MATH-564 Project

House Prices Data Analytics

A20495939 - Jasleen Kaur Bhatia

A20504279 - Sajesh Rao Erabelli

Problem Statement

House price varies based on the condition of itself and the environment. From the number of bedrooms to the location of the house, any variable might be the key that affects the house price the most. In this project , we will use ANOVA and MLR to determine the relation of house situations with sold price and predict the house price. ### Loading Libraries

```
library(readr)
library(stringr)
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(caTools)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(ggplot2)
library(grid)
library(lattice)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

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

```
library(ggpubr)
library(tidyverse)
```

```
## ── Attaching packages
## ───────────────────────────────────────
## tidyverse 1.3.2 ──
```

```
## ✔ tibble  3.1.8     ✔ purrr   0.3.5
## ✔ tidyr   1.2.1     ✔ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ gridExtra::combine() masks dplyr::combine()
## ✖ dplyr::filter()      masks stats::filter()
## ✖ dplyr::lag()         masks stats::lag()
```

```
library(broom)
library(AICcmodavg)
library(caret)
```

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

```
library(leaps)
library(MASS)
```

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

```
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:purrr':
##
##     some
##
## The following object is masked from 'package:dplyr':
##
##     recode
```

Loading Datasets

```
house_datasales <- read_csv("/Users/jasleenkaurbhatia/Desktop/Semester3/Applied_Stats/AS
Project/kc_house_data.csv")
```

```
## Rows: 21597 Columns: 21
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (1): date
## dbl (20): id, price, bedrooms, bathrooms, sqft_living, sqft_lot, floors, wat...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(house_datasales)
```

```
## # A tibble: 6 × 21
##           id date     price bedro…¹ bathr…² sqft_…³ sqft_…⁴ floors water…⁵  view
##        <dbl> <chr>    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>  <dbl>   <dbl> <dbl>
## 1 7129300520 10/13/… 2.22e5       3    1       1180    5650      1       0     0
## 2 6414100192 12/9/2… 5.38e5       3    2.25    2570    7242      2       0     0
## 3 5631500400 2/25/2… 1.8 e5       2    1        770   10000      1       0     0
## 4 2487200875 12/9/2… 6.04e5       4    3       1960    5000      1       0     0
## 5 1954400510 2/18/2… 5.1 e5       3    2       1680    8080      1       0     0
## 6 7237550310 5/12/2… 1.23e6       4    4.5     5420  101930      1       0     0
## # … with 11 more variables: condition <dbl>, grade <dbl>, sqft_above <dbl>,
## #   sqft_basement <dbl>, yr_built <dbl>, yr_renovated <dbl>, zipcode <dbl>,
## #   lat <dbl>, long <dbl>, sqft_living15 <dbl>, sqft_lot15 <dbl>, and
## #   abbreviated variable names ¹bedrooms, ²bathrooms, ³sqft_living, ⁴sqft_lot,
## #   ⁵waterfront
```

```
colnames(house_datasales)
```

```
##  [1] "id"            "date"          "price"         "bedrooms"
##  [5] "bathrooms"     "sqft_living"   "sqft_lot"      "floors"
##  [9] "waterfront"    "view"          "condition"     "grade"
## [13] "sqft_above"    "sqft_basement" "yr_built"      "yr_renovated"
## [17] "zipcode"       "lat"           "long"          "sqft_living15"
## [21] "sqft_lot15"
```

```
dim(house_datasales)
```

```
## [1] 21597    21
```

```
str(house_datasales)
```

```
## spc_tbl_ [21,597 × 21] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ id           : num [1:21597] 7.13e+09 6.41e+09 5.63e+09 2.49e+09 1.95e+09 ...
##  $ date         : chr [1:21597] "10/13/2014" "12/9/2014" "2/25/2015" "12/9/2014" ...
##  $ price        : num [1:21597] 221900 538000 180000 604000 510000 ...
##  $ bedrooms     : num [1:21597] 3 3 2 4 3 4 3 3 3 3 ...
##  $ bathrooms    : num [1:21597] 1 2.25 1 3 2 4.5 2.25 1.5 1 2.5 ...
##  $ sqft_living  : num [1:21597] 1180 2570 770 1960 1680 ...
##  $ sqft_lot     : num [1:21597] 5650 7242 10000 5000 8080 ...
##  $ floors       : num [1:21597] 1 2 1 1 1 1 2 1 1 2 ...
##  $ waterfront   : num [1:21597] 0 0 0 0 0 0 0 0 0 0 ...
##  $ view         : num [1:21597] 0 0 0 0 0 0 0 0 0 0 ...
##  $ condition    : num [1:21597] 3 3 3 5 3 3 3 3 3 3 ...
##  $ grade        : num [1:21597] 7 7 6 7 8 11 7 7 7 7 ...
##  $ sqft_above   : num [1:21597] 1180 2170 770 1050 1680 ...
##  $ sqft_basement: num [1:21597] 0 400 0 910 0 1530 0 0 730 0 ...
##  $ yr_built     : num [1:21597] 1955 1951 1933 1965 1987 ...
##  $ yr_renovated : num [1:21597] 0 1991 0 0 0 ...
##  $ zipcode      : num [1:21597] 98178 98125 98028 98136 98074 ...
##  $ lat          : num [1:21597] 47.5 47.7 47.7 47.5 47.6 ...
##  $ long         : num [1:21597] -122 -122 -122 -122 -122 ...
##  $ sqft_living15: num [1:21597] 1340 1690 2720 1360 1800 ...
##  $ sqft_lot15   : num [1:21597] 5650 7639 8062 5000 7503 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   id = col_double(),
##   ..   date = col_character(),
##   ..   price = col_double(),
##   ..   bedrooms = col_double(),
##   ..   bathrooms = col_double(),
##   ..   sqft_living = col_double(),
##   ..   sqft_lot = col_double(),
##   ..   floors = col_double(),
##   ..   waterfront = col_double(),
##   ..   view = col_double(),
##   ..   condition = col_double(),
##   ..   grade = col_double(),
##   ..   sqft_above = col_double(),
##   ..   sqft_basement = col_double(),
##   ..   yr_built = col_double(),
##   ..   yr_renovated = col_double(),
##   ..   zipcode = col_double(),
##   ..   lat = col_double(),
##   ..   long = col_double(),
##   ..   sqft_living15 = col_double(),
##   ..   sqft_lot15 = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

So the total number of rows in housedata dataset is : 21597 and number of columns is 21.

Understanding the data :

```
summary(house_datasales)
```

```
##        id                 date               price            bedrooms
##  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
##  Max.   :1.000000   Max.   :4.0000   Max.   :5.00   Max.   :13.000
##    sqft_above   sqft_basement       yr_built     yr_renovated
##  Min.   : 370   Min.   :   0.0   Min.   :1900   Min.   :   0.00
##  1st Qu.:1190   1st Qu.:   0.0   1st Qu.:1951   1st Qu.:   0.00
##  Median :1560   Median :   0.0   Median :1975   Median :   0.00
##  Mean   :1789   Mean   : 291.7   Mean   :1971   Mean   :  84.46
##  3rd Qu.:2210   3rd Qu.: 560.0   3rd Qu.:1997   3rd Qu.:   0.00
##  Max.   :9410   Max.   :4820.0   Max.   :2015   Max.   :2015.00
##     zipcode           lat             long        sqft_living15
##  Min.   :98001   Min.   :47.16   Min.   :-122.5   Min.   : 399
##  1st Qu.:98033   1st Qu.:47.47   1st Qu.:-122.3   1st Qu.:1490
##  Median :98065   Median :47.57   Median :-122.2   Median :1840
##  Mean   :98078   Mean   :47.56   Mean   :-122.2   Mean   :1987
##  3rd Qu.:98118   3rd Qu.:47.68   3rd Qu.:-122.1   3rd Qu.:2360
##  Max.   :98199   Max.   :47.78   Max.   :-121.3   Max.   :6210
##    sqft_lot15
##  Min.   :   651
##  1st Qu.:  5100
##  Median :  7620
##  Mean   : 12758
##  3rd Qu.: 10083
##  Max.   :871200
```

```
for (column in house_datasales){
  print( typeof(column))
}
```

```
## [1] "double"
## [1] "character"
## [1] "double"
## [1] "double"
## [1] "double"
## [1] "double"
## [1] "double"
## [1] "double"
## [1] "double"
## [1] "double"
## [1] "double"
## [1] "double"
## [1] "double"
## [1] "double"
## [1] "double"
## [1] "double"
## [1] "double"
## [1] "double"
## [1] "double"
## [1] "double"
## [1] "double"
```

So we understand that we have 21 features and all but one have their datatype as double.Only one specific feature- date has the data type as “character”.

It would be better if we create a dataset without the values of date as that will allow us to undertsand the data better by using correlation and other functions/plots.

```
house_datasales1 <- house_datasales[,-1:-2]
colnames(house_datasales1)
```

```
##  [1] "price"         "bedrooms"      "bathrooms"     "sqft_living"
##  [5] "sqft_lot"      "floors"        "waterfront"    "view"
##  [9] "condition"     "grade"         "sqft_above"    "sqft_basement"
## [13] "yr_built"      "yr_renovated"  "zipcode"       "lat"
## [17] "long"          "sqft_living15" "sqft_lot15"
```

```
dim(house_datasales1)
```

```
## [1] 21597    19
```

```
#View(house_datasales1)
```

Loading USZIPCODE data:

```
zipcode_data <- read_csv("/Users/jasleenkaurbhatia/Desktop/Semester3/Applied_Stats/AS Pr
oject/uszips.csv")
```

```
## Rows: 33121 Columns: 18
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (10): zip, city, state_id, state_name, county_fips, county_name, county_...
## dbl  (4): lat, lng, population, density
## lgl  (4): zcta, parent_zcta, imprecise, military
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(zipcode_data)
```

```
## # A tibble: 6 × 18
##   zip     lat   lng city   state…¹ state…² zcta  paren…³ popul…⁴ density count…⁵
##   <chr> <dbl> <dbl> <chr>  <chr>   <chr>   <lgl> <lgl>     <dbl>   <dbl> <chr>
## 1 00601  18.2 -66.8 Adjun… PR      Puerto… TRUE  NA        17113   103.  72001
## 2 00602  18.4 -67.2 Aguada PR      Puerto… TRUE  NA        37751   476   72003
## 3 00603  18.5 -67.1 Aguad… PR      Puerto… TRUE  NA        47081   575.  72005
## 4 00606  18.2 -66.9 Maric… PR      Puerto… TRUE  NA         6392    58.3 72093
## 5 00610  18.3 -67.1 Anasco PR      Puerto… TRUE  NA        26686   287.  72011
## 6 00612  18.4 -66.7 Areci… PR      Puerto… TRUE  NA        59369   339.  72013
## # … with 7 more variables: county_name <chr>, county_weights <chr>,
## #   county_names_all <chr>, county_fips_all <chr>, imprecise <lgl>,
## #   military <lgl>, timezone <chr>, and abbreviated variable names ¹state_id,
## #   ²state_name, ³parent_zcta, ⁴population, ⁵county_fips
```

```
colnames(zipcode_data)
```

```
##  [1] "zip"              "lat"              "lng"              "city"
##  [5] "state_id"         "state_name"       "zcta"             "parent_zcta"
##  [9] "population"       "density"          "county_fips"      "county_name"
## [13] "county_weights"   "county_names_all" "county_fips_all"  "imprecise"
## [17] "military"         "timezone"
```

```
dim(zipcode_data)
```

```
## [1] 33121    18
```

So, uszips dataset have 33788 rows ansd 18 columns.

Understanding the USZIPS data :

```
summary(zipcode_data)
```

```
##      zip                  lat              lng              city          
##  Length:33121       Min.   :-14.22   Min.   :-176.63   Length:33121      
##  Class :character   1st Qu.: 35.39   1st Qu.: -97.23   Class :character  
##  Mode  :character   Median : 39.49   Median : -88.19   Mode  :character  
##                     Mean   : 38.82   Mean   : -90.92                     
##                     3rd Qu.: 42.12   3rd Qu.: -80.22                     
##                     Max.   : 71.27   Max.   : 145.75                     
##                                                                          
##    state_id          state_name          zcta         parent_zcta   
##  Length:33121       Length:33121       Mode:logical   Mode:logical  
##  Class :character   Class :character   TRUE:33121     NA's:33121    
##  Mode  :character   Mode  :character                                
##                                                                     
##                                                                     
##                                                                     
##                                                                     
##    population        density        county_fips        county_name       
##  Min.   :     0   Min.   :    0.0   Length:33121       Length:33121      
##  1st Qu.:   707   1st Qu.:    7.6   Class :character   Class :character  
##  Median :  2804   Median :   30.5   Mode  :character   Mode  :character  
##  Mean   :  9910   Mean   :  509.7                                        
##  3rd Qu.: 13481   3rd Qu.:  265.1                                        
##  Max.   :128294   Max.   :57641.1                                        
##  NA's   :24       NA's   :24                                             
##  county_weights     county_names_all   county_fips_all    imprecise      
##  Length:33121       Length:33121       Length:33121       Mode :logical  
##  Class :character   Class :character   Class :character   FALSE:33121    
##  Mode  :character   Mode  :character   Mode  :character                  
##                                                                          
##                                                                          
##                                                                          
##                                                                          
##   military         timezone        
##  Mode :logical   Length:33121      
##  FALSE:33121     Class :character  
##                  Mode  :character  
##                                    
##                                    
##                                    
##                                    
```

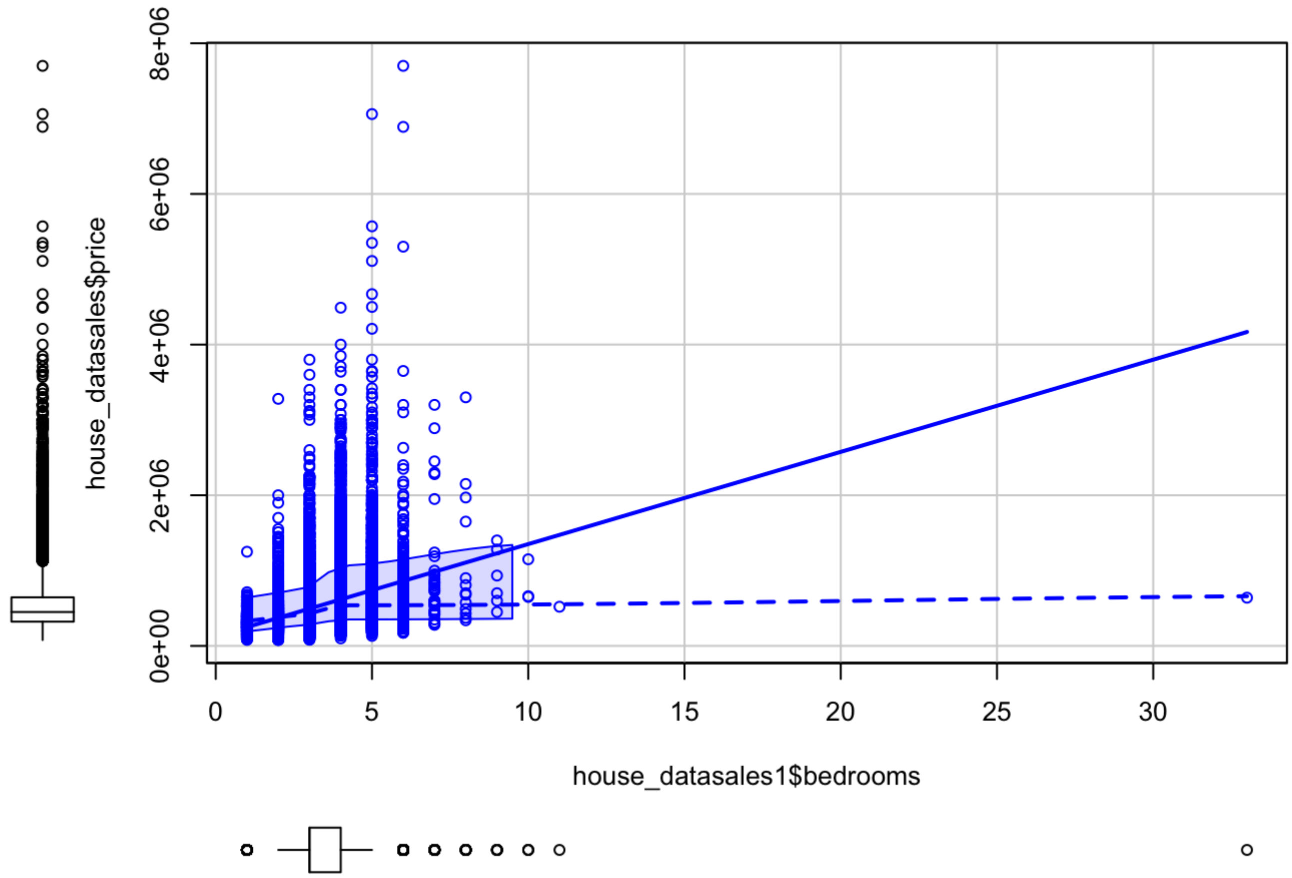
```
for (column in zipcode_data){
  print( typeof(column))
}
```

```
## [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"
```

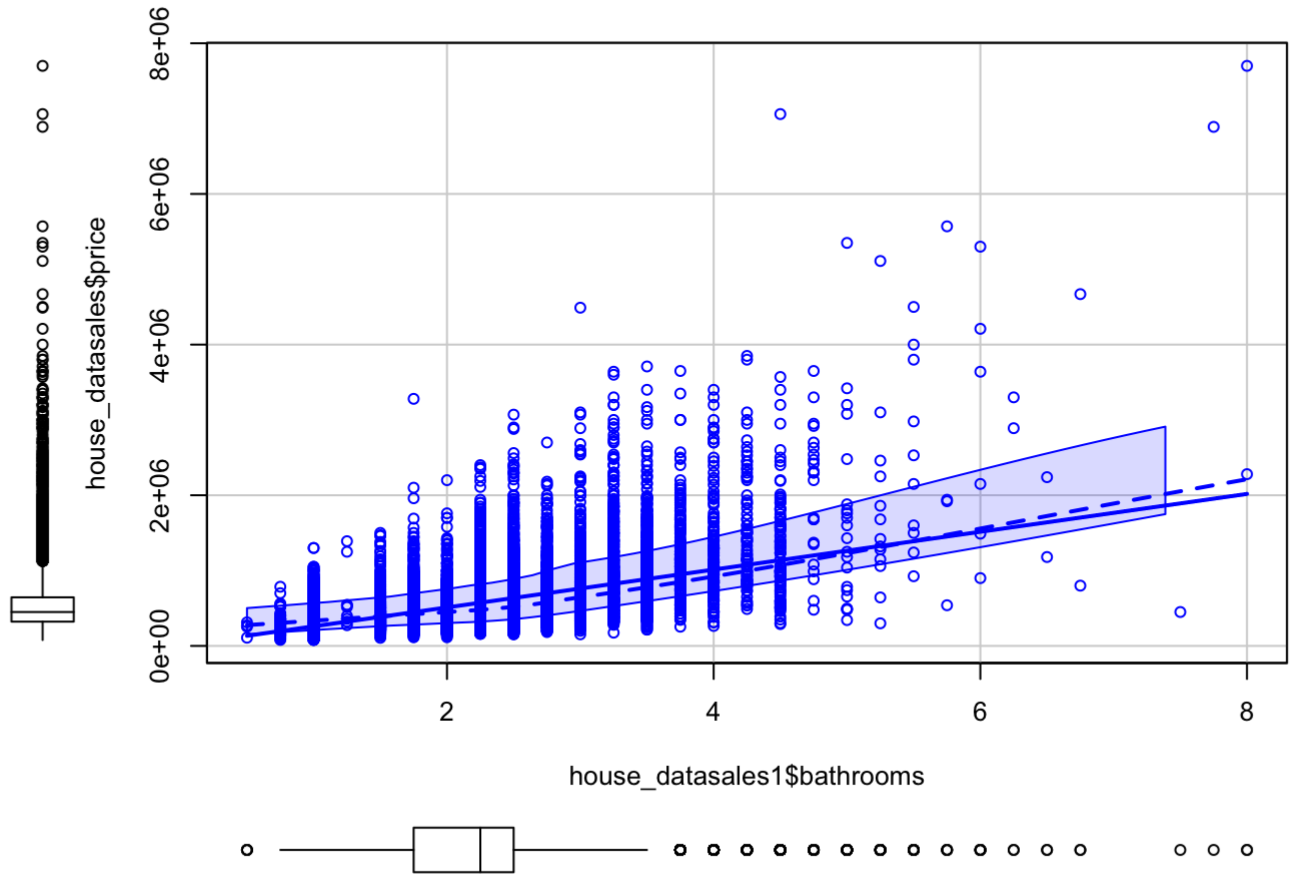
From the above, we understand that 10 features have their data type as character, 4 features have it as double and the remaining 4 are logical.

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.

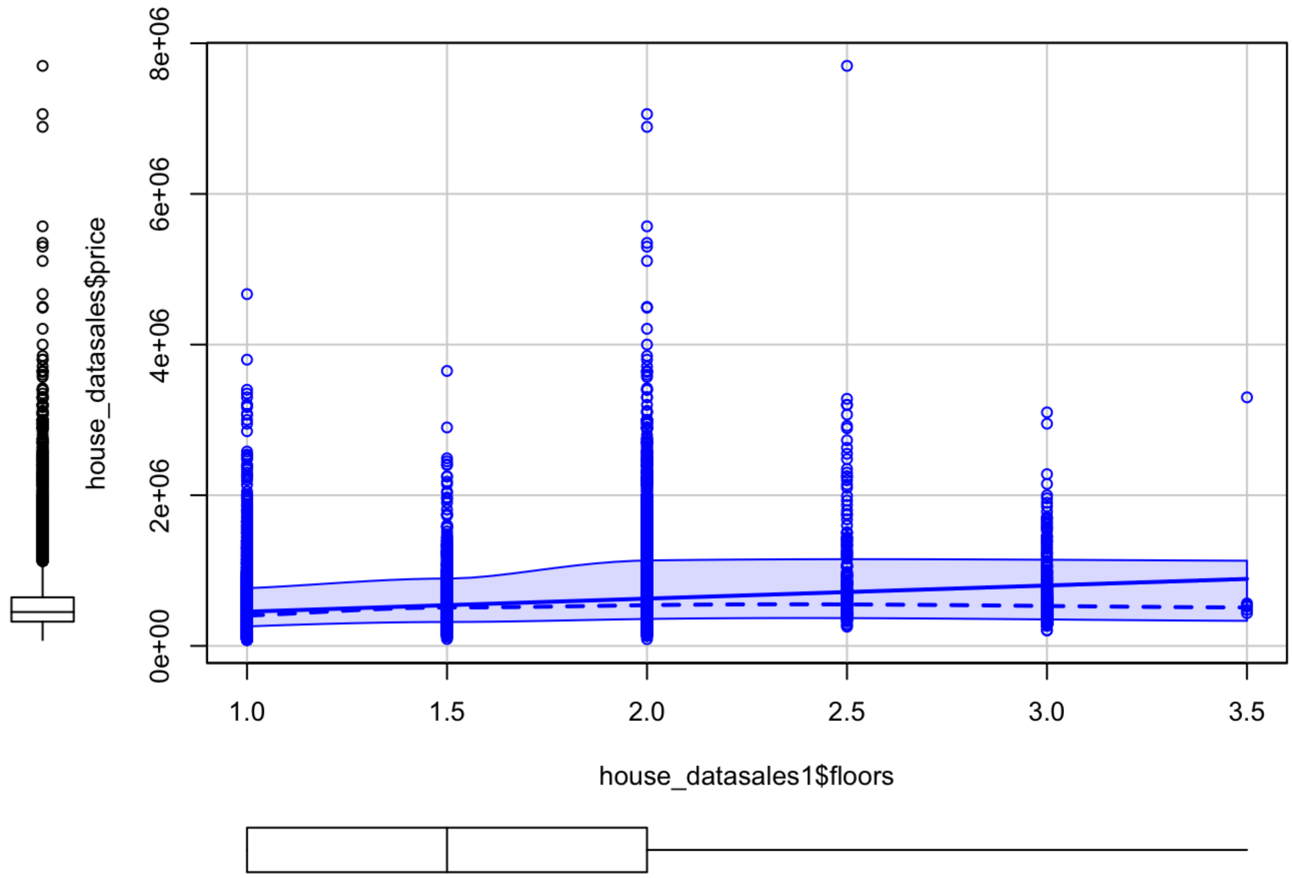
```
par(mfrow=c(4,5))
scatterplot(house_datasales1$bedrooms,house_datasales$price)
```



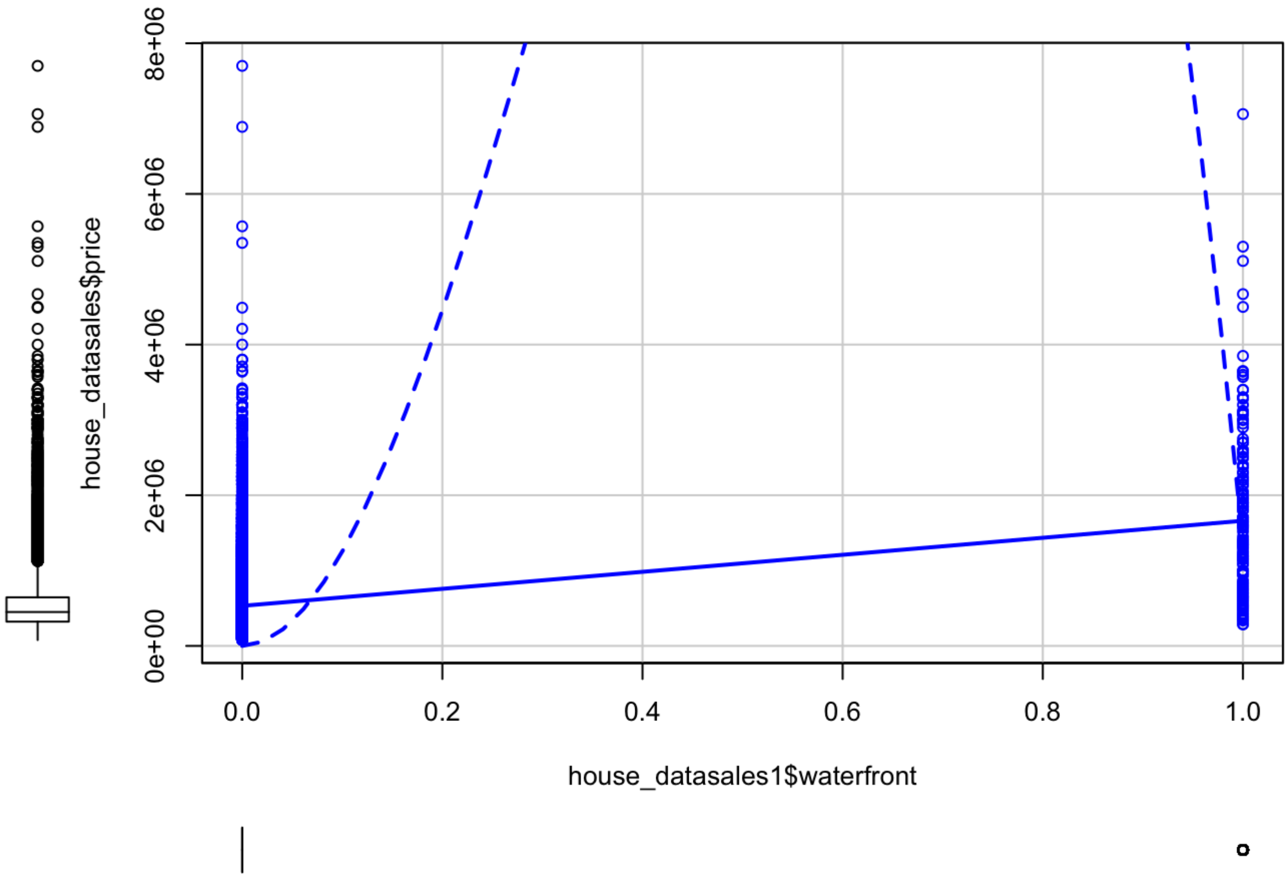

```
scatterplot(house_datasales1$bathrooms,house_datasales$price)
```



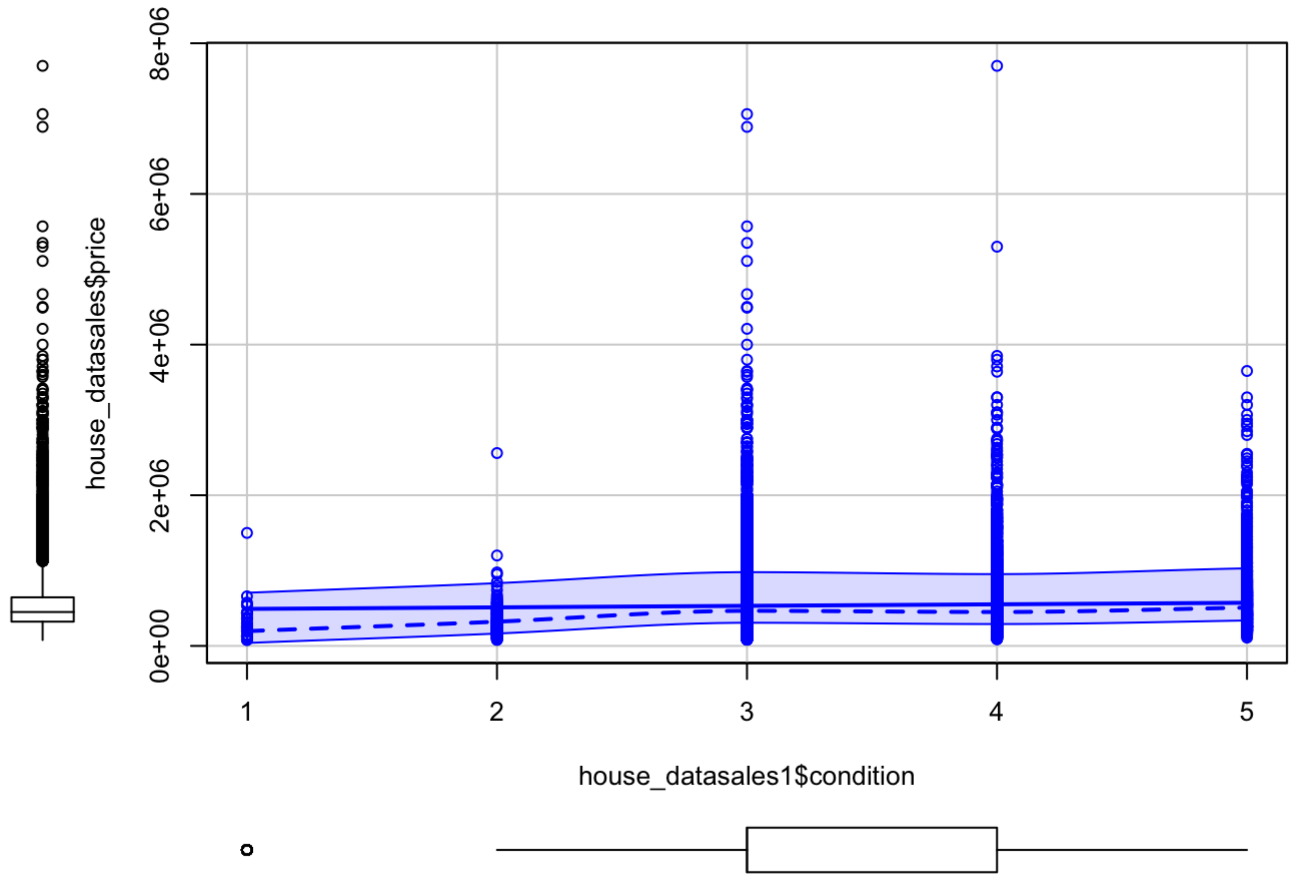

```
scatterplot(house_datasales1$floors,house_datasales$price)
```



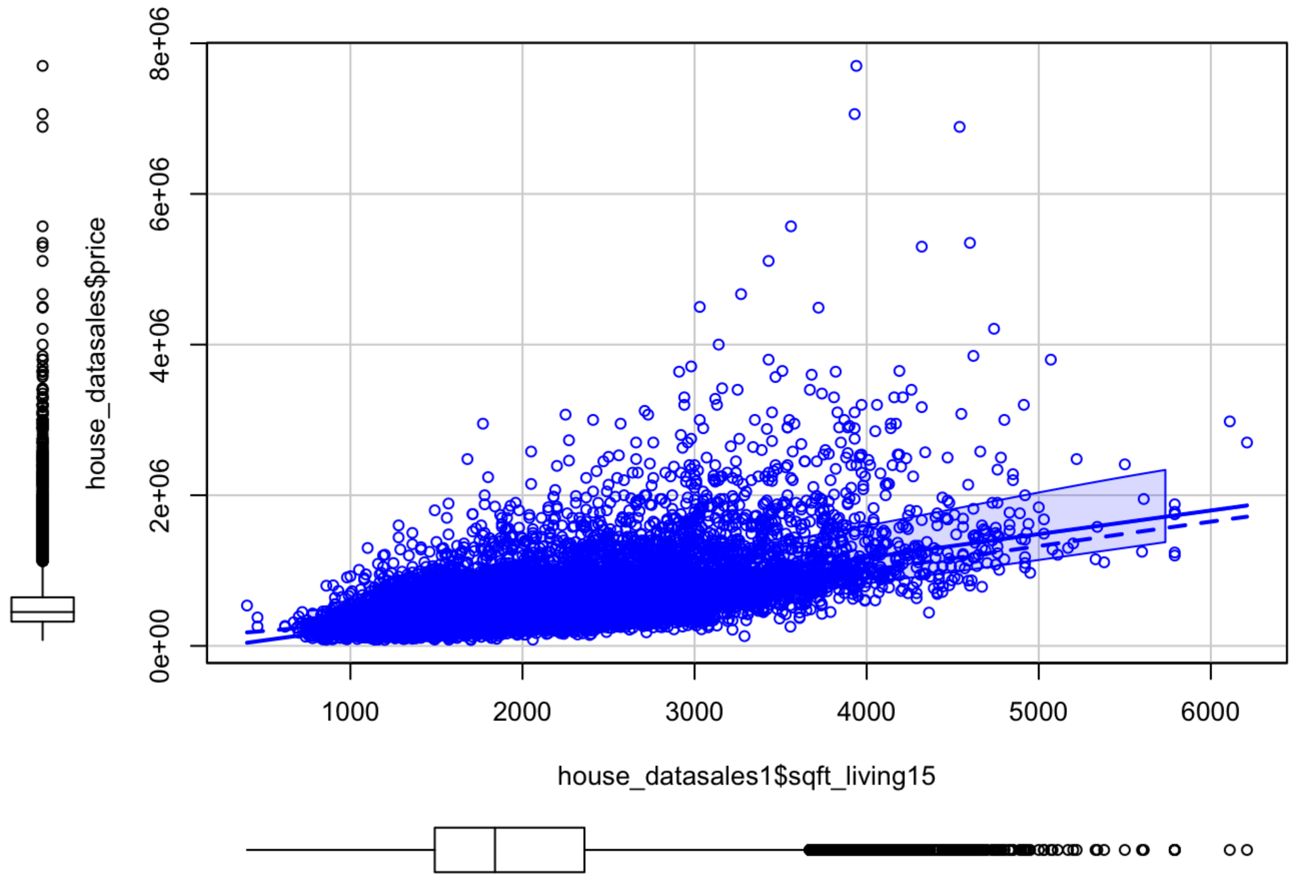

```
scatterplot(house_datasales1$waterfront,house_datasales$price)
```



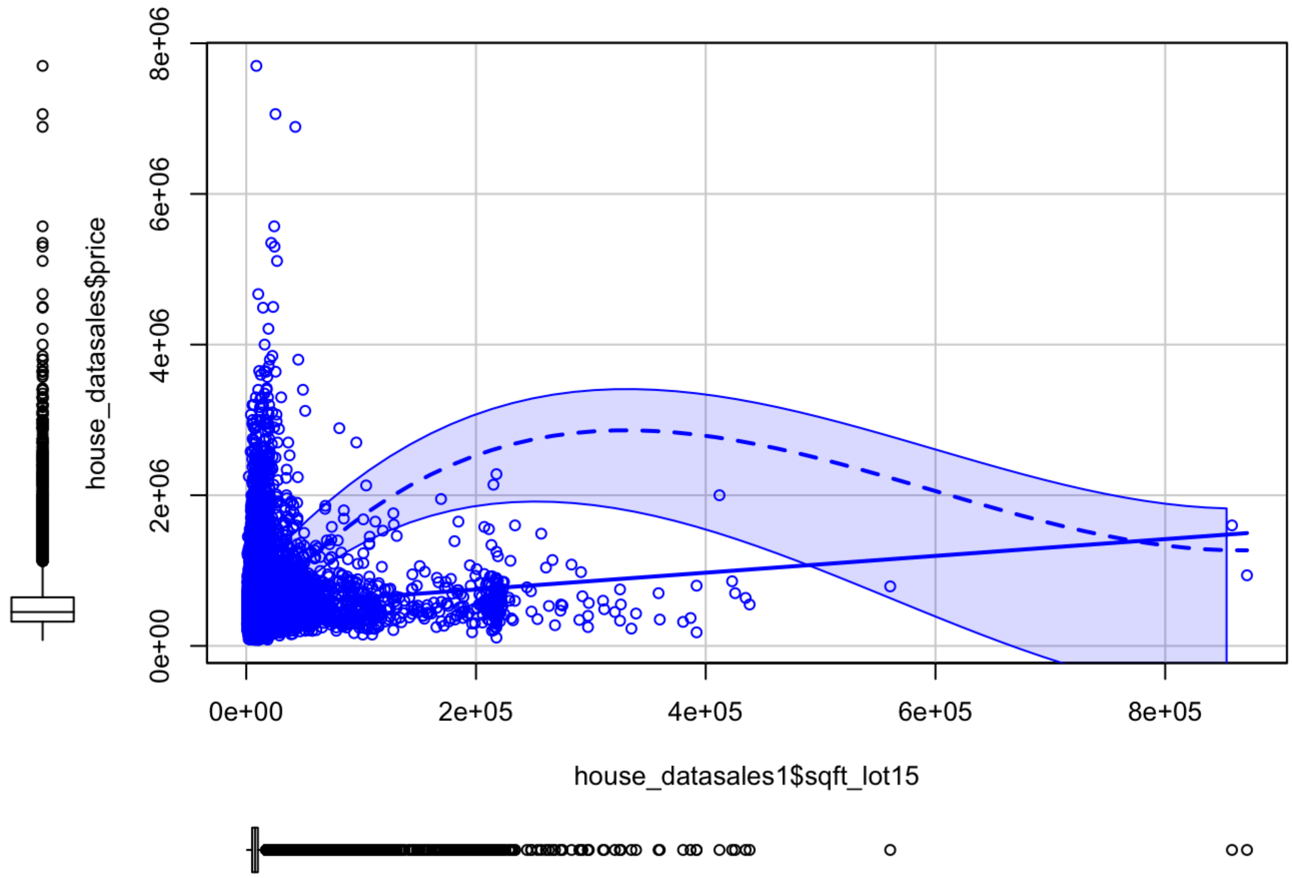

```
scatterplot(house_datasales1$condition,house_datasales$price)
```



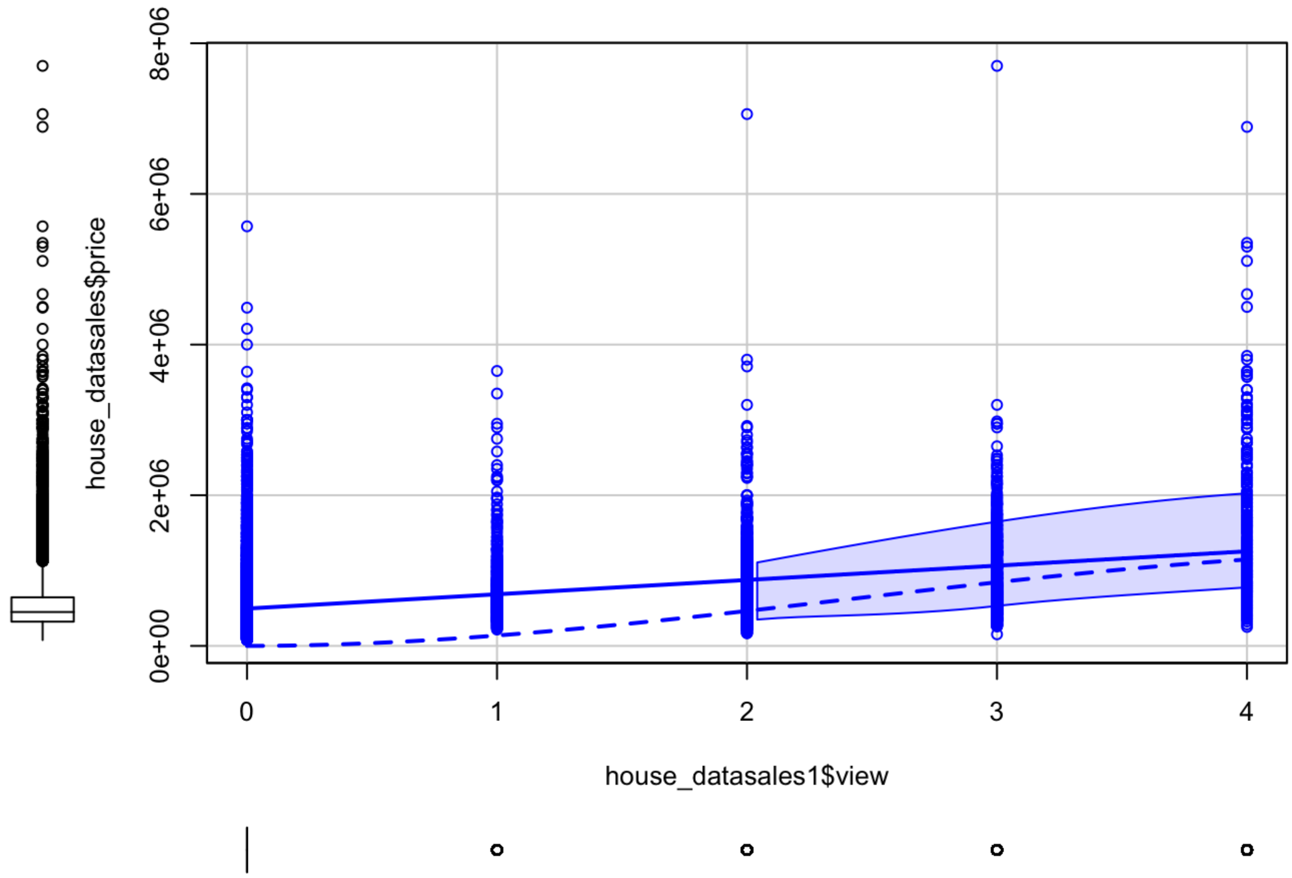

```
scatterplot(house_datasales1$sqft_living15,house_datasales$price)
```



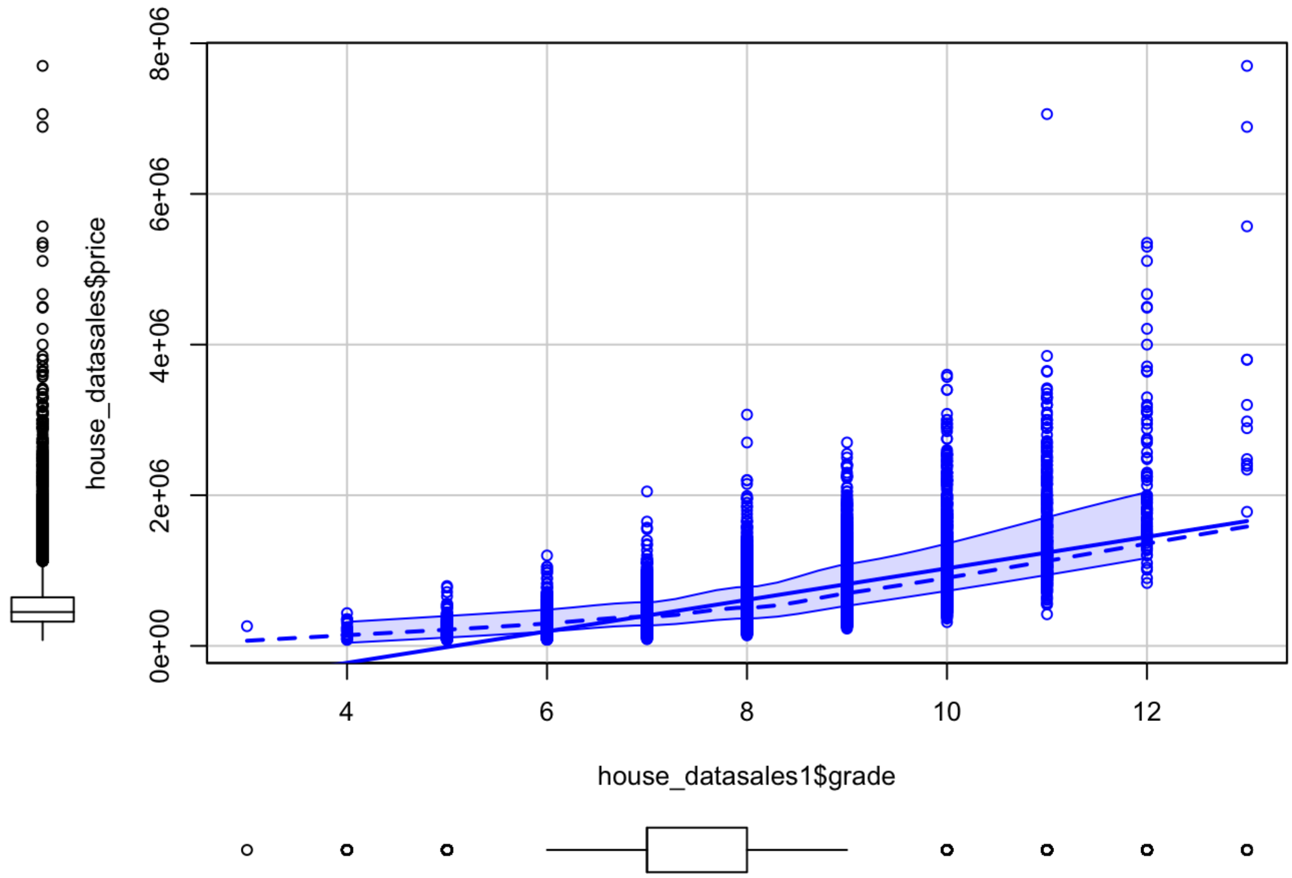

```
scatterplot(house_datasales1$sqft_lot15,house_datasales$price)
```



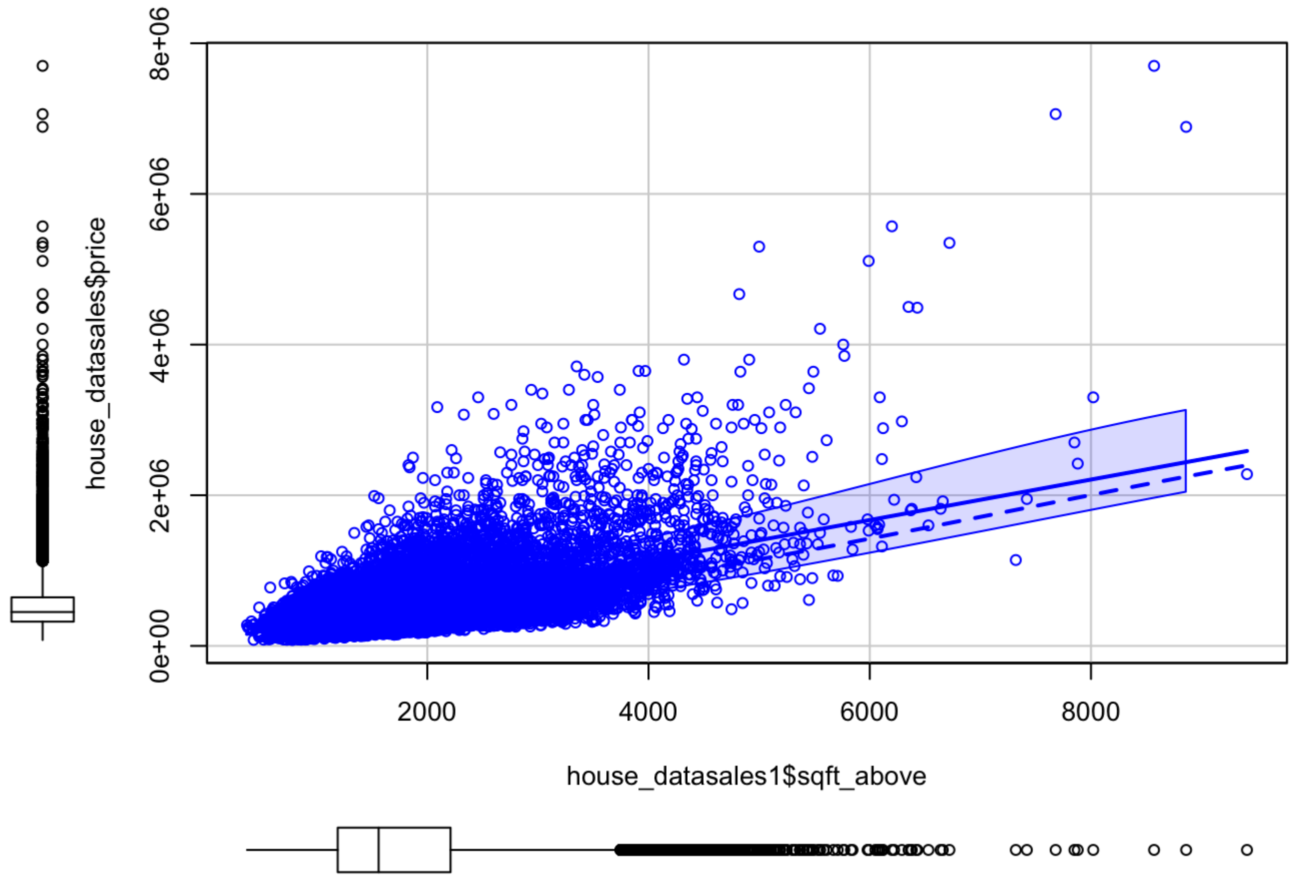

```
scatterplot(house_datasales1$view ,house_datasales$price)
```



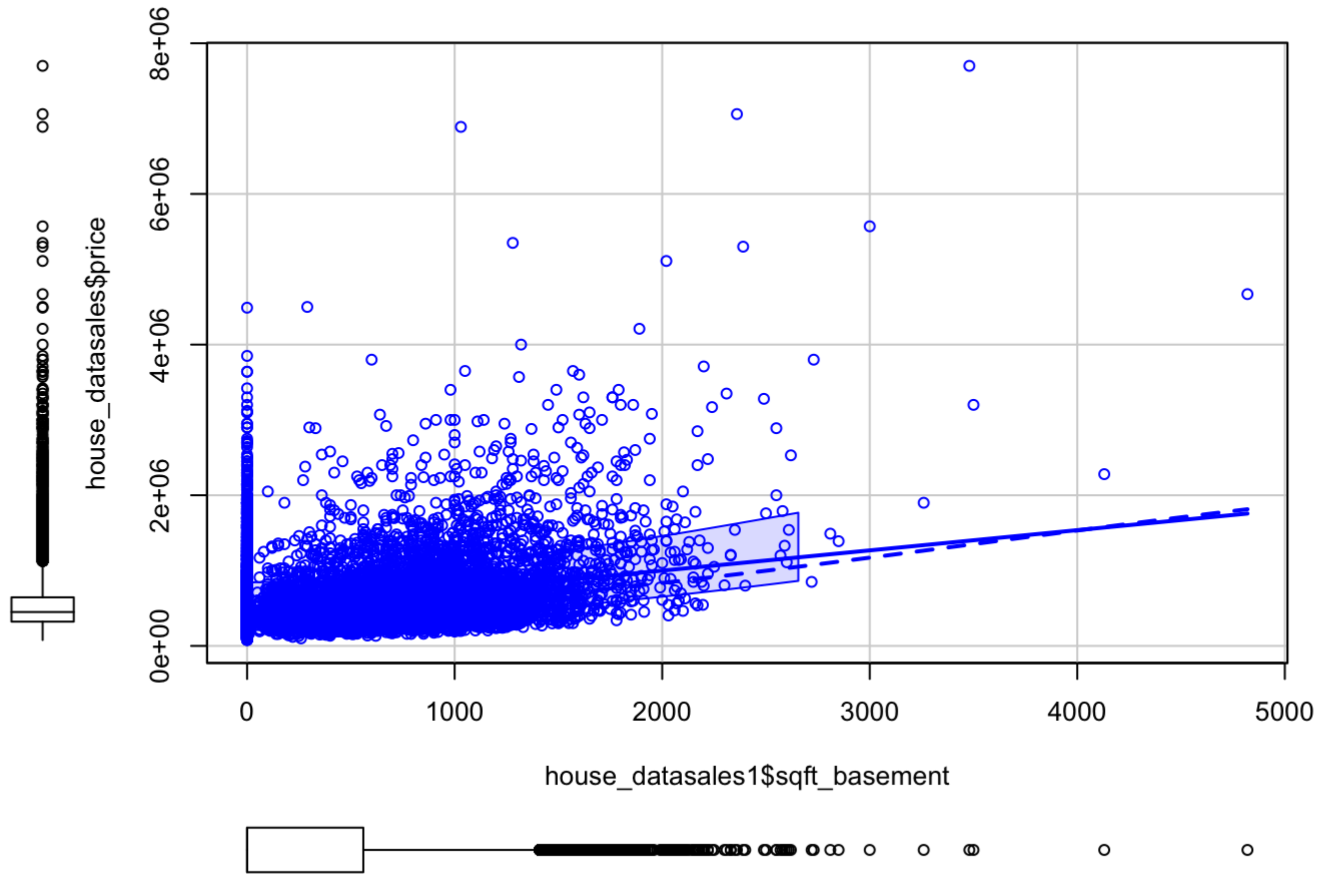

```
scatterplot(house_datasales1$grade,house_datasales$price)
```



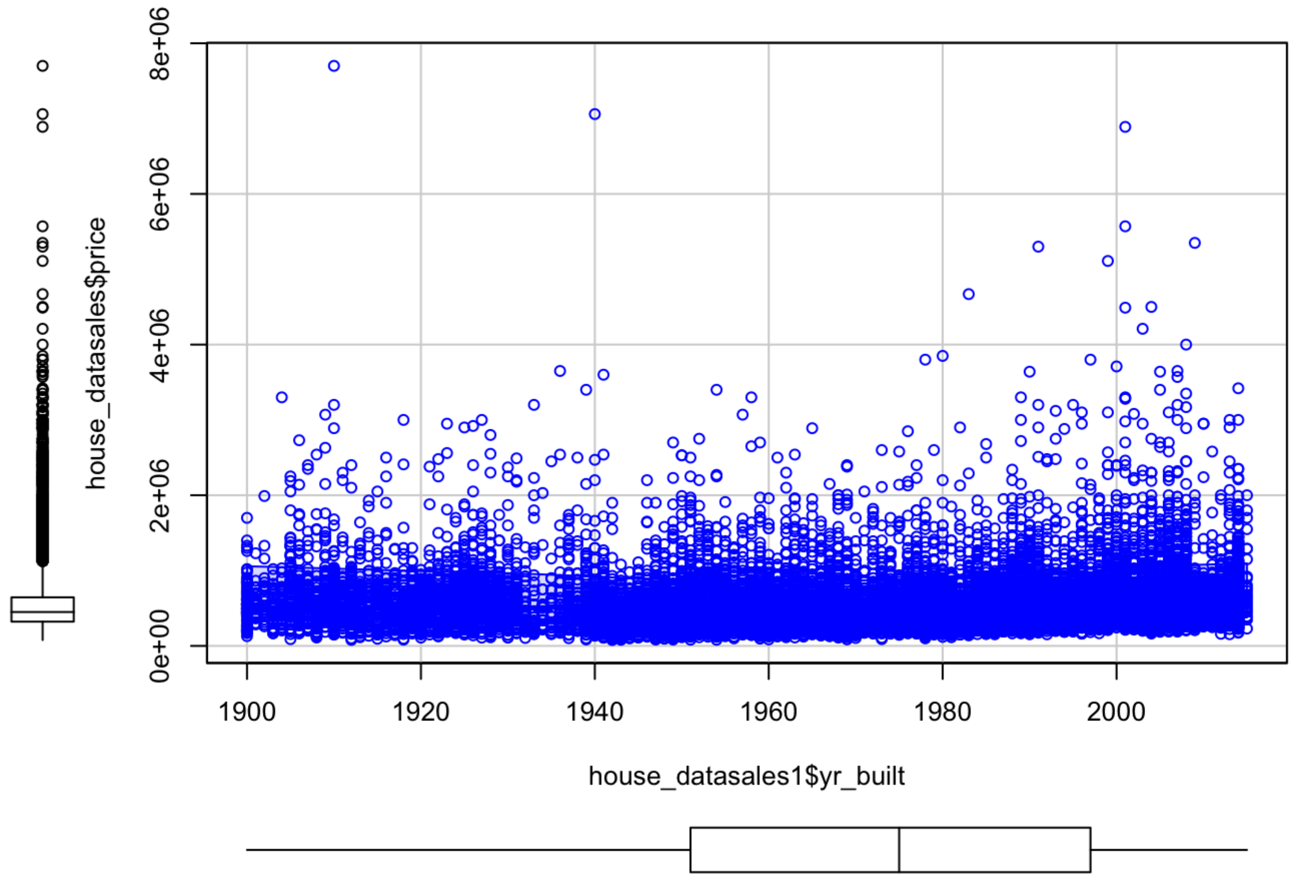

```
scatterplot(house_datasales1$sqft_above,house_datasales$price)
```



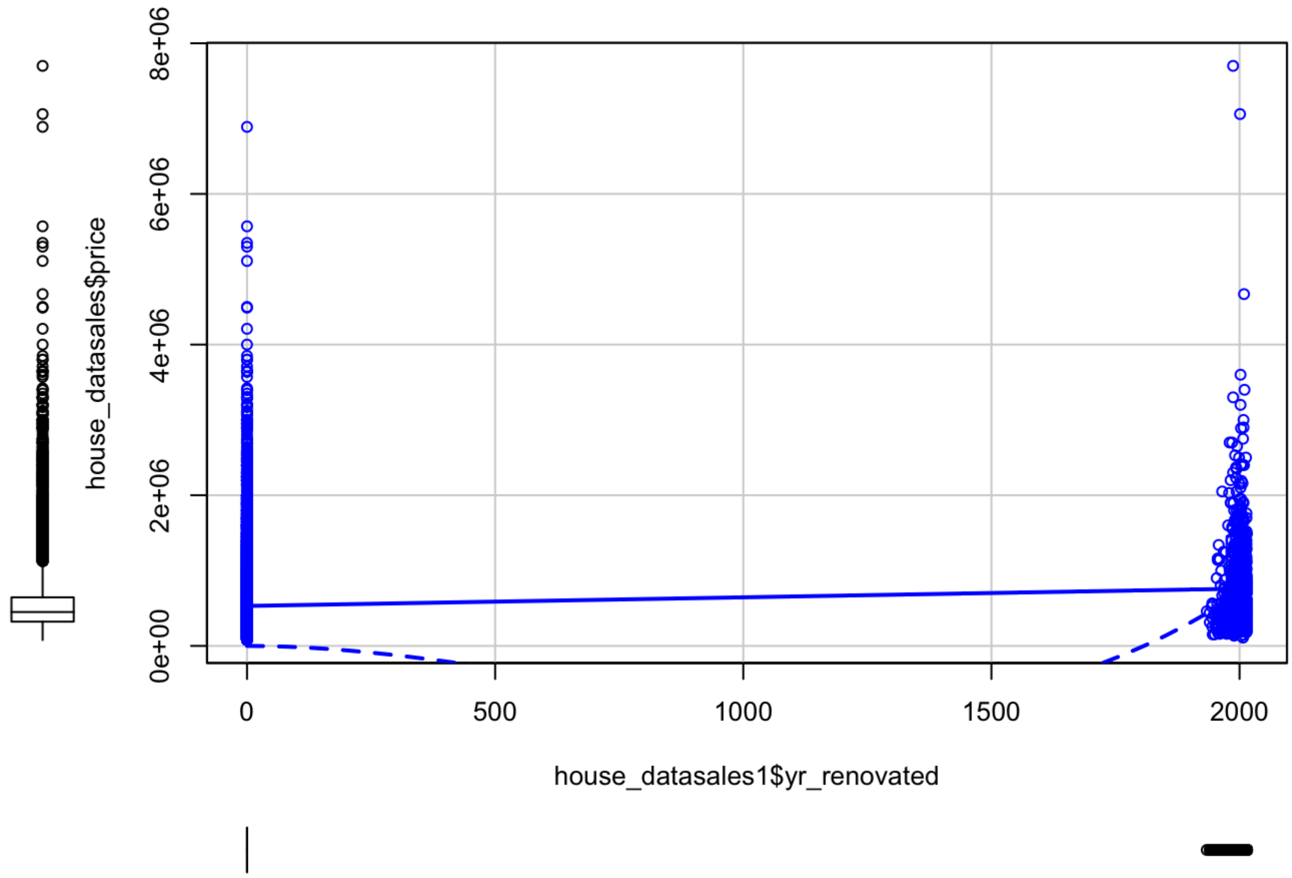

```
scatterplot(house_datasales1$sqft_basement,house_datasales$price)
```



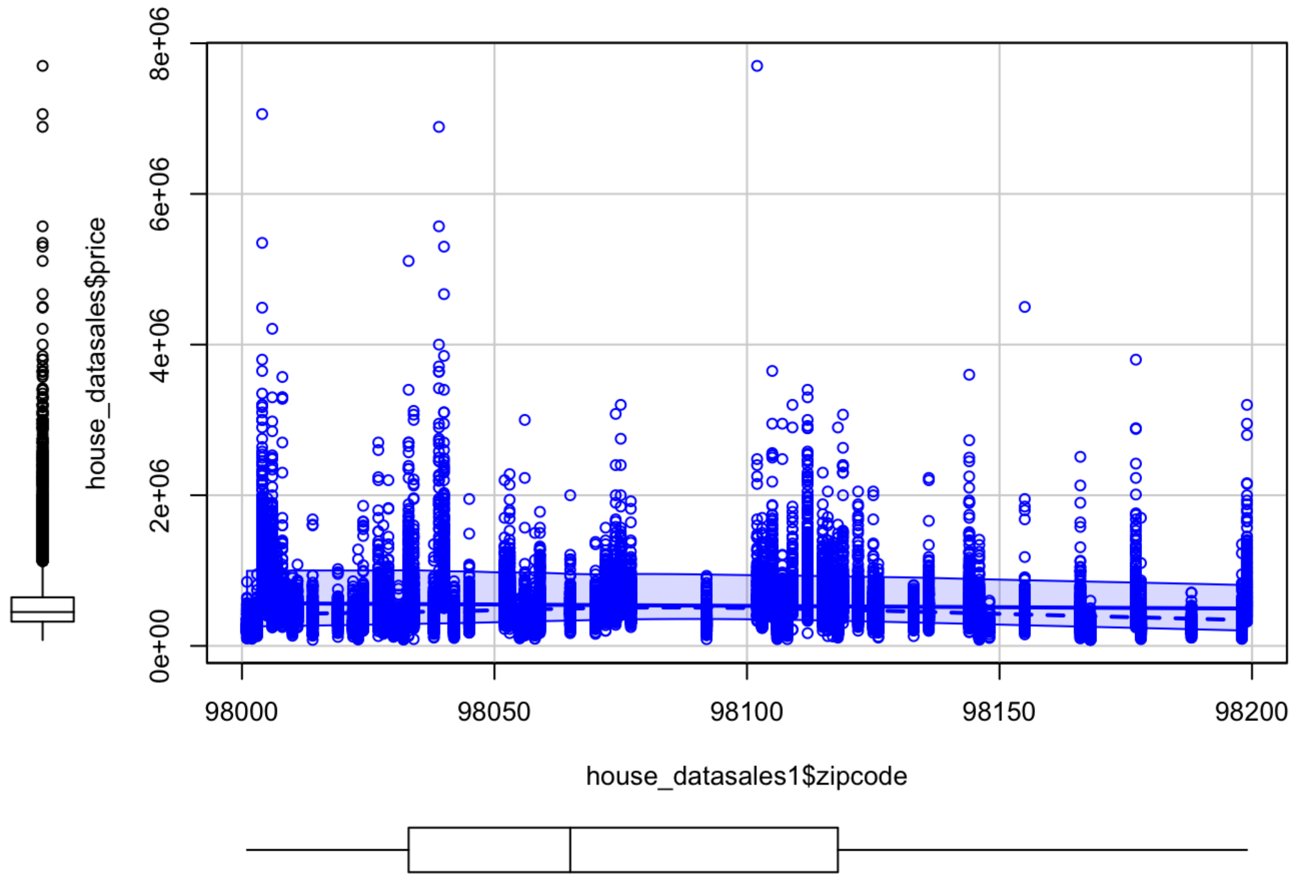

```
scatterplot(house_datasales1$yr_built,house_datasales$price)
```



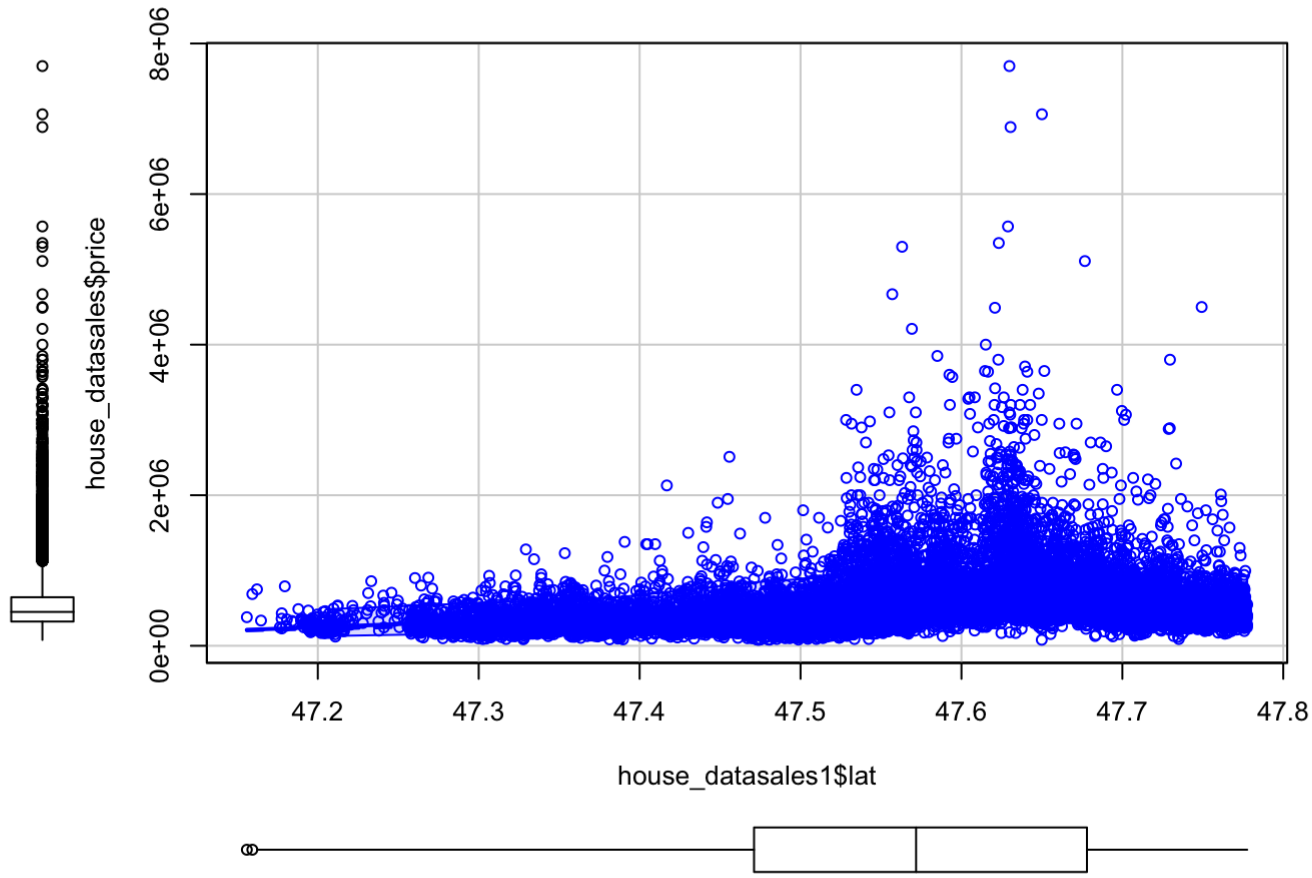

```
scatterplot(house_datasales1$yr_renovated,house_datasales$price)
```



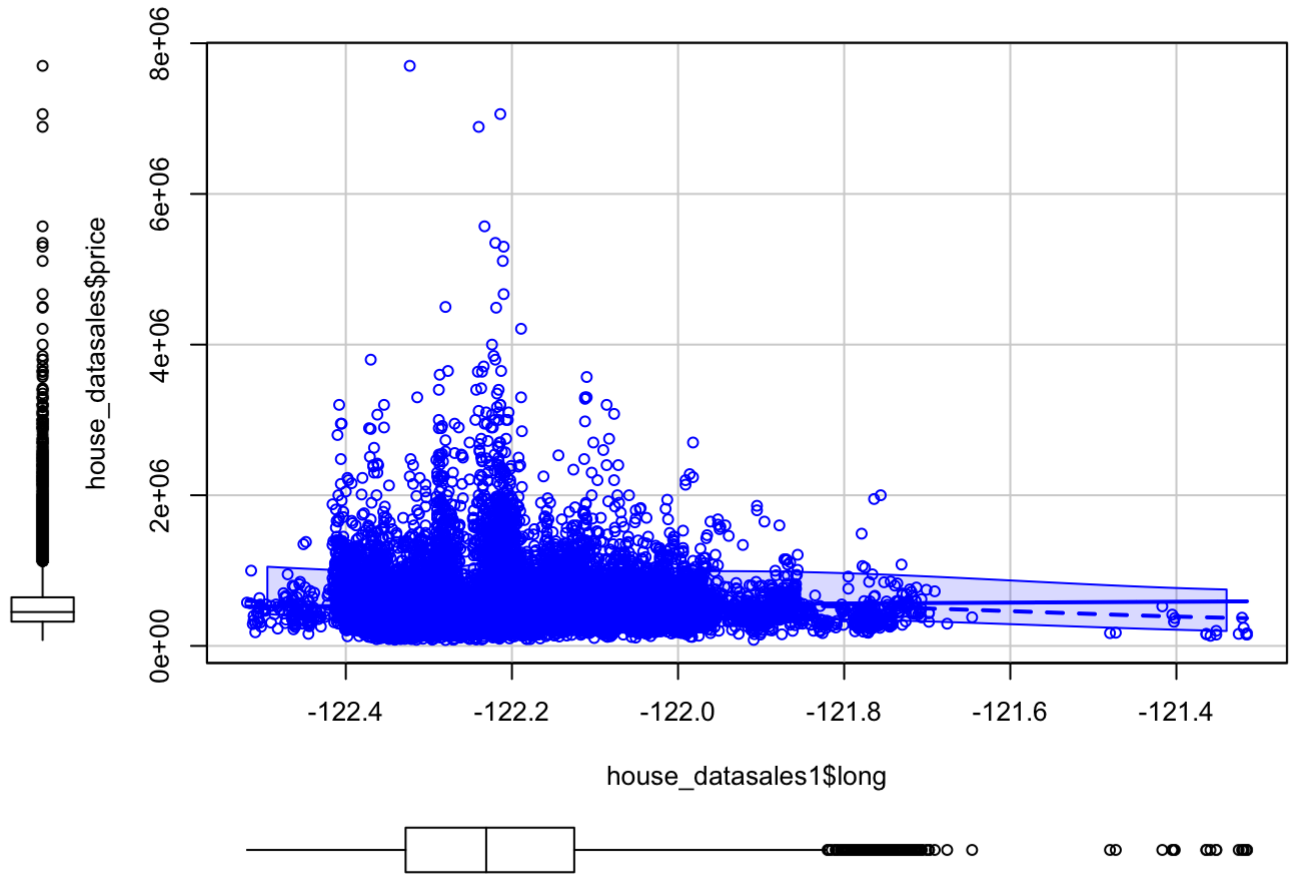

```
scatterplot(house_datasales1$zipcode,house_datasales$price)
```



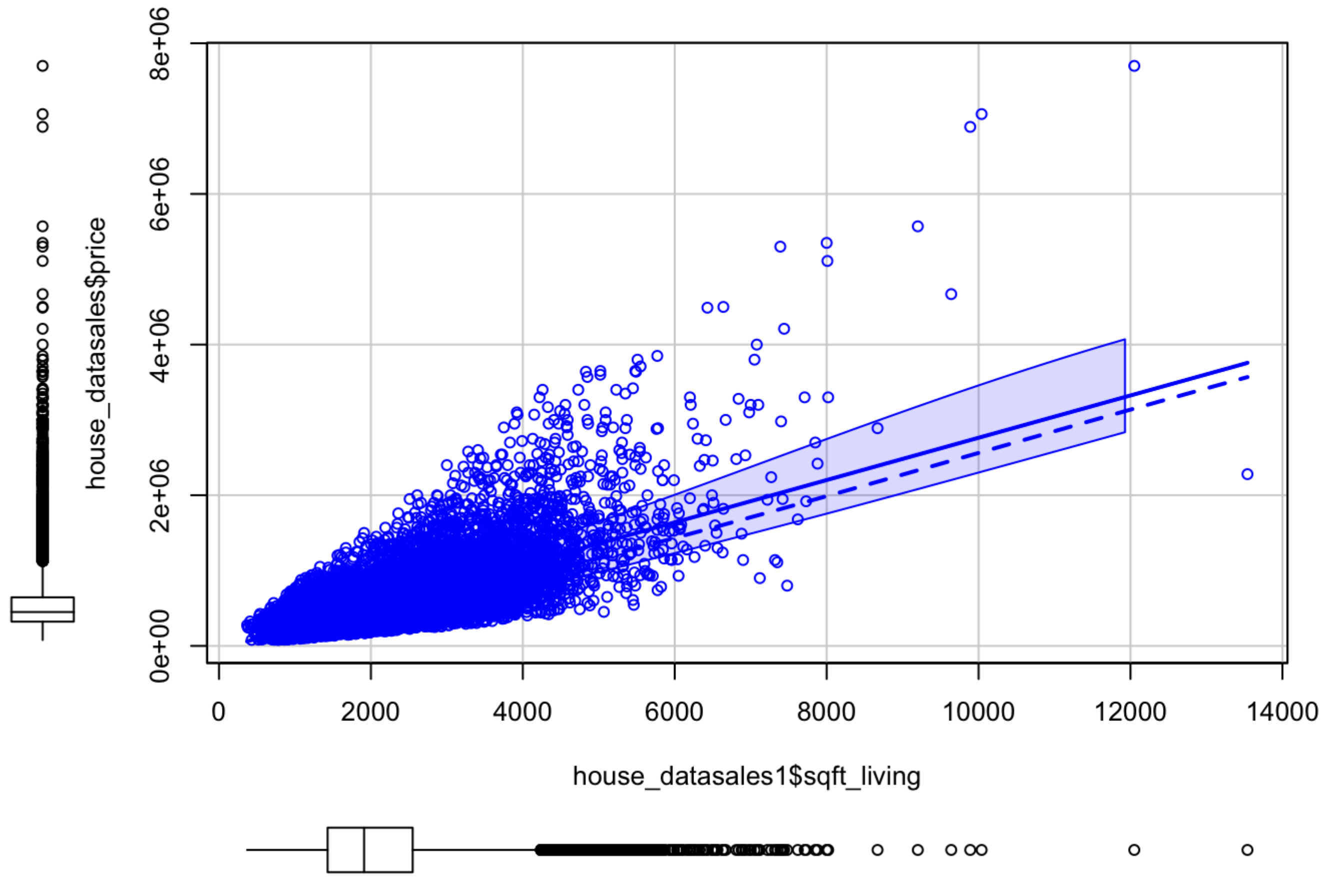

```
scatterplot(house_datasales1$lat,house_datasales$price)
```



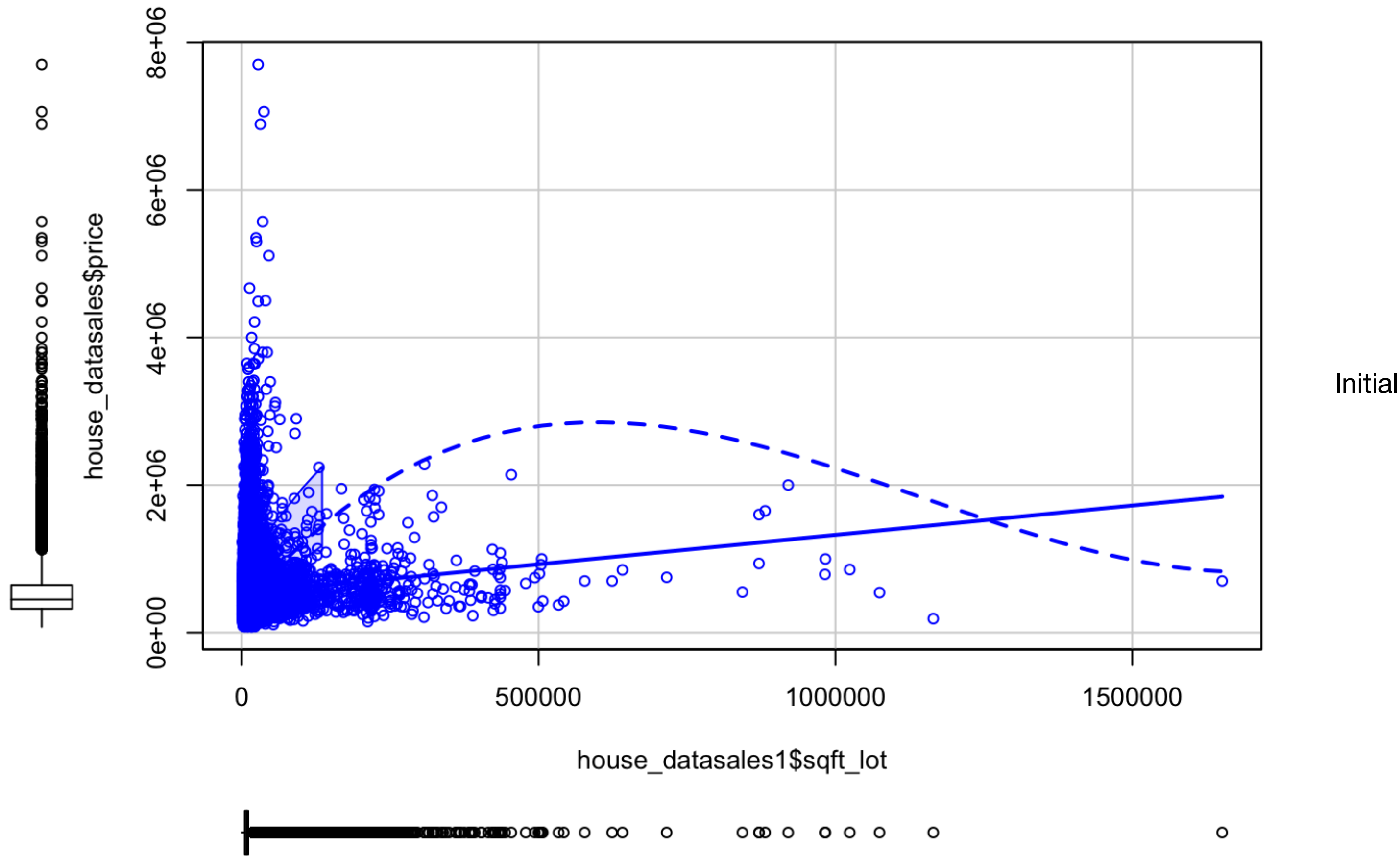

```
scatterplot(house_datasales1$long,house_datasales$price)
```




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

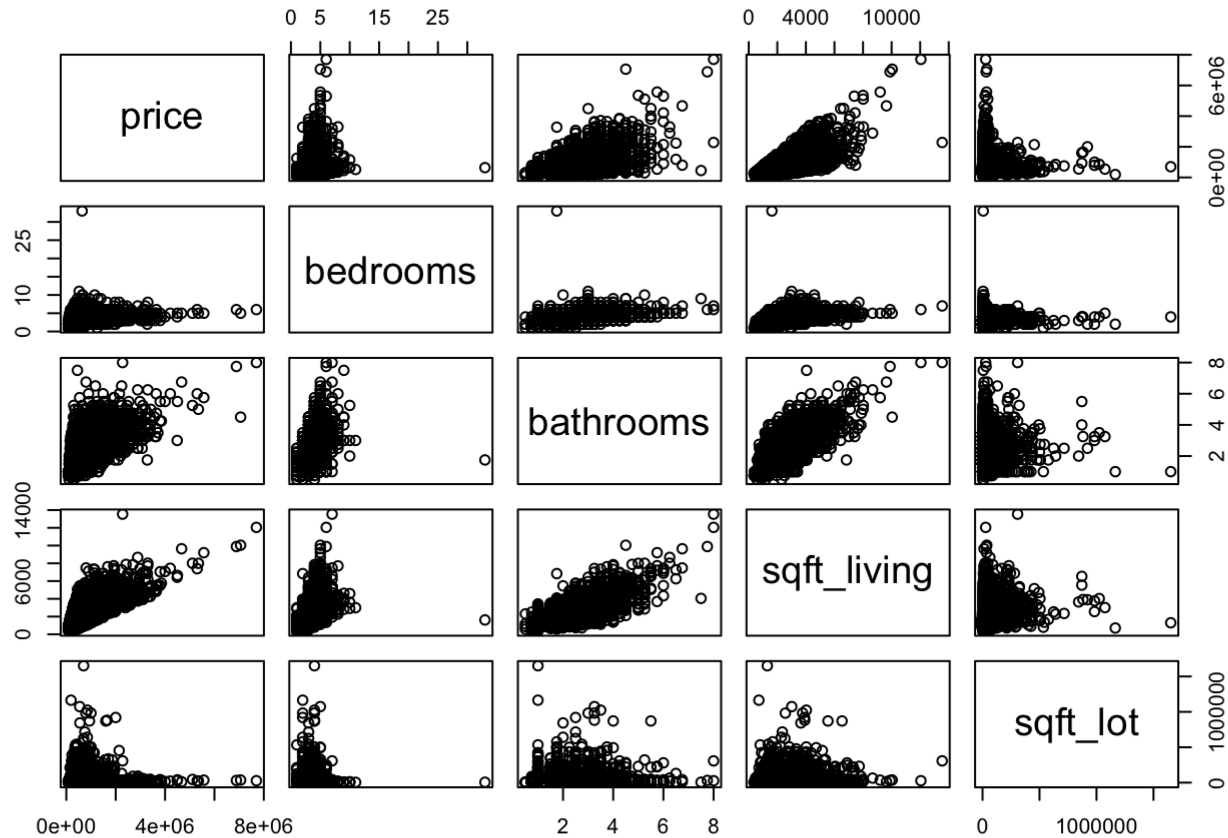



```
scatterplot(house_datasales1$sqft_lot,house_datasales$price)
```

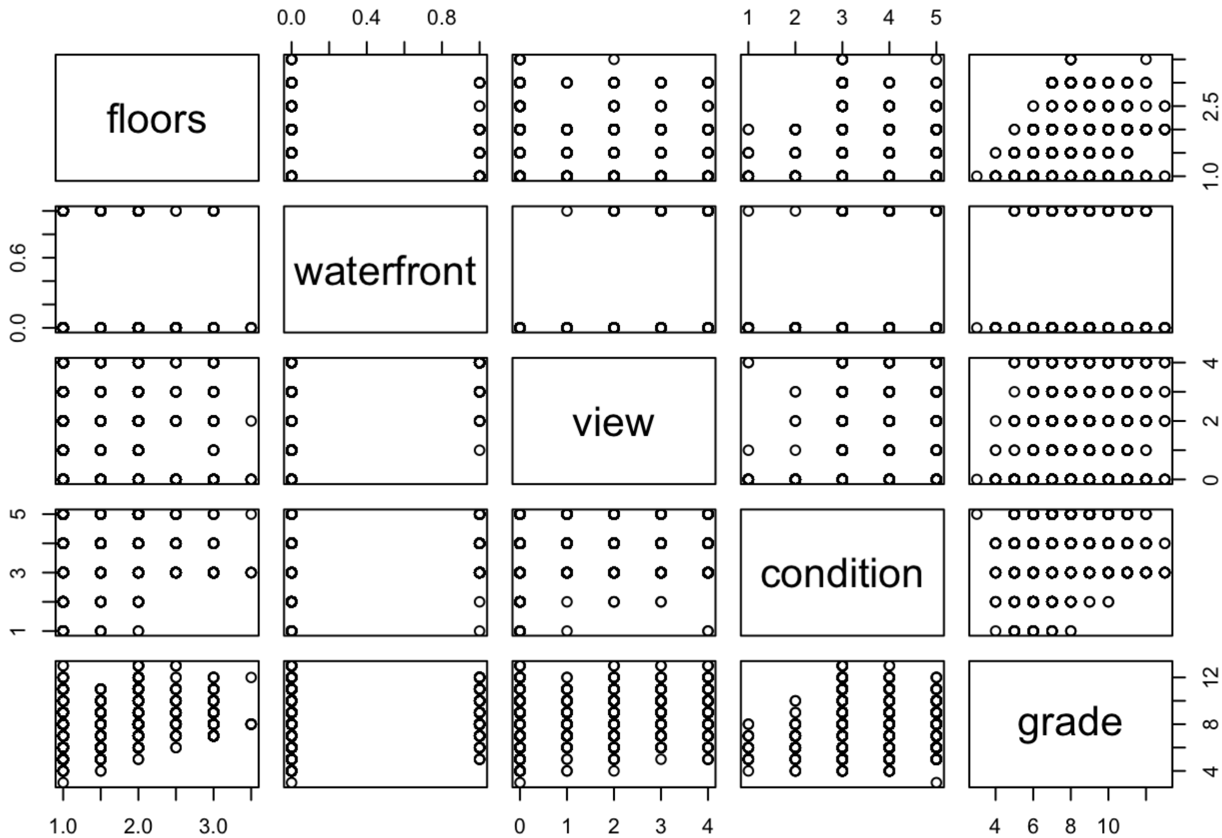


look on the above relation between each variable to the dependent variable - price makes us understand that there are some outliers in the data which we have to take care such that the influence of such points in the creation of the model is less.

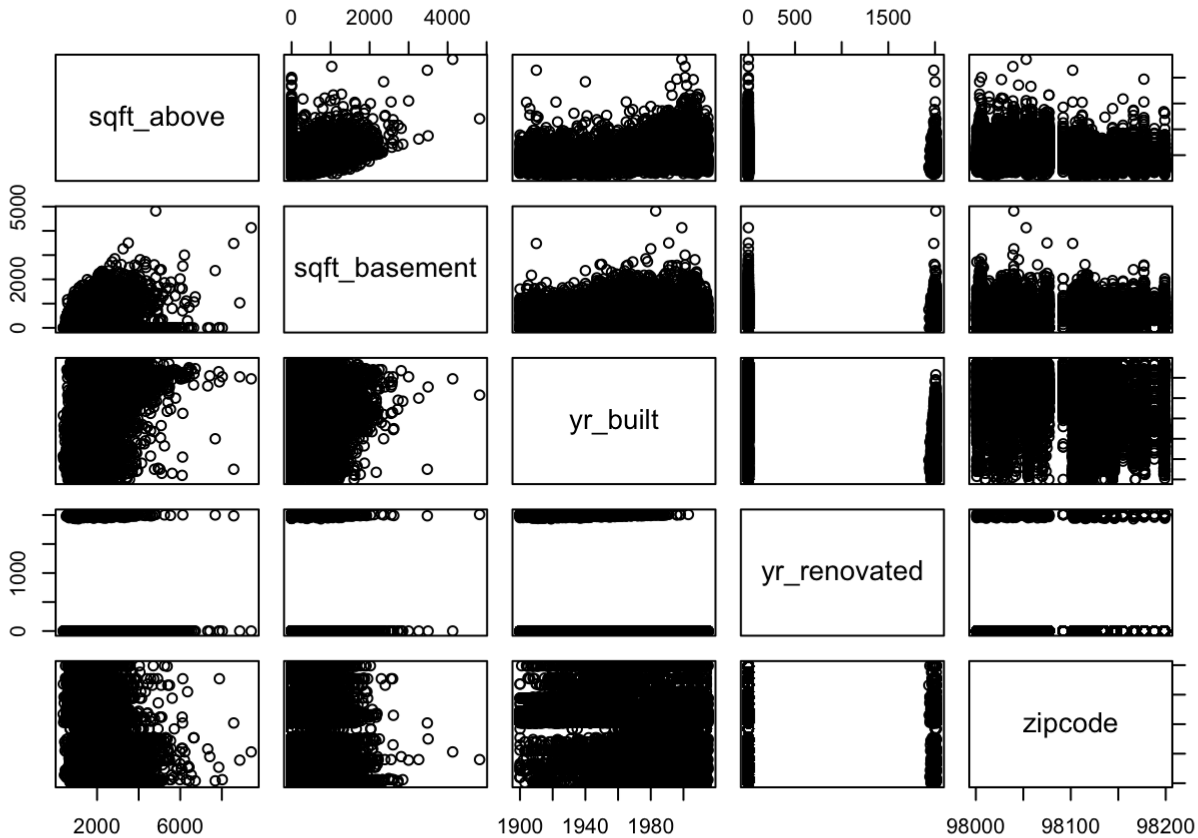
```
plot(house_datasales1[1:5])
```



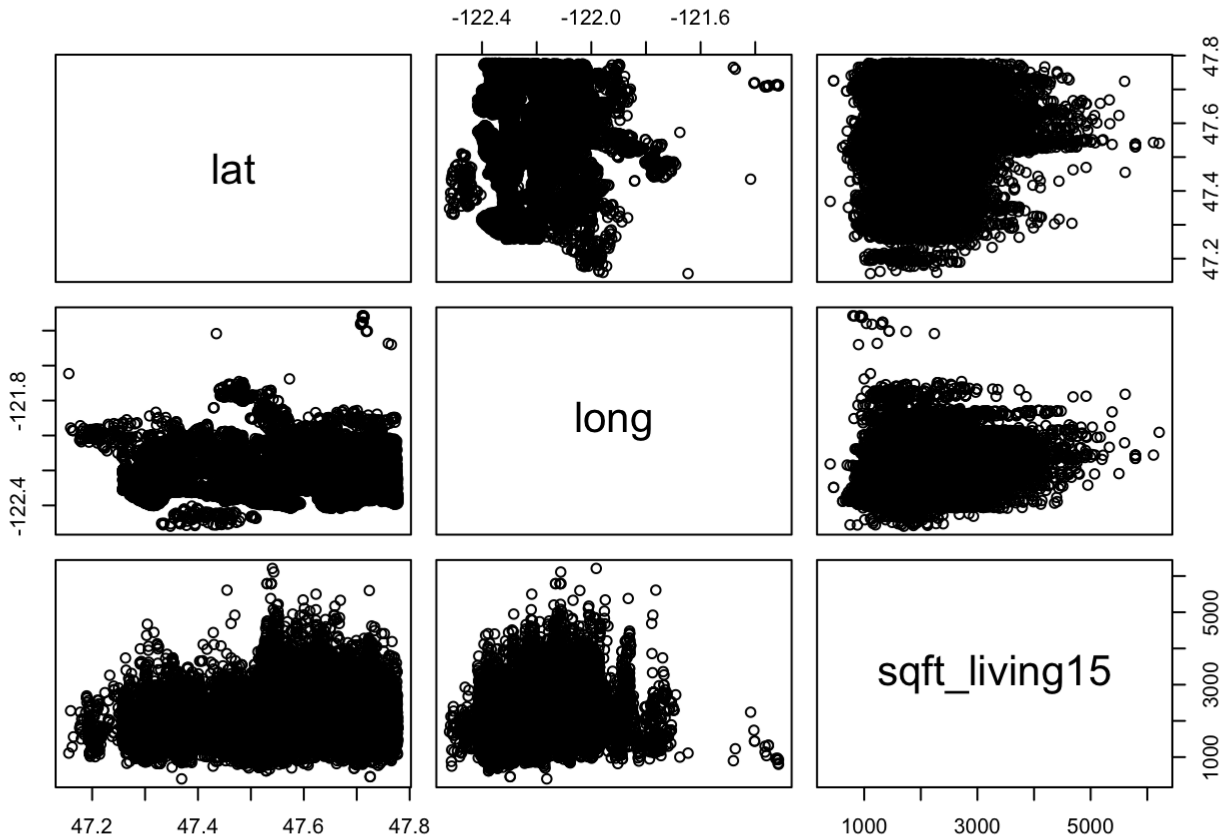

```
plot(house_datasales1[6:10])
```




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




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




```
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
```

```
#View(house_datasales1)
```

Data Preprocessing

Performing Data Sanity Checks before proceeding with analysis

```
house_datasales1$zip <- house_datasales1$zipcode
house_datasales1$zipcode <- NULL
house_datasales1$basement <- house_datasales$sqft_basement
house_datasales1$sqft_basement <- NULL
#View(zipcode_data)
zipcode_data <- zipcode_data[ -c(2:3,5,7:18) ]
#View(zipcode_data)
## Converting categorical values to numeric
house_datasales1$basement = ifelse(house_datasales1$basement>0,"1","0")
#View(house_datasales1)
house_datasales1$renovation = ifelse(house_datasales1$"yr_renovated">0,"1","0")
house_datasales1$yr_renovated <- NULL
```

Checking missing values and duplicate values in the data:

```
## missing values check
print(sum(is.na(house_datasales1)))
```

```
## [1] 0
```

```
print(sum(is.na(zipcode_data)))
```

```
## [1] 0
```

```
## duplicate rows check
zipcode_data %>% distinct(zip, .keep_all= TRUE)
```

```
## # A tibble: 33,121 × 3
##    zip   city        state_name
##    <chr> <chr>       <chr>
##  1 00601 Adjuntas    Puerto Rico
##  2 00602 Aguada      Puerto Rico
##  3 00603 Aguadilla   Puerto Rico
##  4 00606 Maricao     Puerto Rico
##  5 00610 Anasco      Puerto Rico
##  6 00612 Arecibo     Puerto Rico
##  7 00616 Bajadero    Puerto Rico
##  8 00617 Barceloneta Puerto Rico
##  9 00622 Boqueron    Puerto Rico
## 10 00623 Cabo Rojo   Puerto Rico
## # … with 33,111 more rows
```

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

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
```

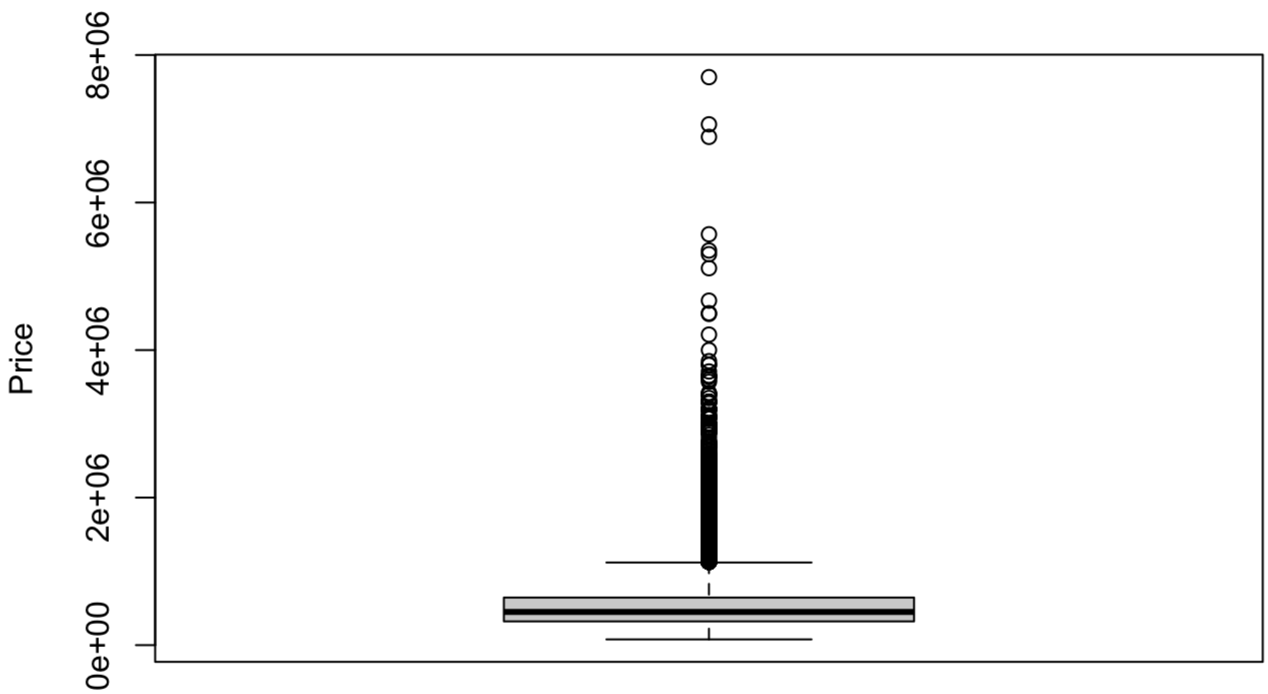
```
table(final_merged_data$state_name)
```

```
## 
## Washington 
##      21597
```

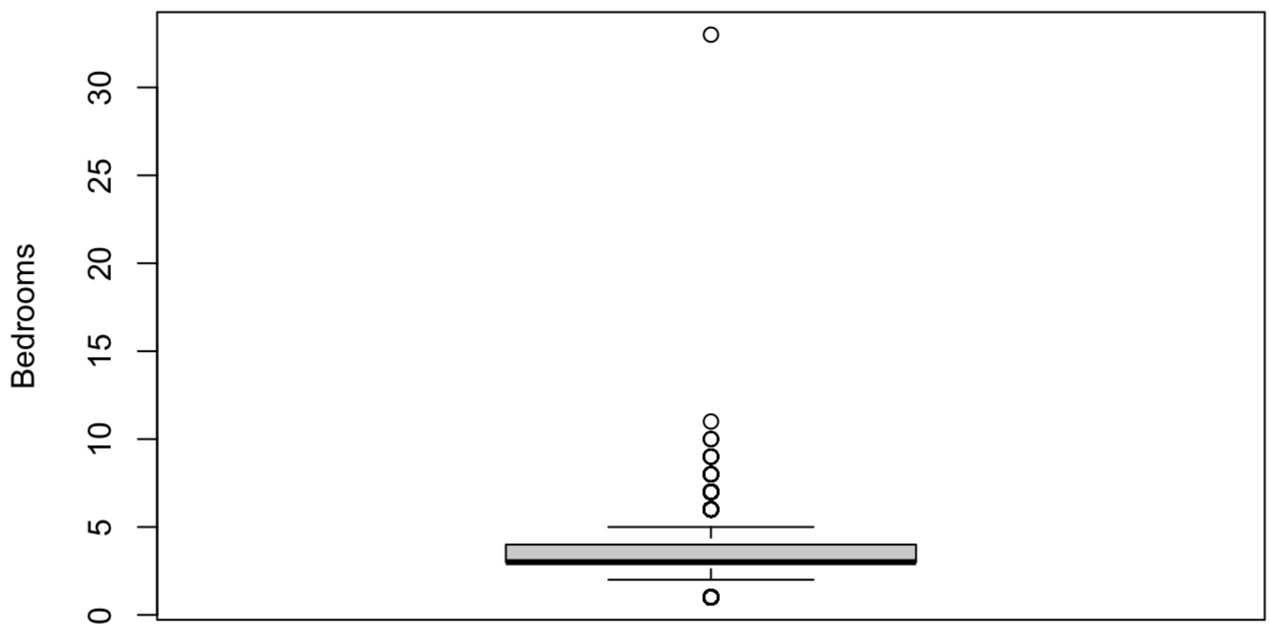
```
final_merged_data$state_name <- NULL
```

Detecting ol if any

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




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



Exploratory Data Analysis

```
#View(final_merged_data)
summary(final_merged_data)
```

```
##       zip            price            bedrooms        bathrooms
##  Min.   :98001   Min.   :  78000   Min.   : 1.000   Min.   :0.500
##  1st Qu.:98033   1st Qu.: 322000   1st Qu.: 3.000   1st Qu.:1.750
##  Median :98065   Median : 450000   Median : 3.000   Median :2.250
##  Mean   :98078   Mean   : 540297   Mean   : 3.373   Mean   :2.116
##  3rd Qu.:98118   3rd Qu.: 645000   3rd Qu.: 4.000   3rd Qu.:2.500
##  Max.   :98199   Max.   :7700000   Max.   :33.000   Max.   :8.000
##   sqft_living       sqft_lot           floors        waterfront
##  Min.   :  370   Min.   :    520   Min.   :1.000   Min.   :0.000000
##  1st Qu.: 1430   1st Qu.:   5040   1st Qu.:1.000   1st Qu.:0.000000
##  Median : 1910   Median :   7618   Median :1.500   Median :0.000000
##  Mean   : 2080   Mean   :  15099   Mean   :1.494   Mean   :0.007547
##  3rd Qu.: 2550   3rd Qu.:  10685   3rd Qu.:2.000   3rd Qu.:0.000000
##  Max.   :13540   Max.   :1651359   Max.   :3.500   Max.   :1.000000
##       view          condition        grade          sqft_above      yr_built
##  Min.   :0.0000   Min.   :1.00   Min.   : 3.000   Min.   : 370   Min.   :1900
##  1st Qu.:0.0000   1st Qu.:3.00   1st Qu.: 7.000   1st Qu.:1190   1st Qu.:1951
##  Median :0.0000   Median :3.00   Median : 7.000   Median :1560   Median :1975
##  Mean   :0.2343   Mean   :3.41   Mean   : 7.658   Mean   :1789   Mean   :1971
##  3rd Qu.:0.0000   3rd Qu.:4.00   3rd Qu.: 8.000   3rd Qu.:2210   3rd Qu.:1997
##  Max.   :4.0000   Max.   :5.00   Max.   :13.000   Max.   :9410   Max.   :2015
##       lat             long        sqft_living15    sqft_lot15
##  Min.   :47.16   Min.   :-122.5   Min.   : 399   Min.   :   651
##  1st Qu.:47.47   1st Qu.:-122.3   1st Qu.:1490   1st Qu.:  5100
##  Median :47.57   Median :-122.2   Median :1840   Median :  7620
##  Mean   :47.56   Mean   :-122.2   Mean   :1987   Mean   : 12758
##  3rd Qu.:47.68   3rd Qu.:-122.1   3rd Qu.:2360   3rd Qu.: 10083
##  Max.   :47.78   Max.   :-121.3   Max.   :6210   Max.   :871200
##    basement          renovation            city
##  Length:21597       Length:21597       Length:21597
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
```

```
## 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
```

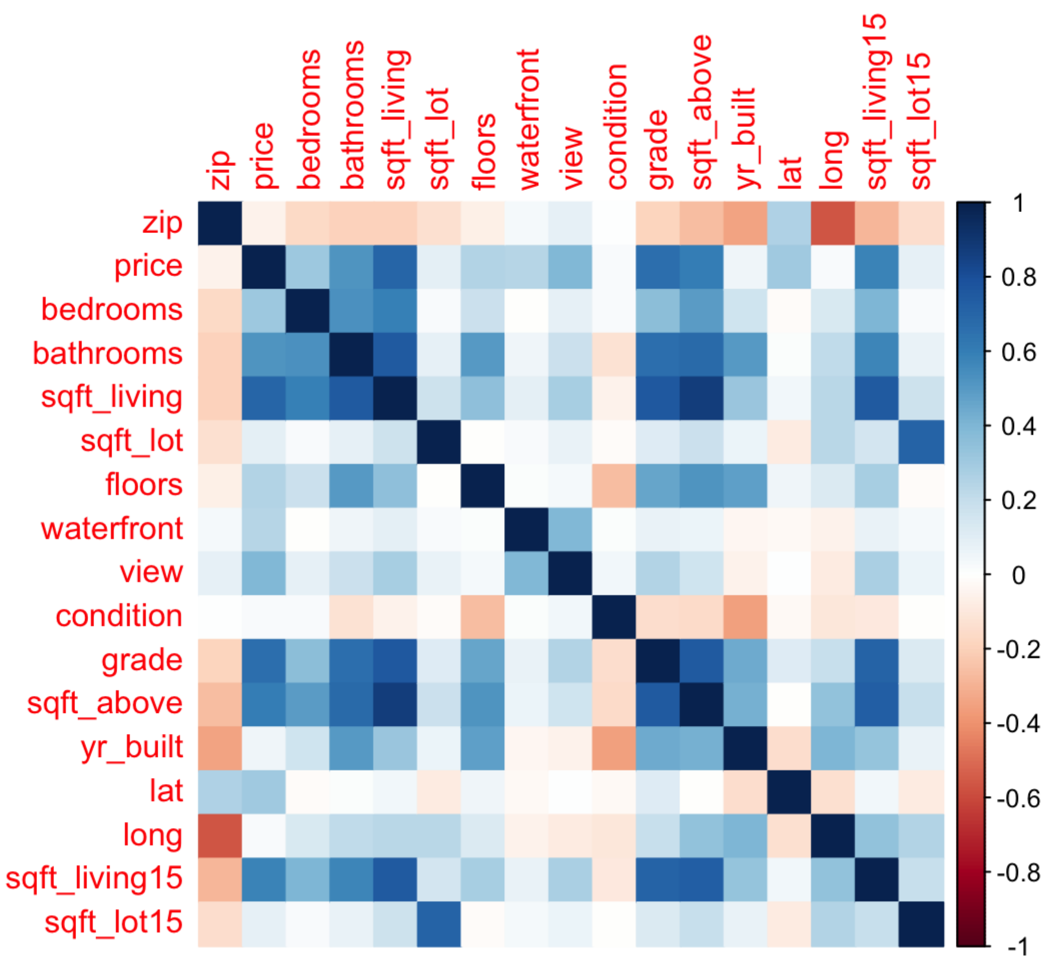
```
#View(train)
dim(test)
```

```
## [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")
```

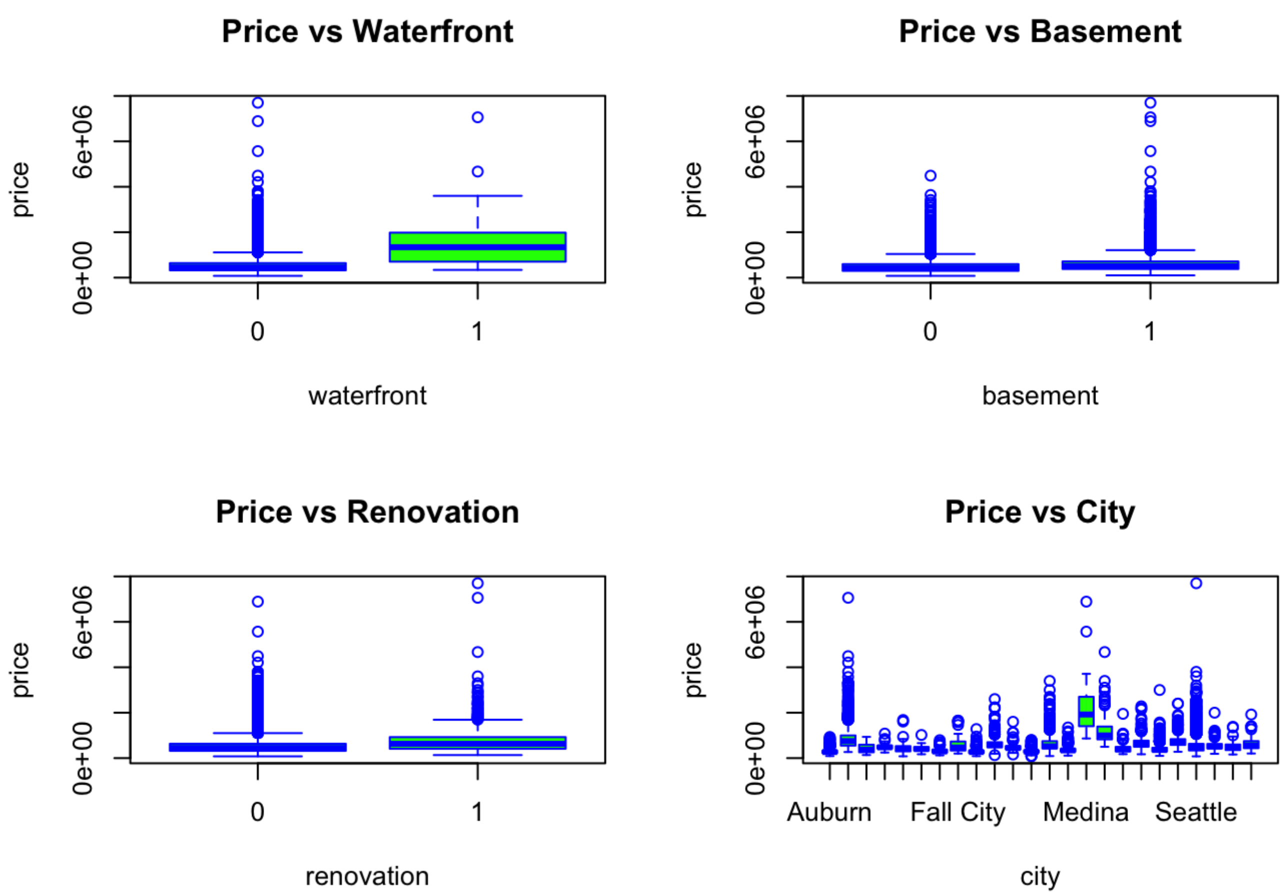


Scatter plots for determining the positive-correlated variables

```
plot1=ggplot(data = train, aes(x = bedrooms, y = price)) +
  geom_jitter() +  geom_smooth(method = "lm", se = FALSE)+labs(title="Scatter plot of Pr
ice vs Bedrooms", x="Bedrooms",y="Price")
plot2=ggplot(data = train, aes(x = bathrooms, y = price)) +
  geom_jitter() +  geom_smooth(method = "lm", se = FALSE)+labs(title="Scatter plot of Pr
ice vs Bathrooms", x="Bathrooms",y="Price")
plot3=ggplot(data = train, aes(x = floors, y = price)) +
  geom_jitter() +  geom_smooth(method = "lm", se = FALSE)+labs(title="Scatter plot of Pr
ice vs Floors", x="Floors",y="Price")
```

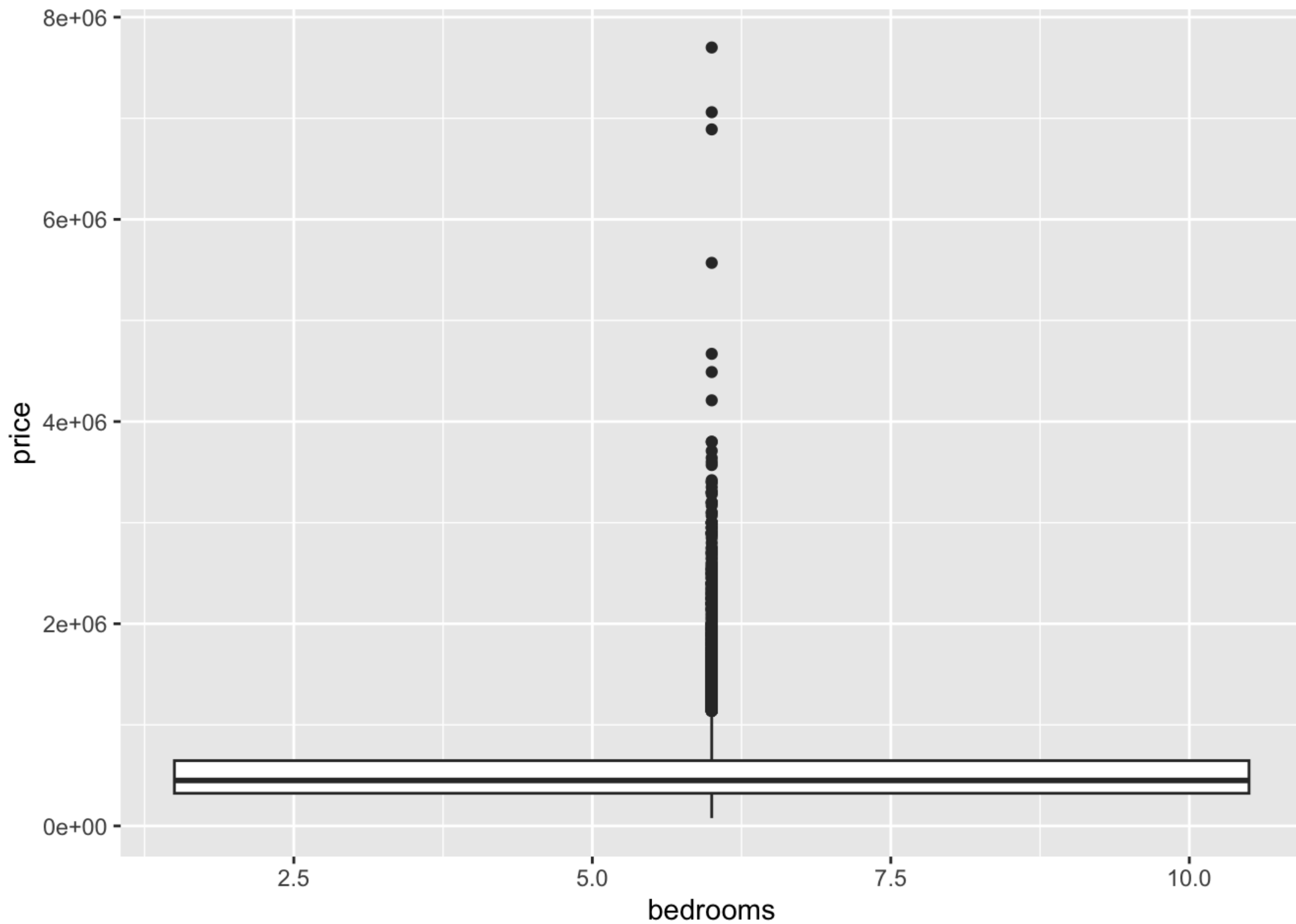
To get clear view of relationships, we plot the boxplots

```
par(mfrow=c(2, 2))
boxplot(price~waterfront,data=train,main="Price vs Waterfront", xlab="waterfront",ylab=
"price",col="green",border="blue")
boxplot(price ~ basement,data=train,main="Price vs Basement", xlab="basement",ylab="pric
e",col="green",border="blue")
boxplot(price~renovation,data=train,main="Price vs Renovation", xlab="renovation",ylab=
"price",col="green",border="blue")
boxplot(price~city,data=train,main="Price vs City", xlab="city",ylab="price",col="green"
,border="blue")
```



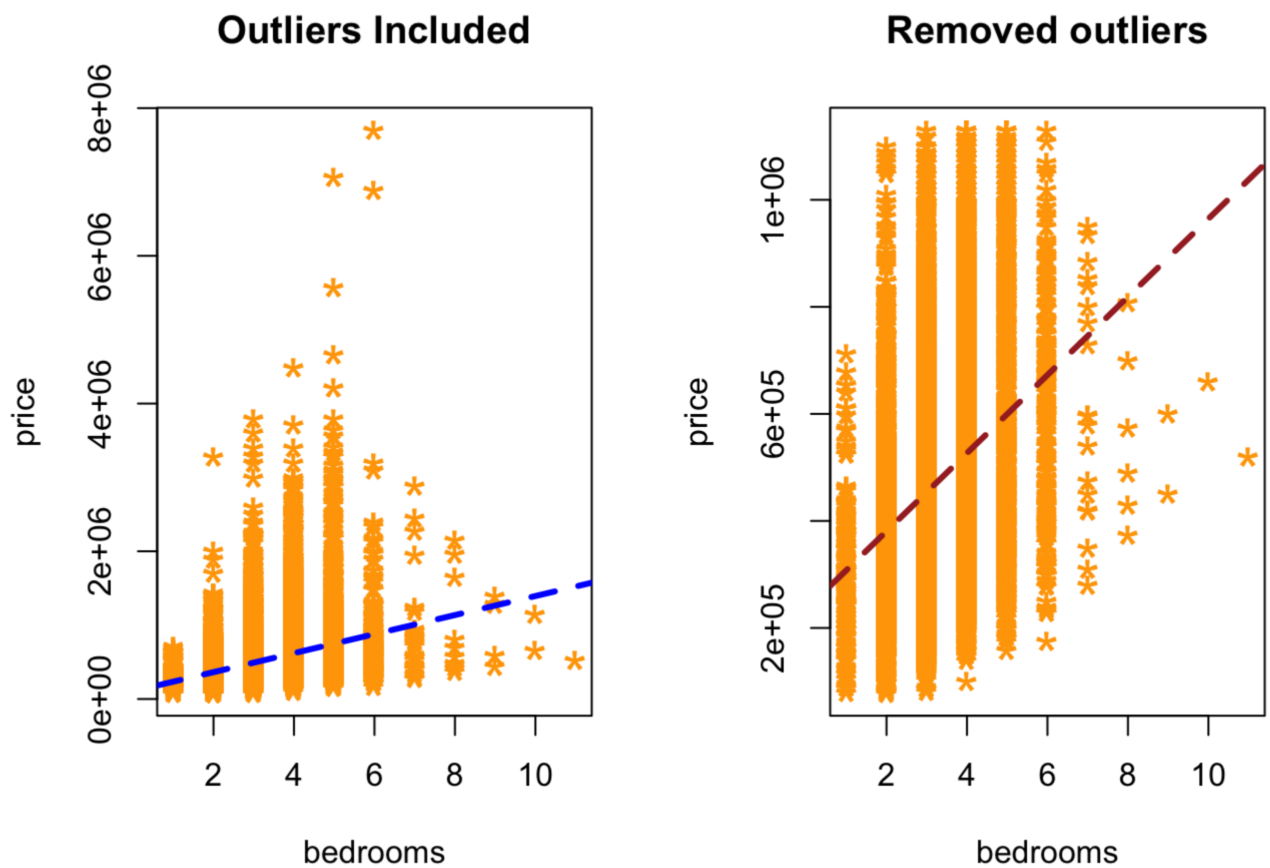
```
ggplot(data=train)+geom_boxplot(aes(x=bedrooms,y=price))
```

```
## Warning: Continuous x aesthetic
## ℹ did you forget `aes(group = ...)`?
```



Plotting data with and without outliers to understand the change in the slope.

```
ol=boxplot(train$price,plot=FALSE)$out
ol_data=train[which(train$price %in% ol),]
train1= train[-which(train$price %in% ol),]
par(mfrow=c(1, 2))
plot(train$bedrooms, train$price, main="Outliers Included", xlab="bedrooms", ylab="pric
e", pch="*", col="orange", cex=2)
abline(lm(price ~ bedrooms, data=train), col="blue", lwd=3, lty=2)
plot(train1$bedrooms, train1$price, main="Removed outliers", xlab="bedrooms", ylab="pric
e", pch="*", col="orange", cex=2)
abline(lm(price ~bedrooms, data=train1), col="brown", lwd=3, lty=2)
```



Analaysis of variance (ANOVA)

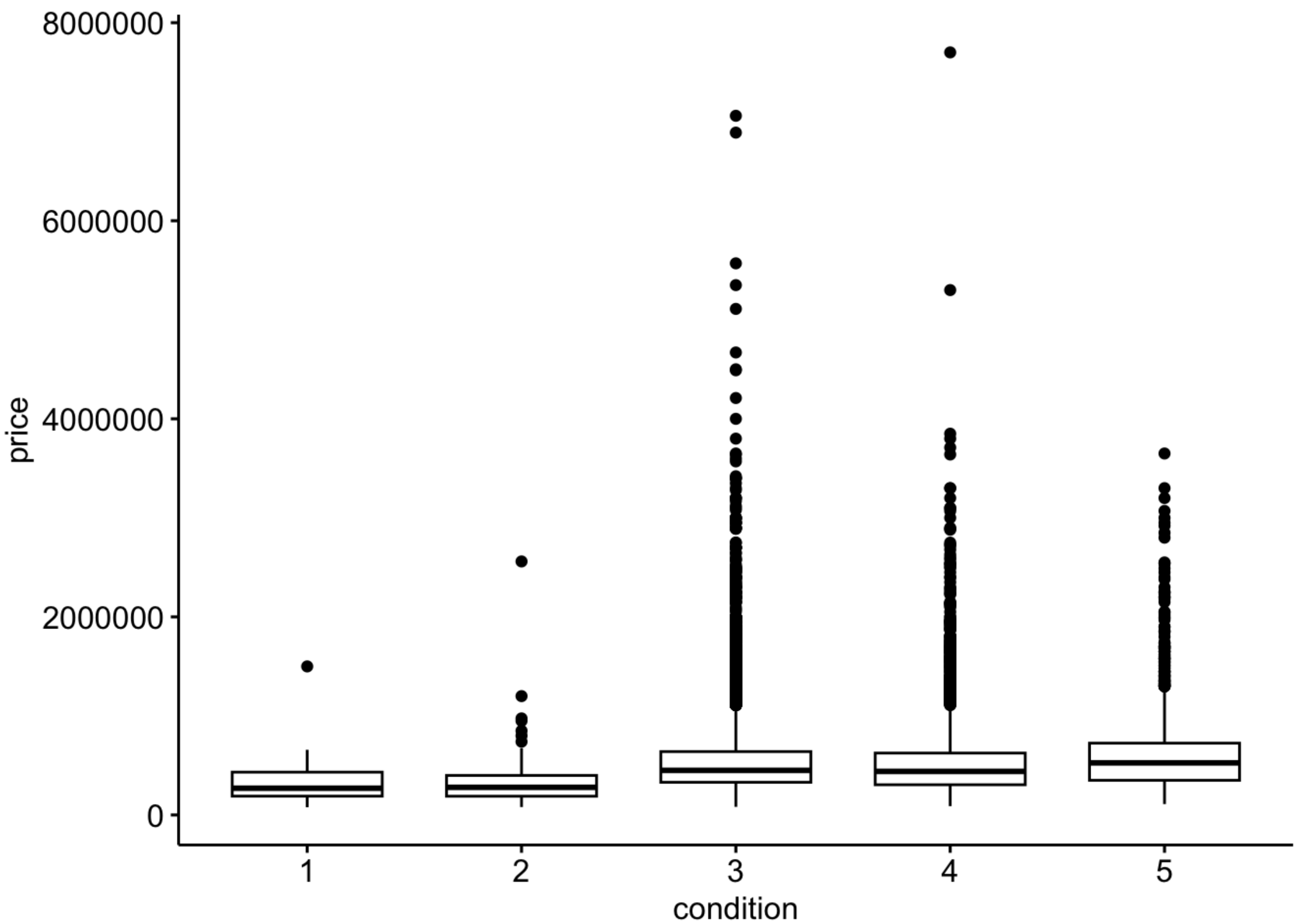
```
## Anova and Turkey test for price vs condition and plotting the distribution
## Calculate frequency, mean and standard deviation
final_merged_data %>% group_by(condition) %>% summarise(condition_freq = n(),price_mean
 = mean(price, na.rm = TRUE), price_sd = sd(price, na.rm = TRUE))
```

```
## # 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(78000,7700000))
```



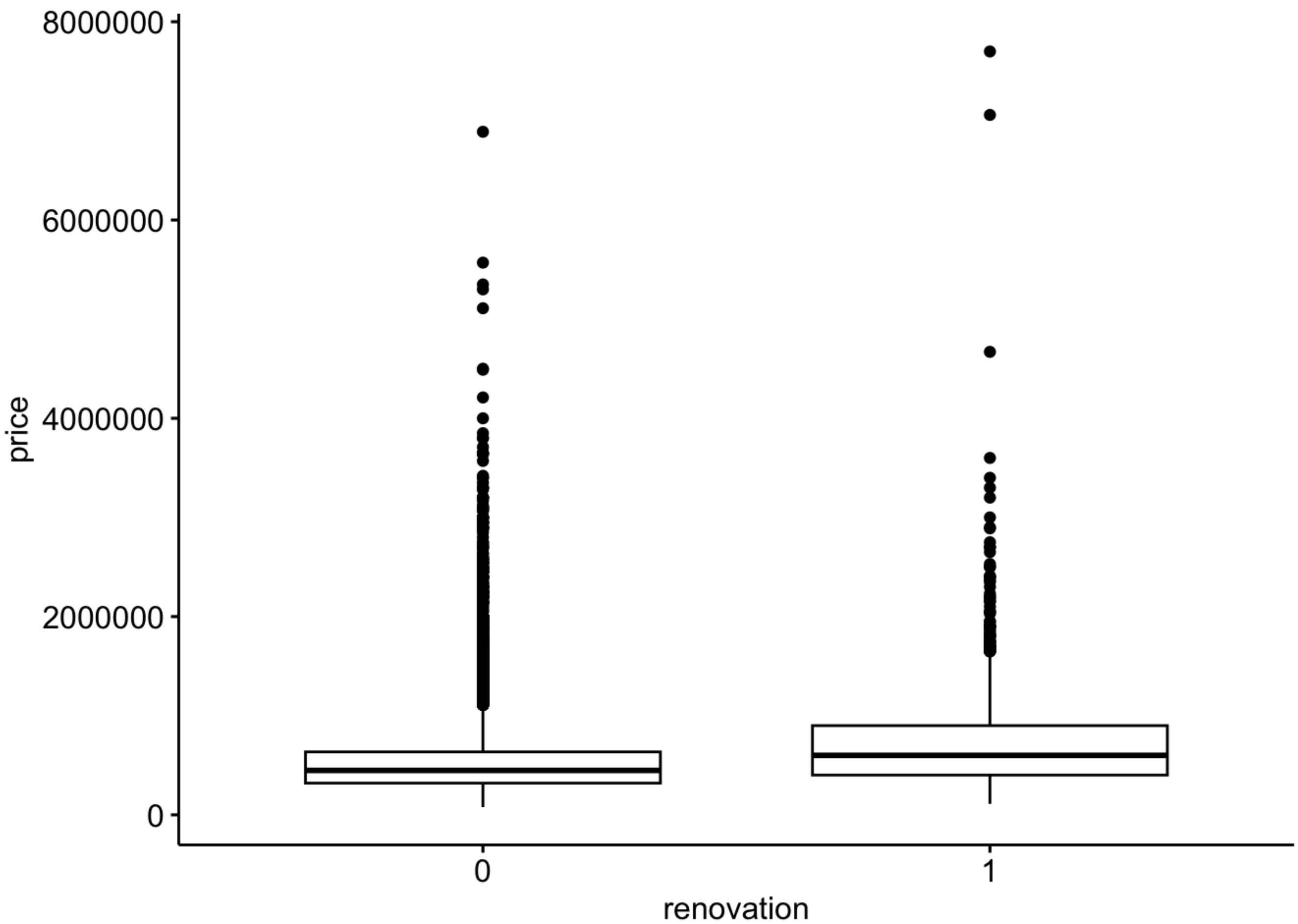

```
## Anova and Turkey test for price vs renovation and plotting the distribution
## Calculate frequency, mean and standard deviation
final_merged_data %>% group_by(renovation) %>% summarise(renovation_freq = n(), price_me
an = mean(price, na.rm = TRUE), price_sd = sd(price, na.rm = TRUE))
```

```
## # 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 <0.0000000000000002
## Residuals   21595 2868250023356209   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(78000,7700000))
```




```
## Anova and Turkey test 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_mean = mean(pr
ice, na.rm = TRUE), price_sd = sd(price, na.rm = TRUE))
```

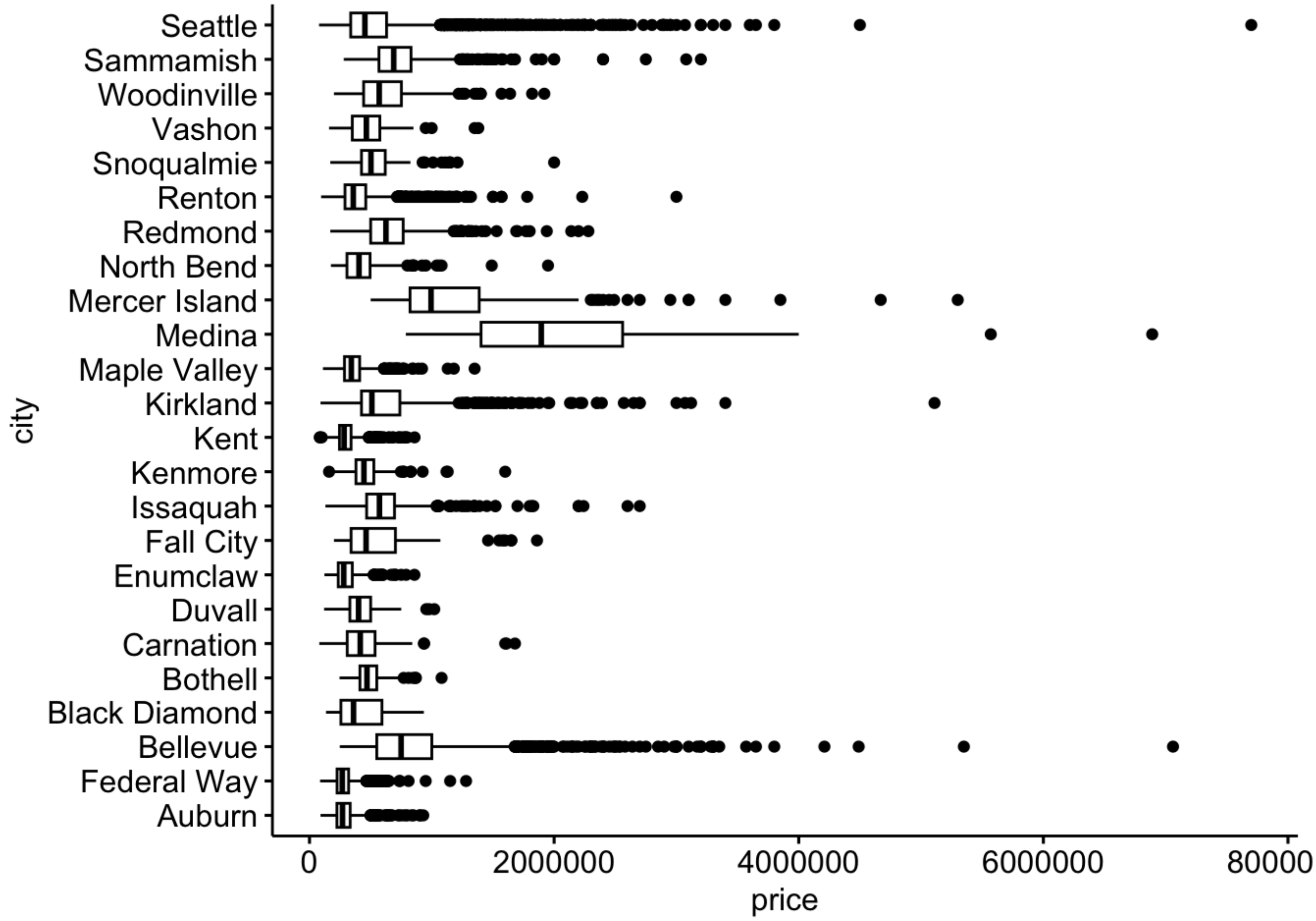
```
## # A tibble: 24 × 4
##    city          city_freq price_mean price_sd
##    <chr>             <int>      <dbl>    <dbl>
##  1 Auburn              911    291648.  108422.
##  2 Bellevue           1407    898466.  559782.
##  3 Black Diamond       100    423666.  195415.
##  4 Bothell             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 <0.0000000000000002
## Residuals    21573 2176477801366957    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_fli
p()
```

```
## Coordinate system already present. Adding new coordinate system, which will
## replace the existing one.
```



Data Modelling

Loading the splitted data pre processed Data

Multiple Linear Regression

```
model <- lm(price~bedrooms+bathrooms+floors+waterfront+condition+sqft_living15+sqft_lot1
5+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
```

```
model_fit <- lm(price~bedrooms+bathrooms+floors+waterfront+condition+sqft_living15+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>                              1068603616353455 377642
## - bedrooms       1    265214779758 1068868831133213 377644
## - sqft_lot15     1    742749127364 1069346365480819 377651
## - floors         1   7538482691711 1076142099045166 377746
## - condition      1  20889274545050 1089492890898504 377933
## - basement       1  24474152529446 1093077768882900 377982
## - renovation     1  24816750138909 1093420366492364 377987
## - bathrooms      1  38041483134714 1106645099488169 378169
## - waterfront     1  62586747317003 1131190363670458 378501
## - sqft_living15  1 248005386066743 1316609002420198 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 1068603616353455 377642.2
```

```
linear_model1 <- lm(price~bedrooms+bathrooms+floors+waterfront+condition+sqft_living15+b
asement+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_living
15+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
```

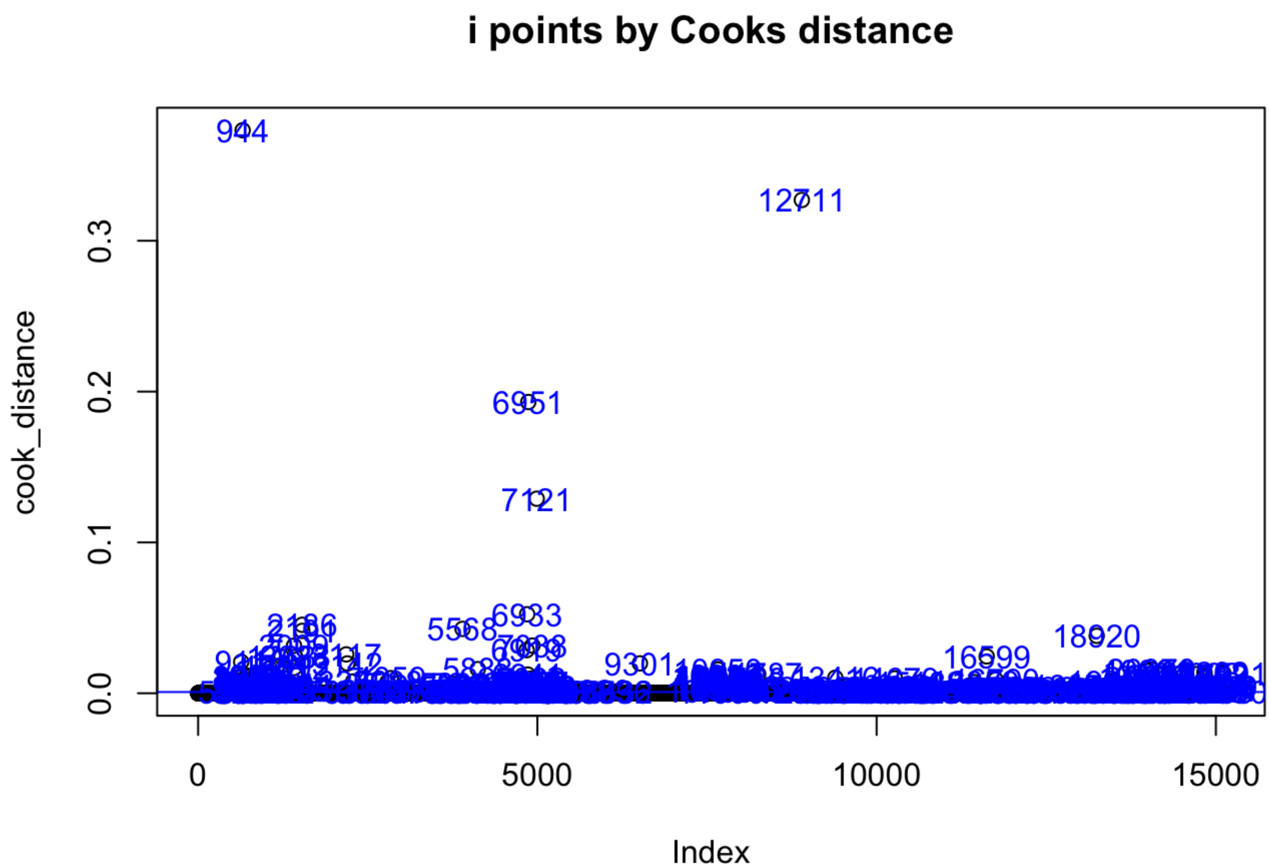
```
model_training_r2 <- summary(model_training$finalModel)$r.squared
model_training_results <- as.data.frame(model_training$results)
```

Influential point Detection using cook's distance

```
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")
```




```
i <- as.numeric(names(cook_distance)[(cook_distance > 4*mean(cook_distance, na.rm=T))])
head(train[i, ])
```

```
##        zip   price bedrooms bathrooms sqft_living sqft_lot floors waterfront
## 802  98003  225900        3       1.0        1510     8800      1          0
## 1213 98005 1000000        5       2.5        3150    50094      2          0
## 1222 98005  596000        3       2.5        1730     2631      2          0
## 1226 98005  556000        4       2.5        2230     7200      1          0
## 1227 98005  851500        3       2.0        3200    18184      1          0
## 1233 98005  699000        4       2.5        2440    14470      1          0
##      view condition grade sqft_above yr_built     lat     long sqft_living15
## 802     0         4     7       1010     1963 47.3290 -122.330          1290
## 1213    0         4     9       3150     1969 47.6387 -122.177          3600
## 1222    0         3     8       1730     2001 47.5878 -122.165          1730
## 1226    0         4     7       1220     1957 47.5890 -122.156          1920
## 1227    0         5     8       2000     1977 47.6034 -122.172          1670
## 1233    0         4     9       1660     1970 47.6401 -122.168          2810
##      sqft_lot15 basement renovation        city
## 802        8470        1          0 Federal Way
## 1213      48787        0          0    Bellevue
## 1222       2751        0          0    Bellevue
## 1226       7200        1          0    Bellevue
## 1227       7416        1          0    Bellevue
## 1233      15564        1          0    Bellevue
```

```
i_data <- train[i, ]
i_ol <- inner_join(ol_data,i_data)
```

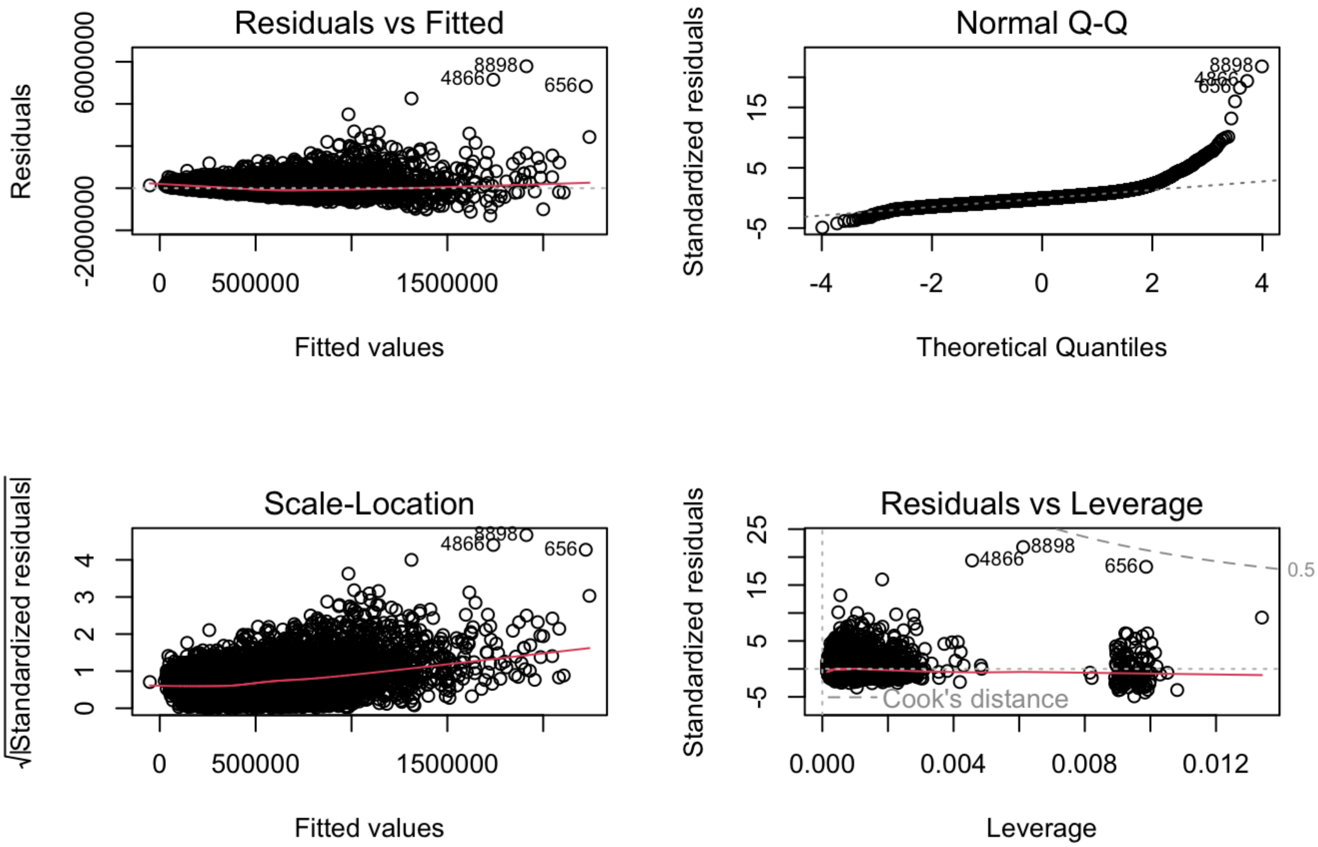
```
## Joining, by = c("zip", "price", "bedrooms", "bathrooms", "sqft_living",
## "sqft_lot", "floors", "waterfront", "view", "condition", "grade", "sqft_above",
## "yr_built", "lat", "long", "sqft_living15", "sqft_lot15", "basement",
## "renovation", "city")
```

```
t2 <- rbind(train,i_ol)
row.names(t2) <- NULL
linear_model2 <- lm(price~bedrooms+bathrooms+floors+waterfront+condition+sqft_living15+b
asement+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
```

Model Evaluation

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




```
## multicollinearilty test
## this shows there is no multicollinearilty in the model.
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$price, predicted=prediction_test)
abs_diff_test = mean(abs(actual_model_fitted_test$actual-actual_model_fitted_test$predic
ted)/actual_model_fitted_test$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"
```