Stat3340_Final_Project

Nicole Torrie & Qinyan Jiang

07/12/2020

1.0 Abstract

The purpose of our project is to research the Real Estate dataset through different images and models. The final product will be a model which can be used to determine housing price based on variables determined to have high influence on housing price. First, we will introduce our topic to give background context on the Real Estate market. We then introduce the dataset and prep the data for analysis. Next we analyze the relationships between variables and build a linear model. Then, we will further validate our model by fitting it to training data and testing it on a separate test subset of the data. Finally, we will summarize the research conclusions based on the results of data experiments and our personal experience.

2.0 Introduction

With the development of the economy, people's demand for houses is increasing, so the housing price has become a very important topic to research. There are also many factors that affect the housing price, such as the age of the house, the market trend when buying the house, the acesss to transportation, the convenience of life, the geographical location of the house and so on. It is important to study the historical data to find the relationship between housing price and explanatory variables. For buyers, this can help them make a better decision when buying a house that is best suited for their lifestyle. For sellers, this can help them make an estimate of the value of the house that they should be selling at. For investors, this can help them understand the trend of the market and make appropriate choices to make profits. Our study use the dataset that consists of six variables and 414 houses. We will conduct data visualization on the dataset and study on different model or methods to summarize the conclusions. We will include our code throughout this analysis so that other users can follow along and reproduce our final model. The main question we hope to address through this analysis is: which variables have highest influence on housing price per unit area?

3.0 Data Description

3.1 Downloading the data

Download the Real Estate dataset from the source: https://www.kaggle.com/quantbruce/real-estate-price-prediction, or find it attached to the github repository: https://github.com/nicoletorrie/STAT3340_Term_Project

Load the necessary packages:

```
library(maps)
library(mapdata)
```

```
## Warning: package 'mapdata' was built under R version 4.0.3
```

```
library(maptools)  #for shapefiles
```

```
## Warning: package 'maptools' was built under R version 4.0.3
```

```
## Loading required package: sp
```

```
## Warning: package 'sp' was built under R version 4.0.3
```

```
## Checking rgeos availability: TRUE
```

```
library(scales)  #for transparency
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.3
```

```
library(sf)
```

```
## Warning: package 'sf' was built under R version 4.0.3
```

```
## Linking to GEOS 3.8.0, GDAL 3.0.4, PROJ 6.3.1
```

```
library(rnaturalearth)
```

```
## Warning: package 'rnaturalearth' was built under R version 4.0.3
```

```
library(rnaturalearthdata)
```

```
## Warning: package 'rnaturalearthdata' was built under R version 4.0.3
```

```
library(rgeos)
```

```
## Warning: package 'rgeos' was built under R version 4.0.3
```

```
## rgeos version: 0.5-5, (SVN revision 640)
##  GEOS runtime version: 3.8.0-CAPI-1.13.1
##  Linking to sp version: 1.4-4
##  Polygon checking: TRUE
```

```
library(ggspatial)
```

```
## Warning: package 'ggspatial' was built under R version 4.0.3
```

```
library(ggmap)
```

```
## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.
```

```
## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```
library(dplyr)
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.0.2
```

Attach and view the Real Estate Data

```
Data <- read.csv("Real estate.csv")
attach(Data)
head(Data)
```

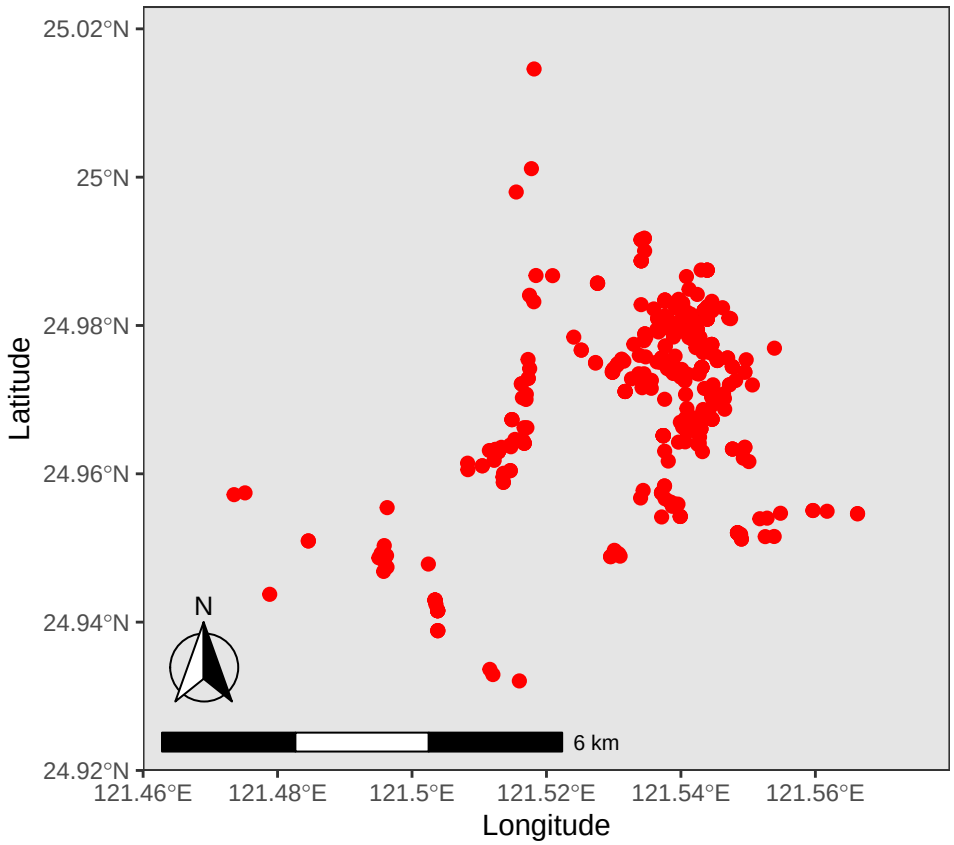
```
##   No X1.transaction.date X2.house.age X3.distance.to.the.nearest.MRT.station
## 1  1            2012.917         32.0                                84.87882
## 2  2            2012.917         19.5                               306.59470
## 3  3            2013.583         13.3                               561.98450
## 4  4            2013.500         13.3                               561.98450
## 5  5            2012.833          5.0                               390.56840
## 6  6            2012.667          7.1                              2175.03000
##   X4.number.of.convenience.stores X5.latitude X6.longitude
## 1                              10    24.98298     121.5402
## 2                               9    24.98034     121.5395
## 3                               5    24.98746     121.5439
## 4                               5    24.98746     121.5439
## 5                               5    24.97937     121.5425
## 6                               3    24.96305     121.5125
##   Y.house.price.of.unit.area
## 1                       37.9
## 2                       42.2
## 3                       47.3
## 4                       54.8
## 5                       43.1
## 6                       32.1
```

3.2 Data Information

Study Area: We first plot the data on a map so that we can get a better understanding of the study area. The code for two mapping methods are below:

Create a basic map with a scalebar to understand the proximity of data points:

```
# gene world map
world <- ne_countries(scale = "medium", returnclass = "sf")
ggplot(data = world) +
  geom_sf() +
  labs( x = "Longitude", y = "Latitude") +
  coord_sf(xlim = c(121.460,121.58), ylim = c(24.92,25.023), expand = FALSE) +
  annotation_scale(location = "bl", width_hint = 0.5) +
  annotation_north_arrow(location = "bl", which_north = "true",
                         pad_x = unit(0.02, "in"), pad_y = unit(0.3, "in"),
                         style = north_arrow_fancy_orienteering) +
  geom_point(data = Data, aes(x = X6.longitude, y =X5.latitude ),col="red", size=2)+
  theme_bw()
```

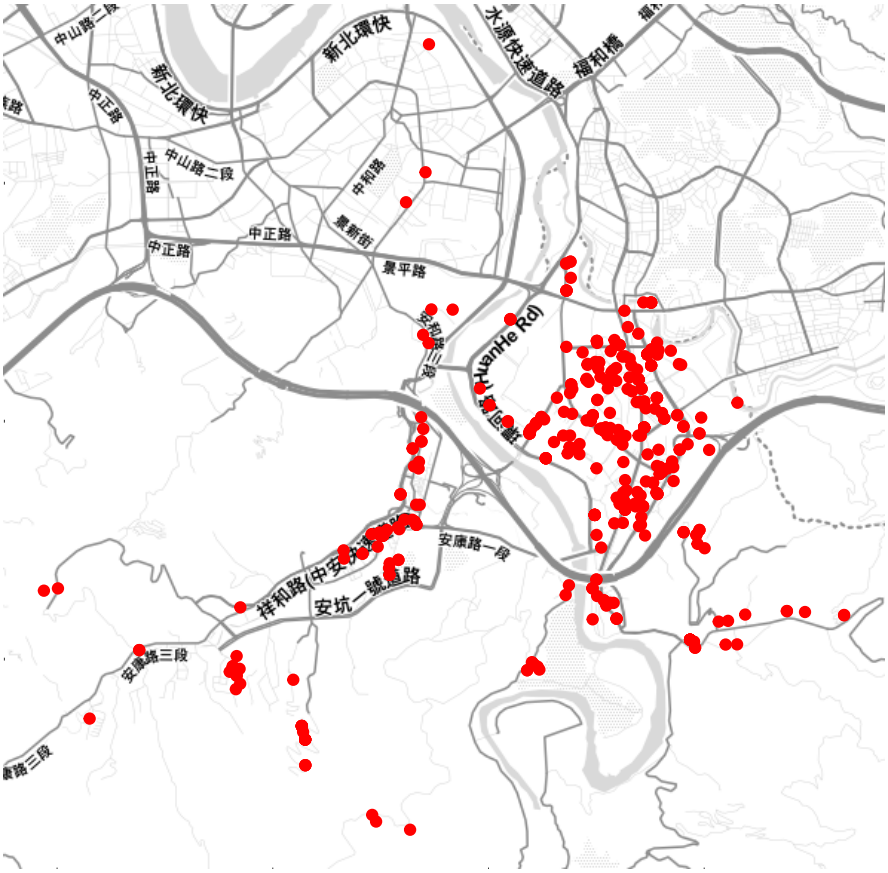


Create a more detailed street map:

```
qmplot(X6.longitude, X5.latitude, data = Data, maptype = "toner-lite", color = I("red"))+
         labs( x = "Longitude", y = "Latitude")
```

```
## Using zoom = 13...
```

```
## Map tiles by Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under ODbL.
```



According to the latitude and longitude information, the dataset is from Taiwan, on the outskirts of the country's capital city, Taipei. Now that we know our study area, we can look at the variables within the context of the study area.

One limitation to understanding the variables included in this analysis is that there are limited variable descriptions included in the online database from which we sourced our data, thus our definitions of the variables below are based on inference.

The independent variables to be included in this analysis are as follows: X1.transaction.date: the transaction date that the house sold on X2.house.age: the age of the house X3.distance.to.the.nearest.MRT.station: The distance between the house and the nearest MRT station. MRT is inferred to mean "Mass Rapid Transit" station. X4.number.of.convenience.stores: The number of convenience stores near the house, with proximity undefined X5.latitude: The latitude of the house location X6.longitude: The longitude of the house location

The dependent variable is: Y.house.price.of.unit.area: The price of the house per unit area. This is presented as per unit area, instead of overall house price to reflect the fact that houses considered in this model are of different sizes. We scale them to price per unit area so that the value can be compared.

We ignore the first column with label "no". Because this is not a variable, and we will not include it in the analysis.

4.0 Methods

We began analysis by analyzing model variables and determining whether it makes sense to consider them in the Real Estate model. Based on these decisions, we re-formatted the dataset accordingly and added our new data point. Once our data was prepped and variables were formatted, we built a linear model with all variables from the new dataset included. This was to get an initial idea of model accuracy.

Crating diagnostic plots of the full model allowed us to identify influential leverage points and outliers in the data. We used cook's distance value to eliminate influential points from the dataset. In total there were 18 influential points identified.

Upon removal of influential points, we re-fit the full model to the new dataset and re-created the diagnostic plots so as to test the following regression assumptions:

1. Linearity of the data.
2. Normality of residuals.
3. Homogeneity of residuals variance.
4. Independence of residuals error terms.

Next we tested the correlation between variables using pairs plots. There did not appear to be strong linear correlation between any of the variables. We also tested for multicollinearity using VIF values. A VIF greater than 10 indicates multicollinearity, and suggests that a variable should be removed.

After confirming the variables and data points that should be considered in the final model, we used stepwise regression to create the strongest model, by eliminating any variables which had high associated AIC values. In addition, we compared the model output from stepwise regression to the model output created by backward regression. Both models were identical, which validated the model. We further validated the final model by splitting it into training and test data, creating a model using the training set, and comparing predicted and observed values of the test data run through the model. The RMSE values and the predicted vs. observed plots allowed us to draw conclusions on how accurately the model had predicted the test data.

The afore-mentioned methods are presented with R-code and analysis in the results section (5.0).

5.0 Results

5.1 Data prepping

We begin by prepping the dataset.

First, take a look at the model variables and decide if any variables or values should be re formatted or removed. Column 1, "No" can be removed, as the values in this column are merely a count of the rows in the dataset.

Column 2, "X1.transaction.date" was divided to just include the transaction year, with the month and day values being eliminated. Year has influence on housing price for several reasons. First, the value of the dollar, (inflation or deflation) will vary by year, thus influencing a house's selling price.

Column 3, "X2.house.age" is an important predictor of price, because house value can either decrease or increase with age. Older houses could be in some cases more desirable due to their antique character or historical value. On the other hand, these houses may be less desirable because of higher infrastructural upkeep costs or outdated features and style.

Column 4, "X3.distance.to.the.nearest.MRT.station" is another important predictor of price. Proximity to public transit can cause house value to increase due to ease of transportation and accessibility that comes along with it.

Column 5, "X4.number.of.convenience.stores" is similarly important. Access to convenience stores in close proximity means homeowners have the ability to pick up food items, or essential small household items last minute. Contrarily, a person living in a home far from any stores will have to rely on transportation to access these services.

Column 6 and 7, "X5.latitude" and "X6.longitude" work together to provide the location of the house. For similar reasons as discussed with the previous variables, location does have impact on the retail price of a house. Often, houses in more desirable locations will be more expensive.

Reference: https://www.opendoor.com/w/blog/factors-that-influence-home-value

Split "X1.transaction.date" to just include the year of transaction date and remove column 1.

```
#split X1.transaction.date
library(tidyr)
Data2<-separate(Data, X1.transaction.date, c("X1.transaction.date",NA,NA))
```

```
## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 414 rows [1, 2,
## 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
#make a new dataset with just columns 2-8 to eliminate column 1
Data3<-Data2[,2:8]
head(Data3)
```

```
##   X1.transaction.date X2.house.age X3.distance.to.the.nearest.MRT.station
## 1                2012         32.0                                84.87882
## 2                2012         19.5                               306.59470
## 3                2013         13.3                               561.98450
## 4                2013         13.3                               561.98450
## 5                2012          5.0                               390.56840
## 6                2012          7.1                              2175.03000
##   X4.number.of.convenience.stores X5.latitude X6.longitude
## 1                              10    24.98298     121.5402
## 2                               9    24.98034     121.5395
## 3                               5    24.98746     121.5439
## 4                               5    24.98746     121.5439
## 5                               5    24.97937     121.5425
## 6                               3    24.96305     121.5125
##   Y.house.price.of.unit.area
## 1                       37.9
## 2                       42.2
## 3                       47.3
## 4                       54.8
## 5                       43.1
## 6                       32.1
```

```
#remove NA values
Data3<-Data3[complete.cases(Data3),]
```

Next, we add an additional data point to our dataset, to make the dataset unique for the purposes of this project. For this data point, we use the average value of each column to create the new value. In order to take the average of each column in the data frame, each column has to be converted from character to numeric type.

```
#check and convert the column types to numeric
Data3<-as.data.frame(Data3)
str(Data3)
```

```
## 'data.frame':    414 obs. of  7 variables:
##  $ X1.transaction.date                   : chr  "2012" "2012" "2013" "2013" ...
##  $ X2.house.age                          : num  32 19.5 13.3 13.3 5 7.1 34.5 20.3 31.7 17.9 ...
```

```
##  $ X3.distance.to.the.nearest.MRT.station: num  84.9 306.6 562 562 390.6 ...
##  $ X4.number.of.convenience.stores       : int  10 9 5 5 5 3 7 6 1 3 ...
##  $ X5.latitude                           : num  25 25 25 25 25 ...
##  $ X6.longitude                          : num  122 122 122 122 122 ...
##  $ Y.house.price.of.unit.area            : num  37.9 42.2 47.3 54.8 43.1 32.1 40.3 46.7 18.8 22.1 ..
```

```
Data4<-data.frame(lapply(Data3,as.numeric))
str(Data4)
```

```
## 'data.frame':    414 obs. of  7 variables:
##  $ X1.transaction.date                   : num  2012 2012 2013 2013 2012 ...
##  $ X2.house.age                          : num  32 19.5 13.3 13.3 5 7.1 34.5 20.3 31.7 17.9 ...
##  $ X3.distance.to.the.nearest.MRT.station: num  84.9 306.6 562 562 390.6 ...
##  $ X4.number.of.convenience.stores       : num  10 9 5 5 5 3 7 6 1 3 ...
##  $ X5.latitude                           : num  25 25 25 25 25 ...
##  $ X6.longitude                          : num  122 122 122 122 122 ...
##  $ Y.house.price.of.unit.area            : num  37.9 42.2 47.3 54.8 43.1 32.1 40.3 46.7 18.8 22.1 ..
```

```
#create values for the additional data point to add to the dataset
Newrow<-data.frame(t(colMeans(x=Data4, na.rm = TRUE)))
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 4.0.3
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
#attach the additional data point to the end of the dataset.
RE_Data <- rbind(Data4,Newrow)
str(RE_Data)
```

```
## 'data.frame':    415 obs. of  7 variables:
##  $ X1.transaction.date                   : num  2012 2012 2013 2013 2012 ...
##  $ X2.house.age                          : num  32 19.5 13.3 13.3 5 7.1 34.5 20.3 31.7 17.9 ...
##  $ X3.distance.to.the.nearest.MRT.station: num  84.9 306.6 562 562 390.6 ...
##  $ X4.number.of.convenience.stores       : num  10 9 5 5 5 3 7 6 1 3 ...
##  $ X5.latitude                           : num  25 25 25 25 25 ...
##  $ X6.longitude                          : num  122 122 122 122 122 ...
##  $ Y.house.price.of.unit.area            : num  37.9 42.2 47.3 54.8 43.1 32.1 40.3 46.7 18.8 22.1 ..
```

```
#RE_Data2<-as.data.frame(RE_Data)
head(RE_Data)
```

```
##   X1.transaction.date X2.house.age X3.distance.to.the.nearest.MRT.station
## 1                2012         32.0                                84.87882
## 2                2012         19.5                               306.59470
## 3                2013         13.3                               561.98450
```

```
## 4                2013         13.3                              561.98450
## 5                2012          5.0                              390.56840
## 6                2012          7.1                             2175.03000
##   X4.number.of.convenience.stores X5.latitude X6.longitude
## 1                              10    24.98298     121.5402
## 2                               9    24.98034     121.5395
## 3                               5    24.98746     121.5439
## 4                               5    24.98746     121.5439
## 5                               5    24.97937     121.5425
## 6                               3    24.96305     121.5125
##   Y.house.price.of.unit.area
## 1                       37.9
## 2                       42.2
## 3                       47.3
## 4                       54.8
## 5                       43.1
## 6                       32.1
```

5.2 Linear Regression Model

Now that the data has been properly formatted, and the new datapoint is added we can start building the regression model.

First, fit a regression model including all independent variables so we can gauge model accuracy.

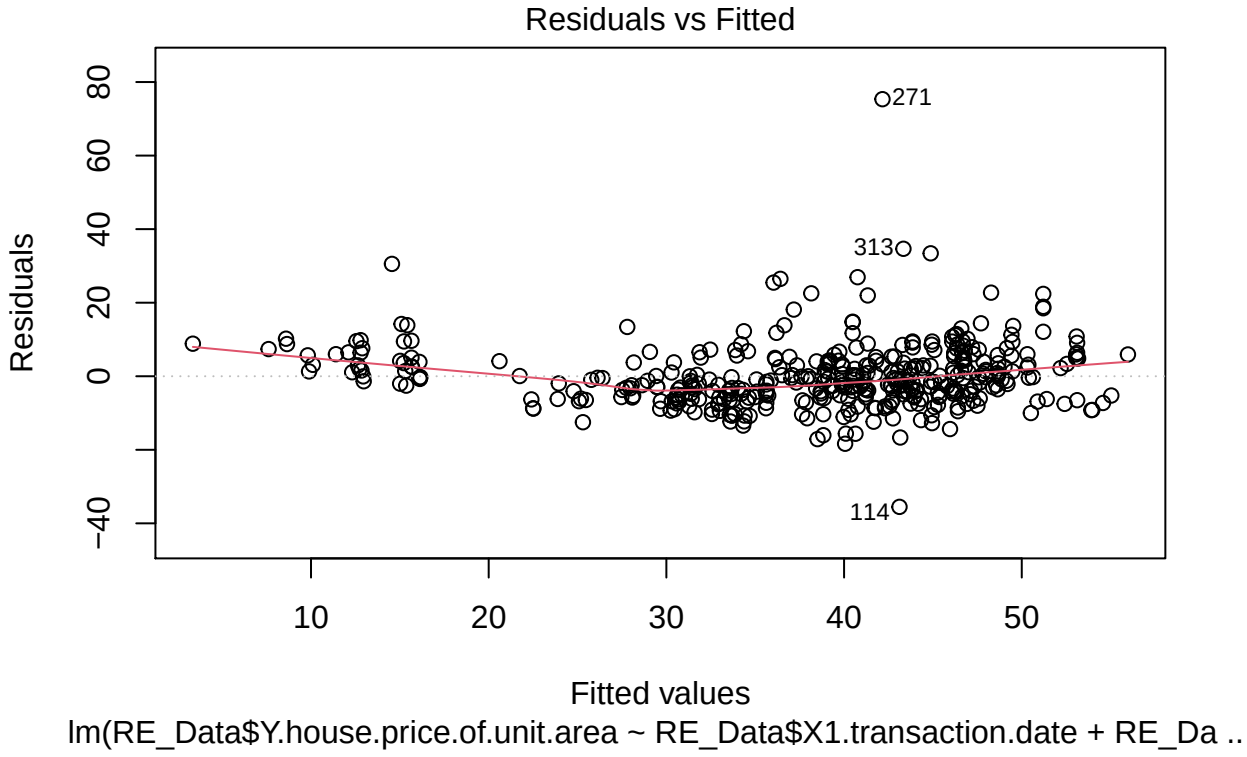
```
#library(MPV)
#library(faraway)

#Fit the base model with all variables included
BaseModel<-lm(RE_Data$Y.house.price.of.unit.area~RE_Data$X1.transaction.date+RE_Data$X2.house.age+RE_Da

summary(BaseModel)
```

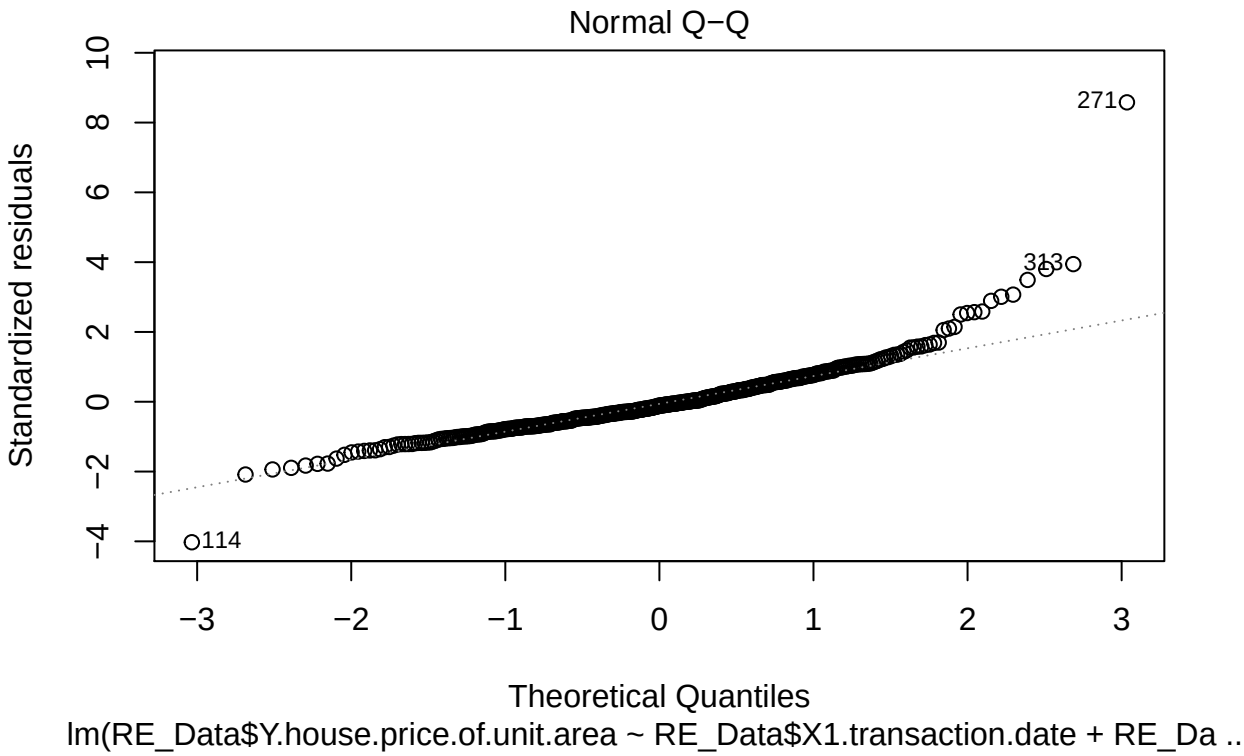
```
##
## Call:
## lm(formula = RE_Data$Y.house.price.of.unit.area ~ RE_Data$X1.transaction.date +
##     RE_Data$X2.house.age + RE_Data$X3.distance.to.the.nearest.MRT.station +
##     RE_Data$X4.number.of.convenience.stores + RE_Data$X5.latitude +
##     RE_Data$X6.longitude)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.520  -5.261  -0.974   4.147  75.338
##
## Coefficients:
##                                               Estimate Std. Error t value
## (Intercept)                                 -9.860e+03  6.342e+03  -1.555
## RE_Data$X1.transaction.date                  2.937e+00  9.491e-01   3.095
## RE_Data$X2.house.age                        -2.747e-01  3.859e-02  -7.119
## RE_Data$X3.distance.to.the.nearest.MRT.station -4.370e-03  7.159e-04  -6.104
## RE_Data$X4.number.of.convenience.stores      1.162e+00  1.880e-01   6.179
## RE_Data$X5.latitude                          2.345e+02  4.444e+01   5.276
## RE_Data$X6.longitude                        -1.534e+01  4.864e+01  -0.315
```

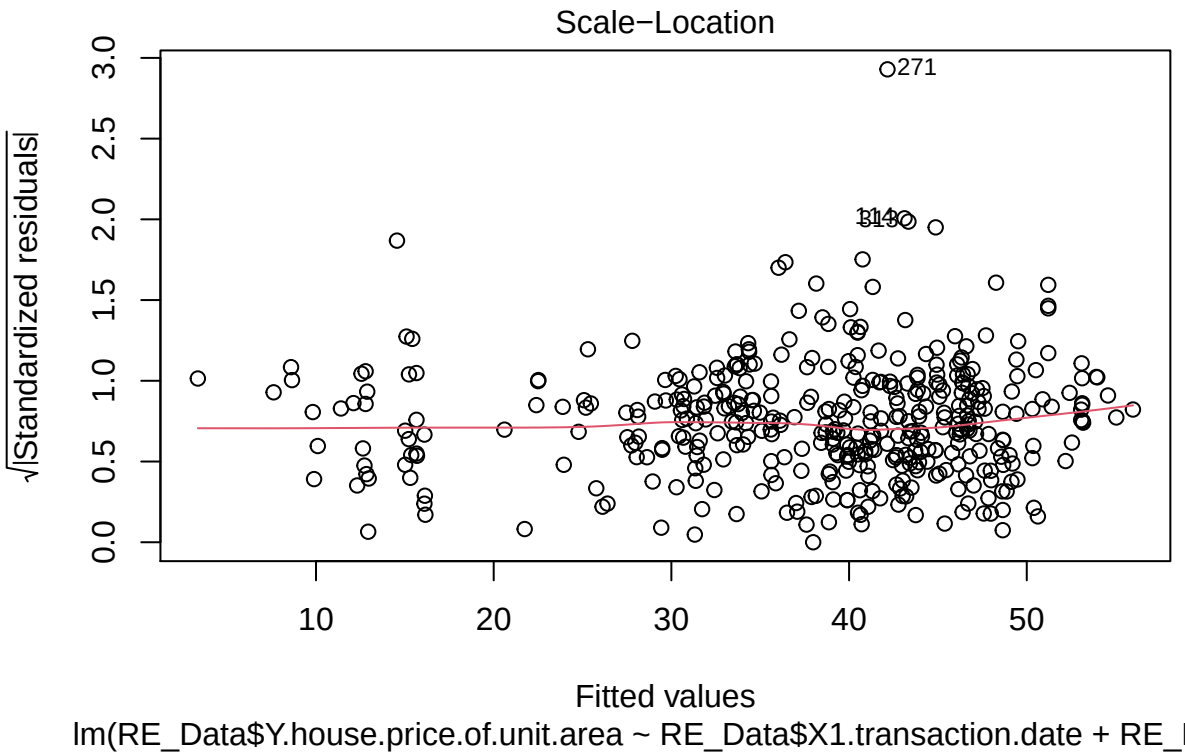
```
##                                                Pr(>|t|)
## (Intercept)                                     0.12078
## RE_Data$X1.transaction.date                     0.00211 **
## RE_Data$X2.house.age                           4.94e-12 ***
## RE_Data$X3.distance.to.the.nearest.MRT.station 2.40e-09 ***
## RE_Data$X4.number.of.convenience.stores        1.56e-09 ***
## RE_Data$X5.latitude                            2.15e-07 ***
## RE_Data$X6.longitude                            0.75272
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.861 on 408 degrees of freedom
## Multiple R-squared:  0.581,  Adjusted R-squared:  0.5748
## F-statistic: 94.29 on 6 and 408 DF,  p-value: < 2.2e-16
```

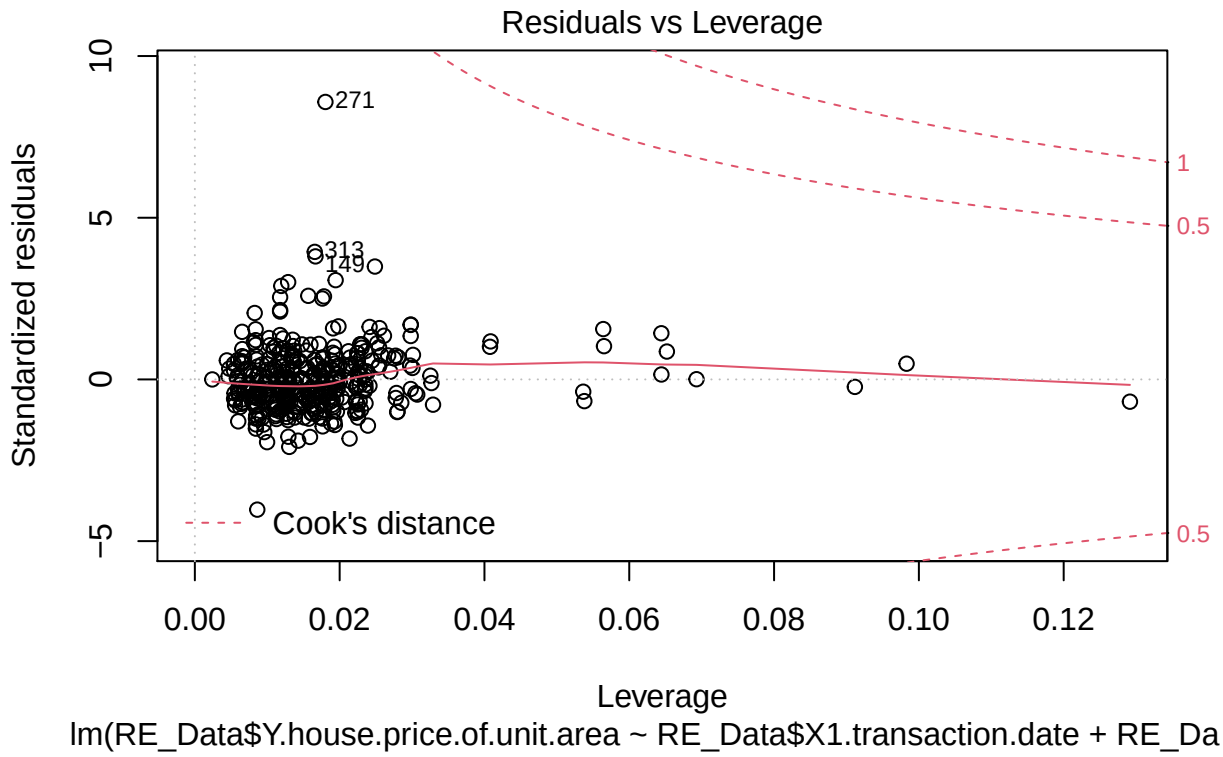
The adjusted R-squared value of this full model is low, at 0.5748. This is indication that the base model is not well performing, and variables should be re-visited to determine if they should be removed from the model to make it more accurate. Before we re-assess the variables to include, check the plots to assess leverage points and outliers.

```
plot(BaseModel)
```




Normal Q−Q
Standardized residuals
10
8
6
4
2
0
−2
−4
271
313
114
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(RE_Data$Y.house.price.of.unit.area ~ RE_Data$X1.transaction.date + RE_Da ..


Scale−Location
271
114
313
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
10
20
30
40
50
Fitted values
lm(RE_Data$Y.house.price.of.unit.area ~ RE_Data$X1.transaction.date + RE_Da ..



Based on the model diagnostic plots, it appears that observations 271, 313, and 114 and potentially a few others are outliers to the dataset. Removing these could improve model accuracy.

Let's remove influential points:

```
#use Cook's distance to eliminate outliers or variables with strong influence.
#for this analysis we eliminate values with a cook's distance value above 4/number of observations

cooksd=cooks.distance(BaseModel)
sample_size=length(RE_Data$X1.transaction.date)
influential<-as.numeric(names(cooksd)[(cooksd>(4/sample_size))])
RE_DataNEW2<-RE_Data[-influential,]

#Fit the base model with all variables included now that the outliers and leverage points have been rem
BaseModel2<-lm(RE_DataNEW2$Y.house.price.of.unit.area~RE_DataNEW2$X1.transaction.date+RE_DataNEW2$X2.hou
```

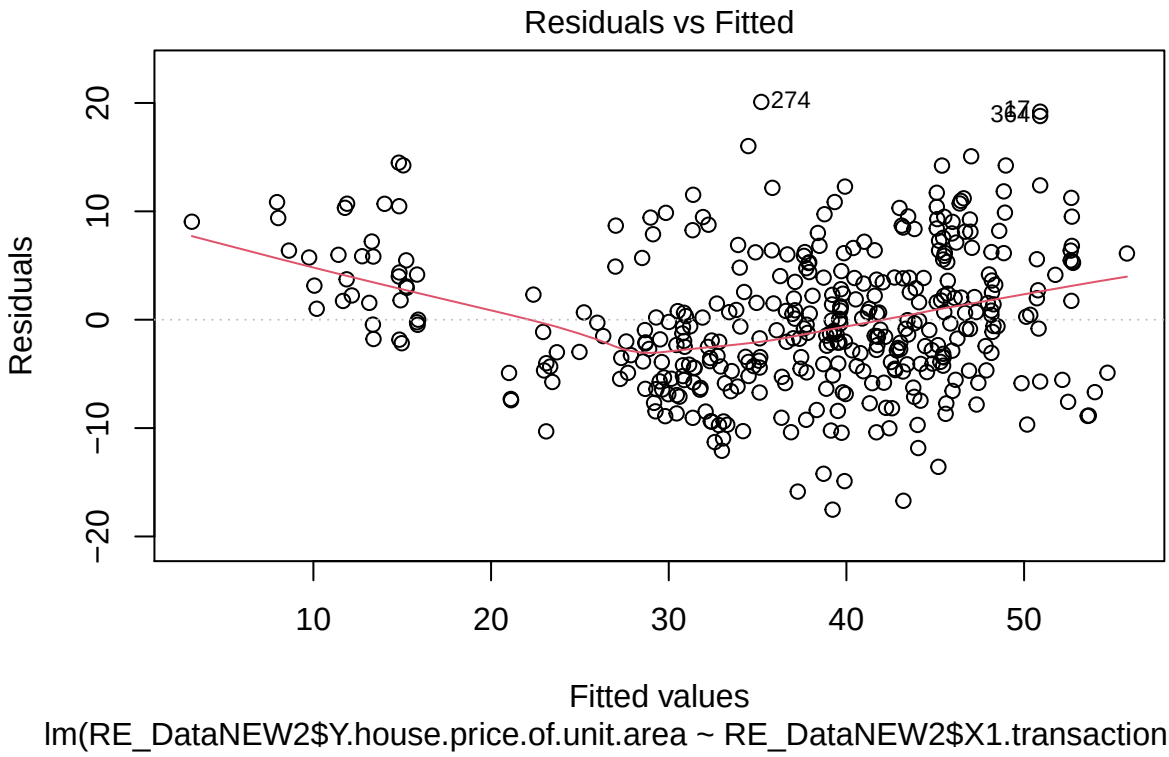
```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'Data' will be disregarded
```

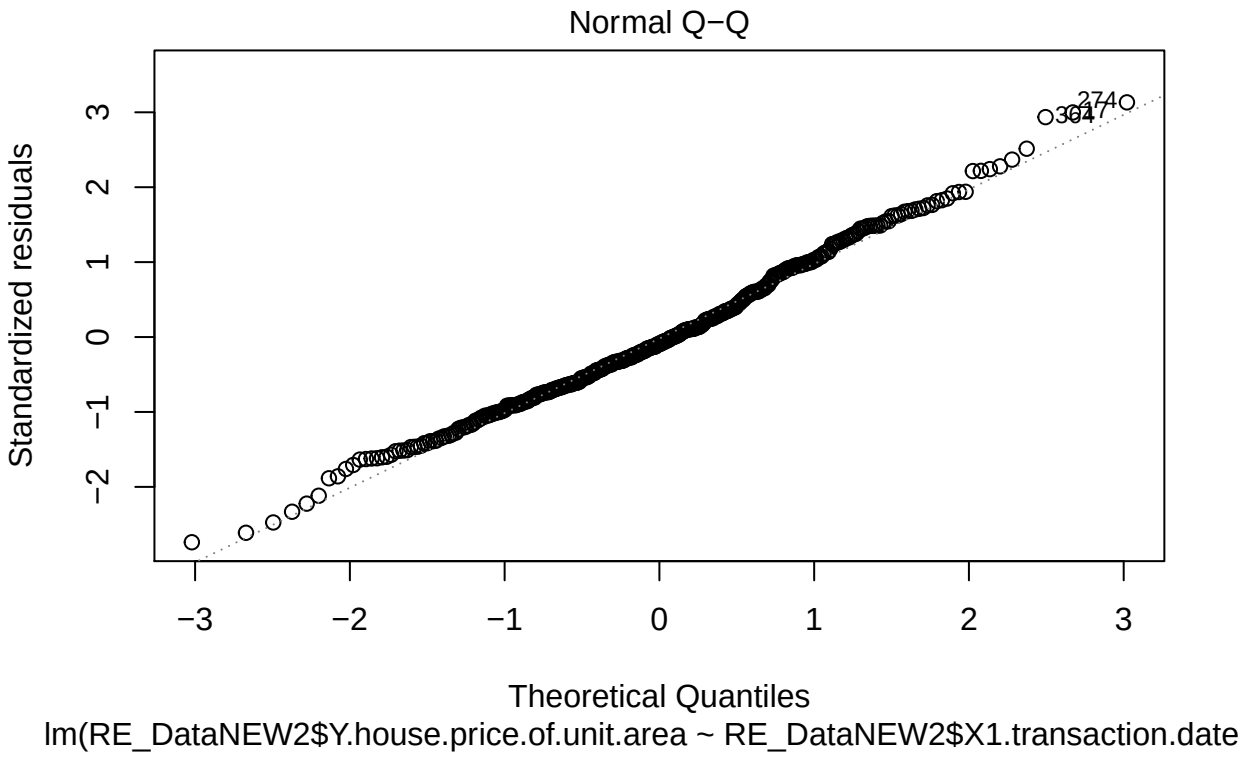
```
summary(BaseModel2)
```

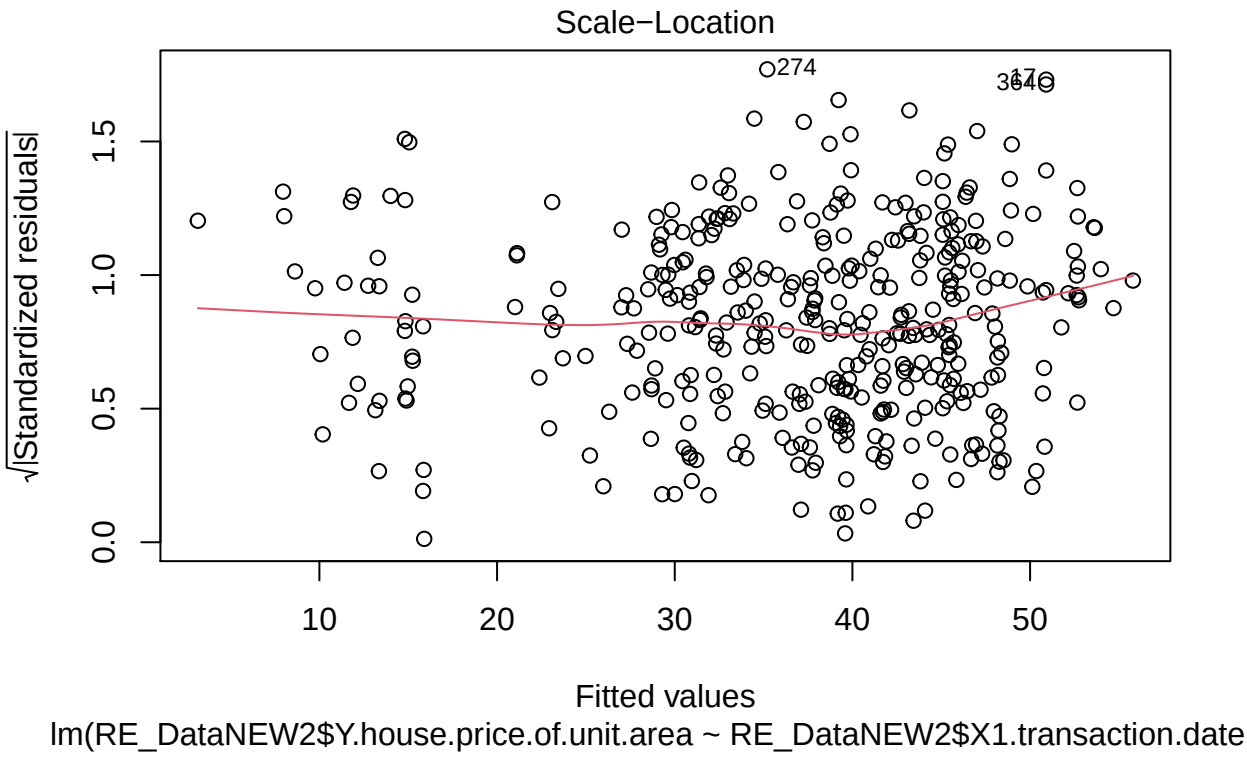
```
##
## Call:
## lm(formula = RE_DataNEW2$Y.house.price.of.unit.area ~ RE_DataNEW2$X1.transaction.date +
##     RE_DataNEW2$X2.house.age + RE_DataNEW2$X3.distance.to.the.nearest.MRT.station +
##     RE_DataNEW2$X4.number.of.convenience.stores + RE_DataNEW2$X5.latitude +
```

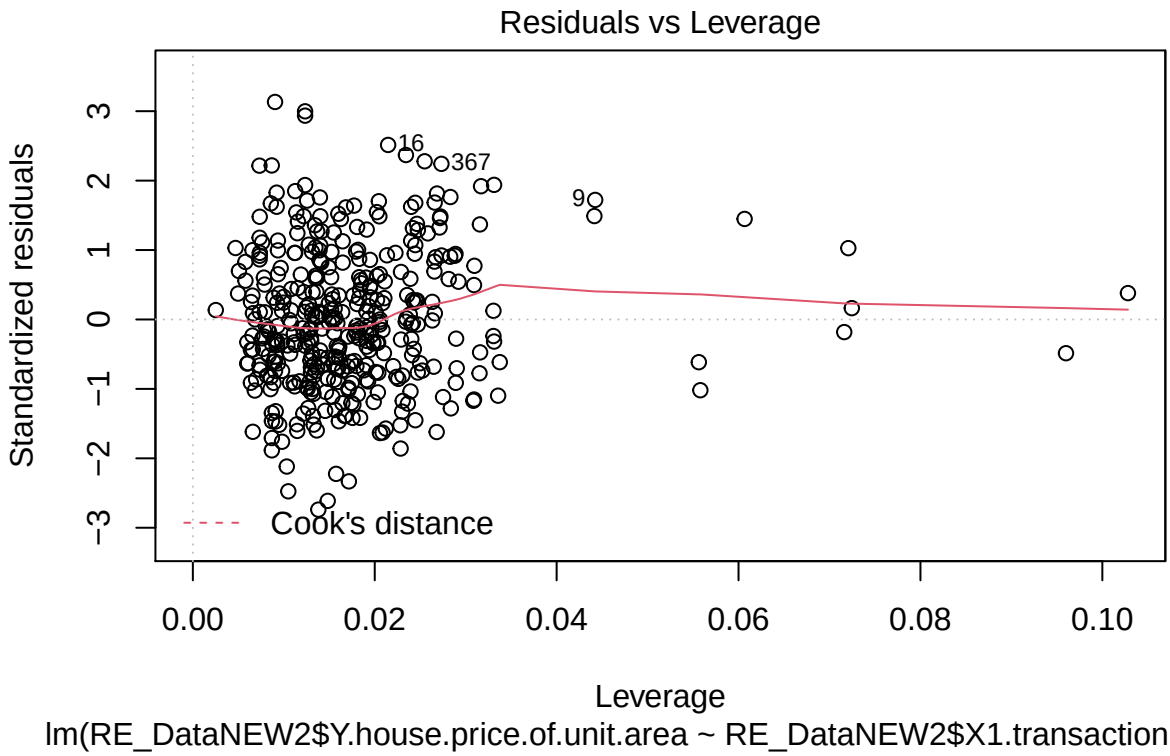
```
##     RE_DataNEW2$X6.longitude, Data = RE_DataNEW2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.5235  -4.4251  -0.6028   4.1651  20.0917
##
## Coefficients:
##                                                  Estimate Std. Error
## (Intercept)                                    -1.154e+04  4.788e+03
## RE_DataNEW2$X1.transaction.date                 2.130e+00  7.020e-01
## RE_DataNEW2$X2.house.age                       -3.203e-01  2.939e-02
## RE_DataNEW2$X3.distance.to.the.nearest.MRT.station -3.746e-03  5.608e-04
## RE_DataNEW2$X4.number.of.convenience.stores     1.272e+00  1.412e-01
## RE_DataNEW2$X5.latitude                         2.318e+02  3.704e+01
## RE_DataNEW2$X6.longitude                        1.245e+01  3.605e+01
##                                                t value Pr(>|t|)
## (Intercept)                                     -2.411  0.01637 *
## RE_DataNEW2$X1.transaction.date                  3.033  0.00258 **
## RE_DataNEW2$X2.house.age                       -10.900  < 2e-16 ***
## RE_DataNEW2$X3.distance.to.the.nearest.MRT.station  -6.680 8.28e-11 ***
## RE_DataNEW2$X4.number.of.convenience.stores      9.006  < 2e-16 ***
## RE_DataNEW2$X5.latitude                          6.257 1.04e-09 ***
## RE_DataNEW2$X6.longitude                         0.345  0.73011
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.443 on 390 degrees of freedom
## Multiple R-squared:  0.7258, Adjusted R-squared:  0.7215
## F-statistic:   172 on 6 and 390 DF,  p-value: < 2.2e-16
```

```
plot(BaseModel2)
```


Residuals vs Fitted
Residuals
Fitted values
lm(RE_DataNEW2$Y.house.price.of.unit.area ~ RE_DataNEW2$X1.transaction.date
274
346
20
10
0
−10
−20
10
20
30
40
50


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(RE_DataNEW2$Y.house.price.of.unit.area ~ RE_DataNEW2$X1.transaction.date
274
347
−3
−2
−1
0
1
2
3


Scale−Location
274
271
270
√|Standardized residuals|
0.0
0.5
1.0
1.5
10
20
30
40
50
Fitted values
lm(RE_DataNEW2$Y.house.price.of.unit.area ~ RE_DataNEW2$X1.transaction.date



18 variables were considered influential, and were removed from the dataset. The remaining values more closely follow a normal distribution, as shown by the Normal Q-Q plot.

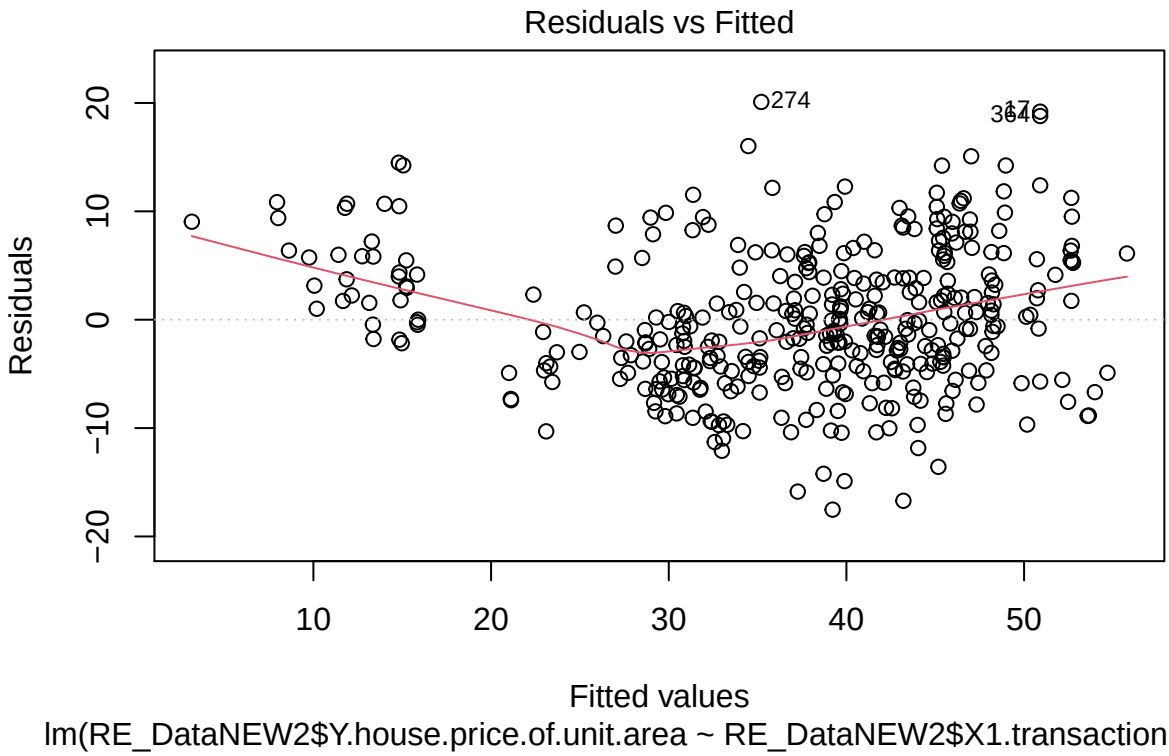
When the full model is fit with the new dataset, the adjusted R-squared value is 0.7215, which is much better than the model fit to the full dataset. Moving forward we will use RE_DataNEW2 as our dataset.

Before we begin assessing which variables to include in the model, lets consider the regression assumptions.

1. Linearity of the data.
2. Normality of residuals.
3. Homogeneity of residuals variance.
4. Independence of residuals error terms.

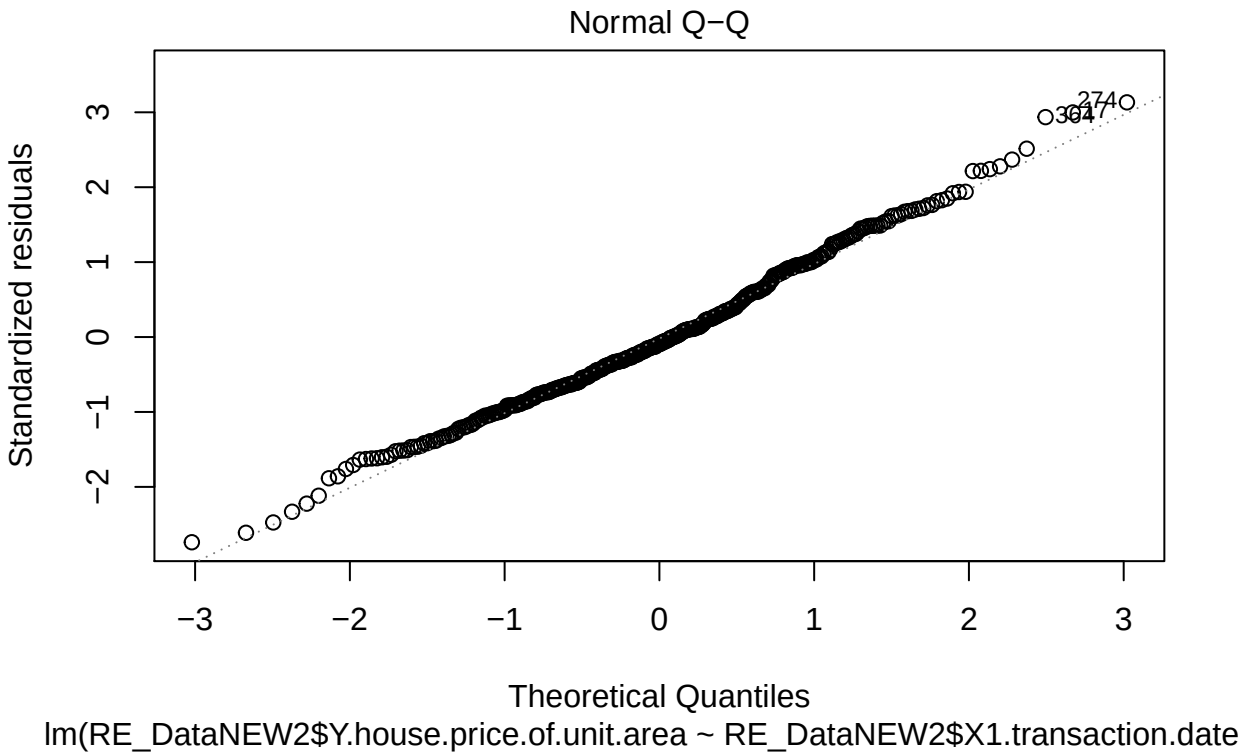
To test these assumptions, re-visit the diagnostics plots:

```
plot(BaseModel2,1)
```



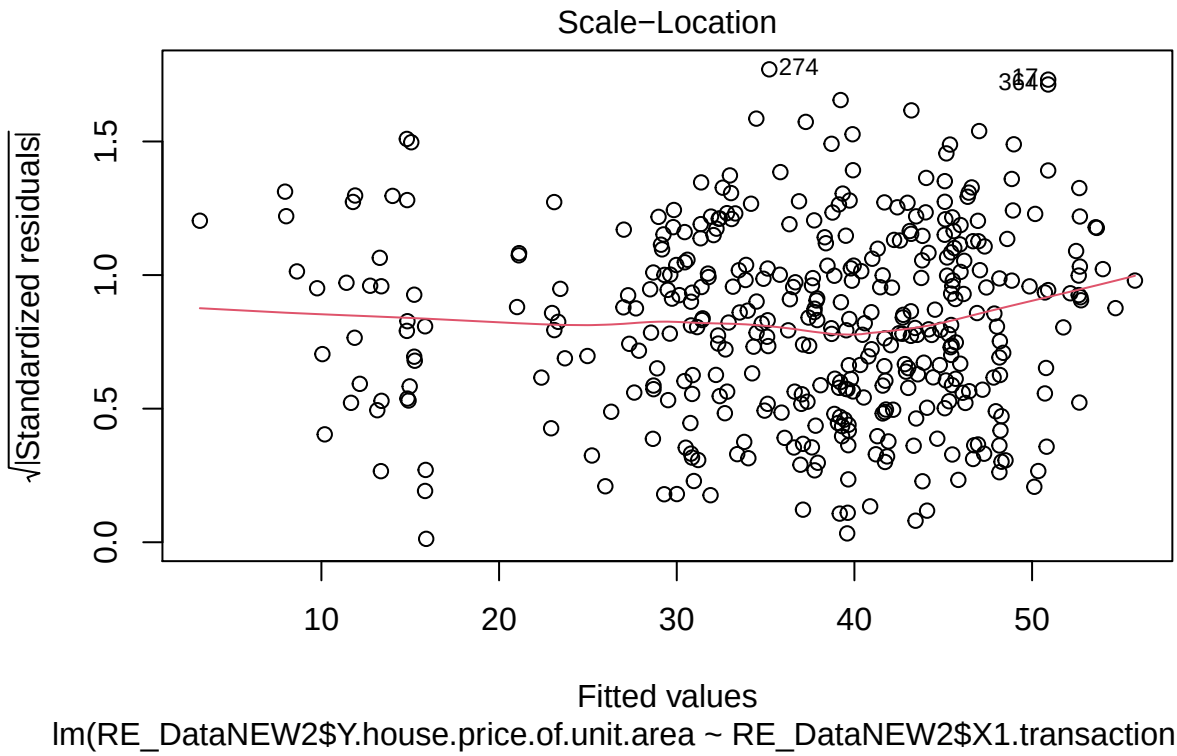
Based on the Residuals vs. Fitted plot there is a slight pattern, but residuals are fairly evenly distributed around zero. For the purposes of continuing with this analysis we will say that this model meets the assumption #1 that the data is linear.

```
plot(BaseModel2,2)
```



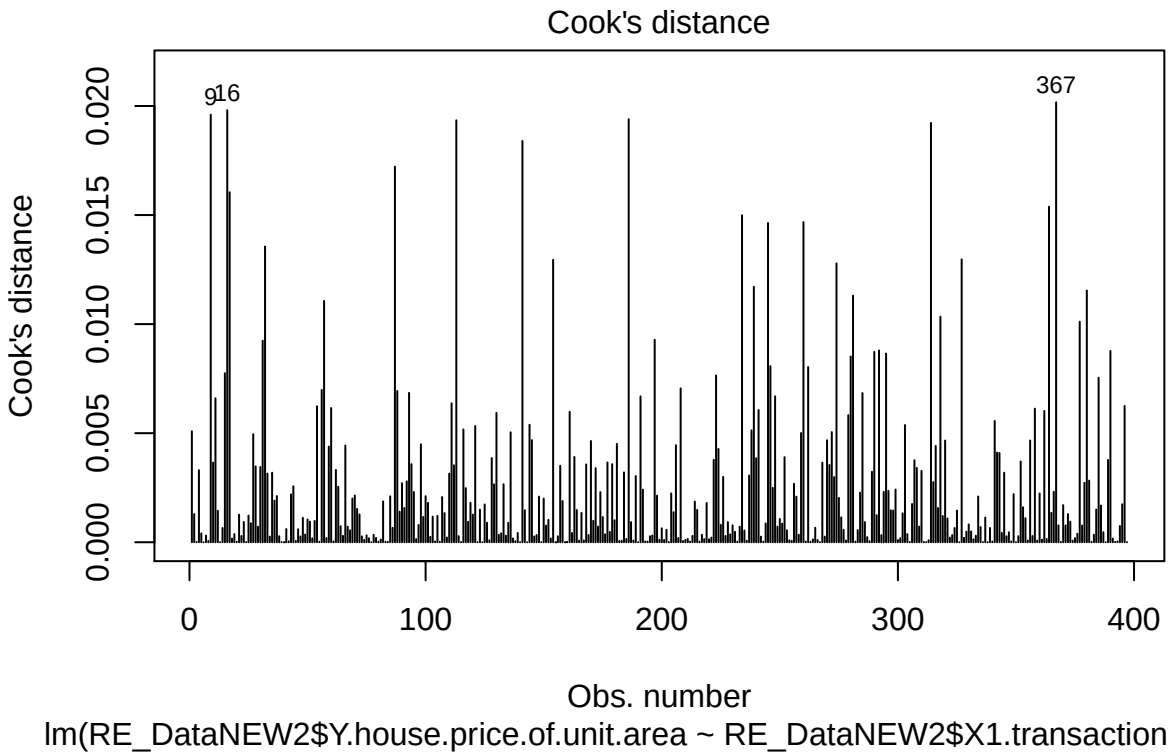
The Normal Q-Q plot shows the points are distributed in a close line, meaning that the data is normally distributed. Thus this data meets assumption #2.

```
plot(BaseModel2,3)
```



The scale-location plot shows a mostly horizontal line with equally spread points, which indicates that there is homoskedasticity of the variables so we can assume that model assumption #3

```
# Cook's distance
plot(BaseModel2, 4)
```



There are no additional leverage points with high Cooke's distance values, so it is confirmed that points of high leverage have been successfully removed from the model.

Now we have confirmed that the data meets model assumptions and outliers and leverage points have been removed. To improve the model further, let's return to the variable selection and variable relationships in the following section.

5.3 Model Adequacy

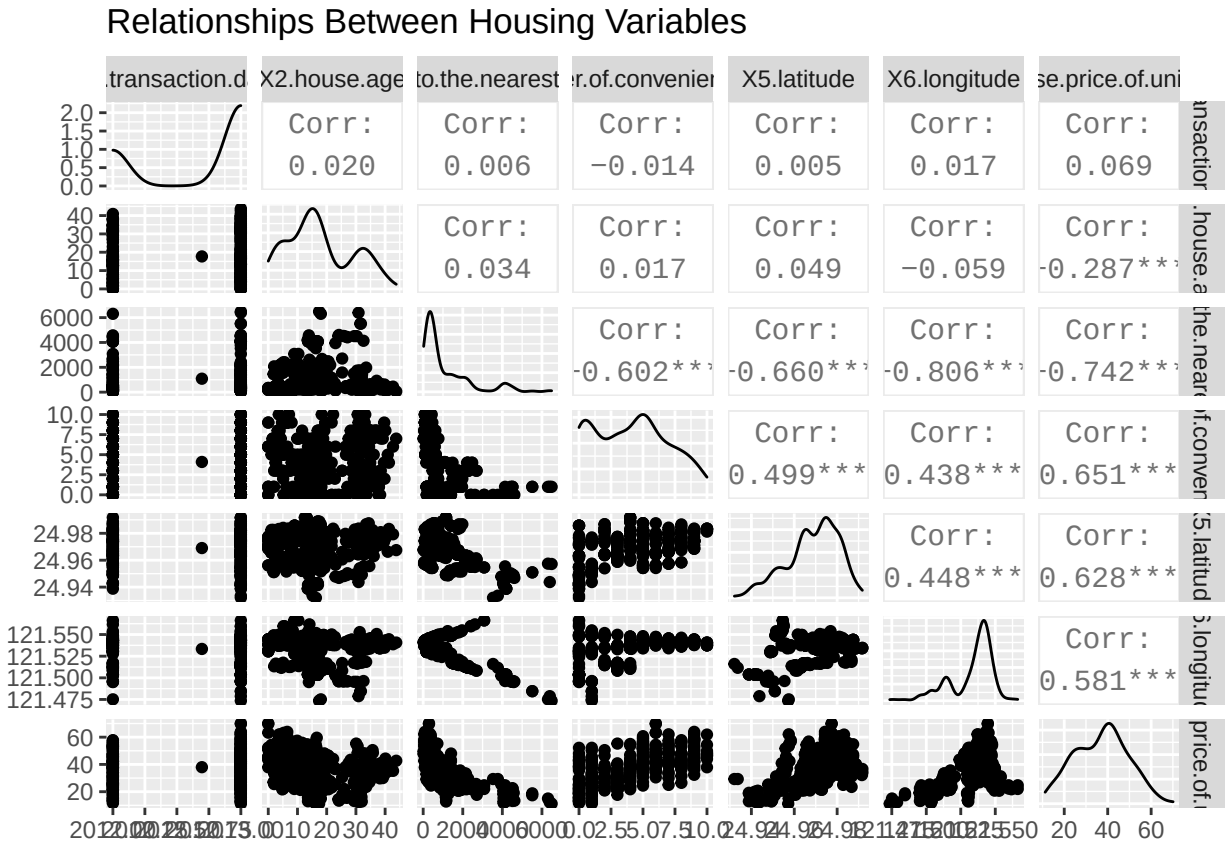
Pairs plots are useful to look at the relationships (correlation) between variables.

```
#a ggpairs plot gives a bit more information about the correlation between variables
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.0.3
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
ggpairs(RE_DataNEW2,title="Relationships Between Housing Variables")
```




```
#(1) Scatterplot Matrix to show correlation
cor(RE_DataNEW2)
```

```
##                                         X1.transaction.date X2.house.age
## X1.transaction.date                              1.000000000   0.01962029
## X2.house.age                                     0.019620287   1.00000000
## X3.distance.to.the.nearest.MRT.station           0.006095796   0.03401488
## X4.number.of.convenience.stores                 -0.014376228   0.01709575
## X5.latitude                                      0.004776867   0.04889947
## X6.longitude                                     0.017042191  -0.05902937
## Y.house.price.of.unit.area                       0.069464101  -0.28698235
##                                         X3.distance.to.the.nearest.MRT.station
## X1.transaction.date                                                 0.006095796
## X2.house.age                                                        0.034014882
## X3.distance.to.the.nearest.MRT.station                              1.000000000
## X4.number.of.convenience.stores                                    -0.601628062
## X5.latitude                                                        -0.660312838
## X6.longitude                                                       -0.806243197
## Y.house.price.of.unit.area                                         -0.741610329
##                                         X4.number.of.convenience.stores
## X1.transaction.date                                          -0.01437623
## X2.house.age                                                  0.01709575
## X3.distance.to.the.nearest.MRT.station                       -0.60162806
## X4.number.of.convenience.stores                               1.00000000
## X5.latitude                                                   0.49876051
```

```
## X6.longitude                                            0.43827434
## Y.house.price.of.unit.area                              0.65096112
##                                        X5.latitude X6.longitude
## X1.transaction.date                     0.004776867   0.01704219
## X2.house.age                            0.048899470  -0.05902937
## X3.distance.to.the.nearest.MRT.station -0.660312838  -0.80624320
## X4.number.of.convenience.stores         0.498760506   0.43827434
## X5.latitude                             1.000000000   0.44777383
## X6.longitude                            0.447773833   1.00000000
## Y.house.price.of.unit.area              0.628089174   0.58084850
##                                       Y.house.price.of.unit.area
## X1.transaction.date                                    0.0694641
## X2.house.age                                          -0.2869823
## X3.distance.to.the.nearest.MRT.station                -0.7416103
## X4.number.of.convenience.stores                        0.6509611
## X5.latitude                                            0.6280892
## X6.longitude                                           0.5808485
## Y.house.price.of.unit.area                             1.0000000
```

There does not appear to be strong linear relationships between any of the variables.

Check for multicollinearity between variables

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.0.3
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
vif(BaseModel2)
```

```
##                     RE_DataNEW2$X1.transaction.date
##                                            1.002362
##                           RE_DataNEW2$X2.house.age
##                                            1.012891
## RE_DataNEW2$X3.distance.to.the.nearest.MRT.station
##                                            4.736584
##        RE_DataNEW2$X4.number.of.convenience.stores
##                                            1.622880
##                            RE_DataNEW2$X5.latitude
##                                            1.897636
##                           RE_DataNEW2$X6.longitude
##                                            2.987266
```

None of these variables had VIF greater than 10. A VIF greater than 10 indicates multicollinearity, and suggests that a variable should be removed.

Next, we use stepwise regression to determine the best model. The stepwise model was also compared to the model output achieved from backward regression. Both methods created the same model output:

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.3
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.0.4     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
## v purrr   0.3.4
```

```
## Warning: package 'tibble' was built under R version 4.0.3
```

```
## Warning: package 'readr' was built under R version 4.0.3
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x data.table::between() masks dplyr::between()
## x readr::col_factor()   masks scales::col_factor()
## x purrr::discard()      masks scales::discard()
## x dplyr::filter()       masks stats::filter()
## x data.table::first()   masks dplyr::first()
## x dplyr::lag()          masks stats::lag()
## x data.table::last()    masks dplyr::last()
## x purrr::map()          masks maps::map()
## x car::recode()         masks dplyr::recode()
## x purrr::some()         masks car::some()
## x purrr::transpose()    masks data.table::transpose()
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.3
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.0.3
```

```r
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.0.3
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
# Stepwise regression model from the full model
step.model2<-step(BaseModel2,direction=c("both"),trace=TRUE)
```

```
## Start:  AIC=1486.1
## RE_DataNEW2$Y.house.price.of.unit.area ~ RE_DataNEW2$X1.transaction.date +
##     RE_DataNEW2$X2.house.age + RE_DataNEW2$X3.distance.to.the.nearest.MRT.station +
##     RE_DataNEW2$X4.number.of.convenience.stores + RE_DataNEW2$X5.latitude +
##     RE_DataNEW2$X6.longitude
##
##                                                 Df Sum of Sq   RSS    AIC
## - RE_DataNEW2$X6.longitude                       1       4.9 16192 1484.2
## <none>                                                       16188 1486.1
## - RE_DataNEW2$X1.transaction.date                1     381.9 16570 1493.4
## - RE_DataNEW2$X5.latitude                        1    1624.8 17812 1522.1
## - RE_DataNEW2$X3.distance.to.the.nearest.MRT.station  1    1851.9 18040 1527.1
## - RE_DataNEW2$X4.number.of.convenience.stores    1    3366.6 19554 1559.1
## - RE_DataNEW2$X2.house.age                       1    4931.1 21119 1589.7
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'Data' will be disregarded
```

```
##
## Step:  AIC=1484.22
## RE_DataNEW2$Y.house.price.of.unit.area ~ RE_DataNEW2$X1.transaction.date +
##     RE_DataNEW2$X2.house.age + RE_DataNEW2$X3.distance.to.the.nearest.MRT.station +
##     RE_DataNEW2$X4.number.of.convenience.stores + RE_DataNEW2$X5.latitude
##
##                                                 Df Sum of Sq   RSS    AIC
## <none>                                                       16192 1484.2
## + RE_DataNEW2$X6.longitude                       1       4.9 16188 1486.1
## - RE_DataNEW2$X1.transaction.date                1     386.0 16579 1491.6
## - RE_DataNEW2$X5.latitude                        1    1643.6 17836 1520.6
## - RE_DataNEW2$X4.number.of.convenience.stores    1    3364.5 19557 1557.2
## - RE_DataNEW2$X3.distance.to.the.nearest.MRT.station  1    4359.4 20552 1576.9
## - RE_DataNEW2$X2.house.age                       1    4948.0 21141 1588.1
```

```r
summary(step.model2)
```

```
##
```

```
## Call:
## lm(formula = RE_DataNEW2$Y.house.price.of.unit.area ~ RE_DataNEW2$X1.transaction.date +
##     RE_DataNEW2$X2.house.age + RE_DataNEW2$X3.distance.to.the.nearest.MRT.station +
##     RE_DataNEW2$X4.number.of.convenience.stores + RE_DataNEW2$X5.latitude,
##     Data = RE_DataNEW2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.6719  -4.4894  -0.5967   4.1572  20.0899
##
## Coefficients:
##                                                  Estimate Std. Error
## (Intercept)                                    -9.995e+03  1.669e+03
## RE_DataNEW2$X1.transaction.date                 2.139e+00  7.007e-01
## RE_DataNEW2$X2.house.age                       -3.207e-01  2.934e-02
## RE_DataNEW2$X3.distance.to.the.nearest.MRT.station -3.889e-03  3.790e-04
## RE_DataNEW2$X4.number.of.convenience.stores     1.268e+00  1.407e-01
## RE_DataNEW2$X5.latitude                         2.295e+02  3.643e+01
##                                                t value Pr(>|t|)
## (Intercept)                                     -5.990 4.77e-09 ***
## RE_DataNEW2$X1.transaction.date                  3.053  0.00242 **
## RE_DataNEW2$X2.house.age                       -10.931  < 2e-16 ***
## RE_DataNEW2$X3.distance.to.the.nearest.MRT.station -10.260  < 2e-16 ***
## RE_DataNEW2$X4.number.of.convenience.stores      9.013  < 2e-16 ***
## RE_DataNEW2$X5.latitude                          6.300 8.03e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.435 on 391 degrees of freedom
## Multiple R-squared:  0.7257, Adjusted R-squared:  0.7222
## F-statistic: 206.9 on 5 and 391 DF,  p-value: < 2.2e-16
```

Backward regression:

```
#backward regression model from the full model
step.model.back<-step(BaseModel2,direction=c("backward"),trace=TRUE)
```

```
## Start:  AIC=1486.1
## RE_DataNEW2$Y.house.price.of.unit.area ~ RE_DataNEW2$X1.transaction.date +
##     RE_DataNEW2$X2.house.age + RE_DataNEW2$X3.distance.to.the.nearest.MRT.station +
##     RE_DataNEW2$X4.number.of.convenience.stores + RE_DataNEW2$X5.latitude +
##     RE_DataNEW2$X6.longitude
##
##                                                 Df Sum of Sq   RSS    AIC
## - RE_DataNEW2$X6.longitude                       1       4.9 16192 1484.2
## <none>                                                       16188 1486.1
## - RE_DataNEW2$X1.transaction.date                1     381.9 16570 1493.4
## - RE_DataNEW2$X5.latitude                        1    1624.8 17812 1522.1
## - RE_DataNEW2$X3.distance.to.the.nearest.MRT.station  1    1851.9 18040 1527.1
## - RE_DataNEW2$X4.number.of.convenience.stores    1    3366.6 19554 1559.1
## - RE_DataNEW2$X2.house.age                       1    4931.1 21119 1589.7

## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'Data' will be disregarded
```

```
##
## Step:  AIC=1484.22
## RE_DataNEW2$Y.house.price.of.unit.area ~ RE_DataNEW2$X1.transaction.date +
##     RE_DataNEW2$X2.house.age + RE_DataNEW2$X3.distance.to.the.nearest.MRT.station +
##     RE_DataNEW2$X4.number.of.convenience.stores + RE_DataNEW2$X5.latitude
##
##                                                 Df Sum of Sq   RSS    AIC
## <none>                                                       16192 1484.2
## - RE_DataNEW2$X1.transaction.date                1     386.0 16579 1491.6
## - RE_DataNEW2$X5.latitude                        1    1643.6 17836 1520.6
## - RE_DataNEW2$X4.number.of.convenience.stores    1    3364.5 19557 1557.2
## - RE_DataNEW2$X3.distance.to.the.nearest.MRT.station  1    4359.4 20552 1576.9
## - RE_DataNEW2$X2.house.age                       1    4948.0 21141 1588.1
```

```
summary(step.model.back)
```

```
##
## Call:
## lm(formula = RE_DataNEW2$Y.house.price.of.unit.area ~ RE_DataNEW2$X1.transaction.date +
##     RE_DataNEW2$X2.house.age + RE_DataNEW2$X3.distance.to.the.nearest.MRT.station +
##     RE_DataNEW2$X4.number.of.convenience.stores + RE_DataNEW2$X5.latitude,
##     Data = RE_DataNEW2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.6719  -4.4894  -0.5967   4.1572  20.0899
##
## Coefficients:
##                                                   Estimate Std. Error
## (Intercept)                                     -9.995e+03  1.669e+03
## RE_DataNEW2$X1.transaction.date                  2.139e+00  7.007e-01
## RE_DataNEW2$X2.house.age                        -3.207e-01  2.934e-02
## RE_DataNEW2$X3.distance.to.the.nearest.MRT.station -3.889e-03  3.790e-04
## RE_DataNEW2$X4.number.of.convenience.stores      1.268e+00  1.407e-01
## RE_DataNEW2$X5.latitude                          2.295e+02  3.643e+01
##                                                 t value Pr(>|t|)
## (Intercept)                                      -5.990 4.77e-09 ***
## RE_DataNEW2$X1.transaction.date                   3.053  0.00242 **
## RE_DataNEW2$X2.house.age                        -10.931  < 2e-16 ***
## RE_DataNEW2$X3.distance.to.the.nearest.MRT.station -10.260  < 2e-16 ***
## RE_DataNEW2$X4.number.of.convenience.stores       9.013  < 2e-16 ***
## RE_DataNEW2$X5.latitude                           6.300 8.03e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.435 on 391 degrees of freedom
## Multiple R-squared:  0.7257, Adjusted R-squared:  0.7222
## F-statistic: 206.9 on 5 and 391 DF,  p-value: < 2.2e-16
```

Based on stepwise regression, the most well-fitted model is: Y.house.price.of.unit.area=X1.transaction.date+X2.house.age+X3

A second model which could be considered is: Y.house.price.of.unit.area=X1.transaction.date+X2.house.age+X3.distance.to.t

```
Simple_REmod<-lm(Y.house.price.of.unit.area~X1.transaction.date+X2.house.age+X3.distance.to.the.nearest
summary(Simple_REmod)
```

```
##
## Call:
## lm(formula = Y.house.price.of.unit.area ~ X1.transaction.date +
##     X2.house.age + X3.distance.to.the.nearest.MRT.station + X4.number.of.convenience.stores,
##     data = RE_DataNEW2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.0838  -4.7382  -0.6715   4.6598  21.0408
##
## Coefficients:
##                                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)                            -4.374e+03  1.478e+03  -2.959  0.00327
## X1.transaction.date                     2.194e+00  7.344e-01   2.988  0.00299
## X2.house.age                           -3.044e-01  3.063e-02  -9.937  < 2e-16
## X3.distance.to.the.nearest.MRT.station -5.140e-03  3.383e-04 -15.195  < 2e-16
## X4.number.of.convenience.stores         1.416e+00  1.455e-01   9.730  < 2e-16
##
## (Intercept)                            **
## X1.transaction.date                    **
## X2.house.age                           ***
## X3.distance.to.the.nearest.MRT.station ***
## X4.number.of.convenience.stores        ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.745 on 392 degrees of freedom
## Multiple R-squared:  0.6978, Adjusted R-squared:  0.6947
## F-statistic: 226.3 on 4 and 392 DF,  p-value: < 2.2e-16
```

This model has no consideration for latitude or longitude. It could be argued that latitude and longitude should not be considered in linear regression modeling, as there are other factors represented by location that aren't reflected in this analysis which merely uses the number values of latitude and longitude. However, excluding latitude causes the adjusted r-squared value to decrease, meaning that this model does not predict house price per unit area as accurately as the model which does include latitude.

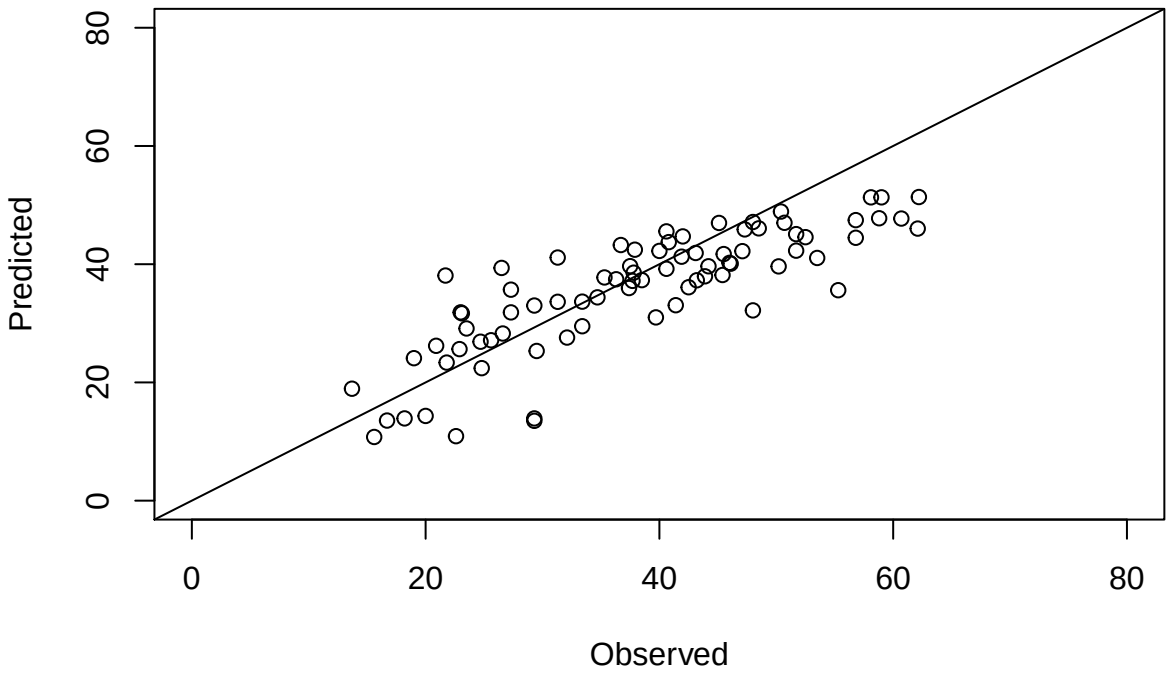
We will continue using the step.model2 model outcome from the stepwise regression: Y.house.price.of.unit.area=X1.transactior

5.4 Model Validation

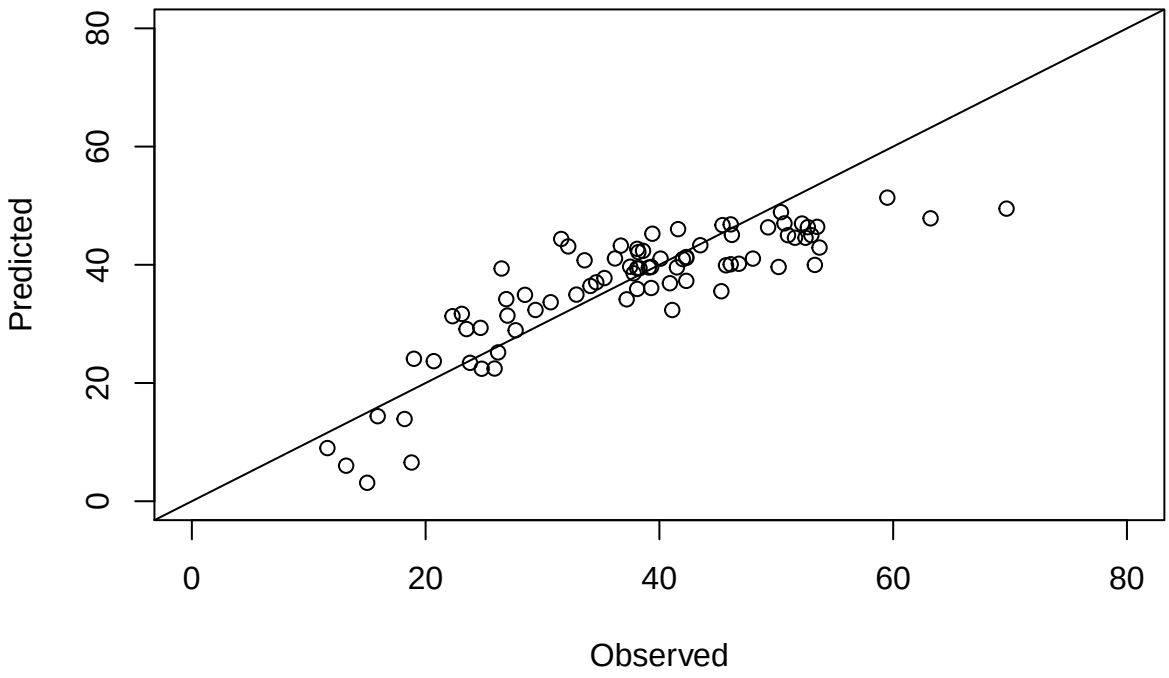
Validate stepmodel by splitting into training and test data

```
#Train 5 times
set.seed(71168)

for(i in 1:5){
nsamp=ceiling(0.8*length(RE_DataNEW2$Y.house.price.of.unit.area))
training_samps=sample(c(1:length(RE_DataNEW2$Y.house.price.of.unit.area)),nsamp)
training_samps=sort(training_samps)
train_data  <- RE_DataNEW2[training_samps, ]
```

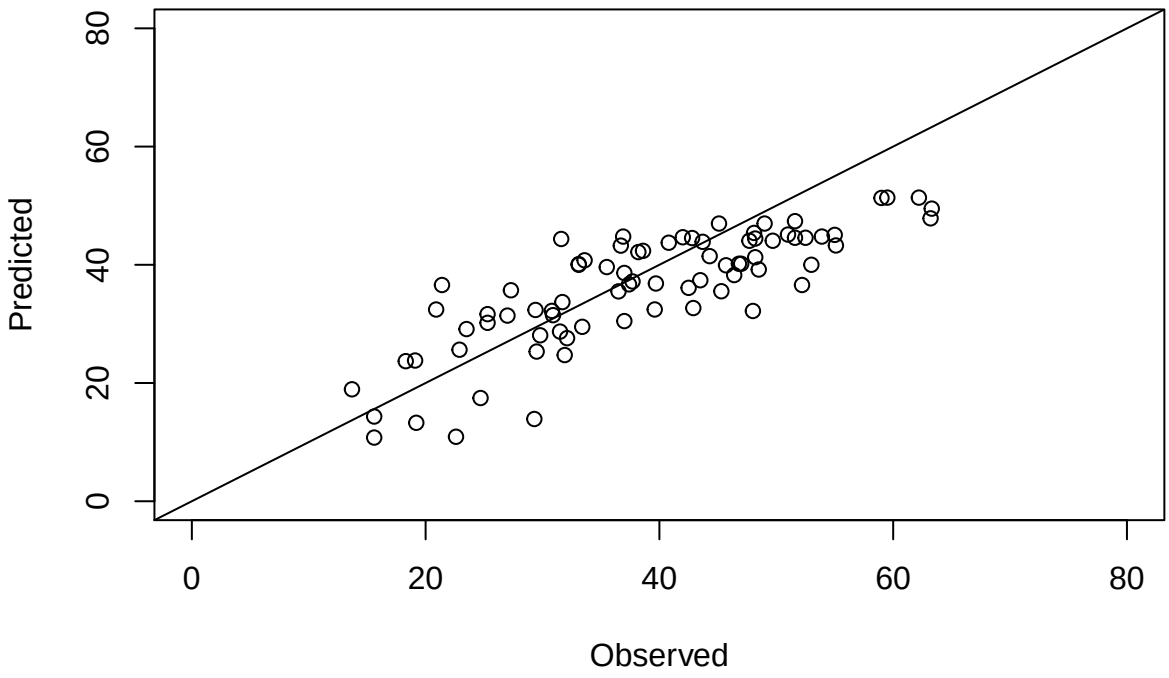
```
test_data <-   RE_DataNEW2[-training_samps, ]

train.lm <- lm(Y.house.price.of.unit.area~X1.transaction.date+X2.house.age+X3.distance.to.the.nearest.M

preds <- predict(train.lm,test_data)

plot(test_data$Y.house.price.of.unit.area,preds,xlim=c(0,80),ylim=c(0,80),xlab = "Observed",ylab = "Pred
abline(c(0,1))

RMSE<-sqrt(sum((preds-test_data$Y.house.price.of.unit.area)^2)/length(preds))


print(c(i,RMSE))

}
```



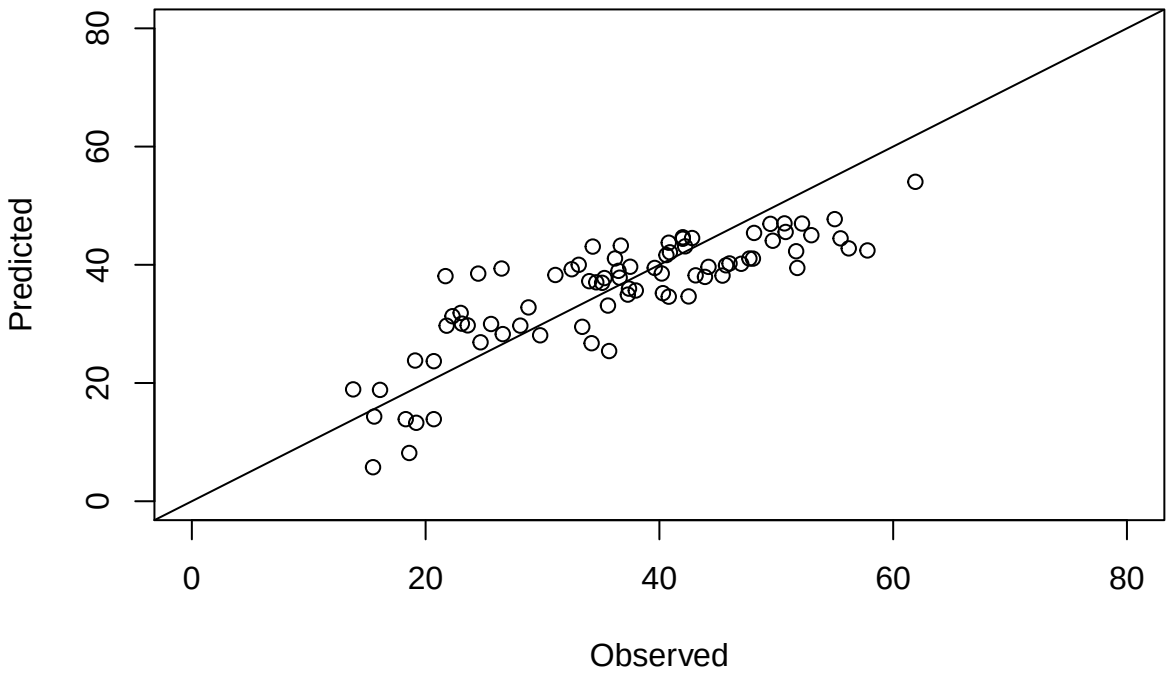

```
## [1] 1.000000 7.460026
```



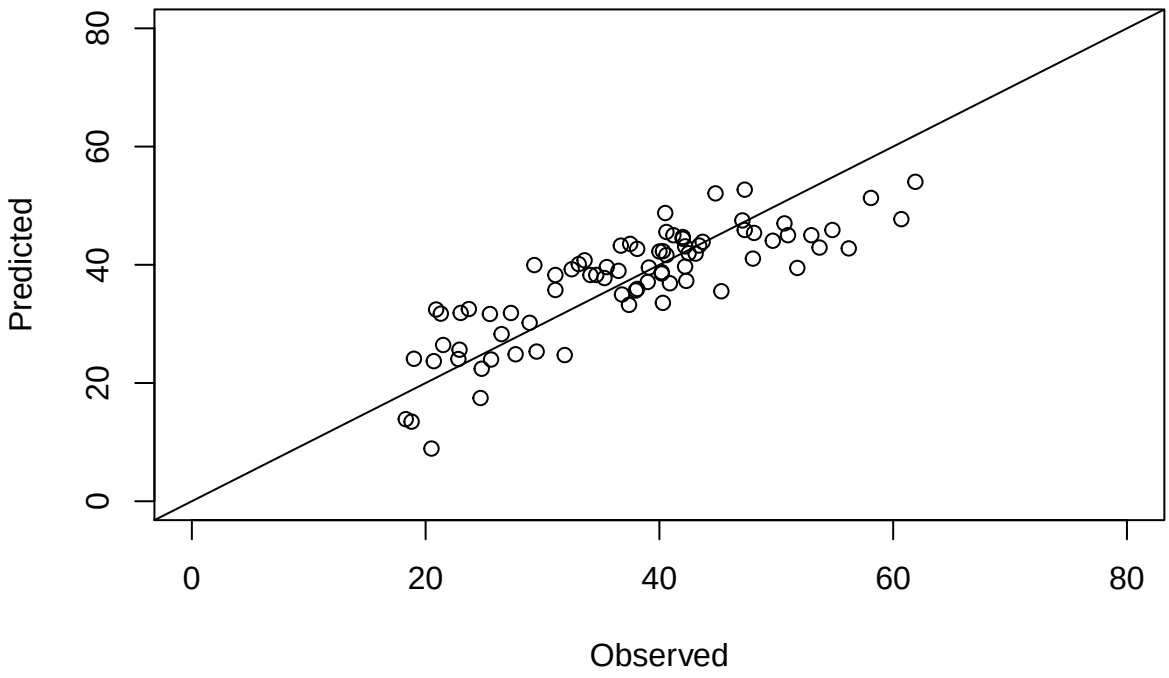

```
## [1] 2.00000 6.50775
```




```
## [1] 3.000000 7.413474
```




```
## [1] 4.000000 6.605285
```




```
## [1] 5.000000 5.947088
```

Lower values of RMSE indicate better fit. In general, points should ideally fall along the centre line of the residuals vs. fitted plot. All RMSE values were below 8. Because the range of the dependent variable in this case is large (roughly 11-70 house price per unit area), 8 is a relatively small RMSE, thus the trained model was a relatively good predictor of the test data.

6 - Conclusion

The final model included the following variables to predict house price per unit area: X1.transaction.date X2.house.age X3.distance.to.the.nearest.MRT.station X4.number.of.convenience.stores X5.latitude

The equation for predicting house price per unit area is:

```
Y.house.price.of.unit.area=2.139e00*X1.transaction.date-3.207e-01*X2.house.age-3.889e-03*X3.distance.to
```

The values of these coefficients is logical. For transaction date, larger value (more recent transaction) has a positive influence on the house price. For house age, larger value (older house) has a negative influence on the house price. For distance to MRT station, larger value (higher distance) has a negative influence on the house price. For number of convenience stores, larger value (more nearby stores) has positive influence on house price. For latitude, larger value (closer proximity to capitol city) has positive influence on house price.

The final R squared value for this model was 0.7222. This means that the model can be used to predict house price per unit area, however it is not always extremely accurate. An R-squared value above 0.9 would be more ideal. There are a few reasons why this model may not be the best predictor for house price per

unit area in Taiwan. Many factors influence housing price. This model did not consider factors such as updated appliances, condition of the house, lot size, proximity to schools, appealing views, and more. In addition, the housing market fluctuates greatly over time. House price per unit area could depend on how well the economy is doing in the month that the house sold, inflation or deflation of currency over time, the current value of currency in comparison to other countries, and demand for housing. In order to build a more accurate model to predict house price per unit area, a more robust dataset should be used, with a larger set of variables to consider.

Source: https://www.opendoor.com/w/blog/factors-that-influence-home-value, https://www.toptal.com/finance/real-estate/real-estate-valuation