# PREDICTING HOUSE PRICES USING LINEAR REGRESSION MODEL IN R

### 1.Introduction

The aim of this project is to assess the use of a linear regression model to predict housing prices in Singapore.

Our houses dataset has several variables. The 'Unit Price' variable will be the response variable whereas the rest of the variables in the dataset will be predictors.

As part of completing the project, I will carry out steps that include Data Cleaning, Exploratory Data Analysis, Data Modelling and Conclusion.

The project will answer the following *questions*: i)Which linear regression model is best suited to predict Price_Unit? ii)How is the model performing in terms of predicted values against actual values?

### 2.Data Pre-processing and Cleaning.

```
#rm(list=ls()) #clearing the environment
```

-loading libraries:

```
library(lattice)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```
library(datasets)
library(class)
library(car)
```

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

-importing the dataset which is a csv file from the directory:

```
Houses <- read.delim("Real estate.csv", sep = ',', header = TRUE)
```

-Checking the structure of the dataset and the datatypes

```
head(Houses)
```

```
##   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
```

...

```r
dim(Houses)
```

```
## [1] 414   8
```

...

```r
str(Houses)
```

```
## ’data.frame’:    414 obs. of  8 variables:
##  $ No                                    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ X1.transaction.date                   : num  2013 2013 2014 2014 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 ..
```

-Our data set have 418 observations and 8 variables. Of the 8 variables, 6 are numerical and 2 are integers.

-After Pre-processing, im now going to do some data cleaning. This involves dealing with missing data, removing unnecessary data and changing names.

-firstly, I will remove the first column as it is just an index and will not affect our predictions.

```r
Houses <- Houses[,-1]
head(Houses)
```

```
##   X1.transaction.date X2.house.age X3.distance.to.the.nearest.MRT.station
## 1            2012.917         32.0                               84.87882
## 2            2012.917         19.5                              306.59470
## 3            2013.583         13.3                              561.98450
## 4            2013.500         13.3                              561.98450
## 5            2012.833          5.0                              390.56840
## 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
```

-redefine the names of the columns to be more presentable.

```
colnames(Houses) <- c("Trans_Date", "Age_of_House", "Dist_to_Stores", "No_Stores", "Lat", "Long", "Pric
head(Houses)
```

```
##   Trans_Date Age_of_House Dist_to_Stores No_Stores      Lat     Long Price_Unit
## 1   2012.917         32.0       84.87882        10 24.98298 121.5402       37.9
## 2   2012.917         19.5      306.59470         9 24.98034 121.5395       42.2
## 3   2013.583         13.3      561.98450         5 24.98746 121.5439       47.3
## 4   2013.500         13.3      561.98450         5 24.98746 121.5439       54.8
## 5   2012.833          5.0      390.56840         5 24.97937 121.5425       43.1
## 6   2012.667          7.1     2175.03000         3 24.96305 121.5125       32.1
```

-check for missing data:

```
which(is.na(Houses) == TRUE)
```

```
## integer(0)
```

-there is no missing values in our Houses dataset, which is a good thing.

- checking the class of each variable in our dataset.

```
apply(Houses, 2, 'class')
```

```
##     Trans_Date   Age_of_House Dist_to_Stores      No_Stores            Lat
##      "numeric"      "numeric"      "numeric"      "numeric"      "numeric"
##           Long     Price_Unit
##      "numeric"      "numeric"
```

- All the variables are of numeric class, they will be no need of changing classes.

### 3.Exploratory Data Analysis

- The next step following data pre-processing & cleaning is exploratory data analysis.
- the reason for the exploratory data analysis is to check the distribution of data for each variable and to assess the relation between the variables in the dataset, more importantly the relation between the predictor variables and the response variable, 'Price Unit'.
- I will start by checking the summary statistics:

```
summary(Houses)
```

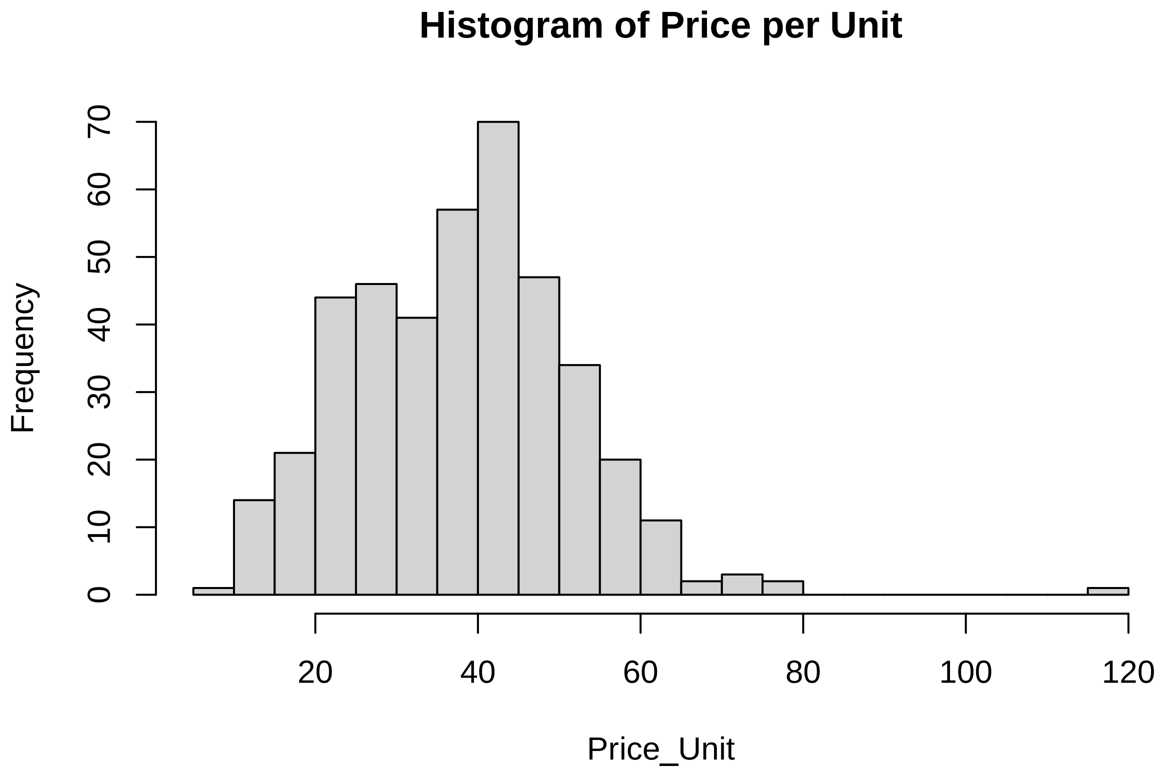
```
##    Trans_Date    Age_of_House    Dist_to_Stores      No_Stores
##  Min.   :2013   Min.   : 0.000   Min.   :  23.38   Min.   : 0.000
##  1st Qu.:2013   1st Qu.: 9.025   1st Qu.: 289.32   1st Qu.: 1.000
##  Median :2013   Median :16.100   Median : 492.23   Median : 4.000
##  Mean   :2013   Mean   :17.713   Mean   :1083.89   Mean   : 4.094
##  3rd Qu.:2013   3rd Qu.:28.150   3rd Qu.:1454.28   3rd Qu.: 6.000
##  Max.   :2014   Max.   :43.800   Max.   :6488.02   Max.   :10.000
##       Lat             Long          Price_Unit
##  Min.   :24.93   Min.   :121.5   Min.   :  7.60
##  1st Qu.:24.96   1st Qu.:121.5   1st Qu.: 27.70
##  Median :24.97   Median :121.5   Median : 38.45
##  Mean   :24.97   Mean   :121.5   Mean   : 37.98
##  3rd Qu.:24.98   3rd Qu.:121.5   3rd Qu.: 46.60
##  Max.   :25.01   Max.   :121.6   Max.   :117.50
```

- From the summary statistics, there is a big difference from the Max value(117.50) to 3rd Quartile(46.60) of the Price_Unit. The same thing can be observed for the Dist_to_Stores variable. This suggest presence of outlier(s).

-To be certain about the presence of outliers, I will do some data visualization.

-I will start with a histogram of 'Price_Unit', to view the distribution of data.

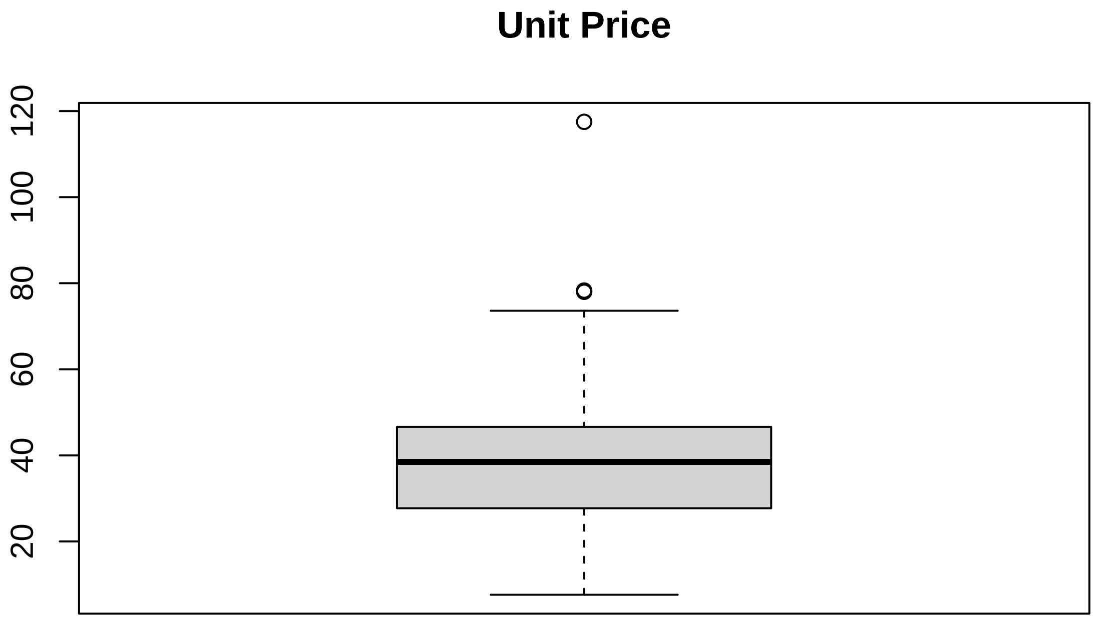
```
hist(Houses$Price_Unit, breaks = 20, main = "Histogram of Price per Unit", xlab = "Price_Unit")
```



-From the histogram, we do have an outlier(s) with a value of around 120.

-I will plot box and whiskers for the Price_Unit and Dist_to_Stores as they are the only 2 variables seemingly with outliers according to the summary statistics. A box plot is a better visualizer of data distribution as it incorporates quartiles.

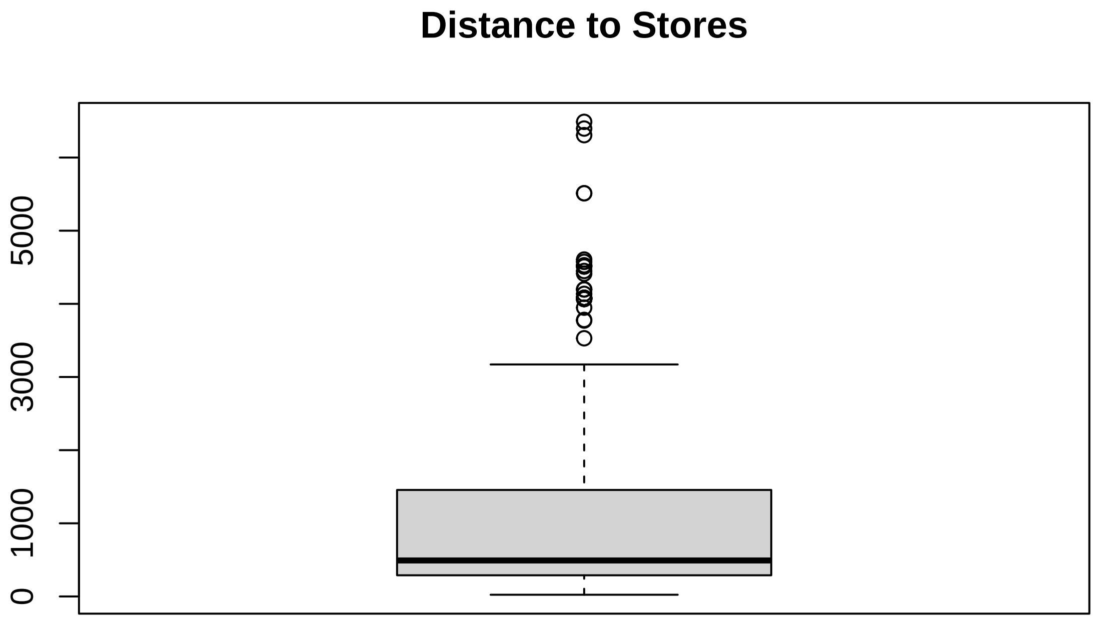
```
boxplot(Houses$Price_Unit, main = "Unit Price")
```



-The boxplot confirms our concern, we do have an outlier with a value of just below 120.

-now boxplot for Dist_to_Stores:

```
boxplot(Houses$Dist_to_Stores, main = "Distance to Stores")
```



-for certain, they are a few outliers for the dist_to_stores data.

-I will eliminate the few outliers.

```
Out_index <- order(Houses$Dist_to_Stores, decreasing = TRUE)[1:5]          #index of outliers in Dist_
Out_index <- c(Out_index, order(Houses$Price_Unit, decreasing = TRUE)[1:2])   #index of outliers in Pri
Out_index <- c(Out_index, order(Houses$Price_Unit, decreasing = FALSE)[1:5])
Houses1 <- Houses[-Out_index, ]                                             #eliminate outliers
```

-recheck the summary statistics

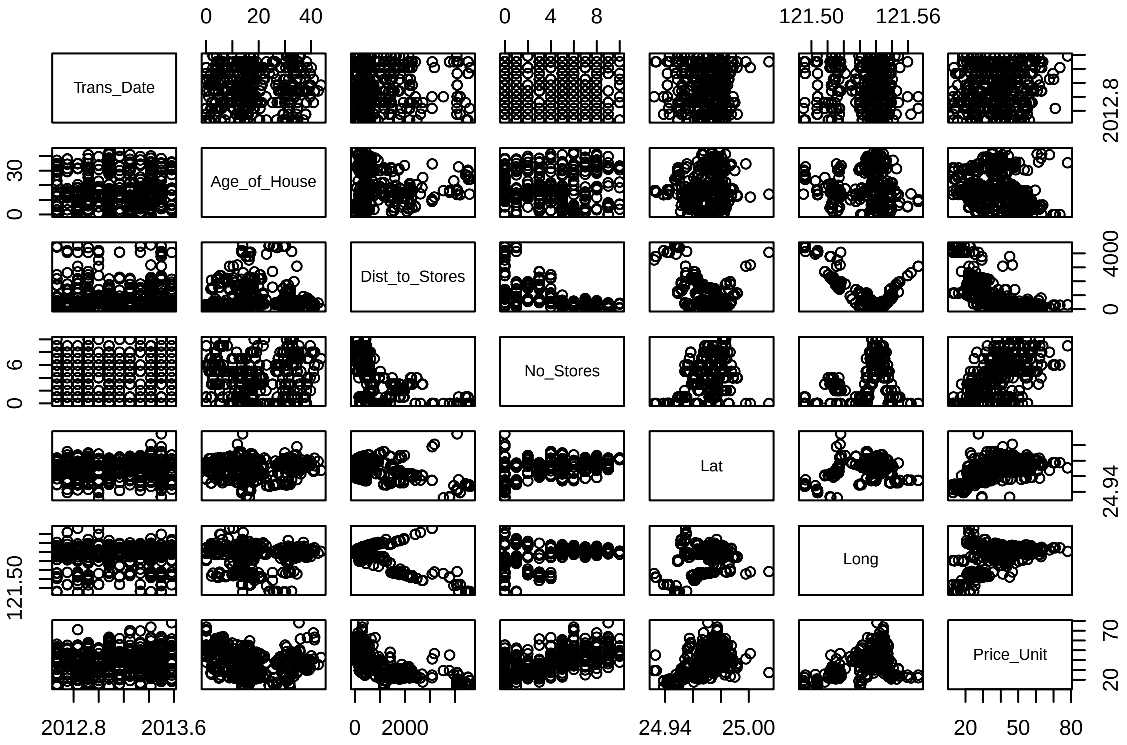
```
summary(Houses1)
```

```
##    Trans_Date    Age_of_House   Dist_to_Stores      No_Stores
##  Min.   :2013   Min.   : 0.00   Min.   :  23.38   Min.   : 0.000
##  1st Qu.:2013   1st Qu.: 8.80   1st Qu.: 289.32   1st Qu.: 1.000
##  Median :2013   Median :16.00   Median : 492.23   Median : 4.000
##  Mean   :2013   Mean   :17.59   Mean   :1013.70   Mean   : 4.144
##  3rd Qu.:2013   3rd Qu.:27.70   3rd Qu.:1414.84   3rd Qu.: 6.000
##  Max.   :2014   Max.   :43.80   Max.   :4605.75   Max.   :10.000
##       Lat             Long          Price_Unit
##  Min.   :24.93   Min.   :121.5   Min.   :12.80
##  1st Qu.:24.96   1st Qu.:121.5   1st Qu.:28.48
##  Median :24.97   Median :121.5   Median :38.85
##  Mean   :24.97   Mean   :121.5   Mean   :38.17
##  3rd Qu.:24.98   3rd Qu.:121.5   3rd Qu.:46.60
##  Max.   :25.01   Max.   :121.6   Max.   :78.00
```

- much better, though there is still a big difference between the minimum value and the 1st quartile for Price_Unit.
- Now I will do further exploratory data analysis to observe the relationship between the variables.

-I will start with a pairs plot to get an overview of all variable relations:

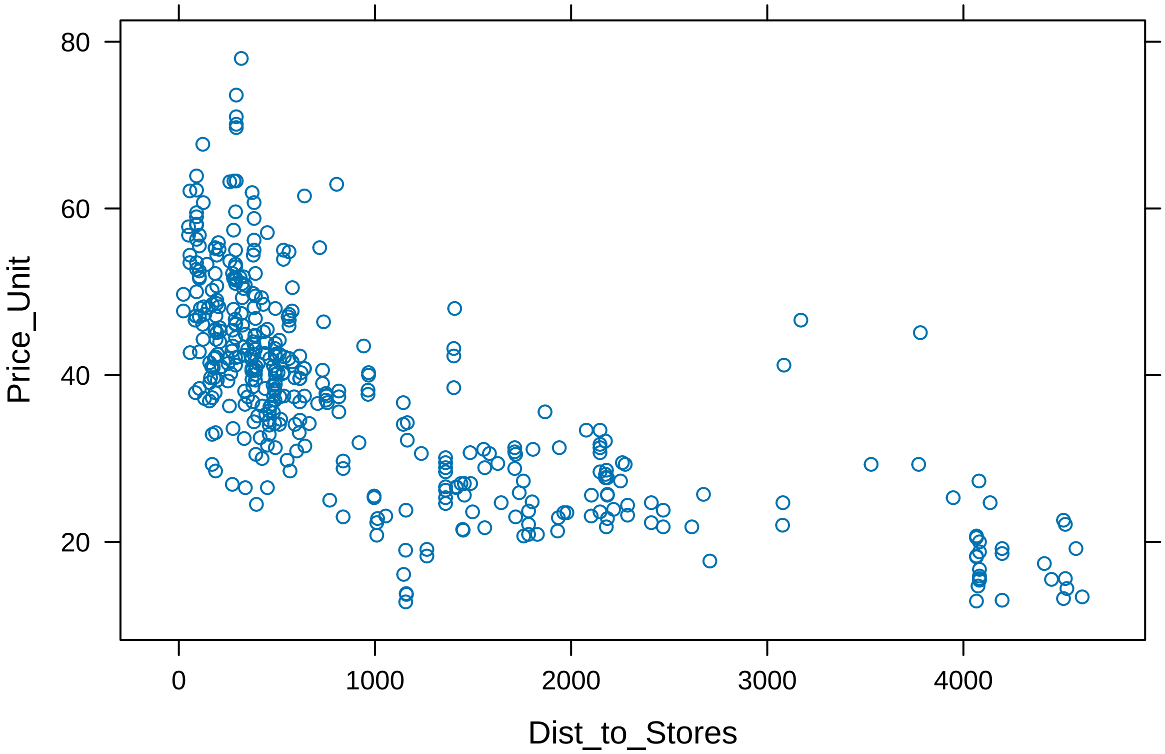
```
pairs(Houses1)
```



-There is some positive as well as negative co-relation between our variables.

-As one of the objective is to identify the influence of each variable on predicting the 'Price_Unit', I will now visualize each predictor against the response.

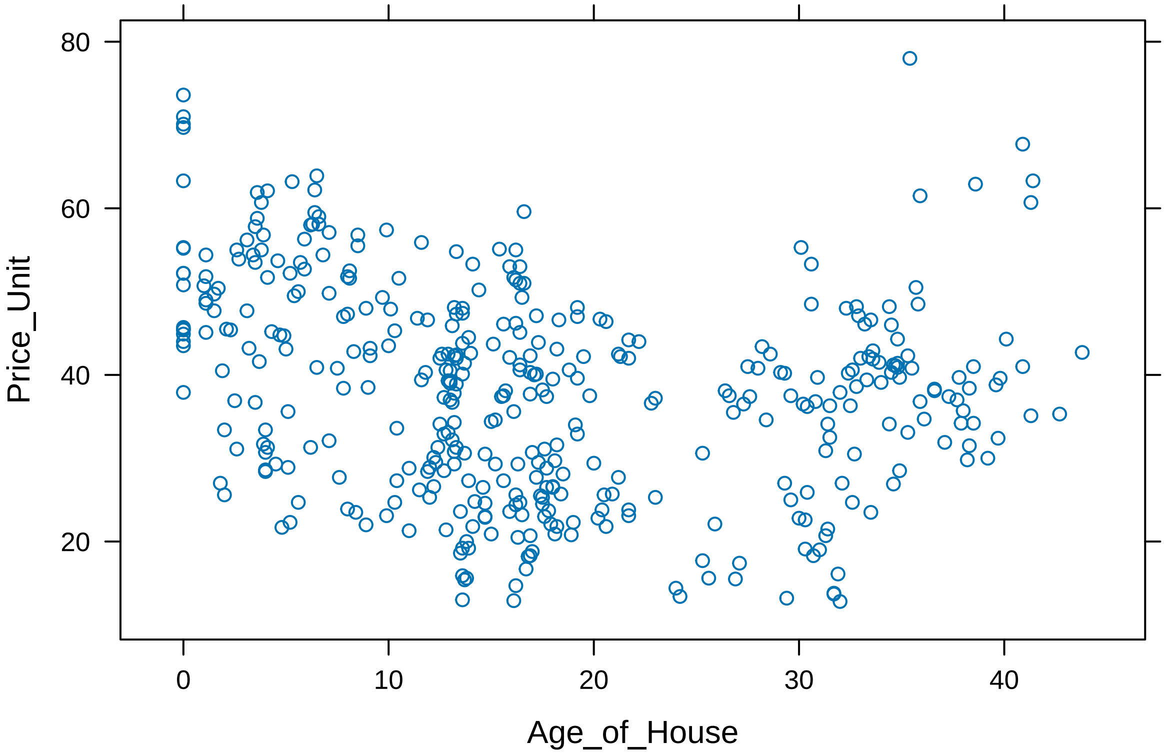
```
#Plot of Price Unit vs Distance to Stores
xyplot(Price_Unit ~ Dist_to_Stores, data = Houses1)
```



- the graph shows a negative correlation between Price Unit and Distance to Stores. The smaller the distance to stores, the higher the price.

- housing units that are close to stores have a higher value whilst those far from the stores have a lower value. This is not a surprise at all as being close to stores is a big advantage.

```r
#Plot of Price Unit vs Age of House
xyplot(Price_Unit ~ Age_of_House, data = Houses1)
```



-The plot is giving a mixed relation vibe! I will break the interpretation of the graph into 3 parts:

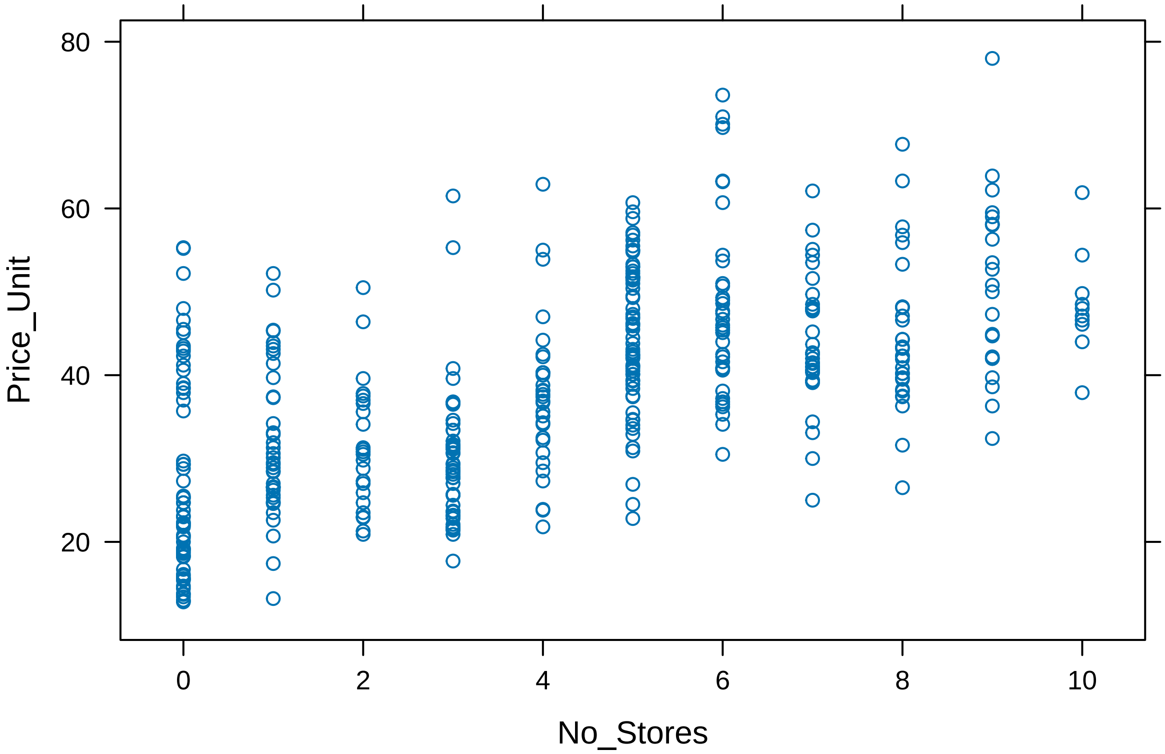
i)0 to 10 years - houses with less than 10 years do not have low prices, they actually have some of the highest priced housing units. In this category it is units with 1 year or less that have large prices.

ii)11 to 33 years - most of the houses are in this category. They have low priced house to medium priced houses. They do not have highly priced units and all the lowest priced units are in this category.

iii) more than 34 years - surprisingly, units in this category do not have any low priced houses. The housing units are priced medium to high only.

-definitely age has an influence on Price of the houses.

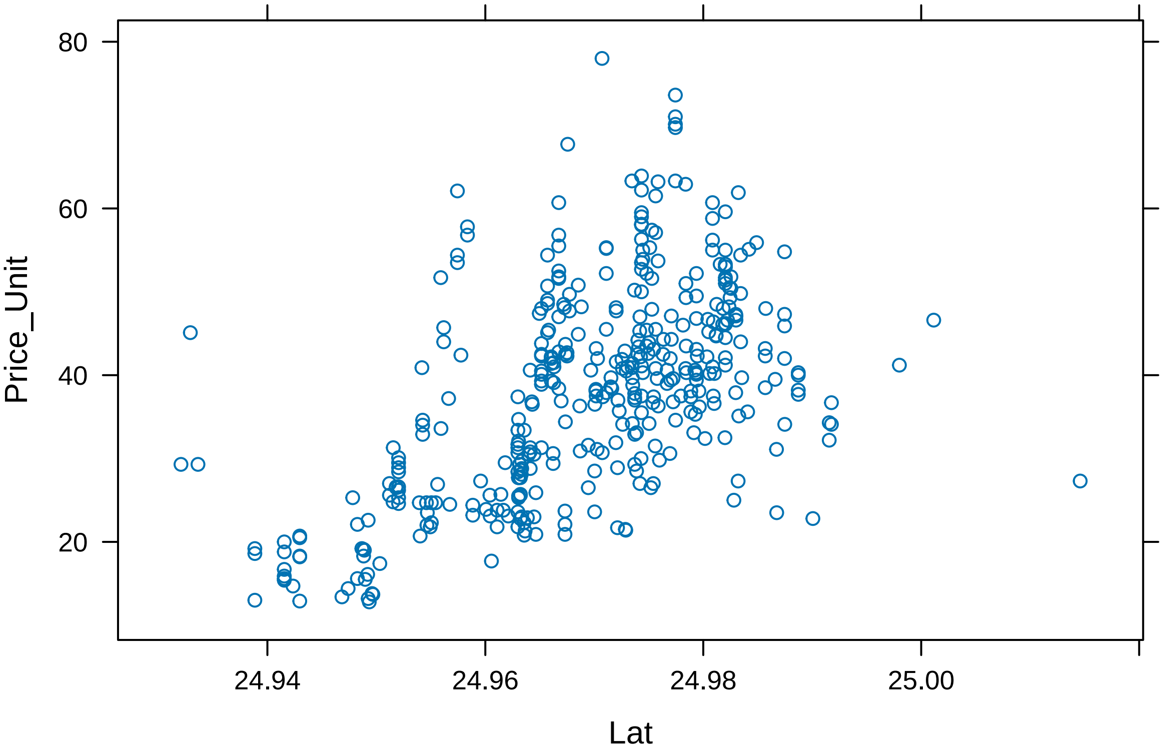
```
#Plot of Price Unit vs Number of Stores
xyplot(Price_Unit ~ No_Stores, data = Houses1)
```



- from the visualization, there is a positive correlation between Price_Unit and No_Stores.

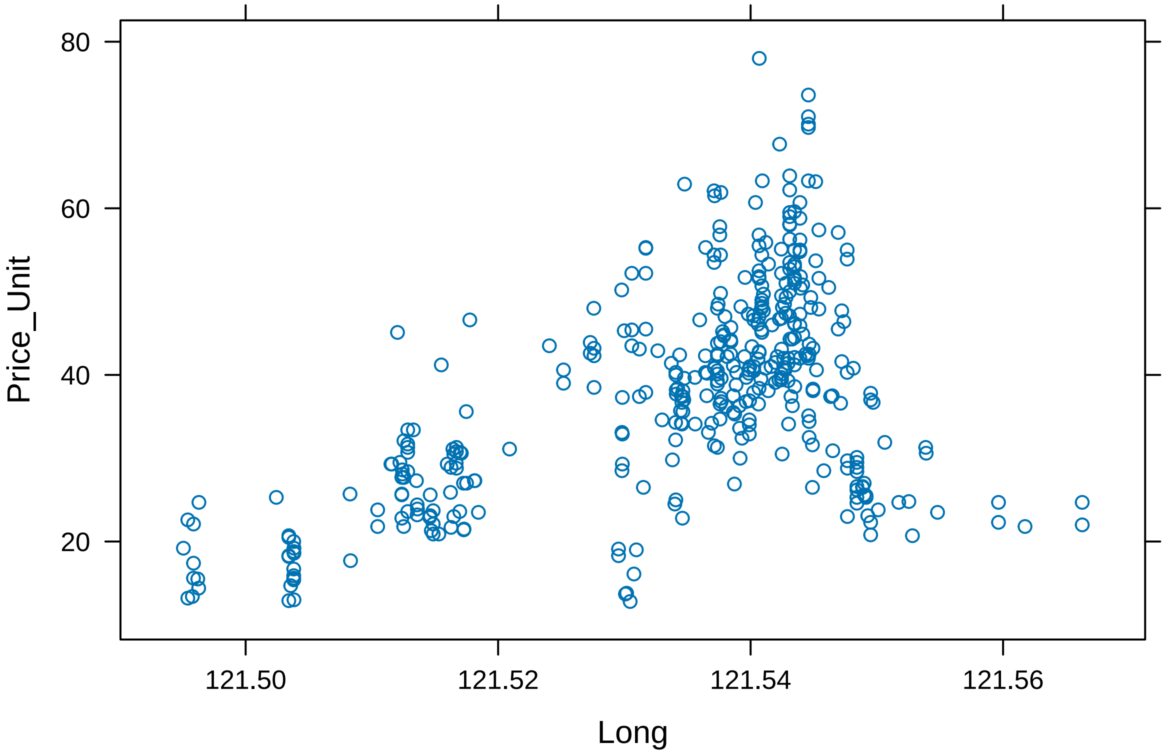
- the higher the number of stores in the area, the higher the house prices.

```
#Plot of Price Unit vs Latitude
xyplot(Price_Unit ~ Lat, data = Houses1)
```



-the higher the latitude, the higher the house price, according to graph of Lat and Price_Unit which is showing a positive correlation.

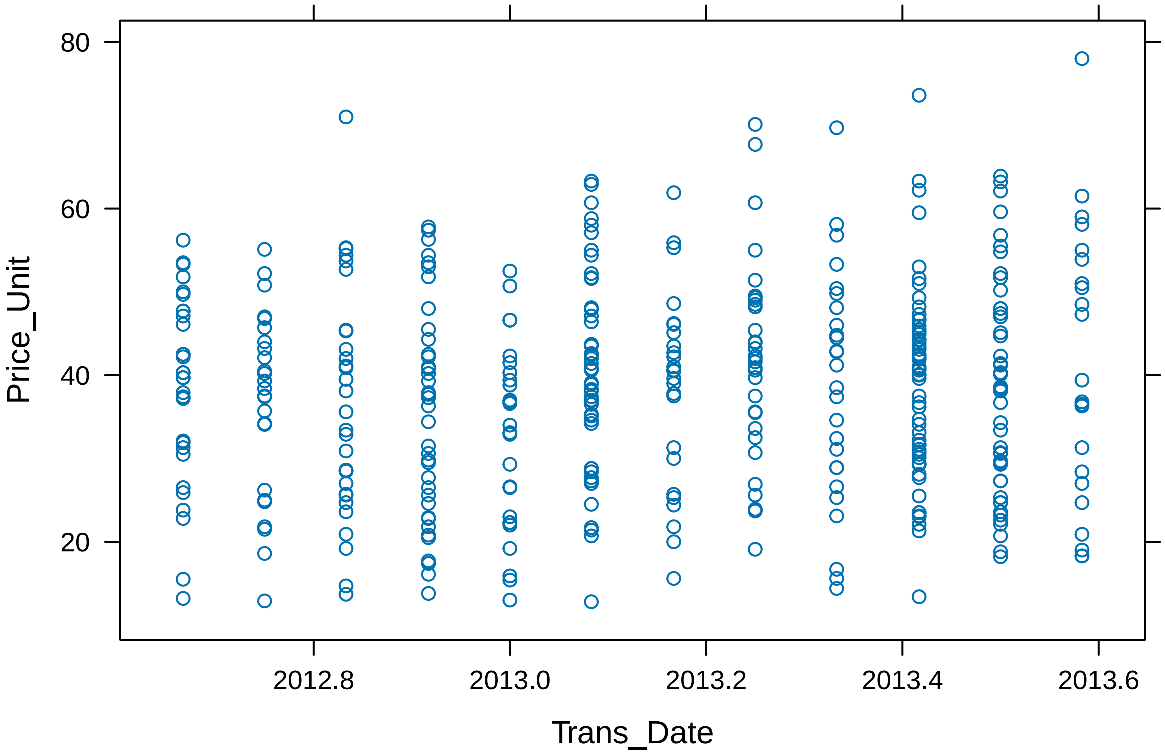
```
#Plot of Price Unit vs Longitude
xyplot(Price_Unit ~ Long, data = Houses1)
```



-the positive correlation between Price unit and longitude seem to be weak. as the longitude increases, the price spread throughout.

-generally from the graph, houses with small longitude are priced very low.

```
#Plot of Price Unit vs Transactional Date
xyplot(Price_Unit ~ Trans_Date, data = Houses1)
```



- price unit and transactional date, have a constant relationship.

- this can be due to the fact that all the houses in this data were sold within a 12 months period hence constant pricing.
- in accordance with the visualizations there is a possibility that all our variables do influence response variable, 'Price_Unit'.

**4.Data Modelling** -This stage of data modelling will involve fitting the linear regression model, dividing our data into test and train sets, predicting and error profiling.

**training and test sets:**

```
N <- length(Houses1[,1]) #let N be the number of observations in dataset mycsv.
set.seed(575)
train_index <- sample(1:N, size = (4/5)*N, replace = FALSE) #getting the random sampled index for the t
```

-dividing the data into test and train set using the random generated index, train_index:

```
train <- Houses1[train_index,]
test <- Houses1[-train_index,]
```

**fitting the model:** -now its time to fit the linear regression model. - I will fit Price_Unit against all the other variables and assess the influence they have on Price_Unit.

```r
themodel <-lm(Price_Unit ~ Trans_Date + Age_of_House + Dist_to_Stores + No_Stores + Lat + Long, data =
```

-will use the summary function to get details of the fitted model.

```r
summary(themodel)
```

```
##
## Call:
## lm(formula = Price_Unit ~ Trans_Date + Age_of_House + Dist_to_Stores +
##     No_Stores + Lat + Long, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.466  -4.982  -0.765   4.295  33.839
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.522e+04  6.610e+03  -2.302 0.021986 *
## Trans_Date      5.511e+00  1.567e+00   3.516 0.000502 ***
## Age_of_House   -3.134e-01  3.864e-02  -8.111 1.12e-14 ***
## Dist_to_Stores -4.710e-03  7.528e-04  -6.256 1.28e-09 ***
## No_Stores       1.150e+00  1.849e-01   6.219 1.59e-09 ***
## Lat             2.071e+02  4.553e+01   4.548 7.72e-06 ***
## Long           -8.264e+00  4.713e+01  -0.175 0.860919
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.759 on 316 degrees of freedom
## Multiple R-squared:  0.6366, Adjusted R-squared:  0.6297
## F-statistic: 92.28 on 6 and 316 DF,  p-value: < 2.2e-16
```

-all the variables are significant except for ‘Long’. This means that Longitudinal has no significance on influencing the Price Unit.

-I will refit the model but this time without ‘Long’ so as to increase the overall influence of the predictors on the response variable.

```r
themodel1 <-lm(Price_Unit ~ Trans_Date + Age_of_House + Dist_to_Stores + No_Stores + Lat, data = train)
```

-checking the summary statistics of the new model:

```r
summary(themodel1)
```

```
##
## Call:
## lm(formula = Price_Unit ~ Trans_Date + Age_of_House + Dist_to_Stores +
##     No_Stores + Lat, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.514  -5.086  -0.782   4.296  33.841
##
```

```
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.622e+04  3.286e+03  -4.937 1.28e-06 ***
## Trans_Date      5.496e+00  1.563e+00   3.517    5e-04 ***
## Age_of_House   -3.133e-01  3.857e-02  -8.122 1.03e-14 ***
## Dist_to_Stores -4.621e-03  5.537e-04  -8.345 2.23e-15 ***
## No_Stores       1.152e+00  1.843e-01   6.250 1.32e-09 ***
## Lat             2.083e+02  4.493e+01   4.636 5.19e-06 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 7.747 on 317 degrees of freedom
## Multiple R-squared:  0.6366, Adjusted R-squared:  0.6309
## F-statistic: 111.1 on 5 and 317 DF,  p-value: < 2.2e-16
```

-Now this is good, all the predictor variables are significant, even the intercept is now more significant.

-The F-statistic has a value of 111.1 with a very small p-value, this shows a strong overall influence of the predictor variables on the response variable, Price_Unit.

-Next, I will check if we have an multicolinearity among our variables. I will use the vif function.

```
vif(themodel1)
```

```
##     Trans_Date   Age_of_House Dist_to_Stores      No_Stores            Lat
##       1.011566       1.020934       2.094808       1.627464       1.634463
```

- there is no multicolinearity among our variables, which is a good thing.

*Model Prediction:* -now I'm going to predict thePrice_Unit, using the predict function with our fitted model on the test set data.

-firstly, I will seperate the response variable from the predictor variables in our traun and test sets:

```
train_X = train[,-7]
train_Y = train[7]
test_X = test[,-7]
test_Y = test[7]
```

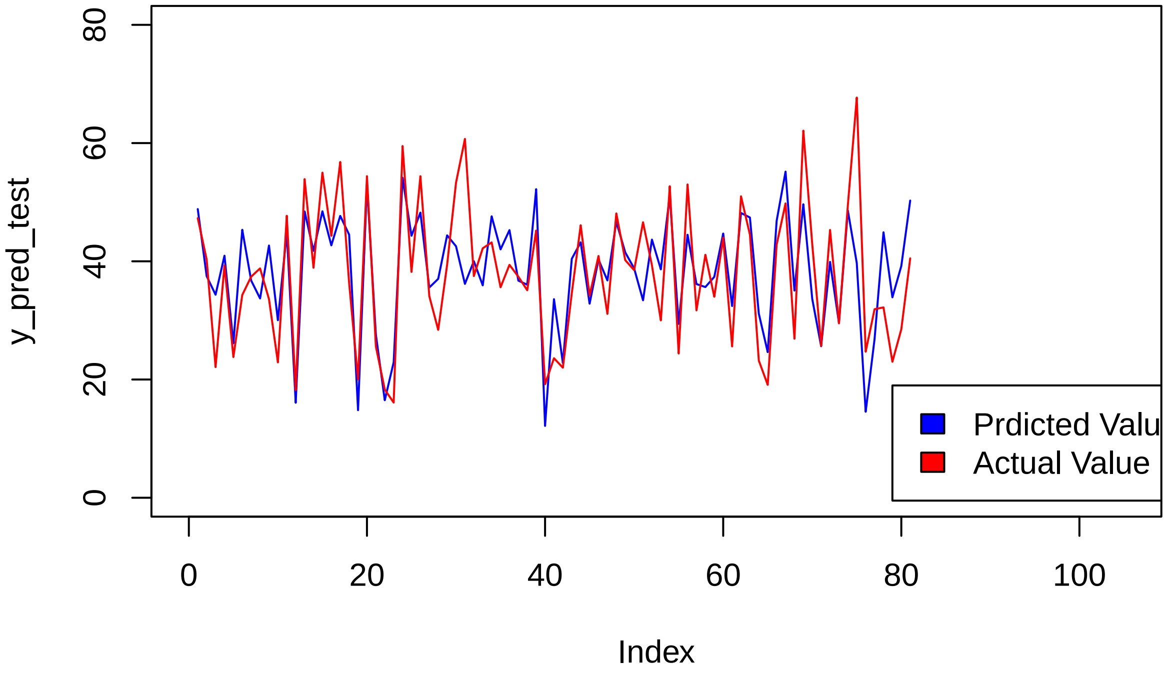
-using the predict function:

```
y_pred_test = predict(themodel1, newdata = test_X)
#y_pred_train = predict(themodel1, newdata = train_X)
```

**error profiling:** -I will profile the test error, to assess the performance of the model in predicting Price_Unit.

-showing response values against the predicted values for the test set.

```
plot(y_pred_test, col = "blue", type = "l", xlim = c(0,105), ylim = c(0,80))
lines(test_Y, col = "red", type = "l")
legend(79, 19, legend=c("Prdicted Value", "Actual Value"),
       fill = c("blue","red")
)
```



-from the graph, its clear our model could not correctly predict several response values. it underestimated most of the high predictor values.

-I will calculate the mean squared error (MSE) to get more insight in the performance of the model:

```
test_error = (1/length(y_pred_test)) * (sum(test_Y[1] - y_pred_test)^2)
```

....

```
test_error
```

```
## [1] 14.2071
```

```
#class(test_Y[1])
```

-The mean squared error 14.21 -The MSE is big due to the underestimating and overestimating of the response variable.

**5.Conclusion**

**Results and discussion:** I was able to fulfill the aim of the project, which is to assess the use of a linear regression model to predict housing prices.

The objectives were to answer 2 research questions. Firstly, after fitting the first model, all the variables were significant except for longitudinal. To enhance the overall impact of the predictors on the response variable, I refitted a new model without longitudinal. The results for the summary statistics of the second model were promising as all predictor variables were now significant, and even the intercept become more significant, demonstrating a strong overall influence of the predictor variables on the response variable, Price Unit.

Secondly, the graph of predicted test values against actual test values indicates that our model fails to accurately predict several response values, particularly underestimating many of the higher predictor values. To gain further insight into the model's performance, I calculated the mean squared error (MSE). The MSE is quite large, reflecting both the under-estimations and over-estimations of the response variable. Overall, the performance of this linear regression model to predict housing prices is not so good.

**Future Recommendation:** To improve the model performance, I recommend implementing cross-validation to ensure that the model generalizes well to unseen data. This can help assess performance more reliably. More so refining the model, I would consider trying different regression techniques, such as Ridge or Lasso to address potential issues like over fitting.