Exploratory Analysis and Price Prediction of Beijing Housing

Jiajun Zhou

UCLA Extension

Data Access

1.1 Project Overview

This class project is designed for us to become productive with the application of our new data science skills into real-life working. Data used for the alternative class project is related to housing prices in Califonia and this is then interesting to take a look at the housing prices in my home country, China since the housing market in China has bloomed for these years. The variation between the home's prices is quite huge among different regions and years. I found one dataset on [Kaggle.com] which contains housing prices of Beijing from 2011 to 2017, fetching from [Lianjia.com] and it contains over 300 thousand transaction records by Lianjia, one of the biggest Chinese real-estate brokerage company founded in 2001.

Therefore, the main objective of this project is to predict the housing price values in Beijing. Despite the over 300 thousand observations in this dataset, I will carry out the tasks outlined that parallel the data science process detailed in the alternative project to balance the difficulties in the following sections:

- Data access: download the data set and load it into the R environment.
- Data Munging: clean the missing values and transform the columns as necessary to select variables that support the hypothesis.
- Data Visualization: use various exploratory data analysis and simple statistical techniques to gain a deep understanding of the data. understanding of the data.
- Supervised Machine Learning: adopt the regression and random forest model to make predictions of housing price values based on the trained algorithm.

1.2 Dataset Description

The dataset records one row per transaction of housing sale in Beijing from the 2011-2017 period scrapped from [Lianjia.com] and shared on [kaggle.com]. Most of the observations are traded in 2011-2017, some of them are traded in Jan 2018, and some are even earlier. It includes the variables URL, ID, Lng., Lat., community ID, trade time, DOM(Days on Market), followers, total price, price, square, number of living room, drawing room, kitchen and bathroom, building type, construction time, renovation condition, building structure, ladder ratio, elevator, property rights for five years, subway, district, and community average price.

Since mapping the data on the map of Beijing is a big challenge, I removed the columns of latitude and longitude when accessing the data. All the other useless columns: URL, ID, and followers are not loaded, either. Therefore, there are 318851 observations of 21 variables loaded and the complete description of the features considered in this dataset are:

- `Cid`: community ID;
- `DOM`: active days on market;
- `tradeTime`: the date of the transaction;
- `totalPrice`: the final price of the house (10,000¥);
- `price`: price per square meter of housing;
- `square`: the square meter of the house;
- `livingRoom`: the number of the living room (Supposed to be bedroom after I checked in Chinese);
- `drawingRoom`: the number of drawing-room (Supposed to be living room after I checked in Chinese);
- `kitchen`: the number of the kitchen;
- `bathroom` the number of the bathroom;
- `floor`: the location of the house in the building and the floor number of the housing;
- `buildingType`: the type of building including (1) tower, (2) bungalow, (3)combination of plate and tower, (4)plate;
- `constructionTime`: the year of building constructed;

- `renovationCondition`: the condition of renovation including (1)other, (2)rough, (3)Simplicity, (4)hardcover;
- `buildingStructure`: the building structure including (1)unknown, (2)mixed, (3)brick and wood, (4)brick and concrete, (5)steel, and (6)steel-concrete composite;
- `ladderRatio`: the proportion between the number of residents on the same floor and number of the elevator of a ladder. It describes how many ladders a resident has on average;
- `elevator` whether the housing (1) have or (0) not have an elevator;
- `fiveYearsProperty`: whether the owner has the property for (1) less than or (0) more than 5 years (It's related to China restricted the purchase of houses policy);
- `subway` : whether the housing is (1) close to subway or (0) not
- `district` :(1)"DongCheng",(2)"FengTai",(3)"Yizhuang", (4)"DaXing", (5)"FangShang", (6)"ChangPing",(7)"ChaoYang",(8)"HaiDian",(9)"ShiJingShan",(10)"XiCheng",(11)"Tong Zhou",(12)"ShunYi",(13)"MenTouGou"
- `commuityAverage` : the average price per square meter in the corresponding community;

Data Munging

In this section, I first performed a 'summary()' function on the data frame to display the data class, range of values for numeric variables, and levels for any factor variable.

```
summary(df)
```

```
##       Cid              tradeTime              DOM         
##  Min.   :1.111e+12   Length:318851      Min.   :   1.00  
##  1st Qu.:1.111e+12   Class :character   1st Qu.:   1.00  
##  Median :1.111e+12   Mode  :character   Median :   6.00  
##  Mean   :1.129e+12                      Mean   :  28.82  
##  3rd Qu.:1.111e+12                      3rd Qu.:  37.00  
##  Max.   :1.115e+15                      Max.   :1677.00  
##                                         NA's   :157977   
##    totalPrice          price             square         livingRoom       
##  Min.   :    0.1   Min.   :     1   Min.   :   6.90   Length:318851     
```

```
##  1st Qu.:  205.0   1st Qu.: 28050   1st Qu.:  57.90   Class :character
##  Median :  294.0   Median : 38737   Median :  74.26   Mode  :character
##  Mean   :  349.0   Mean   : 43530   Mean   :  83.24
##  3rd Qu.:  425.5   3rd Qu.: 53820   3rd Qu.:  98.71
##  Max.   :18130.0   Max.   :156250   Max.   :1745.50
##
##  drawingRoom           kitchen         bathRoom            floor
##  Length:318851      Min.   :0.0000   Length:318851      Length:318851
##  Class :character   1st Qu.:1.0000   Class :character   Class :character
##  Mode  :character   Median :1.0000   Mode  :character   Mode  :character
##                     Mean   :0.9946
##                     3rd Qu.:1.0000
##                     Max.   :4.0000
##
##   buildingType   constructionTime   renovationCondition buildingStructure
##  Min.   :0.048   Length:318851      Min.   :0.000       Min.   :0.000
##  1st Qu.:1.000   Class :character   1st Qu.:1.000       1st Qu.:2.000
##  Median :4.000   Mode  :character   Median :3.000       Median :6.000
##  Mean   :3.010                      Mean   :2.606       Mean   :4.451
##  3rd Qu.:4.000                      3rd Qu.:4.000       3rd Qu.:6.000
##  Max.   :4.000                      Max.   :4.000       Max.   :6.000
##  NA's   :2021
##   ladderRatio          elevator     fiveYearsProperty     subway
##  Min.   :       0   Min.   :0.000   Min.   :0.0000    Min.   :0.0000
##  1st Qu.:       0   1st Qu.:0.000   1st Qu.:0.0000    1st Qu.:0.0000
##  Median :       0   Median :1.000   Median :1.0000    Median :1.0000
##  Mean   :      63   Mean   :0.577   Mean   :0.6456    Mean   :0.6011
##  3rd Qu.:       0   3rd Qu.:1.000   3rd Qu.:1.0000    3rd Qu.:1.0000
##  Max.   :10009400   Max.   :1.000   Max.   :1.0000    Max.   :1.0000
##                     NA's   :32      NA's   :32        NA's   :32
##     district      communityAverage
##  Min.   : 1.000   Min.   : 10847
##  1st Qu.: 6.000   1st Qu.: 46339
```

```
##  Median : 7.000   Median : 59015
##  Mean   : 6.764   Mean   : 63682
##  3rd Qu.: 8.000   3rd Qu.: 75950
##  Max.   :13.000   Max.   :183109
##                   NA's   :463
```

2.1 Cleaning the missing values

From the summary output, I found that the columns of `DOM`,`buildingType`,`elevator`,`fiveYearsProperty`,`subway`, and `communityAverage` have NAs. The biggest problem is that nearly half of the days on market are NA. The distribution of `DOM` shows that its minimum is 1 day on the market and 50% of the transactions staying under 6 days. It suggests that the transaction in Beijing is very fast-paced and thus, I consider that its missing value to supposed to be the value of 0 instead. To clean the missing values of `buildingType` then, I classify the NAs as the unknown group.

```
df$DOM[is.na(df$DOM)]<-0
df$buildingType[is.na(df$buildingType)]<-5
df$ladderRatio[df$ladderRatio==10009400] <- 1.0009400
nrow(df[!complete.cases(df),])

## [1] 495
```

I finally got 495 rows of observations with NAs by performing the above function and I removed all these observations by performing a 'na.omit()' function since the little proportion of the observations would not affect the data largely. On the other hand, I found the maximum of the ladder ratio is 10009400, a huge outlier. I replaced the number with 1.0009400 after I inspected the original webpage. After cleaning the missing values, there are 318356 observations in this dataset.

2.2 Transforming the Numeric Variables

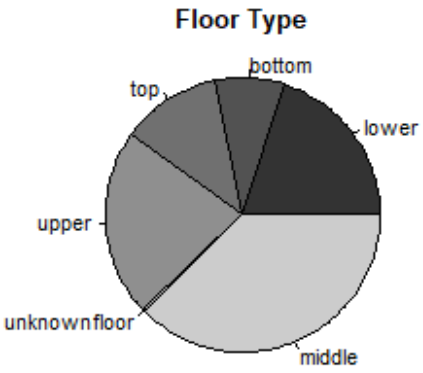
From the summary output, I found that there are many columns with character supposed to be numeric variables. I converted them by performing `as.numeric` function. Besides, I added several new variables such as the age of the housing by the substruction of the construction year

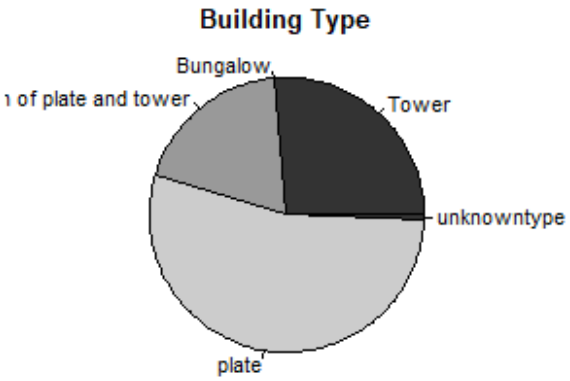
from the trading year. When converting the variable of `constructionTime`, I found 19283 rows of observations with "未知", the meaning of unknowns in Chinese and I imputed these missing values with the median of the rest numbers. Besides, I split the variable of `floor` as the location of the house floor in the building and the floor number of the housing. The types of the location of the house are at the bottom, lower, middle, upper, top and unknown floor of the building. After the transformation, I got 11 numeric variables and the descriptive statistics of these variables are shown in the below table.

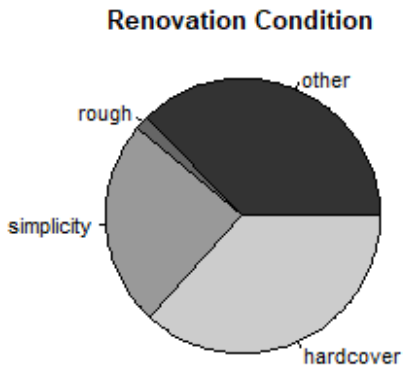
	vars	n	mean	sd	min	max	range	se
price	1	318356	43495.409	21644.037	1.000	156250.0	156249.000	38.360
totalPrice	2	318356	349.159	230.731	0.100	18130.0	18129.900	0.409
square	3	318356	83.283	37.184	7.370	1745.5	1738.130	0.066
floor	4	318356	13.309	7.823	1.000	63.0	62.000	0.014
DOM	5	318356	14.550	38.494	0.000	1677.0	1677.000	0.068
age	6	318356	15.455	8.607	-7.000	84.0	91.000	0.015
livingRoom	7	318356	2.011	0.777	0.000	9.0	9.000	0.001
drawingRoom	8	318356	1.173	0.522	0.000	5.0	5.000	0.001
kitchen	9	318356	0.995	0.104	0.000	4.0	4.000	0.000
bathRoom	10	318356	1.189	0.437	0.000	7.0	7.000	0.001
ladderRatio	11	318356	0.381	0.178	0.014	10.0	9.986	0.000
communityAverage	12	318356	63684.227	22328.807	10847.000	183109.0	172262.000	39.574

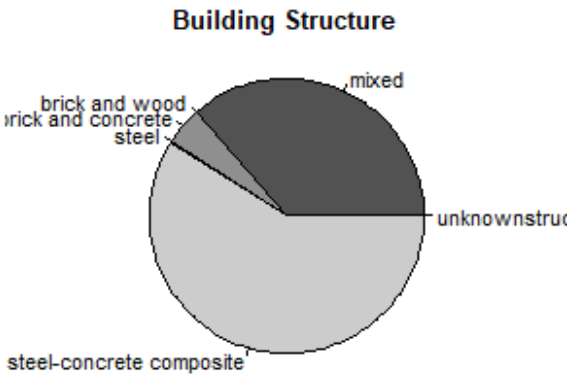
2.3 Relabeling the Categorial Variables

In this section, I converted the rest of categorical variables into factors labeled in the data description and I drew multiple pie charts to get a sense of the distribution of these factors. There are 6 levels in the location of floor type, 5 levels of the type of building, 4 levels in the renovation condition, 6 levels in the types of the building structure, 2 levels in the `elevator`, `subway` and `fiveYearsProperty` and 13 levers in the district in Beijing.


Floor Type
bottom
top
lower
upper
unknownfloor
middle


Building Type
Bungalow
Tower
unknowntype
plate

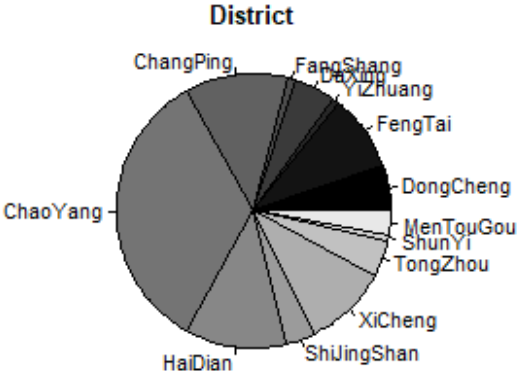

Renovation Condition
other
rough
simplicity
hardcover


Building Structure
mixed
brick and wood
steel
unknownstruc
steel-concrete composite


Elevator
without elevator
with elevator


Five Years Property
no less than 5 years
less than 5 years


Subway
no subway
close to subway

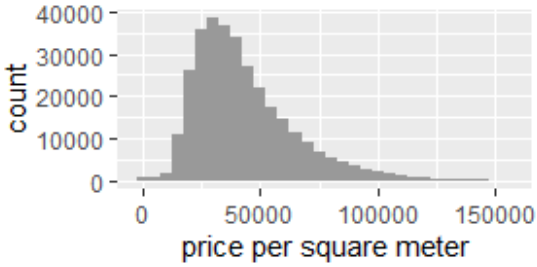

District
ChangPing
FangShang
YiZhuang
FengTai
DongCheng
ChaoYang
MenTouGou
ShunYi
TongZhou
XiCheng
HaiDian
ShiJingShan

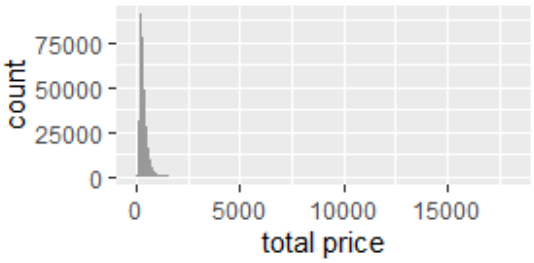
Data Visualization

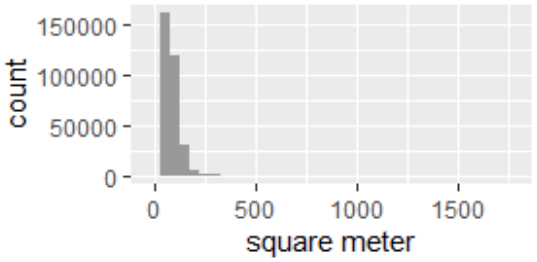
After the transformation of the dataset, I got 12 numeric and 10 well-labeled categorial variables. In the following section, I did some exploratory data analysis (EDA) to gain a deeper understanding of the datasets. I drew multiple histograms to observe the distribution and a corrplot to check the correlation between the numeric variables.I made the plots of the relationship between housing prices and other numeric variables and boxplots of the effect of factors on housing prices.

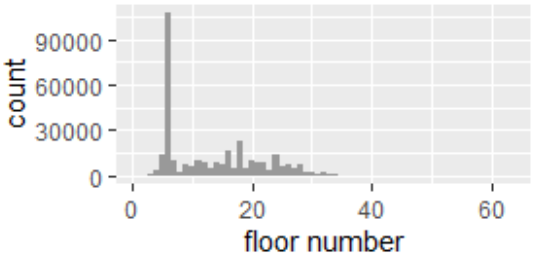
3.1 Histograms for each numeric variable

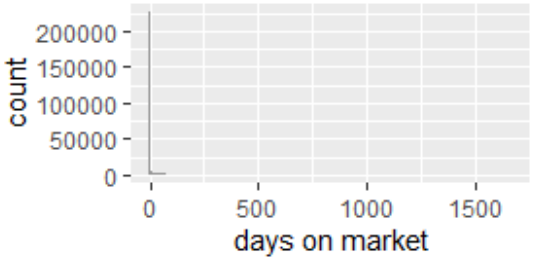
Histograms for each numeric variable could be shown to help us get a more intuitive sense of the distributions. From the below figure, it shows that the distributions of variables `price`, `house age` and `community mean` are much closer to the "bell shape", but the distribution is all right-skewed, while other variables have extremely long tails. Therefore, transformation is needed to make it appear normal when predicting the housing price. The distribution of variables `living room`, `drawing room`,`kitchen`,`bathroom` are more similar to the one of the ordinal variables and it is questionable that these variables are converted into the factors with an order. I treated these variables as numeric variables in the prediction model like in the alternative project despite the issue.

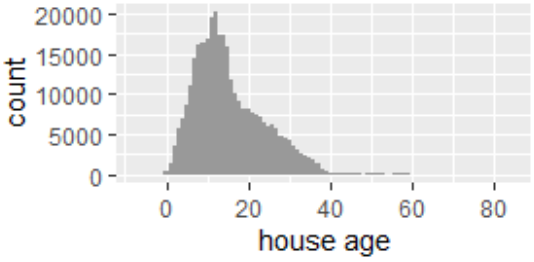

count
40000
30000
20000
10000
0
0
50000
100000
150000
price per square meter

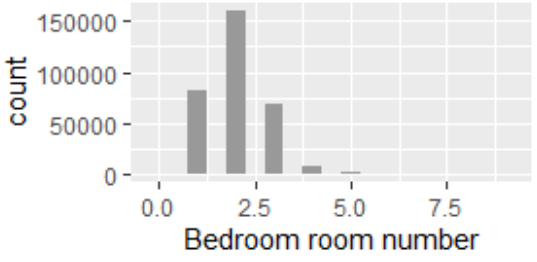

count
75000
50000
25000
0
0
5000
10000
15000
total price

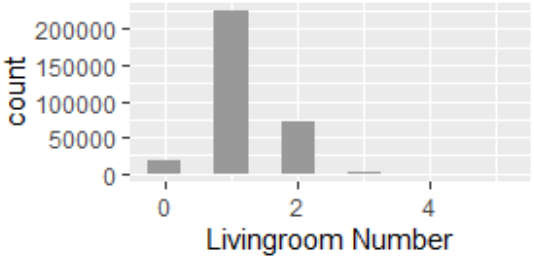

count
150000
100000
50000
0
0
500
1000
1500
square meter


count
90000
60000
30000
0
0
20
40
60
floor number

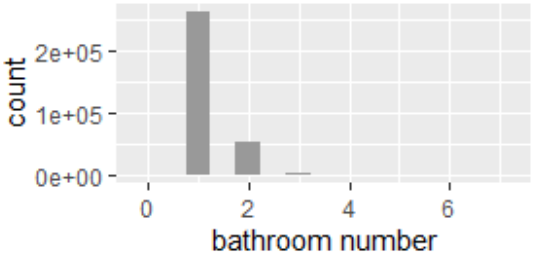

count
200000
150000
100000
50000
0
0
500
1000
1500
days on market

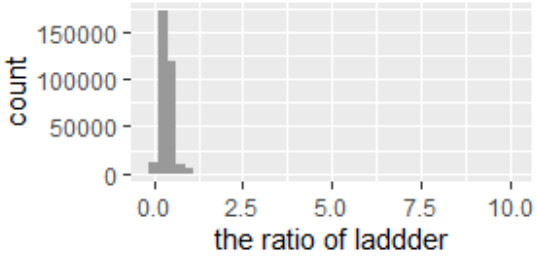

count
20000
15000
10000
5000
0
0
20
40
60
80
house age

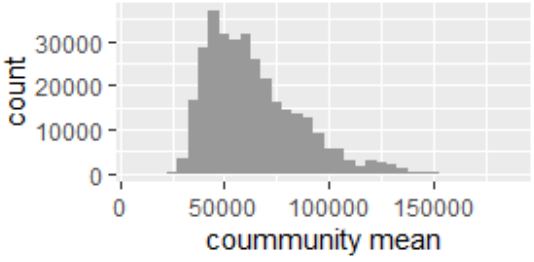

count
150000
100000
50000
0
0.0
2.5
5.0
7.5
Bedroom room number


count
200000
150000
100000
50000
0
0
2
4
Livingroom Number

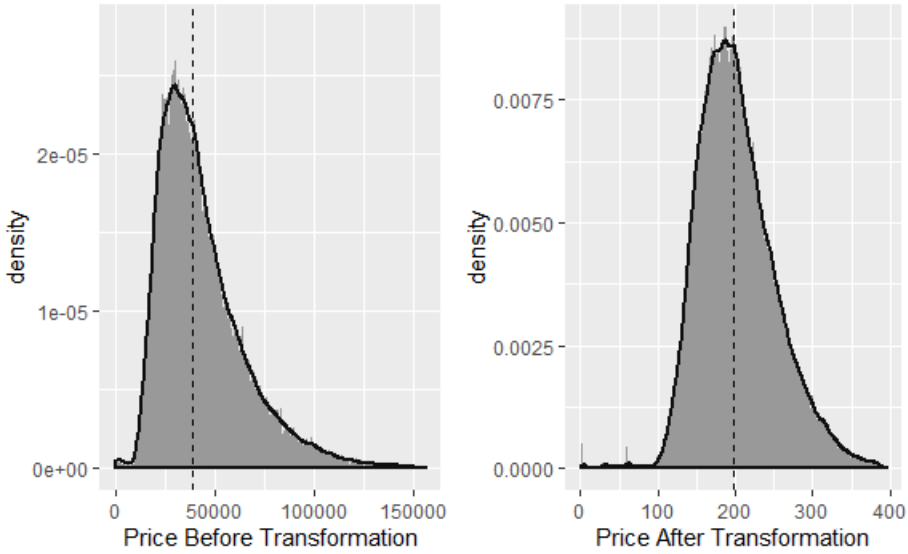

count
3e+05
2e+05
1e+05
0e+00
0
1
2
3
4
kitchen number


count
2e+05
1e+05
0e+00
0
2
4
6
bathroom number


count
150000
100000
50000
0
0.0
2.5
5.0
7.5
10.0
the ratio of laddder

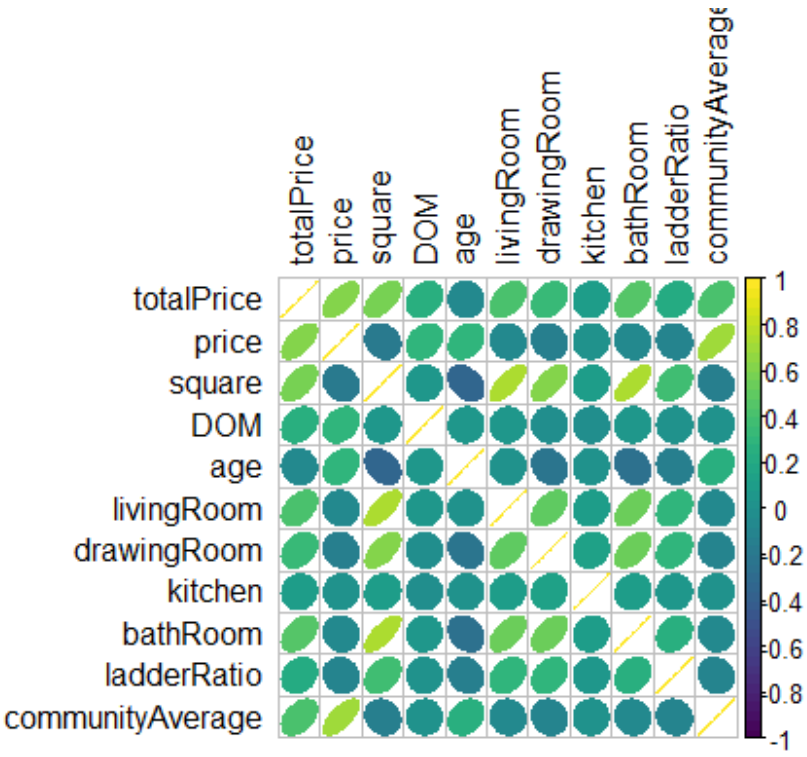

count
30000
20000
10000
0
0
50000
100000
150000
coummunity mean

Since the rest of the variables have a long tail, the square root transformation by performing `sqrt()` is applied for the data. I took the column of `price` as an example and it is shown the square root transformation makes the distribution of the data appear more normal somewhat in the following figure. Therefore, it could be a compromise to take advantage of square root transformation to generate new variables in further analysis.



3.2 Corrplot for the numeric variables

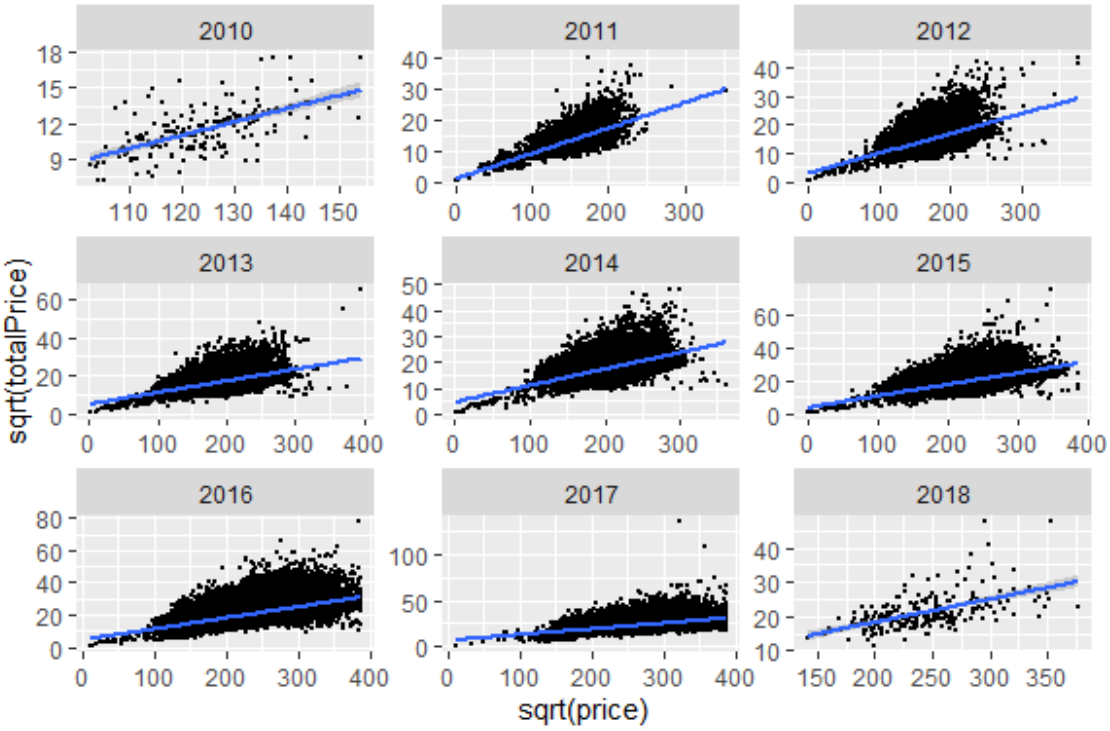
The corrplot below indicates the correlation between the numeric variables. The positive relationship is stronger as the color becomes green while the negative relationship is stronger as the color becomes dark blue. The round shape corresponds to the little relationship between variables. There are several interesting findings:



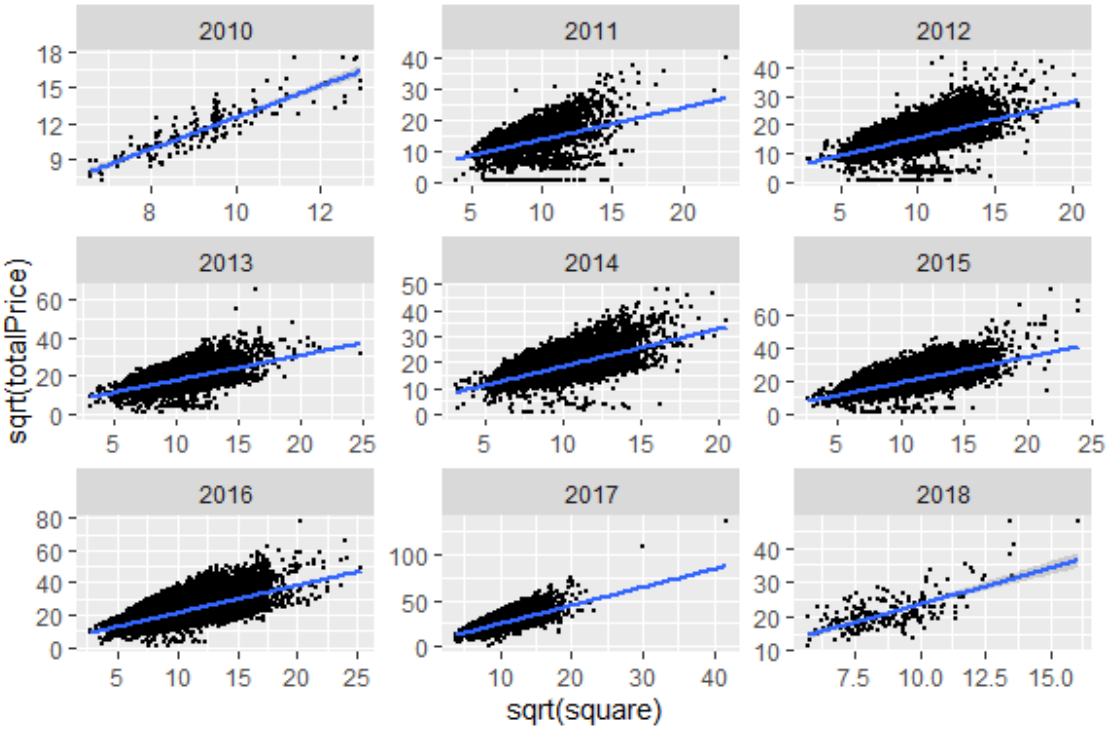
- `price` has a strong positive correlation with `totalPrice` and `communityAverage` and it has some positive correlation with the date on the market and the age of the house. It is a little surprising that the older house has higher price values per square, like the antique.
- The correlation of `totalPrice` with the other variables is much higher than the rest of the variables and it would be better to predict the total price of the house instead of the price per square of the house.
- `square` variable has some positive correlation with `totalPrice` while some negative correlation with `price`. It suggests that the bigger house tends to have a lower price value per square.

3.3 Scatterplot for the price and other variables

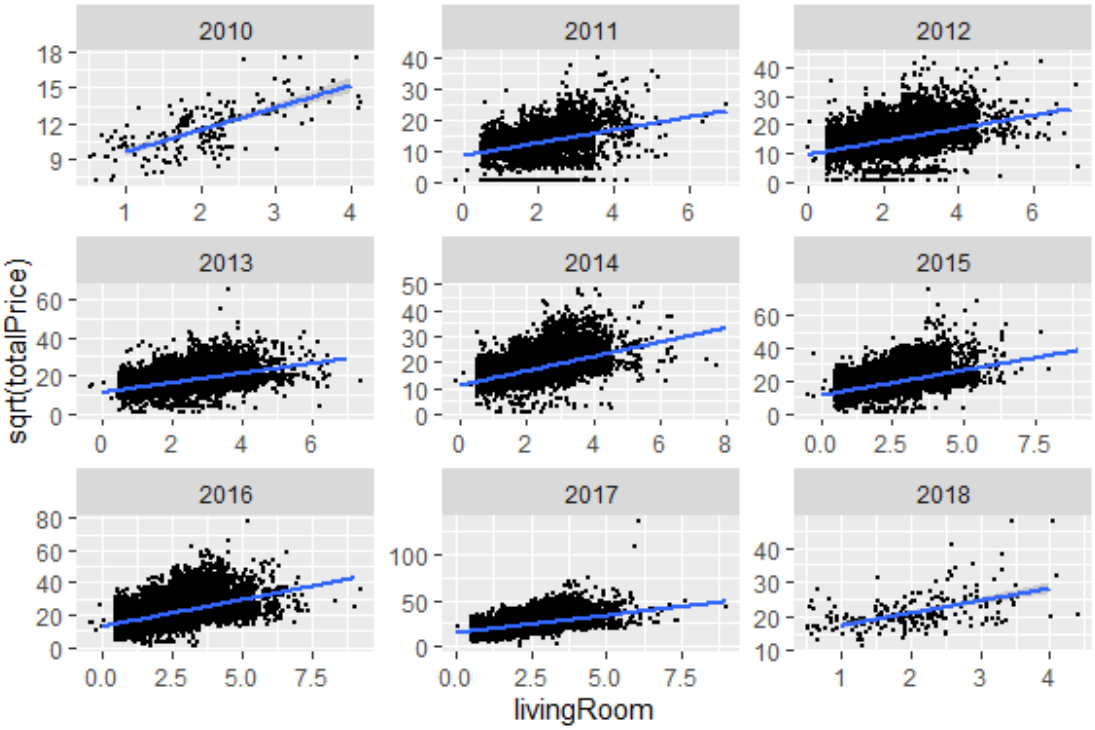
To explore the further relationship of these numeric variables, I plotted several meaningful pairs of relationships throughout the trading years excluding data before 2010 and applied the transformation mentioned above into the used the data. The scatterplots are divided by every year because of the high density of the data and it also helps us check the yearly trend of the total price of the house.



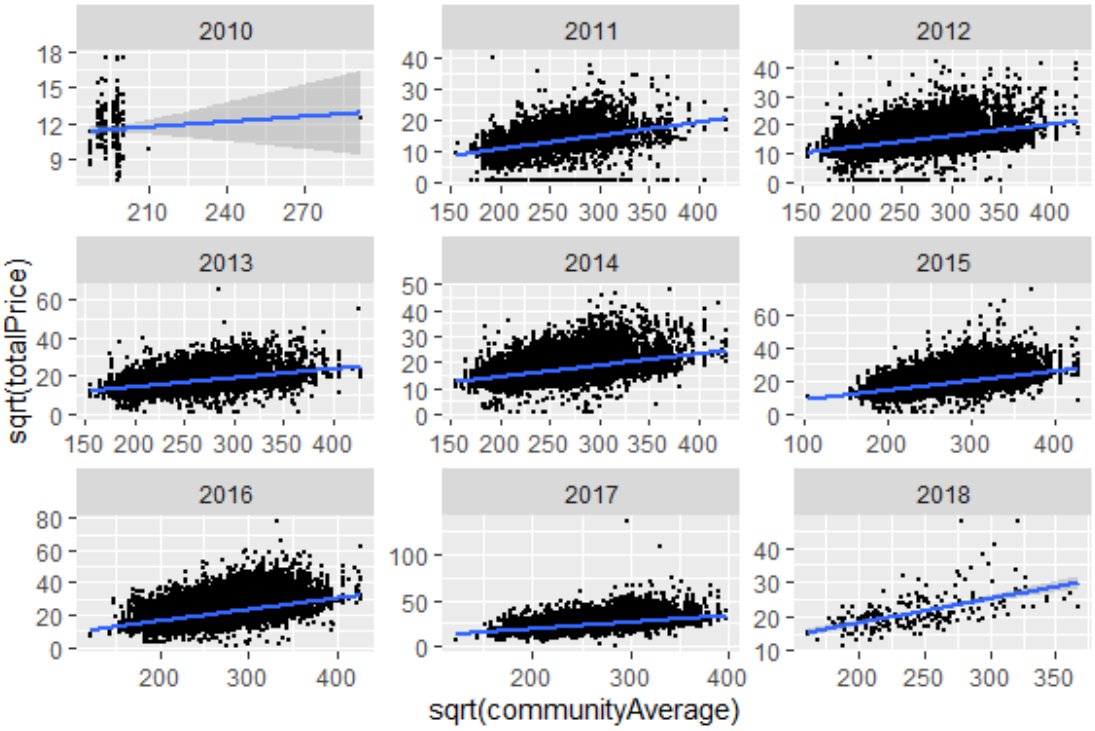
The above scatterplots demonstrate the relationship between the transformed `price` and `totalPrice` and the trend line indicates the positive relationship between them. The points are more concentrated in the area with a higher price and total price meaning the houses in Beijing are very expensive.



The above scatterplots demonstrate the relationship between the transformed `square` and `totalPrice` and the trend line indicates the positive relationship between them. The houses above 200 square meters are a relatively small proportion but much more expensive and thus some of the values are likely to be outliers in the model of predicting the total price of the house. The expanding of the limit on the y-axis shows indirectly that the rise of the total price throughout the nine years.



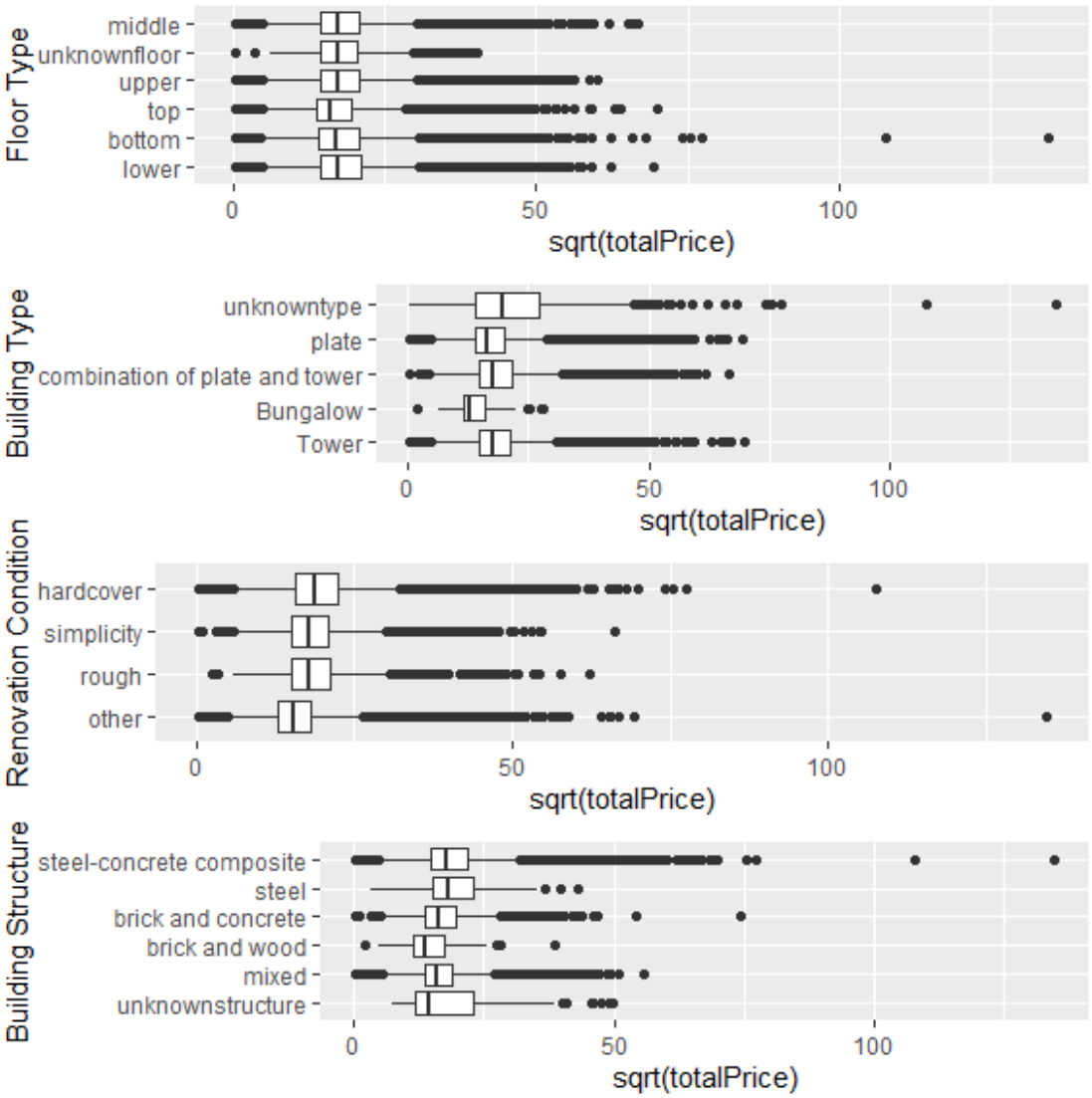
The above scatterplots demonstrate the relationship between the number of the bedroom and transformed `totalPrice` and the trend line indicates the positive relationship between them. It is obvious that the more rooms the house has, the bigger it is, the more expensive it is. The relationship of the `totalPrice` with the number of the living room and bathroom is similar to the above one.



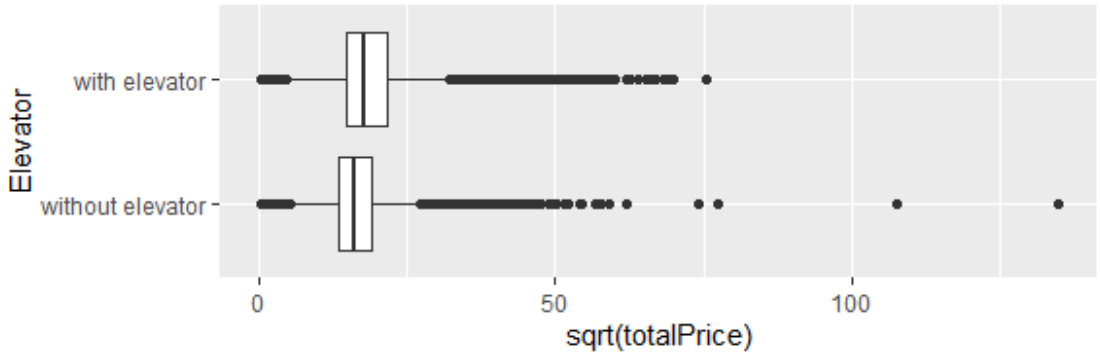
The above scatterplots demonstrate the relationship between the average of community price and `totalPrice` and the trend line indicates the positive relationship between them. Since the scatterplots all shows some positive relationship, it would be useful to consider these variable as the predictor in the building the model.

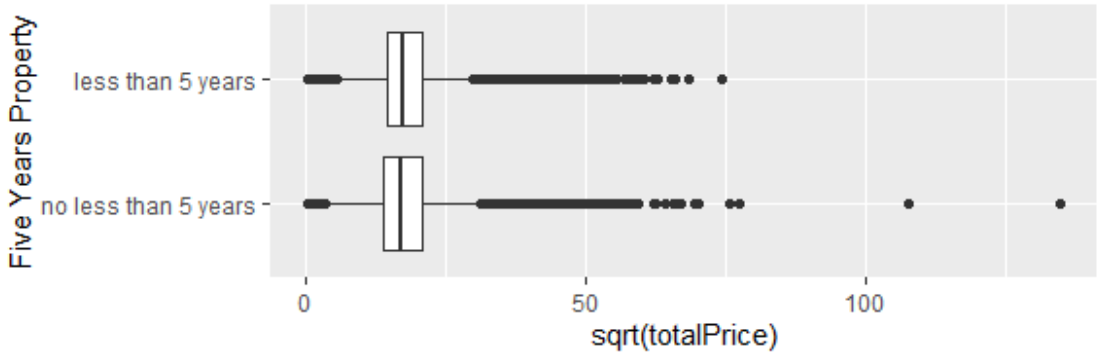
3.4 Boxplots for each Categorial Variables

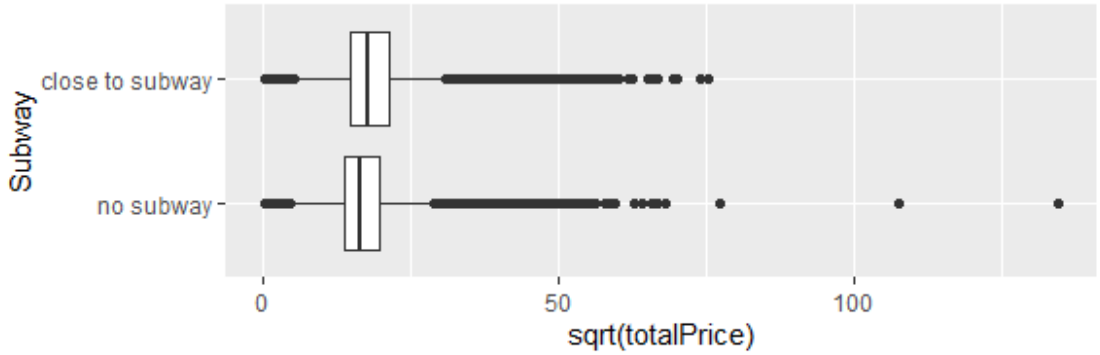
To examine the effects of all the factors on the total price of the house, I drew the eight boxplots below to show the distribution of square-rooted `totalPrice` by each level of the factor. I found that there are still many outliers in the distributions and the majority of them are right-skewed.

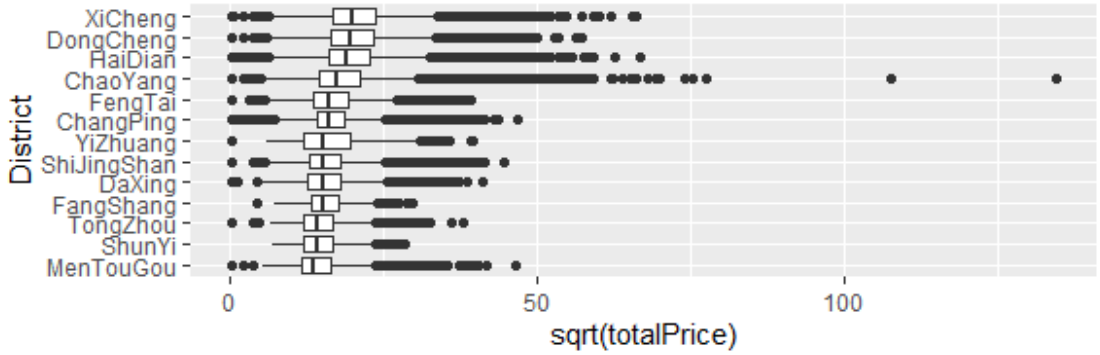


The above boxplots examined the effect of the `Floortype`,`BuildingType`,`renovationCondition`,and `buildingStructure` on the total price of the house. There is a mean difference within these four factors and I found that the houses at the top of the building are the least expensive among the other location of the house. The houses in the bungalow building type and the housed I the building made with brick and wood are the cheapest among the other categories. It makes sense that most of the bungalow building is made with brick and wood. Hardcover-renovated housing is the most expensive among the other categories.





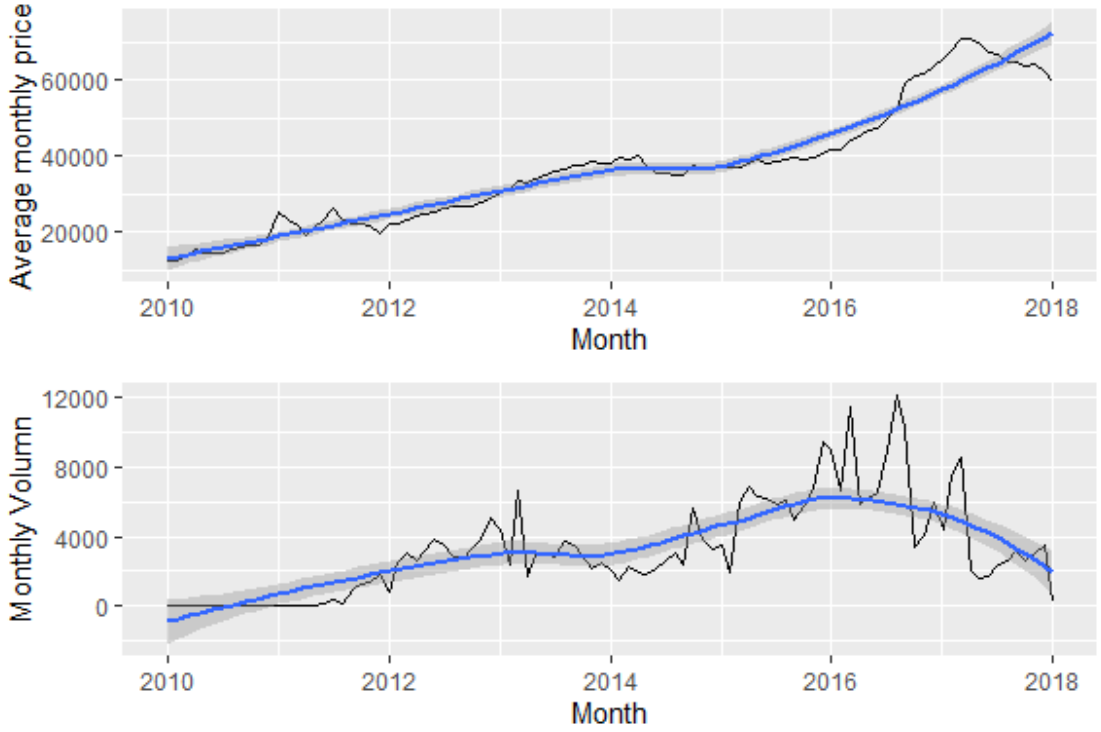




The above boxplots examined the effect of the `Floortype`,`BuildingType`,`renovationCondition`,and `buildingStructure` on the total price of the house. There is a mean difference within these four factors and I found that the houses

with evalotor, less than five years and close to the subway are more expensive. After I reordered the district by the total price, XiCheng and DongCheng are the most expensive area in Beijing and it makes sense that both districts are the center area of Beijing. Mapping the data is a better way to examine the geographic effects on the total price in future analysis.

3.5 Time trends for the average price per square



The above figure plots the monthly average price vaues and the count for transactions to show the growth of the market volumn and price in Beijing housing market. Because of the imcomplete data before 2011, the trend is not obvious. However, the market grew every year and reached the peak around 2017. More time series analysis and the prediction based on the time will be more helpful in the future analysis.

Supervised Machine Learning

4.1 Split Traning and Test Sets

In the following section, I build two models of regression and random forest based on the training set to predict the total price of the house on the testing set and compared them by

evaulating the model performance. Besause of the large volumn of the data, I selected one district `HaiDian`with approciate amount of the data, around 38 thousand of observations to build the models. Since there are too many factors in the dataset, I removed the two factors related to the preperties of the building `BuildingType` and `BuidlingStructure` to speed up computer calculations.

Then I split the dataset into training and test set using a random sample index. I created a training set named train consisting of 80% of the rows while a test set named test consisting of 20% of the rows of the housing data frame. The variables used in the training are listed blow

```
##  [1] "livingRoom"
##  [2] "drawingRoom"
##  [3] "kitchen"
##  [4] "age"
##  [5] "ladderRatio"
##  [6] "communityAverage"
##  [7] "totalprice"
##  [8] "square"
##  [9] "dom"
## [10] "elevator_with elevator"
## [11] "floortype_bottom"
## [12] "floortype_top"
## [13] "floortype_upper"
## [14] "floortype_unknownfloor"
## [15] "floortype_middle"
## [16] "subway_close to subway"
## [17] "renovationCondition_rough"
## [18] "renovationCondition_simplicity"
## [19] "renovationCondition_hardcover"
## [20] "fiveYearsProperty_less than 5 years"
```

4.2 Build the multiple regression model

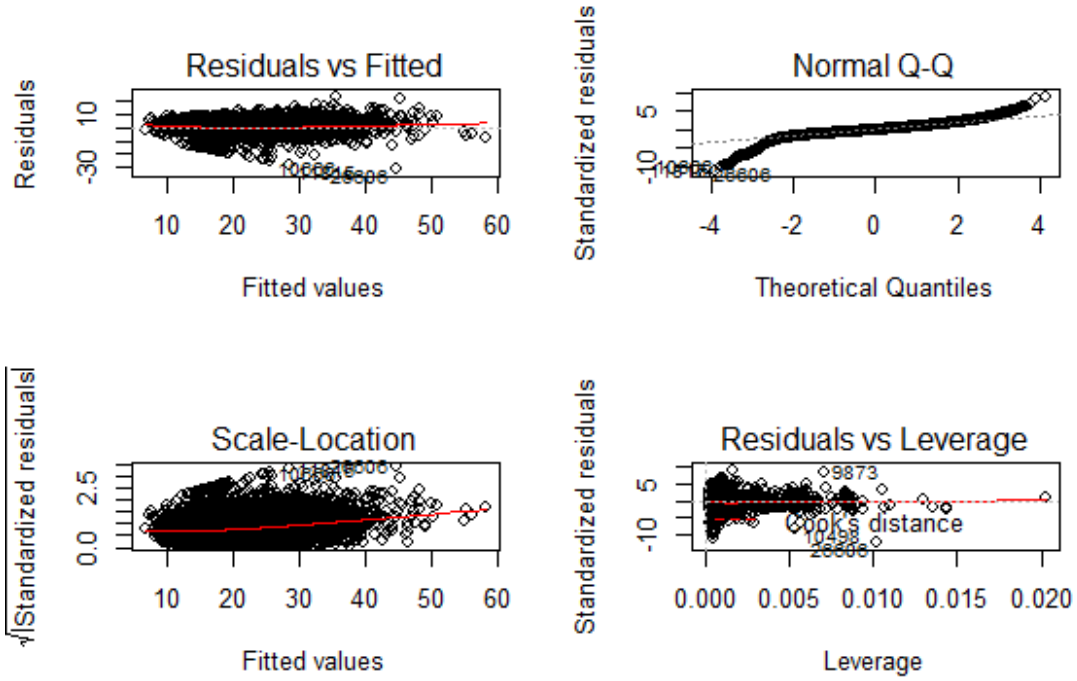
I first trained the linear regression model to predict total housing values using all other predictors and all the categorical variable “type” are all encoded into dummy variables. The summary results are shown below and I found that 77.66% variability in the total housing price was account for the model of all predictors forecasting the values of housing price. Surprisingly, the predictors of `subway` don’t have any effect on the total housing price. Besides, I made the 4 plots of regression diagnostics.

```
m1 <- lm(totalprice~., data=train)
summary(m1)

##
## Call:
## lm(formula = totalprice ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -30.5781  -1.5156  -0.1015   1.4937  22.2105
##
## Coefficients:
##                                     Estimate Std. Error t value
## (Intercept)                       -1.248e+01  2.203e-01 -56.653
## livingRoom                         1.321e-01  3.220e-02   4.102
## drawingRoom                        1.259e-01  4.166e-02   3.023
## kitchen                            8.200e-01  1.800e-01   4.556
## age                                3.496e-02  2.598e-03  13.457
## ladderRatio                        1.124e+00  9.770e-02  11.504
## communityAverage                   1.281e-04  1.083e-06 118.259
## square                             2.042e+00  1.703e-02 119.907
## dom                                4.957e-01  4.873e-03 101.718
## `elevator_with elevator`           1.001e-01  3.781e-02   2.646
## floortype_bottom                  -1.025e-01  6.038e-02  -1.697
## floortype_top                     -5.754e-01  5.582e-02 -10.308
```

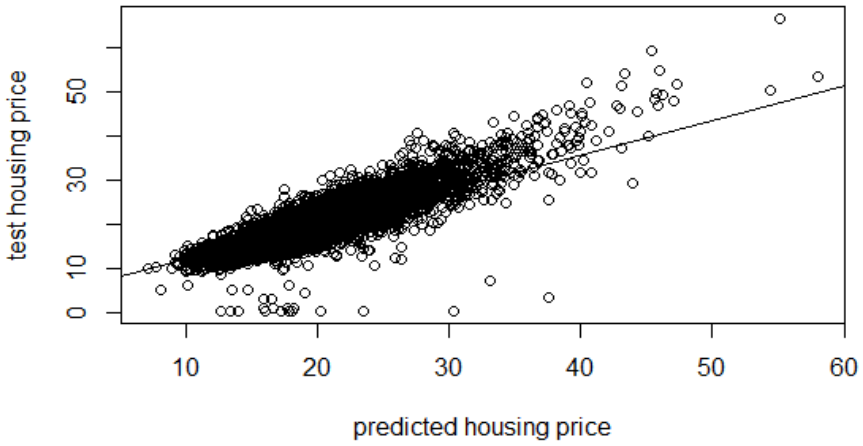
```
## floortype_upper                           8.481e-02  4.640e-02   1.828
## floortype_unknownfloor                   -7.144e-01  2.376e-01  -3.007
## floortype_middle                          8.608e-02  4.128e-02   2.085
## `subway_close to subway`                  1.690e-02  3.267e-02   0.517
## renovationCondition_rough                 1.654e+00  1.343e-01  12.309
## renovationCondition_simplicity            2.119e+00  4.097e-02  51.720
## renovationCondition_hardcover             2.417e+00  3.662e-02  66.000
## `fiveYearsProperty_less than 5 years`    -3.307e-01  3.367e-02  -9.824
##                                          Pr(>|t|)
## (Intercept)                               < 2e-16 ***
## livingRoom                               4.11e-05 ***
## drawingRoom                               0.00250 **
## kitchen                                  5.23e-06 ***
## age                                       < 2e-16 ***
## ladderRatio                               < 2e-16 ***
## communityAverage                          < 2e-16 ***
## square                                    < 2e-16 ***
## dom                                       < 2e-16 ***
## `elevator_with elevator`                  0.00814 **
## floortype_bottom                          0.08970 .
## floortype_top                             < 2e-16 ***
## floortype_upper                           0.06756 .
## floortype_unknownfloor                    0.00264 **
## floortype_middle                          0.03706 *
## `subway_close to subway`                  0.60488
## renovationCondition_rough                 < 2e-16 ***
## renovationCondition_simplicity            < 2e-16 ***
## renovationCondition_hardcover             < 2e-16 ***
## `fiveYearsProperty_less than 5 years`     < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.591 on 30522 degrees of freedom
```

```
## Multiple R-squared:  0.7768, Adjusted R-squared:  0.7766
## F-statistic:  5590 on 19 and 30522 DF,  p-value: < 2.2e-16
```

The four residual V.S fitted, normal Q-Q, scale-location and residual V.S leverage plots show an acceptable distribution of the residuals and the four assumptions of the linear regression: linearity, independence, normality, and equality of variance has been verified as well.



To evaluate the model, I first plotted the relationship between the predicted and actual total housing price in the test set shown below. Besides, the standard deviation of the residuals is evaluated to show how well the algorithm was able to predict the response variable. The function for calculating RMSE is defined as `sqrt(mean((y_hat-y)^2)`.



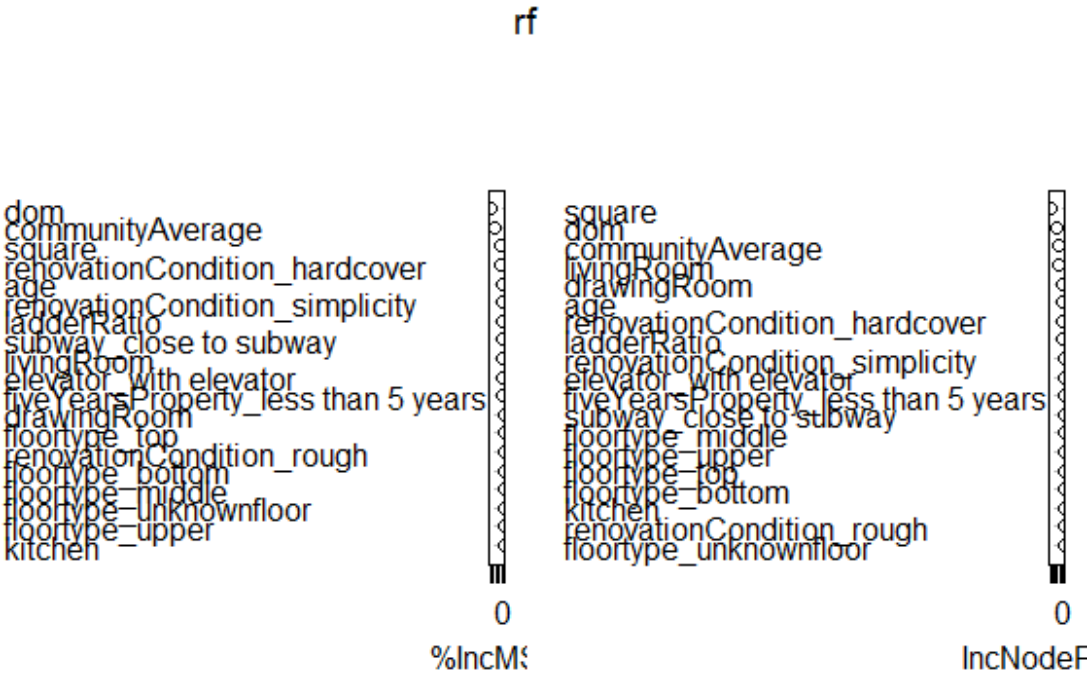
Compared with the 'RMSE' of 2.59 for training set, the test set generates 'RMSE' score of only 2.62. The model scored roughly the same on the training and test data and it suggests that it made a good prediction.

```
# Calculate RMSE for training set
rmse_train <- rmse(predict(m1),train$totalprice)
rmse_train

## [1] 2.590036

# Calculate RMSE for test set
rmse_test <- rmse(predict(m1, newdata=test),
                  test$totalprice)
rmse_test

## [1] 2.624965
```

4.3 Build the Radom Forest

Next I adopted the `randomForest()` algorithm for training and inference. I construct 500 decision trees for this random forest to achieve good performance. The `rf$importance` is displayed below to indicate the importance of given predictors in the performance of the model by checking the number of mean squared error.

```
##                                        %IncMSE IncNodePurity
## square                               23.382574225   301481.4655
## dom                                  10.886665056   174794.7494
## communityAverage                      9.030880083   137656.4032
## livingRoom                            3.061143499   103811.7827
## drawingRoom                           0.985420542    46326.3091
## age                                   2.680520718    36636.6610
## renovationCondition_hardcover         1.400784694    23282.9059
## ladderRatio                           1.115165906    22201.0395
## renovationCondition_simplicity        0.783492083    11735.8849
## elevator_with elevator                0.729244754     7563.0099
## fiveYearsProperty_less than 5 years   0.232427582     5145.0044
## subway_close to subway                0.439475190     5137.9023
## floortype_middle                      0.022055399     3873.6338
## floortype_upper                       0.002023204     3136.8847
## floortype_top                         0.064929089     3023.1207
## floortype_bottom                      0.029439657     2923.7116
## kitchen                              -0.008125663     1326.3831
## renovationCondition_rough             0.011300981     1003.7900
## floortype_unknownfloor                0.001531455      220.5869
```



The out-of-bag (oob) error estimate in this random forrest is 1.99 and the resulting RMSE is the prediction of total price of a house in a given district to within a RMSE delta of the actual total house pric.If applying into the test se, the RMSE is 2.06. The model scored roughly the same on the training and test data and it suggests that it made a good prediction.

```
# Compute the out-of-bag (oob) error estimate
oob_prediction <- predict(rf)
train_mse <- mean(as.numeric((oob_prediction -train$totalprice)^2))
oob_rmse = sqrt(train_mse)
oob_rmse

## [1] 1.989642

# Calculate RMSE for test set
y_pred = predict(rf , test[,-7])
test_mse = mean(((y_pred - test$totalprice)^2))
sqrt(test_mse)

## [1] 2.070075
```

Compared with the regression model, square, age, community and date on market are more important features in both models. The RMSE for the test set in the regression is higher than that in the random forest algorithm and it suggests that the random forest algorithm is a better fit predicting the total price of a house.

Limitation and Conclusion

One of the limitations in this analysis is the lack of data mapping and time series analysis, which helps us get a deeper understanding of the geographic and time effects on the housing price. With the mapping of the data, future work can be to visualize the variation of housing prices in each region, even in each community on the map of Beijing. On the other hand, we could consider predicting the values based on the previous time trends in the application time series analysis. Besides, I used the data within the district "HaiDian" to establish the prediction model and the rest of the districts can be analyzed in the future work as well.

I adopted the variable of `totalPRICE` as the main response variable but the columns of `price` and `community average` are ought to be considered as well. I guess that the variable of `totalPRICE` is more affected by the interior property of the house while `price` and `community average` are more influenced by the exterior environment of the community. This assumption requires future analysis to work on it.

Although it is a quite general and exploratory analysis of the housing price in Beijing, I reach abundant information and conclusions. The total price and price values per square have a significant growth from 2010 to 2018 and there is a mean difference in the thirteen regions of Beijing. Like the square, the date on market and the number of the living room and bedroom increase, the total price and price values per square increase. The variables of the square, age, community and date on market are more influential features in both models predicting the total price of a house. We know that Beijing is the capital of China with the increasing population and the Winter Olympics will open in Beijing in 2020. With these factors, I believe the housing price in Beijing will still increase in Beijing.