House Price Prediction

University of Texas at Dallas

BUAN-6356.003

Shujing sun

Group – 7

Sai Madhan Muthyam

Hari Shanker Reddy Madana

Yashwanth Yalam

Prashanth Thamshetti

Hari Sai Krishna

Sadgun Chikoti

Data Source:

We have taken the data source from Kaggle Website, and the data set comprises different property attributes from different cities in Washington state in the US. The table below shows and simplifies the description of the considered dataset.

1	date	This refers to the date on which the information was collected.
2	price	Price of the house
3	bedrooms	No. of bedrooms in the house
4	Bathrooms	No. of bathrooms in the house
5	sqft_living	Area of the plot used for living
6	sqft_lot	The total area of the plot
7	floors	No. of floors in the house
8	waterfront	No. of waterfronts in the house
9	View	View of the house (on a scale of 0 to 4)
10	condition	Condition of the house (on a scale of 0 to 5)
11	sqft_above	The area above the ground
12	sqft_basement	Area of the basement
13	yr_built	Year in which the house was built
14	yr_renovated	Year in which the house was renovated
15	street	Indicates Street name and location
16	City	Indicates city and location
17	state zip	Indicates state zip and location
18	Country	Indicates country

Data Mining Objective:

We would like to develop a multiple linear regression method to develop a predictive model that can accurately predict the price of a house based on its features such as the number of bedrooms, bathrooms, square footage, age of the house, and other relevant variables. The aim is to identify the key predictors of house prices and create a model that can generalize well on unseen data. The multiple linear regression models are evaluated and compared based on their performance metrics such as R-squared, adjusted R-squared, and p-values, to determine the most suitable model for predicting house prices. And also include exploratory data analysis and outlier detection techniques to ensure the validity and reliability of the predictive model.

Input:

In this project, we aim to predict the price of houses based on various attributes such as the number of bedrooms and bathrooms, the area of the plot used for living, the total area of the plot, the number of floors, the presence of waterfront, the view of the house, the condition of the house, the area above the ground, the area of the basement, the year in which the house was built, the year in which the house was renovated, the street name and location, the city and location, the state zip and location, and the country. We will use multiple linear regression methods to model the relationship between the independent variables and the dependent variable (house price) and evaluate the accuracy of the model using various performance metrics such as mean squared error and R-squared.

Output:

The output of this study should be a predictive model that can predict the price of a house based on its attributes such as the number of bedrooms, bathrooms, living area, lot area, location, and other relevant factors. The model should be able to provide accurate predictions for the price of a house, given its characteristics and other relevant information.

Libraries to be Loaded:

```
library(ggplot2)

library(caret)
```

Reading the data:

```
house_price <- read.csv (file='data.csv', stringsAsFactors = FALSE)

str(house_price)
```

displays the structure of the data frame, including the number of observations (rows) and variables (columns), the data types of each variable, and the first few values of each variable.

```
> str(house_price)
'data.frame':   4600 obs. of  18 variables:
 $ date         : chr  "2014-05-02 00:00:00" "2014-05-02 00:00:00" "2014-05-02
00:00:00" "2014-05-02 00:00:00" ...
 $ price        : num  313000 2384000 342000 420000 550000 ...
 $ bedrooms     : num  3 5 3 3 4 2 2 4 3 4 ...
 $ bathrooms    : num  1.5 2.5 2 2.25 2.5 1 2 2.5 2.5 2 ...
 $ sqft_living  : int  1340 3650 1930 2000 1940 880 1350 2710 2430 1520 ...
 $ sqft_lot     : int  7912 9050 11947 8030 10500 6380 2560 35868 88426 6200
...
 $ floors       : num  1.5 2 1 1 1 1 1 2 1 1.5 ...
 $ waterfront   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ view         : int  0 4 0 0 0 0 0 0 0 0 ...
 $ condition    : int  3 5 4 4 4 3 3 3 4 3 ...
 $ sqft_above   : int  1340 3370 1930 1000 1140 880 1350 2710 1570 1520 ...
 $ sqft_basement: int  0 280 0 1000 800 0 0 0 860 0 ...
 $ yr_built     : int  1955 1921 1966 1963 1976 1938 1976 1989 1985 1945 ...
 $ yr_renovated : int  2005 0 0 0 1992 1994 0 0 0 2010 ...
 $ street       : chr  "18810 Densmore Ave N" "709 W Blaine St" "26206-26214 14
3rd Ave SE" "857 170th Pl NE" ...
 $ city         : chr  "Shoreline" "Seattle" "Kent" "Bellevue" ...
 $ statezip     : chr  "WA 98133" "WA 98119" "WA 98042" "WA 98008" ...
 $ country      : chr  "USA" "USA" "USA" "USA" ...
```

head(house_price,5)

displays the first 5 rows of the house_price data frame, showing the values of each variable for those 5 observations. This can be useful for quickly getting a sense of what the data looks like and what variables are included.

```
> head(house_price,5)
                 date   price bedrooms bathrooms sqft_living sqft_lot
1 2014-05-02 00:00:00  313000        3      1.50        1340     7912
2 2014-05-02 00:00:00 2384000        5      2.50        3650     9050
3 2014-05-02 00:00:00  342000        3      2.00        1930    11947
4 2014-05-02 00:00:00  420000        3      2.25        2000     8030
5 2014-05-02 00:00:00  550000        4      2.50        1940    10500
  floors waterfront view condition sqft_above sqft_basement yr_built
1    1.5          0    0         3       1340             0     1955
2    2.0          0    4         5       3370           280     1921
3    1.0          0    0         4       1930             0     1966
4    1.0          0    0         4       1000          1000     1963
5    1.0          0    0         4       1140           800     1976
  yr_renovated                 street      city statezip country
1         2005     18810 Densmore Ave N Shoreline WA 98133     USA
2            0          709 W Blaine St   Seattle WA 98119     USA
3            0 26206-26214 143rd Ave SE      Kent WA 98042     USA
4            0          857 170th Pl NE  Bellevue WA 98008     USA
5         1992         9105 170th Ave NE   Redmond WA 98052     USA
```

summary(house_price)

Summary statistics of house_price

```
> summary(house_price)
     date               price             bedrooms       bathrooms
 Length:4600        Min.   :       0   Min.   :0.000   Min.   :0.000
 Class :character   1st Qu.:  322875   1st Qu.:3.000   1st Qu.:1.750
 Mode  :character   Median :  460943   Median :3.000   Median :2.250
                    Mean   :  551963   Mean   :3.401   Mean   :2.161
                    3rd Qu.:  654962   3rd Qu.:4.000   3rd Qu.:2.500
                    Max.   :26590000   Max.   :9.000   Max.   :8.000
  sqft_living       sqft_lot            floors         waterfront
 Min.   :  370   Min.   :    638   Min.   :1.000   Min.   :0.000000
 1st Qu.: 1460   1st Qu.:   5001   1st Qu.:1.000   1st Qu.:0.000000
 Median : 1980   Median :   7683   Median :1.500   Median :0.000000
 Mean   : 2139   Mean   :  14852   Mean   :1.512   Mean   :0.007174
 3rd Qu.: 2620   3rd Qu.:  11001   3rd Qu.:2.000   3rd Qu.:0.000000
 Max.   :13540   Max.   :1074218   Max.   :3.500   Max.   :1.000000
      view           condition       sqft_above    sqft_basement
 Min.   :0.0000   Min.   :1.000   Min.   : 370   Min.   :   0.0
 1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:1190   1st Qu.:   0.0
 Median :0.0000   Median :3.000   Median :1590   Median :   0.0
 Mean   :0.2407   Mean   :3.452   Mean   :1827   Mean   : 312.1
 3rd Qu.:0.0000   3rd Qu.:4.000   3rd Qu.:2300   3rd Qu.: 610.0
 Max.   :4.0000   Max.   :5.000   Max.   :9410   Max.   :4820.0
    yr_built     yr_renovated        street              city
 Min.   :1900   Min.   :   0.0   Length:4600        Length:4600
 1st Qu.:1951   1st Qu.:   0.0   Class :character   Class :character
 Median :1976   Median :   0.0   Mode  :character   Mode  :character
 Mean   :1971   Mean   : 808.6
 3rd Qu.:1997   3rd Qu.:1999.0
 Max.   :2014   Max.   :2014.0
   statezip           country
 Length:4600        Length:4600
 Class :character   Class :character
 Mode  :character   Mode  :character
```

<u>Finding</u>:

Waterfront, View, Condition, Street, City, State Zip, and Country are Categorical Variables, and the remaining are Continuous variables.

<u>Data Pre-processing</u>:

Removing columns that are not helpful in developing a model for predicting house prices.

```
> nrow(house_price[house_price$waterfront == 0,])
[1] 4567
```

df <- house_price[,-c(1,8,15,16,17,18)]

Out of 4600 rows for the waterfront column, 4567 rows' value is 0 so we are removing columns like date, street, city, state zip, and the country will not contribute to the model, so we are removing these columns.

cor(df)

```
> cor(df)
                    price    bedrooms  bathrooms sqft_living     sqft_lot      floors
price          1.00000000  0.20033629  0.3271099  0.43041003  0.050451295  0.15146080
bedrooms       0.20033629  1.00000000  0.5459199  0.59488406  0.068819355  0.17789490
bathrooms      0.32710992  0.54591993  1.0000000  0.76115370  0.107837479  0.48642757
sqft_living    0.43041003  0.59488406  0.7611537  1.00000000  0.210538454  0.34485027
sqft_lot       0.05045130  0.06881935  0.1078375  0.21053845  1.000000000  0.00374975
floors         0.15146080  0.17789490  0.4864276  0.34485027  0.003749750  1.00000000
view           0.22850417  0.11102800  0.2119602  0.31100944  0.073906741  0.03121095
condition      0.03491454  0.02507986 -0.1199943 -0.06282598  0.000558114 -0.27501339
sqft_above     0.36756960  0.48470534  0.6899184  0.87644325  0.216454651  0.52281374
sqft_basement  0.21042657  0.33416525  0.2980202  0.44720554  0.034842303 -0.25550982
yr_built       0.02185683  0.14246104  0.4634977  0.28777522  0.050706346  0.46748066
yr_renovated  -0.02877365 -0.06108157 -0.2158862 -0.12281688 -0.022730309 -0.23399567
                     view    condition  sqft_above sqft_basement    yr_built yr_renovated
price          0.22850417  0.034914537  0.36756960    0.21042657  0.02185683  -0.02877365
bedrooms       0.11102800  0.025079856  0.48470534    0.33416525  0.14246104  -0.06108157
bathrooms      0.21196025 -0.119994341  0.68991841    0.29802018  0.46349768  -0.21588624
sqft_living    0.31100944 -0.062825979  0.87644325    0.44720554  0.28777522  -0.12281688
sqft_lot       0.07390674  0.000558114  0.21645465    0.03484230  0.05070635  -0.02273031
floors         0.03121095 -0.275013395  0.52281374   -0.25550982  0.46748066  -0.23399567
view           1.00000000  0.063077281  0.17432671    0.32160180 -0.06446506   0.02296700
condition      0.06307728  1.000000000 -0.17819634    0.20063235 -0.39969823  -0.18681841
sqft_above     0.17432671 -0.178196344  1.00000000   -0.03872299  0.40853521  -0.16042556
sqft_basement  0.32160180  0.200632350 -0.03872299    1.00000000 -0.16167480   0.04312492
yr_built      -0.06446506 -0.399698234  0.40853521   -0.16167480  1.00000000  -0.32134228
yr_renovated   0.02296700 -0.186818414 -0.16042556    0.04312492 -0.32134228   1.00000000
>
```

nrow(df[df$view == 0,])

```
> nrow(df[df$view == 0,])
[1] 4140
```

df <- df[,-7]

And also for view column, 4140 rows value is 0 almost 90 percent of values and also there is no strongly correlated with other predictor variables and response variable. So, it will not contribute to model and we are removing the view column.

current_year <- as.numeric(format(Sys.time(), "%Y"))

df$age_of_house <- current_year - df$yr_built

df <- df[,-10]

df$year_renovated <- df[,"yr_renovated"]

df <- df[,-10]

nrow(df[df$year_renovated ==0,])

df$year_renovated <- ifelse(df$year_renovated > 0, 1, 0)

nrow(df[df$sqft_basement ==0,])

df$sqft_basement<- ifelse(df$sqft_basement > 0, 1, 0)

Other than these parameters, there are a few parameters that still require some modifications. We can find the age of the house by subtracting the year in which it was built from the current year. For parameters such as 'sqft_basement' and 'yr_renovated' most of the values in the dataset are equal to 0, indicating that the house doesn't have a basement and it has not been renovated even once respectively. Thus, we will change the variable 'sqft_basement' to a zero-one variable (1 indicating the house has a basement and 0 indicating the doesn't have a basement). In a similar manner, we change 'yr_renovated' into a dichotomous variable (1 indicating the house has been renovated and 0 indicating the house has not been renovated).

colMeans(is.na(df))

Checking for Null values

```
> colMeans(is.na(df))
         price       bedrooms      bathrooms    sqft_living       sqft_lot         floors
             0              0              0              0              0              0
     condition     sqft_above  sqft_basement   age_of_house year_renovated
             0              0              0              0              0
```

summary(df$price)

```
> summary(df$price)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
       0   322875   460943   551963   654962 26590000
```

table(df$price==0)

```
> table(df$price==0)

FALSE  TRUE
 4551    49
```

df <- df[df$price!=0,]

However, there are 49 rows for which the price of the house is 0. Since the price cannot be zero, we will remove these rows for now.

Checking for outliers:

A useful method for checking for outliers in data is to plot a boxplot.

We will be searching for outliers in our dataset, which are data points that significantly differ from the rest of the data and can potentially cause errors in our analysis. It is important to identify and remove outliers, as our dataset contains such values.

par(mfrow=c(2, 4))

boxplot(df$price, main="Price")

boxplot(df$bedrooms, main="Bedrooms")

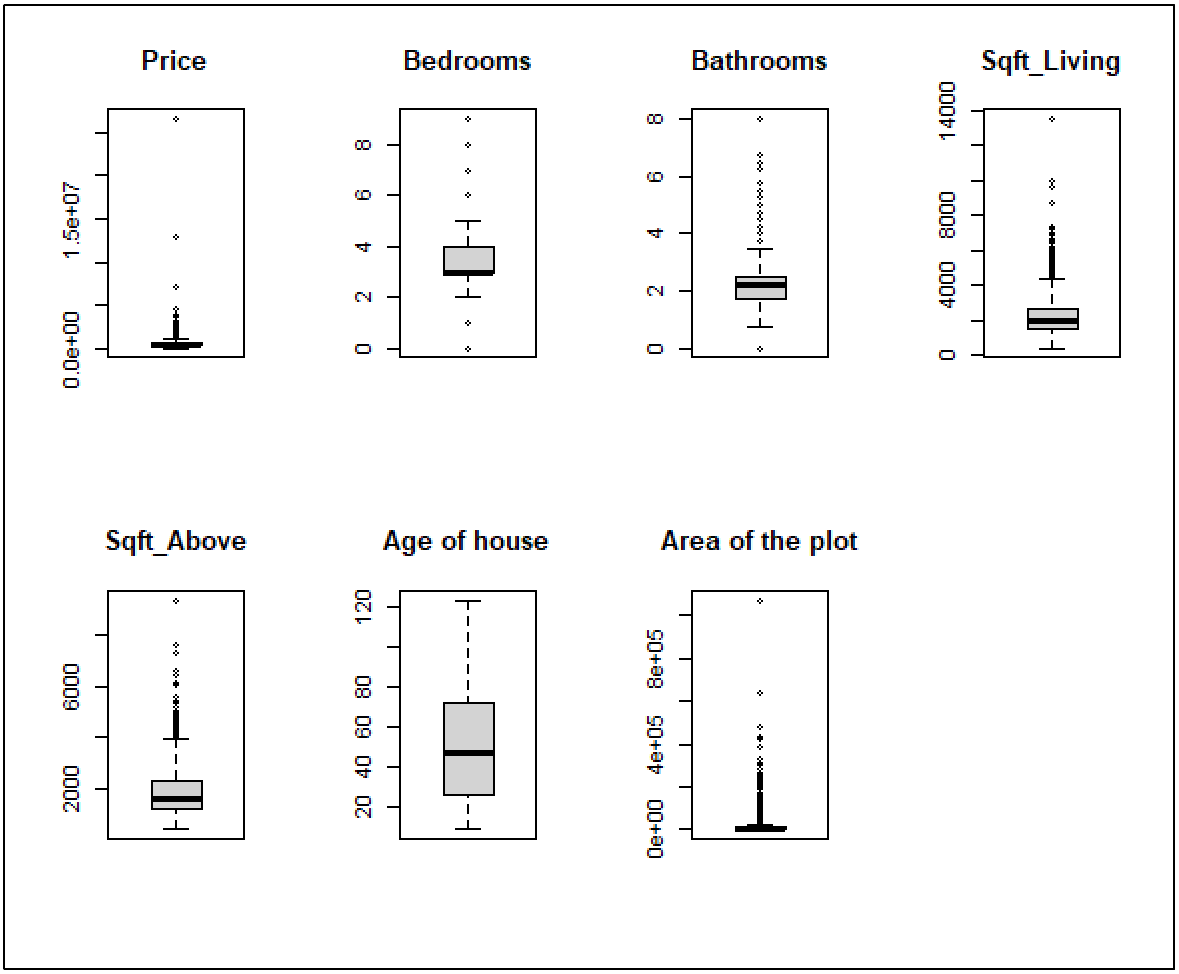
boxplot(df$bathrooms, main="Bathrooms")

boxplot(df$sqft_living, main="Sqft_Living")

boxplot(df$sqft_above, main="Sqft_Above")

boxplot(df$age_of_house, main="Age of house")

boxplot(df$sqft_lot, main="Area of the plot")



Clearly from the plots there are outliers.

To prepare the data for modeling, it is necessary to eliminate any outliers present in the dataset.

1st Method :

```
outlier_treat1 <- function(x){
  UC = quantile(x, p=0.99,na.rm=T)
  LC = quantile(x, p=0.01,na.rm=T)
  x=ifelse(x>UC,UC, x)
  x=ifelse(x<LC,LC, x)
  return(x)
}
```

df1 = data.frame(apply(df, 2, FUN=outlier_treat1))

min(df1$price)

max(df1$price)

```
> min(df1$price)
[1] 148000
> max(df1$price)
[1] 2016000
```

cor_matrix1 <- cor(df1)

cor_matrix1

```
> cor_matrix1 <- cor(df1)
> cor_matrix1
                      price    bedrooms  bathrooms sqft_living     sqft_lot      floors
price            1.00000000  0.34952968  0.5315714  0.68970401  0.111761200  0.27470702
bedrooms         0.34952968  1.00000000  0.5419382  0.61085931  0.090812769  0.17773065
bathrooms        0.53157139  0.54193815  1.0000000  0.75480449  0.122584463  0.49973155
sqft_living      0.68970401  0.61085931  0.7548045  1.00000000  0.244721341  0.35309178
sqft_lot         0.11176120  0.09081277  0.1225845  0.24472134  1.000000000 -0.00165711
floors           0.27470702  0.17773065  0.4997316  0.35309178 -0.001657110  1.00000000
condition        0.04967053  0.01298960 -0.1391405 -0.07426651 -0.003821659 -0.29022480
sqft_above       0.59485599  0.49290839  0.6824021  0.87142173  0.257463513  0.53267666
sqft_basement    0.18337977  0.17799248  0.1615690  0.20248146 -0.039771020 -0.27363642
age_of_house    -0.02926788 -0.14288783 -0.4739220 -0.29718722 -0.073717393 -0.46766500
year_renovated  -0.04794277 -0.06409787 -0.2227909 -0.12817210 -0.022218629 -0.23476973
                   condition sqft_above sqft_basement age_of_house year_renovated
price            0.049670534  0.5948560    0.18337977  -0.02926788    -0.04794277
bedrooms         0.012989601  0.4929084    0.17799248  -0.14288783    -0.06409787
bathrooms       -0.139140456  0.6824021    0.16156896  -0.47392200    -0.22279089
sqft_living     -0.074266509  0.8714217    0.20248146  -0.29718722    -0.12817210
sqft_lot        -0.003821659  0.2574635   -0.03977102  -0.07371739    -0.02221863
floors          -0.290224803  0.5326767   -0.27363642  -0.46766500    -0.23476973
condition        1.000000000 -0.1929195    0.17661086   0.42125915    -0.19845489
sqft_above      -0.192919476  1.0000000   -0.22682081  -0.41715382    -0.16429689
sqft_basement    0.176610856 -0.2268208    1.00000000   0.20064722     0.06195220
age_of_house     0.421259151 -0.4171538    0.20064722   1.00000000     0.32204870
year_renovated  -0.198454888 -0.1642969    0.06195220   0.32204870     1.00000000
```

model1 <- lm(price~.,data=df1)

summary(model1)

```
> summary(model1)

Call:
lm(formula = price ~ ., data = df1)

Residuals:
     Min       1Q   Median       3Q      Max 
-1373676  -123964   -14424    91746  1843894 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)    -3.183e+05  2.791e+04 -11.405  < 2e-16 ***
bedrooms       -5.598e+04  4.862e+03 -11.515  < 2e-16 ***
bathrooms       6.364e+04  7.812e+03   8.147 4.80e-16 ***
sqft_living     1.980e+02  1.473e+01  13.436  < 2e-16 ***
sqft_lot       -6.613e-01  1.292e-01  -5.118 3.22e-07 ***
floors          5.715e+04  8.356e+03   6.840 8.98e-12 ***
condition       3.023e+04  6.041e+03   5.005 5.81e-07 ***
sqft_above      7.791e+01  1.602e+01   4.863 1.20e-06 ***
sqft_basement   5.363e+04  1.261e+04   4.253 2.15e-05 ***
age_of_house    2.775e+03  1.525e+02  18.188  < 2e-16 ***
year_renovated  1.586e+04  7.680e+03   2.065    0.039 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 218700 on 4540 degrees of freedom
Multiple R-squared:  0.5481,	Adjusted R-squared:  0.5471 
F-statistic: 550.7 on 10 and 4540 DF,  p-value: < 2.2e-16
```

The model's residual standard error is 218700, and it has been calculated using 4540 degrees of freedom. The multiple R-squared value for the model is 0.5481, with an adjusted R-squared value of 0.5471. The F-statistic for the model is 550.7, using 10 and 4540 degrees of freedom. The p-value for the F-statistic is less than 2.2e-16.

 # 2nd Method :

```
SD <- function(v){
  sqrt(sum((v-mean(v))^2)/length(v))
}
outlier_treat2 <- function(x){
  mean_val = mean(x)
  sd_val = SD(x)
  UC = mean_val + 3 * sd_val
  LC = mean_val - 3 * sd_val
  x = ifelse(x > UC, UC, x)
  x = ifelse(x < LC, LC, x)
  return(x)
}
```

From this we can calculate the correlation matrix `cor_matrix2` for `df2`, and then runs a multiple linear regression model `model2` with `price` as the response variable and all other columns as predictor variables.

```
df2 <- data.frame(lapply(df, FUN = outlier_treat2))
min(df2$price)
max(df2$price)
```

```
> min(df2$price)
[1] 7800
> max(df2$price)
[1] 2249510
>
```

summary(df2)

```
> summary(df2)
     price            bedrooms        bathrooms      sqft_living      sqft_lot          floors     
 Min.   :   7800   Min.   :0.6812   Min.   :0.000   Min.   : 370   Min.   :   638   Min.   :1.000  
 1st Qu.: 326264   1st Qu.:3.0000   1st Qu.:1.750   1st Qu.:1460   1st Qu.:  5000   1st Qu.:1.000  
 Median : 465000   Median :3.0000   Median :2.250   Median :1970   Median :  7680   Median :1.500  
 Mean   : 545788   Mean   :3.3907   Mean   :2.150   Mean   :2119   Mean   : 12930   Mean   :1.512  
 3rd Qu.: 657500   3rd Qu.:4.0000   3rd Qu.:2.500   3rd Qu.:2610   3rd Qu.: 10978   3rd Qu.:2.000  
 Max.   :2249510   Max.   :6.1081   Max.   :4.484   Max.   :5000   Max.   :122716   Max.   :3.128  
   condition       sqft_above    sqft_basement     age_of_house   year_renovated  
 Min.   :1.424   Min.   : 370   Min.   :0.0000   Min.   :  9.0   Min.   :0.0000  
 1st Qu.:3.000   1st Qu.:1190   1st Qu.:0.0000   1st Qu.: 26.0   1st Qu.:0.0000  
 Median :3.000   Median :1590   Median :0.0000   Median : 47.0   Median :0.0000  
 Mean   :3.450   Mean   :1814   Mean   :0.4028   Mean   : 52.2   Mean   :0.4054  
 3rd Qu.:4.000   3rd Qu.:2300   3rd Qu.:1.0000   3rd Qu.: 72.0   3rd Qu.:1.0000  
 Max.   :5.000   Max.   :4385   Max.   :1.0000   Max.   :123.0   Max.   :1.0000  
```

cor_matrix2 <- cor(df2)

cor_matrix2

```
> cor_matrix2 <- cor(df2)
> cor_matrix2
                     price    bedrooms  bathrooms sqft_living     sqft_lot      floors    condition
price           1.00000000  0.34536807  0.5257257   0.6837489  0.127276636  0.27058387  0.056762597
bedrooms        0.34536807  1.00000000  0.5463298   0.6104388  0.106159963  0.17863993  0.023069013
bathrooms       0.52572573  0.54632976  1.0000000   0.7530274  0.133505917  0.49780194 -0.122497799
sqft_living     0.68374885  0.61043880  0.7530274   1.0000000  0.278002908  0.35337832 -0.063014996
sqft_lot        0.12727664  0.10615996  0.1335059   0.2780029  1.000000000 -0.01715452 -0.005836799
floors          0.27058387  0.17863993  0.4978019   0.3533783 -0.017154521  1.00000000 -0.275278784
condition       0.05676260  0.02306901 -0.1224978  -0.0630150 -0.005836799 -0.27527878  1.000000000
sqft_above      0.59017727  0.49345536  0.6806572   0.8714273  0.288653890  0.53325159 -0.180455837
sqft_basement   0.18295830  0.17825531  0.1620456   0.2028087 -0.044021832 -0.27363972  0.174179257
age_of_house   -0.02883912 -0.14601702 -0.4728284  -0.2975965 -0.076555292 -0.46688261  0.401059461
year_renovated -0.04707209 -0.06477761 -0.2231796  -0.1283638 -0.017372638 -0.23467902 -0.184199546
               sqft_above sqft_basement age_of_house year_renovated
price           0.5901773    0.18295830  -0.02883912    -0.04707209
bedrooms        0.4934554    0.17825531  -0.14601702    -0.06477761
bathrooms       0.6806572    0.16204563  -0.47282845    -0.22317960
sqft_living     0.8714273    0.20280872  -0.29759654    -0.12836377
sqft_lot        0.2886539   -0.04402183  -0.07655529    -0.01737264
floors          0.5332516   -0.27363972  -0.46688261    -0.23467902
condition      -0.1804558    0.17417926   0.40105946    -0.18419955
sqft_above      1.0000000   -0.22669580  -0.41783943    -0.16479513
sqft_basement  -0.2266958    1.00000000   0.20041597     0.06195220
age_of_house   -0.4178394    0.20041597   1.00000000     0.32181329
year_renovated -0.1647951    0.06195220   0.32181329     1.00000000
```

model2 <- lm(price~.,data=df2)

summary(model2)

```
> summary(model2)

Call:
lm(formula = price ~ ., data = df2)

Residuals:
     Min       1Q   Median       3Q      Max 
-1387360  -127151   -15104    93125  2033099 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)    -3.329e+05  2.822e+04 -11.799  < 2e-16 ***
bedrooms       -5.714e+04  4.955e+03 -11.532  < 2e-16 ***
bathrooms       6.534e+04  8.087e+03   8.080 8.23e-16 ***
sqft_living     1.975e+02  1.536e+01  12.859  < 2e-16 ***
sqft_lot       -1.037e+00  1.860e-01  -5.574 2.63e-08 ***
floors          5.283e+04  8.723e+03   6.056 1.50e-09 ***
condition       3.093e+04  6.017e+03   5.141 2.84e-07 ***
sqft_above      8.934e+01  1.674e+01   5.337 9.91e-08 ***
sqft_basement   5.922e+04  1.314e+04   4.506 6.76e-06 ***
age_of_house    2.834e+03  1.565e+02  18.110  < 2e-16 ***
year_renovated  1.601e+04  7.886e+03   2.030   0.0424 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 227600 on 4540 degrees of freedom
Multiple R-squared:  0.5399,	Adjusted R-squared:  0.5389 
F-statistic: 532.8 on 10 and 4540 DF,  p-value: < 2.2e-16
```

The residual standard error of the multiple linear regression model is 227600, and it has been computed using 4540 degrees of freedom. The multiple R-squared value for the model is 0.5399, with an adjusted R-squared value of 0.5389. The F-statistic for the model is 532.8, computed using 10 and 4540 degrees of freedom. The p-value for the F-statistic is less than 2.2e-16.

```
# 3rd Method :
outlier_treat3 <- function(x) {
  q1 <- quantile(x, probs = 0.25)
  q3 <- quantile(x, probs = 0.75)
  iqr <- q3 - q1
  UC <- q3 + 1.5 * iqr
  LC <- q1 - 1.5 * iqr
  x <- ifelse(x > UC, UC, x)
  x <- ifelse(x < LC, LC, x)
  return(x)
}
df3 <- data.frame(lapply(df, FUN = outlier_treat3))
min(df3$price)
max(df3$price)
```

```
> min(df3$price)
[1] 7800
> max(df3$price)
[1] 1154354
```

cor_matrix3 <- cor(df3)

cor_matrix3

```
> cor_matrix3 <- cor(df3)
> cor_matrix3
                    price    bedrooms  bathrooms sqft_living    sqft_lot     floors
price          1.00000000  0.36231826  0.5320875  0.69830870  0.16816193  0.2966047
bedrooms       0.36231826  1.00000000  0.5428209  0.62045231  0.21309949  0.1801038
bathrooms      0.53208752  0.54282089  1.0000000  0.74609338  0.13057540  0.5067546
sqft_living    0.69830870  0.62045231  0.7460934  1.00000000  0.36719621  0.3555830
sqft_lot       0.16816193  0.21309949  0.1305754  0.36719621  1.00000000 -0.1654969
floors         0.29660467  0.18010375  0.5067546  0.35558302 -0.16549688  1.0000000
condition      0.05722951  0.02016595 -0.1267138 -0.06323968  0.04327781 -0.2754411
sqft_above     0.60156078  0.50199269  0.6730708  0.86855012  0.34950405  0.5389669
sqft_basement  0.18864173  0.17641644  0.1592123  0.20163607 -0.03715389 -0.2736109
age_of_house  -0.04212497 -0.15274126 -0.4906996 -0.30224919 -0.06349536 -0.4666906
year_renovated -0.06072986 -0.06713978 -0.2324807 -0.13034460  0.02064419 -0.2343817
                condition sqft_above sqft_basement age_of_house year_renovated
price          0.05722951  0.6015608    0.18864173  -0.04212497    -0.06072986
bedrooms       0.02016595  0.5019927    0.17641644  -0.15274126    -0.06713978
bathrooms     -0.12671375  0.6730708    0.15921231  -0.49069963    -0.23248067
sqft_living   -0.06323968  0.8685501    0.20163607  -0.30224919    -0.13034460
sqft_lot       0.04327781  0.3495040   -0.03715389  -0.06349536     0.02064419
floors        -0.27544113  0.5389669   -0.27361090  -0.46669056    -0.23438170
condition      1.00000000 -0.1828054    0.17418417   0.40142783    -0.18440324
sqft_above    -0.18280536  1.0000000   -0.23047181  -0.42286957    -0.16721026
sqft_basement  0.17418417 -0.2304718    1.00000000   0.20041597     0.06195220
age_of_house   0.40142783 -0.4228696    0.20041597   1.00000000     0.32181329
year_renovated -0.18440324 -0.1672103    0.06195220   0.32181329     1.00000000
```

model3 <- lm(price~.,data=df3)

summary(model3)

```
> summary(model3)

Call:
lm(formula = price ~ ., data = df3)

Residuals:
     Min       1Q   Median       3Q      Max 
-1047538  -110385    -5628    98967   933613 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)    -1.737e+05  2.196e+04  -7.911 3.18e-15 ***
bedrooms       -4.322e+04  3.883e+03 -11.133  < 2e-16 ***
bathrooms       4.644e+04  6.418e+03   7.236 5.40e-13 ***
sqft_living     1.654e+02  1.177e+01  14.052  < 2e-16 ***
sqft_lot       -3.197e+00  5.684e-01  -5.624 1.97e-08 ***
floors          4.542e+04  6.988e+03   6.499 8.95e-11 ***
condition       2.637e+04  4.518e+03   5.837 5.67e-09 ***
sqft_above      7.102e+01  1.266e+01   5.611 2.14e-08 ***
sqft_basement   4.891e+04  9.751e+03   5.016 5.49e-07 ***
age_of_house    2.137e+03  1.188e+02  17.998  < 2e-16 ***
year_renovated  1.032e+04  5.912e+03   1.745   0.0811 .  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 170500 on 4540 degrees of freedom
Multiple R-squared:  0.5624,	Adjusted R-squared:  0.5614 
F-statistic: 583.5 on 10 and 4540 DF,  p-value: < 2.2e-16
```

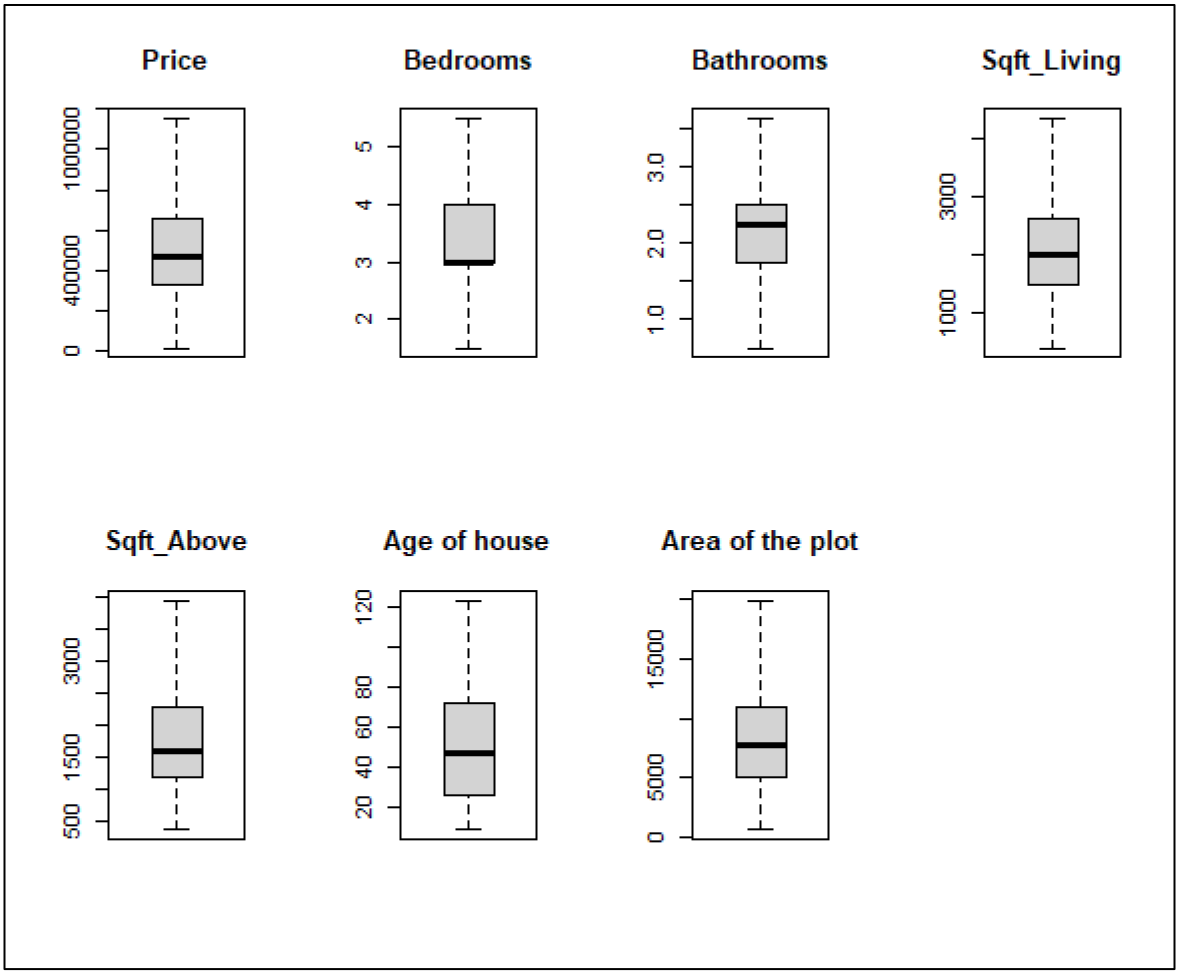
The given output shows the statistical analysis of a regression model. The Residual standard error is 170500 on 4540 degrees of freedom. The Multiple R-squared value is 0.5624, and the Adjusted R-squared value is 0.5614. The F-statistic is 583.5 on 10 and 4540 degrees of freedom, and the p-value is less than 2.2e-16.

Three different methods were used for removing outliers, and their summary statistics were compared. Among these methods, the **3rd method** has the highest R-squared value and the lowest Residual standard error. Additionally, the F-statistic is higher for the third method, and the p-values are also comparable to the other methods.

A high R-squared value indicates that more variation in the response variable is explained by the predictors. Similarly, a high F-statistic suggests a significant relationship between the predictors and the response variable. A low Residual standard error means that the third method has greater precision. Therefore, the third method is preferred for removing outliers, and the analysis will be performed on the data obtained after applying this method (df3).

Plotting boxplot to determine whether the outliers have been eliminated or not.

```
par(mfrow=c(2, 4))

boxplot(df3$price, main="Price")

boxplot(df3$bedrooms, main="Bedrooms")

boxplot(df3$bathrooms, main="Bathrooms")

boxplot(df3$sqft_living, main="Sqft_Living")

boxplot(df3$sqft_above, main="Sqft_Above")

boxplot(df3$age_of_house, main="Age of house")

boxplot(df3$sqft_lot, main="Area of the plot")
```



outliers have been removed

Visualizing the data:

par(mfrow=c(2, 3))

hist(df3$bedrooms, breaks = 5, col = "violet", main = "Histogram for no. of bedrooms", xlab = "Bedrooms")

#bedrooms with 2-3 bins have the highest frequency

hist(df3$bathrooms, breaks = 10, col = "green", main = "Histogram for no. of bathrooms", xlab = "Bathrooms")

#bathrooms with 2-2.5 bins have the highest frequency

hist(df3$price, breaks = 10, col = "red", main = "Histogram for price", xlab = "Price")

#price with 200000-600000 have the highest frequency

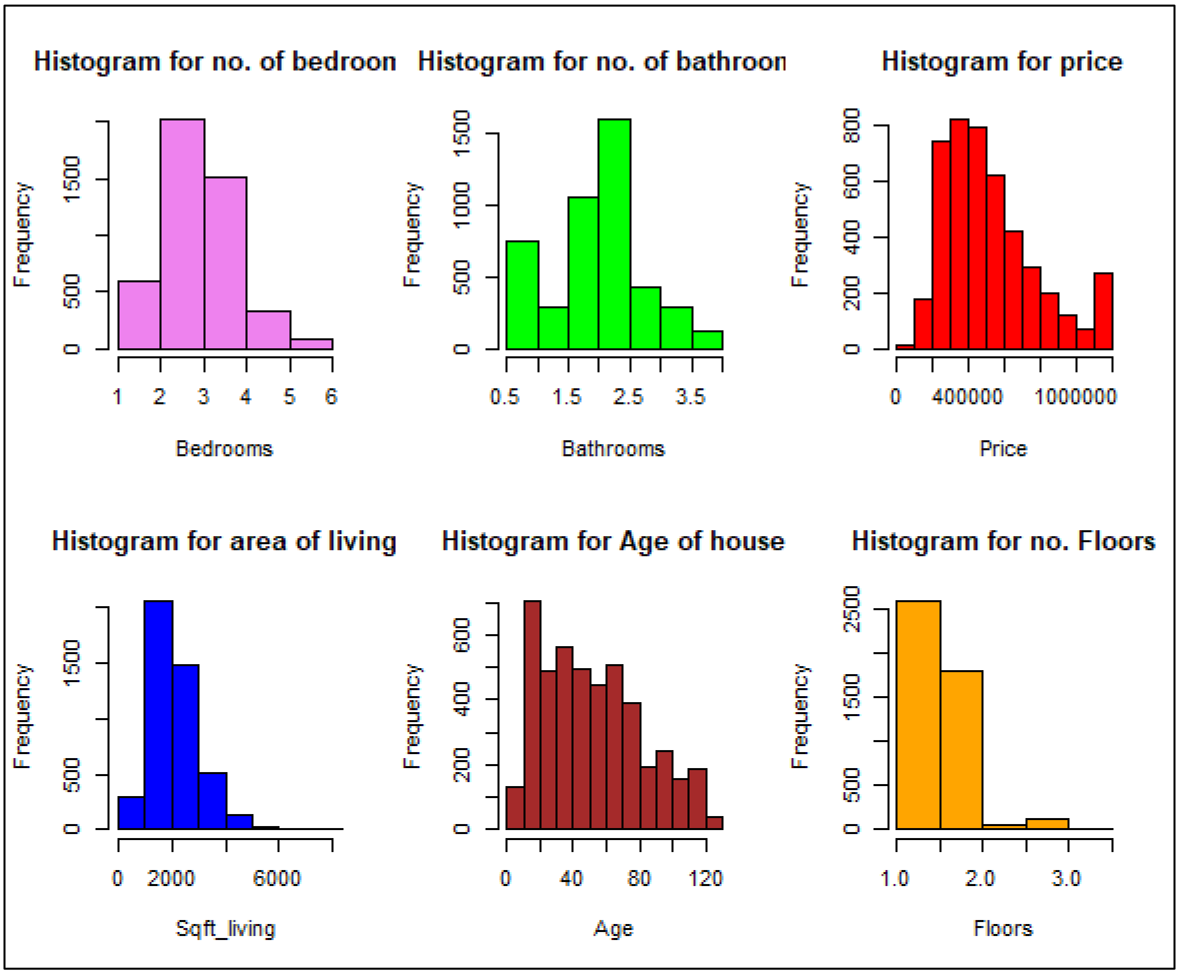
hist(df$sqft_living, breaks = 10, col = "blue", main = "Histogram for area of living", xlab = "Sqft_living",xlim = c(0,8000))

#sqft_living with 1000-2000sqft have the highest frequency

hist(df$age_of_house, breaks = 10, col = "brown", main = "Histogram for Age of house", xlab = "Age")

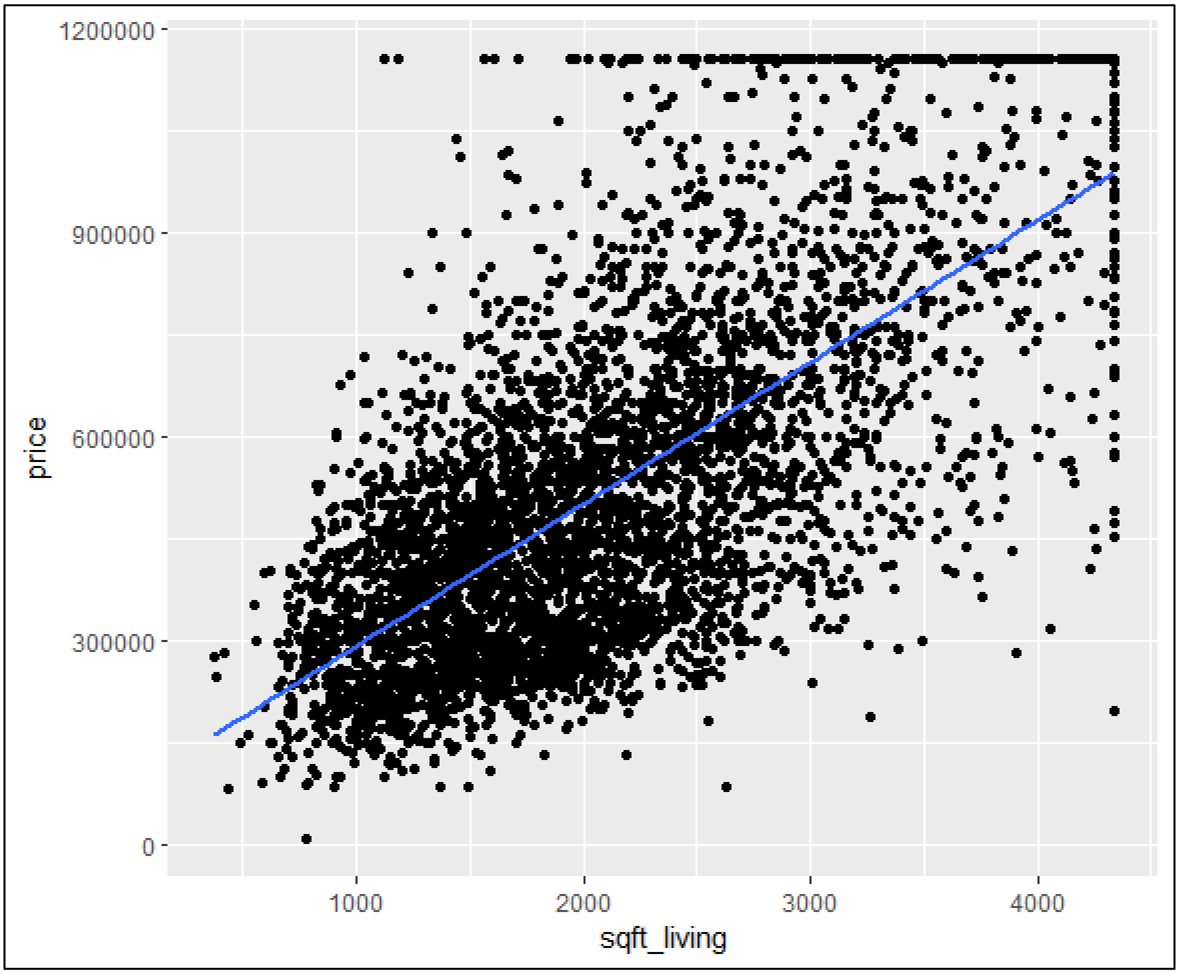
hist(df$floors, breaks = 6, col = "orange", main = "Histogram for no. Floors", xlab = "Floors")

#floors with 1-1.5 have the highest frequency



Scatter plot between price and sqft_living

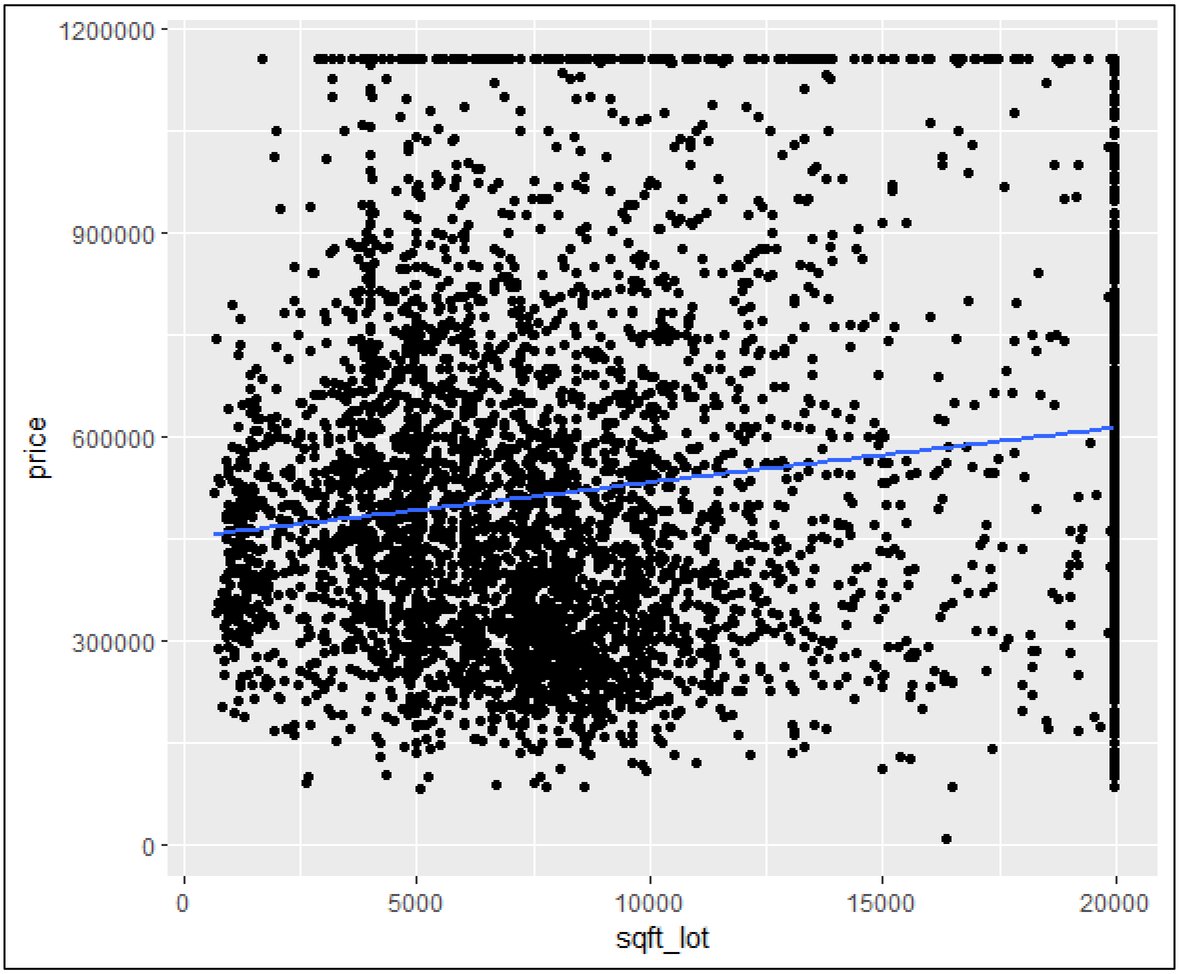
ggplot(data=df3,aes(x=sqft_living,y=price))+geom_point()+geom_smooth(method="lm",se=F)



From the plot, we can observe that there is a linear relationship between price and sqft_living. The curve is positive, so it means more area of plot used for a living(sqft_living) more the price.

Scatter plot between price and sqft_lot

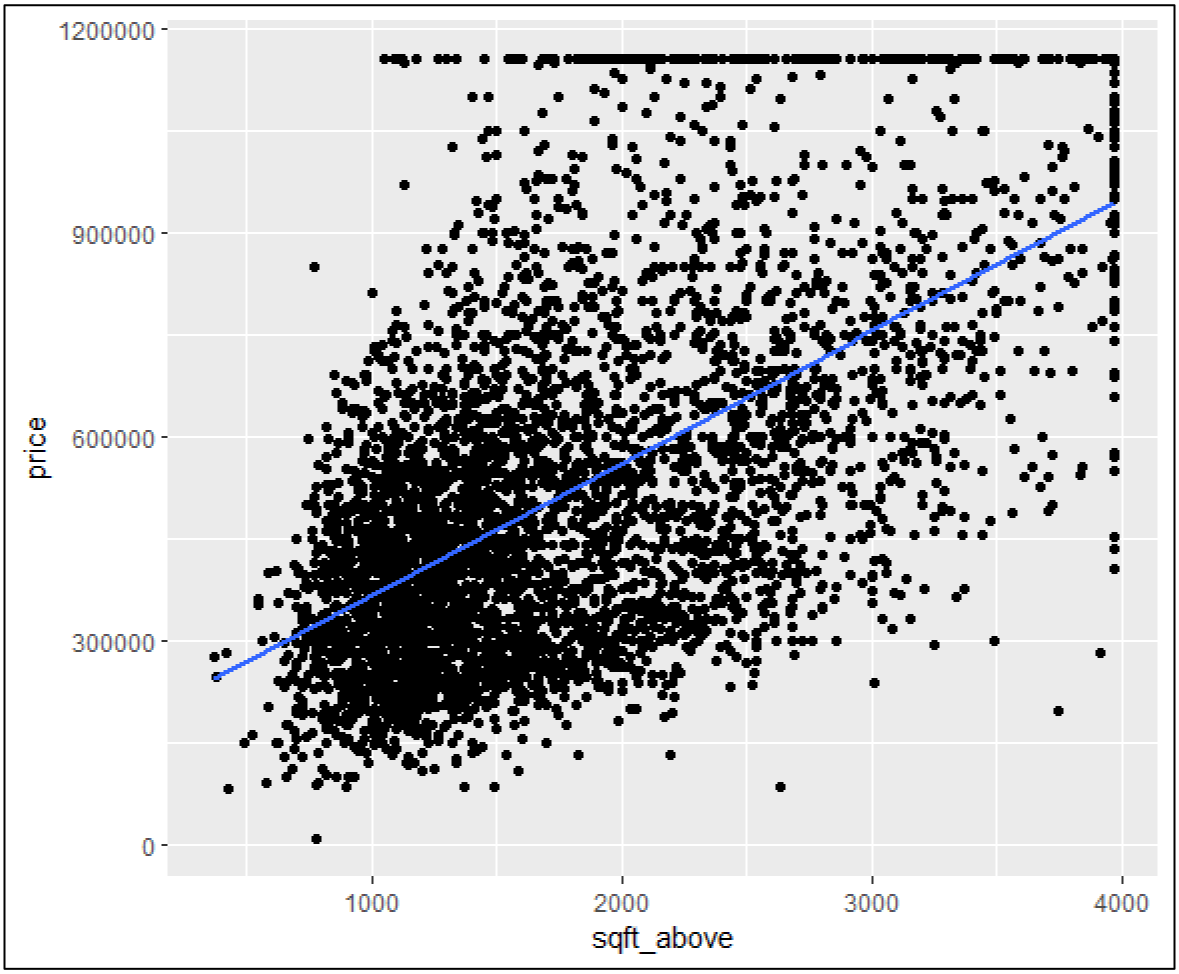
ggplot(data=df3,aes(x=sqft_lot,y=price))+geom_point()+geom_smooth(method ="lm",se=F)



From the plot, we can observe that there is a linear relationship between price and sqft_lot. The curve is positive, so it means the greater the total area of the plot(sqft_lot) more the price. But the linear relationship is less strong compared to sqft_living.

Scatter plot between price and sqft_above

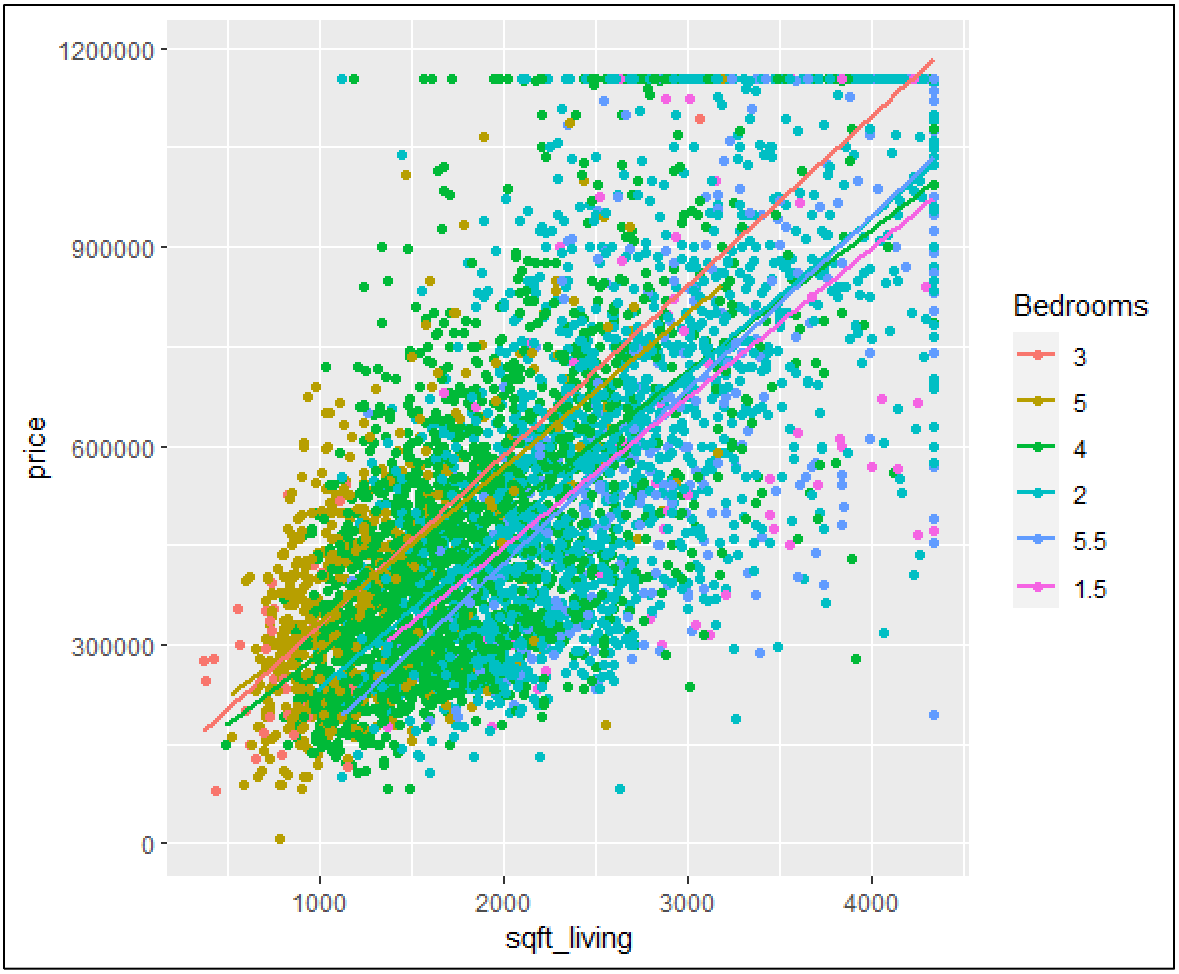
ggplot(data=df3,aes(x=sqft_above,y=price))+geom_point()+geom_smooth(method="lm",se=F)



From the plot, we can observe that there is a linear relationship between price and sqft_above. The line is positive, so it means more total area above ground(sqft_above) more the price. The linear relationship is almost as strong as compared to sqft_living.

Scatter plot between sqft_living and price

```
g <- ggplot(df3,aes(x=sqft_living,y=price,col=factor(bedrooms)))

g+geom_point() +geom_smooth(method="lm",se=F)+ labs(col="Bedrooms")  +
scale_color_discrete(labels = unique(df3$bedrooms))
```

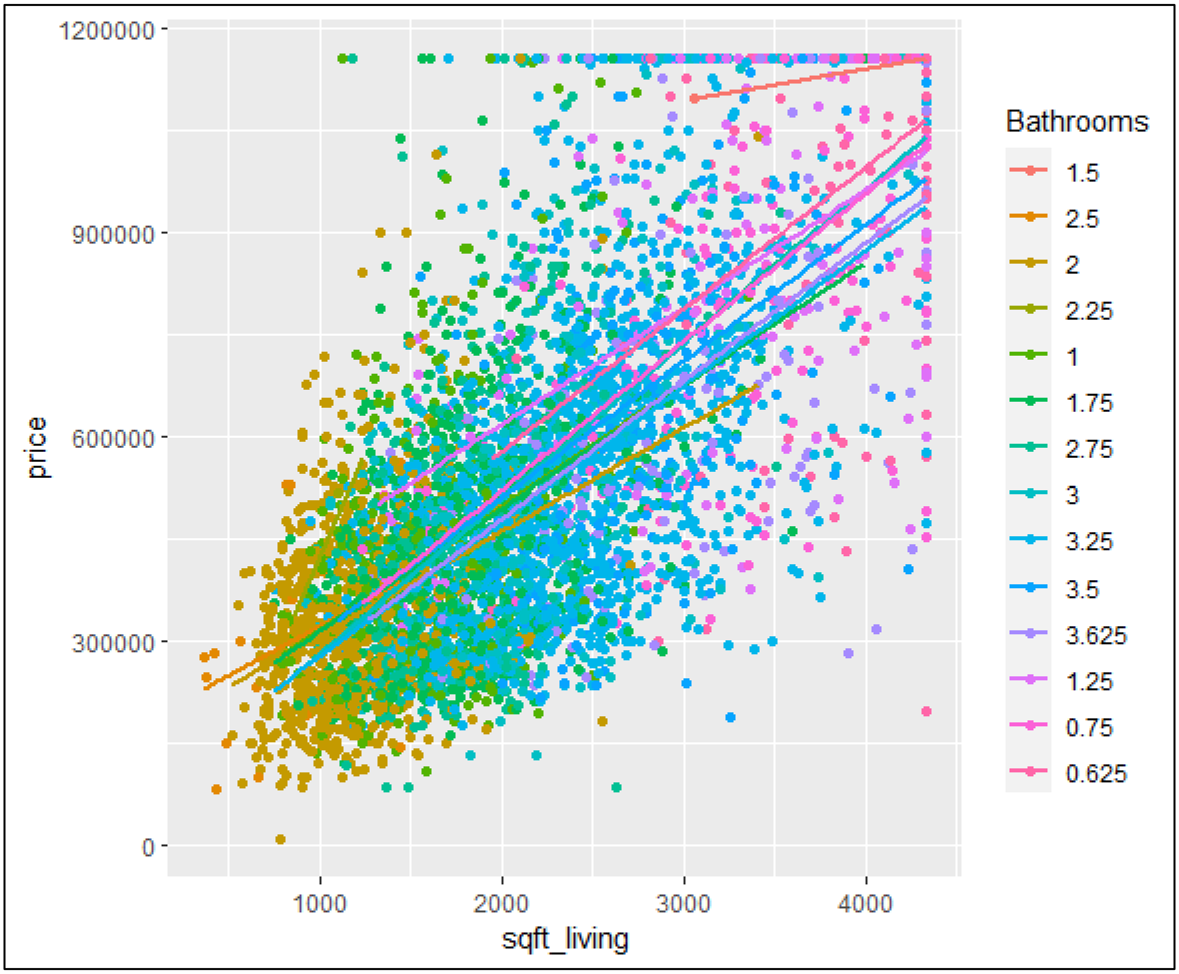


From the plot we can visually explore the relationship between `sqft_living` and `price`, and how it differs based on the number of bedrooms in the house. The relationship between `sqft_living` and `price` varies by the number of bedrooms.

Scatter plot between the price of the house and the number of bathrooms

h <- ggplot(df3,aes(x=sqft_living,y=price,col=factor(bathrooms)))

h+geom_point() +geom_smooth(method="lm",se=F)+ labs(col="Bathrooms") + scale_color_discrete(labels = unique(df3$bathrooms))



From the scatterplot, we can observe the relationship between the variables `sqft_living` and `price`. We can see how the price varies as the living area (sqft_living) increases or decreases.

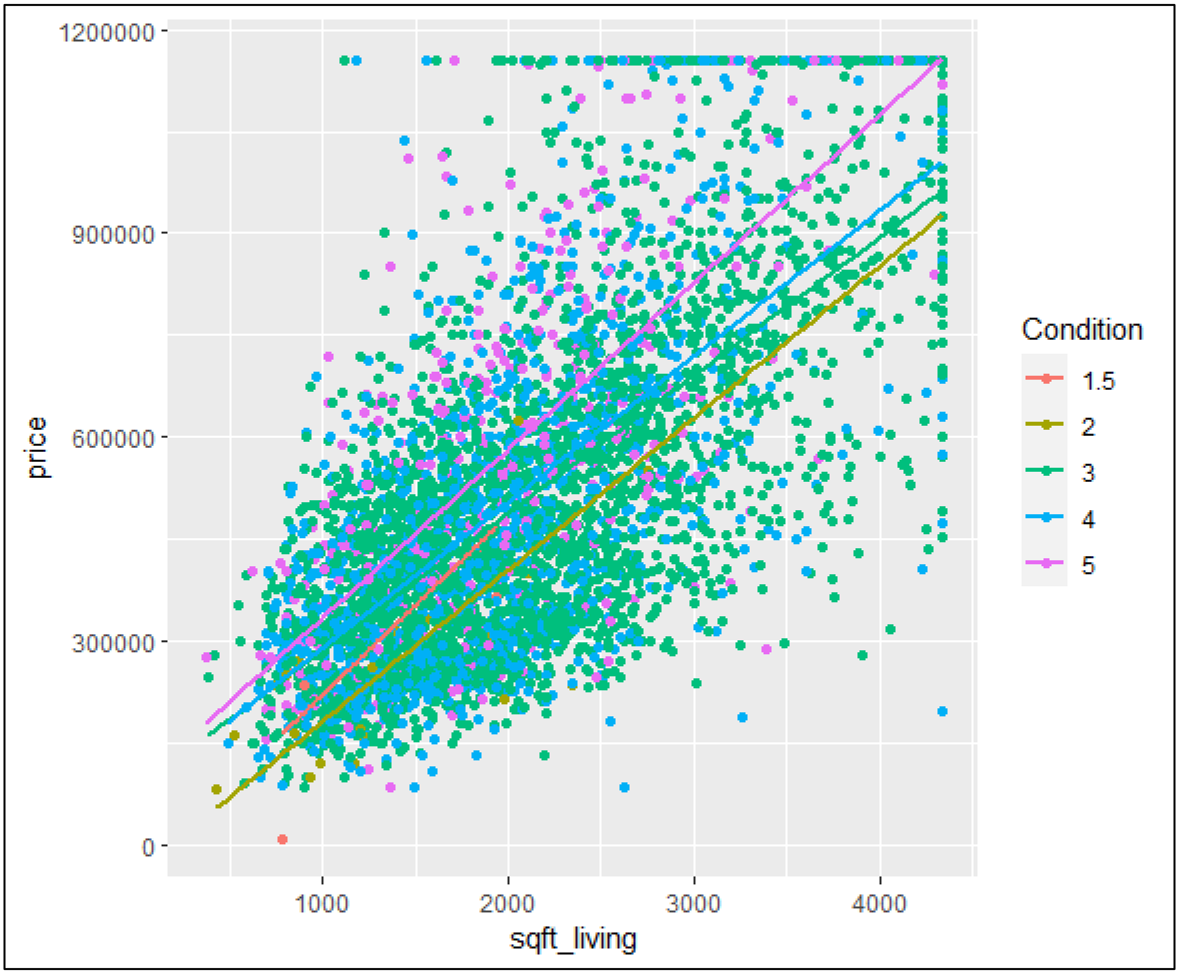
Additionally, we can observe the relationship between the number of bathrooms (`bathrooms`) and the price. The scatterplot represents the different number of bathrooms in different colors, so we can also observe how the price varies with the number of bathrooms.

Overall, the scatterplot and the multiple linear regression line provide a visual representation of the relationship between the variables, which can help us to understand the data and identify any patterns or trends.

Scatter plot between the price of house and condition of the house

i <- ggplot(df3,aes(x=sqft_living,y=price,col=factor(condition)))

i+geom_point() +geom_smooth(method="lm",se=F)+ labs(col="Condition"



From the plot, we can observe that there is a positive correlation between the square footage of living space and the price of a house, as expected.

The scatter plot shows that there is quite a bit of variability in the price of houses with similar square footage. This could be due to other factors such as location, age, or style of the house.

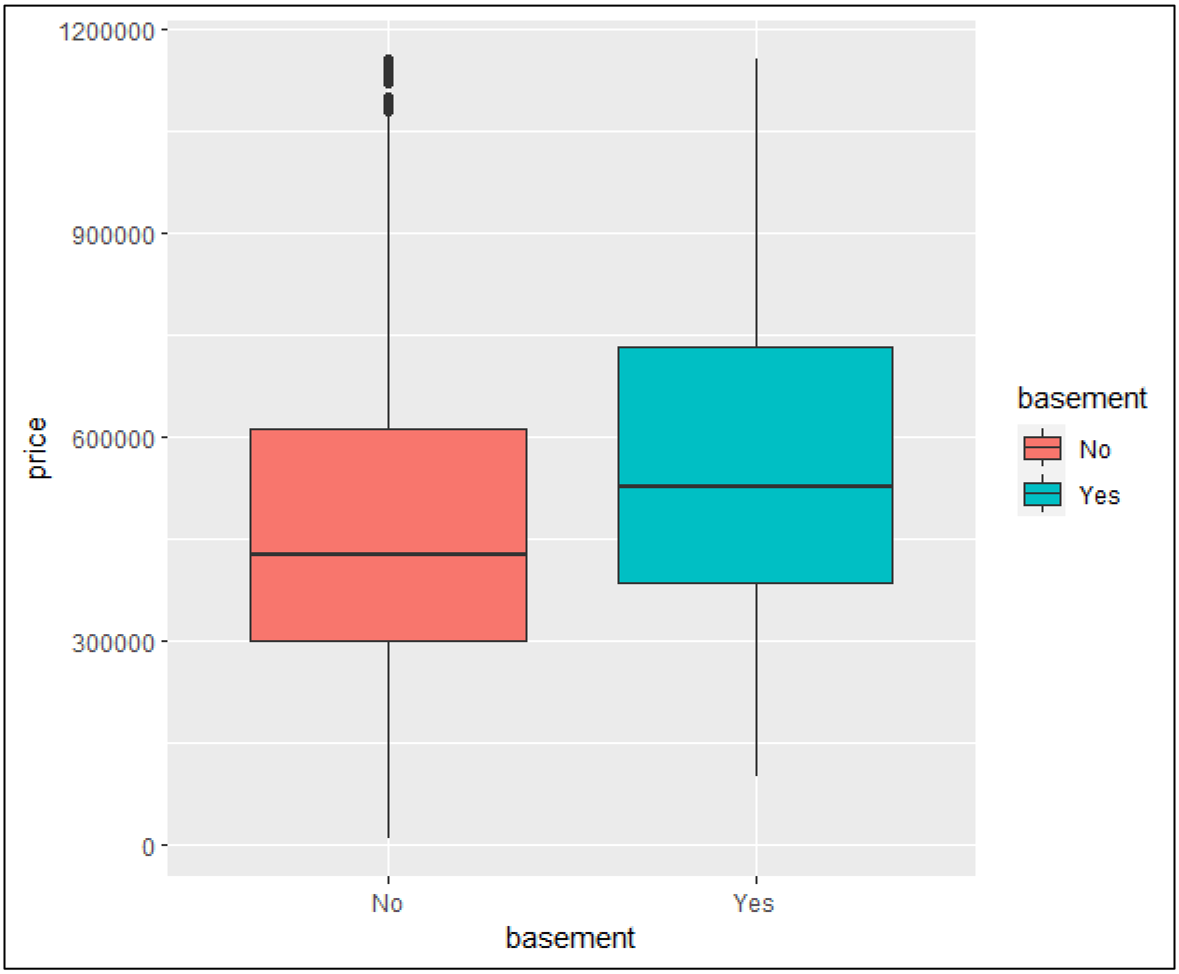
The trend line (generated by the "geom_smooth" function with method="lm") shows a positive slope, indicating a positive linear relationship between square footage and price.

The color legend shows that houses in better condition (condition=3 or 4) tend to have higher prices compared to those in worse condition (condition=1 or 2).

Boxplot for basement & price

basement<-ifelse(df$sqft_basement > 0, "Yes", "No")

ggplot(data=df3,aes(y=price,x=basement, fill=basement))+geom_boxplot()



From the box plot, we can observe that houses with a basement tend to have a higher median price than those without a basement. The box for houses with a basement is also slightly box plot suggests that having a basement may be a factor in determining the price of a house, but there are other factors that may also influence the price.

From the above graphs we can infer that all the independent variables are related to the target variable. And the trend line (generated by the "geom_smooth" function with method="lm") shows a positive slope, indicating a positive linear relationship.

Moving further we are dividing the dataset into training and testing data. We used the library caret. The createDataPartition() function from the caret package is used to randomly split your data into training and test sets based on a specified proportion. Here we took a 0.60 proportion. Based on that we built our model.

library(caret)

set.seed(123)

4551*0.6

trainIndex <- createDataPartition(df3$price, p = 0.6, list = FALSE)

training_set <- df3[trainIndex,]

test_set <- df3[-trainIndex,]

cat("No. of rows for training:", nrow(training_set), "\n")

cat("No. of rows for testing:", nrow(test_set), "\n")

```
> cat("No. of rows for training:", nrow(training_set), "\n")
No. of rows for training: 2732
> cat("No. of rows for testing:", nrow(test_set), "\n")
No. of rows for testing: 1819
```

 # We started building our model using multiple linear regression.

mod1 <- lm(price~.,data=training_set)

summary(mod1)

```
> mod1 <- lm(price~.,data=training_set)
> summary(mod1)

Call:
lm(formula = price ~ ., data = training_set)

Residuals:
     Min       1Q   Median       3Q      Max 
-1045002  -113463    -9450    99771   932760 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)    -1.730e+05  2.894e+04  -5.977 2.58e-09 ***
bedrooms       -4.428e+04  4.926e+03  -8.988  < 2e-16 ***
bathrooms       3.889e+04  8.381e+03   4.640 3.64e-06 ***
sqft_living     1.640e+02  1.522e+01  10.774  < 2e-16 ***
sqft_lot       -4.106e+00  7.477e-01  -5.491 4.36e-08 ***
floors          4.875e+04  9.157e+03   5.324 1.10e-07 ***
condition       3.028e+04  5.961e+03   5.080 4.03e-07 ***
sqft_above      7.648e+01  1.623e+01   4.713 2.57e-06 ***
sqft_basement   5.084e+04  1.273e+04   3.995 6.65e-05 ***
age_of_house    2.101e+03  1.528e+02  13.749  < 2e-16 ***
year_renovated  1.325e+04  7.594e+03   1.744   0.0812 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 171800 on 2721 degrees of freedom
Multiple R-squared:  0.5552,	Adjusted R-squared:  0.5536 
F-statistic: 339.6 on 10 and 2721 DF,  p-value: < 2.2e-16
```

 # We are checking the pairwise correlation coefficients between all columns.

cor(df3)

```
> cor(df3)
                    price    bedrooms  bathrooms sqft_living     sqft_lot     floors   condition sqft_above sqft_basement age_of_house
price          1.00000000  0.36231826  0.5320875  0.69830870  0.16816193  0.2966047  0.05722951  0.6015608    0.18864173  -0.04212497
bedrooms       0.36231826  1.00000000  0.5428209  0.62045231  0.21309949  0.1801038  0.02016595  0.5019927    0.17641644  -0.15274126
bathrooms      0.53208752  0.54282089  1.0000000  0.74609338  0.13057540  0.5067546 -0.12671375  0.6730708    0.15921231  -0.49069963
sqft_living    0.69830870  0.62045231  0.7460934  1.00000000  0.36719621  0.3555830 -0.06323968  0.8685501    0.20163607  -0.30224919
sqft_lot       0.16816193  0.21309949  0.1305754  0.36719621  1.00000000 -0.1654969  0.04327781  0.3495040   -0.03715389  -0.06349536
floors         0.29660467  0.18010375  0.5067546  0.35558302 -0.16549688  1.0000000 -0.27544113  0.5389669   -0.27361090  -0.46669056
condition      0.05722951  0.02016595 -0.1267138 -0.06323968  0.04327781 -0.2754411  1.00000000 -0.1828054    0.17418417   0.40142783
sqft_above     0.60156078  0.50199269  0.6730708  0.86855012  0.34950405  0.5389669 -0.18280536  1.0000000   -0.23047181  -0.42286957
sqft_basement  0.18864173  0.17641644  0.1592123  0.20163607 -0.03715389 -0.2736109  0.17418417 -0.2304718    1.00000000   0.20041597
age_of_house  -0.04212497 -0.15274126 -0.4906996 -0.30224919 -0.06349536 -0.4666906  0.40142783 -0.4228696    0.20041597   1.00000000
year_renovated -0.06072986 -0.06713978 -0.2324807 -0.13034460  0.02064419 -0.2343817 -0.18440324 -0.1672103    0.06195220   0.32181329
              year_renovated
price            -0.06072986
bedrooms         -0.06713978
bathrooms        -0.23248067
sqft_living      -0.13034460
sqft_lot          0.02064419
floors           -0.23438170
condition        -0.18440324
sqft_above       -0.16721026
sqft_basement     0.06195220
age_of_house      0.32181329
year_renovated    1.00000000
```

Next, we observed the correlation coefficient between price and all other variables.

As observed, the correlation coefficient between year_renovated and the price column & the condition of the house with the price is very low. So, we removed these columns and checked the model performance.

 # Moving forward we have to see the correlation coefficient between variables.

 # And develop models by removing columns and check model performance

mod2 <- lm(price~. -year_renovated, data=training_set)

summary(mod2)

mod3 <- lm(price~. -year_renovated -age_of_house, data=training_set)

summary(mod3)

```
> mod2 <- lm(price~. -year_renovated,data=training_set)
> summary(mod2)

Call:
lm(formula = price ~ . - year_renovated, data = training_set)

Residuals:
     Min       1Q   Median       3Q      Max 
-1038094  -112919    -9005    99028   937566 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -1.559e+05  2.726e+04  -5.721 1.17e-08 ***
bedrooms     -4.406e+04  4.927e+03  -8.942  < 2e-16 ***
bathrooms     3.831e+04  8.378e+03   4.572 5.04e-06 ***
sqft_living   1.645e+02  1.522e+01  10.809  < 2e-16 ***
sqft_lot     -4.064e+00  7.476e-01  -5.436 5.94e-08 ***
floors        4.696e+04  9.103e+03   5.159 2.66e-07 ***
condition     2.639e+04  5.530e+03   4.772 1.92e-06 ***
sqft_above    7.613e+01  1.623e+01   4.690 2.87e-06 ***
sqft_basement 5.056e+04  1.273e+04   3.971 7.33e-05 ***
age_of_house  2.183e+03  1.455e+02  15.004  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 171900 on 2722 degrees of freedom
Multiple R-squared:  0.5547,	Adjusted R-squared:  0.5532 
F-statistic: 376.8 on 9 and 2722 DF,  p-value: < 2.2e-16
```

```
> mod3 <- lm(price~. -year_renovated -age_of_house,data=training_set)
> summary(mod3)

Call:
lm(formula = price ~ . - year_renovated - age_of_house, data = training_set)

Residuals:
    Min      1Q  Median      3Q     Max 
-876034 -121834   -7988  101302  925943 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)   -4.933e+04  2.738e+04  -1.802 0.071697 .  
bedrooms      -3.897e+04  5.113e+03  -7.622 3.42e-14 ***
bathrooms     -7.954e+03  8.104e+03  -0.981 0.326467    
sqft_living    1.748e+02  1.582e+01  11.052  < 2e-16 ***
sqft_lot      -4.994e+00  7.751e-01  -6.444 1.37e-10 ***
floors         3.149e+04  9.409e+03   3.347 0.000827 ***
condition      5.459e+04  5.411e+03  10.089  < 2e-16 ***
sqft_above     7.287e+01  1.689e+01   4.315 1.65e-05 ***
sqft_basement  6.983e+04  1.318e+04   5.300 1.25e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 178800 on 2723 degrees of freedom
Multiple R-squared:  0.5179,	Adjusted R-squared:  0.5165 
F-statistic: 365.6 on 8 and 2723 DF,  p-value: < 2.2e-16
```

If we remove year_renovated and condition columns there is not much difference in the model performance. So, we removed these columns from the model.

But if we remove the age_of_house column there is a difference in model performance. So, we are not removing age_of_house.

Our final model will be on all predictors except year_renovated and condition of the house. By removing these columns there is almost the same model performance.

Now we develop our final model on the training_set and observe summary statistics of the model.

 # Final Model on training_set

mod_final <- lm(price~. -year_renovated -condition, data=training_set)

summary(mod_final)

```
> mod_final <- lm(price~. -year_renovated -condition,data=training_set)
> summary(mod_final)

Call:
lm(formula = price ~ . - year_renovated - condition, data = training_set)

Residuals:
     Min       1Q   Median       3Q      Max 
-1035680  -113309    -9421    98484   923494 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -7.900e+04  2.206e+04  -3.580 0.000349 ***
bedrooms     -4.292e+04  4.941e+03  -8.688  < 2e-16 ***
bathrooms     4.167e+04  8.381e+03   4.972 7.03e-07 ***
sqft_living   1.699e+02  1.524e+01  11.148  < 2e-16 ***
sqft_lot     -3.962e+00  7.503e-01  -5.280 1.39e-07 ***
floors        4.309e+04  9.103e+03   4.733 2.32e-06 ***
sqft_above    6.835e+01  1.621e+01   4.215 2.57e-05 ***
sqft_basement 4.712e+04  1.276e+04   3.692 0.000227 ***
age_of_house  2.419e+03  1.374e+02  17.607  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 172600 on 2723 degrees of freedom
Multiple R-squared:  0.551,	Adjusted R-squared:  0.5497 
F-statistic: 417.7 on 8 and 2723 DF,  p-value: < 2.2e-16
```

The given output shows the statistical analysis of a regression model. The Residual standard error is 172600 on 2723 degrees of freedom. The Multiple R-squared value is 0.551, and the Adjusted R-squared value is 0.5497. The F-statistic is 417.7 on 8 and 2723 degrees of freedom, and the p-value is less than 2.2e-16.

A high R-squared value indicates that more variation in the response variable is explained by the predictors. Similarly, a high F-statistic suggests a significant relationship between the predictors and the response variable. A low Residual standard error means that the third method has greater precision. Therefore, the third method is preferred for removing outliers, and the analysis will be performed on the data obtained after applying this method (df3).

From the summary of the model we can see, all the t-values are greater than 2 and their corresponding p-values are very small (less than 0.05), indicating that all the coefficients are statistically significant at the 5% level of significance. So, we can conclude that all the predictor variables have a statistically significant relationship with the response variable.

 # Predicting the model with test data set

 # creating data frame with columns actual value, predicted value and error value

test <- predict(mod_final, test_set)

result_diff <- cbind(actual=test_set$price, predicted=test)

result_diff <- as.data.frame(result_diff)

error <- result_diff$actual-result_diff$predicted

error <- as.data.frame(error)

final_result <- cbind(result_diff,error)

final_result

```
> test <- predict(mod_final,test_set)
> result_diff <- cbind(actual=test_set$price,predicted=test)
> result_diff <- as.data.frame(result_diff)
> error <- result_diff$actual-result_diff$predicted
> error <- as.data.frame(error)
> final_result <- cbind(result_diff,error)
> final_result
    actual predicted        error
1   313000  371765.0  -58764.98504
7   335000  386762.1  -51762.10517
8   482000  588509.3 -106509.26480
12 1154354  851792.9  302560.67126
15 1154354  744529.2  409824.40855
17  419000  407397.2   11602.77211
19  257950  298505.9  -40555.88873
20  275000  234994.3   40005.68277
26  285000  478799.5 -193799.46321
27  615000  601802.1   13197.92608
31  382500  337972.5   44527.47718
32  499950  655042.4 -155092.42795
39  403000  519302.5 -116302.52214
42  260000  312694.0  -52694.00569
44  439950  472700.3  -32750.30271
45  235000  349960.7 -114960.71559
47  437500  473755.2  -36255.18916
```

We have created data frame with columns actual value, predicted value and the error value.

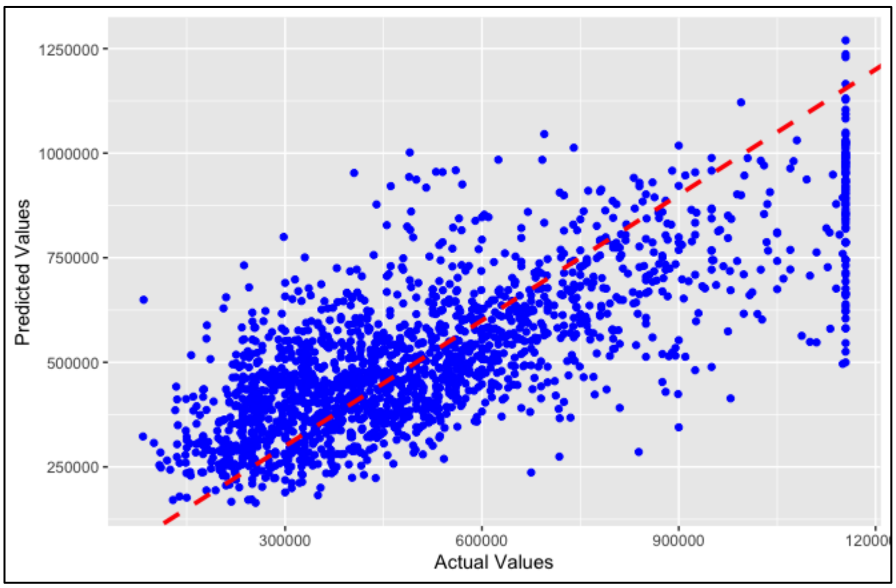
We got the error value from subtracting actual value and predicted value.

Next, we created final_result data frame with columns actual, predicted and error values to create scatter plot between actual and predicted values.

Scatterplot between actual and predicted values

k <- ggplot(data=test_set, aes(y=test, x=price))

k+geom_point(colour='blue')+labs(y='Predicted Values', x='Actual Values') + geom_abline(intercept = 0, slope = 1, color = "red", linetype ="dashed",size=1.2)



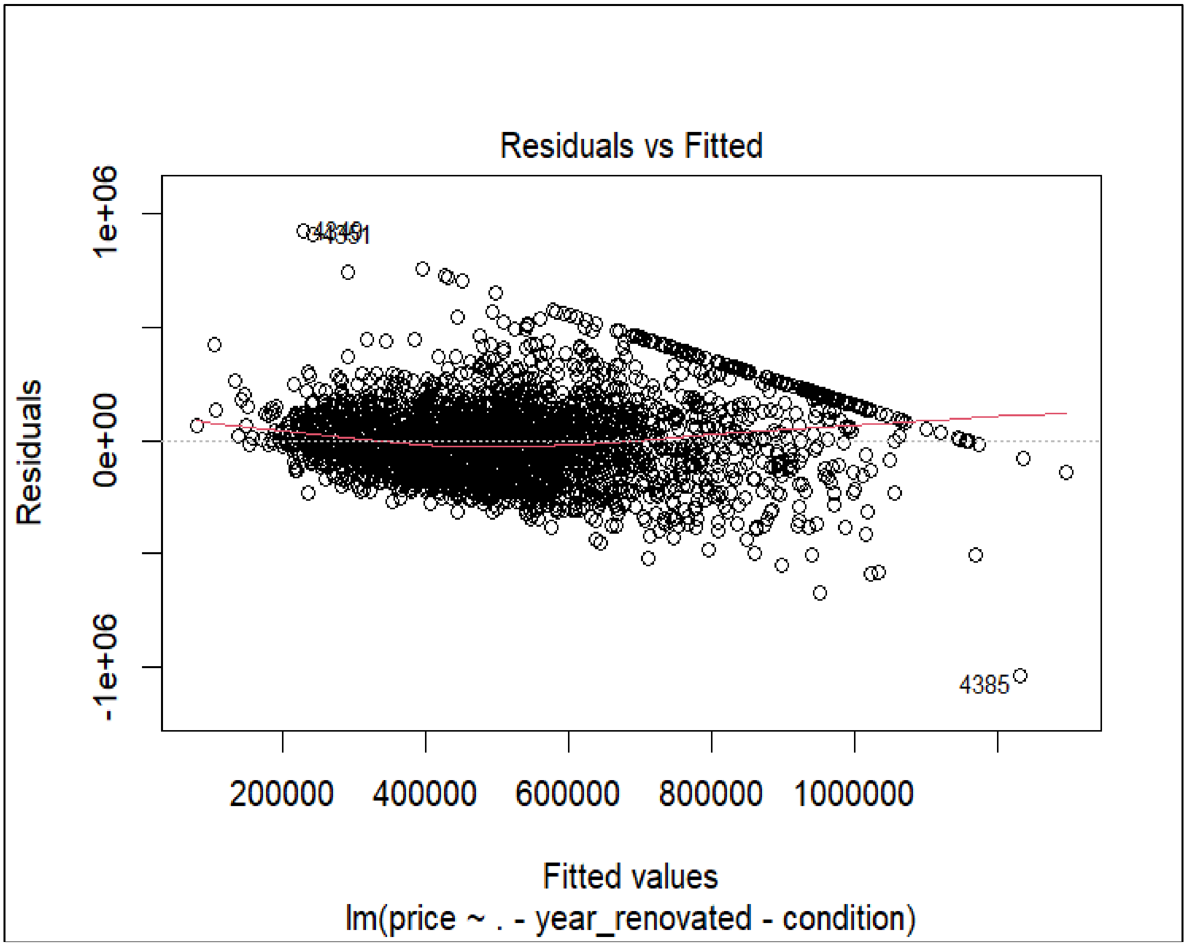
From the scatterplot, we can observe that points are tightly clustered around the diagonal line.

We can infer that predicted values are very close to the actual values.

From this, we can say the model has high accuracy and there is clearly linear relationship between variables.

Residual Plot for Final Model

plot(mod_final,which=1)



- From the residual plot, we can observe that all points clustered around the 0 line and there is no specific pattern.
- We can infer that model is the model is making accurate predictions on average, as the residuals (i.e., the differences between the observed values and the predicted values) are close to zero for most of the data points.
- Clearly, we can see there is no correlation between residuals and fitted values and curve is relatively flat indicating model is working fine.

- The residuals are close to zero for most of the data points means model is unbiased and model is not consistently underestimating or overestimating the data.

- The absence of a specific pattern in the residuals (i.e., no discernible shape or trend in the points) indicates that the model is capturing the underlying patterns in the data, and that there are no obvious outliers or influential observations that are affecting the model's performance.
- The flatness of the curve is a good indication that the model is working fine and that there is no evidence of a non-linear relationship between the predictor variables and the response variable that the model is failing to capture.

 # After evaluating the model, we can infer from the coefficients of the model for each feature:

coef_matrix <- summary(mod_final)$coefficients

model_coefficients <- coef_matrix[, 1]

as.data.frame(model_coefficients)

```
> coef_matrix <- summary(mod_final)$coefficients
> model_coefficients <- coef_matrix[ , 1]
> as.data.frame(model_coefficients)
              model_coefficients
(Intercept)        -79001.013891
bedrooms           -42924.950011
bathrooms           41672.689299
sqft_living           169.882051
sqft_lot               -3.961604
floors              43087.086566
sqft_above             68.351123
sqft_basement       47120.433093
age_of_house         2419.307973
>
```

We have created the model_coefficients data frame to find coefficient of each variable in the final multiple linear regression equation and see which feature has highest impact on price.

- Holding all other features fixed, an increase of one bedroom is associated with a decrease of $42924.95 in price.
- Holding all other features fixed, an increase of one bathroom is associated with an increase of $41672.69 in price.
- Holding all other features fixed, an increase of one sq feet area for living is associated with an increase of $169.88 in price.
- Holding all other features fixed, an increase of one sq feet area for the lot is associated with a decrease of $3.96 in price.
- Holding all other features fixed, an increase of one floor is associated with an increase of $43087.09 in price.
- Holding all other features fixed, an increase of one sq feet area above ground is associated with an increase of $68.35 in price.
- Holding all other features fixed, a house having a basement is associated with an increase of $47120.43 in price.
- Holding all other features fixed, an increase in age of house by 1 year is associated with an increase of $2419.31 in price.

Out of all the features, the feature that has the highest impact on the price of a house is house having basement, so basement feature.

After we have all the coefficients and intercept, we created the final multiple linear regression equation from all the coefficients and intercept of the model.

So final Multiple Linear regression equation:

House_Price_Prediction = -79001.01 - (42924.95)X1+ (41672.69)X2+

(169.88)X3- (3.96)X4+ (43087.09)X5+ (68.35)X6+ (47120.43)X7+ (2419.31)*X8

 # X1 = Number of Bedrooms

 # X2 = Number of Bathrooms

 # X3 = Square feet area of living

 # X4 = Square feet area of lot

 # X5 = Number of floors

 # X6 = Square feet are above ground

 # X7 = A house having basement

 # X8 = Age of the House

Now predicting house price for the sample data:

 # No. of bedrooms = 3

 # No. of bathrooms = 2

 # Area of living(sq feet) = 1250 sq feet

 # Area of lot(sq feet)= 2400 sq feet

 # No. of floors = 2

 # Area above ground(sq feet) = 1200 sq feet

 # Basement = Yes

 # Age of the house = 35 years old

predicted_price = -79001.01 - (42924.95)*(3)+ (41672.69)*(2)+ (169.88)*(1250)-(3.96)*(2400)+ (43087.09)*(2)+ (68.35)*(1200)+ (47120.43)*(1)+ (2419.31)*(35)

```
> cat("Predicted price of the house: $", formatC(predicted_price, digits = 0, format = "f", big.mark = ","), sep = "")
Predicted price of the house: $378,406
```

So, after substituting all the sample data in the multiple linear regression equation, we got the predicted price of the house as $378,406.

<u>CONCLUSION</u>:

We have successfully built a predictive model for house prices using multiple linear regression. The model can be used by real estate industry professionals as well as by people who are looking to buy or sell a house to get an estimate of the house price based on its attributes.

<u>References</u>:

Kaggle Website

https://www.kaggle.com/code/;muskanbhasin/house-price-prediction/input