New Taipei City Housing Valuation

Ismail Olasege

6/11/2020

New Taipei City Housing Valuation Project

This project is about the housing valuation of New Taipei City neighborhood data are for a neighborhood of New Taipei City. New Taipei City is the most populous city in Taiwan. It is well known that the housing prices in Taipei is expensive and for the reason, the residents of the city find it difficult to own a house.

There are factors that contributes the house valuation such as date, age, distance, stores, latitude, and longitude. Where

a. Date: This represents the date of the transaction.
b. Age: This represents the age of the house in years.
c. Distance: This represents the distance in meters from the unit to the nearest metro station.
d. Stores: This represents the number of conveniences stores within the radius of 1000 meters of the unit.
e. Latitude: This represents the latitude for the geographical location of the house.
f. Longitude: This represents the longitude for the geographical location of the house.
g. Price: This represents the price of the house per square foot.

The goal of the project is to build a reliable statistical model that can predict the housing price in New Taipei City, Taiwan.

```
# data importation
hv <- read.csv("training_data.csv")
head(hv)
```

```
##    No Year  Age   Distance Stores Latitude Longitude Price log_dist
## 1 344 2013 33.5  563.28540      8 24.98223  121.5360  46.6 6.333786
## 2 373 2013 33.9  157.60520      7 24.96628  121.5420  41.5 5.060093
## 3 117 2013 30.9 6396.28300      1 24.94375  121.4788  12.2 8.763472
## 4 288 2013 19.2  461.10160      5 24.95425  121.5399  32.9 6.133618
## 5 330 2013 13.6 4197.34900      0 24.93885  121.5038  19.2 8.342208
## 6 386 2013 18.3   82.88643     10 24.98300  121.5403  46.6 4.417471
```

Data Exploration

```
# the data structure.

str(hv)
```

```
## 'data.frame':    250 obs. of  9 variables:
##  $ No       : int  344 373 117 288 330 386 205 383 41 188 ...
##  $ Year     : int  2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 ...
##  $ Age      : num  33.5 33.9 30.9 19.2 13.6 18.3 18 16.3 13.6 8.9 ...
##  $ Distance : num  563 158 6396 461 4197 ...
##  $ Stores   : int  8 7 1 5 0 10 1 0 0 0 ...
##  $ Latitude : num  25 25 24.9 25 24.9 ...
##  $ Longitude: num  122 122 121 122 122 ...
##  $ Price    : num  46.6 41.5 12.2 32.9 19.2 46.6 26.6 29.3 15.9 22 ...
##  $ log_dist : num  6.33 5.06 8.76 6.13 8.34 ...
```

There are `9` number of columns and `250` number of rows

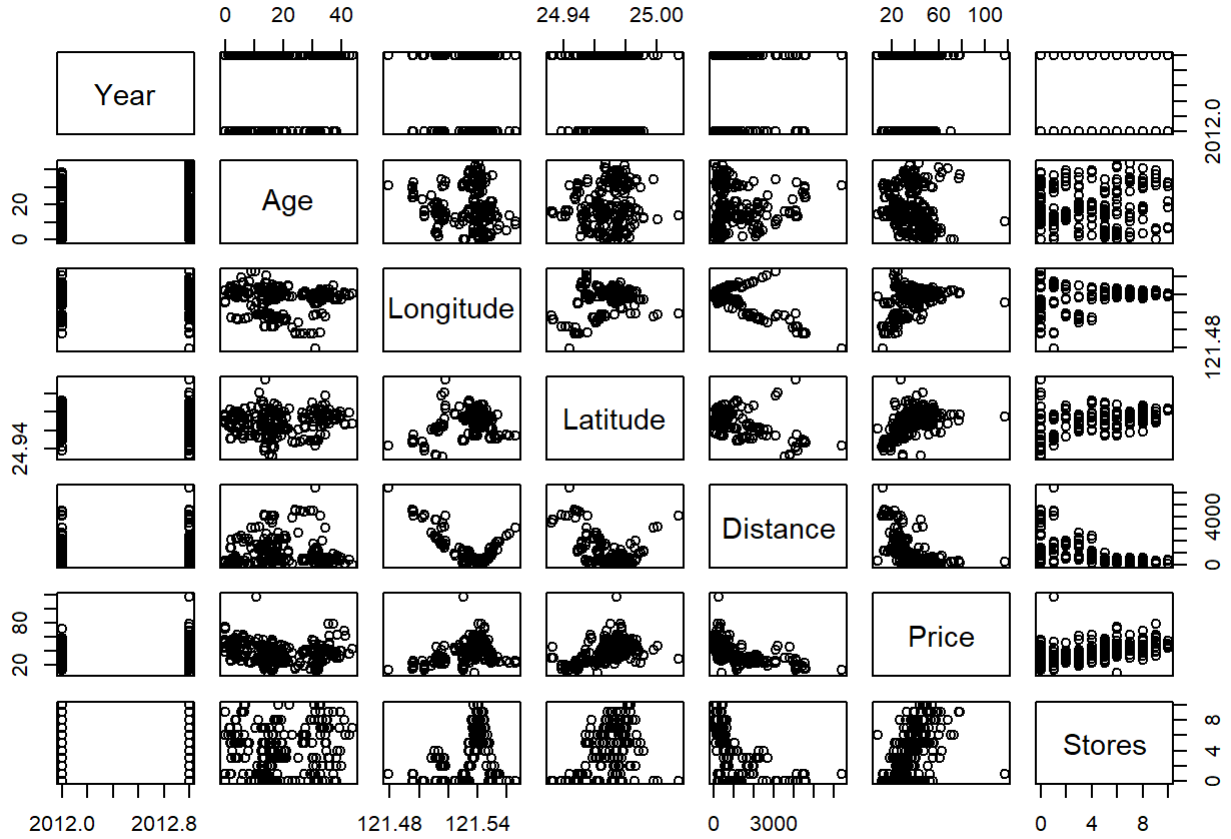
The variable names are `No, Year, Age, Distance, Stores, Latitude, Longitude, Price, log_dist`

```
# statistical properties of the data set.
summary(hv)
```

```
##        No             Year           Age           Distance
##  Min.   :  3.0   Min.   :2012   Min.   : 0.00   Min.   :  23.38
##  1st Qu.:115.5   1st Qu.:2012   1st Qu.:10.57   1st Qu.: 325.24
##  Median :222.0   Median :2013   Median :16.40   Median : 554.96
##  Mean   :216.5   Mean   :2013   Mean   :18.22   Mean   :1140.93
##  3rd Qu.:316.8   3rd Qu.:2013   3rd Qu.:28.98   3rd Qu.:1695.35
##  Max.   :412.0   Max.   :2013   Max.   :43.80   Max.   :6396.28
##      Stores          Latitude       Longitude         Price
##  Min.   : 0.000   Min.   :24.93   Min.   :121.5   Min.   :  7.60
##  1st Qu.: 1.000   1st Qu.:24.96   1st Qu.:121.5   1st Qu.: 26.50
##  Median : 4.000   Median :24.97   Median :121.5   Median : 37.25
##  Mean   : 3.928   Mean   :24.97   Mean   :121.5   Mean   : 37.13
##  3rd Qu.: 6.000   3rd Qu.:24.98   3rd Qu.:121.5   3rd Qu.: 45.35
##  Max.   :10.000   Max.   :25.01   Max.   :121.6   Max.   :117.50
##     log_dist
##  Min.   :3.152
##  1st Qu.:5.785
##  Median :6.319
##  Mean   :6.465
##  3rd Qu.:7.435
##  Max.   :8.763
```

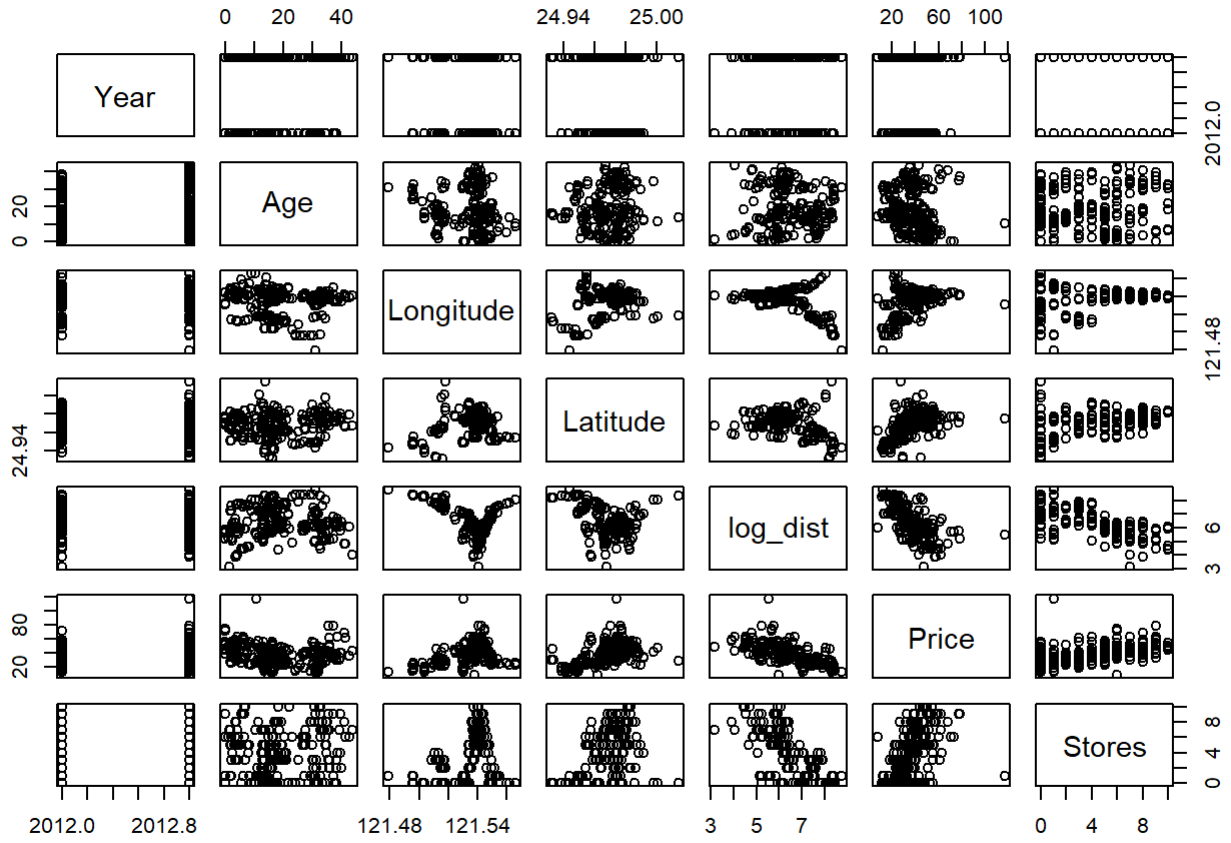
```
# The pair plot shows the relationship between the variables

plot(~Year+Age+Longitude+Latitude+Distance+Price+Stores, data = hv)
```




```
# Taking the log of distance gives better relationship

plot(~Year+Age+Longitude+Latitude+log_dist+Price+Stores, data = hv)
```




```
hist(hv$'Price', main = "Housing Prices", xlab = "Price", col = "blue")
```


```
library(dplyr)
```

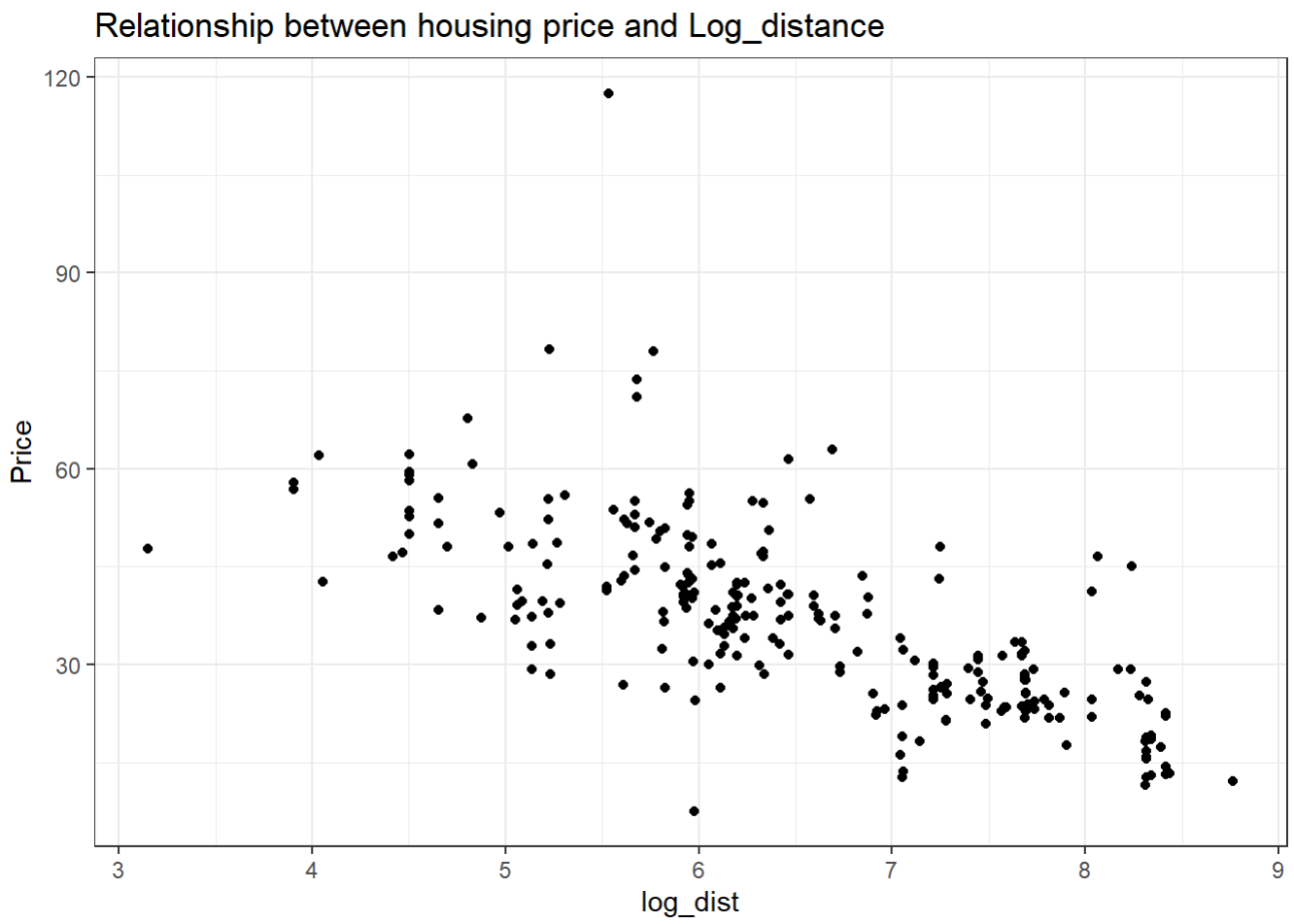
```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)

ggplot(hv, aes(x = log_dist, y = Price)) +
  geom_point() +
  theme_bw() +
  ggtitle("Relationship between housing price and Log_distance")
```




```
ggplot(hv, aes(x = Age, y = Price)) +
  geom_point() +
  theme_bw() +
  ggtitle("Relationship between housing price and Age")
```


```
boxplot(hv$Price, main = "Housing price distribution", xlab = "Price", ylab = "Frequency", col =
"red")
```

Checking for the outlier case.

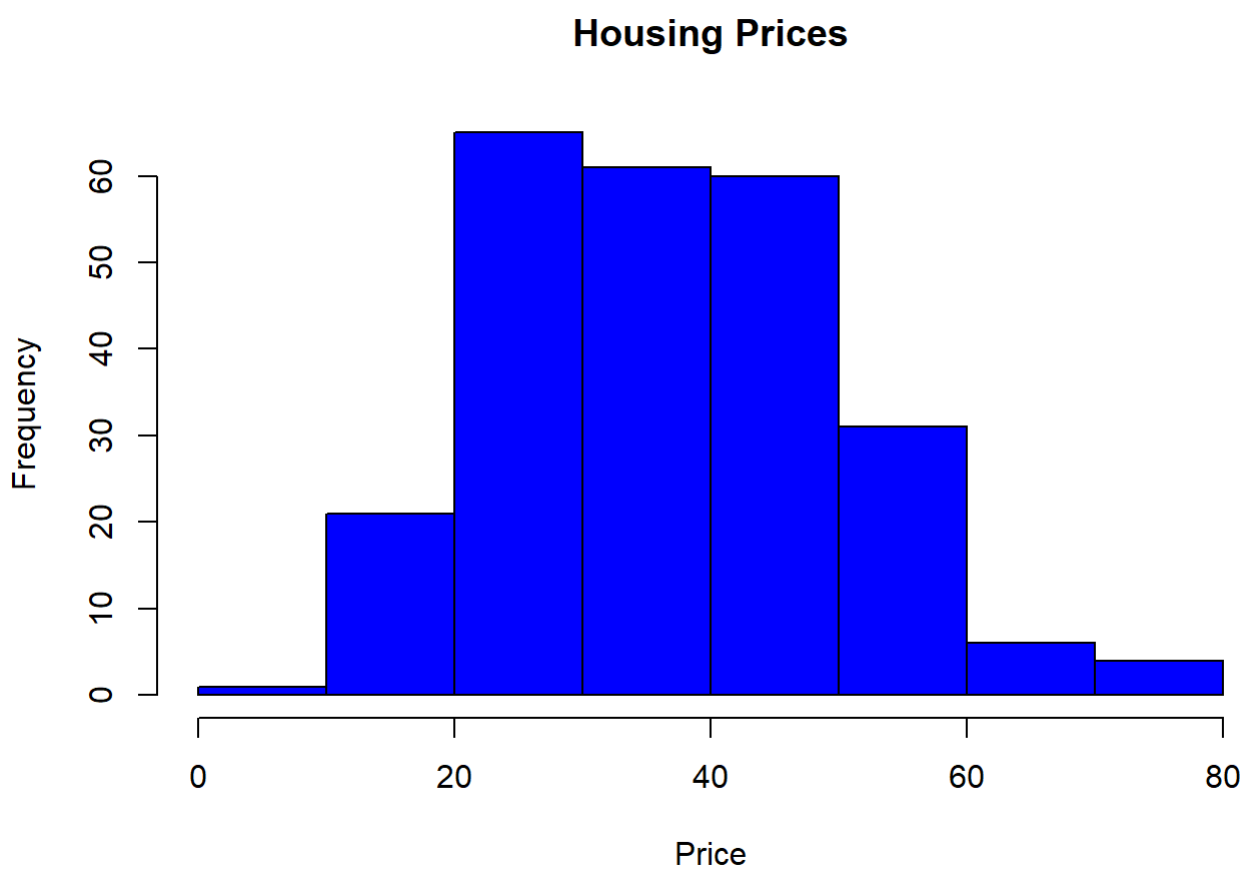
```
subset(hv, Price> 80)
```

```
##     No Year  Age Distance Stores Latitude Longitude Price log_dist
## 88 271 2013 10.8 252.5822      1  24.9746  121.5305 117.5 5.531737
```

```
# Removing the outlier
hv01 <- hv[-c(88),]
str(hv01)
```

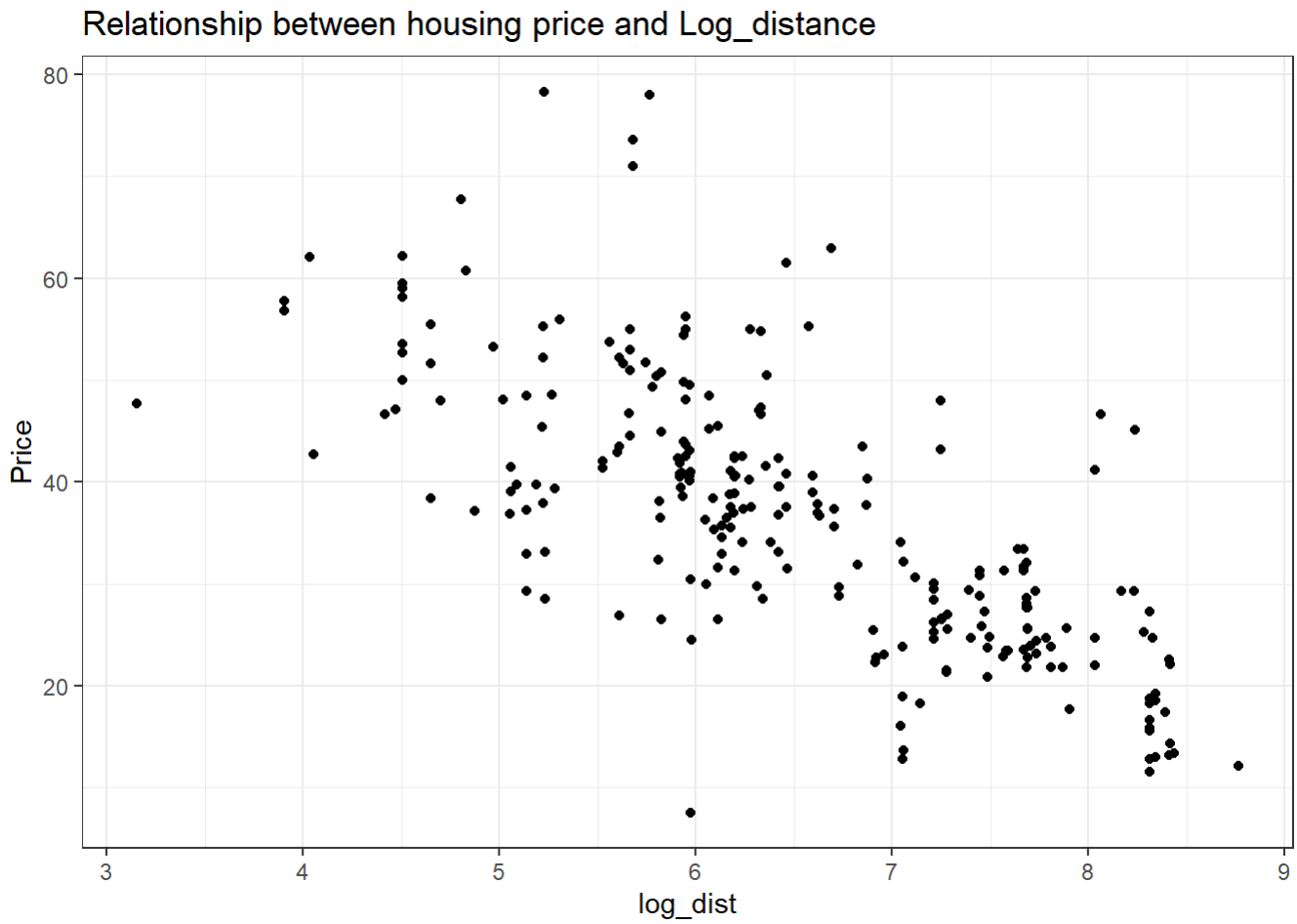
```
## 'data.frame':    249 obs. of  9 variables:
##  $ No       : int  344 373 117 288 330 386 205 383 41 188 ...
##  $ Year     : int  2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 ...
##  $ Age      : num  33.5 33.9 30.9 19.2 13.6 18.3 18 16.3 13.6 8.9 ...
##  $ Distance : num  563 158 6396 461 4197 ...
##  $ Stores   : int  8 7 1 5 0 10 1 0 0 0 ...
##  $ Latitude : num  25 25 24.9 25 24.9 ...
##  $ Longitude: num  122 122 121 122 122 ...
##  $ Price    : num  46.6 41.5 12.2 32.9 19.2 46.6 26.6 29.3 15.9 22 ...
##  $ log_dist : num  6.33 5.06 8.76 6.13 8.34 ...
```

```
hist(hv01$'Price', main = "Housing Prices", xlab = "Price", col = "blue")
```




```
boxplot(hv01$Price, main = "Housing price distribution", xlab = "Price", ylab = "Frequency", col
= "red")
```


```
ggplot(hv01, aes(x = log_dist, y = Price)) +
  geom_point() +
  theme_bw() +
  ggtitle("Relationship between housing price and Log_distance")
```



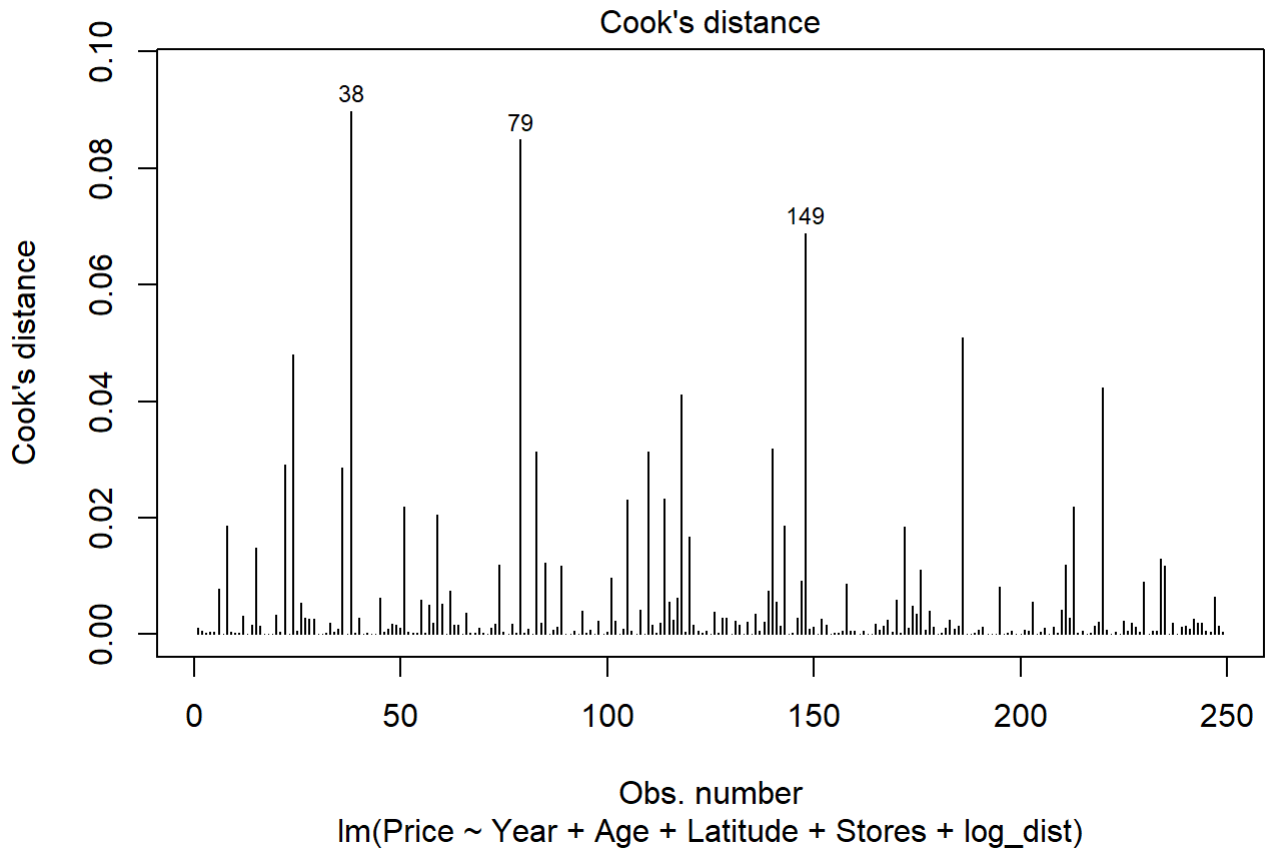
Fitting the statistical model

```
model_fit <- lm(Price ~ Year+Age+Latitude+Stores+log_dist, data = hv01)
summary(model_fit)
```

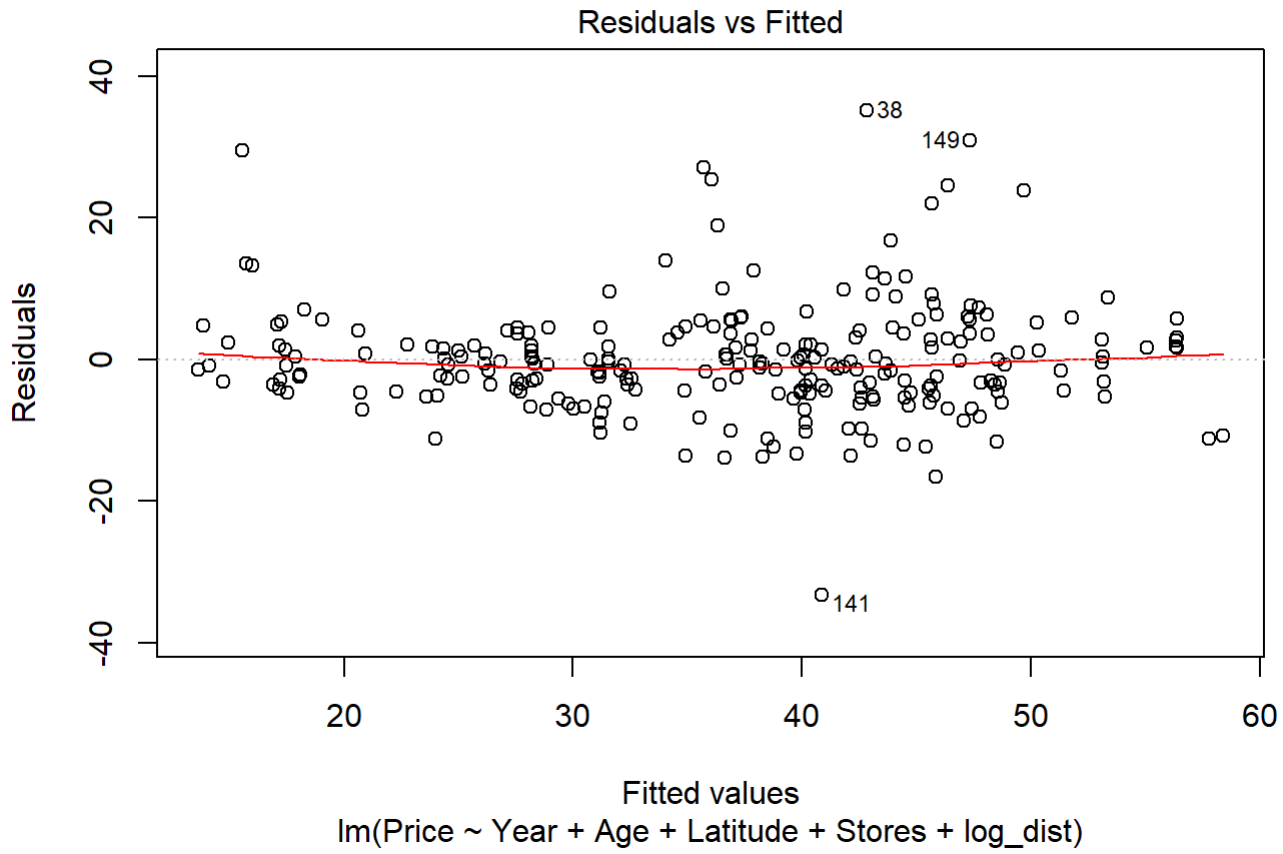
```
##
## Call:
## lm(formula = Price ~ Year + Age + Latitude + Stores + log_dist,
##     data = hv01)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.286  -4.346  -0.643   3.493  35.154
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.362e+04  2.570e+03  -5.300 2.60e-07 ***
## Year         3.330e+00  1.121e+00   2.970  0.00328 **
## Age         -1.848e-01  4.579e-02  -4.036 7.28e-05 ***
## Latitude     2.802e+02  4.552e+01   6.155 3.09e-09 ***
## Stores       6.845e-01  2.398e-01   2.855  0.00467 **
## log_dist    -5.672e+00  6.514e-01  -8.706 4.89e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.076 on 243 degrees of freedom
## Multiple R-squared:  0.6321, Adjusted R-squared:  0.6246
## F-statistic: 83.51 on 5 and 243 DF,  p-value: < 2.2e-16
```

Outlier strategy to remove the outliers

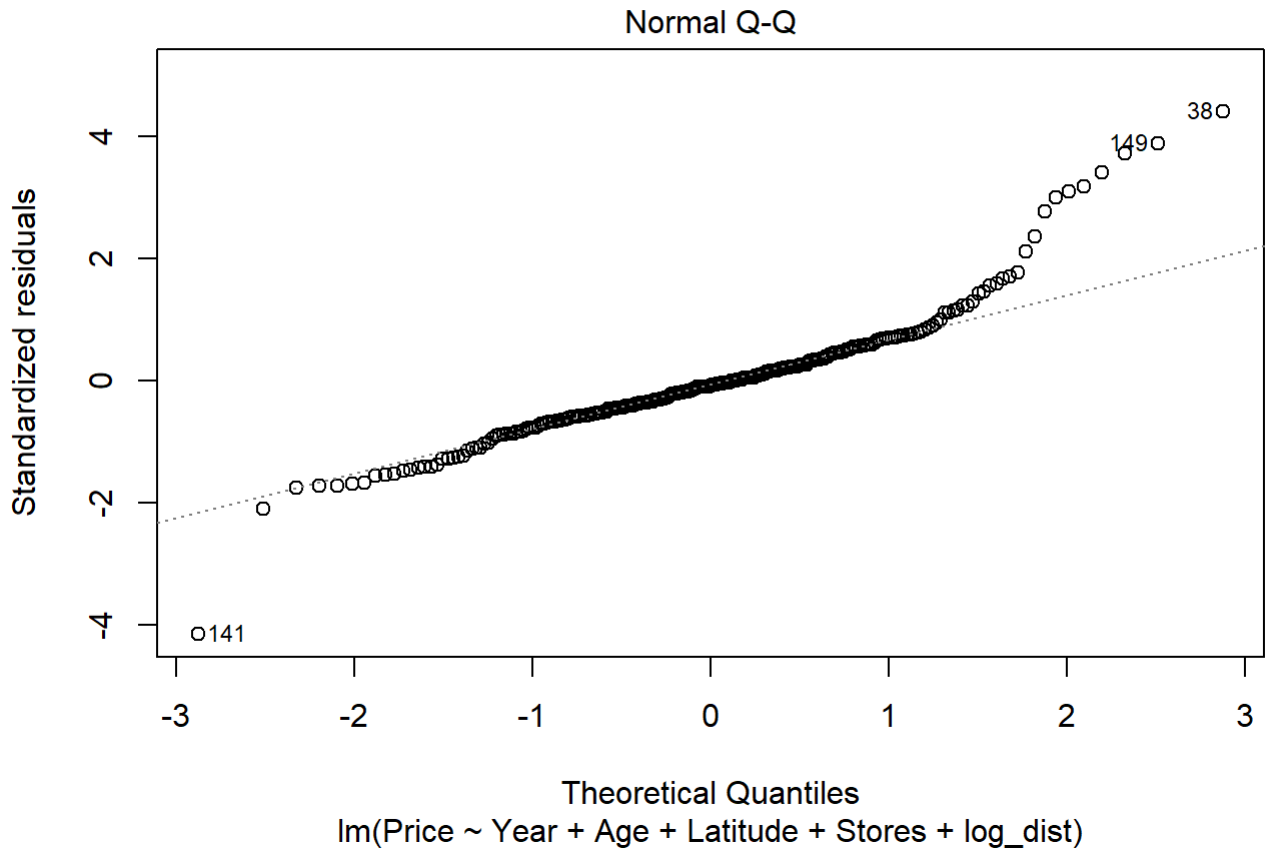
```
#Cook's Distance
plot(model_fit, which=4)
```




```
plot(model_fit, 1)
```




```
plot(model_fit, 2)
```




```
#Outlier Removal
hv02 <- hv01[-c(24, 38,79, 141,24, 149, 116,138,145, 216),]
row.names(hv02) <- 1:nrow(hv02)
```

Model re-fit

```
model_refit <- lm(Price ~ Year+Age+Latitude+Stores+log_dist, data = hv02)
summary(model_refit)
```

```
##
## Call:
## lm(formula = Price ~ Year + Age + Latitude + Stores + log_dist,
##     data = hv02)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.531  -3.688  -0.246   3.490  32.223
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.393e+04  2.386e+03  -5.836 1.77e-08 ***
## Year         2.999e+00  1.040e+00   2.885  0.00428 **
## Age         -2.189e-01  4.334e-02  -5.051 8.84e-07 ***
## Latitude     3.190e+02  4.272e+01   7.467 1.61e-12 ***
## Stores       5.943e-01  2.260e-01   2.629  0.00913 **
## log_dist    -5.607e+00  6.065e-01  -9.245  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.431 on 234 degrees of freedom
## Multiple R-squared:  0.6731, Adjusted R-squared:  0.6661
## F-statistic: 96.35 on 5 and 234 DF,  p-value: < 2.2e-16
```

Model Evaluation

```
library(broom)
lm(Price ~ Year+Age+Latitude+Stores+log_dist, data = hv02)%>%
  augment()%>%
  mutate(residual  = Price - .fitted)%>%
  summarize(r_sqd = 1 - var(residual)/var(Price))
```

```
## # A tibble: 1 x 1
##   r_sqd
##   <dbl>
## 1 0.673
```

```
lm(Price ~ Year+Age+Latitude+Stores+log_dist, data = hv02)%>% # checking if training_data04 woul
d fit
  augment()%>%
  mutate(residual  = Price - .fitted)%>%
  mutate(resi_sqd = residual^2)%>%
  summarize(rmse = sqrt(mean(resi_sqd)))
```

```
## # A tibble: 1 x 1
##    rmse
##   <dbl>
## 1  7.34
```

```
test_data = read.csv("test_data.csv")
test_data = test_data[-c(1),]
head(test_data)
```

```
##    No Year  Age  Distance Stores Latitude Longitude Price log_dist
## 2 413 2013  8.1  104.8101      5 24.96674  121.5407  52.5 4.652150
## 3 159 2013 11.6  390.5684      5 24.97937  121.5425  39.4 5.967603
## 4 197 2013 22.8  707.9067      2 24.98100  121.5471  36.6 6.562312
## 5 270 2013 17.6  837.7233      0 24.96334  121.5477  23.0 6.730688
## 6 410 2013 13.7 4082.0150      0 24.94155  121.5038  15.4 8.314346
## 7  67 2013  1.0  193.5845      6 24.96571  121.5409  50.7 5.265714
```

```
lm(Price ~ Year+Age+Latitude+Stores+log_dist, data = hv02)%>%
  augment()%>%
  mutate(residual  = Price - .fitted)%>%
  summarize(r_sqd = 1 - var(residual)/var(Price))
```

```
## # A tibble: 1 x 1
##   r_sqd
##   <dbl>
## 1 0.673
```

```
lm(Price ~ Year+Age+Latitude+Stores+log_dist, data = hv02)%>%
  augment()%>%
  mutate(residual  = Price - .fitted)%>%
  mutate(resi_sqd = residual^2)%>%
  summarize(rmse = sqrt(mean(resi_sqd)))
```

```
## # A tibble: 1 x 1
##    rmse
##   <dbl>
## 1  7.34
```

```
lm(Price ~ Year+Age+Latitude+Stores+log_dist, data = hv02)%>%
  augment(newdata = test_data)%>%
  mutate(residual = Price - .fitted)%>%
  mutate(resi_sqd = residual^2)%>%
  summarize(rmse = sqrt(mean(resi_sqd)))
```

```
## # A tibble: 1 x 1
##    rmse
##   <dbl>
## 1  6.06
```