DC Properties Qualification's Binary Logistic Regression Report

Aaron Niecestro

May 7, 2019

Introduction

This report is about using binary logistic regression to figure out which variables and their effect those variables have on what makes a property qualified to sell. Qualified property means that the paperwork needed to sell a house, the deed of the property, approval from banks (if needed), etc. is completed, and the property inspection is passed. The reason this type of study is being conducted is that my partner and I both go to American University which is in the District of Columbia. We thought and believed that since a lot of students live in either DC or one of the surrounding states (West Virginia, Virginia, Maryland), this would be something we could analyze and learn from. Also, we felt that maybe some of our fellow students will be property owners or apartment renters in the coming future, if not already, and this would give insight into whether they will be able to pick a qualified and right place for themselves to live.

The next step was finding data that we could use for a logistic regression model. We found our data relatively fast from Kaggle D.C. Residential Property, and agreed upon using binary logistic regression analysis, although we could have also used nominal multinomial and ordinal logistic regression by using a different response variable then qualification. Once the data and type of logistic regression analysis were decided, the next step was coming up with questions we wished to answer. The questions we created and tried to answer were as follows: 1) What does the Qualification column in the dataset mean? 2) What qualifies a residential property to be sold on the housing market? 3) Is the property pricing the most important factor in determining whether a property is qualified to go on the market? 4) Do the realtors even care about whether a property is qualified to sell before listing it or is it all about the money? 5) Are we creating the most optimal regression for modeling properties? 6) Do we follow previous linear regression housing model approaches for predictor variables, or should we come up with our own model and approaches from scratch? and 7) Is money the most important thing? If so how does that define the world? With these questions in mind, we started to clean, assess, and manipulate the data, so no extremes were used in the analysis and modeling processes.

Data

Following the download of the data we started to assess and figure out what each column represented and the importance of each column. The original data had 158957 rows and 49 columns. To do this we had to read the description of the columns on the Kaggle site and google what we might not have known since this is not our area of expertise. In the beginning, we ran into a slight problem with not knowing what the qualification column in the original dataset meant. To resolve this problem of ours, we tried to reach out to the original uploaders of this dataset, but the original uploaders have not gotten back to us yet. So, we researched ourselves what a qualified property might be and the things a person should do to sell their property. The research later becomes what the qualification response variable description.

Since the dataset had 49 columns, this report will describe only the variables used in the models and analysis. This is because it will take up too many pages otherwise. The model variables were as follows: 1) PRICE, price of most recent sale, 2) BATHRM, the number of full bathrooms, 3) HF_BATHRM, the number of half bathrooms (no bathtub or shower), 4) AC, whether the property has air conditioning, 5) ROOMS, the number of Rooms, 6) BEDRM, the number of bedrooms, 7) STORIES, the number of stories in the building or property, 8) QUALIFIED, whether a property is qualified to sell, 9) STYLE, the style of the property, 10) CNDTN, a verbal rating of the condition of the property, 11) KITCHENS, the number of kitchens, 12) FIREPLACES, the number of fireplaces, and 13) WARD, the ward and the ward number (District is divided into eight wards, each with approximately 75,000 residents). Although there were more variables, these variables were not used in our model and shall be described in a later report.

Unfortunately, we started to have a lot more issues even before the cleaning process began. One of the key issues we noticed right away and were coming across was that a lot of data was missing in most the columns. Each column besides the ID column had missing rows between 1 observation to over fifty percent. We also had two columns (complex number and living GBA) that had to be taken out since they had no data entries in any of the rows. We come to the decision not to add the data which could have been found on realtor sites that had similar qualities. We did not fill in the blanks for the missing data because it would increase bias dramatically and who knows whether the data we could have added it would be the correct data. The executive decision we came down to was taking out all the blanks from our dataset and working with only the data that was downloaded. Although this method was working great, we ran into some more problems. Some of these issues we were facing were data being entered either incorrectly, data having errors, and values incorrectly labeled. One example air conditioning column (AC). The AC column was supposed to be Y, yes, and N, no, but it had a third value of 0 which had to be later changed to N. Once the data cleaning was completed, we decided the bounds we wished to use for our analysis. The bounds we came with were rather long but necessary in lowering bias. The bounds we came up with were as follows: Price between $10,000 and $1,000,000, Fireplaces less then 8, kitchens less than or equal to 10, rooms less than 26, bedrooms less than 20, stories of your building less than 100, bathrooms greater than 0, and half bathrooms great than 0. With these bounds in mind, we started to use visualizations to see what kinds of graphs we could create and data we are working with.

Although this worked great for our visualizations, the dataset we used for a model had only the essential variables we wish to use. So, in total there were three datasets for this project which were called in order, the original dataset, the visualization dataset, and the model dataset. The original dataset as stated above had 158957 rows and 49 columns. The visualization dataset which used the specified bounds had 17522 rows and 46 columns. The model dataset which used the specified bounds had 33671 rows and 32 columns. Now that the data cleaning was completed we can move onto the analysis section where we will report on how we compiled our model and the model process. It should be noted that we were not happy and tried to figure out ways to get more than 1/5 of the original dataset to no avail.

```
library(tidyverse)
library(readxl)
library(broom)
library(ggplot2)
library(modelr)
library(purrr)
library(boot)
library(scales)

## Data

DC_Properties <- read_excel("~/Documents/STAT 616 Generalizd Linear
Models/GLM Project/Data/DC_Properties.xlsx", na ="")
summary(DC_Properties)

##        ID             BATHRM          HF_BATHRM           HEAT          
##  Min.   :     0   Min.   : 0.000   Min.   : 0.0000   Length:158957     
##  1st Qu.: 39739   1st Qu.: 1.000   1st Qu.: 0.0000   Class :character  
##  Median : 79478   Median : 2.000   Median : 0.0000   Mode  :character  
##  Mean   : 79478   Mean   : 1.811   Mean   : 0.4582                     
##  3rd Qu.:119217   3rd Qu.: 2.000   3rd Qu.: 1.0000                     
##  Max.   :158956   Max.   :14.000   Max.   :11.0000                     
##                                                                        
##       AC              NUM_UNITS         ROOMS            BEDRM       
##  Length:158957      Min.   :0.0     Min.   : 0.000   Min.   : 0.000  
##  Class :character   1st Qu.:1.0     1st Qu.: 4.000   1st Qu.: 2.000  
##  Mode  :character   Median :1.0     Median : 6.000   Median : 3.000  
##                     Mean   :1.2     Mean   : 6.188   Mean   : 2.733  
##                     3rd Qu.:1.0     3rd Qu.: 7.000   3rd Qu.: 3.000  
##                     Max.   :6.0     Max.   :48.000   Max.   :24.000  
##                     NA's   :52261                                    
##       AYB          YR_RMDL          EYB          STORIES      
##  Min.   :1754   Min.   :  20    Min.   :1800   Min.   :  0.00  
##  1st Qu.:1918   1st Qu.:1985    1st Qu.:1954   1st Qu.:  2.00  
##  Median :1937   Median :2004    Median :1963   Median :  2.00  
##  Mean   :1942   Mean   :1998    Mean   :1964   Mean   :  2.09  
##  3rd Qu.:1960   3rd Qu.:2010    3rd Qu.:1975   3rd Qu.:  2.00  
##  Max.   :2019   Max.   :2019    Max.   :2018   Max.   :826.00  
##  NA's   :271    NA's   :78029                  NA's   :52305   
##     SALEDATE                         PRICE             QUALIFIED        
##  Min.   :1947-05-14 00:00:00   Min.   :        1   Length:158957     
##  1st Qu.:2005-04-14 00:00:00   1st Qu.:   240000   Class :character  
##  Median :2011-05-13 00:00:00   Median :   399999   Mode  :character  
##  Mean   :2009-12-06 10:43:20   Mean   :   931352                     
##  3rd Qu.:2015-08-26 00:00:00   3rd Qu.:   652000                     
##  Max.   :2018-07-12 00:00:00   Max.   :137427545                     
##  NA's   :26770                 NA's   :60741                         
```

```
##     SALE_NUM          GBA           BLDG_NUM        STYLE          
##  Min.   : 1.00   Min.   :    0   Min.   :1.000   Length:158957     
##  1st Qu.: 1.00   1st Qu.: 1190   1st Qu.:1.000   Class :character  
##  Median : 1.00   Median : 1480   Median :1.000   Mode  :character  
##  Mean   : 1.68   Mean   : 1715   Mean   :1.001                     
##  3rd Qu.: 2.00   3rd Qu.: 1966   3rd Qu.:1.000                     
##  Max.   :15.00   Max.   :45384   Max.   :5.000                     
##                  NA's   :52261                                     
##     STRUCT             GRADE               CNDTN          
##  Length:158957      Length:158957      Length:158957      
##  Class :character   Class :character   Class :character   
##  Mode  :character   Mode  :character   Mode  :character   
##                                                           
##                                                           
##                                                           
##                                                           
##    EXTWALL              ROOF              INTWALL             KITCHENS    
##  Length:158957      Length:158957      Length:158957      Min.   : 0.00  
##  Class :character   Class :character   Class :character   1st Qu.: 1.00  
##  Mode  :character   Mode  :character   Mode  :character   Median : 1.00  
##                                                           Mean   : 1.22  
##                                                           3rd Qu.: 1.00  
##                                                           Max.   :44.00  
##                                                           NA's   :52262  
##    FIREPLACES          USECODE          LANDAREA     
##  Min.   :     0.00   Min.   : 11.00   Min.   :     0  
##  1st Qu.:     0.00   1st Qu.: 11.00   1st Qu.:   697  
##  Median :     0.00   Median : 13.00   Median :  1649  
##  Mean   :     2.37   Mean   : 14.25   Mean   :  2473  
##  3rd Qu.:     1.00   3rd Qu.: 17.00   3rd Qu.:  3000  
##  Max.   :293920.00   Max.   :117.00   Max.   :942632  
##                                                      
##  GIS_LAST_MOD_DTTM                SOURCE          CMPLX_NUM     
##  Min.   :2018-07-22 18:01:00   Length:158957      Mode:logical  
##  1st Qu.:2018-07-22 18:01:00   Class :character   TRUE:52261    
##  Median :2018-07-22 18:01:00   Mode  :character   NA's:106696   
##  Mean   :2018-07-22 18:01:00                                    
##  3rd Qu.:2018-07-22 18:01:00                                    
##  Max.   :2018-07-22 18:01:00                                    
##                                                                 
##  LIVING_GBA      FULLADDRESS            CITY              STATE          
##  Mode :logical   Length:158957      Length:158957      Length:158957     
##  FALSE:1         Class :character   Class :character   Class :character  
##  TRUE :52260     Mode  :character   Mode  :character   Mode  :character  
##  NA's :106696                                                            
##                                                                          
##                                                                          
##                                                                          
##     ZIPCODE      NATIONALGRID          LATITUDE       LONGITUDE     
##  Min.   :20001   Length:158957      Min.   :38.82   Min.   :-77.11  
```

```
##  1st Qu.:20007   Class :character   1st Qu.:38.90   1st Qu.:-77.04
##  Median :20011   Mode  :character   Median :38.92   Median :-77.02
##  Mean   :20013                      Mean   :38.91   Mean   :-77.02
##  3rd Qu.:20018                      3rd Qu.:38.94   3rd Qu.:-76.99
##  Max.   :20392                      Max.   :39.00   Max.   :-76.91
##  NA's   :1                          NA's   :1       NA's   :1
##  ASSESSMENT_NBHD    ASSESSMENT_SUBNBHD  CENSUS_TRACT   CENSUS_BLOCK
##  Length:158957      Length:158957      Min.   :  100   Length:158957
##  Class :character   Class :character   1st Qu.: 2102   Class :character
##  Mode  :character   Mode  :character   Median : 5201   Mode  :character
##                                        Mean   : 5348
##                                        3rd Qu.: 8302
##                                        Max.   :11100
##                                        NA's   :1
##      WARD               SQUARE           X                Y
##  Length:158957      Min.   :   4   Min.   :-77.11   Min.   :38.82
##  Class :character   1st Qu.:1053   1st Qu.:-77.04   1st Qu.:38.90
##  Mode  :character   Median :2591   Median :-77.02   Median :38.92
##                     Mean   :2641   Mean   :-77.02   Mean   :38.91
##                     3rd Qu.:3924   3rd Qu.:-76.99   3rd Qu.:38.94
##                     Max.   :6277   Max.   :-76.91   Max.   :38.99
##                     NA's   :237    NA's   :237      NA's   :237
##    QUADRANT
##  Length:158957
##  Class :character
##  Mode  :character
##
##
##
##
```

```
dim(DC_Properties)
```

```
## [1] 158957     49
```

```
## Minimum you have to take away since it was not need in the analysis or
Visualisations

## Visualisation Data

DC_Properties_Visualisations <- DC_Properties %>%
  select(-CMPLX_NUM, -LIVING_GBA, -SALE_NUM, -GIS_LAST_MOD_DTTM) %>%
  filter(PRICE > 10000 & PRICE < 10000000,
         HEAT != "No Data",
         CNDTN != "No Data",
         CNDTN != "Default",
         STRUCT != "Default",
         GRADE != " No Data",
         STYLE != "Default",
         KITCHENS <= 10,
```

```
         ROOMS < 26,
         BEDRM < 20,
         STORIES <100,
         BATHRM > 0,
         HF_BATHRM > 0) %>%
  mutate(QUALIFIED_2 = QUALIFIED) %>%
  mutate(QUALIFIED_2 = ifelse(QUALIFIED == "Q", 1, 0))

dcproperty <- na.omit(DC_Properties_Visualisations)

dcproperty$AC[dcproperty$AC == "0"] <- "N"
dcproperty$GRADE[dcproperty$GRADE == "Exceptional-A"] <- "Exceptional"
dcproperty$GRADE[dcproperty$GRADE == "Exceptional-B"] <- "Exceptional"
dcproperty$GRADE[dcproperty$GRADE == "Exceptional-C"] <- "Exceptional"
dcproperty$GRADE[dcproperty$GRADE == "Exceptional-D"] <- "Exceptional"

dim(dcproperty)

## [1] 17522    46

## Final Cleaned Dataset

DC_Properties_Final <- DC_Properties %>%
  select(-NUM_UNITS, -YR_RMDL, -SALEDATE, -GBA, -STRUCT, -EXTWALL, -ROOF, -
INTWALL, -CMPLX_NUM, -LIVING_GBA, -FULLADDRESS, -CITY, -STATE, -NATIONALGRID,
-ASSESSMENT_SUBNBHD, -CENSUS_BLOCK, -SALE_NUM, -GIS_LAST_MOD_DTTM) %>%
  filter(CNDTN != "No Data",
         CNDTN != "Default",
         GRADE != " No Data",
         STYLE != "Default",
         PRICE > 10000 & PRICE < 10000000,
         FIREPLACES < 8,
         KITCHENS <= 10,
         ROOMS < 26,
         BEDRM < 20,
         STORIES <100,
         BATHRM > 0,
         HF_BATHRM > 0) %>%
  mutate(QUALIFIED_2 = QUALIFIED) %>%
  mutate(QUALIFIED_2 = ifelse(QUALIFIED == "Q", 1, 0))

DC_Final <- na.omit(DC_Properties_Final)

DC_Final$AC[DC_Final$AC == "0"] <- "N"

dim(DC_Final)

## [1] 33671    32

summary(DC_Final)
```

```
##        ID              BATHRM          HF_BATHRM          HEAT          
##  Min.   :     2   Min.   : 1.000   Min.   : 1.000   Length:33671      
##  1st Qu.: 22708   1st Qu.: 1.000   1st Qu.: 1.000   Class :character  
##  Median : 43851   Median : 2.000   Median : 1.000   Mode  :character  
##  Mean   : 47835   Mean   : 2.288   Mean   : 1.109                     
##  3rd Qu.: 72416   3rd Qu.: 3.000   3rd Qu.: 1.000                     
##  Max.   :106668   Max.   :11.000   Max.   :11.000                     
##       AC                ROOMS            BEDRM             AYB      
##  Length:33671       Min.   : 0.000   Min.   : 0.000   Min.   :1765  
##  Class :character   1st Qu.: 6.000   1st Qu.: 3.000   1st Qu.:1912  
##  Mode  :character   Median : 7.000   Median : 3.000   Median :1929  
##                     Mean   : 7.551   Mean   : 3.519   Mean   :1938  
##                     3rd Qu.: 8.000   3rd Qu.: 4.000   3rd Qu.:1952  
##                     Max.   :24.000   Max.   :12.000   Max.   :2018  
##       EYB          STORIES          PRICE          QUALIFIED        
##  Min.   :1932   Min.   : 0.00   Min.   :  10273   Length:33671      
##  1st Qu.:1964   1st Qu.: 2.00   1st Qu.: 310000   Class :character  
##  Median :1969   Median : 2.00   Median : 550000   Mode  :character  
##  Mean   :1975   Mean   : 2.16   Mean   : 685729                     
##  3rd Qu.:1984   3rd Qu.: 2.00   3rd Qu.: 855000                     
##  Max.   :2018   Max.   :25.00   Max.   :9100000                     
##     BLDG_NUM    STYLE               GRADE               CNDTN          
##  Min.   :1   Length:33671       Length:33671       Length:33671      
##  1st Qu.:1   Class :character   Class :character   Class :character  
##  Median :1   Mode  :character   Mode  :character   Mode  :character  
##  Mean   :1                                                           
##  3rd Qu.:1                                                           
##  Max.   :2                                                           
##     KITCHENS       FIREPLACES        USECODE         LANDAREA     
##  Min.   :0.000   Min.   :0.0000   Min.   :11.00   Min.   :     0  
##  1st Qu.:1.000   1st Qu.:0.0000   1st Qu.:11.00   1st Qu.:  1520  
##  Median :1.000   Median :1.0000   Median :12.00   Median :  2264  
##  Mean   :1.155   Mean   :0.8053   Mean   :12.86   Mean   :  3389  
##  3rd Qu.:1.000   3rd Qu.:1.0000   3rd Qu.:12.00   3rd Qu.:  4362  
##  Max.   :4.000   Max.   :7.0000   Max.   :24.00   Max.   :102340  
##     SOURCE             ZIPCODE         LATITUDE       LONGITUDE     
##  Length:33671       Min.   :20001   Min.   :38.82   Min.   :-77.11  
##  Class :character   1st Qu.:20005   1st Qu.:38.90   1st Qu.:-77.05  
##  Mode  :character   Median :20011   Median :38.92   Median :-77.01  
##                     Mean   :20012   Mean   :38.92   Mean   :-77.02  
##                     3rd Qu.:20017   3rd Qu.:38.94   3rd Qu.:-76.99  
##                     Max.   :20052   Max.   :38.99   Max.   :-76.91  
##  ASSESSMENT_NBHD     CENSUS_TRACT        WARD              SQUARE    
##  Length:33671       Min.   :  100   Length:33671       Min.   :  14  
##  Class :character   1st Qu.: 1600   Class :character   1st Qu.:1306  
##  Mode  :character   Median : 4901   Mode  :character   Median :2697  
##                     Mean   : 5184                      Mean   :2773  
##                     3rd Qu.: 8402                      3rd Qu.:3920  
##                     Max.   :11100                      Max.   :6250  
##        X                 Y            QUADRANT          QUALIFIED_2
```

```
##   Min.   :-77.11   Min.   :38.82   Length:33671       Min.   :0.0000
##   1st Qu.:-77.05   1st Qu.:38.90   Class :character   1st Qu.:1.0000
##   Median :-77.01   Median :38.92   Mode  :character   Median :1.0000
##   Mean   :-77.02   Mean   :38.92                      Mean   :0.8368
##   3rd Qu.:-76.99   3rd Qu.:38.94                      3rd Qu.:1.0000
##   Max.   :-76.91   Max.   :38.99                      Max.   :1.0000

## random case number

cases <- c(1:2773, 4624:6649, 8000:12724, 15874:22079, 26216:31903)

Final_T <- DC_Final[cases,]
Final_V <- DC_Final[-cases,]
```

I choose the observations randomly for the training and validation set data to decrease bias.

Section 1: Analysis

To complete our analysis, we tried to use our analysis processing skills. Before we conducted any analysis, we decided to break the model dataset into two parts called the training dataset and the validation dataset. The training dataset included approximately sixty percent of the models' datasets. The validation dataset included approximately forty percent of the models' dataset. The analysis processes in order were composed of model building, diagnostic processes, stepwise AIC, stepwise BIC, hypothesis testing which included Likelihood ratio test and goodness of fits tests, rebuilding the model, checking for multicollinearity issues, fixing multicollinearity issues, creating interaction terms, and creating the ROC Curve, and more diagnostic processes.

The first step in our analysis process was creating a lot of diagnostic plots and noting our observations of these plots. From these plots, we could come to a few conclusions. These conclusions were as follows: 1) our data is not normally distributed so we will have to use transformations, 2) we have multicollinearity, so we will have to create variance inflation factor graphs and numbers to fix this issue (refer to VIF section) we will need to apply transformations to our model. With these things in mind, we moved on to the model building process.

The second step in our analysis process was to build some models and choose which predictor variables we wish to use for our model. The first model we built was very simple but it had our basic requirements on what we believed was necessary at that time to model qualification. Our first model was using the qualification as the response variable and price as our only predictor variable. In the beginning, we believed that price is the most important variable and it should not be eliminated from our model because most housing model price as the response so we should have at a minimum price as a predictor variable. The basic model equation was $\text{Log}(\pi/1-\pi) = 0.6811 + 0.000001726*\text{Price}$ and the AIC was equal to 17,988. So, for every additional pricing dollar, the odds a property being qualified to sell increases by a factor of 1.000002. When we tried to use the likelihood ratio test with a null hypothesis that $\beta 1$ is equal to 0. We did this to determine if the price was supposed to be in the model going forward or not. Although if the result was not to reject the null hypothesis, we might have just noted it and continued with the analysis keeping price as a predictor variable anyway. Thankfully though the results stated to reject the null hypothesis and keep the price as a predictor variable.

Moving on we created a new model with all the predictor variables we felt were necessary. This new model included the price, the number of bathrooms, the number of half bathrooms, having air conditioning dummy variable, number of stories the property has or the building the property is in, the type of style of the property (14 categorical variables), the condition of the property (4 categorical variables), the number of kitchens, the number of fireplaces, and the ward number where the property was located (ward 1 – 5). It should be noted that although there were originally 8 wards, after the data cleaning and creating the training and validation datasets we ended up with only 5 wards. This binary logistic regression model had an AIC equal to 17,410. From this model, we could tell we were on the right track in our model building process since the AIC decreased by 578 from our first basic model, but the further analysis still needed to be completed since this AIC was still very high.

The next step was figuring out if we could reduce the 32-betas in our model. To reduce the model, we used stepwise AIC and BIC. The stepwise AIC and BIC results showed us that we should keep the following predictor variables: price, bathrooms, AC=Yes, bedrooms, 14 Style dummy variables, 5 condition dummy variables, kitchens, and the 4 ward dummy variables. My partner and I made sure to double check these results with the likelihood ratio test by finding G-squared and p-value to make sure these variables that were being taken out were correct, and there were no other variables we missed taking out before moving on. The reason that we were so meticulous with getting rid of variables is that we wished to eliminate any cases of hidden multicollinearity, and our belief that having too many variables would disrupt the model. It should be noted that before moving on we decided to take out the style dummy variables from our model, even though it slightly increased the AIC. This is because the style was creating too many betas, majority of style dummy variables were statistically insignificant, and they were also making the rest of our predictor variables be statistically insignificant.

Once the final single predictor variables were selected we decided to create all possible 2-way interactions terms. We could have created interaction terms on our own from what we felt was the most important, but we did not wish to miss any type of interaction terms that could have been beneficial to creating a better binary logistic regression model for qualification. From this 2-way interaction model, we used stepwise AIC, BIC, and likelihood ratio tests to determine which variable and interactions terms because these variables would become the founding base for our final model with transformations. The variable and the interaction terms that were created from all this analysis were as follows: price, AC=Yes, bedrooms, 5 condition dummy variables, the 4 ward categorical variables, price times AC=Yes, price times rooms, and price times the 4 ward categorical variables.

Following the creation of our interaction terms model, we looked back at the diagnostic plots and created some more diagnostic plots to see if we if the underlying concerns we had were still around. It seemed that our data was still not normally distributed so we would have to create some transformations to our interaction terms model. The good thing was that multicollinearity we noticed and were concerned about was no valid. However, we did notice that there were some outliers in training set data, so we checked to see whether they were significantly impacting our model and if they were significantly impacting our model we took them out. Although because the training set dataset were we working was large, we might have missed taking out outliers.

The next step was to add some transformations to our model to make the data and our model more normally distributed. We found that although we could add transformations to the single predictor variables, when we applied these transformations to the interaction terms, the AIC and BIC numbers were increased as a result and some variables were becoming statistically insignificant. So, with the little option left, my partner and I left the single predictor variable transformations in the model and the non-transformation interaction terms. We then used stepwise AIC, stepwise BIC, and likelihood ratio tests on the new predictors to determine if the model needed further changes. The final model we created from stepwise AIC selection results was as follows:

Log(π/1-π) = - 3.158 - 0.000004374*Price + 0.007901*√Price - 0.2907*(AC=Yes) + 0.1821*Rooms + 2.221*(Rooms^0.2) - 0.04816*√BEDRM + 0.1069*(CNDTN=Excellent) - 1.196*(CNDTN=Fair) + 0.2849*(CNDTN=Good) – 1.373*(CNDTN=Poor) + 0.6739(CNDTN=Very Good) – 0.4306*(Ward=2) – 0.2858*(Ward=3) – 0.0515*(Ward=4) + 0.2901(Ward=5) + 0.000001085PRICE*(AC=Yes) + 0.00000002.698*(PRICE*ROOMS) + 0.0000003.897(Price*Ward=2) + 0.0000005486*(Price*Ward=3) + 0.000001052*(Price*Ward=4) – 0.0000003.313*(Price*Ward=5)

This model had an AIC of 16748. The interpretation of this model is as follows:

For every additional pricing dollar, the odds a property being qualified to sell decrease by a factor of 0.00000437402. For every additional square root of a pricing dollar, the odds a property is qualified to sell increase by a factor of 1.01. If a property has air conditioning, then the odds a property is qualified to sell increase by a factor of 0.34. For every additional room, the odds a property being qualified to sell decreases by a factor of 0.2. For every additional room raised to the one-fifth power, the odds a property being qualified to sell increases by a factor of 9.22. For every additional square root of a bedroom, the odds a property being qualified to sell decrease by a factor of 0.62. If a property has an excellent condition rating, then the odds that property being qualified to sell decreases by a factor of 2.31. If a property has a good condition rating, then the odds that property being qualified to sell increases by a factor of 1.33. If a property has a poor condition rating, then the odds that property being qualified to sell decreases by a factor of 2.95. If a property has a very good condition rating, then the odds that property being qualified to sell increase by a factor of 1.96. If a property is in Ward 2, then the odds that property being qualified to sell will decrease by a factor 0.54. If a property is in Ward 3, then the odds that property being qualified to sell will decrease by a factor 0.33. If a property is in Ward 4, then the odds that property being qualified to sell will decrease by a factor of 0.05. For every additional dollar added to the price and if the property has air conditioning, the odds a property being qualified to sell will increase by a factor of 1. For every additional dollar added to the price and for every additional room, the odds a property being qualified to sell will increase by a factor of 1. For every additional dollar added to the price and if the property is in ward 2, the odds a property being qualified to sell will increase by a factor of 1. For every additional dollar added to the price and if the property is in ward 3, the odds a property being qualified to sell will increase by a factor of 1. For every additional dollar added to the price and if the property is in ward 4, the odds a property being qualified to sell will increase by a factor of 1. For every additional dollar added to the price and if the property is in ward 5, the odds a property being qualified to sell will decrease by a factor of 0.0000003313001.

The reason we choose the stepwise AIC model was that we felt that the stepwise selection for BIC was penalizing our variables too much. With the creation of the final model, there was only the goodness of fit test left to run. Kingsley conducted this test and told me that the results were that the model fits the data. I do not believe this is accurate with such a high AIC but one will have to trust Kingsley's judgment in this case. We also used the ROC Curve to determine well the training set model works compared with the validation set model. We can conclude that our model works well because the two areas were greater than seventy percent, even though we would have liked the areas to be above .85, and the training set and validation set lines were very similar, almost the same.

All of the analysis work is below

Model Selection

```
## Basic that we originally thought to work with

basic_model <- glm(as.factor(QUALIFIED_2) ~ PRICE, data = Final_T, family =
binomial(link=logit))
summary(basic_model)

##
## Call:
## glm(formula = as.factor(QUALIFIED_2) ~ PRICE, family = binomial(link =
logit),
##     data = Final_T)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.5409   0.3234   0.5264   0.6699   0.8891
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 6.811e-01  3.445e-02   19.77   <2e-16 ***
## PRICE       1.726e-06  6.139e-08   28.11   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 19081  on 21417  degrees of freedom
## Residual deviance: 17984  on 21416  degrees of freedom
## AIC: 17988
##
## Number of Fisher Scoring iterations: 5

## Model with all the variables we wished to use

model3 <- glm(as.factor(QUALIFIED_2) ~ PRICE + BATHRM + HF_BATHRM +
as.factor(AC) + ROOMS + BEDRM + STORIES + as.factor(STYLE) + as.factor(CNDTN)
+ KITCHENS + FIREPLACES + as.factor(WARD), data = Final_T, family =
binomial(link=logit))
summary(model3)
```

```
## 
## Call:
## glm(formula = as.factor(QUALIFIED_2) ~ PRICE + BATHRM + HF_BATHRM + 
##     as.factor(AC) + ROOMS + BEDRM + STORIES + as.factor(STYLE) + 
##     as.factor(CNDTN) + KITCHENS + FIREPLACES + as.factor(WARD), 
##     family = binomial(link = logit), data = Final_T)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -5.3076   0.3167   0.4660   0.6330   1.6162  
## 
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)    
## (Intercept)                     7.072e-01  2.160e-01   3.274 0.001060 ** 
## PRICE                           1.742e-06  8.155e-08  21.355  < 2e-16 ***
## BATHRM                         -5.859e-02  3.017e-02  -1.942 0.052110 .  
## HF_BATHRM                      -1.165e-01  5.809e-02  -2.006 0.044853 *  
## as.factor(AC)Y                  2.874e-01  5.230e-02   5.495 3.90e-08 ***
## ROOMS                          -6.969e-02  1.461e-02  -4.770 1.84e-06 ***
## BEDRM                          -1.014e-01  2.868e-02  -3.537 0.000405 ***
## STORIES                         2.154e-01  1.416e-01   1.522 0.128058    
## as.factor(STYLE)1.5 Story Fin  -2.082e-01  1.885e-01  -1.105 0.269374    
## as.factor(STYLE)1.5 Story Unfin  8.060e-01  1.111e+00   0.726 0.467980    
## as.factor(STYLE)2 Story         1.539e-01  1.769e-01   0.870 0.384117    
## as.factor(STYLE)2.5 Story Fin   2.275e-02  2.404e-01   0.095 0.924624    
## as.factor(STYLE)2.5 Story Unfin  2.129e-01  2.979e-01   0.715 0.474735    
## as.factor(STYLE)3 Story        -2.339e-01  3.021e-01  -0.774 0.438760    
## as.factor(STYLE)3.5 Story Fin  -1.027e+00  5.477e-01  -1.876 0.060714 .  
## as.factor(STYLE)3.5 Story Unfin  8.616e+00  1.970e+02   0.044 0.965108    
## as.factor(STYLE)4 Story        -1.213e+00  5.122e-01  -2.368 0.017875 *  
## as.factor(STYLE)4.5 Story Fin  -1.190e+01  1.970e+02  -0.060 0.951818    
## as.factor(STYLE)4.5 Story Unfin  9.559e+00  1.970e+02   0.049 0.961293    
## as.factor(STYLE)Bi-Level        9.938e+00  1.382e+02   0.072 0.942678    
## as.factor(STYLE)Split Foyer    -3.020e-01  2.814e-01  -1.073 0.283276    
## as.factor(STYLE)Split Level     2.130e-01  3.636e-01   0.586 0.557921    
## as.factor(CNDTN)Excellent       2.837e-01  1.507e-01   1.882 0.059772 .  
## as.factor(CNDTN)Fair           -1.123e+00  2.206e-01  -5.090 3.58e-07 ***
## as.factor(CNDTN)Good            4.755e-01  4.590e-02  10.361  < 2e-16 ***
## as.factor(CNDTN)Poor           -1.244e+00  6.570e-01  -1.893 0.058313 .  
## as.factor(CNDTN)Very Good       9.511e-01  8.346e-02  11.396  < 2e-16 ***
## KITCHENS                        5.396e-02  5.530e-02   0.976 0.329180    
## FIREPLACES                     -6.799e-02  2.640e-02  -2.575 0.010018 *  
## as.factor(WARD)Ward 2          -8.850e-02  7.177e-02  -1.233 0.217554    
## as.factor(WARD)Ward 3          -4.662e-03  6.038e-02  -0.077 0.938449    
## as.factor(WARD)Ward 4           1.535e-01  7.845e-02   1.957 0.050356 .  
## as.factor(WARD)Ward 5          -7.673e-02  6.973e-02  -1.100 0.271140    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 19081  on 21417  degrees of freedom
## Residual deviance: 17344  on 21385  degrees of freedom
## AIC: 17410
##
## Number of Fisher Scoring iterations: 10
```

AIC & BIC Analysis

```
step(model3,direction="both")

## Start:  AIC=17410.18
## as.factor(QUALIFIED_2) ~ PRICE + BATHRM + HF_BATHRM + as.factor(AC) +
##     ROOMS + BEDRM + STORIES + as.factor(STYLE) + as.factor(CNDTN) +
##     KITCHENS + FIREPLACES + as.factor(WARD)
##
##                      Df Deviance   AIC
## - KITCHENS            1    17345 17409
## <none>                     17344 17410
## - STORIES             1    17347 17411
## - BATHRM              1    17348 17412
## - HF_BATHRM           1    17348 17412
## - as.factor(WARD)     4    17357 17415
## - FIREPLACES          1    17351 17415
## - BEDRM               1    17357 17421
## - as.factor(STYLE) 14    17392 17430
## - ROOMS               1    17367 17431
## - as.factor(AC)       1    17374 17438
## - as.factor(CNDTN)    5    17560 17616
## - PRICE               1    17876 17940
##
## Step:  AIC=17409.14
## as.factor(QUALIFIED_2) ~ PRICE + BATHRM + HF_BATHRM + as.factor(AC) +
##     ROOMS + BEDRM + STORIES + as.factor(STYLE) + as.factor(CNDTN) +
##     FIREPLACES + as.factor(WARD)
##
##                      Df Deviance   AIC
## <none>                     17345 17409
## + KITCHENS            1    17344 17410
## - STORIES             1    17348 17410
## - BATHRM              1    17348 17410
## - HF_BATHRM           1    17349 17411
## - FIREPLACES          1    17352 17414
## - as.factor(WARD)     4    17358 17414
## - BEDRM               1    17357 17419
## - as.factor(STYLE) 14    17392 17428
## - ROOMS               1    17367 17429
## - as.factor(AC)       1    17375 17437
## - as.factor(CNDTN)    5    17560 17614
## - PRICE               1    17877 17939
```

```
##
## Call:  glm(formula = as.factor(QUALIFIED_2) ~ PRICE + BATHRM + HF_BATHRM +
##     as.factor(AC) + ROOMS + BEDRM + STORIES + as.factor(STYLE) +
##     as.factor(CNDTN) + FIREPLACES + as.factor(WARD), family =
binomial(link = logit),
##     data = Final_T)
##
## Coefficients:
##                    (Intercept)                             PRICE
##                      7.458e-01                         1.742e-06
##                         BATHRM                         HF_BATHRM
##                     -5.345e-02                        -1.151e-01
##                 as.factor(AC)Y                             ROOMS
##                      2.872e-01                        -6.815e-02
##                          BEDRM                           STORIES
##                     -9.880e-02                         2.160e-01
##   as.factor(STYLE)1.5 Story Fin  as.factor(STYLE)1.5 Story Unfin
##                     -2.089e-01                         8.103e-01
##         as.factor(STYLE)2 Story    as.factor(STYLE)2.5 Story Fin
##                      1.547e-01                         1.571e-02
## as.factor(STYLE)2.5 Story Unfin          as.factor(STYLE)3 Story
##                      2.076e-01                        -2.270e-01
##   as.factor(STYLE)3.5 Story Fin  as.factor(STYLE)3.5 Story Unfin
##                     -1.026e+00                         8.653e+00
##         as.factor(STYLE)4 Story    as.factor(STYLE)4.5 Story Fin
##                     -1.208e+00                        -1.195e+01
## as.factor(STYLE)4.5 Story Unfin         as.factor(STYLE)Bi-Level
##                      9.556e+00                         9.938e+00
##     as.factor(STYLE)Split Foyer      as.factor(STYLE)Split Level
##                     -3.031e-01                         2.082e-01
##       as.factor(CNDTN)Excellent             as.factor(CNDTN)Fair
##                      2.716e-01                        -1.121e+00
##            as.factor(CNDTN)Good             as.factor(CNDTN)Poor
##                      4.735e-01                        -1.247e+00
##       as.factor(CNDTN)Very Good                       FIREPLACES
##                      9.445e-01                        -6.877e-02
##            as.factor(WARD)Ward 2            as.factor(WARD)Ward 3
##                     -1.054e-01                        -1.236e-02
##            as.factor(WARD)Ward 4            as.factor(WARD)Ward 5
##                      1.426e-01                        -8.878e-02
##
## Degrees of Freedom: 21417 Total (i.e. Null);  21386 Residual
## Null Deviance:       19080
## Residual Deviance: 17350     AIC: 17410
```

```
model_aic <- glm(as.factor(QUALIFIED_2) ~ PRICE + BATHRM + as.factor(AC) +
ROOMS + BEDRM + as.factor(STYLE) + as.factor(CNDTN) + KITCHENS +
as.factor(WARD), family = binomial(link = logit), data = Final_T)
summary(model_aic)
```

```
## 
## Call:
## glm(formula = as.factor(QUALIFIED_2) ~ PRICE + BATHRM + as.factor(AC) + 
##     ROOMS + BEDRM + as.factor(STYLE) + as.factor(CNDTN) + KITCHENS + 
##     as.factor(WARD), family = binomial(link = logit), data = Final_T)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -5.2554   0.3166   0.4685   0.6347   1.6361  
## 
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)    
## (Intercept)                     8.051e-01  1.540e-01   5.227 1.72e-07 ***
## PRICE                           1.708e-06  8.066e-08  21.181  < 2e-16 ***
## BATHRM                         -5.932e-02  2.996e-02  -1.980 0.047714 *  
## as.factor(AC)Y                  2.829e-01  5.228e-02   5.410 6.29e-08 ***
## ROOMS                          -7.686e-02  1.441e-02  -5.333 9.68e-08 ***
## BEDRM                          -1.018e-01  2.865e-02  -3.555 0.000378 ***
## as.factor(STYLE)1.5 Story Fin  -1.056e-01  1.755e-01  -0.602 0.547360    
## as.factor(STYLE)1.5 Story Unfin  9.592e-01  1.105e+00   0.868 0.385220    
## as.factor(STYLE)2 Story         3.850e-01  1.087e-01   3.543 0.000396 ***
## as.factor(STYLE)2.5 Story Fin   3.198e-01  1.305e-01   2.450 0.014284 *  
## as.factor(STYLE)2.5 Story Unfin  4.825e-01  2.291e-01   2.106 0.035169 *  
## as.factor(STYLE)3 Story         1.861e-01  1.265e-01   1.471 0.141374    
## as.factor(STYLE)3.5 Story Fin  -5.179e-01  4.205e-01  -1.232 0.218127    
## as.factor(STYLE)3.5 Story Unfin  9.080e+00  1.970e+02   0.046 0.963231    
## as.factor(STYLE)4 Story        -6.375e-01  3.361e-01  -1.897 0.057854 .  
## as.factor(STYLE)4.5 Story Fin  -1.165e+01  1.970e+02  -0.059 0.952834    
## as.factor(STYLE)4.5 Story Unfin  1.002e+01  1.970e+02   0.051 0.959440    
## as.factor(STYLE)Bi-Level        9.972e+00  1.380e+02   0.072 0.942412    
## as.factor(STYLE)Split Foyer    -2.356e-01  2.787e-01  -0.845 0.397963    
## as.factor(STYLE)Split Level     3.288e-01  3.546e-01   0.927 0.353783    
## as.factor(CNDTN)Excellent       3.427e-01  1.493e-01   2.295 0.021758 *  
## as.factor(CNDTN)Fair           -1.125e+00  2.205e-01  -5.100 3.40e-07 ***
## as.factor(CNDTN)Good            4.830e-01  4.586e-02  10.531  < 2e-16 ***
## as.factor(CNDTN)Poor           -1.237e+00  6.563e-01  -1.885 0.059436 .  
## as.factor(CNDTN)Very Good       9.776e-01  8.293e-02  11.789  < 2e-16 ***
## KITCHENS                        5.588e-02  5.518e-02   1.013 0.311218    
## as.factor(WARD)Ward 2          -1.164e-01  7.136e-02  -1.631 0.102879    
## as.factor(WARD)Ward 3           2.827e-04  5.988e-02   0.005 0.996233    
## as.factor(WARD)Ward 4           1.621e-01  7.785e-02   2.082 0.037341 *  
## as.factor(WARD)Ward 5          -6.343e-02  6.891e-02  -0.921 0.357306    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 19081  on 21417  degrees of freedom
## Residual deviance: 17358  on 21388  degrees of freedom
## AIC: 17418
```

```
##
## Number of Fisher Scoring iterations: 10

# AIC: 29329

sampsize <- length(model3$fitted)
step(model3, direction="both", k=log(sampsize))

## Start:  AIC=17673.26
## as.factor(QUALIFIED_2) ~ PRICE + BATHRM + HF_BATHRM + as.factor(AC) +
##     ROOMS + BEDRM + STORIES + as.factor(STYLE) + as.factor(CNDTN) +
##     KITCHENS + FIREPLACES + as.factor(WARD)
##
##                    Df Deviance   AIC
## - as.factor(STYLE) 14    17392 17581
## - as.factor(WARD)   4    17357 17646
## - KITCHENS          1    17345 17664
## - STORIES           1    17347 17666
## - BATHRM            1    17348 17667
## - HF_BATHRM         1    17348 17667
## - FIREPLACES        1    17351 17670
## <none>                   17344 17673
## - BEDRM             1    17357 17676
## - ROOMS             1    17367 17686
## - as.factor(AC)     1    17374 17693
## - as.factor(CNDTN)  5    17560 17839
## - PRICE             1    17876 18195
##
## Step:  AIC=17581.25
## as.factor(QUALIFIED_2) ~ PRICE + BATHRM + HF_BATHRM + as.factor(AC) +
##     ROOMS + BEDRM + STORIES + as.factor(CNDTN) + KITCHENS + FIREPLACES +
##     as.factor(WARD)
##
##                    Df Deviance   AIC
## - as.factor(WARD)   4    17407 17556
## - KITCHENS          1    17392 17572
## - STORIES           1    17392 17572
## - HF_BATHRM         1    17396 17575
## - BATHRM            1    17397 17576
## <none>                   17392 17581
## - FIREPLACES        1    17402 17581
## - BEDRM             1    17406 17585
## - ROOMS             1    17417 17597
## - as.factor(AC)     1    17419 17599
## + as.factor(STYLE) 14    17344 17673
## - as.factor(CNDTN)  5    17629 17768
## - PRICE             1    17916 18095
##
## Step:  AIC=17556.16
## as.factor(QUALIFIED_2) ~ PRICE + BATHRM + HF_BATHRM + as.factor(AC) +
```

```
##     ROOMS + BEDRM + STORIES + as.factor(CNDTN) + KITCHENS + FIREPLACES
##
##                      Df Deviance   AIC
## - STORIES             1    17408 17547
## - KITCHENS            1    17408 17547
## - HF_BATHRM           1    17411 17550
## - BATHRM              1    17411 17551
## <none>                     17407 17556
## - FIREPLACES          1    17418 17557
## - BEDRM               1    17422 17561
## - as.factor(AC)       1    17431 17570
## - ROOMS               1    17435 17574
## + as.factor(WARD)     4    17392 17581
## + as.factor(STYLE) 14    17357 17646
## - as.factor(CNDTN)  5    17662 17762
## - PRICE               1    18066 18205
##
## Step:  AIC=17547.18
## as.factor(QUALIFIED_2) ~ PRICE + BATHRM + HF_BATHRM + as.factor(AC) +
##     ROOMS + BEDRM + as.factor(CNDTN) + KITCHENS + FIREPLACES
##
##                      Df Deviance   AIC
## - KITCHENS            1    17409 17539
## - HF_BATHRM           1    17412 17541
## - BATHRM              1    17412 17542
## <none>                     17408 17547
## - FIREPLACES          1    17418 17548
## - BEDRM               1    17422 17552
## + STORIES             1    17407 17556
## - as.factor(AC)       1    17432 17561
## - ROOMS               1    17435 17565
## + as.factor(WARD)     4    17392 17572
## + as.factor(STYLE) 14    17360 17639
## - as.factor(CNDTN)  5    17663 17753
## - PRICE               1    18073 18203
##
## Step:  AIC=17538.66
## as.factor(QUALIFIED_2) ~ PRICE + BATHRM + HF_BATHRM + as.factor(AC) +
##     ROOMS + BEDRM + as.factor(CNDTN) + FIREPLACES
##
##                      Df Deviance   AIC
## - BATHRM              1    17413 17532
## - HF_BATHRM           1    17413 17533
## <none>                     17409 17539
## - FIREPLACES          1    17420 17540
## - BEDRM               1    17423 17543
## + KITCHENS            1    17408 17547
## + STORIES             1    17408 17547
## - as.factor(AC)       1    17433 17552
## - ROOMS               1    17435 17555
```

```
## + as.factor(WARD)   4    17393 17562
## + as.factor(STYLE) 14    17362 17631
## - as.factor(CNDTN)  5    17663 17743
## - PRICE             1    18074 18194
##
## Step:  AIC=17532.43
## as.factor(QUALIFIED_2) ~ PRICE + HF_BATHRM + as.factor(AC) +
##     ROOMS + BEDRM + as.factor(CNDTN) + FIREPLACES
##
##                    Df Deviance   AIC
## - HF_BATHRM         1    17416 17526
## <none>                   17413 17532
## - FIREPLACES        1    17426 17535
## + BATHRM            1    17409 17539
## + STORIES           1    17412 17542
## + KITCHENS          1    17412 17542
## - as.factor(AC)     1    17434 17543
## - BEDRM             1    17435 17545
## - ROOMS             1    17445 17554
## + as.factor(WARD)   4    17397 17557
## + as.factor(STYLE) 14    17365 17624
## - as.factor(CNDTN)  5    17664 17734
## - PRICE             1    18104 18214
##
## Step:  AIC=17525.92
## as.factor(QUALIFIED_2) ~ PRICE + as.factor(AC) + ROOMS + BEDRM +
##     as.factor(CNDTN) + FIREPLACES
##
##                    Df Deviance   AIC
## <none>                   17416 17526
## - FIREPLACES        1    17430 17530
## + HF_BATHRM         1    17413 17532
## + BATHRM            1    17413 17533
## + STORIES           1    17415 17535
## + KITCHENS          1    17415 17535
## - as.factor(AC)     1    17437 17537
## - BEDRM             1    17439 17538
## + as.factor(WARD)   4    17400 17550
## - ROOMS             1    17451 17550
## + as.factor(STYLE) 14    17369 17618
## - as.factor(CNDTN)  5    17668 17728
## - PRICE             1    18108 18207

##
## Call:  glm(formula = as.factor(QUALIFIED_2) ~ PRICE + as.factor(AC) +
##     ROOMS + BEDRM + as.factor(CNDTN) + FIREPLACES, family = binomial(link
= logit),
##     data = Final_T)
##
## Coefficients:
```

```
##                 (Intercept)                          PRICE
##                   1.291e+00                      1.703e-06
##              as.factor(AC)Y                          ROOMS
##                   2.285e-01                     -8.270e-02
##                       BEDRM  as.factor(CNDTN)Excellent
##                  -1.232e-01                      2.326e-01
##        as.factor(CNDTN)Fair       as.factor(CNDTN)Good
##                  -1.141e+00                      4.857e-01
##        as.factor(CNDTN)Poor  as.factor(CNDTN)Very Good
##                  -1.260e+00                      9.858e-01
##                  FIREPLACES
##                  -9.347e-02
##
## Degrees of Freedom: 21417 Total (i.e. Null);  21407 Residual
## Null Deviance:       19080
## Residual Deviance: 17420     AIC: 17440
```

```
## Model Created from BIC Results is below

model_bic <- glm(as.factor(QUALIFIED_2) ~ PRICE + BATHRM + as.factor(AC) +
ROOMS + BEDRM + as.factor(STYLE) + as.factor(CNDTN) + KITCHENS +
as.factor(WARD), data = Final_T, family = binomial(link=logit))
summary(model_bic)
```

```
##
## Call:
## glm(formula = as.factor(QUALIFIED_2) ~ PRICE + BATHRM + as.factor(AC) +
##     ROOMS + BEDRM + as.factor(STYLE) + as.factor(CNDTN) + KITCHENS +
##     as.factor(WARD), family = binomial(link = logit), data = Final_T)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.2554   0.3166   0.4685   0.6347   1.6361
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   8.051e-01  1.540e-01   5.227 1.72e-07 ***
## PRICE                         1.708e-06  8.066e-08  21.181  < 2e-16 ***
## BATHRM                       -5.932e-02  2.996e-02  -1.980 0.047714 *
## as.factor(AC)Y                2.829e-01  5.228e-02   5.410 6.29e-08 ***
## ROOMS                        -7.686e-02  1.441e-02  -5.333 9.68e-08 ***
## BEDRM                        -1.018e-01  2.865e-02  -3.555 0.000378 ***
## as.factor(STYLE)1.5 Story Fin   -1.056e-01  1.755e-01  -0.602 0.547360
## as.factor(STYLE)1.5 Story Unfin  9.592e-01  1.105e+00   0.868 0.385220
## as.factor(STYLE)2 Story         3.850e-01  1.087e-01   3.543 0.000396 ***
## as.factor(STYLE)2.5 Story Fin    3.198e-01  1.305e-01   2.450 0.014284 *
## as.factor(STYLE)2.5 Story Unfin  4.825e-01  2.291e-01   2.106 0.035169 *
## as.factor(STYLE)3 Story         1.861e-01  1.265e-01   1.471 0.141374
## as.factor(STYLE)3.5 Story Fin   -5.179e-01  4.205e-01  -1.232 0.218127
## as.factor(STYLE)3.5 Story Unfin  9.080e+00  1.970e+02   0.046 0.963231
```

```
## as.factor(STYLE)4 Story          -6.375e-01  3.361e-01  -1.897 0.057854 .
## as.factor(STYLE)4.5 Story Fin    -1.165e+01  1.970e+02  -0.059 0.952834
## as.factor(STYLE)4.5 Story Unfin   1.002e+01  1.970e+02   0.051 0.959440
## as.factor(STYLE)Bi-Level          9.972e+00  1.380e+02   0.072 0.942412
## as.factor(STYLE)Split Foyer      -2.356e-01  2.787e-01  -0.845 0.397963
## as.factor(STYLE)Split Level       3.288e-01  3.546e-01   0.927 0.353783
## as.factor(CNDTN)Excellent         3.427e-01  1.493e-01   2.295 0.021758 *
## as.factor(CNDTN)Fair             -1.125e+00  2.205e-01  -5.100 3.40e-07 ***
## as.factor(CNDTN)Good              4.830e-01  4.586e-02  10.531  < 2e-16 ***
## as.factor(CNDTN)Poor             -1.237e+00  6.563e-01  -1.885 0.059436 .
## as.factor(CNDTN)Very Good         9.776e-01  8.293e-02  11.789  < 2e-16 ***
## KITCHENS                          5.588e-02  5.518e-02   1.013 0.311218
## as.factor(WARD)Ward 2            -1.164e-01  7.136e-02  -1.631 0.102879
## as.factor(WARD)Ward 3             2.827e-04  5.988e-02   0.005 0.996233
## as.factor(WARD)Ward 4             1.621e-01  7.785e-02   2.082 0.037341 *
## as.factor(WARD)Ward 5            -6.343e-02  6.891e-02  -0.921 0.357306
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 19081  on 21417  degrees of freedom
## Residual deviance: 17358  on 21388  degrees of freedom
## AIC: 17418
##
## Number of Fisher Scoring iterations: 10
```

```
# AIC: 29329, BIC = 32182

## tested further to see what could be eliminated using AIC and BIC
## got rid of STYLE - messing with model approaches, too many variables

## from AIC & BIC results

model_inter <- glm(QUALIFIED_2 ~ PRICE + as.factor(AC) +ROOMS + BEDRM +
as.factor(CNDTN) + as.factor(WARD) + PRICE*as.factor(AC) + PRICE*ROOMS +
PRICE*BEDRM + PRICE*as.factor(CNDTN) + PRICE*as.factor(WARD) +
as.factor(CNDTN)*as.factor(WARD), family = binomial(link = logit), data =
Final_T)

## AIC
step(model_inter, direction = "both")
```

```
## Start:  AIC=16959.22
## QUALIFIED_2 ~ PRICE + as.factor(AC) + ROOMS + BEDRM + as.factor(CNDTN) +
##     as.factor(WARD) + PRICE * as.factor(AC) + PRICE * ROOMS +
##     PRICE * BEDRM + PRICE * as.factor(CNDTN) + PRICE * as.factor(WARD) +
##     as.factor(CNDTN) * as.factor(WARD)
##
##                                     Df Deviance   AIC
```

```
## <none>                                        16869 16959
## - PRICE:as.factor(AC)                   1    16875 16963
## - PRICE:BEDRM                           1    16876 16964
## - PRICE:as.factor(CNDTN)                5    16896 16976
## - PRICE:ROOMS                           1    16897 16985
## - as.factor(CNDTN):as.factor(WARD) 19    16961 17013
## - PRICE:as.factor(WARD)                 4    17005 17087

##
## Call:  glm(formula = QUALIFIED_2 ~ PRICE + as.factor(AC) + ROOMS + BEDRM +
##     as.factor(CNDTN) + as.factor(WARD) + PRICE * as.factor(AC) +
##     PRICE * ROOMS + PRICE * BEDRM + PRICE * as.factor(CNDTN) +
##     PRICE * as.factor(WARD) + as.factor(CNDTN) * as.factor(WARD),
##     family = binomial(link = logit), data = Final_T)
##
## Coefficients:
##                                    (Intercept)
##                                      7.943e-01
##                                          PRICE
##                                      2.349e-06
##                                as.factor(AC)Y
##                                      2.614e-02
##                                          ROOMS
##                                     -4.967e-03
##                                          BEDRM
##                                     -8.343e-02
##                     as.factor(CNDTN)Excellent
##                                      5.019e-01
##                          as.factor(CNDTN)Fair
##                                     -2.092e+00
##                          as.factor(CNDTN)Good
##                                      1.676e-01
##                          as.factor(CNDTN)Poor
##                                     -2.655e+00
##                     as.factor(CNDTN)Very Good
##                                      4.192e-01
##                         as.factor(WARD)Ward 2
##                                      5.509e-01
##                         as.factor(WARD)Ward 3
##                                     -5.317e-01
##                         as.factor(WARD)Ward 4
##                                     -7.332e-01
##                         as.factor(WARD)Ward 5
##                                     -4.412e-01
##                          PRICE:as.factor(AC)Y
##                                      3.829e-07
##                                    PRICE:ROOMS
##                                     -1.097e-07
##                                    PRICE:BEDRM
##                                     -1.223e-07
```

```
##                   PRICE:as.factor(CNDTN)Excellent
##                                         1.457e-07
##                        PRICE:as.factor(CNDTN)Fair
##                                         2.594e-06
##                        PRICE:as.factor(CNDTN)Good
##                                         7.358e-07
##                        PRICE:as.factor(CNDTN)Poor
##                                         1.119e-05
##                   PRICE:as.factor(CNDTN)Very Good
##                                         7.872e-07
##                       PRICE:as.factor(WARD)Ward 2
##                                        -4.382e-07
##                       PRICE:as.factor(WARD)Ward 3
##                                         1.307e-06
##                       PRICE:as.factor(WARD)Ward 4
##                                         3.040e-06
##                       PRICE:as.factor(WARD)Ward 5
##                                         2.281e-06
## as.factor(CNDTN)Excellent:as.factor(WARD)Ward 2
##                                        -1.419e-01
##      as.factor(CNDTN)Fair:as.factor(WARD)Ward 2
##                                        -2.736e-01
##      as.factor(CNDTN)Good:as.factor(WARD)Ward 2
##                                        -2.825e-01
##      as.factor(CNDTN)Poor:as.factor(WARD)Ward 2
##                                        -1.827e+01
## as.factor(CNDTN)Very Good:as.factor(WARD)Ward 2
##                                        -1.226e+00
## as.factor(CNDTN)Excellent:as.factor(WARD)Ward 3
##                                         9.150e+00
##      as.factor(CNDTN)Fair:as.factor(WARD)Ward 3
##                                        -8.612e-01
##      as.factor(CNDTN)Good:as.factor(WARD)Ward 3
##                                        -1.510e-01
##      as.factor(CNDTN)Poor:as.factor(WARD)Ward 3
##                                        -2.300e+00
## as.factor(CNDTN)Very Good:as.factor(WARD)Ward 3
##                                         2.717e-01
## as.factor(CNDTN)Excellent:as.factor(WARD)Ward 4
##                                        -7.510e-01
##      as.factor(CNDTN)Fair:as.factor(WARD)Ward 4
##                                        -4.479e-02
##      as.factor(CNDTN)Good:as.factor(WARD)Ward 4
##                                         1.751e-01
##      as.factor(CNDTN)Poor:as.factor(WARD)Ward 4
##                                        -2.046e+00
## as.factor(CNDTN)Very Good:as.factor(WARD)Ward 4
##                                        -1.410e-01
## as.factor(CNDTN)Excellent:as.factor(WARD)Ward 5
##                                        -3.615e-01
```

```
##      as.factor(CNDTN)Fair:as.factor(WARD)Ward 5
##                                      8.068e-01
##      as.factor(CNDTN)Good:as.factor(WARD)Ward 5
##                                     -3.390e-01
##      as.factor(CNDTN)Poor:as.factor(WARD)Ward 5
##                                             NA
## as.factor(CNDTN)Very Good:as.factor(WARD)Ward 5
##                                      7.845e-01
## 
## Degrees of Freedom: 21417 Total (i.e. Null);  21373 Residual
## Null Deviance:       19080 
## Residual Deviance: 16870     AIC: 16960

## BIC
sampsize <- length(model_inter$fitted)
step(model_inter, direction="both", k=log(sampsize))

## Start:  AIC=17317.96
## QUALIFIED_2 ~ PRICE + as.factor(AC) + ROOMS + BEDRM + as.factor(CNDTN) + 
##     as.factor(WARD) + PRICE * as.factor(AC) + PRICE * ROOMS + 
##     PRICE * BEDRM + PRICE * as.factor(CNDTN) + PRICE * as.factor(WARD) + 
##     as.factor(CNDTN) * as.factor(WARD)
## 
##                                    Df Deviance   AIC
## - as.factor(CNDTN):as.factor(WARD) 19    16961 17220
## - PRICE:as.factor(CNDTN)            5    16896 17294
## - PRICE:as.factor(AC)               1    16875 17314
## - PRICE:BEDRM                       1    16876 17314
## <none>                                   16869 17318
## - PRICE:ROOMS                       1    16897 17336
## - PRICE:as.factor(WARD)             4    17005 17414
## 
## Step:  AIC=17220.54
## QUALIFIED_2 ~ PRICE + as.factor(AC) + ROOMS + BEDRM + as.factor(CNDTN) + 
##     as.factor(WARD) + PRICE:as.factor(AC) + PRICE:ROOMS + PRICE:BEDRM + 
##     PRICE:as.factor(CNDTN) + PRICE:as.factor(WARD)
## 
##                                    Df Deviance   AIC
## - PRICE:as.factor(CNDTN)            5    16995 17204
## - PRICE:BEDRM                       1    16969 17218
## - PRICE:as.factor(AC)               1    16969 17219
## <none>                                   16961 17220
## - PRICE:ROOMS                       1    16986 17236
## + as.factor(CNDTN):as.factor(WARD) 19    16869 17318
## - PRICE:as.factor(WARD)             4    17119 17338
## 
## Step:  AIC=17203.99
## QUALIFIED_2 ~ PRICE + as.factor(AC) + ROOMS + BEDRM + as.factor(CNDTN) + 
##     as.factor(WARD) + PRICE:as.factor(AC) + PRICE:ROOMS + PRICE:BEDRM + 
##     PRICE:as.factor(WARD)
```

```
##
##                                      Df Deviance   AIC
## - PRICE:BEDRM                         1    17000 17200
## <none>                                     16995 17204
## - PRICE:as.factor(AC)                 1    17011 17210
## + PRICE:as.factor(CNDTN)              5    16961 17220
## - PRICE:ROOMS                         1    17026 17225
## - as.factor(CNDTN)                    5    17112 17272
## + as.factor(CNDTN):as.factor(WARD) 19    16896 17294
## - PRICE:as.factor(WARD)               4    17172 17342
##
## Step:  AIC=17199.68
## QUALIFIED_2 ~ PRICE + as.factor(AC) + ROOMS + BEDRM + as.factor(CNDTN) +
##     as.factor(WARD) + PRICE:as.factor(AC) + PRICE:ROOMS +
PRICE:as.factor(WARD)
##
##                                      Df Deviance   AIC
## <none>                                     17000 17200
## + PRICE:BEDRM                         1    16995 17204
## - PRICE:as.factor(AC)                 1    17016 17205
## + PRICE:as.factor(CNDTN)              5    16969 17218
## - BEDRM                               1    17035 17224
## - as.factor(CNDTN)                    5    17119 17268
## - PRICE:ROOMS                         1    17083 17273
## + as.factor(CNDTN):as.factor(WARD) 19    16901 17290
## - PRICE:as.factor(WARD)               4    17187 17347

##
## Call:  glm(formula = QUALIFIED_2 ~ PRICE + as.factor(AC) + ROOMS + BEDRM +
##     as.factor(CNDTN) + as.factor(WARD) + PRICE:as.factor(AC) +
##     PRICE:ROOMS + PRICE:as.factor(WARD), family = binomial(link = logit),
##     data = Final_T)
##
## Coefficients:
##                 (Intercept)                        PRICE
##                   8.887e-01                    2.459e-06
##             as.factor(AC)Y                        ROOMS
##                  -6.852e-02                    7.034e-03
##                       BEDRM    as.factor(CNDTN)Excellent
##                  -1.557e-01                    1.475e-01
##        as.factor(CNDTN)Fair         as.factor(CNDTN)Good
##                  -1.176e+00                    3.397e-01
##        as.factor(CNDTN)Poor    as.factor(CNDTN)Very Good
##                  -1.240e+00                    6.949e-01
##       as.factor(WARD)Ward 2        as.factor(WARD)Ward 3
##                   4.625e-01                   -5.449e-01
##       as.factor(WARD)Ward 4        as.factor(WARD)Ward 5
##                  -6.263e-01                   -5.117e-01
##        PRICE:as.factor(AC)Y                  PRICE:ROOMS
##                   6.464e-07                   -1.364e-07
```

```
## PRICE:as.factor(WARD)Ward 2  PRICE:as.factor(WARD)Ward 3
##                 -7.334e-07                  1.204e-06
## PRICE:as.factor(WARD)Ward 4  PRICE:as.factor(WARD)Ward 5
##                  2.731e-06                  2.265e-06
##
## Degrees of Freedom: 21417 Total (i.e. Null);  21398 Residual
## Null Deviance:       19080
## Residual Deviance: 17000     AIC: 17040
```

Final Model

```
final_model <- glm(as.factor(QUALIFIED_2) ~ PRICE + I(PRICE^0.5) +
as.factor(AC) + ROOMS + I(ROOMS^.2) + I(BEDRM^0.5) + as.factor(CNDTN) +
as.factor(WARD) + PRICE*as.factor(AC) + PRICE*ROOMS + PRICE*as.factor(WARD),
family = binomial(link = logit), data = Final_T)
summary(final_model)

##
## Call:
## glm(formula = as.factor(QUALIFIED_2) ~ PRICE + I(PRICE^0.5) +
##     as.factor(AC) + ROOMS + I(ROOMS^0.2) + I(BEDRM^0.5) + as.factor(CNDTN)
+
##     as.factor(WARD) + PRICE * as.factor(AC) + PRICE * ROOMS +
##     PRICE * as.factor(WARD), family = binomial(link = logit),
##     data = Final_T)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7555   0.3175   0.4181   0.5929   1.9791
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -3.158e+00  9.183e-01  -3.439 0.000583 ***
## PRICE                        -4.374e-06  3.945e-07 -11.089  < 2e-16 ***
## I(PRICE^0.5)                  7.901e-03  4.029e-04  19.609  < 2e-16 ***
## as.factor(AC)Y               -2.907e-01  8.067e-02  -3.603 0.000314 ***
## ROOMS                        -1.821e-01  3.423e-02  -5.321 1.03e-07 ***
## I(ROOMS^0.2)                  2.221e+00  7.669e-01   2.896 0.003775 **
## I(BEDRM^0.5)                 -4.816e-01  1.055e-01  -4.564 5.02e-06 ***
## as.factor(CNDTN)Excellent     1.069e-01  1.513e-01   0.707 0.479819
## as.factor(CNDTN)Fair         -1.196e+00  2.259e-01  -5.294 1.19e-07 ***
## as.factor(CNDTN)Good          2.849e-01  4.681e-02   6.085 1.16e-09 ***
## as.factor(CNDTN)Poor         -1.373e+00  6.671e-01  -2.057 0.039646 *
## as.factor(CNDTN)Very Good     6.739e-01  8.426e-02   7.998 1.26e-15 ***
## as.factor(WARD)Ward 2        -4.306e-01  1.222e-01  -3.523 0.000426 ***
## as.factor(WARD)Ward 3        -2.858e-01  1.055e-01  -2.709 0.006747 **
## as.factor(WARD)Ward 4        -5.150e-02  1.432e-01  -0.360 0.719011
## as.factor(WARD)Ward 5         2.901e-01  1.243e-01   2.334 0.019583 *
## PRICE:as.factor(AC)Y          1.085e-06  1.653e-07   6.561 5.36e-11 ***
## PRICE:ROOMS                   2.698e-08  1.453e-08   1.856 0.063431 .
## PRICE:as.factor(WARD)Ward 2   3.897e-07  1.451e-07   2.685 0.007246 **
```

```
## PRICE:as.factor(WARD)Ward 3  5.486e-07  1.733e-07   3.166 0.001547 **
## PRICE:as.factor(WARD)Ward 4  1.052e-06  3.618e-07   2.908 0.003635 **
## PRICE:as.factor(WARD)Ward 5 -3.313e-07  3.750e-07  -0.883 0.377046
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 19081  on 21417  degrees of freedom
## Residual deviance: 16704  on 21396  degrees of freedom
## AIC: 16748
##
## Number of Fisher Scoring iterations: 5
```

Section 2: Visualisation

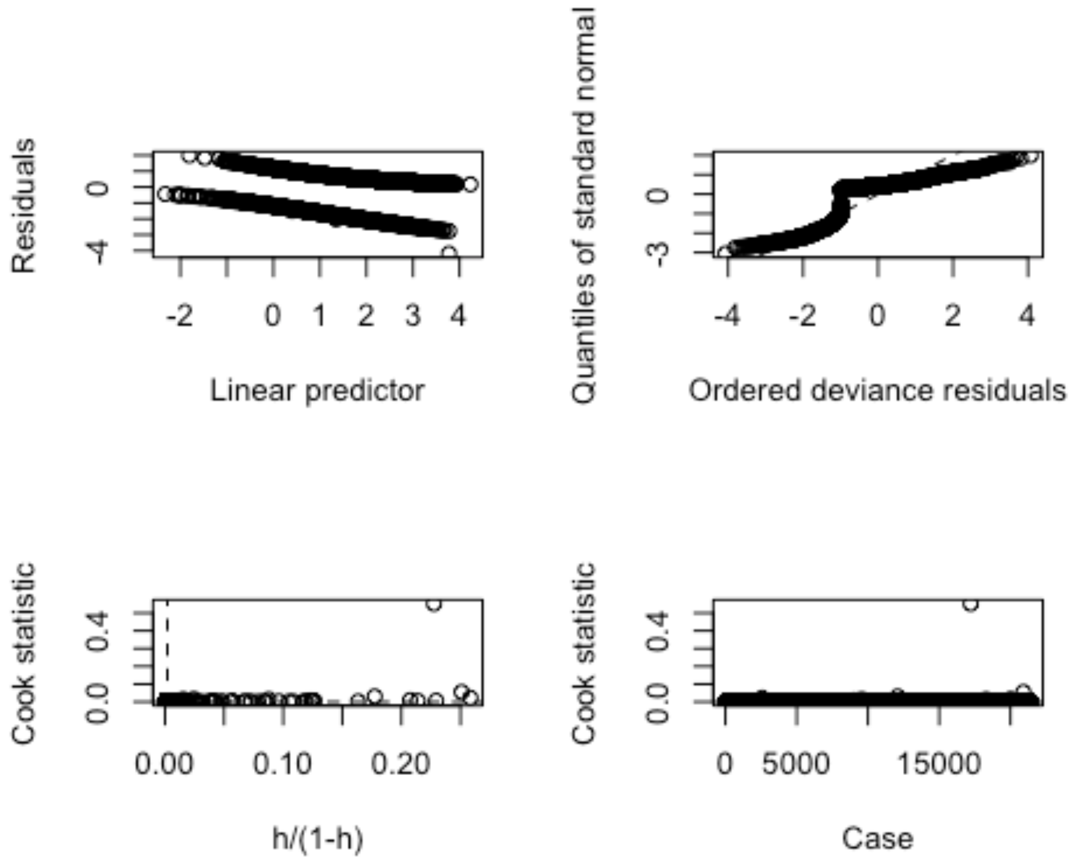
Below are some diagnostic plot and visualisation that helped me understand more of the data and the models I tried to create.

Section 2.1: Analysis Visualisations

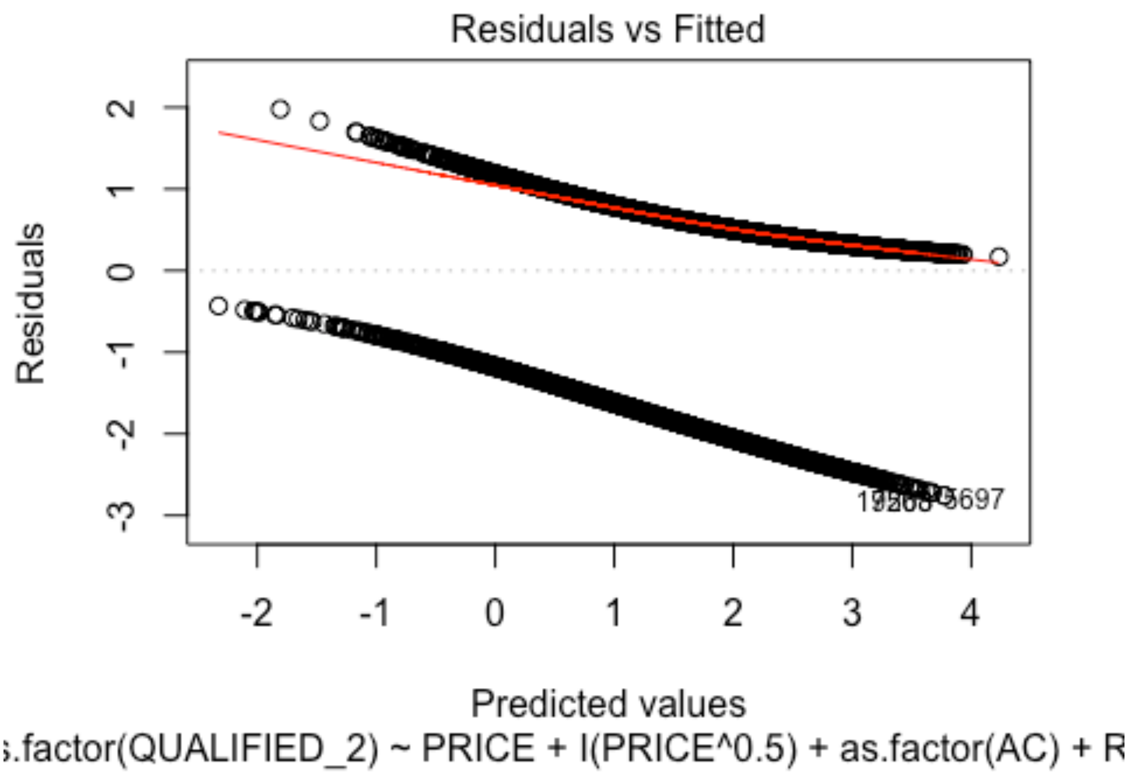
Diagnostic Plots

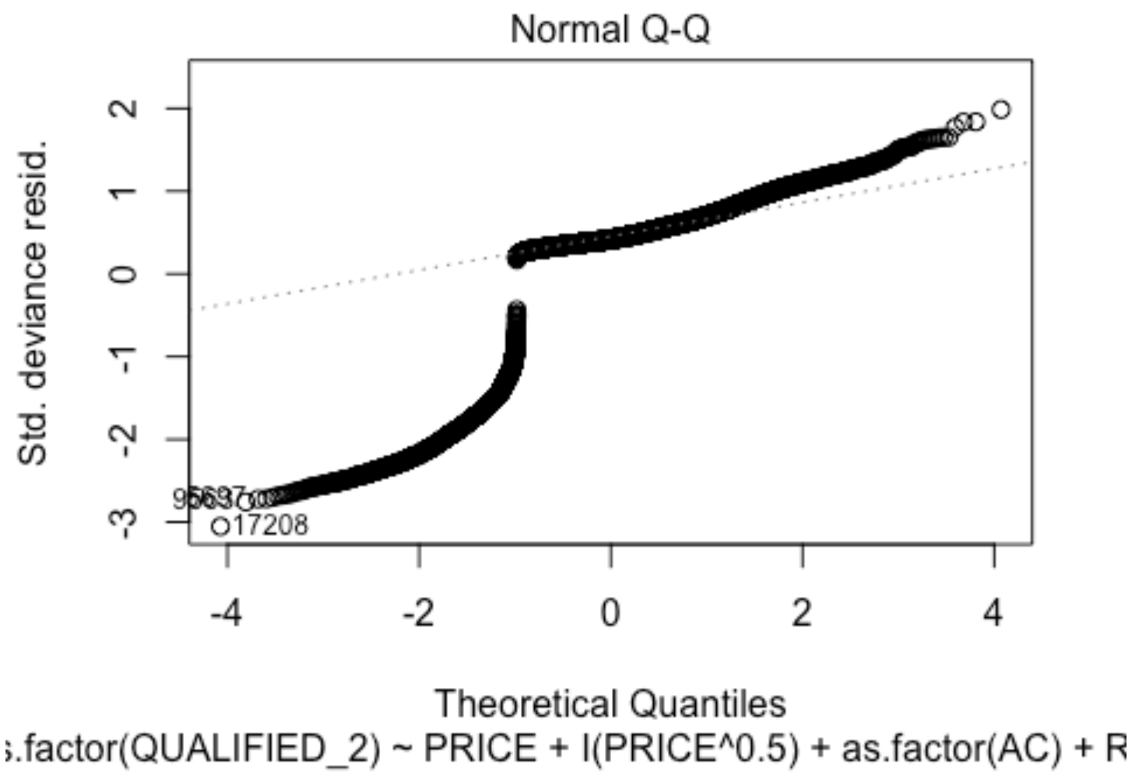
```
a <- c(1:10)

final_modelT.diag <- glm.diag(final_model)
final_modelT.diag$rd[a]  # Standardized Deviance Residuals

##          1          2          3          4          5          6
##  0.3108682  0.3530772  0.3783794  0.4236119  0.2552592  0.2488656
##          7          8          9         10
##  0.2708673  0.4840084 -0.6986174  0.2640823

final_modelT.diag$rp[a]  # Standardized Person Residual

##          1          2          3          4          5          6
##  0.2224944  0.2536005  0.2724057  0.3063813  0.1819748  0.1773450
##          7          8          9         10
##  0.1933014  0.3525130 -0.5255236  0.1883735

final_modelT.diag$cook[a]  # Cook's D

##            1            2            3            4            5
## 4.060001e-06 3.120757e-06 6.435024e-06 2.324495e-06 7.918558e-07
##            6            7            8            9           10
## 7.445751e-07 7.139247e-07 1.899438e-06 7.616752e-05 6.319234e-07

final_modelT.diag$h[a]  # Leverages

##  [1] 0.0018010582 0.0010663970 0.0019042003 0.0005444904 0.0005257965
##  [6] 0.0005205554 0.0004201674 0.0003361643 0.0060308836 0.0003916313
```

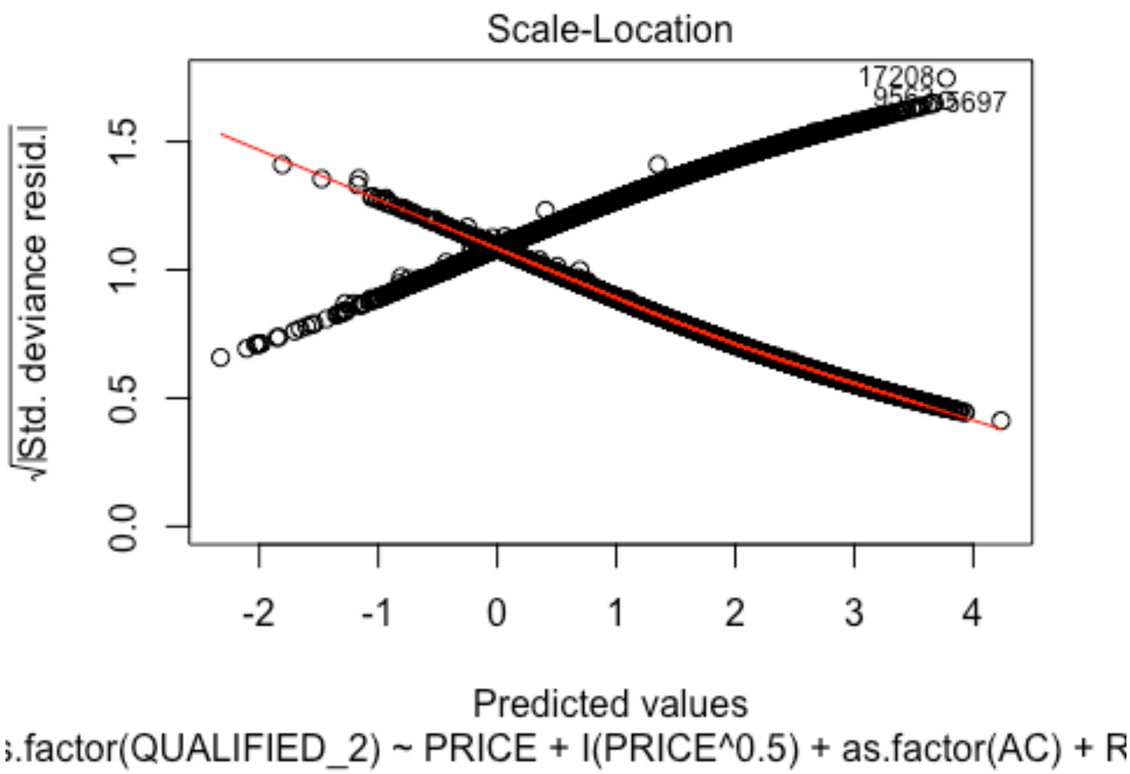
```
glm.diag.plots(final_model)
```

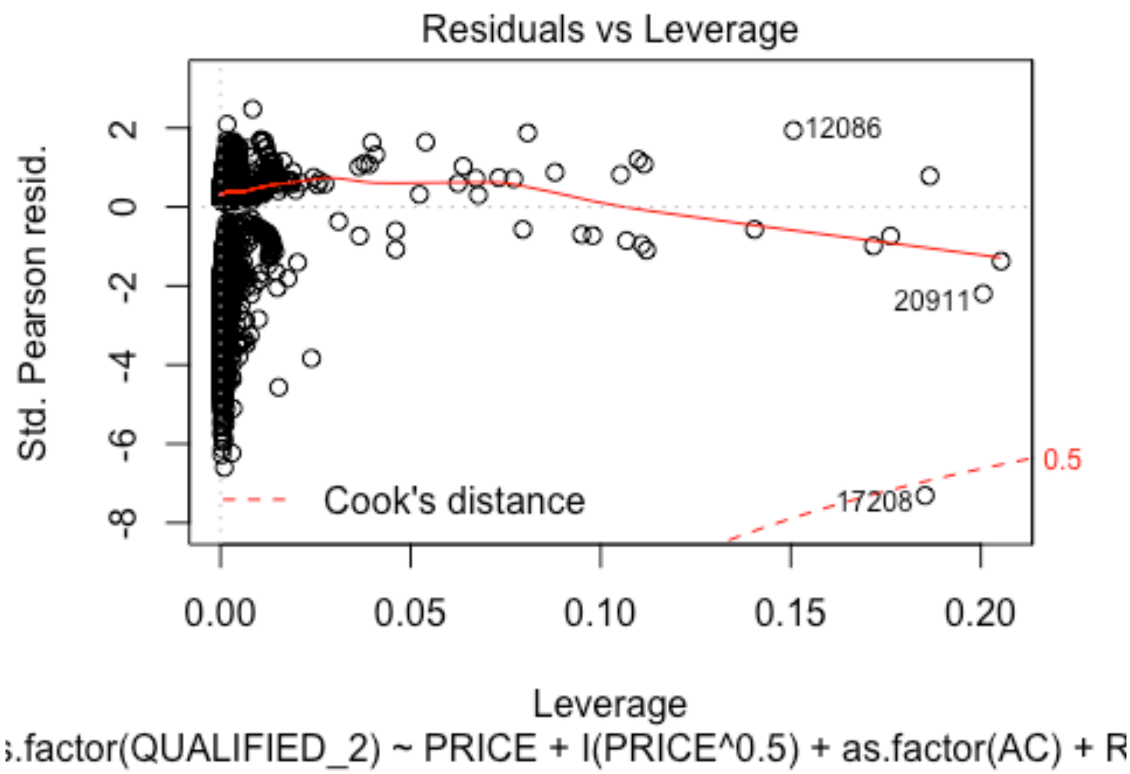



```
plot(final_model)
```


Residuals vs Fitted
Residuals
2
1
0
-1
-2
-3
-2
-1
0
1
2
3
4
Predicted values
s.factor(QUALIFIED_2) ~ PRICE + I(PRICE^0.5) + as.factor(AC) + R


Normal Q-Q
Std. deviance resid.
Theoretical Quantiles
2
1
0
-1
-2
-3
-4
-2
0
2
4
17208
s.factor(QUALIFIED_2) ~ PRICE + I(PRICE^0.5) + as.factor(AC) + R


Scale-Location
17208
95
5697
1.5
1.0
0.5
0.0
√|Std. deviance resid.|
-2
-1
0
1
2
3
4
Predicted values
s.factor(QUALIFIED_2) ~ PRICE + I(PRICE^0.5) + as.factor(AC) + R

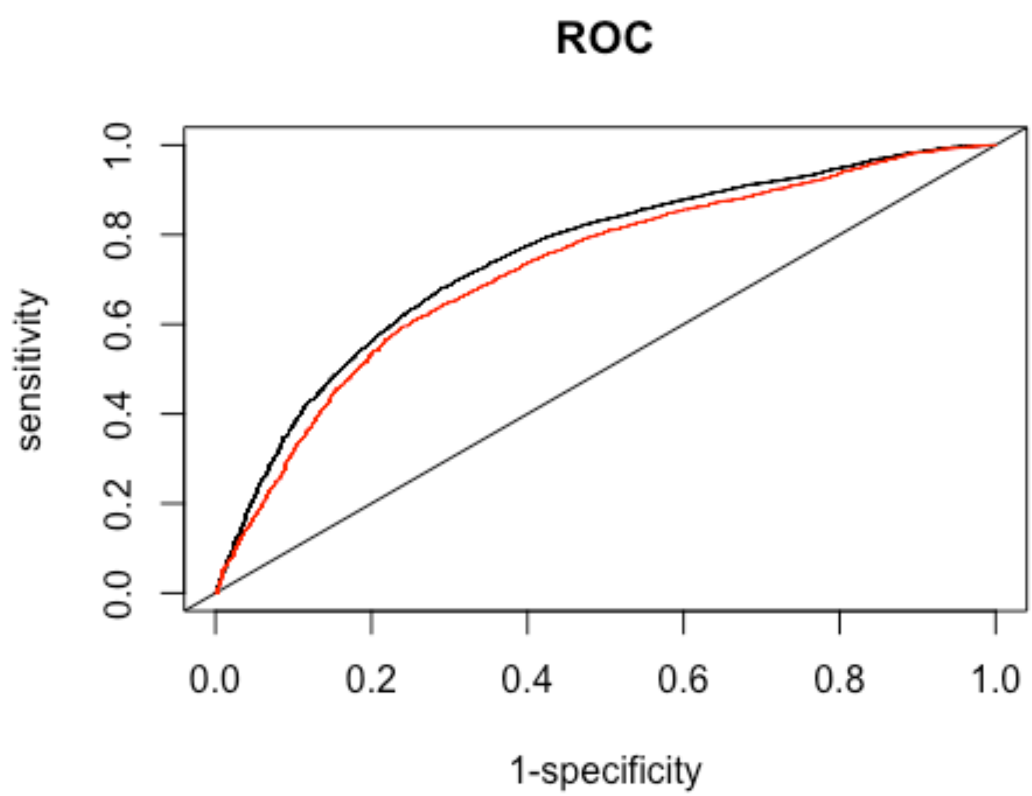


One can see from these diagnostic plots that we do not have normally distributed data. So the model will have to apply transformations to the predictor variables to make the data and model more normally distributed.

ROC Curve

```
roc.analysis <-function (object, newdata = NULL, newplot=TRUE)
{
  if (is.null(newdata)) {
    pi.tp <- object$fitted[object$y == 1]
    pi.tn <- object$fitted[object$y == 0]
  }
  else {
    pi.tp <- predict(object, newdata, type = "response")[newdata$y == 1]
    pi.tn <- predict(object, newdata, type = "response")[newdata$y == 0]
  }

  pi.all <- sort(c(pi.tp, pi.tn))
  sens <- rep(1, length(pi.all)+1)
  specc <- rep(1, length(pi.all)+1)
  for (i in 1:length(pi.all)) {
    sens[i+1] <- mean(pi.tp >= pi.all[i], na.rm = T)
    specc[i+1] <- mean(pi.tn >= pi.all[i], na.rm = T)
  }
```

```
  npoints <- length(sens)
  area <- sum(0.5 * (sens[-1] + sens[-npoints]) * (specc[-npoints] -
        specc[-1]))
  lift <- (sens - specc)[-1]
  cutoff <- pi.all[lift == max(lift)][1]
  sensopt <- sens[-1][lift == max(lift)][1]
  specopt <- 1 - specc[-1][lift == max(lift)][1]

  if (newplot){
  plot(specc, sens, xlim = c(0, 1), ylim = c(0, 1), type = "s",
            xlab = "1-specificity", ylab = "sensitivity", main="ROC")
  abline(0, 1)
  }
  else lines(specc, sens, type="s", lty=2, col=2)

  list(pihat=as.vector(pi.all), sens=as.vector(sens[-1]),
  spec=as.vector(1-specc[-1]), area = area, cutoff = cutoff,
  sensopt = sensopt, specopt = specopt)
}

b <- c(1:10, 34317:34327)
trainingROC <- roc.analysis(final_model)
trainingROC$area

## [1] 0.7497219

trainingROC$cutoff

##     18551
## 0.8461571

trainingROC$sensopt

## [1] 0.6789462

trainingROC$specopt

## [1] 0.7118789

Final_V$y <- Final_V$QUALIFIED_2
validationROC <- roc.analysis(final_model, newdata=Final_V, newplot=F)
```




```
validationROC$area

## [1] 0.7233407

validationROC$cutoff

##      7238
## 0.8782725

validationROC$sensopt

## [1] 0.595809

validationROC$specopt

## [1] 0.7596588
```

As one can see the training set and validation set lines are very close to one another which is very good. The closer the black (training) and red (validation) lines are to one another the better our model is. However both areas are still relatively very low and it would of been better if they were above 0.85. Since the areas are below 0.85, we can conclude that the model needs work.

Variance Inflation Factors (VIF)

```
library(rms)

vif(glm(as.factor(QUALIFIED_2) ~ PRICE + I(PRICE^0.5) + as.factor(AC) + ROOMS
+ I(ROOMS^.2) + I(BEDRM^0.5) + as.factor(CNDTN) + as.factor(WARD) +
PRICE*as.factor(AC) + PRICE*ROOMS + PRICE*as.factor(WARD), family =
binomial(link = logit), data = Final_T))

##                        PRICE                 I(PRICE^0.5)
##                   103.585353                    31.046266
##               as.factor(AC)Y                        ROOMS
##                     3.078162                    11.516670
##                 I(ROOMS^0.2)                 I(BEDRM^0.5)
##                     9.250033                     1.983626
##    as.factor(CNDTN)Excellent         as.factor(CNDTN)Fair
##                     1.162295                     1.013751
##         as.factor(CNDTN)Good         as.factor(CNDTN)Poor
##                     1.377496                     1.002527
##    as.factor(CNDTN)Very Good        as.factor(WARD)Ward 2
##                     1.334604                     5.393325
##        as.factor(WARD)Ward 3        as.factor(WARD)Ward 4
##                     6.147429                     4.980247
##        as.factor(WARD)Ward 5         PRICE:as.factor(AC)Y
##                     6.713264                    20.203751
##                  PRICE:ROOMS PRICE:as.factor(WARD)Ward 2
##                    22.719759                    14.479582
## PRICE:as.factor(WARD)Ward 3 PRICE:as.factor(WARD)Ward 4
##                     5.167960                     4.013284
## PRICE:as.factor(WARD)Ward 5
##                     4.430496
```

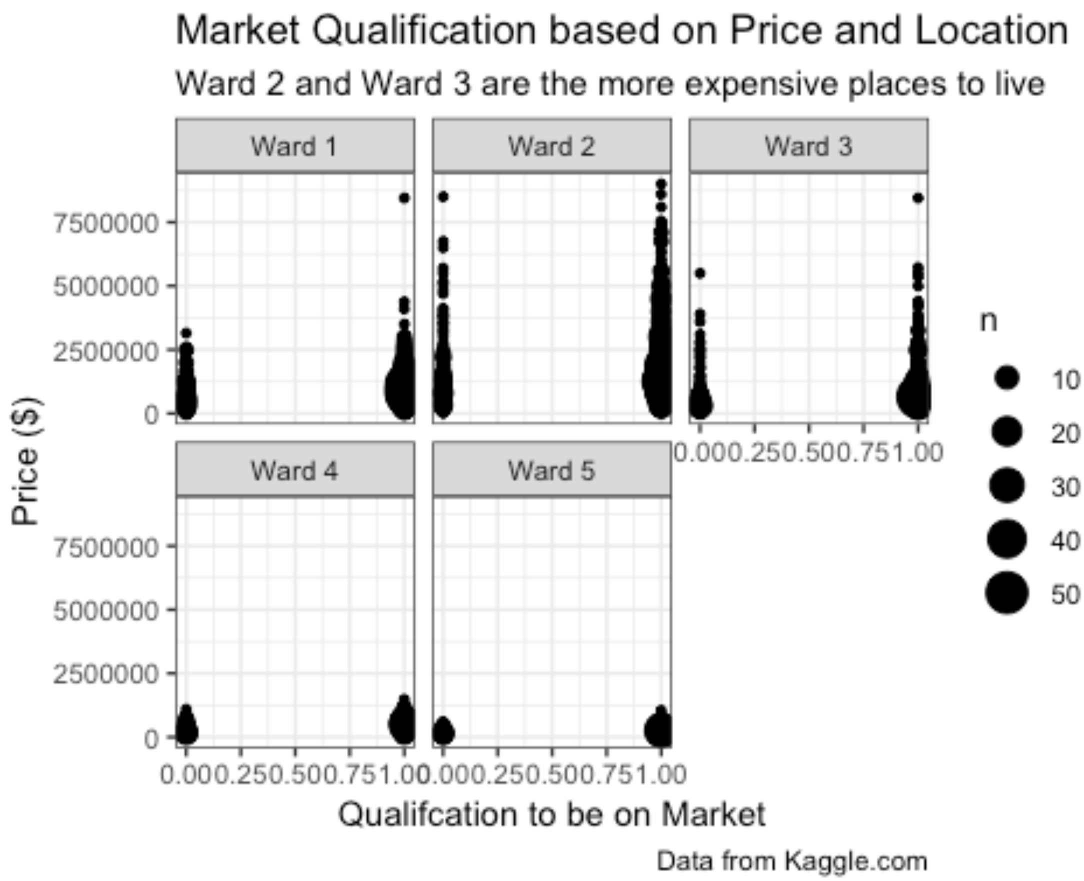
It seems that multicollinearity is still an issue with the interaction terms and transformations. Number that are above 5 means that the variables are too closely correlated with one another and that is a bad thing to have. The saving grace here is that the numbers are only above 5 in the interaction terms and transformation variables which makes sense.

Section 2.2: Model Visualisation

Basic Model Graph

```
(graph <- ggplot(dcproperty, aes(y=PRICE, x=QUALIFIED_2)) +
    stat_sum() +
    stat_smooth(method="glm",
                method.args = list(family="binomial"), se=TRUE,
                fullrange=TRUE) +
    labs(title = "Market Qualification based on Price and Location",
       subtitle = "Ward 2 and Ward 3 are the more expensive places to live",
       caption = "Data from Kaggle.com",
       y = "Price ($)",
       x = "Qualifcation to be on Market",
```

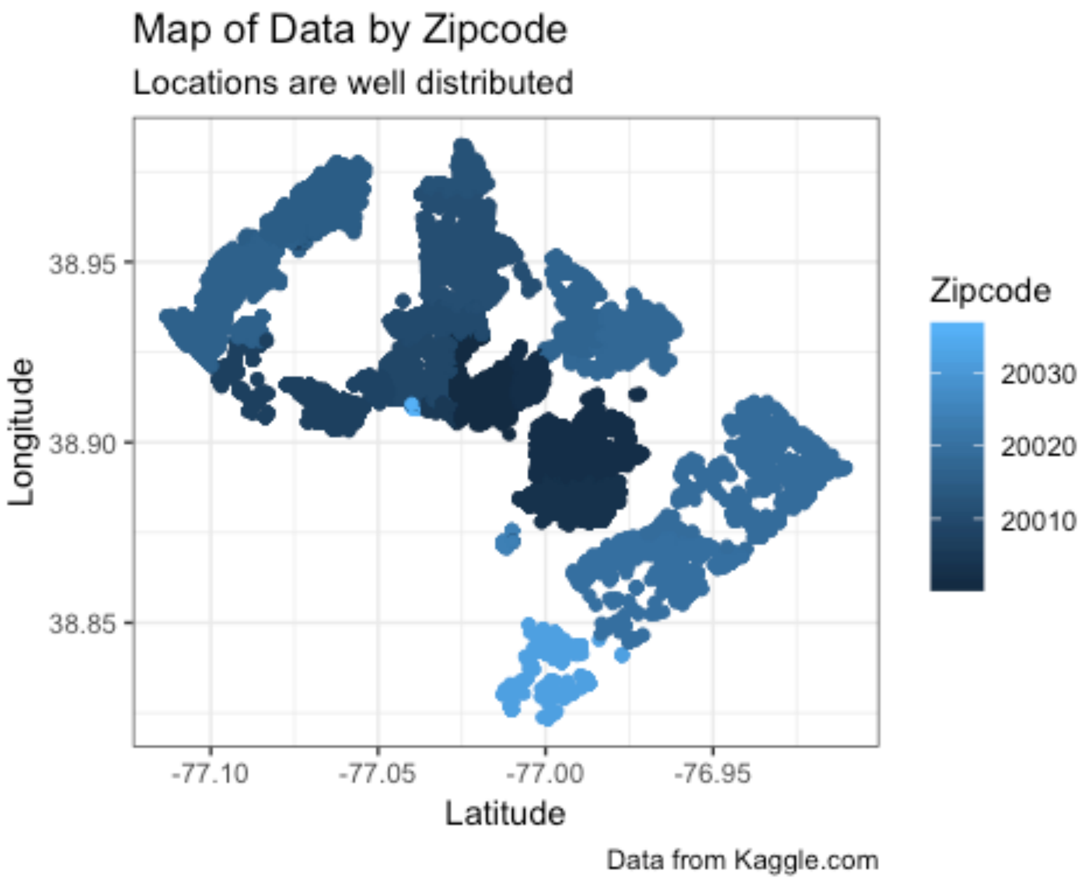
```
      color = "Qualification") +
  facet_wrap(~WARD) +
  theme_bw())
```



One can see that Ward 2 and 3 has a higher distribution of price rangers, while Ward 4 and 5 have low priced property but the qualifation ratios are nearly identical.

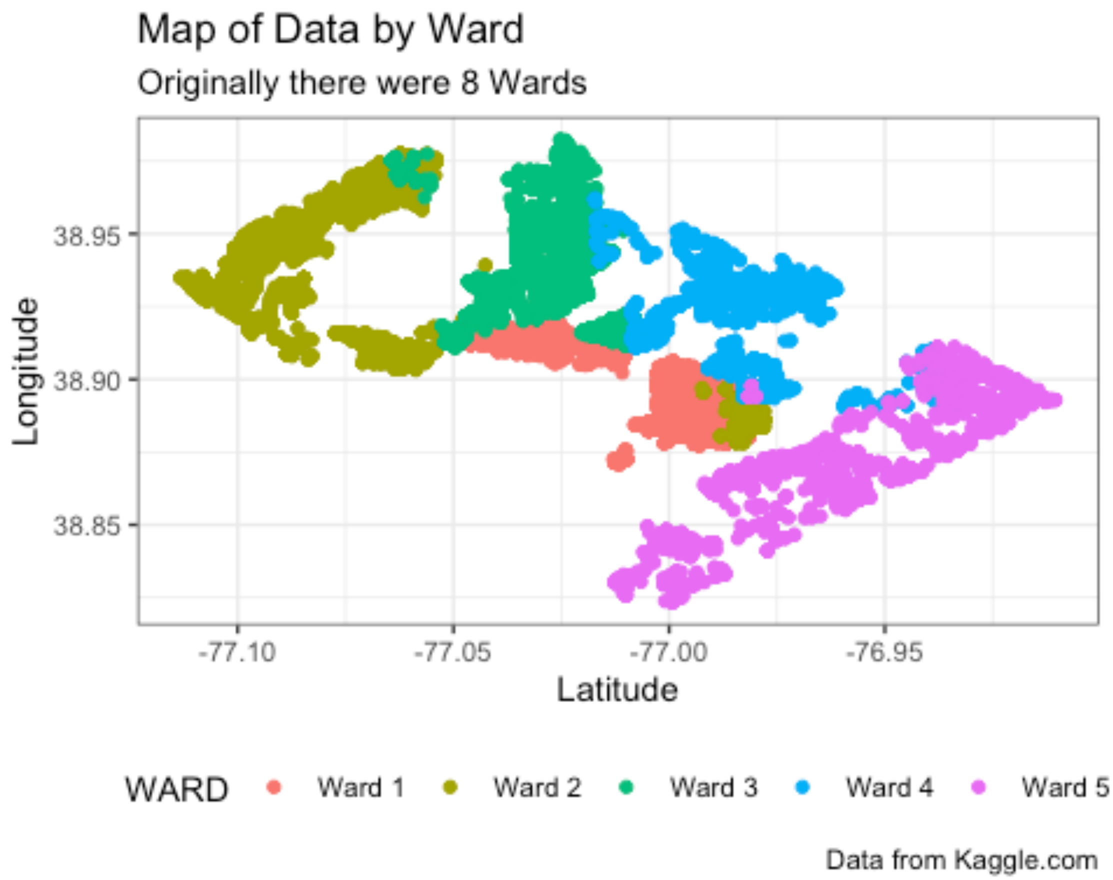
Map Graphs

```
ggplot(dcproperty, aes(x=X, y=Y)) +
  geom_point(aes(color=ZIPCODE)) +
  labs(title = "Map of Data by Zipcode",
      subtitle = "Locations are well distributed",
      caption = "Data from Kaggle.com",
      y = "Longitude",
      x = "Latitude",
      color = "Zipcode") +
  theme_bw() +
  theme(legend.position = "right")
```



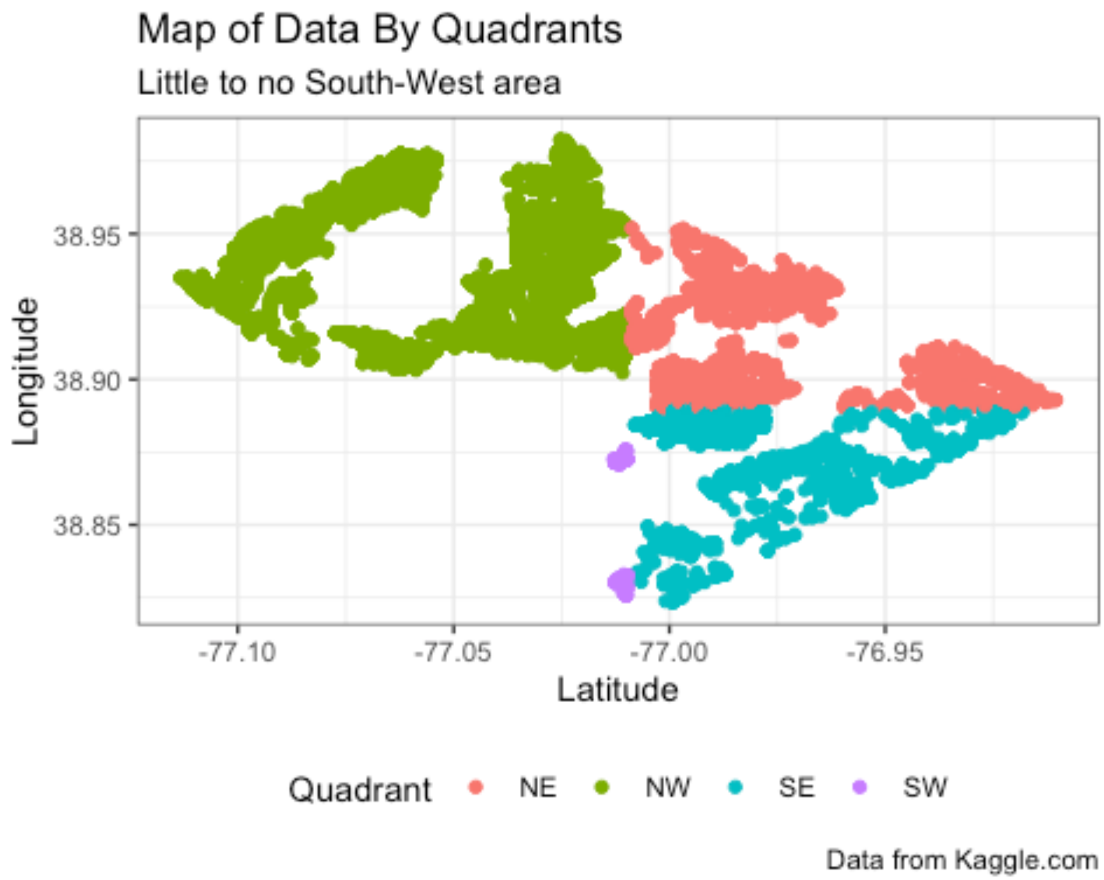
It seems that the zipcodes are distributed very well across our DC data.

```
ggplot(dcproperty, aes(x=X, y=Y)) +
  geom_point(aes(color=WARD)) +
  labs(title = "Map of Data by Ward",
       subtitle = "Originally there were 8 Wards",
       caption = "Data from Kaggle.com",
       y = "Longitude",
       x = "Latitude",
       color = "WARD") +
  theme_bw() +
  theme(legend.position = "bottom")
```



It seems that the Wards numbers have changed over time so that is why some of these colored dots are not where they are suppose to be. Through cleaning of the data we lost Wards 6-8 but there is nothing we can do since if we kept them our model would not run properly.

```
ggplot(dcproperty, aes(x=X, y=Y)) +
  geom_point(aes(color=QUADRANT)) +
  labs(title = "Map of Data By Quadrants",
       subtitle = "Little to no South-West area",
       caption = "Data from Kaggle.com",
       y = "Longitude",
       x = "Latitude",
       color = "Quadrant") +
  theme_bw() +
  theme(legend.position = "bottom")
```



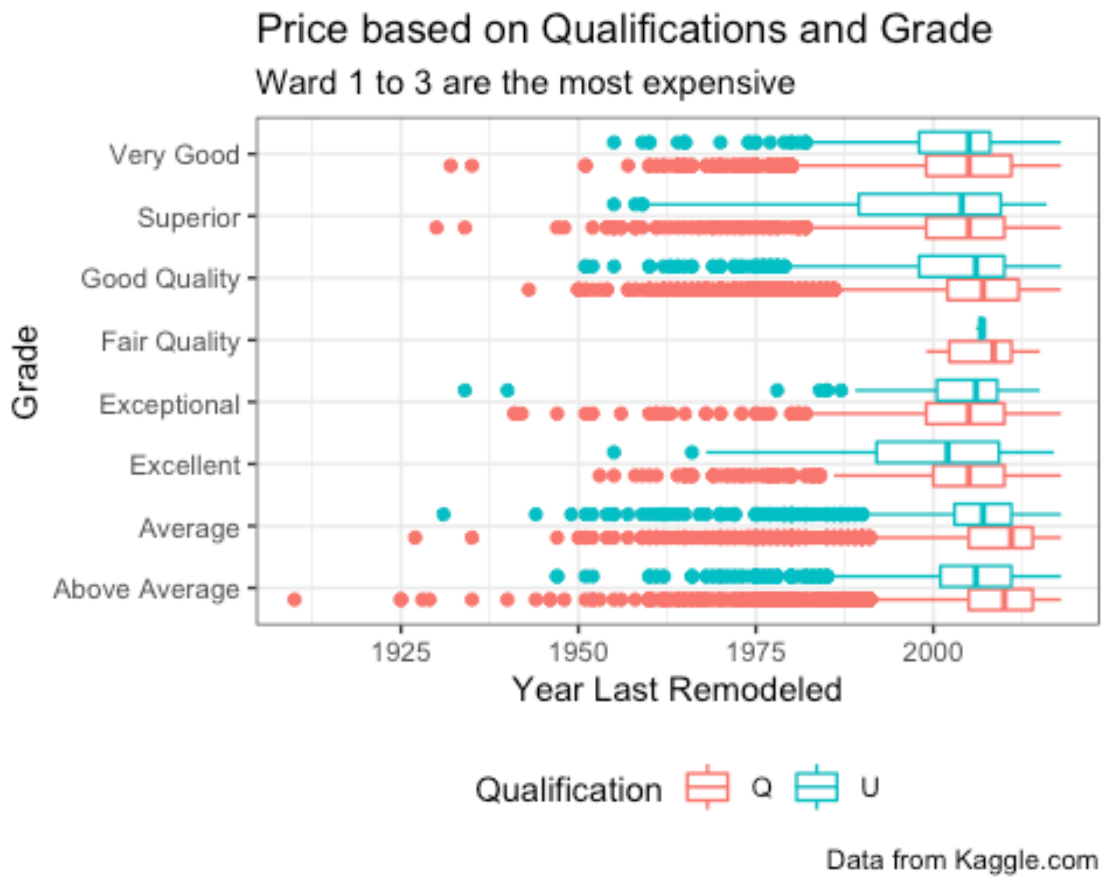
It seems that most of our properties are in the northwest region of DC. This makes sense considering a lot of the schools and unviersities are in this area. Southwest is the smallest because it is the smallest region in the map anyway. Also I believe that the southwest dots listed above are waterfront property or at least have nice views, otherwise we may not of had any observations in the southwest region.

Year Graphs

```
## use these graphs below

dcproperty %>%
  filter(YR_RMDL > 1800) %>%
  ggplot(mapping = aes(y=YR_RMDL, x=QUADRANT, color = QUALIFIED)) +
  geom_boxplot() +
  labs(title = "Price based on Qualifications and Quadrant",
       subtitle = "Ward 1 to 3 are the most expensive",
       caption = "Data from Kaggle.com",
       y = "Year Last Remodeled",
       x = "Quadrant",
       color = "Qualification") +
  theme_bw() +
  theme(legend.position = "bottom")
```

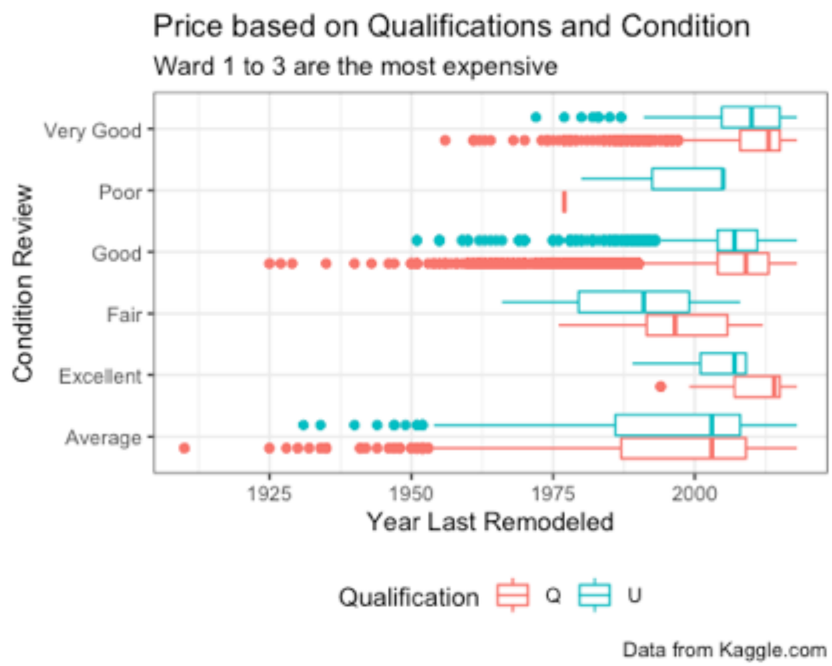
The southwest region having such a large boxplot can be explained through the Quadrant DC map. As for the rest of the regions, it is interesting that all the means are between 2000-2010, but the first remodeled year was before 1920 in the southeast region.

```
dcproperty %>%
  filter(YR_RMDL > 1800) %>%
  ggplot(mapping = aes(y=YR_RMDL, x=GRADE, color = QUALIFIED)) +
  geom_boxplot() +
  coord_flip() +
  labs(title = "Price based on Qualifications and Grade",
       subtitle = "Ward 1 to 3 are the most expensive",
       caption = "Data from Kaggle.com",
       y = "Year Last Remodeled",
       x = "Grade",
       color = "Qualification") +
  theme_bw() +
  theme(legend.position = "bottom")
```



The most interesting thing here is that there were no outliers in the plot for the fair quality grade. It is also interesting to note that all the mean year remodeled grades are between 2005-2010. It seems that either the system got much better or because communication and reviiewing become more popular in the last 20 years.

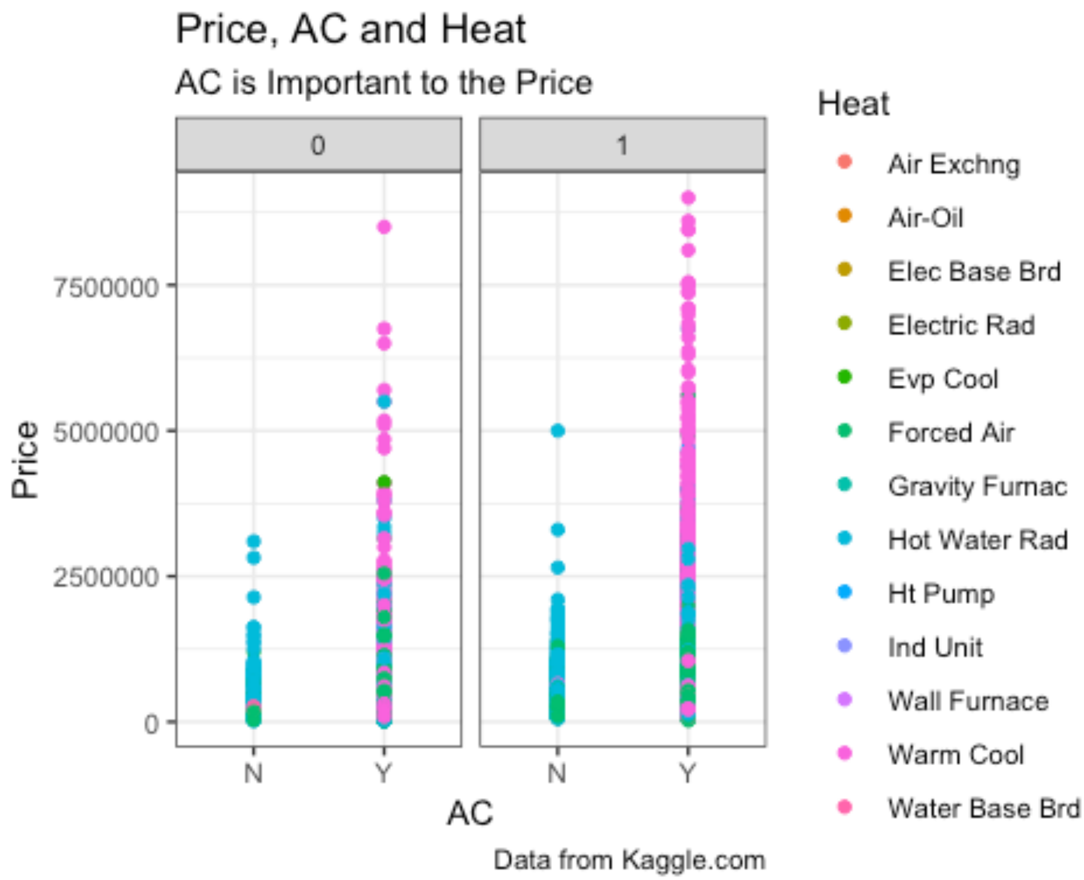
```
dcproperty %>%
  filter(YR_RMDL > 1800) %>%
  ggplot(mapping = aes(y=YR_RMDL, x=CNDTN, color = QUALIFIED)) +
  geom_boxplot() +
  coord_flip() +
  labs(title = "Price based on Qualifications and Condition",
       subtitle = "Ward 1 to 3 are the most expensive",
       caption = "Data from Kaggle.com",
       y = "Year Last Remodeled",
       x = "Condition Review",
       color = "Qualification") +
  theme_bw() +
  theme(legend.position = "bottom")
```



This is a very interesting plot to observe. It is interesting that the excellent condition review had the least outliers, fair condition review had no outliers, and good condition review had the most outliers. It seems that a good condition review is the most popular. I believe that everything above can be explained by people have too high of an opinion and people being lazy and loving to comemnt everything as good and average.

Building Heat and AC Graph

```
ggplot(dcproperty, aes(x=AC, y=PRICE)) +
  geom_point(aes(color=HEAT, )) +
  facet_wrap(~QUALIFIED_2) +
  labs(title = "Price, AC and Heat",
       subtitle = "AC is Important to the Price",
       caption = "Data from Kaggle.com",
       y = "Price",
       x = "AC",
       color = "Heat") +
  theme_bw() +
  theme(legend.position = "right")
```

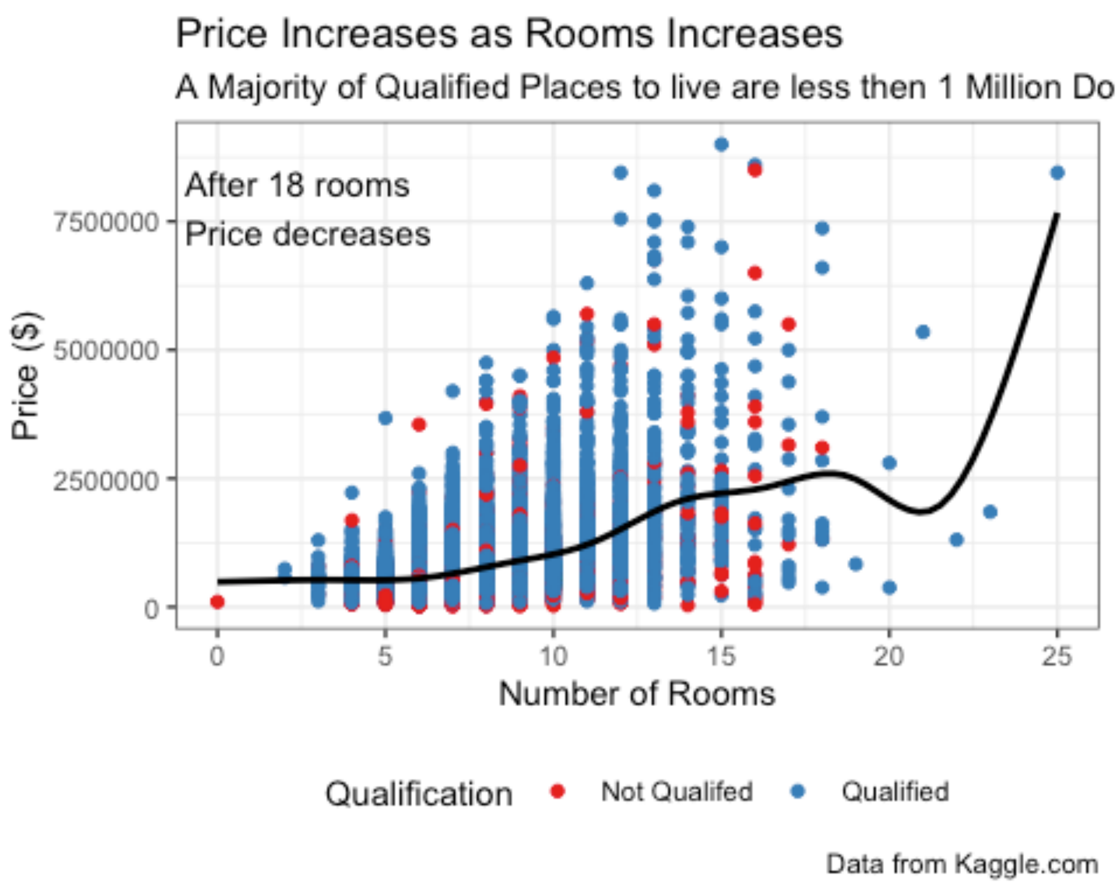


It seems that the distribution of price between having AC and being a qualified property is about the same. Yet when it comes to warm heat that is what decides how high a price was able to be raised. I do not understand the differences in heat that much so I am unable to comment further on this plot.

Continuous Variable Plots

```
text_df <- tibble(text = " \n After 18 rooms \n Price decreases", x = -Inf, y = Inf)
ggplot(dcproperty, aes(ROOMS, PRICE)) +
  geom_point(aes(color = factor(QUALIFIED_2, labels = c("Not Qualifed", "Qualified")))) +
  geom_smooth(se = FALSE, color = "black") +
  labs(title = "Price Increases as Rooms Increases",
       subtitle = "A Majority of Qualified Places to live are less then 1 Million Dollars",
       caption = "Data from Kaggle.com",
       y = "Price ($)",
       x = "Number of Rooms",
       color = "Qualification") +
  geom_text(aes(x, y, label = text), data = text_df, vjust = "top", hjust = "left") +
  scale_colour_brewer(palette = "Set1") +
```

```
  theme_bw() +
  theme(legend.position = "bottom")
```



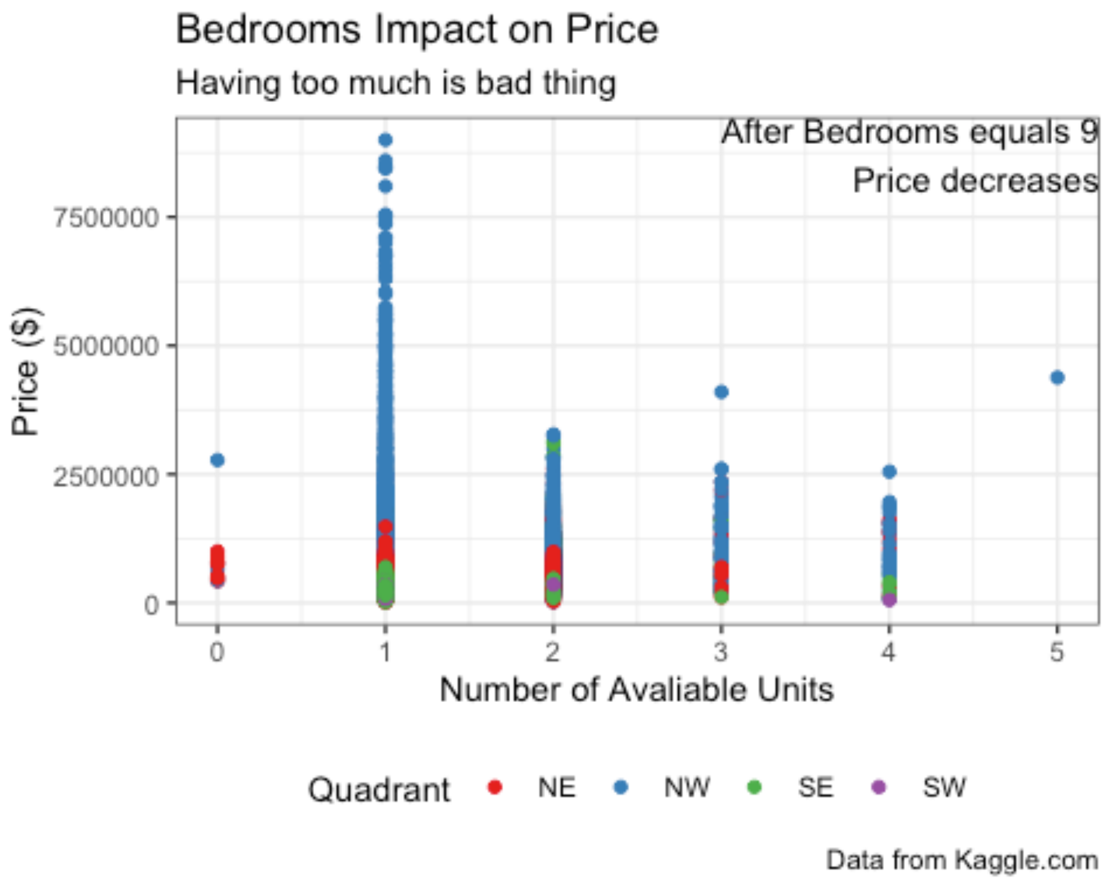
It seems that only when you have at least 5 rooms does hte price of a property start to increase but if you have more than 18 rooms the price will start to drop slowly. I believe that after having 22 rooms might be outliers and that is why we are seeing such a sharp increase in price. Also the more rooms you have the more we see that a property is unqualified to sell.

```
text_df <- tibble(text = "More Kitchens equals\nLower Price & Less
Qualified", x = Inf, y = Inf)
ggplot(dcproperty, aes(KITCHENS, PRICE)) +
  geom_point(aes(color = factor(QUALIFIED_2, labels = c("Not Qualifed",
"Qualified")))) +
  labs(title = "Kitchens Impact on Price",
       subtitle = "Having Fewer Kitchens Increases the Price",
       caption = "Data from Kaggle.com",
       y = "Price ($)",
       x = "Number of Kitchens",
       color = "Qualification") +
  geom_text(aes(x, y, label = text), data = text_df, vjust = "top", hjust =
"right") +
  scale_colour_brewer(palette = "Set1") +
```

```
  theme_bw() +
  theme(legend.position = "bottom")
```

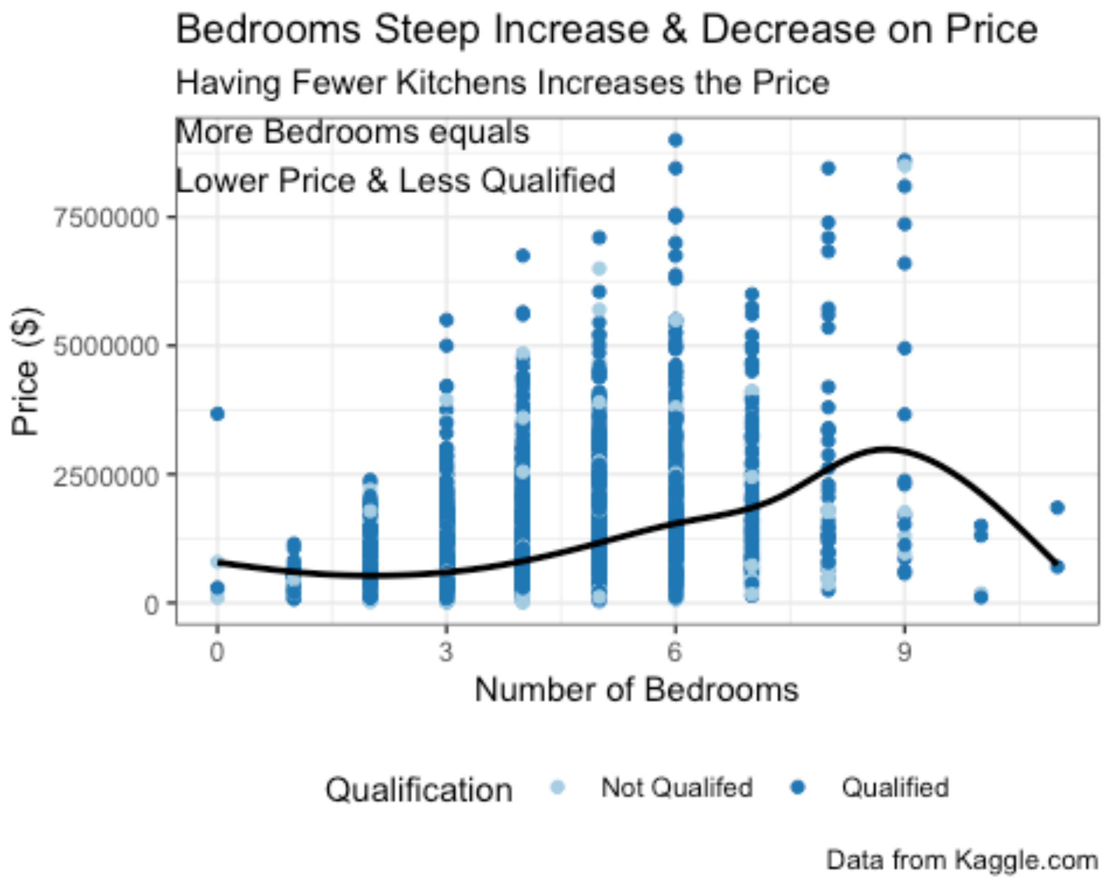
The difference in having a kitchen and not having a kitchen is as clear as day. Not having a kitechen will get you veyr little in price over at least having one kitchen in your property. It is interesting though that price trend starts to decrease after having kitchen. I guess people do not like having too many places to cook and clean.

```
text_df <- tibble(text = "After Bedrooms equals 9\nPrice decreases", x = Inf,
y = Inf)
ggplot(dcproperty, aes(NUM_UNITS, PRICE)) +
  geom_point(aes(color = as.factor(QUADRANT))) +
  labs(title = "Bedrooms Impact on Price",
       subtitle = "Having too much is bad thing",
       caption = "Data from Kaggle.com",
       y = "Price ($)",
       x = "Number of Avaliable Units",
       color = "Quadrant") +
  geom_text(aes(x, y, label = text), data = text_df, vjust = "top", hjust =
"right") +
  scale_colour_brewer(palette = "Set1") +
  theme_bw() +
  theme(legend.position = "bottom")
```



It seems that having at least one property to sell will have a large range of values. having between 2 and 4 similar properties to seems is about the same. If we follow the principles behind supply and demand this graph makes perfect sense.

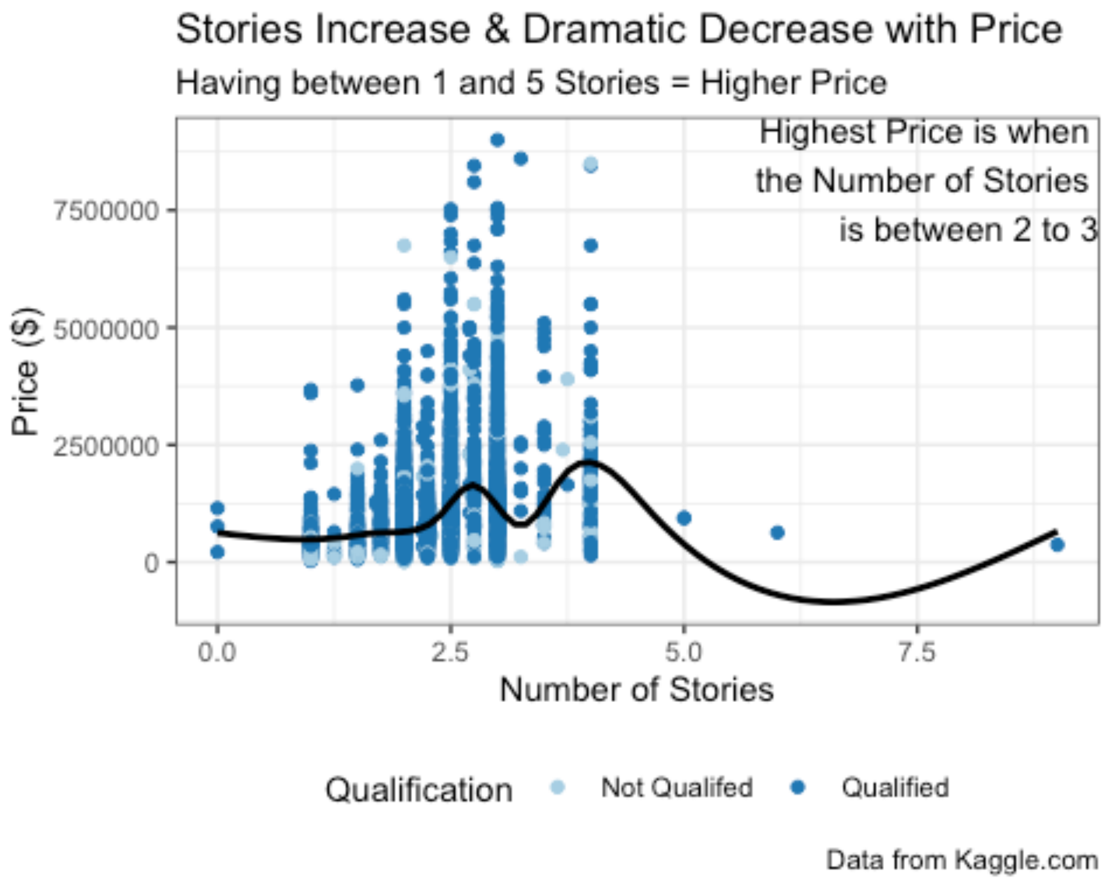
```
text_df <- tibble(text = "More Bedrooms equals\nLower Price & Less
Qualified", x = -Inf, y = Inf)
ggplot(dcproperty, aes(BEDRM, PRICE)) +
  geom_point(aes(color = factor(QUALIFIED_2, labels = c("Not Qualifed",
"Qualified")))) +
  geom_smooth(se = FALSE, color = "black") +
  labs(title = "Bedrooms Increase & Decrease with Price",
       subtitle = "Having Fewer Kitchens Increases the Price",
       caption = "Data from Kaggle.com",
       y = "Price ($)",
       x = "Number of Bedrooms",
       color = "Qualification") +
  geom_text(aes(x, y, label = text), data = text_df, vjust = "top", hjust =
"left") +
  scale_colour_brewer(palette = "Paired") +
  theme_bw() +
  theme(legend.position = "bottom")
```



It seems that the more bedrooms you have will make the price increase until you have 9 bedrooms that is. If you look closely enough one can see that the price trend drops from studio (no bedrooms) and 2 bedrooms. I thought that the price trend would have been always increasing but it semes from graph my logic and guess was wrong. However we can see that at the beginnning and end our confidence interval starts to increase while in the middle the confidence interval we had was the same as the line.

```
text_df <- tibble(text = "Highest Price is when \n the Number of Stories \n
is between 2 to 3", x = Inf, y = Inf)
ggplot(dcproperty, aes(STORIES, PRICE)) +
  geom_point(aes(color = factor(QUALIFIED_2, labels = c("Not Qualifed",
"Qualified")))) +
  geom_smooth(se = FALSE, color = "black") +
  labs(title = "Stories Steep Increase & Dramatic Decrease on Price",
       subtitle = "Having between 1 and 5 Stories = Higher Price",
       caption = "Data from Kaggle.com",
       y = "Price ($)",
       x = "Number of Stories",
       color = "Qualification") +
  geom_text(aes(x, y, label = text), data = text_df, vjust = "top", hjust =
"right") +
  scale_colour_brewer(palette = "Paired") +
```

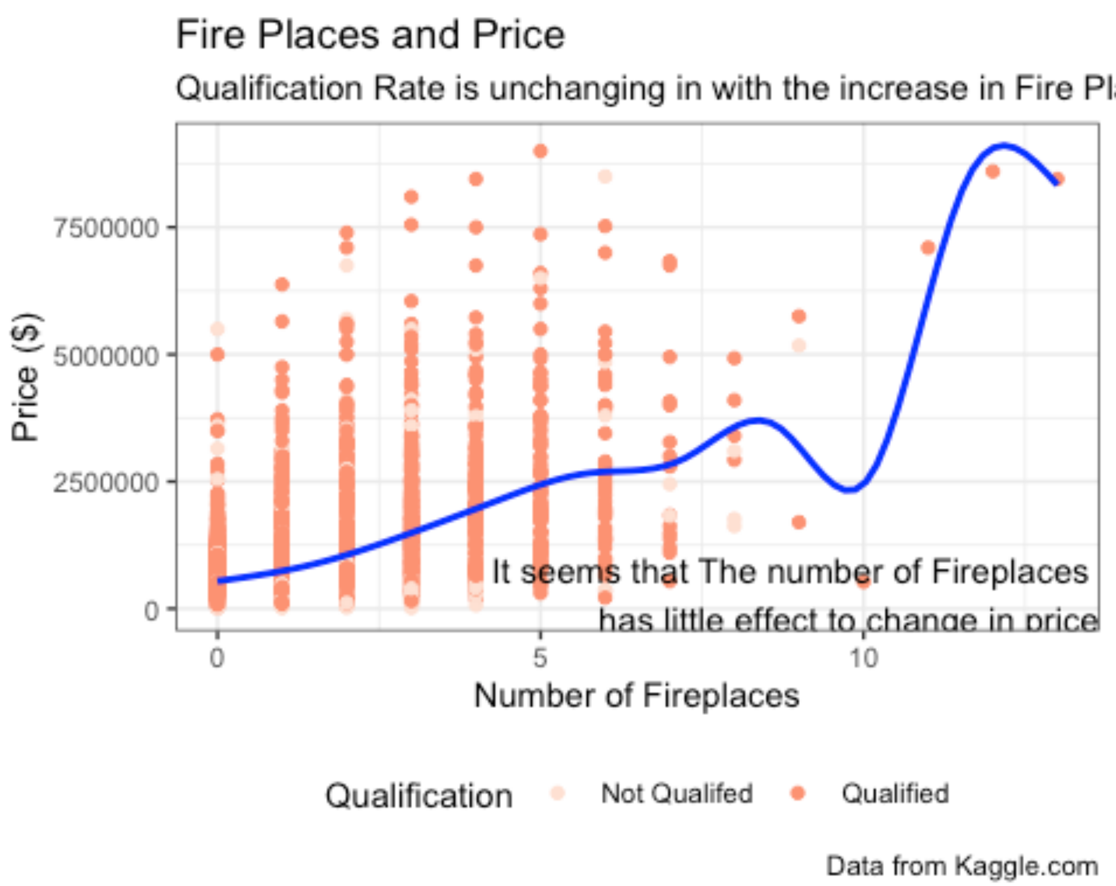
```
  theme_bw() +
  theme(legend.position = "bottom")
```



It is interesting how the price trend line is increasing and decreasing throughout this graph. It seems that people like to have property that hs few floors but once you leave in a building the the price can change in many ways. Also it is interesting that how the middle (1-4 stories) has an almost eqla distributions of qualfieid and unqualified properties.

```
text_df <- tibble(text = "It seems that The number of Fireplaces \n has
little effect to change in price", x = Inf, y = -Inf)
ggplot(dcproperty, aes(FIREPLACES, PRICE)) +
  geom_point(aes(color = factor(QUALIFIED_2, labels = c("Not Qualifed",
"Qualified")))) +
  geom_smooth(se = FALSE, color = "blue") +
  labs(title = "Fire Places and Price",
       subtitle = "Qualification Rate is unchanging in with the increase in
Fire Places",
       caption = "Data from Kaggle.com",
       y = "Price ($)",
       x = "Number of Fireplaces",
       color = "Qualification") +
  geom_text(aes(x, y, label = text), data = text_df, vjust = "bottom", hjust
= "right") +
```

```
  scale_colour_brewer(palette = "Reds") +
  theme_bw() +
  theme(legend.position = "bottom")
```

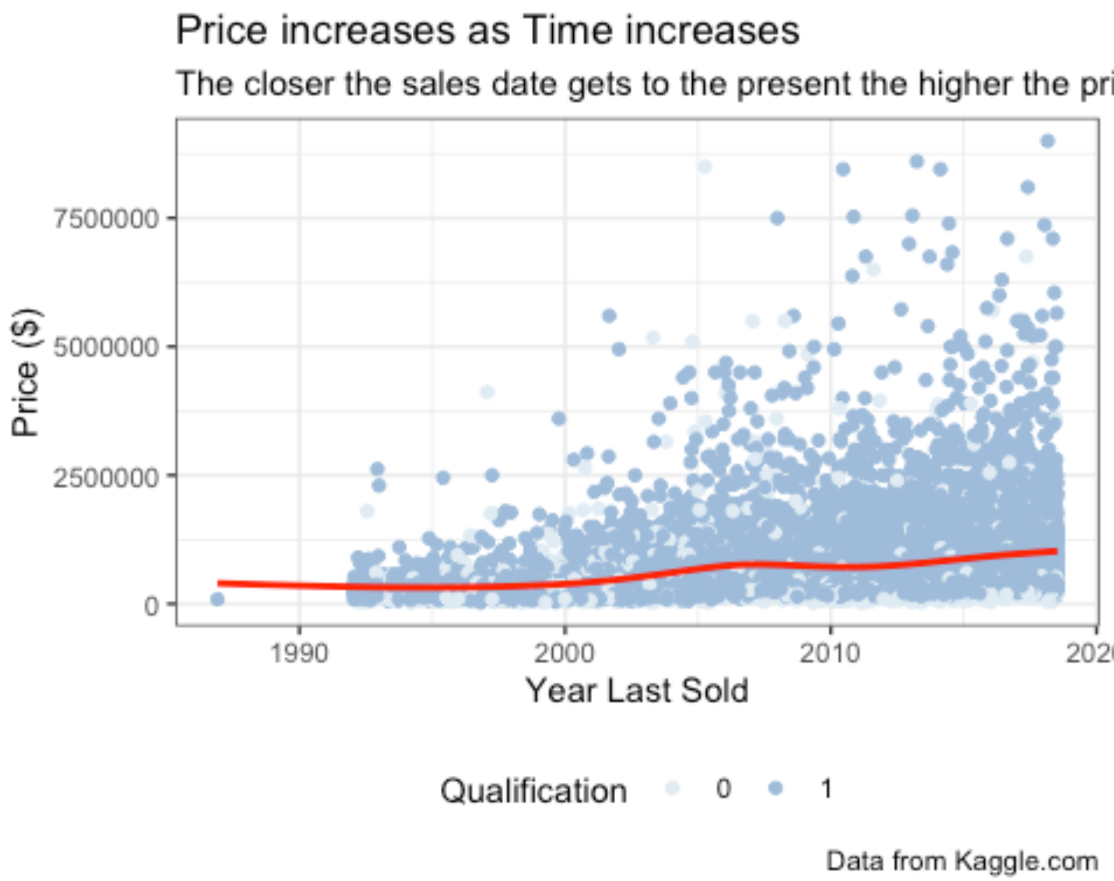


This is the only graph in my analysis that is confusing and has a high upward sloping curve for half of the graph. I say that this is confusing because I can not believe someone would want more then one fireplace and that the price increases for every additional fireplace. (I choose the colors to be similar to fire.) Next time I have to set the number of fireplaces to greater then or 8.

Time Graphs

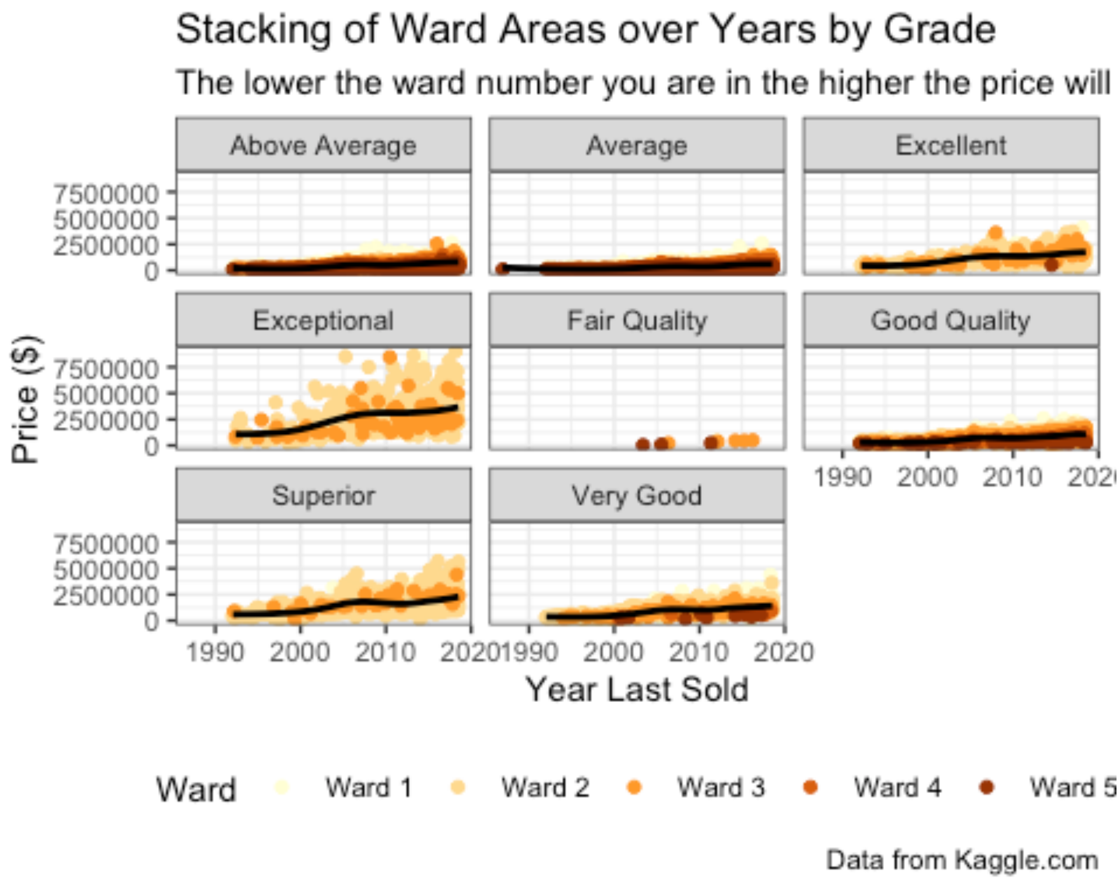
```
ggplot(dcproperty, aes(SALEDATE, PRICE)) +
  geom_point(aes(color = as.factor(QUALIFIED_2))) +
  geom_smooth(se = FALSE, color = "red") +
  labs(title = "Price increases as Time increases",
       subtitle = "The closer the sales date gets to the present the higher
the price",
       caption = "Data from Kaggle.com",
       y = "Price ($)",
       x = "Year Last Sold",
       color = "Qualification") +
```

```
  scale_colour_brewer(palette = "BuPu") +
  theme_bw() +
  theme(legend.position = "bottom")
```



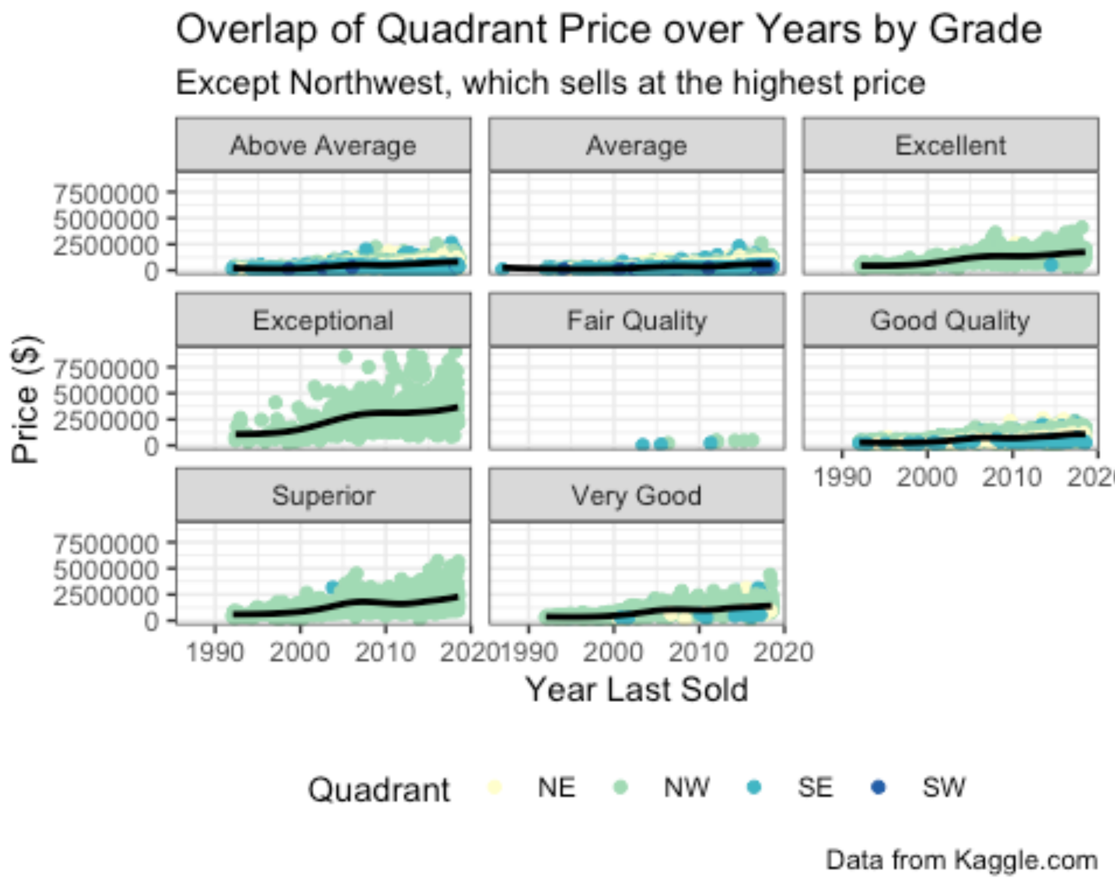
It seems that the closer one gets to the presense in property sales the higher the price will be. Makes perfect sense to me. The trend line not increasing shaprly can be explained if one checks inflation rate of each year.

```
ggplot(dcproperty, aes(SALEDATE, PRICE)) +
  geom_point(aes(color = as.factor(WARD))) +
  geom_smooth(se = FALSE, color = "black") +
  labs(title = "Stacking of Ward Areas over Years by Grade",
       subtitle = "The lower the ward number you are in the higher the price
will be",
       caption = "Data from Kaggle.com",
       y = "Price ($)",
       x = "Year Last Sold",
       color = "Ward") +
  scale_colour_brewer(palette = "YlOrBr") +
  facet_wrap(~GRADE) +
  theme_bw() +
  theme(legend.position = "bottom")
```



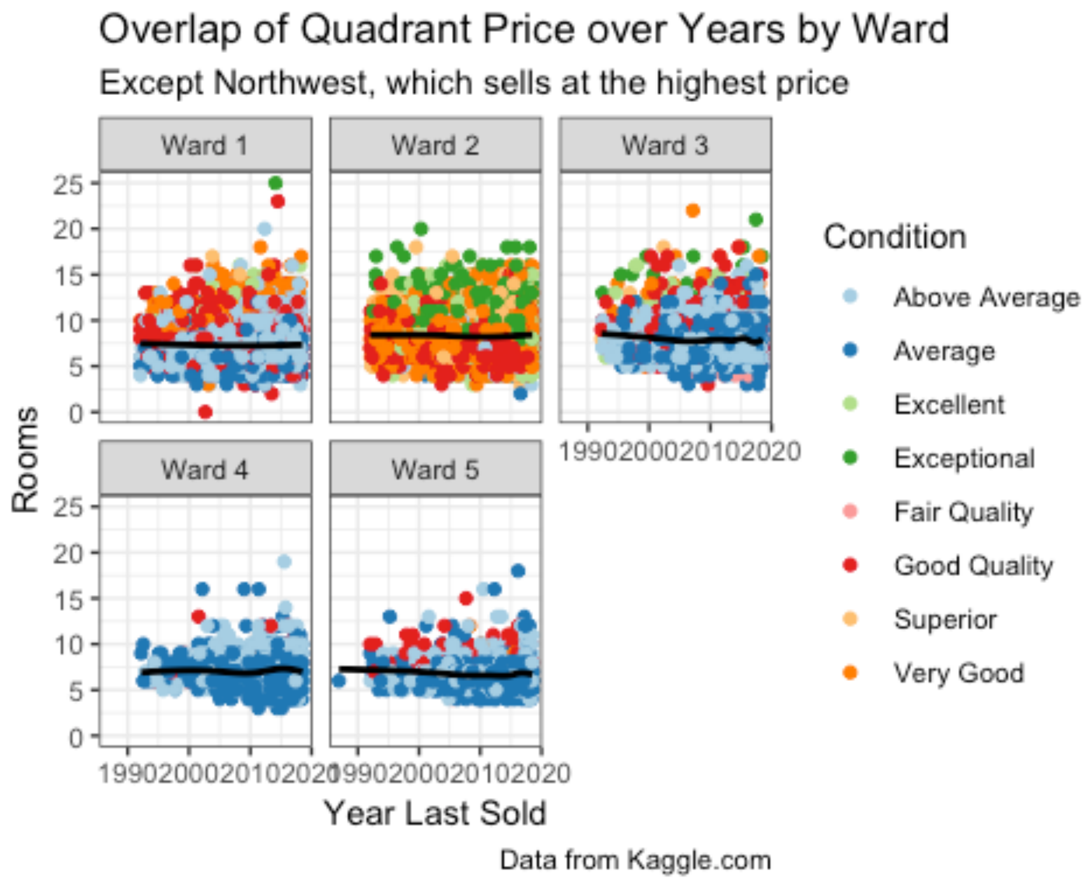
This is an interesting graph to look and observe. The Ward numbers are almost stacked one on top of the other in eveyr grade category except for aver and above average. The better the grade a property recieved the higher the price will be which makes perfect sense.

```
ggplot(dcproperty, aes(SALEDATE, PRICE)) +
  geom_point(aes(color = as.factor(QUADRANT))) +
  geom_smooth(se = FALSE, color = "black") +
  labs(title = "Overlap of Quadrant Price over Years by Grade",
       subtitle = "Except Northwest, which sells at the highest price",
       caption = "Data from Kaggle.com",
       y = "Price ($)",
       x = "Year Last Sold",
       color = "Quadrant") +
  facet_wrap(~GRADE) +
  scale_colour_brewer(palette = "YlGnBu") +
  theme_bw() +
  theme(legend.position = "bottom")
```



This graph has a similar outlook to the graph above. It seems taht only the northwest region was able to get the highest conditin reviews. We can see like the graph above there is some stacking in good quality, above average, and average but not so much in very good or fair.It is interesting though the shape of the data points for very good, above, average, good quality, and excellent are almost identical but only excellent is dominated by the northwest region.

```
ggplot(dcproperty, aes(SALEDATE, ROOMS)) +
  geom_point(aes(color = as.factor(GRADE))) +
  geom_smooth(se = FALSE, color = "black") +
  labs(title = "Overlap of Quadrant Price over Years by Ward",
       subtitle = "Except Northwest, which sells at the highest price",
       caption = "Data from Kaggle.com",
       y = "Rooms",
       x = "Year Last Sold",
       color = "Condition") +
  scale_colour_brewer(palette = "Paired") +
  facet_wrap(~WARD) +
  theme_bw() +
  theme(legend.position = "right")
```



I like to call this graph the color mess. It seems that the shape of the top ward (Ward 1-3) and the bottom wards (Wards 4-5) are similar between themselves but the top and bottom are not. Also Wards 1-3 have a a lot more color then Wards 4,5 do. It seems that rooms started to increase as we get closer to the present. THis makes sense considering people lived within their means back in the 20th century.

Conclusion

To conclude we have created a binary logistic model for what determines whether a property is qualified enough to sell. Although the AIC is very high as 16,748, it seems that the model fits that data well. It was a long analysis process but we could complete it even with the time restrictions we had. Now we will answer the 7 questions we had at the beginning of this study. The answers to our seven questions were as follows: 1) We were not able to hear back from Chris, the provider of this dataset on Kaggle, we so can still not answer what the qualification column in the original dataset means. 2) The qualifications for a residential property to be sold on the market is that the paperwork is completed and submitted, the bank approves any transaction that the buyers and sellers need and the inspection of the property is passed. 3) From all our analysis so far, we can conclude that property pricing is the most important factor in determining whether a property is qualified to go on the market 4) From all our analysis so far, we can conclude realtors do care whether the property is qualified to sell and money is the most important to them.

Since the more the realtor sells and the higher the price, the property is sold for the more money from the deal they receive. 4) We believe that we were creating the most optimal regression for modeling properties based on our response variable being qualification. Overall though a multiple linear type regression would work best when it comes to making housing, property, and apartment models. 5) We did follow the previous housing model approaches for predictor variables at the beginning of our model building but our analysis later had different predictor variables from those models creating using linear regression. 6) Yes, money is the most important thing. We will not say how this defines this world since we do not wish to be labeled as pessimistic people. Thankfully, we could answer our questions based on our analysis results and work, yet this does not mean we will stop the analysis being conducted.

For future analysis, we would do many things differently and add many different types of things. We will do the following things in the future: 1) conduct more time analysis and visualizations, 2) conduct some sentiment analysis on the street, neighborhood, and State one lives in since people are sometimes superstitious, 3) Try to see if we can hear back on what qualification meant in the dataset, 4) Add a few more variables – for example: Heat, and the interaction terms of heat and AC, 5) Collect data from realtor's websites and fill in the information ourselves since we were losing data constantly, 6) Add neighborhood rating, neighborhood review 7) Collect data from the surrounding states (West Virginia, Virginia, Maryland).

In conclusion, through all our analysis and graphs our model might have been able to measure the odds for determining qualifications. This model is nowhere near good enough to be published or presented in a conference. This was a great learning experience for careers even though the model we created is still inferior to a linear regression pricing model since we believe that money is the most important to realtors and people when it comes to the housing market. If you wish to know more about the data and analysis we have completed visit reference link 1. As a famous person once said, "failure is the mother of success."

References

1. https://aaronniecestro.shinyapps.io/DC-Housing/
2. McKay, Allie W. "Farmers' Markets vs. Food Deserts: Which Are Winning in DC?" The Capital's Markets, 31 July 2014, thecapitalsmarkets.wordpress.com/2014/07/31/farmers-markets-vs-food-deserts-which-is-winning-in-dc/.
3. Johnson, Matt. "Washington's Systemic Streets." Greater Greater Washington, ggwash.org/view/2530/washingtons-systemic-streets.
4. "Money Is The Root Of All Evil Stock Photos and Images." Alamy, www.alamy.com/stock-photo/money-is-the-root-of-all-evil.html.
5. "Types of Housing Models and Programs." The 519, www.the519.org/education-training/lgbtq2s-youth-homelessness-in-canada/types-of-housing-models-and-programs.
6. Dobbins, Tim, and John Burke. "Predicting Housing Prices with Linear Regression Using Python, Pandas, and Statsmodels." Learn Data Science - Tutorials, Books,

Courses, and More, www.learndatasci.com/tutorials/predicting-housing-prices-linear-regression-using-python-pandas-statsmodels/.

7. Corsini, Kenneth Richard. “STATISTICAL ANALYSIS OF RESIDENTIAL HOUSING PRICES IN AN UP AND DOWN REAL ESTATE MARKET: A GENERAL FRAMEWORK AND STUDY OF COBB COUNTY, GA .” A Thesis Presented to The Academic Faculty, Georgia Institute of Technology, Dec. 2009, smartech.gatech.edu/bitstream/handle/1853/31763/Corsini_Kenneth_R_200912_mast.pdf.
8. “Regression Data for Inclusionary Housing Simulation Model | DataSF | City, and County of San Francisco.” San Francisco Data, data.sfgov.org/Economy-and-Community/Regression-data-for-Inclusionary-Housing-Simulatio/vcwn-f2xk/data.
9. Leonard, Kimberlee. “What Forms Are Needed to Sell a Home by Owner?” Home Guides | SF Gate, 29 Dec. 2018, homeguides.sfgate.com/forms-needed-sell-home-owner-7271.html.
10. Leonard, Kimberlee. “What Is the Procedure for Closing a for Sale by Owner House Sale?” Home Guides | SF Gate, 15 Dec. 2018, homeguides.sfgate.com/procedure-closing-sale-owner-house-sale-65511.html.

I worked with Kingsley Iyawe in STAT-616 Generalized Linear Models to complete this project and report. He deserves some credit for this report.