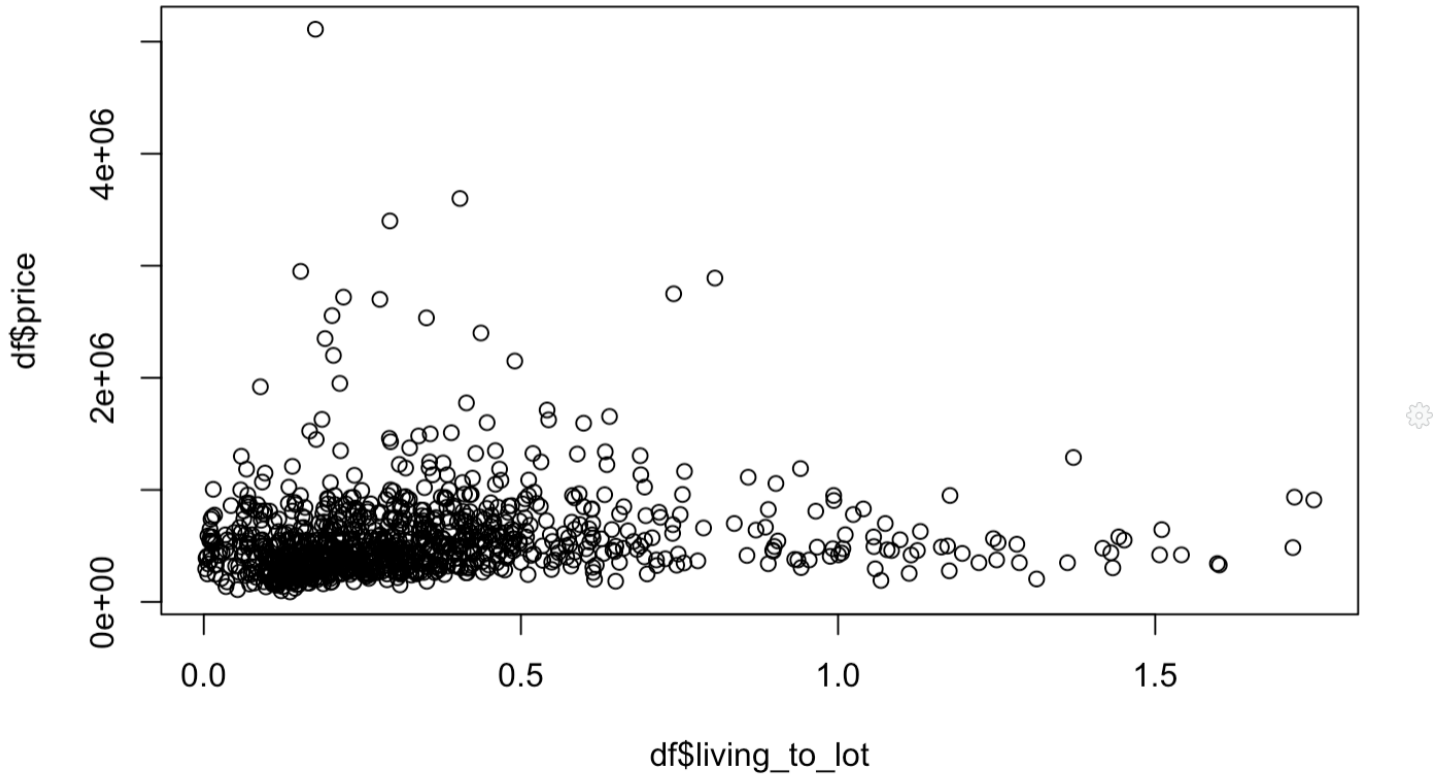
Graph 6 Distribution Plot of sqft_basement



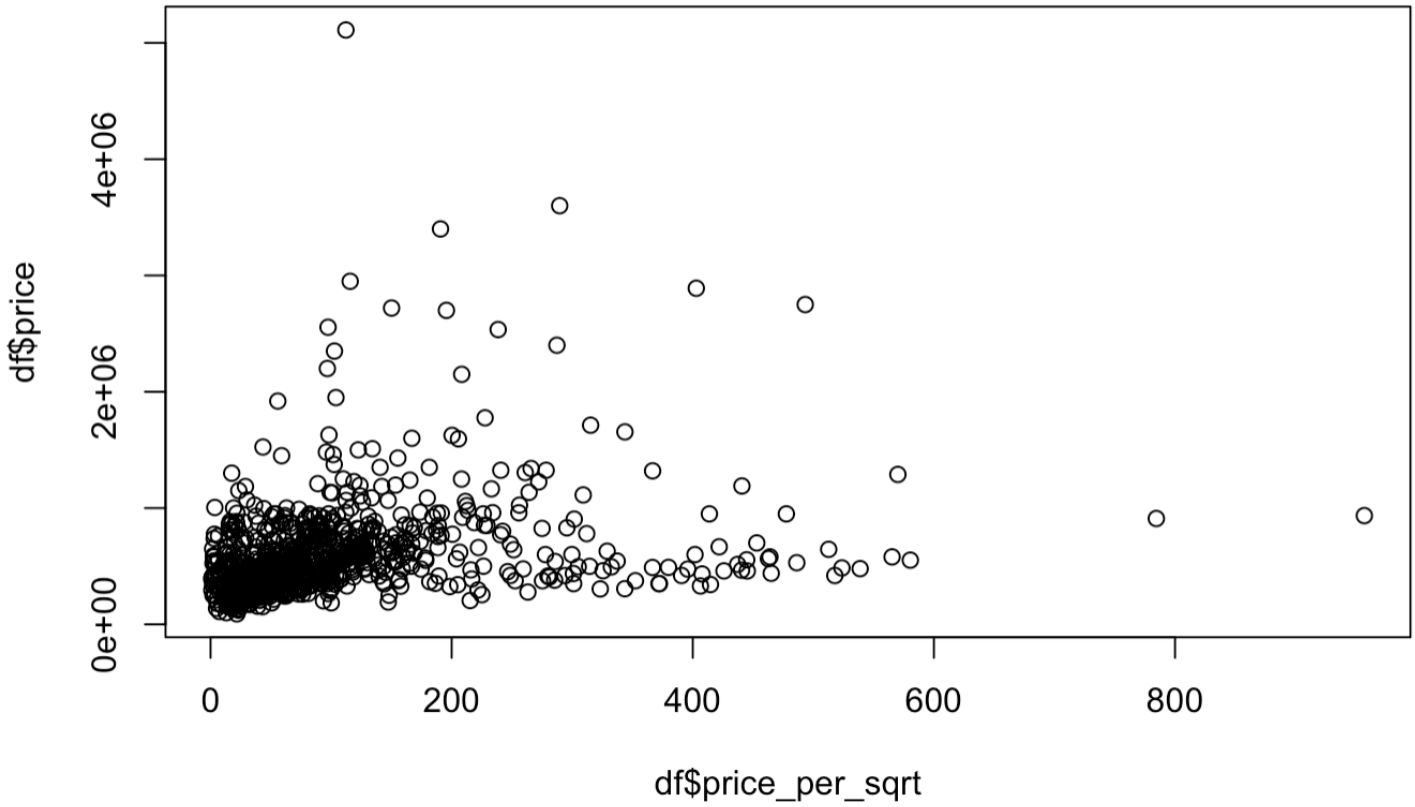
Graph 7 Distribution Plot of distance to center



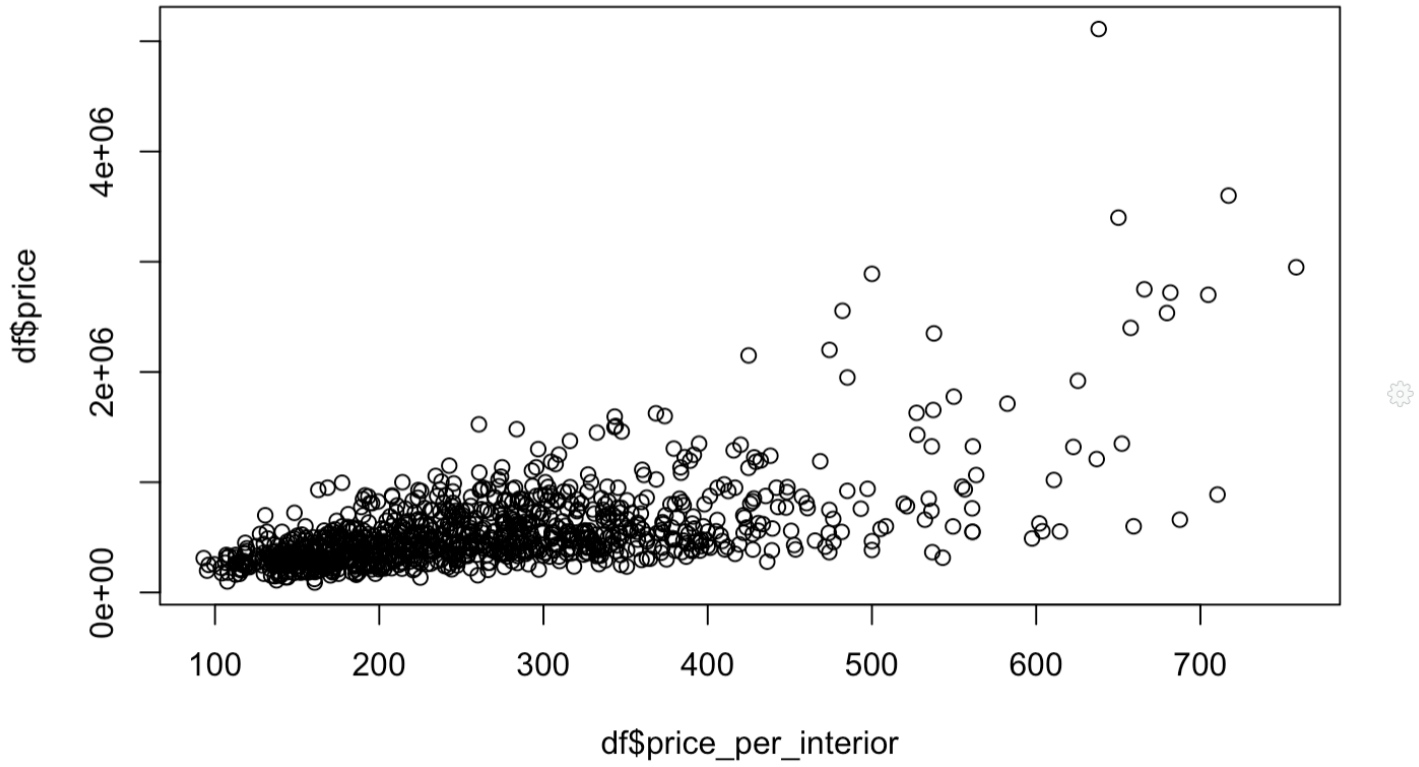
Graph 8 Distribution Plot of Living to Lot



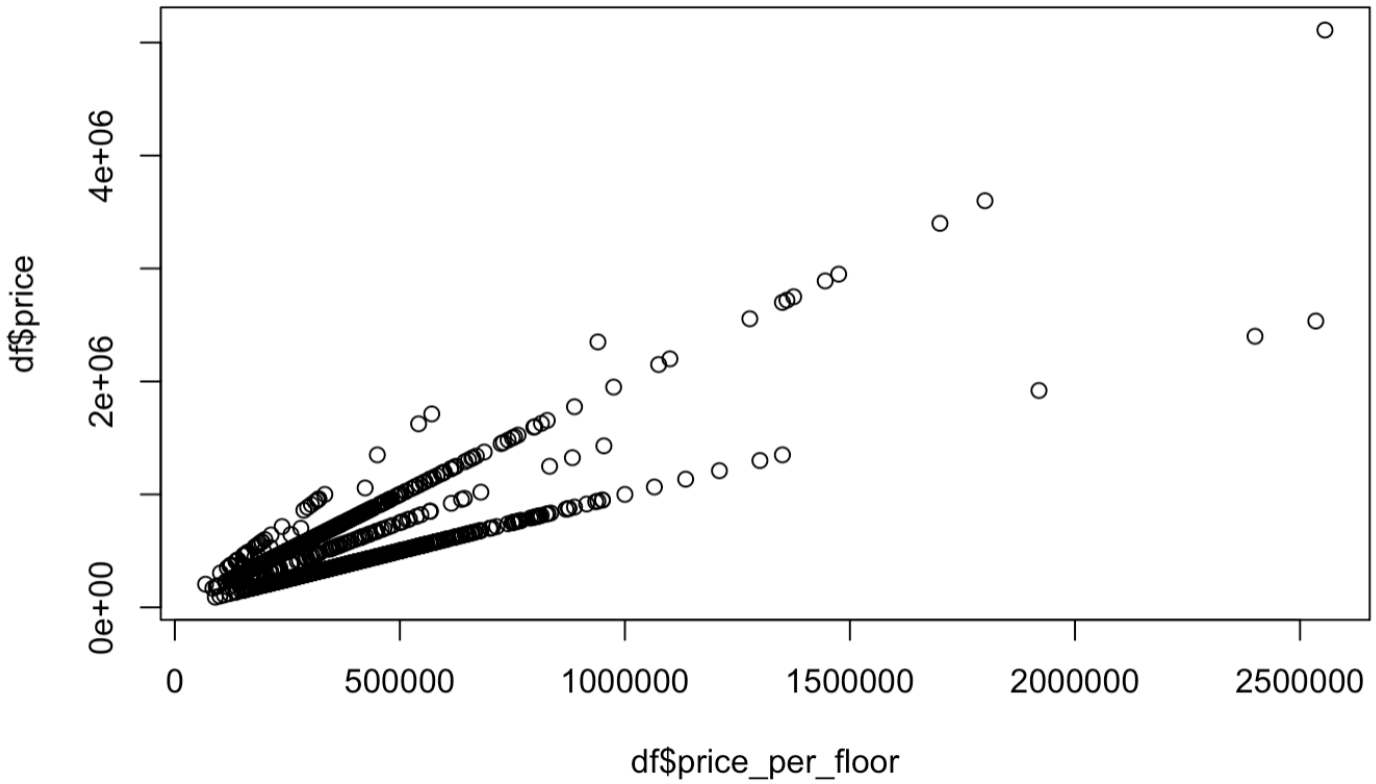
Graph 9 Distribution Plot of Price Per Sqrt



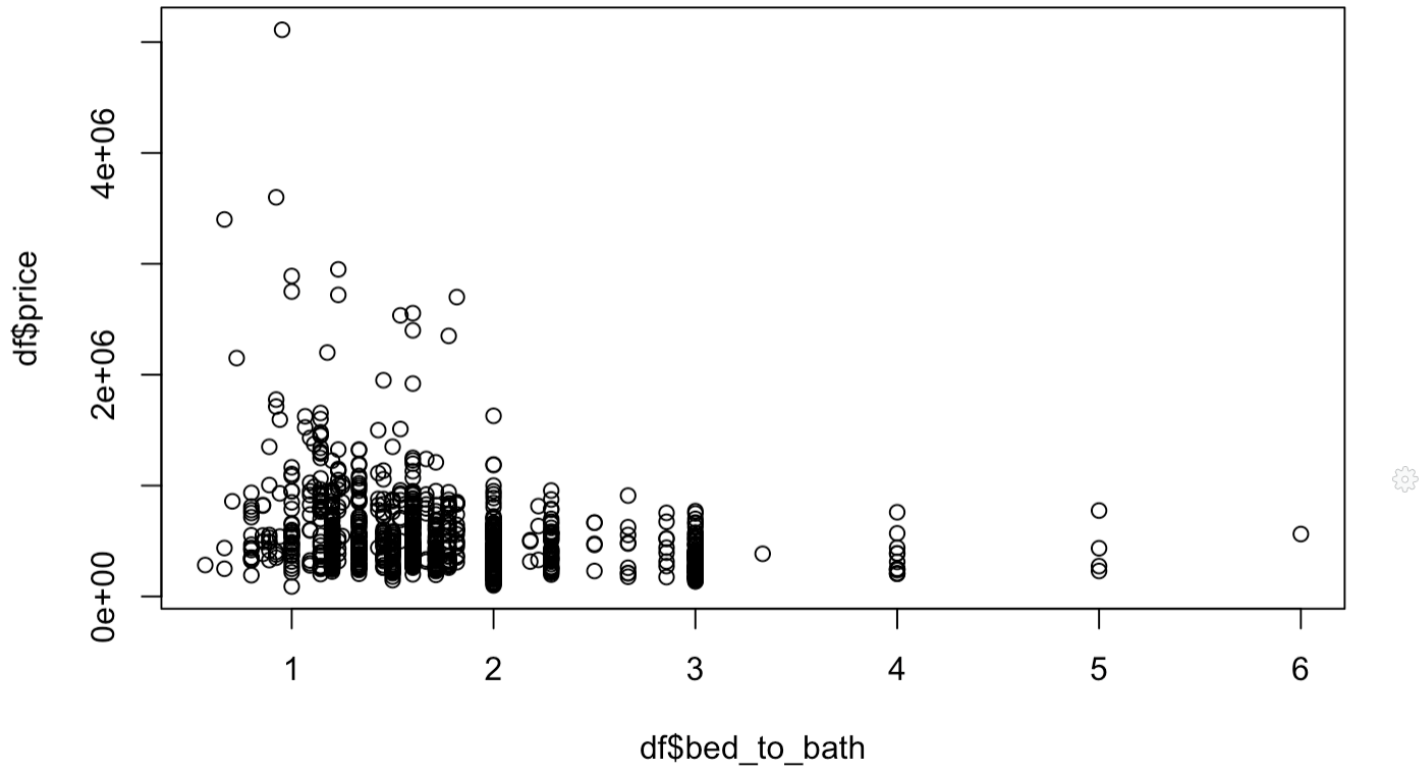
Graph 10 Distribution Plot of Price Per interior



Graph 11 Distribution Plot of Price Per Floor



Graph 12 Distribution Plot of Bed to Bath



Graph 13 Distribution Plot of age_dif_renovate