MGT2003 (Slot C2 + TC2)
Fundamentals of Business Analytics Project Assignment

Phase 2

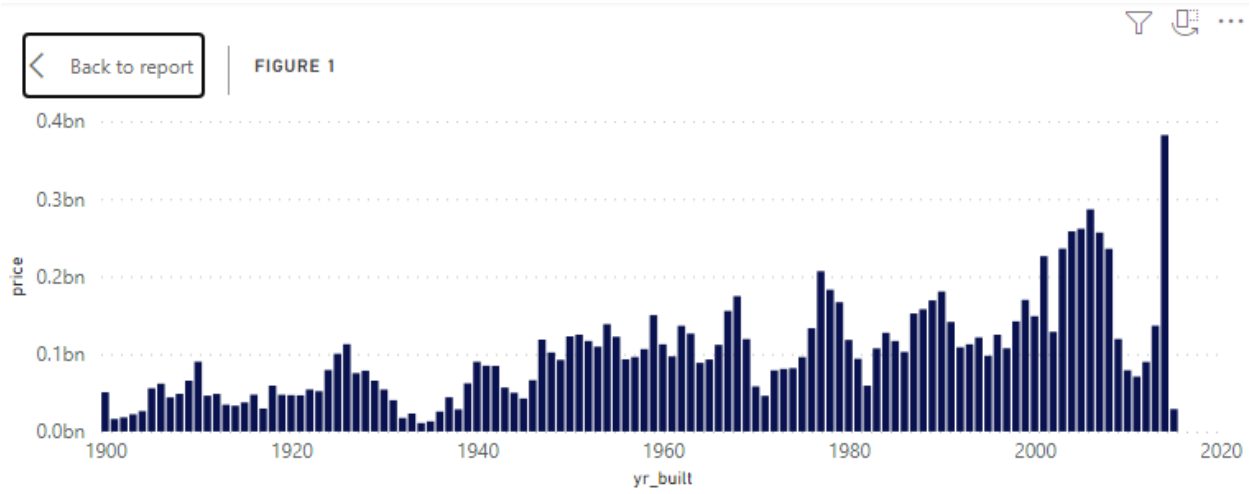
House Price Prediction

By:
Athrwa Deshmukh - 19BCE7381
Riya Deulkar - 19BEC7040

Dataset Dimensions:
Columns: 21
Rows/Samples: 21,613
Dataset Link

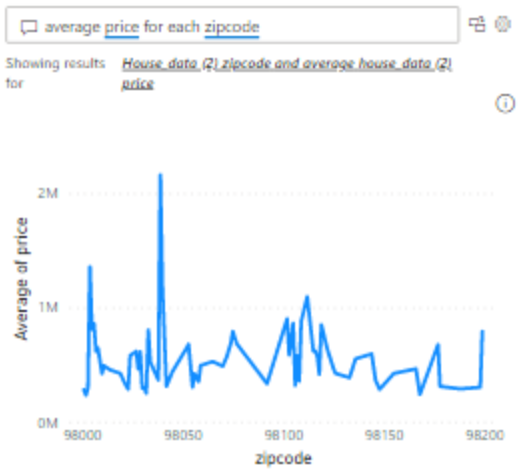
Tools Used - Power BI, MS Excel, R Studio

Analysis using Power BI :



Attribute information -
yr_built tells us in which year the house was built ranging from 1900 to 2015
Insight -
The above graph represents the yearly average price and as a general trend it keeps on increasing and took a dip around 2010.

Figure 2



zipcode	Average of price
98002	234284.04
98168	240328.37
98032	251296.24
98001	280804.69
98148	284908.60
98023	286732.79
98188	289078.35
98003	294111.28
98030	296187.98
Total	**540088.14**

zipcode	Average of price
98039	2160606.60
98004	1355927.08
98040	1194230.02
98112	1095499.34
98102	901258.27
98109	879623.62
98105	862825.23
98006	859684.78
98119	849448.02
Total	**540088.14**

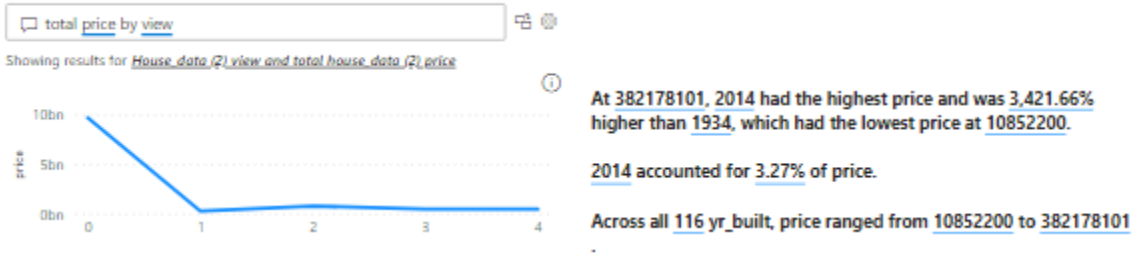
Attribute information -

The zipcode tells us about the location of the house in the form of zipcode/pincode.

Insight -

From the above graph and tables, we can see the average price for each zip code and we can conclude that zipcode 98002 is the cheapest to live in, whereas zipcode 98039 is the most expensive to live in.

Figure 3



Attribute information -

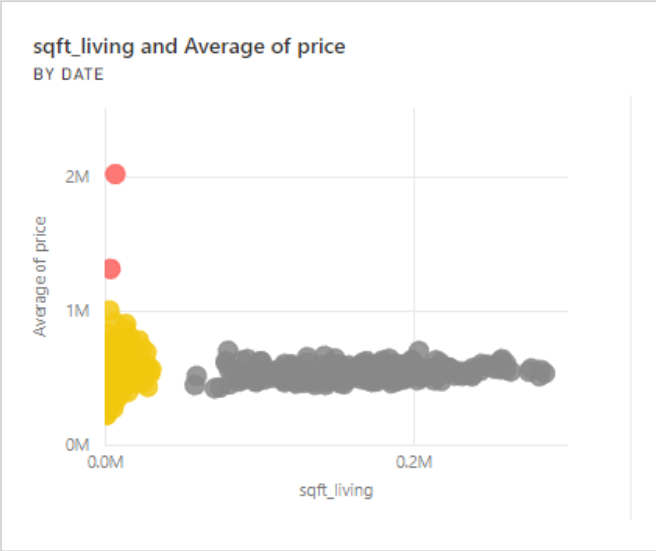
View tells us the number of times the house has been viewed

Minimum value - 0

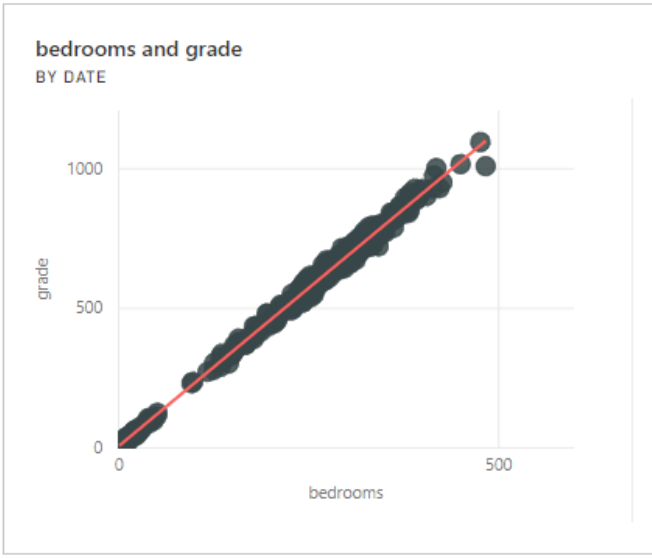
Maximum value - 4

Insight -

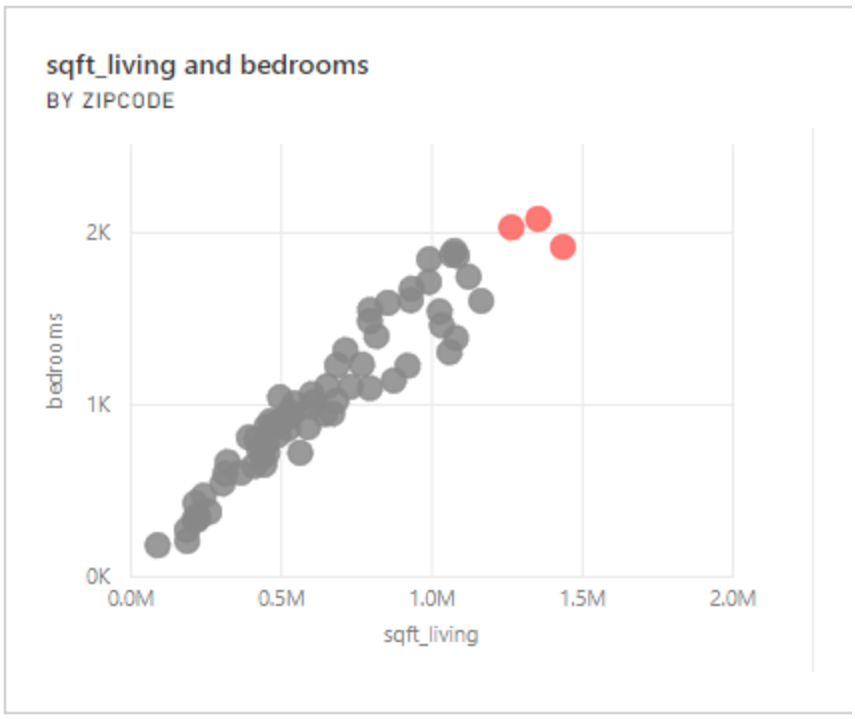
From Figure 3, we can see that as the views increase the price of the house decreases.



Insight -
Sqft_living and price form clusters when grouped by date except for 2 dates.



Attribute information -
Grade - overall grade given to the housing unit, based on a certain grading system ranging from 1 to 13
Bedrooms - Number of bedrooms in the house ranging from 0 to 33.
Insight -
From the above figure we can see that there is a nearly perfect positive correlation between bedrooms and grade



Insight -

From the above figure we can see that there is a positive correlation between sqft_living and bedrooms with respect to the zipcode.

R Code and Output :

(Lines starting with '#' are comments)

#Setting the working directory, in which the data is stored

```
setwd("F:/FBA Project")
```

#Reading the dataset as a data frame

```
data = read.csv("house_data.csv")
```

#Getting dimensions of the dataset

```
dim(data)
```

```
> #Setting the working directory, in which the data is stored
> setwd("F:/FBA Project")
> #Reading the dataset as a data frame
> data = read.csv("house_data.csv")
> #Getting dimensions of the dataset
> dim(data)
[1] 21613    21
```

#Preprocessing + Data Cleaning:

#Getting the summary of the data

```
summary(data)
```

```
> summary(data)
       id                 date               price            bedrooms        bathrooms
 Min.   :1.000e+06   Length:21613       Min.   :  75000   Min.   : 0.000   Min.   :0.000
 1st Qu.:2.123e+09   Class :character   1st Qu.: 321950   1st Qu.: 3.000   1st Qu.:1.750
 Median :3.905e+09   Mode  :character   Median : 450000   Median : 3.000   Median :2.250
 Mean   :4.580e+09                      Mean   : 540088   Mean   : 3.371   Mean   :2.115
 3rd Qu.:7.309e+09                      3rd Qu.: 645000   3rd Qu.: 4.000   3rd Qu.:2.500
 Max.   :9.900e+09                      Max.   :7700000   Max.   :33.000   Max.   :8.000
  sqft_living       sqft_lot           floors        waterfront            view          condition
 Min.   :  290   Min.   :    520   Min.   :1.000   Min.   :0.000000   Min.   :0.0000   Min.   :1.000
 1st Qu.: 1427   1st Qu.:   5040   1st Qu.:1.000   1st Qu.:0.000000   1st Qu.:0.0000   1st Qu.:3.000
 Median : 1910   Median :   7618   Median :1.500   Median :0.000000   Median :0.0000   Median :3.000
 Mean   : 2080   Mean   :  15107   Mean   :1.494   Mean   :0.007542   Mean   :0.2343   Mean   :3.409
 3rd Qu.: 2550   3rd Qu.:  10688   3rd Qu.:2.000   3rd Qu.:0.000000   3rd Qu.:0.0000   3rd Qu.:4.000
 Max.   :13540   Max.   :1651359   Max.   :3.500   Max.   :1.000000   Max.   :4.0000   Max.   :5.000
     grade          sqft_above   sqft_basement       yr_built     yr_renovated       zipcode
 Min.   : 1.000   Min.   : 290   Min.   :   0.0   Min.   :1900   Min.   :   0.0   Min.   :98001
 1st Qu.: 7.000   1st Qu.:1190   1st Qu.:   0.0   1st Qu.:1951   1st Qu.:   0.0   1st Qu.:98033
 Median : 7.000   Median :1560   Median :   0.0   Median :1975   Median :   0.0   Median :98065
 Mean   : 7.657   Mean   :1788   Mean   : 291.5   Mean   :1971   Mean   :  84.4   Mean   :98078
 3rd Qu.: 8.000   3rd Qu.:2210   3rd Qu.: 560.0   3rd Qu.:1997   3rd Qu.:   0.0   3rd Qu.:98118
 Max.   :13.000   Max.   :9410   Max.   :4820.0   Max.   :2015   Max.   :2015.0   Max.   :98199
      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   : 12768
 3rd Qu.:47.68   3rd Qu.:-122.1   3rd Qu.:2360   3rd Qu.: 10083
 Max.   :47.78   Max.   :-121.3   Max.   :6210   Max.   :871200
```

#Cleaning the date column to remove extra characters

```
data$date = substr(data$date,1,8)
```

#converting the string YYYYMMDD to YYYY-MM-DD date format

```
data$date <- as.Date(data$date, "%Y%m%d")
```

#checking the first few rows of the data

head(data)

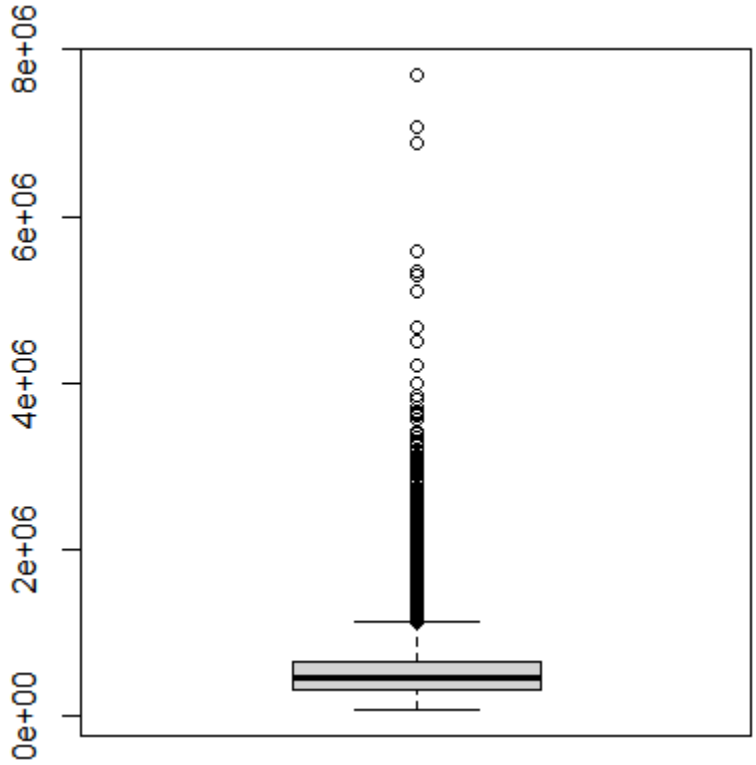
```
> #Cleaning the date column to remove extra characters
> #Cleaning the date column to remove extra characters
> data$date = substr(data$date,1,8)
> #converting the string YYYYMMDD to YYYY-MM-DD date format
> data$date <- as.Date(data$date, "%Y%m%d")
> #checking the first few rows of the data
> head(data)
           id       date   price bedrooms bathrooms sqft_living sqft_lot floors waterfront view
1 7129300520 2014-10-13  221900        3      1.00        1180     5650      1          0    0
2 6414100192 2014-12-09  538000        3      2.25        2570     7242      2          0    0
3 5631500400 2015-02-25  180000        2      1.00         770    10000      1          0    0
4 2487200875 2014-12-09  604000        4      3.00        1960     5000      1          0    0
5 1954400510 2015-02-18  510000        3      2.00        1680     8080      1          0    0
6 7237550310 2014-05-12 1225000        4      4.50        5420   101930      1          0    0
  condition grade sqft_above sqft_basement yr_built yr_renovated zipcode     lat     long
1         3     7       1180             0     1955            0   98178 47.5112 -122.257
2         3     7       2170           400     1951         1991   98125 47.7210 -122.319
3         3     6        770             0     1933            0   98028 47.7379 -122.233
4         5     7       1050           910     1965            0   98136 47.5208 -122.393
5         3     8       1680             0     1987            0   98074 47.6168 -122.045
6         3    11       3890          1530     2001            0   98053 47.6561 -122.005
  sqft_living15 sqft_lot15
1          1340       5650
2          1690       7639
3          2720       8062
4          1360       5000
5          1800       7503
6          4760     101930
```

#Checking for missing values in the dataset

summary(is.na(data))

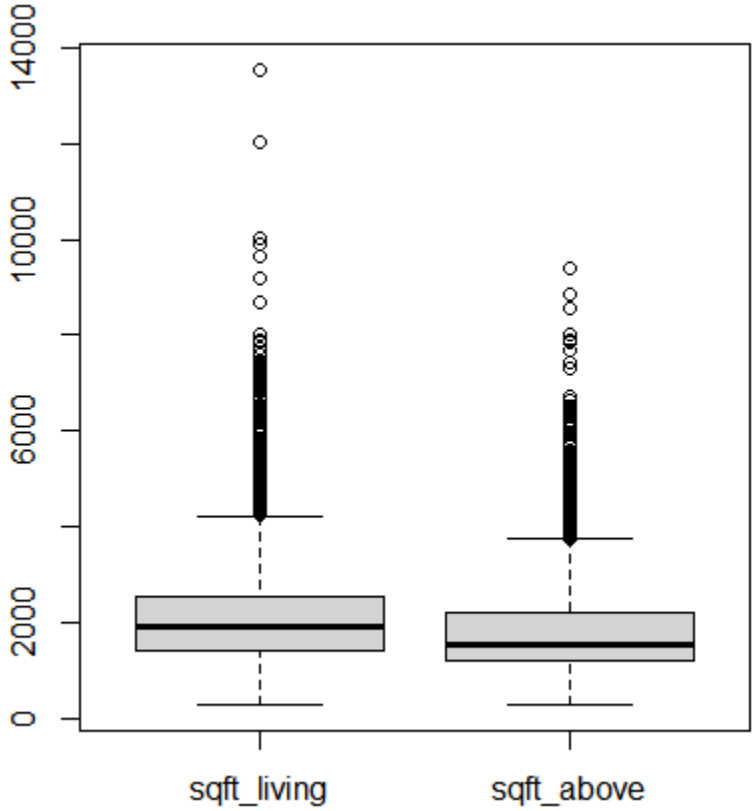
```
> #Checking for missing values in the dataset
> summary(is.na(data))
     id              date            price          bedrooms       bathrooms       sqft_living
 Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical
 FALSE:21613     FALSE:21613     FALSE:21613     FALSE:21613     FALSE:21613     FALSE:21613
  sqft_lot          floors        waterfront         view         condition         grade
 Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical
 FALSE:21613     FALSE:21613     FALSE:21613     FALSE:21613     FALSE:21613     FALSE:21613
 sqft_above      sqft_basement    yr_built       yr_renovated     zipcode            lat
 Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical
 FALSE:21613     FALSE:21613     FALSE:21613     FALSE:21613     FALSE:21613     FALSE:21613
    long         sqft_living15   sqft_lot15
 Mode :logical   Mode :logical   Mode :logical
 FALSE:21613     FALSE:21613     FALSE:21613
```

#Checking for outliers
boxplot(data[,'price'])



Attribute Information -
Name - Price
Minimum value - 75000
Maximum value - 7700000
The price column tells us about the pricing of the houses
Insight -
For the price column which is our target column, we observe from the above boxplot that there are various outliers including values nearing 80 Lacs.
We can also note that the range for the price lies approximately between 1 Lac to 15 Lacs
(Note : The prices in our dataset are in the form of dollars)

```
boxplot(data[,c('sqft_living','sqft_above')])
```

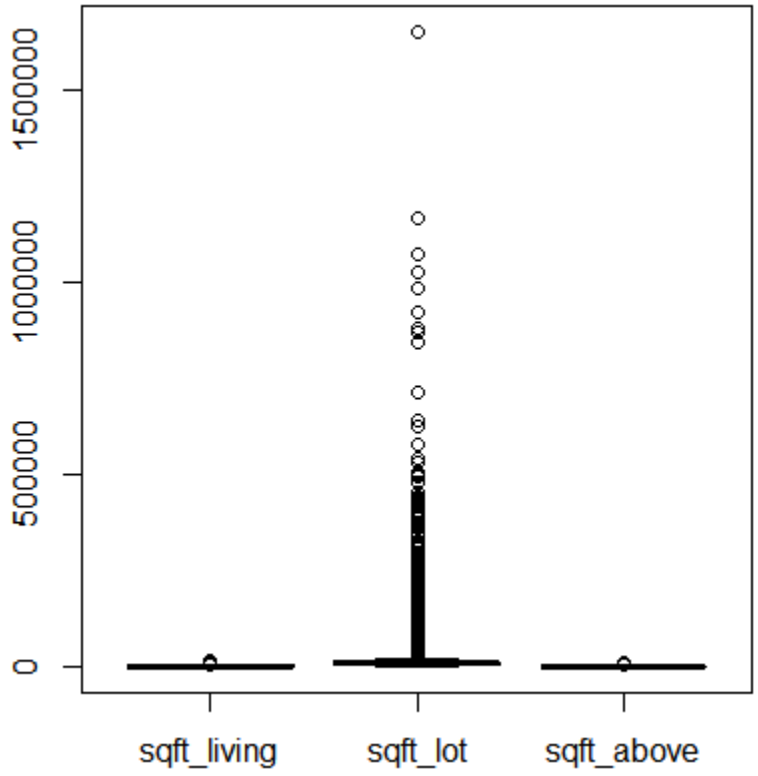


Attribute Information -
Name - sqft_living
Minimum value - 290
Maximum value - 13540
The sqft_living column tells us about the living area of the house measured in sq ft.

Name - sqft_above
Minimum value - 290
Maximum value - 9410
The sqft_above column tells us about the square foot of the house apart from the basement.

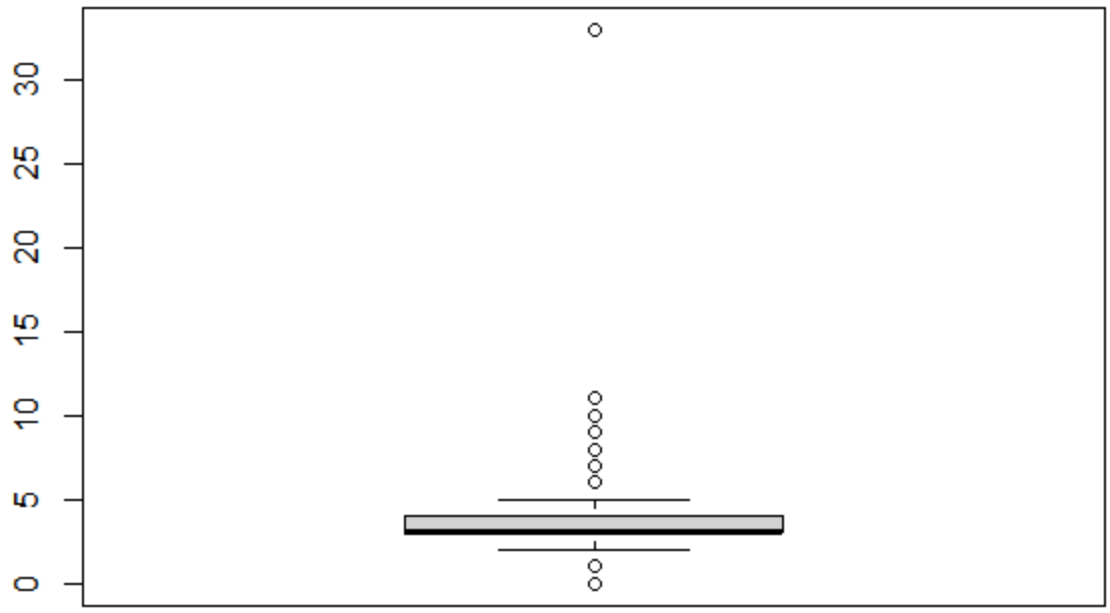
Insight -
For the sqft_living column and sqft_above column, we observe from the above boxplot that there are various outliers including values nearing 14000 and 10000 respectively.
We can also note that the range for maximum rows for sqft_living lies approximately between 1500 to 3000 and the range for maximum rows for sqft_above lies approximately between 1000 to 2000.

```
boxplot(data[,c('sqft_living','sqft_lot','sqft_above')])
```



Attribute Information -
Name - sqft_lot
Minimum value - 520
Maximum value - 1651359
The sqft_lot column tells us about the square footage of the lot of the house.

Insight -
For the sqft_lot, we observe from the above boxplot that there are various outliers including values nearing 1600000.
However, the range for sqft_lot is not evident from the above boxplot due to very high values of outliers.
As compared to sqft_living and sqft_above, there is a major difference in the number and value of outliers.



Attribute Information -
Name - Bedrooms
Minimum value - 33
Maximum value - 0
The bedrooms column tells us about the number of bedrooms in the house.

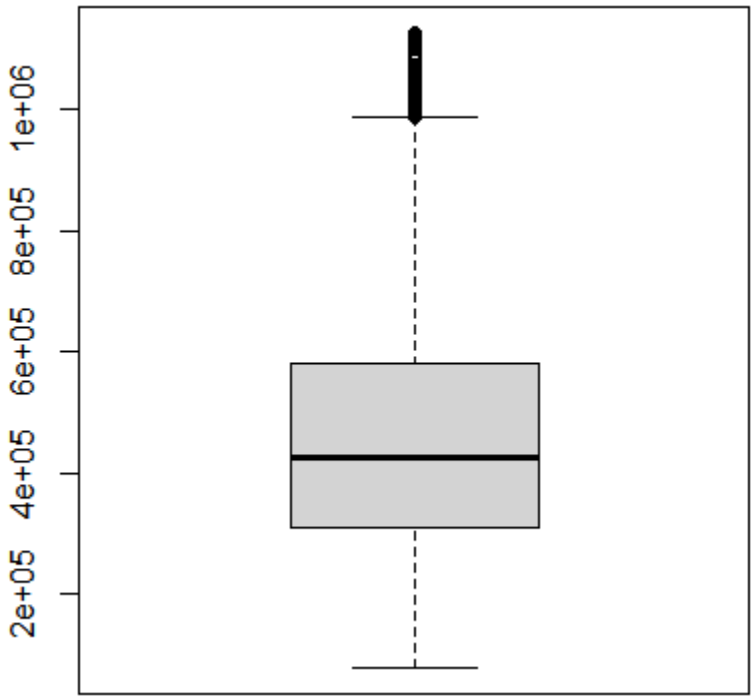
Insight -
For the bedrooms column, we observe from the above boxplot that there are various outliers including values nearing 30 and 0.
We can also note that the range for the price lies approximately between 2 to 4.

```
#Replacing outliers with NA
for(x in c('price','sqft_living','sqft_lot','sqft_above','bedrooms')){
  val = data[,x][data[,x] %in% boxplot.stats(data[,x])$out]
  data[,x][data[,x] %in% val] = NA
}
#Dropping rows with null values
data=tidyr::drop_na(data)
#Checking if all NA values have been dealt with
summary(is.na(data))
```

```
> #Replacing outliers with NA
> for(x in c('price','sqft_living','sqft_lot','sqft_above','bedrooms')){
+   val = data[,x][data[,x] %in% boxplot.stats(data[,x])$out]
+   data[,x][data[,x] %in% val] = NA
+ }
> #Dropping rows with null values
> data=tidyr::drop_na(data)
> #Checking if all NA values have been dealt with
> summary(is.na(data))
     id              date            price          bedrooms       bathrooms       sqft_living
 Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical
 FALSE:17716     FALSE:17716     FALSE:17716     FALSE:17716     FALSE:17716     FALSE:17716
  sqft_lot          floors        waterfront          view         condition         grade
 Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical
 FALSE:17716     FALSE:17716     FALSE:17716     FALSE:17716     FALSE:17716     FALSE:17716
 sqft_above      sqft_basement    yr_built       yr_renovated     zipcode           lat
 Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical   Mode :logical
 FALSE:17716     FALSE:17716     FALSE:17716     FALSE:17716     FALSE:17716     FALSE:17716
    long         sqft_living15   sqft_lot15
 Mode :logical   Mode :logical   Mode :logical
 FALSE:17716     FALSE:17716     FALSE:17716
```

#Checking the boxplot after removing outliers

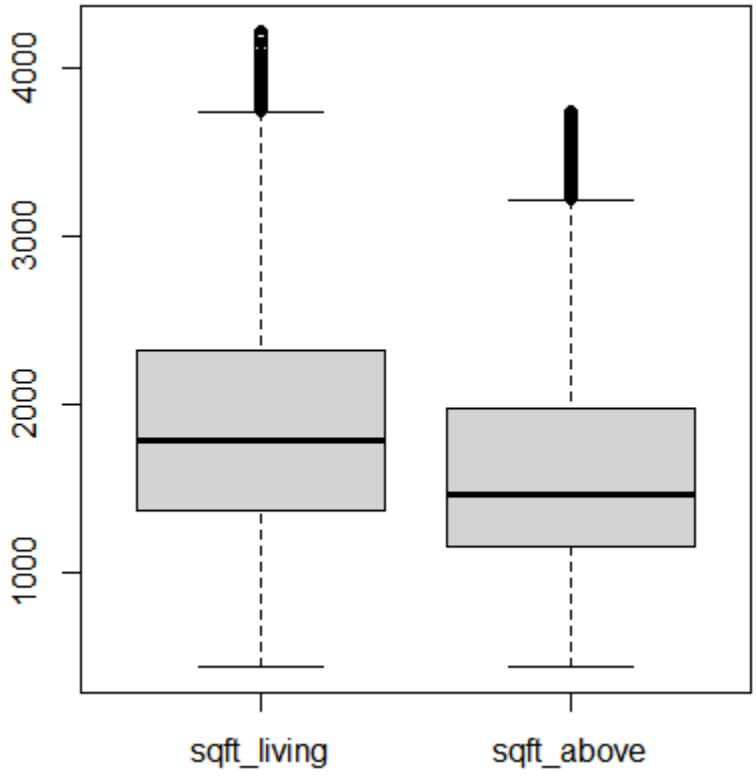
```
boxplot(data[,'price'])
```



Insight -

After removal of outliers, the range, the average value is clearly visible in the box plot and we can confirm that there are no more outliers.

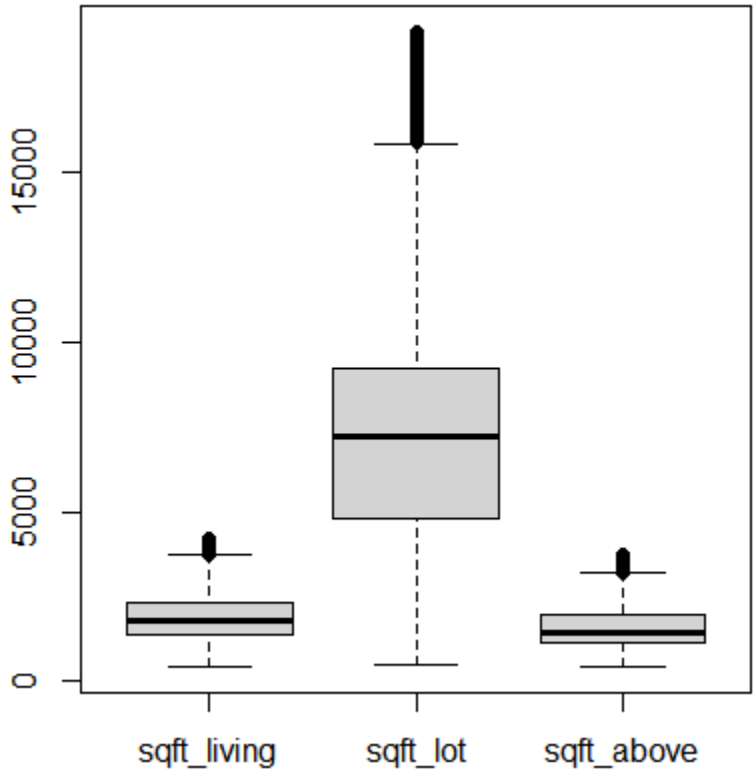
```
boxplot(data[,c('sqft_living','sqft_above')])
```



Insight -

After removal of outliers, the range, the average values for both the attributes are clearly visible in the box plot and we can confirm that there are no more outliers.

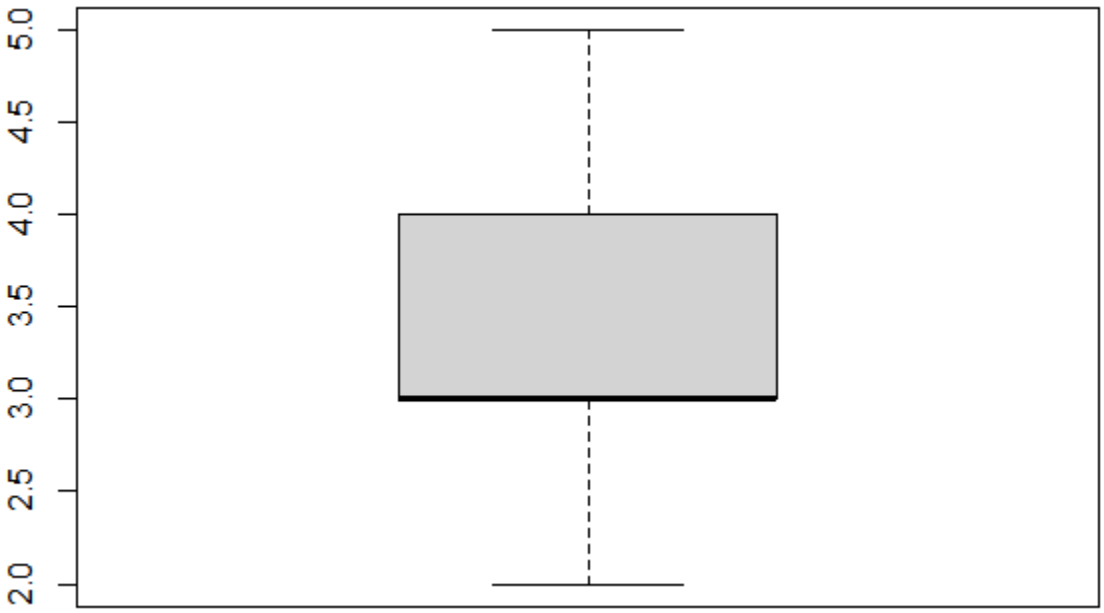
```
boxplot(data[,c('sqft_living','sqft_lot','sqft_above')])
```



Insight -

After removal of outliers, the range, the average values for all 3 attributes are clearly visible in the box plot and we can also note that all 3 attributes are comparable to a certain extent. Additionally we can confirm that there are no more outliers.

boxplot(data[,'bedrooms'])



Insight -

After removal of outliers, the range, the average value is clearly visible in the box plot and we can confirm that there are no more outliers.

#Dropping redundant columns

```
data = subset(data, select = -c(id,lat,long) )
#Checking new dimensions of the data
dim(data)
```

```
> #Dropping redundant columns
> data = subset(data, select = -c(id,lat,long) )
> #Checking new dimesnions of the data
> dim(data)
[1] 17716    18
```

Insight -

The ID column contains unique values and in no way is helpful to us in determining the price, making it redundant.

Similarly, the latitude (lat) and longitude (long) are not helpful in predicting the price and hence are removed along with the ID column.

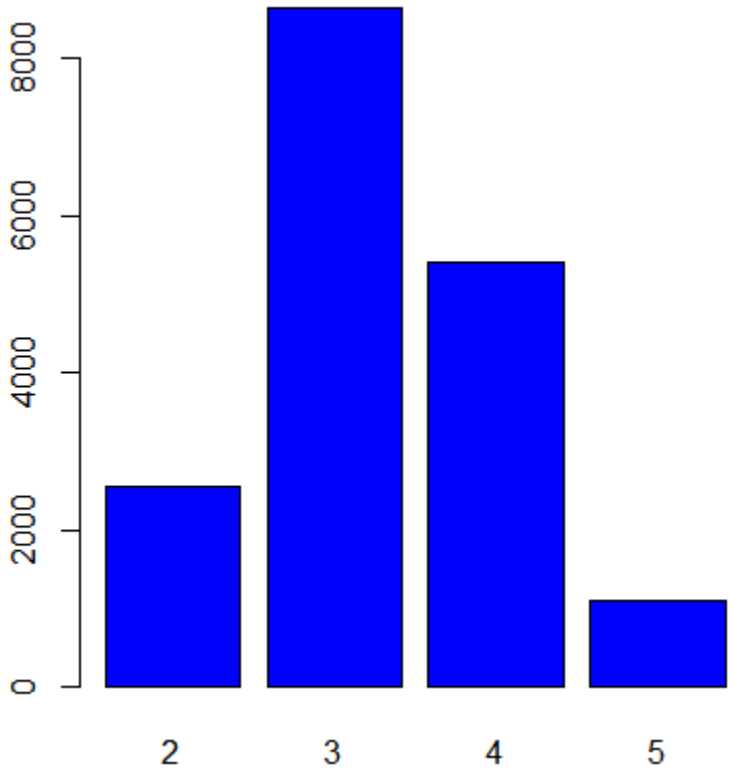
#Data Visualization:

```
library(ggplot2)
library(ggpubr)
#Plotting the Frequency of Unique values for various columns
table(data$bedrooms)
```

```
> #Plotting the Frequency of Unique values for various columns
> table(data$bedrooms)

   2    3    4    5
2553 8643 5418 1102
```

```
barplot(table(data$bedrooms),col='blue')
```



Attribute information -

It tells us about the number of bedrooms in the house.

Insight -

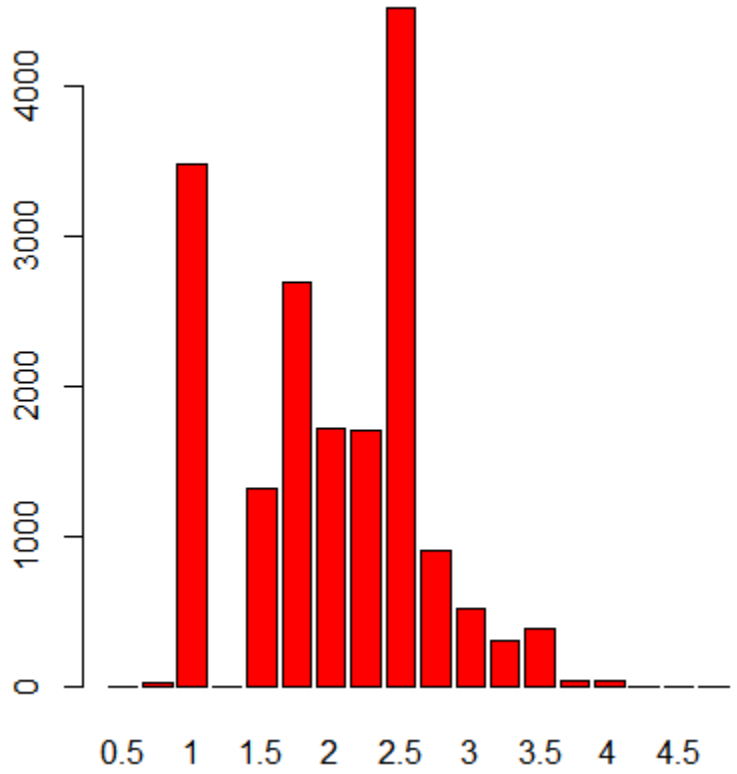
From the above table and graph, we observe that the maximum houses have 3 bedrooms whereas minimum houses have 5 bedrooms.

table(data$bathrooms)

```
> table(data$bathrooms)

  0.5 0.75    1 1.25  1.5 1.75    2 2.25  2.5 2.75    3 3.25  3.5 3.75    4 4.25  4.5 4.75
    3   32 3486    6 1317 2694 1722 1708 4516  915  521  308  392   40   37    9    9    1
```

barplot(table(data$bathrooms),col='red')



Attribute information -

It tells us about the number of bathrooms in the house.

Insight -

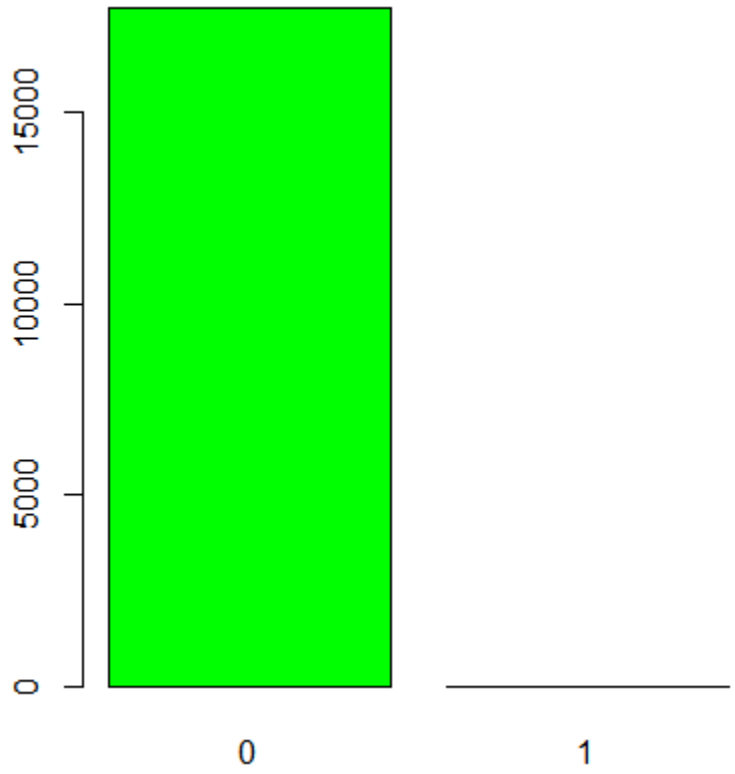
From the above table and graph, we observe that the maximum houses have 2.5 bathrooms whereas minimum houses have 4.75 bathrooms.

table(data$waterfront)

```
> table(data$waterfront)

    0     1
17685    31
```

barplot(table(data$waterfront),col='green')



Attribute information -

It tells us about houses which have a view of the waterfront.

Insight -

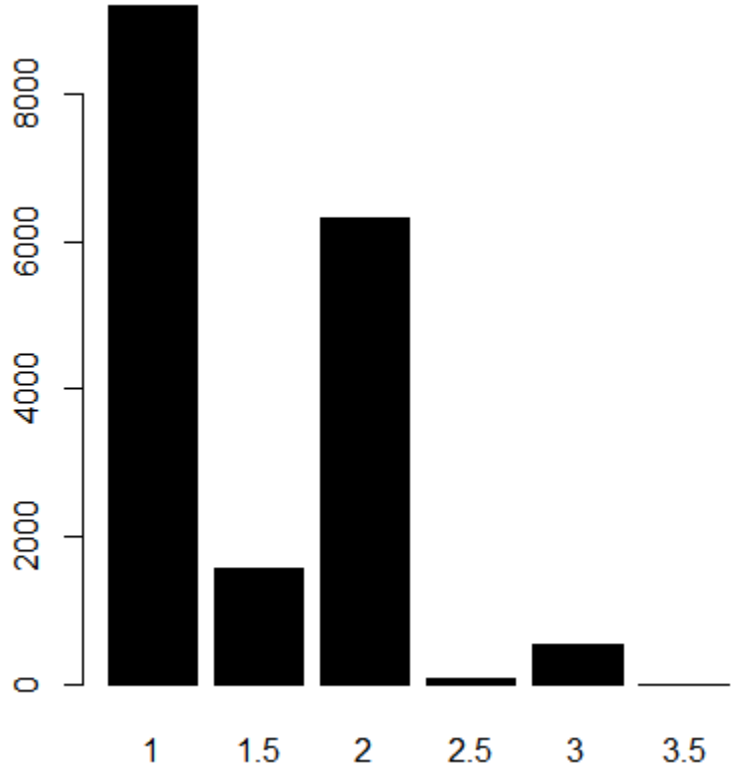
From the above table and graph, we observe that the maximum houses have no waterfront whereas minimum houses have waterfront.

table(data$floors)

```
> table(data$floors)

   1  1.5    2  2.5    3  3.5
9177 1571 6319   85  558    6
```

barplot(table(data$floors),col='black')



Attribute -

This tells us about no. of floors in the house.

Insight -

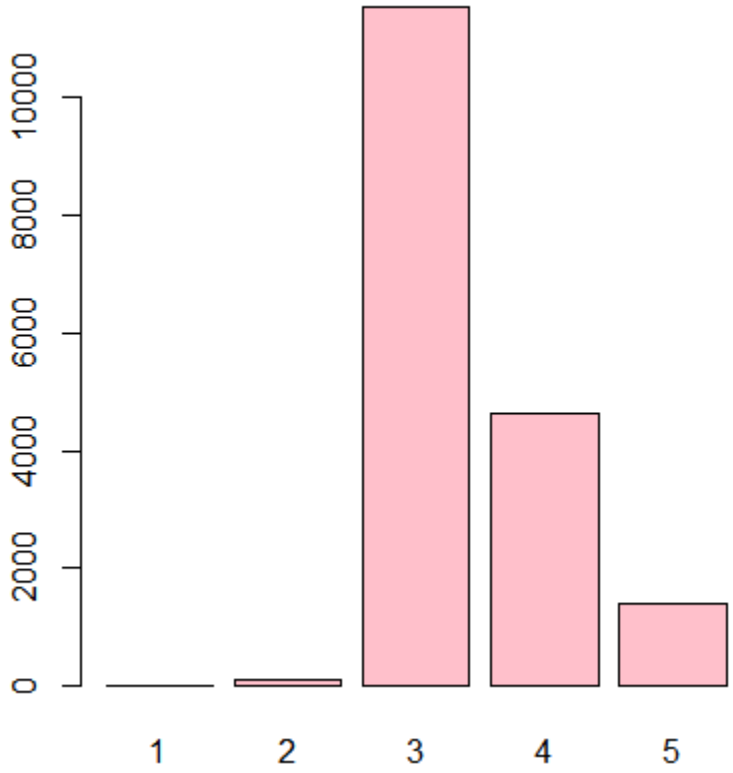
From the above table and graph, we observe that the maximum houses have 1 floor whereas minimum houses have 3.5 floors.

table(data$condition)

```
> table(data$condition)

    1     2     3     4     5
   17   120 11524  4645  1410
```

barplot(table(data$condition),col='pink')



Attribute Information -

Condition tells us about how good the condition is overall.

Insight -

From the above table and graph, we observe that the maximum houses are in condition 3 whereas minimum houses are in condition 1.

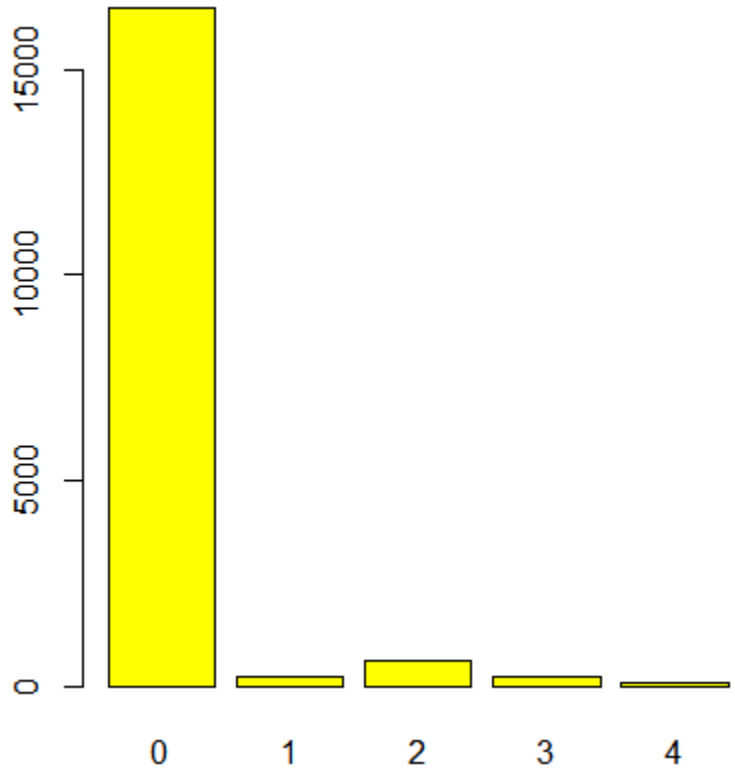
Majority of the houses fall under condition 3, 4 or 5.

table(data$view)

```
> table(data$view)

    0     1     2     3     4
16476   241   648   253    98
```

barplot(table(data$view),col='yellow')



Attribute Information -

It tells us about the number of times the house has been viewed.

Insight -

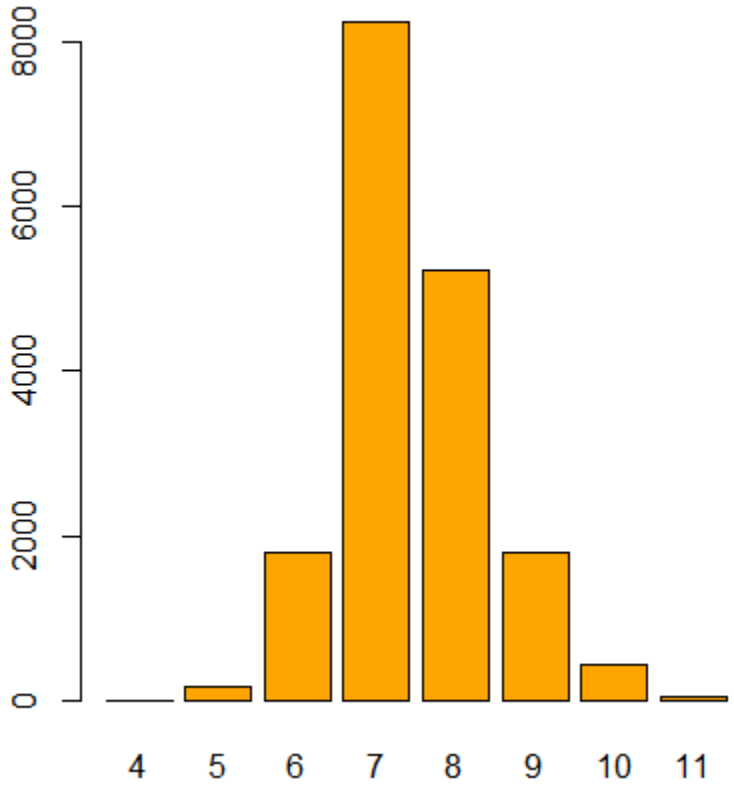
From the above table and graph, we observe that the maximum view is 0 whereas the minimum is 4.

table(data$grade)

```
> table(data$grade)

   4    5    6    7    8    9   10   11
  10  167 1806 8228 5213 1808  441   43
```

barplot(table(data$grade),col='orange')



Attribute Information -

It tells us about the overall grade given to the housing unit.

Insight -

From the above table and graph, we observe that the maximum grade given is 8 whereas the minimum grade given is 4.

#For Correlation HeatMap

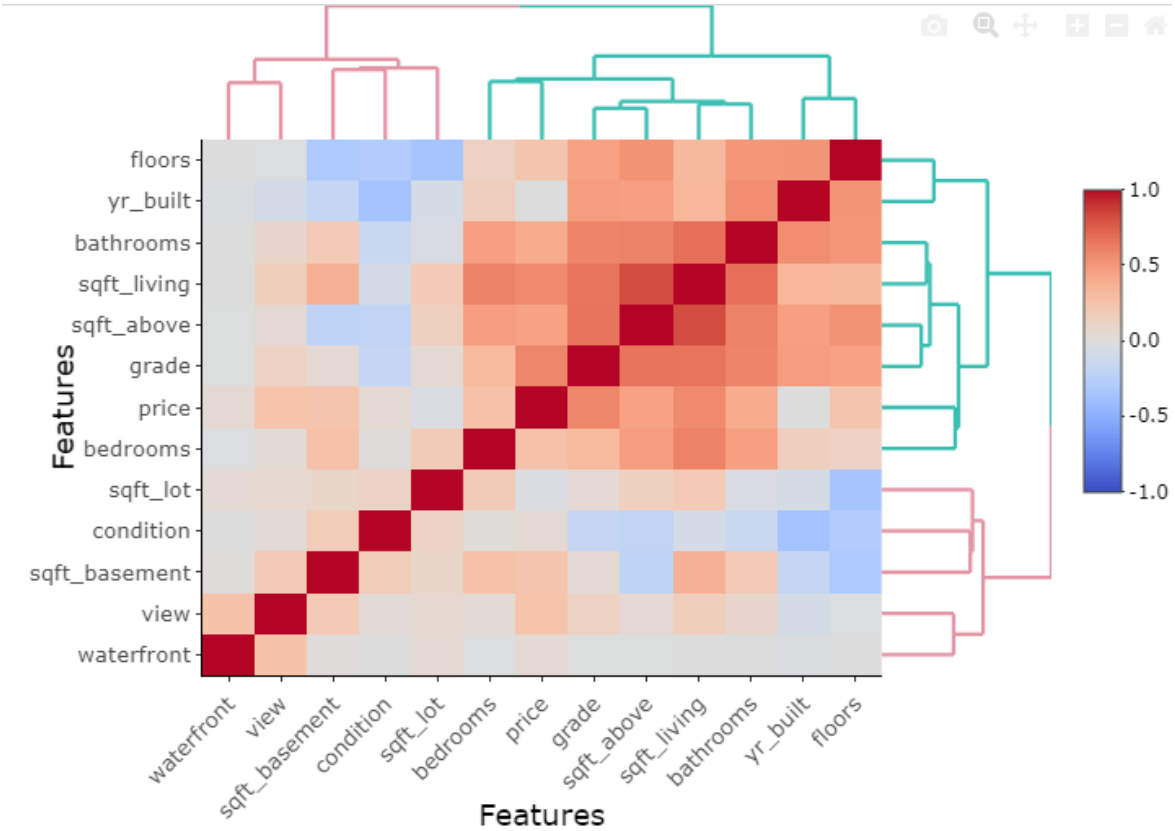
library(heatmaply)

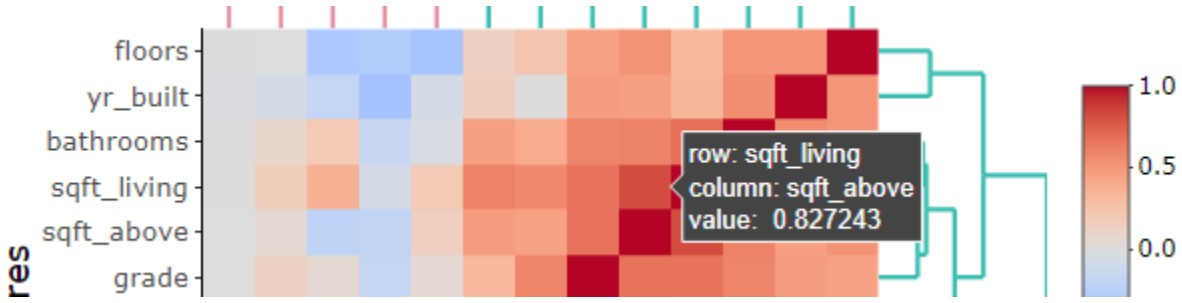
 # Plotting corr heatmap

df = subset(data, select = -c(date,zipcode,yr_renovated,sqft_lot15,sqft_living15))

```
> # Plotting corr heatmap
> df = subset(data, select = -c(date,zipcode,yr_renovated,sqft_lot15,sqft_living15) )
> heatmaply_cor(x = cor(df), xlab = "Features", ylab = "Features", k_col = 2, k_row = 2)
```

heatmaply_cor(x = cor(df), xlab = "Features", ylab = "Features", k_col = 2, k_row = 2)





Insight -

The above heatmap tells us about the correlation between various features.
The possible range of values for the correlation coefficient is -1.0 to 1.0. In other words, the values cannot exceed 1.0 or be less than -1.0. A correlation of -1.0 indicates a perfect negative correlation, and a correlation of 1.0 indicates a perfect positive correlation. If the correlation coefficient is greater than zero, it is a positive relationship. Conversely, if the value is less than zero, it is a negative relationship. A value of zero indicates that there is no relationship between the two variables.
From the above heatmap , we observe that the brown areas represent positive correlation, the blue areas represent negative correlation and the white areas represent nearly no correlation.
We can eliminate one of the columns from sqft_above and sqft_living as their correlation coefficient is 0.827243, which is nearly a perfect positive correlation (As both the features are nearly similar while training the model)

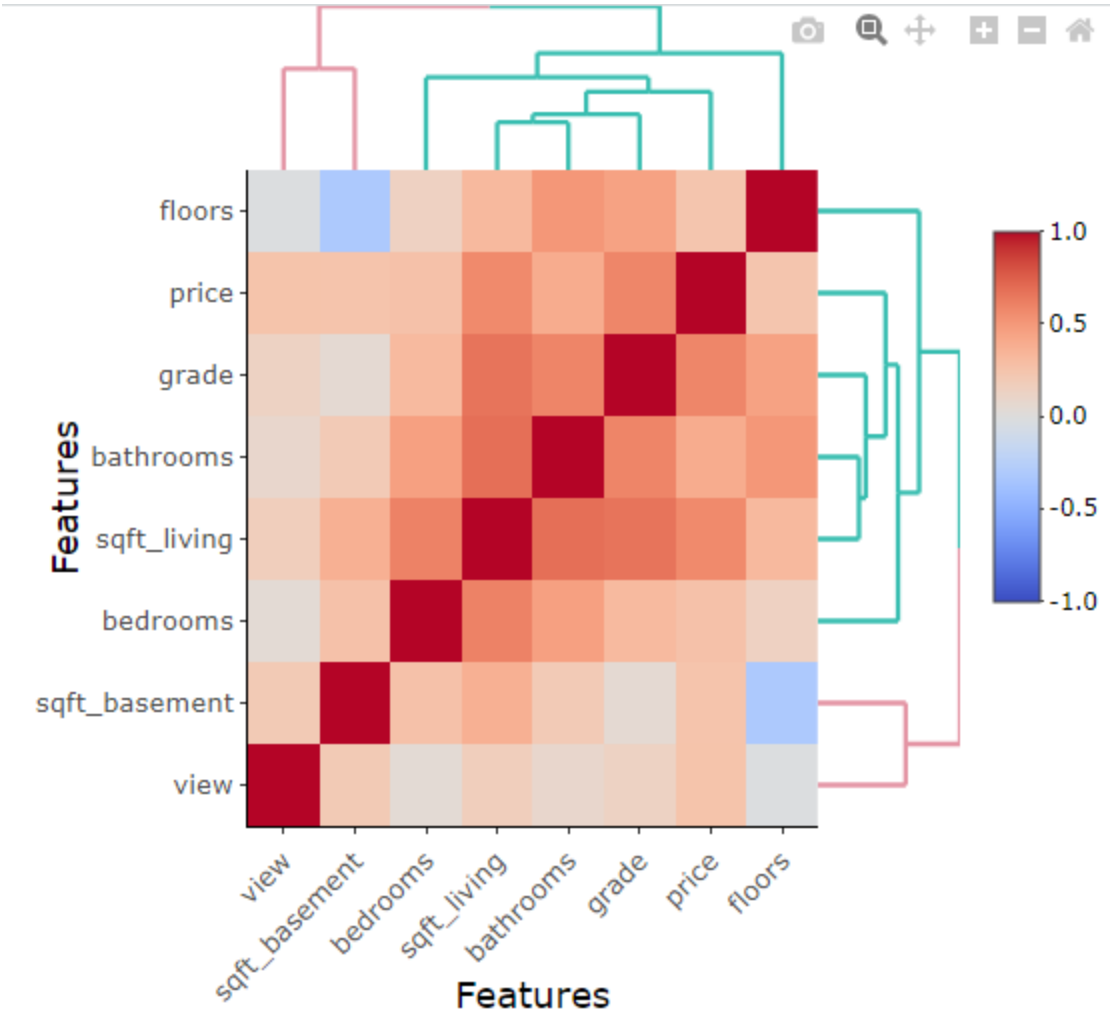
```
#dropping redundant columns based on heatmap of correlation
data=subset(data,select=-c(sqft_above))
#Checking Correlations of features with target i.e price
cor(df,df$price)
```

```
> #dropping redundant columns based on heatmap of correlation
> data=subset(data,select=-c(sqft_above))
> #Checking Correlations of features with target i.e price
> cor(df,df$price)
                      [,1]
price          1.000000000
bedrooms       0.270048302
bathrooms      0.405073221
sqft_living    0.577727087
sqft_lot      -0.024130076
floors         0.245623106
waterfront     0.054975075
view           0.257060886
condition      0.058778714
grade          0.590723988
sqft_above     0.457236027
sqft_basement  0.254261101
yr_built       0.002065127
```

```
#Dropping Factors that do not have much effect on the price
data=subset(data,select=-c(yr_built,condition,waterfront,sqft_lot))
dim(data)
#Checking Heatmap after dropping redundant columns
df = subset(data, select = -c(date,zipcode,yr_renovated,sqft_lot15,sqft_living15) )
```

```
> #Dropping Factors that do not have much effect on the price
> data=subset(data,select=-c(yr_built,condition,waterfront,sqft_lot))
> dim(data)
[1] 17716    13
> #Checking Heatmap after dropping redundant columns
> df = subset(data, select = -c(date,zipcode,yr_renovated,sqft_lot15,sqft_living15) )
```

```
heatmaply_cor(x = cor(df), xlab = "Features", ylab = "Features", k_col = 2, k_row = 2)
```



Insight -

From the above heatmap , we observe that grade and sqft_living have the best correlation with the price.