ILLINOIS INSTITUTE OF TECHNOLOGY

APPLIED STATISTICS PROJECT REPORT
MATH-569 (FALL 2022)

HOUSE PRICES DATA ANALYTICS

Team Members

Jasleen Kaur Bhatia (A20495939)

Sajesh Rao Erabelli (A20504279)

1. Introduction

There are many factors that affect the sale price of the Houses. The market forces that affect the housing prices include interest rates, economic factors - GDP, employment, manufacturing, prices of goods, import/export - and government subsidies. These forces are out of our control and not easily predictable. Sometimes the prices rise, other times the prices falls. In this project we understand the data and try to predict the sale price of houses based on different internal factors like - number of bedrooms, bathrooms, square foot, etc. - that affect the housing prices.

2. Overview of the Study

The dataset we have taken is House sales in King County, USA. The data contains the prices of houses against a variety of parameters. The objective of the study is to use statistical analysis to find the dependence of these variables on the price of houses, and which parameters affect the housing prices and which variables have minimal effect on the price of houses. We use various statistical tools for analysis of data and create a model based on such analysis.

Data:

The is the House sales in King County, USA. The dataset is taken from Kaggle. The specific URL is https://www.kaggle.com/harlfoxem/housesalesprediction. There are 19 house features and 21 overall columns. There are 21597 observations. The descriptions of the specific columns are:

id - a notation for the house; a numeric data type
date - date the house was sold; string
price - Price of the house; numeric
bedrooms - no.of bedrooms in a house; numeric
bathrooms - no. of bathrooms; numeric
sqft_living - square footage of the home; numeric
sqft_lot - square footage of the lot; numeric
floors - no.of floors in the house; numeric
waterfront - House which has a view to a waterfront; numeric
view - has been viewed; numeric
condition - how good is the condition; numeric
grade - overall grade given to the housing unit, based on King County grading system; numeric
sqft_above - square footage of house apart from basement; numeric
sqft_basement - square footage of the basement; numeric
yr_built - built year; numeric
yr_renovated - year when house was renovated; numeric
zipcode - zip; numeric
lat - latitiude; numeric
long - longitude; numeric
sqft_living15 - Living room area in 2015(implies– some renovations). This might or might not have affected the lot size area; numeric
sqft_lot15 - lot Size area in 2015(implies– some renovations); numeric

Solution:

Our goal is to understand the relation between the price (dependent variable) and all other features and create a model which will be able to predict the price when such details are given. For this purpose, we first understand and analyse the data post which we try to create a model. We plan to use ANOVA and MLR(Multiple Linear Regression) in creation of the model and try to find the best fit from the options considered.

Software used:

We will be using R language for this project as it is easy to apply statistical methods on the data using R.

Understanding the data :

```
summary(house_datasales)
##        id                date               price            bedroems
##  Min.   :1.000e+06   Length:21597       Min.   :  78000   Min.   : 1.000
##  1st Qu.:2.123e+09   Class :character   1st Qu.: 322000   1st Qu.: 3.000
##  Median :3.905e+09   Mode  :character   Median : 450000   Median : 3.000
##  Mean   :4.580e+09                      Mean   : 540297   Mean   : 3.373
##  3rd Qu.:7.309e+09                      3rd Qu.: 645000   3rd Qu.: 4.000
##  Max.   :9.900e+09                      Max.   :7700000   Max.   :33.000
##    bathrooms      sqft_living       sqft_lot           floors
##  Min.   :0.500   Min.   :  370   Min.   :    520   Min.   :1.000
##  1st Qu.:1.750   1st Qu.: 1430   1st Qu.:   5040   1st Qu.:1.000
##  Median :2.250   Median : 1910   Median :   7618   Median :1.500
##  Mean   :2.116   Mean   : 2080   Mean   :  15099   Mean   :1.494
##  3rd Qu.:2.500   3rd Qu.: 2550   3rd Qu.:  10685   3rd Qu.:2.000
##  Max.   :8.000   Max.   :13540   Max.   :1651359   Max.   :3.500
##    waterfront            view          condition         grade
##  Min.   :0.000000   Min.   :0.0000   Min.   :1.00   Min.   : 3.000
##  1st Qu.:0.000000   1st Qu.:0.0000   1st Qu.:3.00   1st Qu.: 7.000
##  Median :0.000000   Median :0.0000   Median :3.00   Median : 7.000
##  Mean   :0.007547   Mean   :0.2343   Mean   :3.41   Mean   : 7.658
##  3rd Qu.:0.000000   3rd Qu.:0.0000   3rd Qu.:4.00   3rd Qu.: 8.000
```

From the above , we understand that :

- The price of the houses varies from 78000(which is the minimum value) to 7700000(max. value).
- The minimum no. of bathrooms is 0.5 and the more than 75% of our data has bathrooms upto 2.5
- The houses we consider have a minimum living area of 370 units.

We use uszip data set as it has various information like county names, population, zipcode, city name and some other interesting information. It has 18 columns and 33788 rows.

Understanding the USZIPS data :

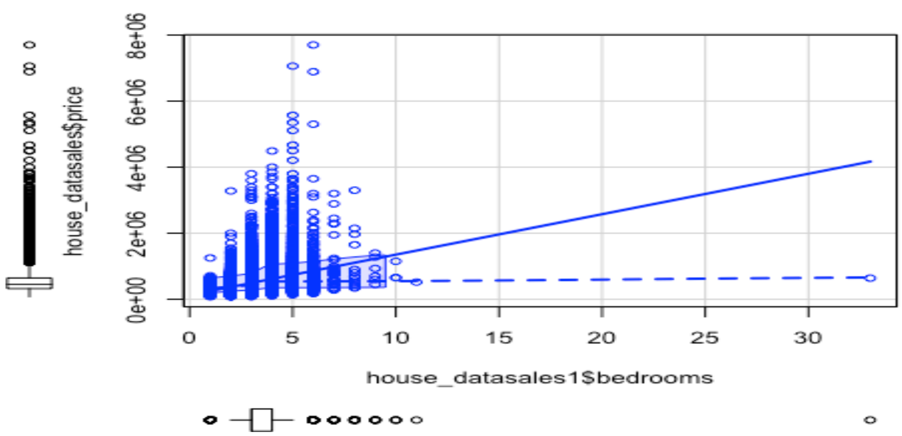
```
for (column in zipcode_data){
  print( typeof(column))
}
## [1] "character"
## [1] "double"
## [1] "double"
## [1] "character"
## [1] "character"
## [1] "character"
## [1] "logical"
## [1] "logical"
## [1] "double"
## [1] "double"
## [1] "character"
```

```
## [1] "character"
## [1] "character"
## [1] "character"
## [1] "character"
## [1] "logical"
## [1] "logical"
## [1] "character"
```

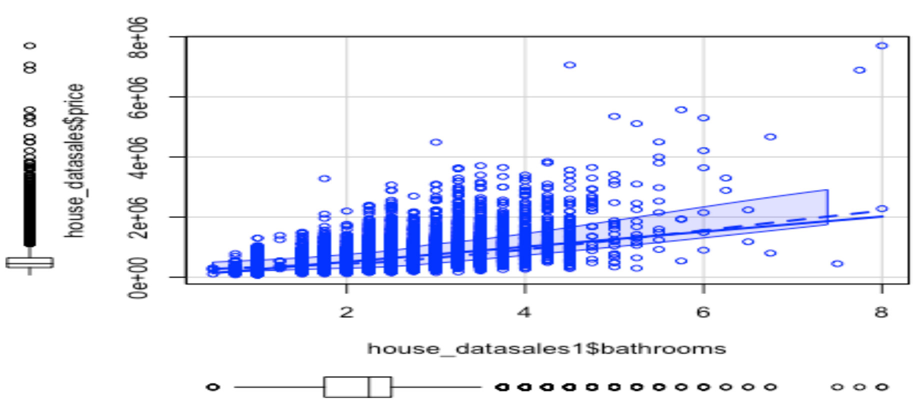
From the above, we understand that 10 features have their data type as character, 4 features have it as double and the remaining 4 are logical. As our main focus is on prediction of sold price, we remove values that do not have much impact on the change in the value of price.

We need to understand how each feature changes with change in the price and to understand that better, we plot them individually and see how the price change with increase in the value of the feature.

```
par(mfrow=c(4,5))
scatterplot(house_datasales1$bedrooms,house_datasales$price)
```

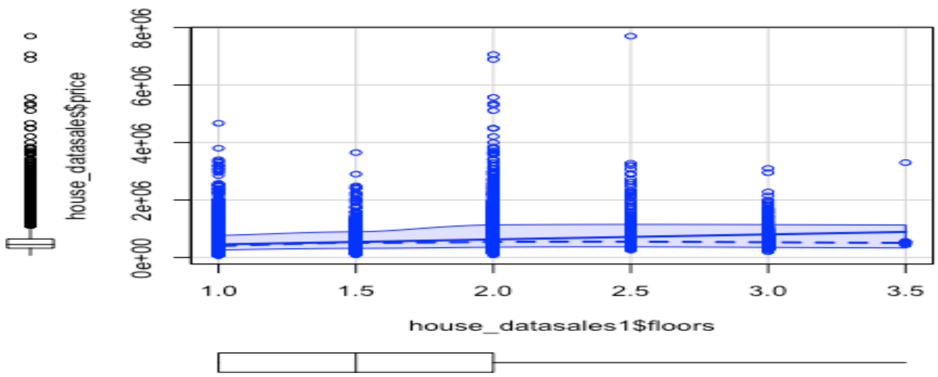



```
scatterplot(house_datasales1$bathrooms,house_datasales$price)
```

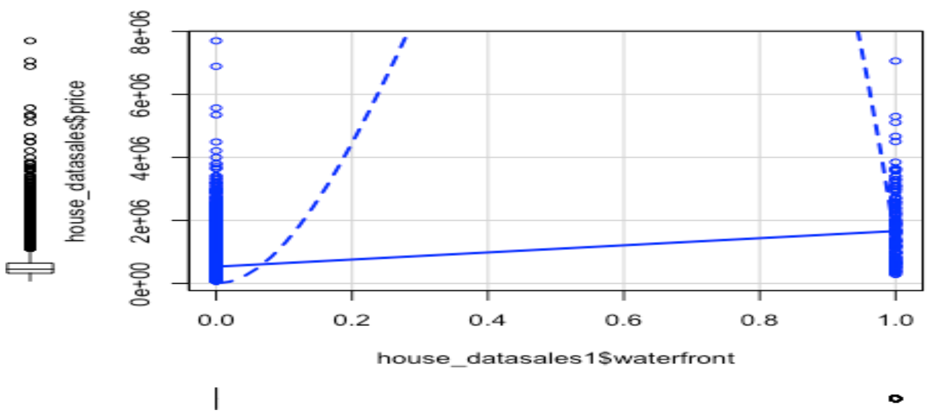


We see that as the number of bathrooms of a house increase, the price increases in most of the cases.

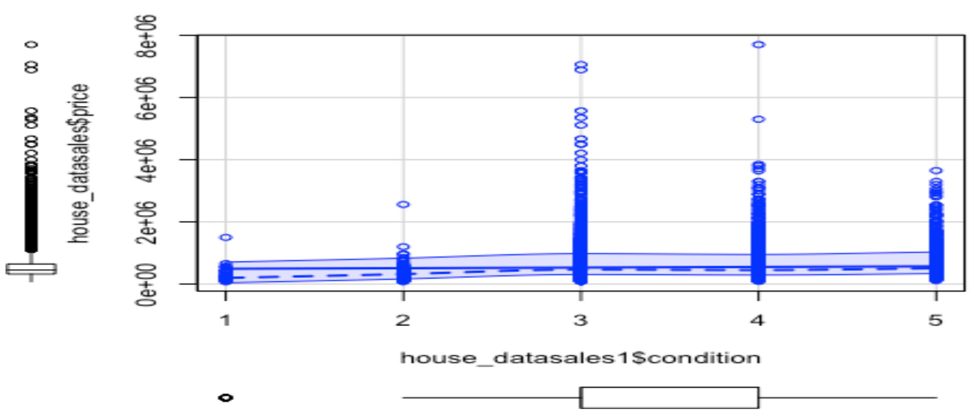
```
scatterplot(house_datasales1$floors,house_datasales$price)
```



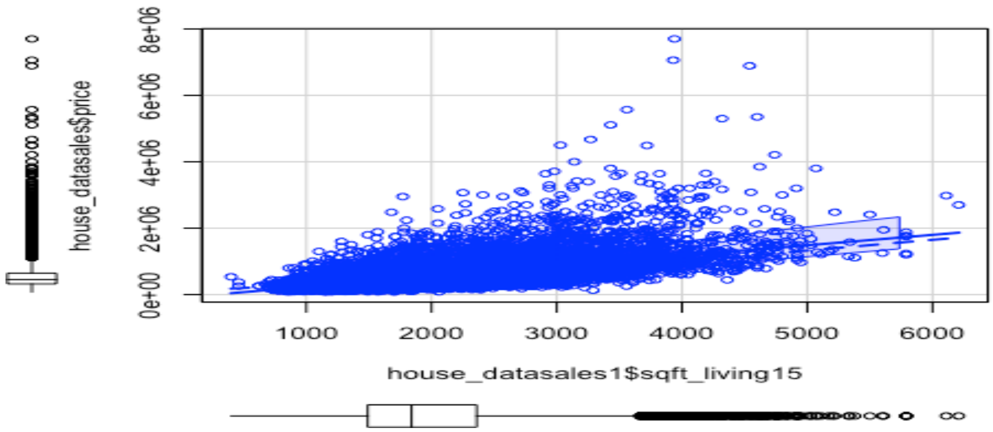

```
scatterplot(house_datasales1$waterfront,house_datasales$price)
```



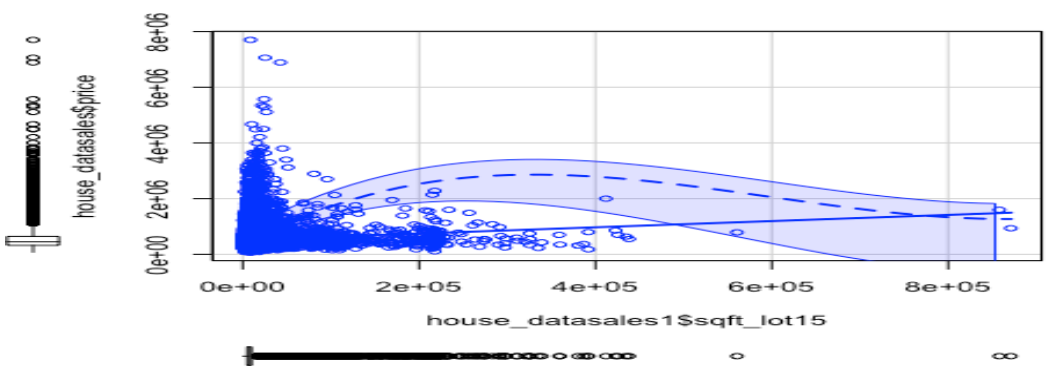

```
scatterplot(house_datasales1$condition,house_datasales$price)
```



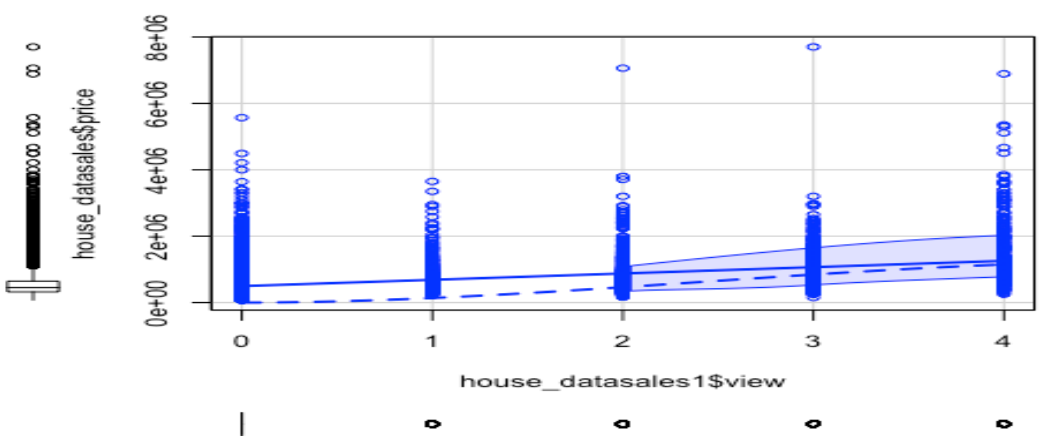
```
scatterplot(house_datasales1$sqft_living15,house_datasales$price)
```



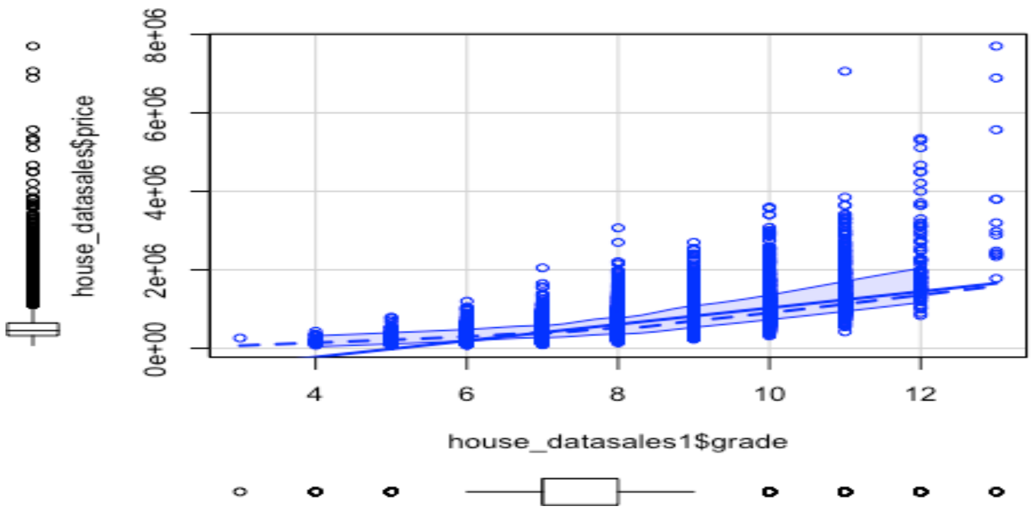

```
scatterplot(house_datasales1$sqft_lot15,house_datasales$price)
```



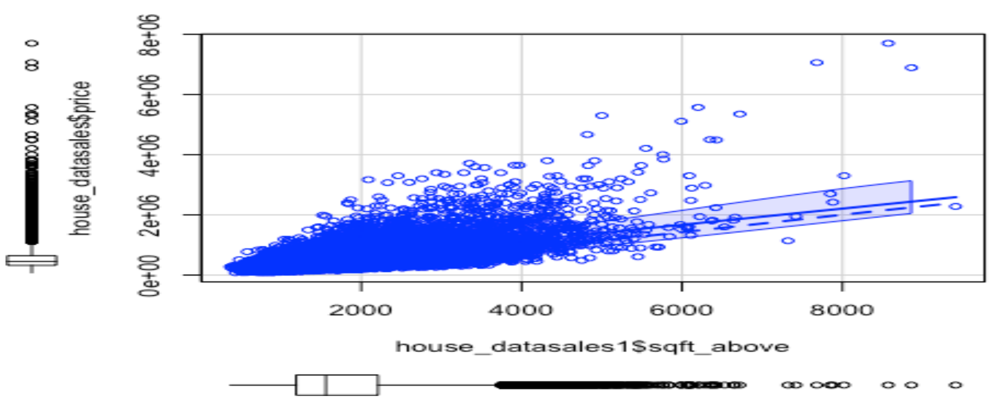

```
scatterplot(house_datasales1$view ,house_datasales$price)
```



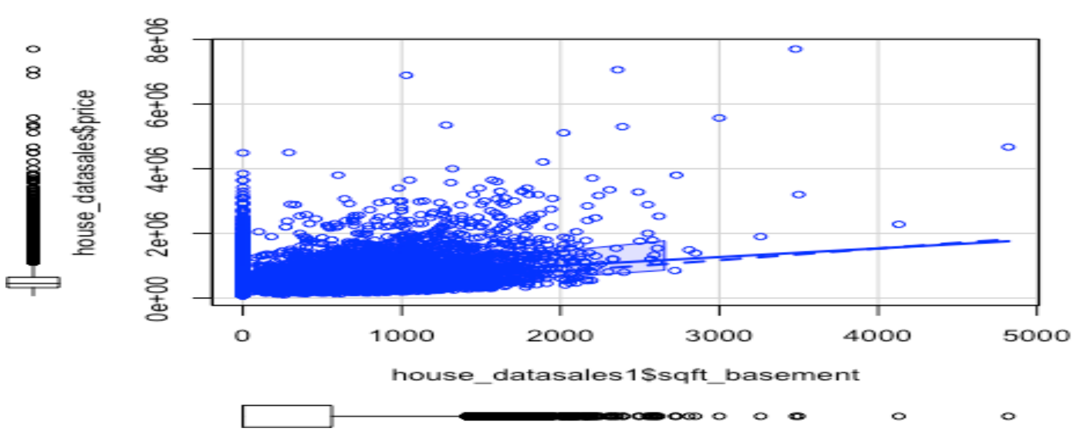
```
scatterplot(house_datasales1$grade,house_datasales$price)
```



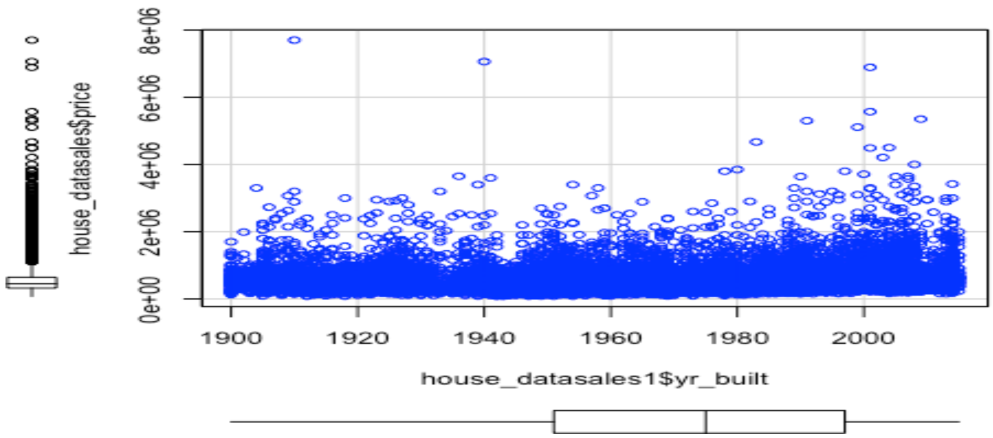

```
scatterplot(house_datasales1$sqft_above,house_datasales$price)
```



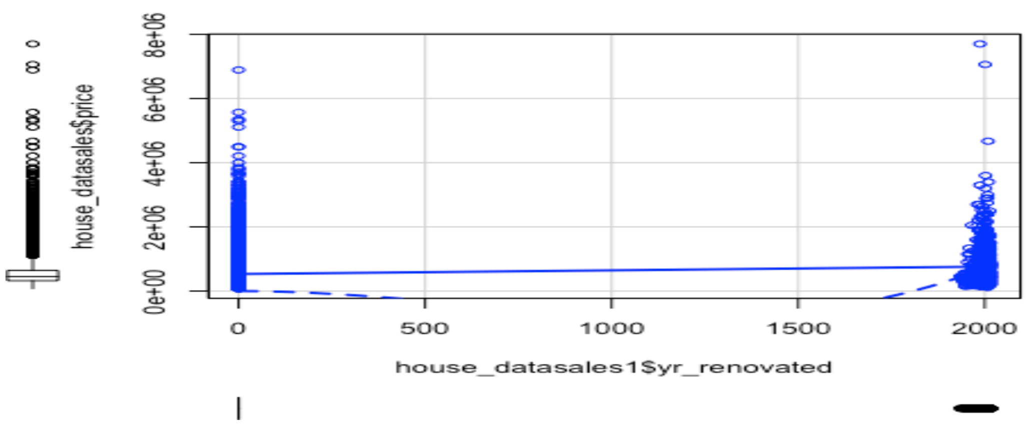

```
scatterplot(house_datasales1$sqft_basement,house_datasales$price)
```



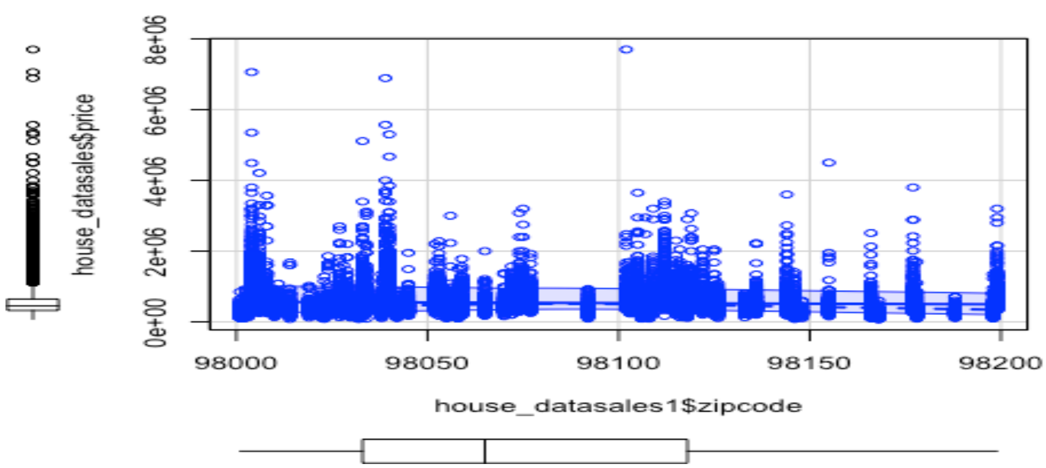
```
scatterplot(house_datasales1$yr_built,house_datasales$price)
```



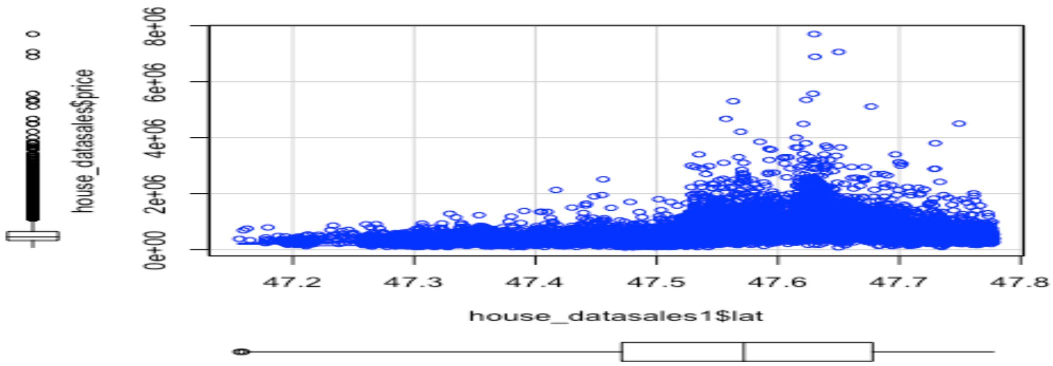

```
scatterplot(house_datasales1$yr_renovated,house_datasales$price)
```



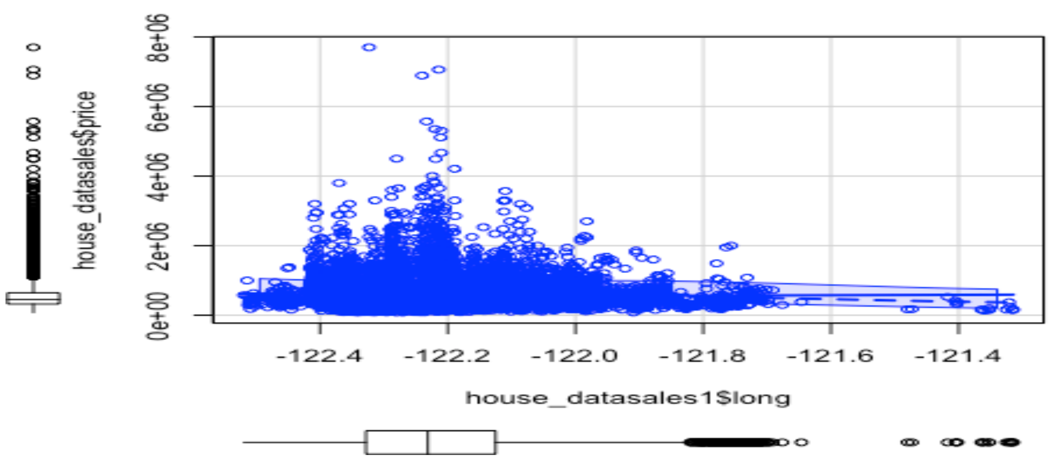

```
scatterplot(house_datasales1$zipcode,house_datasales$price)
```



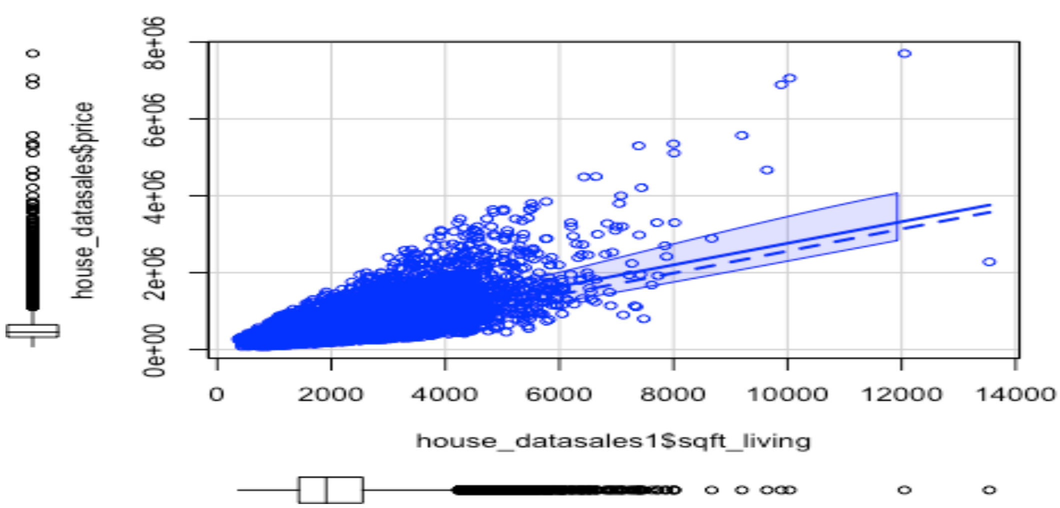
```
scatterplot(house_datasales1$lat,house_datasales$price)
```



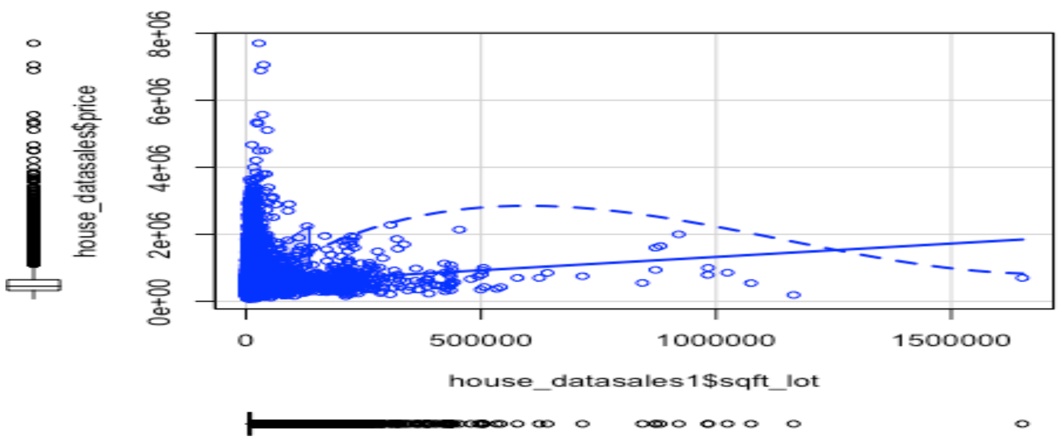

```
scatterplot(house_datasales1$long,house_datasales$price)
```




```
scatterplot(house_datasales1$sqft_living,house_datasales$price)
```

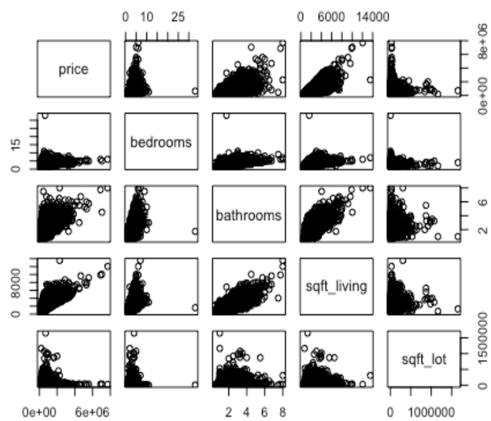


```
scatterplot(house_datasales1$sqft_lot,house_datasales$price)
```

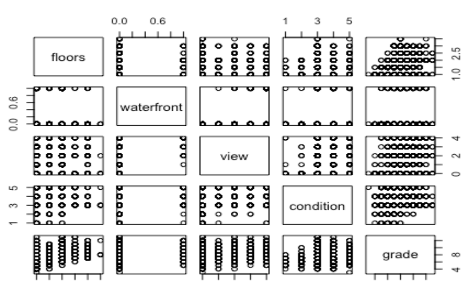


Initial look on the above relation between each variable to the dependent variable - price makes us understand that there are some outliers in the data which we have to take care such that the influence of such points in the creation of the model is less.

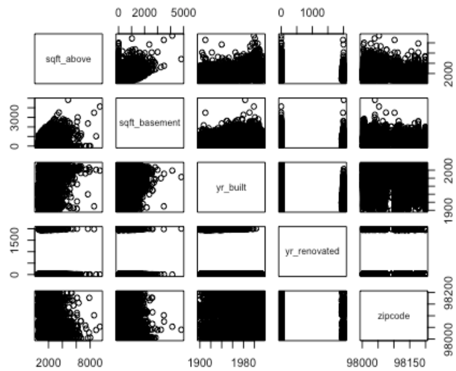
```
plot(house_datasales1[1:5])
```



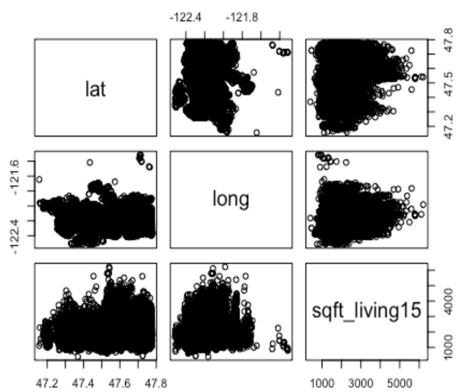

```
plot(house_datasales1[6:10])
```



```
plot(house_datasales1[11:15])
```




```
plot(house_datasales1[16:18])
```



We have checked the correlation of features with the price and we got the values as mentioned below:

```
cor(house_datasales1[1:5],house_datasales1$price)
##                     [,1]
## price       1.00000000
## bedrooms    0.30878747
## bathrooms   0.52590562
## sqft_living 0.70191730
## sqft_lot    0.08987622
cor(house_datasales1[6:10],house_datasales1$price)
##                   [,1]
## floors     0.25680354
## waterfront 0.26639846
## view       0.39737030
## condition  0.03605638
## grade      0.66795077
cor(house_datasales1[11:19],house_datasales1$price)
```

```
##                        [,1]
## sqft_above     0.60536794
## sqft_basement  0.32379891
## yr_built       0.05395333
## yr_renovated   0.12642362
## zipcode       -0.05340243
## lat            0.30669231
## long           0.02203632
## sqft_living15  0.58524120
## sqft_lot15     0.08284493
```

The above values indicate that :

a. There is a strong correlation between the price and sqft_living space. Also there is good correlation between bathrooms, bedrooms, grade, sqft_above to price.

b. There is minimal or minute correlation between price to condition, yr_built, sqft_lot15, and condition

So while creating the model, we need not use the features which has minimal or minute correlation and hence remove them from the data.

We the check the data and see if there are any missing and/or duplicate values. We ensure there is no such value in our data.

Checking missing values and duplicate values in the data:

```
## missing values check and finally merge the data:
print(sum(is.na(house_datasales1)))
## [1] 0
print(sum(is.na(zipcode_data)))

final_merged_data <- merge(house_datasales1,zipcode_data,by="zip")
```

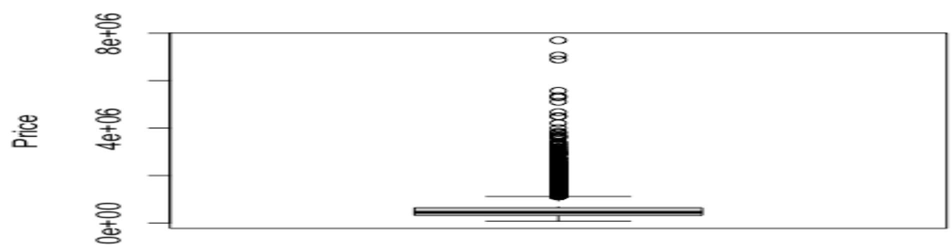
We go ahead with merging 2 datasets as it will then be easy for us to create the model.

```
# Merging 2 datasets
final_merged_data <- merge(house_datasales1,zipcode_data,by="zip")
View(final_merged_data)
# Examine the frequency table of city and state_name
table(final_merged_data$city)
##
##         Auburn      Bellevue Black Diamond       Bothell     Carnation
##            911          1407           100           195           124
##         Duvall      Enumclaw     Fall City   Federal Way      Issaquah
##            190           233            80           779           733
##        Kenmore          Kent      Kirkland  Maple Valley        Medina
##            283          1201           977           589            50
## Mercer Island    North Bend       Redmond        Renton     Sammamish
##            282           220           977          1597           800
##        Seattle    Snoqualmie        Vashon   Woodinville
##           8973           308           117           471
```

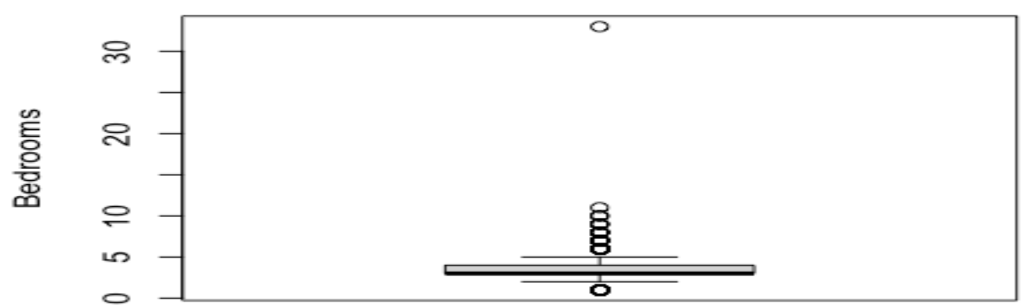
```
table(final_merged_data$state_name)
##
## Washington
##      21597
final_merged_data$state_name <- NULL
```

Detecting Outliers if any :

```
boxplot(final_merged_data$price, ylab = "Price")
```




```
boxplot(final_merged_data$bedrooms, ylab = "Bedrooms")
```



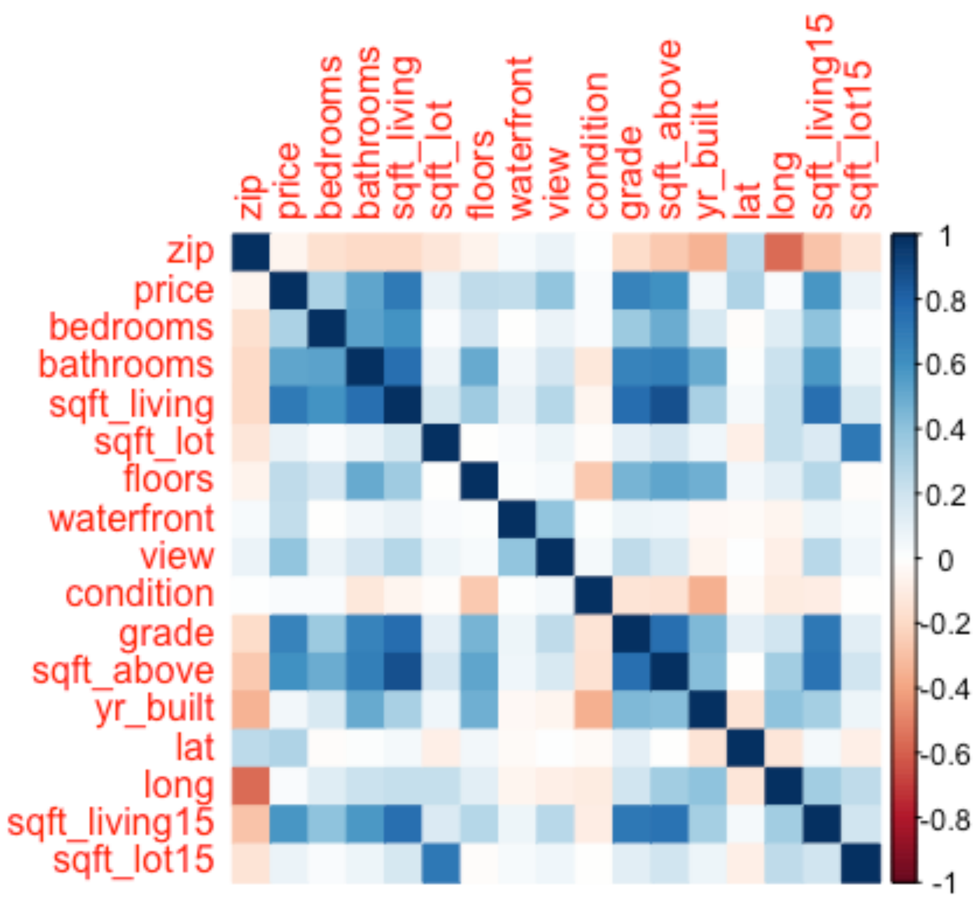
3. Exploratory Data Analysis

```
## missing value check
na_check=data.frame(no_of_na_values=colSums(is.na(final_merged_data)))
head(na_check,5)
##              no_of_na_values
## zip                        0
## price                      0
## bedrooms                   0
```

```
## bathrooms                     0
## sqft_living                   0
## Sampling the data
set.seed(123)
split = sample.split(final_merged_data$zip,SplitRatio = 0.7)
train =subset(final_merged_data,split == TRUE)
test =subset(final_merged_data, split == FALSE)
dim(train)
## [1] 15116    20
View(train)
dim(test)
## [1] 6481   20
```

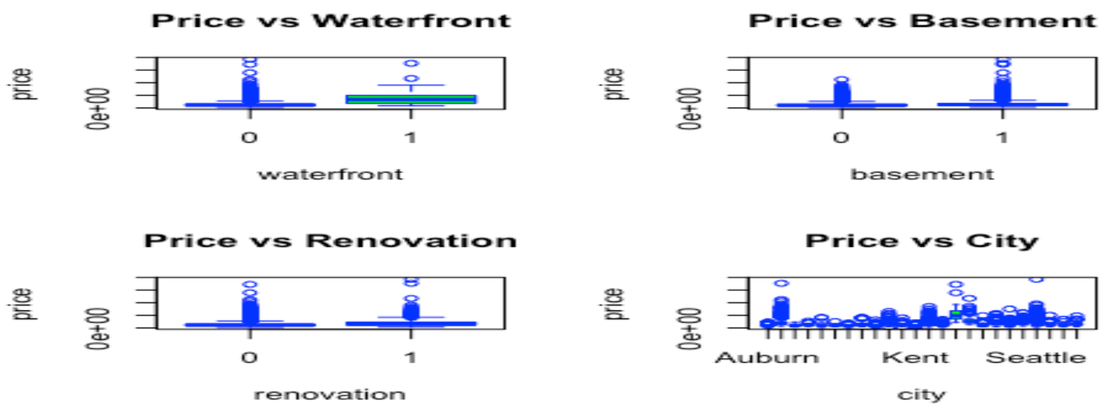
Finding the correlation and plotting the features using heatmap

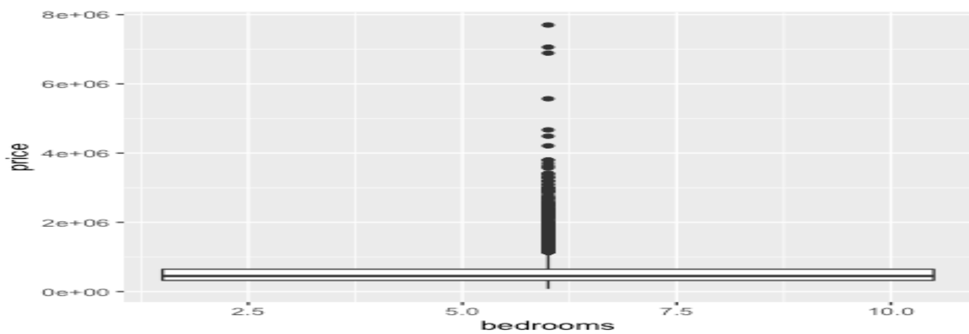
```
corr_data=data.frame(train[,1:20])
corr_data = corr_data[, -c(18:21)]

correlation=cor(corr_data)
par(mfrow=c(1, 1))
corrplot(correlation,method="color")
```



To get clear view of relationships, we plot the boxplots.

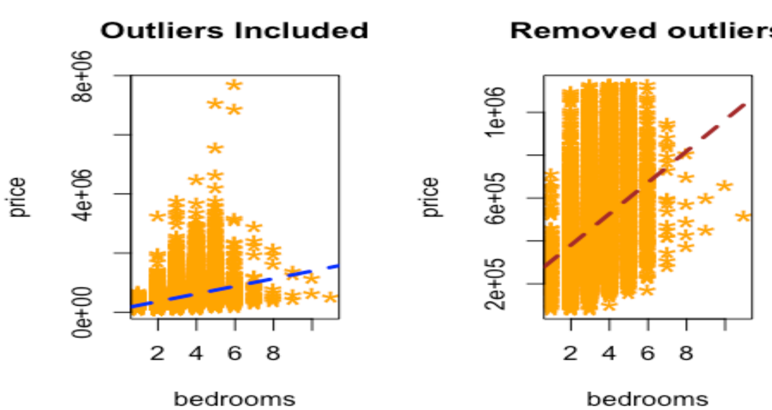
In descriptive statistics, a box plot or boxplot (also known as box and whisker plot) is a type of chart often used in explanatory data analysis. Box plots visually show the distribution of numerical data and skewness through displaying the data quartiles (or percentiles) and averages.





Outliers and its effect:

An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. The outliers contained in sample data introduce bias into statistical estimates such as mean values, leading to under- or over-estimated resulting values. So we need to ensure such points in the data are found and taken care of.We try plotting the data with and without outliers to understand the change in the slope.



The above diagram shows how the model changes drastically based on whether the outliers are considered or not.To solve the question as to whether there any significant difference in the house sale price based on the house features, we use ANOVA.

Using ANOVA:

An ANOVA test is a type of statistical test used to determine if there is a statistically significant difference between two or more categorical groups by testing for differences of means using variance. Another Key part of ANOVA is that it splits the independent variable into 2 or more groups. For example, one or more groups might be expected to influences the dependent variable while the other group is used as a control group, and is not expected to influence the dependent variable.

There are different types of ANOVA tests. The two most common are a "One-Way" and a "Two-Way."The difference between these two types depends on the number of independent variables in your test.The ANOVA *F* value can tell you if there is a significant difference between the levels of the independent variable, when $p < .05$. So, a higher F value indicates that the treatment variables are significant.

Our Objective:

To investigate if the condition, renovation and city located has any effect on the house sale price. For this purpose , we use one way ANOVA to find out if the categorical variables caused a difference in the mean of the house prices. The Null Hypothesis (H0) and the alternate hypothesis are taken as mentioned below:

H0: No difference in the means

H1: the means are different from one another.

As ANOVA only reveals those that are different between group means, we use Turkey's Honestly Significant Difference (HSD) test to tell which groups are statistically different from each other.

Step Implemented:

1. Analyse the factor level of the variable
2. Perform ANOVA test.
3. Boxplot the price and variable to check the distribution.

Analysis of variance (ANOVA) for variable "condition"

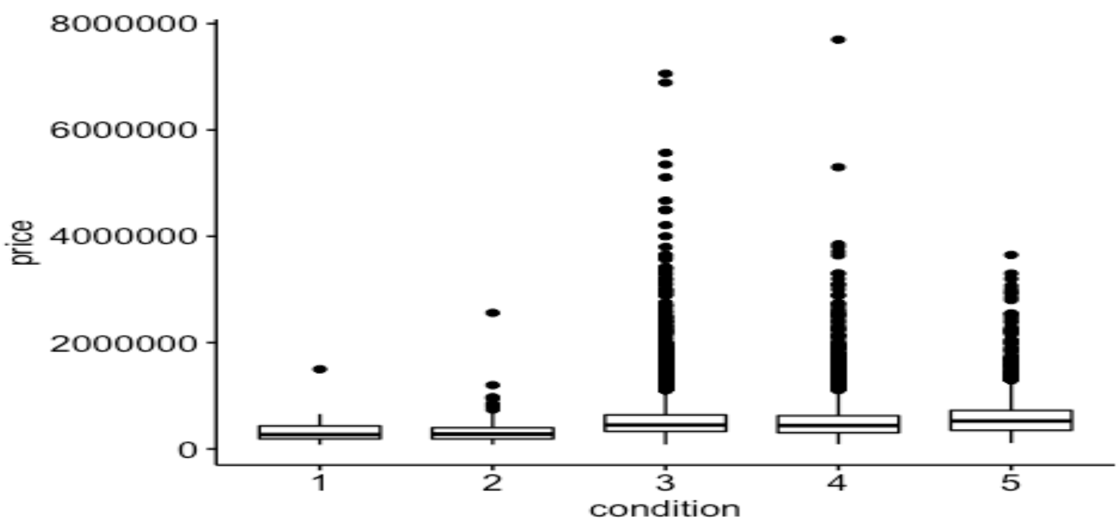
Hypothesis:

H0: The mean price is equal for all levels of " condition " categories.

Ha: At least one of the " condition " categories has a mean " price" that is not the same as the other " c condition" categories.

```
## Anova for price vs condition and plotting the distribution
## Calculate frequency, mean and standard deviation
final_merged_data %>% group_by(condition) %>% summarise(condition_freq = n(),
price_mean = mean(price, na.rm = TRUE), price_sd = sd(price, na.rm = TRUE))
## # A tibble: 5 × 4
##   condition condition_freq price_mean price_sd
##       <dbl>          <int>      <dbl>    <dbl>
## 1         1             29    341067.  273483.
## 2         2            170    328179.  246987.
## 3         3          14020    542173.  364650.
```

```
## 4         4           5677    521374.  358796.
## 5         5           1701    612578.  411318.
anova_cond <- aov(price ~ condition, data = final_merged_data)
summary(anova_cond)
##                Df    Sum Sq   Mean Sq F value   Pr(>F)
## condition       1 3.789e+12 3.789e+12   28.11 1.16e-07 ***
## Residuals   21595 2.911e+15 1.348e+11
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
options(scipen=999)
ggboxplot(final_merged_data, x = "condition", y = "price", ylim=c(78000,77000
00))
```



From the above, we see that the p- value of the condition variable is vey low (p<0.0001), which implies that the value of condition has impact on the sale price of the house. Hence our assumption of NULL HYPOTHESIS is rejected we accept the alternate hypothesis.

Analysis of variance (ANOVA) for variable “renovation”:

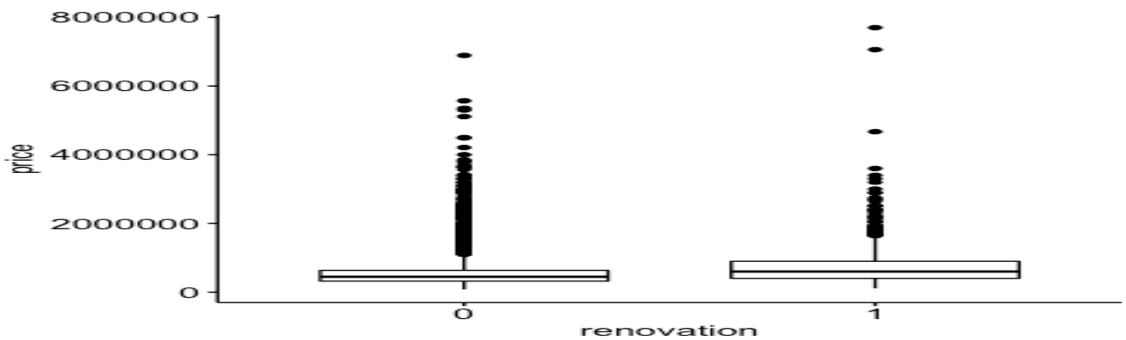
Hypothesis:

H0: The mean “ price” is equal for all levels of “renovation” charges

Ha: At least one of the “ renovation” categories have a mean “price” that is not the same as the other “renovation” categories.

```
## Anova for price vs renovation and plotting the distribution
## Calculate frequency, mean and standard deviation
final_merged_data %>% group_by(renovation) %>% summarise(renovation_freq = n(
), price_mean = mean(price, na.rm = TRUE), price_sd = sd(price, na.rm = TRUE)
)
## # A tibble: 2 × 4
##   renovation renovation_freq price_mean price_sd
##   <chr>                <int>      <dbl>    <dbl>
```

```
## 1 0                      20683    530560.  349805.
## 2 1                        914    760629.  608017.
anova_reno <- aov(price ~ renovation, data = final_merged_data)
summary(anova_reno)
##                Df           Sum Sq        Mean Sq F value              Pr(
>F)
## renovation      1   46332107051977 46332107051977   348.8 <0.0000000000000
002
## Residuals   21595 2868250023356214   132820098326
##
## renovation  ***
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
options(scipen=999)
ggboxplot(final_merged_data, x = "renovation", y = "price", ylim=c(78000,7700
000))
```



The p-value of the "renovation" variable is low(p<0.001), which again implied that the null hypothesis is rejected, Hence we confirm that there is a significant difference in the average price of the house based on the renovation state of the house.

Analysis of variance (ANOVA) for variable "city":

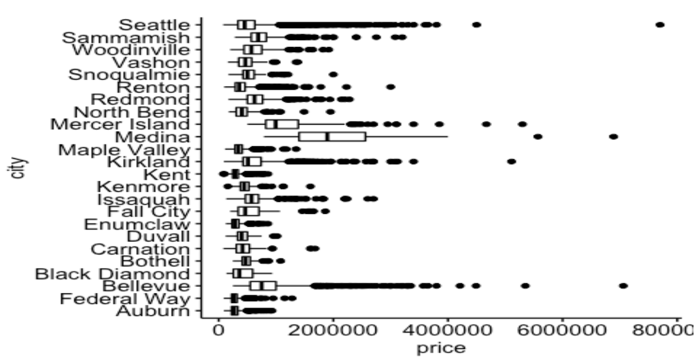
Hypothesis:

H0: The mean "price" is equal for all levels of "city" categories.

H1: At least one of the " city" categories has a mean price that is not the same as the other "city" categories.

```
## Anova for price vs city and plotting the distribution
## Calculate frequency, mean and standard deviation
options(dplyr.print_max = 1e9)
final_merged_data %>% group_by(city) %>% summarise(city_freq = n(), price_mea
n = mean(price, na.rm = TRUE), price_sd = sd(price, na.rm = TRUE))
## # A tibble: 24 × 4
##    city          city_freq price_mean price_sd
##    <chr>             <int>      <dbl>    <dbl>
##  1 Auburn              911    291648.  108422.
##  2 Bellevue           1407    898466.  559782.
##  3 Black Diamond       100    423666.  195415.
```

```
##   4 Bothell              195    490377.  121971.
##   5 Carnation            124    455617.  258603.
##   6 Duvall               190    424815.  130638.
##   7 Enumclaw             233    316742.  122329.
##   8 Fall City             80    586121.  376719.
##   9 Federal Way          779    289391.  108399.
##  10 Issaquah             733    615122.  260451.
##  11 Kenmore              283    462489.  149530.
##  12 Kent                1201    299470.   91647.
##  13 Kirkland             977    646543.  409633.
##  14 Maple Valley         589    367091.  132721.
##  15 Medina                50   2161300  1166904.
##  16 Mercer Island        282   1194874.  607768.
##  17 North Bend           220    440232.  207554.
##  18 Redmond              977    658432.  231136.
##  19 Renton              1597    403468.  200725.
##  20 Sammamish            800    732821.  280951.
##  21 Seattle             8973    535086.  340519.
##  22 Snoqualmie           308    529630.  185254.
##  23 Vashon               117    489382.  201501.
##  24 Woodinville          471    617498.  244298.
anova_city <- aov(price ~ city, data = final_merged_data)
summary(anova_city)
##                Df          Sum Sq        Mean Sq F value              Pr(
>F)
## city           23   738104329040975 32091492566999   318.1 <0.0000000000000
002
## Residuals   21573 2176477801366934   100888972390
##
## city        ***
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
options(scipen=999)
ggboxplot(final_merged_data, x = "city", y = "price", ylim=c(78000,7700000))
+ coord_flip()
## Coordinate system already present. Adding new coordinate system, which wil
l replace the existing one.
```



The p- value of the “city” variable is low(p<0.001), which again implies that location of the city gives impact on the house sale price. Hence the null hypothesis H0 is rejected which means there is a significant difference in the average price of house based on the location of the house. Now, we use multiple linear regression to get a prediction model on the house price based on the selected variables such as – bedrooms, bathrooms, floors, waterfront, condition,sqft_living15, sqft_lot15,basement and renovation. We will apply machine learning algorithm on the multiple linear regression analysis.

Steps we follow:

a. Construct a linear model.

b. Remove outliers but keep the influential points for further analysis.

c. Prediction of house price.

4. Model Building

Multiple Linear Regression:

```
model <- lm(price~bedrooms+bathrooms+floors+waterfront+condition+sqft_living1
5+sqft_lot15+basement+renovation,data=train)
summary(model)
##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + floors + waterfront +
##     condition + sqft_living15 + sqft_lot15 + basement + renovation,
##     data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1291533  -149571   -25169   103034  5787440
##
## Coefficients:
##                   Estimate    Std. Error t value             Pr(>|t|)
## (Intercept)   -455045.81774   16437.09336 -27.684 <0.0000000000000002 ***
## bedrooms        -5625.32457    2905.24193  -1.936               0.0529 .
## bathrooms      101112.98774    4360.25414  23.190 <0.0000000000000002 ***
## floors          53625.05780    5194.68697  10.323 <0.0000000000000002 ***
## waterfront     749134.27427   25185.58577  29.745 <0.0000000000000002 ***
## condition       59809.81332    3480.52171  17.184 <0.0000000000000002 ***
## sqft_living15     235.64690       3.97983  59.210 <0.0000000000000002 ***
## sqft_lot15         -0.27190       0.08391  -3.240               0.0012 **
## basement1       91667.10118    4928.25585  18.600 <0.0000000000000002 ***
## renovation1    201428.42659   10754.29353  18.730 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 266000 on 15106 degrees of freedom
## Multiple R-squared:  0.4709, Adjusted R-squared:  0.4706
## F-statistic:  1494 on 9 and 15106 DF,  p-value: < 0.00000000000000022
```

```r
model_fit <- lm(price~bedrooms+bathrooms+floors+waterfront+condition+sqft_liv
ing15+sqft_lot15+basement+renovation, data=train)
s <- stepAIC(model_fit, direction="both")
## Start:  AIC=377642.2
## price ~ bedrooms + bathrooms + floors + waterfront + condition + 
##     sqft_living15 + sqft_lot15 + basement + renovation
## 
##                  Df       Sum of Sq              RSS    AIC
## <none>                              1068603616353456 377642
## - bedrooms       1    265214779753 1068868831133209 377644
## - sqft_lot15     1    742749127358 1069346365480814 377651
## - floors         1   7538482691713 1076142099045169 377746
## - condition      1  20889274545049 1089492890898505 377933
## - basement       1  24474152529446 1093077768882902 377982
## - renovation     1  24816750138906 1093420366492362 377987
## - bathrooms      1  38041483134715 1106645099488171 378169
## - waterfront     1  62586747317008 1131190363670464 378501
## - sqft_living15  1 248005386066731 1316609002420186 380795
s$anova
## Stepwise Model Path 
## Analysis of Deviance Table
## 
## Initial Model:
## price ~ bedrooms + bathrooms + floors + waterfront + condition + 
##     sqft_living15 + sqft_lot15 + basement + renovation
## 
## Final Model:
## price ~ bedrooms + bathrooms + floors + waterfront + condition + 
##     sqft_living15 + sqft_lot15 + basement + renovation
## 
## 
##   Step Df Deviance Resid. Df       Resid. Dev      AIC
## 1                      15106 1068603616353456 377642.2
linear_model1 <- lm(price~bedrooms+bathrooms+floors+waterfront+condition+sqft
_living15+basement+renovation, data=train)
summary(linear_model1)
## 
## Call:
## lm(formula = price ~ bedrooms + bathrooms + floors + waterfront + 
##     condition + sqft_living15 + basement + renovation, data = train)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -1296654  -150346   -25449   102864  5792383 
## 
## Coefficients:
##                   Estimate  Std. Error t value            Pr(>|t|)    
## (Intercept)   -456822.205   16433.113 -27.799 <0.0000000000000002 ***
## bedrooms        -5206.009    2903.271  -1.793               0.073 .  
## bathrooms      100696.931    4359.733  23.097 <0.0000000000000002 ***
```

```
## floors          55046.626    5177.755  10.631 <0.0000000000000002 ***
## waterfront     747541.902   25188.707  29.678 <0.0000000000000002 ***
## condition       59763.067    3481.586  17.165 <0.0000000000000002 ***
## sqft_living15     233.324       3.916  59.583 <0.0000000000000002 ***
## basement1       92843.074    4916.420  18.884 <0.0000000000000002 ***
## renovation1    201291.101   10757.591  18.712 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 266100 on 15107 degrees of freedom
## Multiple R-squared:  0.4706, Adjusted R-squared:  0.4703
## F-statistic:  1678 on 8 and 15107 DF,  p-value: < 0.00000000000000022
# train the model and store the bootstrap in a dataframe
model_training <- train(price~bedrooms+bathrooms+floors+waterfront+condition+
sqft_living15+basement+renovation, data=train, method="lm")
summary(model_training)
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1296654  -150346   -25449   102864  5792383
##
## Coefficients:
##                  Estimate  Std. Error t value            Pr(>|t|)
## (Intercept)   -456822.205   16433.113 -27.799 <0.0000000000000002 ***
## bedrooms        -5206.009    2903.271  -1.793               0.073 .
## bathrooms      100696.931    4359.733  23.097 <0.0000000000000002 ***
## floors          55046.626    5177.755  10.631 <0.0000000000000002 ***
## waterfront     747541.902   25188.707  29.678 <0.0000000000000002 ***
## condition       59763.067    3481.586  17.165 <0.0000000000000002 ***
## sqft_living15     233.324       3.916  59.583 <0.0000000000000002 ***
## basement1       92843.074    4916.420  18.884 <0.0000000000000002 ***
## renovation1    201291.101   10757.591  18.712 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 266100 on 15107 degrees of freedom
## Multiple R-squared:  0.4706, Adjusted R-squared:  0.4703
## F-statistic:  1678 on 8 and 15107 DF,  p-value: < 0.00000000000000022
model_training_r2 <- summary(model_training$finalModel)$r.squared
model_training_results <- as.data.frame(model_training$results)
```

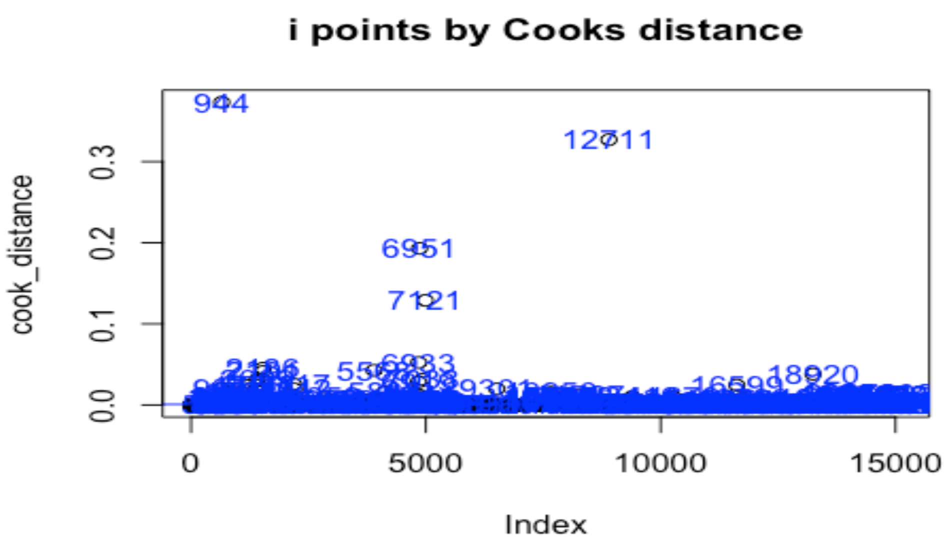
Note that full model of multiple linear regression model gave a R-square value of 0.4703. Also note that the step wise regression also showed that all the variables should be retained in the model. We tried removing sqft_lot15 above from the model and even after that, the R-square value remains the same. Hence we may remove that variable from the model.

Detection of Influential points:

An influential point is a point that has a large impact on the regression. Surprisingly, outlier and influential points are not the same thing. A point can be an outlier without being influential. A point can be influential without being an outlier. In other words, an influential observation is an observation for a statistical calculation whose deletion from the dataset would noticeably change the result of the calculation. In particular, in regression analysis an influential observation is one whose deletion has a large effect on the parameter estimates.
To check on influential points, three possible methods can be used. They are scatter plots, partial plots, and Cook's distances. Simple scatterplots will display the values of each independent variable plotted against the dependent variable.

We use Cook's distance in our project to detect the points.

```
cook_distance <- cooks.distance(linear_model1)
sprintf("The mean of Cook's distance is : %f ", mean(cook_distance))
## [1] "The mean of Cook's distance is : 0.000203 "
par(mfrow=c(1, 1))
plot(cook_distance, main="i points by Cooks distance")
abline(h = 4*mean(cook_distance, na.rm=T), col="blue")
text(x=1:length(cook_distance)+1,y=cook_distance,labels=ifelse(cook_distance>
4*mean(cook_distance,na.rm=T),names(cook_distance),""), col="blue")
```



After finding the influence points and outliers in our training data based on the cook distance, we modify the data and create a new model.

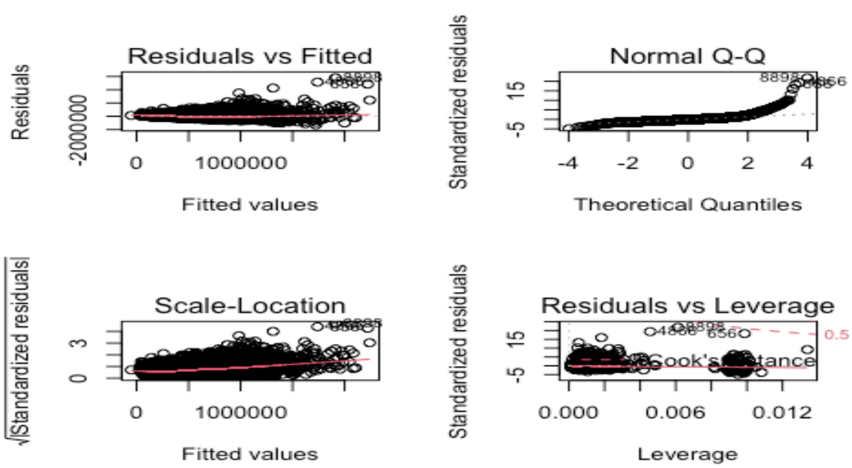
```
t2 <- rbind(train,i_ol)
row.names(t2) <- NULL
linear_model2 <- lm(price~bedrooms+bathrooms+floors+waterfront+condition+sqft
_living15+basement+renovation, data=t2)
summary(linear_model2)
##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + floors + waterfront +
##     condition + sqft_living15 + basement + renovation, data = t2)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1298667  -150507   -25503   103345  5787812
##
## Coefficients:
##                 Estimate  Std. Error t value             Pr(>|t|)
## (Intercept)  -457734.108   16440.466 -27.842 <0.0000000000000002 ***
## bedrooms       -5671.758    2903.978  -1.953               0.0508 .
## bathrooms     101452.341    4358.815  23.275 <0.0000000000000002 ***
## floors         54386.975    5178.322  10.503 <0.0000000000000002 ***
## waterfront    748345.795   25097.016  29.818 <0.0000000000000002 ***
## condition      59751.564    3483.271  17.154 <0.0000000000000002 ***
## sqft_living15    234.441       3.909  59.970 <0.0000000000000002 ***
## basement1      92803.861    4917.781  18.871 <0.0000000000000002 ***
## renovation1   200856.736   10765.092  18.658 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 266300 on 15119 degrees of freedom
## Multiple R-squared:  0.473,  Adjusted R-squared:  0.4727
## F-statistic:  1696 on 8 and 15119 DF,  p-value: < 0.00000000000000022
```

We see that our R-square value increased a little bit in this approach. Also the Residual Standard Error decreased as compared to the model with outliers data. The model with revised data is significant with all the variables strongly related to the price.

5. Model Evaluation

```
## regression diagonstics
par(mfrow = c(2, 2))
plot(linear_model2)
```



Residual vs. Fitted Plot shows that the relationship between price and predictors is linear. The normal Q-Q plot on the other hand shows a straight line which indicates that the residuals are normally distributed. The Scale- Location plot shows almost uniform dispersion of the fitted values which means the homoscedasticity of the residuals (equal variance). Finally the Residuals vs Leverage Plot shows there is no leverage out of the boundaries.

Multicollinearity test:

Multicollinearity is a statistical concept where several independent variables in a model are correlated. Two variables are considered to be perfectly collinear if their correlation coefficient is +/- 1.0. Multicollinearity among independent variables will result in less reliable statistical inferences.

This correlation is a problem because independent variables should be independent. If the degree of correlation between variables is high enough, it can cause problems when you fit the model and interpret the results.
A simple method to detect multicollinearity in a model is by using something called the **variance inflation factor** or the **VIF** for each predicting variable.

VIF measures the ratio between the variance for a given regression coefficient with only that variable in the model versus the variance for a given regression coefficient with all variables in the model.A VIF of 1 (the minimum possible VIF) means the tested predictor is not correlated with the other predictors.A VIF of 1 (the minimum possible VIF) means the tested predictor is not correlated with the other predictors.
An acceptable VIF is if it's less than the max of 10

```
## multicollinearilty test

vif(linear_model2)
##      bedrooms     bathrooms        floors    waterfront     condition
##      1.465613      2.424348      1.656469      1.022634      1.096146
## sqft_living15      basement    renovation
##      1.550625      1.235218      1.022778
## accuracy
prediction_test=predict(newdata=test, linear_model2)
actual_model_fitted_test=data.frame(actual=test$price, predicted=prediction_t
est)
abs_diff_test = mean(abs(actual_model_fitted_test$actual-actual_model_fitted_
test$predicted)/actual_model_fitted_test$actual)
accuracy=1-abs_diff_test
sprintf(" The accuracy of the prediction on test data is : %f",accuracy*100)
## [1] " The accuracy of the prediction on test data is : 63.784002"
```

6. Conclusion

From the values of VIF above, we can clearly say that there is no multicollinearity in the model. After training the model on training data using the Multiple Linear Regression, we test our model for the test data and accuracy of the model on the test data is 63.78%.

7. Future Works

We can increase the accuracy by using different machine learning models like boosting and bagging techniques. Also we can use other advanced machine learning models which help us in better prediction of the house prices.

8. References

1. https://www.kaggle.com/harlfoxem/housesalesprediction.
2. https://www.researchgate.net/publication/347584803_House_Price_Prediction_using_a_Machine_Learning_Model_A_Survey_of_Literature
3. C. R. Madhuri, G. Anuradha and M. V. Pujitha, "House Price Prediction Using Regression Techniques: A Comparative Study," 2019 International Conference on Smart Structures and Systems (ICSSS), 2019, pp. 1-5, doi: 10.1109/ICSSS.2019.8882834.

9. Project Link

GitHub