House Price Prediction

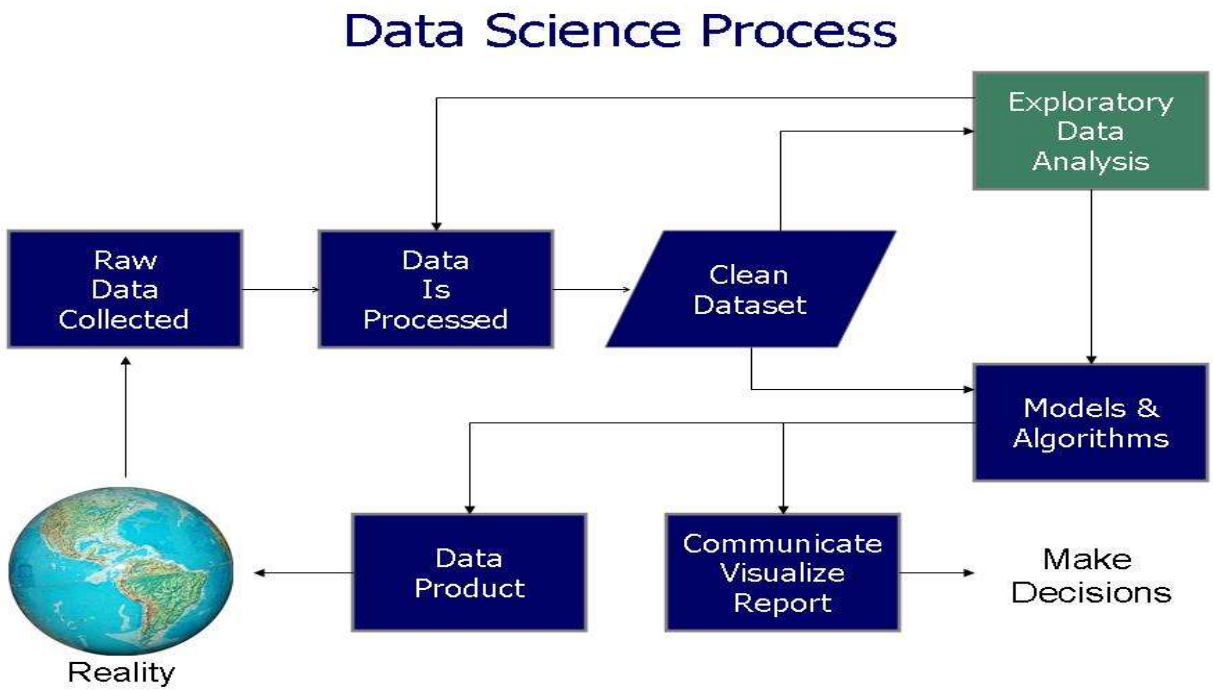
"MULTIPLE LINEAR REGRESSION, TIME SERIES FORECASTING, HYPOTHESIS TESTING AND ANOVA"

Shephalika Shekhar

Table of Contents:

1. Introduction:

If we ask a person going to buy a house to describe their dream house, then there would be many factors they would tell such as the number the bedrooms, condition and grade of the house, number of bathrooms etc. But there are many more factors involved for predicting the price of houses. A person won't probably begin with the height of the basement ceiling or the proximity to the roads. Here the dataset consists of many factors that can be useful for predicting house prices. There are many steps involved to make accurate predictions on the dataset which includes data preprocessing, exploratory data analysis then model building, evaluating the model based on their accuracy and then make predictions.



2. Data

The data set consists of many numerical as well as categorical variables that might have an affect on house price prediction which can be determined by model building using Multiple Linear Regression. Also, there is “date” variable through which we can also use time series analysis and forecasting methods to see if we can predict price in the future based on the past data.

Data has been taken from: https://www.kaggle.com/harlfoxem/housesalesprediction
The dataset consists of house sale prices (21000 rows) with many factors associated to it.

Raw data was collected, and it consists on many variables – Dependent variable: Price

Independent Variables that affect price:

1. Id: numeric
2. Date: converted to numeric
3. Bedrooms: categorical
4. Bathrooms: categorical
5. Sqft_living: numeric
6. Sqft_lot: numeric
7. Floors: categorical
8. Waterfront: categorical
9. View: categorical
10. Condition: categorical
11. Grade: categorical
12. Sqft_above: numeric
13. Sqft_basement: numeric
14. Yr_built: numeric
15. Yr_renovated: numeric
16. Zipcode: categorical
17. Lat: numeric
18. Long: numeric
19. Sqft_living15: numeric
20. Sqft_lot15: numeric

Below screenshot shows the data types of the variables of the data set.

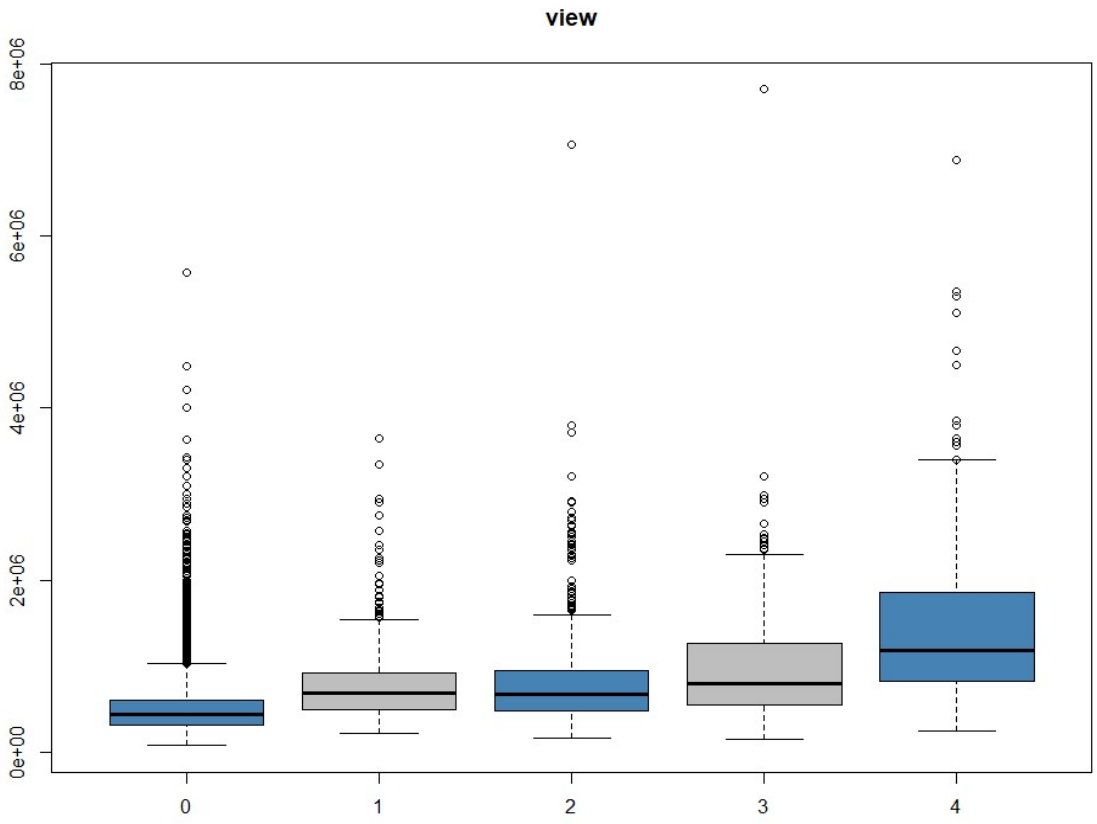
```
> str(housedata)
'data.frame':   21613 obs. of  21 variables:
 $ id           : num  7.13e+09 6.41e+09 5.63e+09 2.49e+09 1.95e+09 ...
 $ date         : Factor w/ 372 levels "20140502T000000",..: 165 221 291 221 284 11 57 252 340 306 ...
 $ price        : num  221900 538000 180000 604000 510000 ...
 $ bedrooms     : int  3 3 2 4 3 4 3 3 3 3 ...
 $ bathrooms    : num  1 2.25 1 3 2 4.5 2.25 1.5 1 2.5 ...
 $ sqft_living  : int  1180 2570 770 1960 1680 5420 1715 1060 1780 1890 ...
 $ sqft_lot     : int  5650 7242 10000 5000 8080 101930 6819 9711 7470 6560 ...
 $ floors       : num  1 2 1 1 1 1 2 1 1 2 ...
 $ waterfront   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ view         : int  0 0 0 0 0 0 0 0 0 0 ...
 $ condition    : int  3 3 3 5 3 3 3 3 3 3 ...
 $ grade        : int  7 7 6 7 8 11 7 7 7 7 ...
 $ sqft_above   : int  1180 2170 770 1050 1680 3890 1715 1060 1050 1890 ...
 $ sqft_basement: int  0 400 0 910 0 1530 0 0 730 0 ...
 $ yr_built     : int  1955 1951 1933 1965 1987 2001 1995 1963 1960 2003 ...
 $ yr_renovated : int  0 1991 0 0 0 0 0 0 0 0 ...
 $ zipcode      : int  98178 98125 98028 98136 98074 98053 98003 98198 98146 98038 ...
 $ lat          : num  47.5 47.7 47.7 47.5 47.6 ...
 $ long         : num  -122 -122 -122 -122 -122 ...
 $ sqft_living15: int  1340 1690 2720 1360 1800 4760 2238 1650 1780 2390 ...
 $ sqft_lot15   : int  5650 7639 8062 5000 7503 101930 6819 9711 8113 7570 ...
>
```

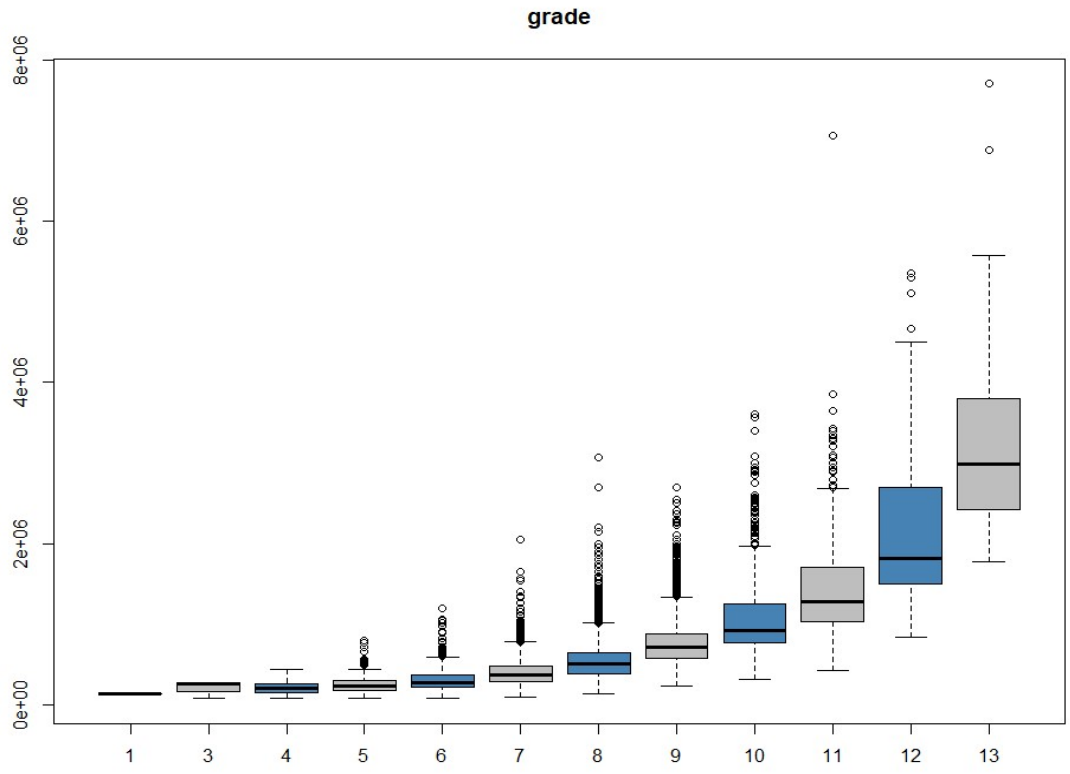
3. Problems to be Solved

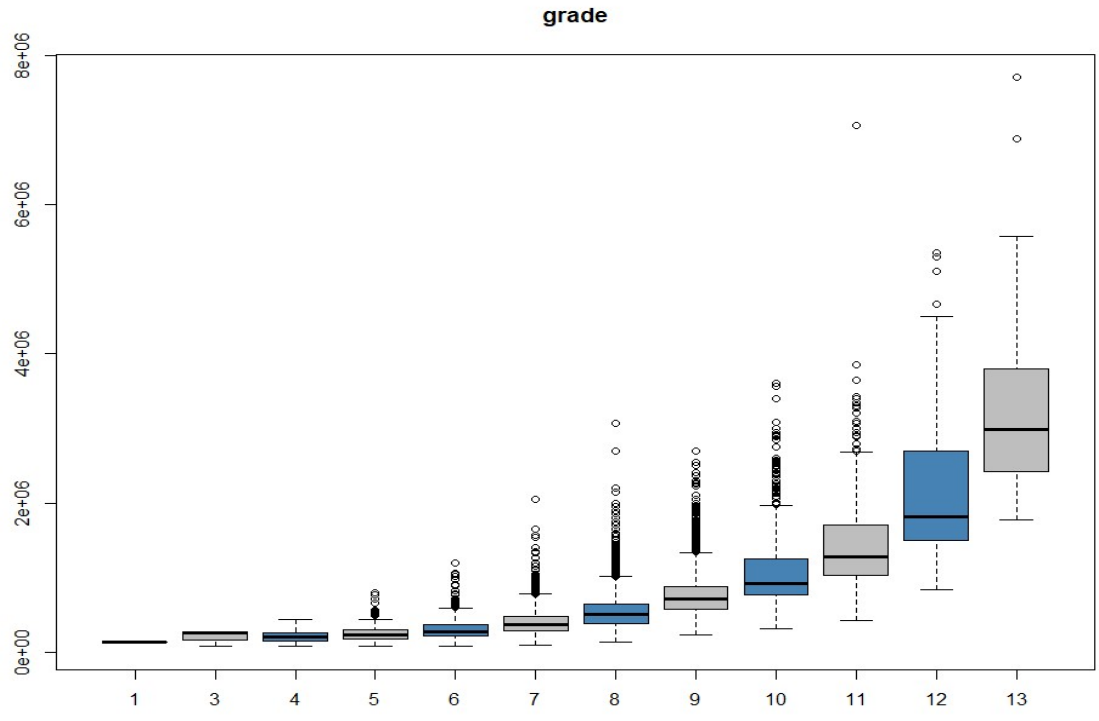
The goal is to determine which factors that influence the house price the most and may be useful to predict the house sales. Also, to see if the house sales depends on the past data based on the date variable that is present in the dataset. These problems are to be solved by building various models and applying algorithms and analysis to determine the best factors and future values of the price.

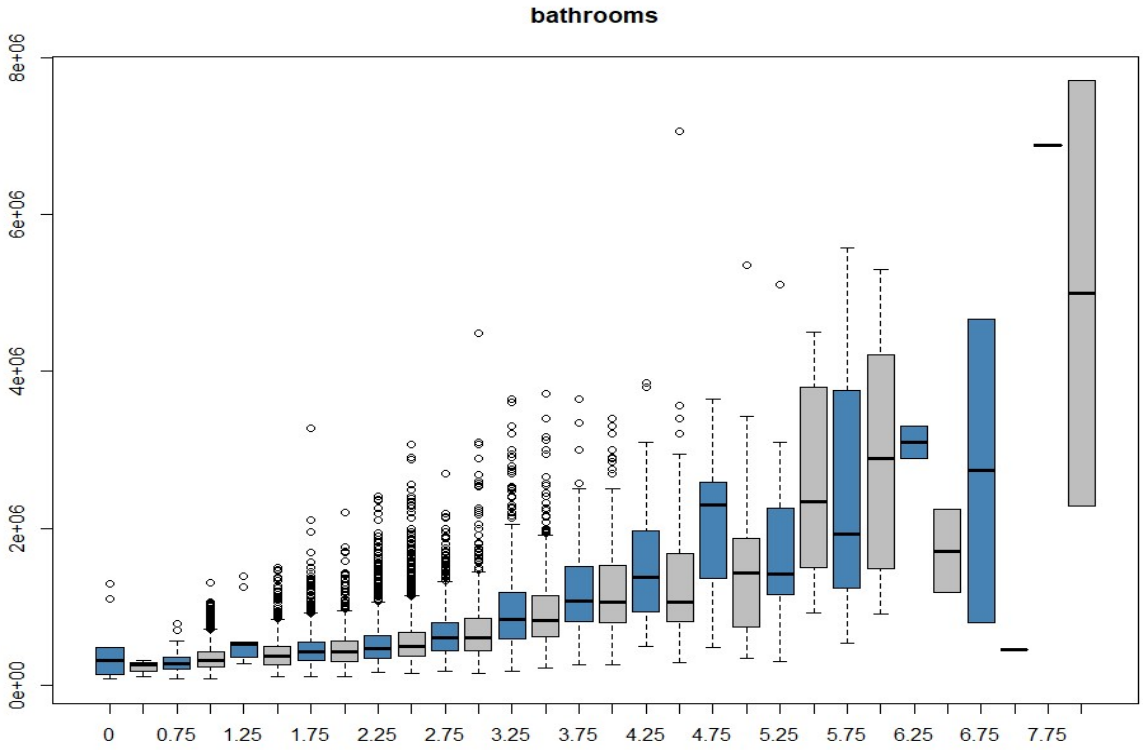
4. Data Processing

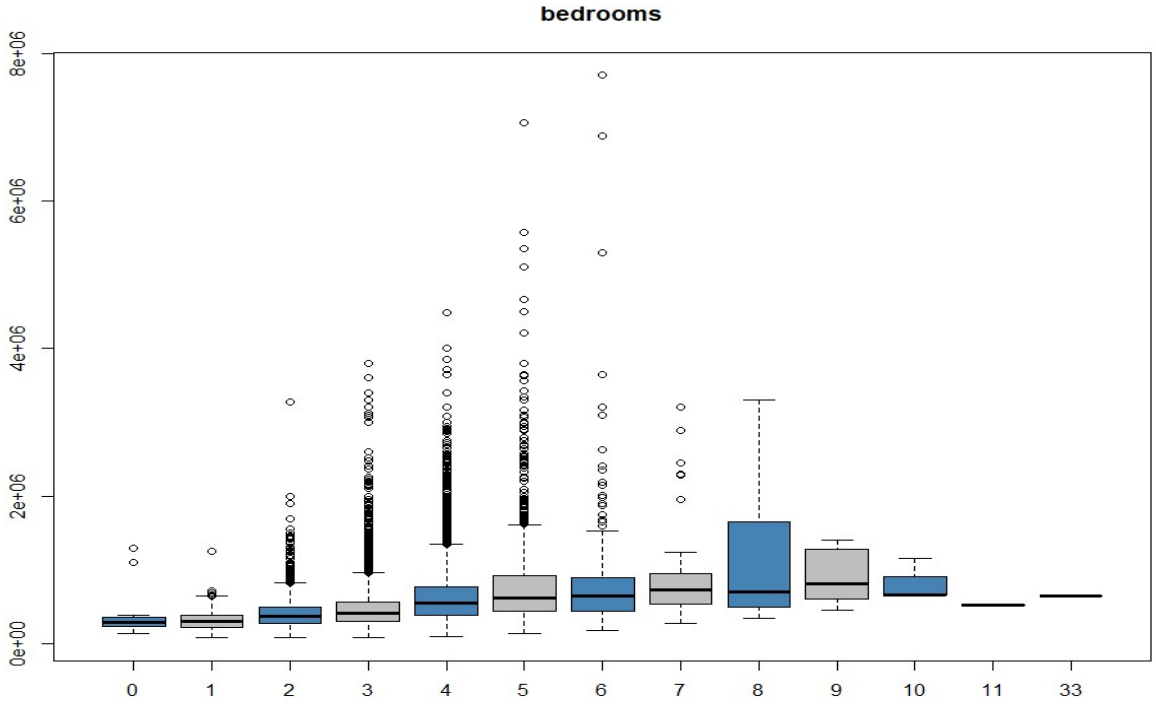
There we no missing values in the data. Bedrooms variable had an outlier which has been rectified before building the model. The dataset has been divided into training and testing data set. Date was converted to numeric variable for multiple linear regression while Date was converted to date format as required for time series analysis. Below are the box plots for independent variables vs Price which show that these are categorical variables and how they vary with price.




grade
0e+00
2e+06
4e+06
6e+06
8e+06
1
3
4
5
6
7
8
9
10
11
12
13


grade
0e+00
2e+06
4e+06
6e+06
8e+06
1
3
4
5
6
7
8
9
10
11
12
13

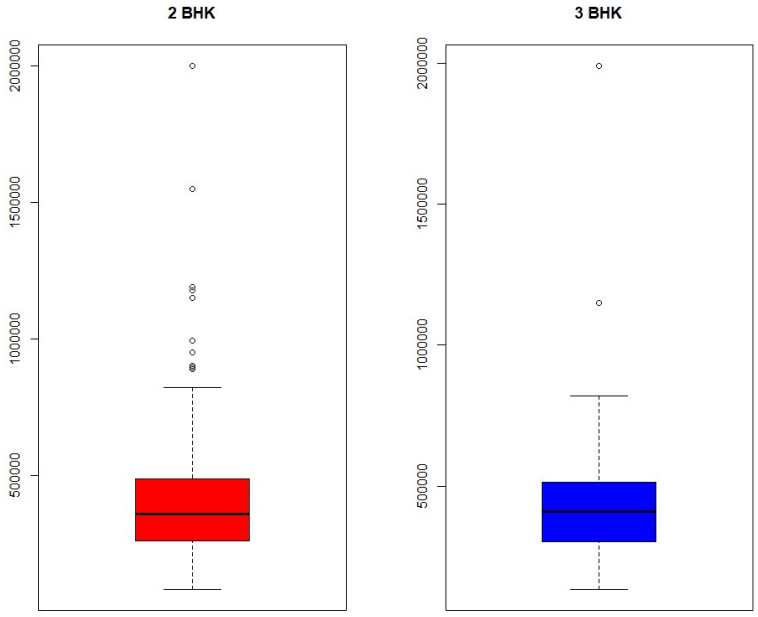

bathrooms
0e+00
2e+06
4e+06
6e+06
8e+06
0
0.75
1.25
1.75
2.25
2.75
3.25
3.75
4.25
4.75
5.25
5.75
6.25
6.75
7.75


bedrooms
0e+00
2e+06
4e+06
6e+06
8e+06
0
1
2
3
4
5
6
7
8
9
10
11
33

Two sample One-tailed Hypothesis Testing:

Null Hypothesis: Mean price of 2 BHK houses is less than 3 BHK houses
Alternative Hypothesis: Mean price of 2 BHK houses is greater than 3 BHK houses.
To check hypothesis, z test has been done where p value is greater than 0.05. Thus, at 95% confidence interval, there is enough evidence to accept null hypothesis.

This test can be modified by 2BHK houses in a particular downtown area to be compared with 3BHK houses in another suburb area: limitations due to no such variable present in data set.




```
> z.test(price_2bhk,price_3bhk,alternative = "greater", mu=0, sigma.x = sd(price_2bhk), sigma.y = sd(price_3bhk
), conf.level = 0.95)

        Two-sample z-Test

data:  price_2bhk and price_3bhk
z = -1.383, p-value = 0.9167
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 -80001.57        NA
sample estimates:
mean of x mean of y
 402073.4  438614.6
```

Confidence Intervals for 2BHK and 3BHK Houses –
Taken random samples from main dataset and this test data produces the confidence interval.

```
> x2=mean(price_2bhk)
> s2 = sd(price_2bhk)
> n2=length(price_2bhk)
> err2 = (qnorm(0.975)*s2)/sqrt(n2)
> n2
[1] 276
> left2 = x2 - err2
> right2 = x2+err2
> left2
[1] 376069.8
> right2
[1] 428077
> x3=mean(price_3bhk)
> s3 = sd(price_3bhk)
> n3=length(price_3bhk)
> n3
[1] 98
> err3 = (qnorm(0.975)*s3)/sqrt(n3)
> left3 = x3 - err3
> right3 = x3 + err3
> left3
[1] 393830.3
> right3
[1] 483398.8
```

```
> mydata_2bhk = mydata %>% filter(bedrooms==2)
> mean(mydata_2bhk$price)
[1] 401387.7
> mydata_3bhk = mydata %>% filter(bedrooms==3)
> mean(mydata_3bhk$price) # actual mean of 3 BHK greater
[1] 466276.6
>
```

5. Methods and Processes:

Multiple Linear Regression:

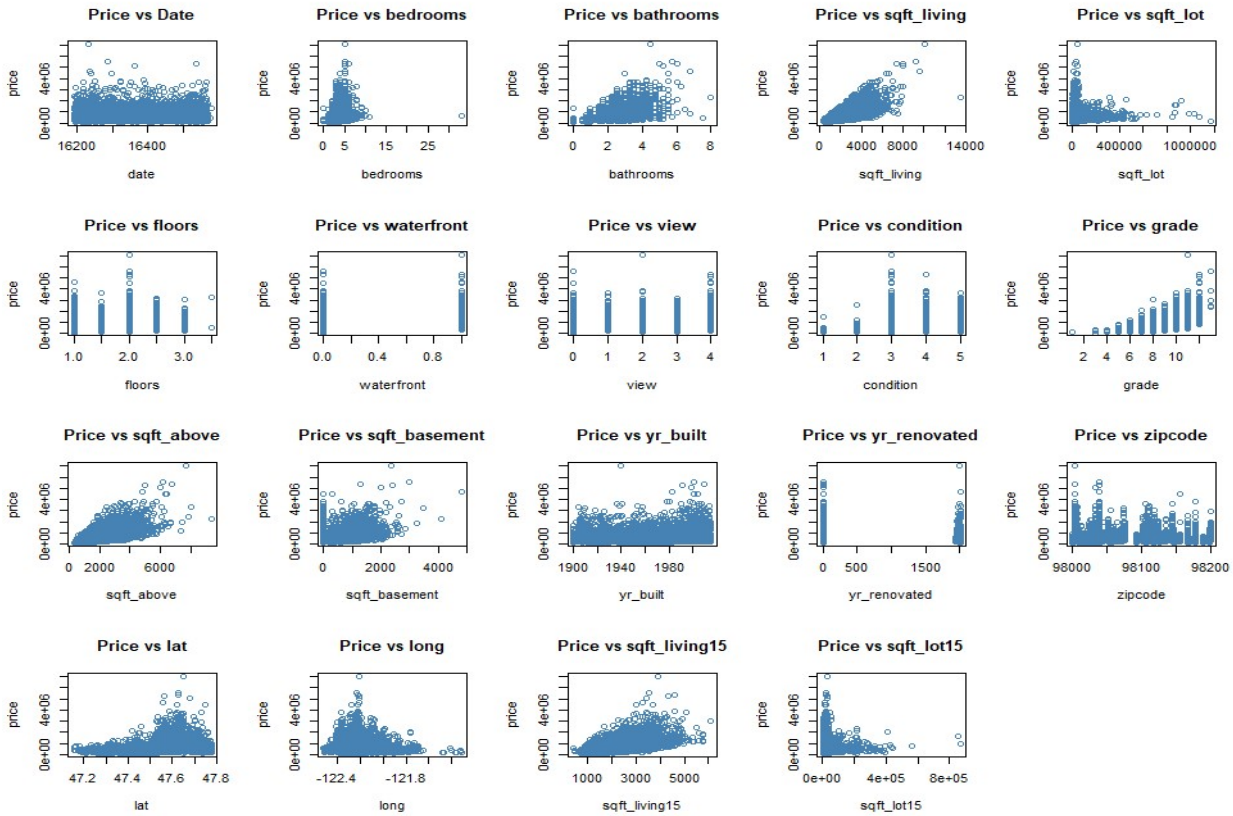
Steps Involved -

1. Load the dataset and libraries in R
2. Divide the data set into training and testing
3. Checking the linear association between variables
4. Multicollinearity exists between variables, but they will taken care later after model building and calculating VIF
5. Convert the categorical variables converted to factors and then to be used in model.

6. Build different regression models using training data set and find the best model based on RMSE value.
7. Selection of different models based on different metrics like AIC, BIC and p value
8. Residual analysis
9. Make predictions on test dataset and find the accuracy

<u>Checking correlation values among the variables</u> - Below is the correlation plot of data:

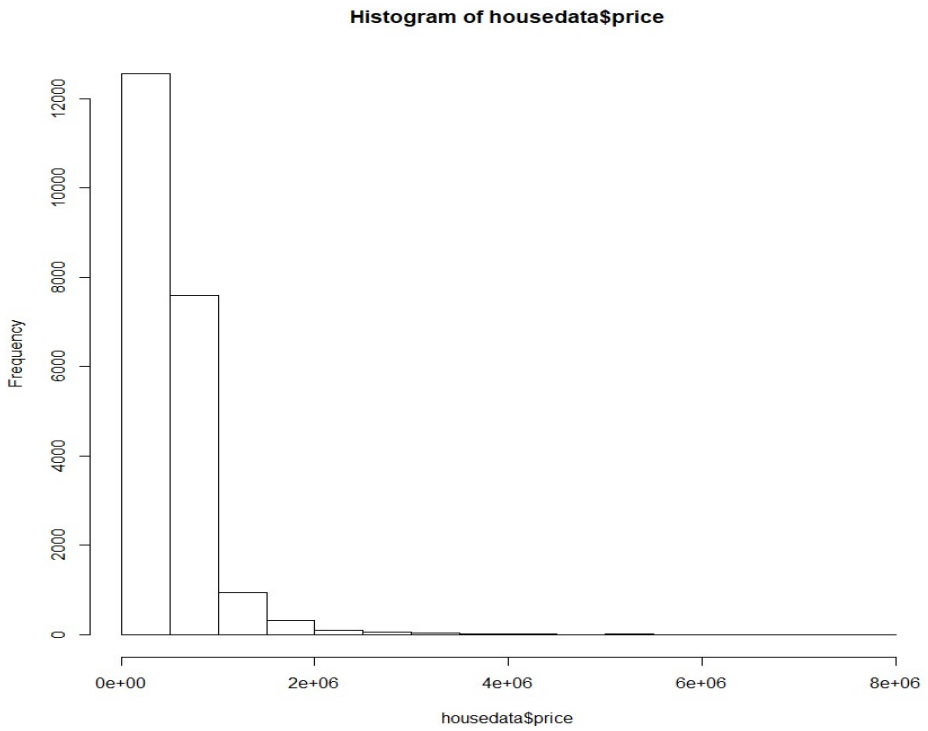
	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sqft_above	sqft_basement	yr_built	yr_renovated	zipcode	lat	long	sqft_living15	sqft_lot15
id	1	0.01	-0.02	0	0.01	-0.01	-0.13	0.02	0	0.01	-0.02	0.01	-0.01	-0.01	0.02	-0.02	-0.01	0	0.02	0	-0.14
date	0.01	1	0	-0.02	-0.03	-0.03	0.01	-0.02	0	0	-0.05	-0.04	-0.03	-0.02	0	-0.02	0	-0.03	-0.01	-0.03	0
price	-0.02	0	1	0.31	0.53	0.7	0.09	0.26	0.27	0.4	0.04	0.67	0.61	0.32	0.05	0.13	-0.05	0.31	0.02	0.59	0.08
bedrooms	0	-0.02	0.31	1	0.52	0.58	0.03	0.18	-0.01	0.08	0.03	0.36	0.48	0.3	0.15	0.02	-0.15	-0.01	0.13	0.39	0.03
bathrooms	0.01	-0.03	0.53	0.52	1	0.75	0.09	0.5	0.06	0.19	-0.12	0.66	0.69	0.28	0.51	0.05	-0.2	0.02	0.22	0.57	0.09
sqft_living	-0.01	-0.03	0.7	0.58	0.75	1	0.17	0.35	0.1	0.28	-0.06	0.76	0.88	0.44	0.32	0.06	-0.2	0.05	0.24	0.76	0.18
sqft_lot	-0.13	0.01	0.09	0.03	0.09	0.17	1	-0.01	0.02	0.07	-0.01	0.11	0.18	0.02	0.05	0.01	-0.13	-0.09	0.23	0.14	0.72
floors	0.02	-0.02	0.26	0.18	0.5	0.35	-0.01	1	0.02	0.03	-0.26	0.46	0.52	-0.25	0.49	0.01	-0.06	0.05	0.13	0.28	-0.01
waterfront	0	0	0.27	-0.01	0.06	0.1	0.02	0.02	1	0.4	0.02	0.08	0.07	0.08	-0.03	0.09	0.03	-0.01	-0.04	0.09	0.03
view	0.01	0	0.4	0.08	0.19	0.28	0.07	0.03	0.4	1	0.05	0.25	0.17	0.28	-0.05	0.1	0.08	0.01	-0.08	0.28	0.07
condition	-0.02	-0.05	0.04	0.03	-0.12	-0.06	-0.01	-0.26	0.02	0.05	1	-0.14	-0.16	0.17	-0.36	-0.06	0	-0.01	-0.11	-0.09	0
grade	0.01	-0.04	0.67	0.36	0.66	0.76	0.11	0.46	0.08	0.25	-0.14	1	0.76	0.17	0.45	0.01	-0.18	0.11	0.2	0.71	0.12
sqft_above	-0.01	-0.03	0.61	0.48	0.69	0.88	0.18	0.52	0.07	0.17	-0.16	0.76	1	-0.05	0.42	0.02	-0.26	0	0.34	0.73	0.19
sqft_basement	-0.01	-0.02	0.32	0.3	0.28	0.44	0.02	-0.25	0.08	0.28	0.17	0.17	-0.05	1	-0.13	0.07	0.07	0.11	-0.14	0.2	0.02
yr_built	0.02	0	0.05	0.15	0.51	0.32	0.05	0.49	-0.03	-0.05	-0.36	0.45	0.42	-0.13	1	-0.22	-0.35	-0.15	0.41	0.33	0.07
yr_renovated	-0.02	-0.02	0.13	0.02	0.05	0.06	0.01	0.01	0.09	0.1	-0.06	0.01	0.02	0.07	-0.22	1	0.06	0.03	-0.07	0	0.01
zipcode	-0.01	0	-0.05	-0.15	-0.2	-0.2	-0.13	-0.06	0.03	0.08	0	-0.18	-0.26	0.07	-0.35	0.06	1	0.27	-0.56	-0.28	-0.15
lat	0	-0.03	0.31	-0.01	0.02	0.05	-0.09	0.05	-0.01	0.01	-0.01	0.11	0	0.11	-0.15	0.03	0.27	1	-0.14	0.05	-0.09
long	0.02	-0.01	0.02	0.13	0.22	0.24	0.23	0.13	-0.04	-0.08	-0.11	0.2	0.34	-0.14	0.41	-0.07	-0.56	-0.14	1	0.33	0.25
sqft_living15	0	-0.03	0.59	0.39	0.57	0.76	0.14	0.28	0.09	0.28	-0.09	0.71	0.73	0.2	0.33	0	-0.28	0.05	0.33	1	0.18
sqft_lot15	-0.14	0	0.08	0.03	0.09	0.18	0.72	-0.01	0.03	0.07	0	0.12	0.19	0.02	0.07	0.01	-0.15	-0.09	0.25	0.18	1

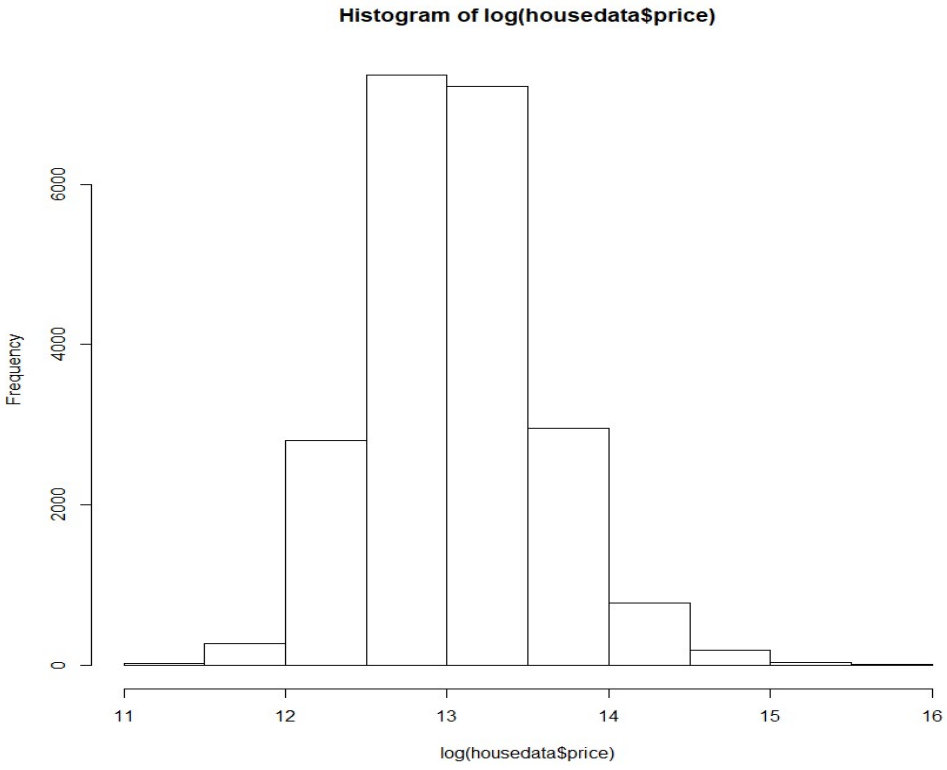


- Linear associations exist between price and some independent variables like sqft-living, bathrooms, sqft_above. So linear model can be built using them.
- Multicollinearity exist between independent variables:
    i. Correlation between sqft_living and sqft_above is (0.88)
    ii. Correlation between sqft_living and sqft_living15 is (0.76)
    iii. Correlation between sqft_living and grade is (0.76)
    iv. Correlation between sqft living and bathrooms is (0.75)

But multicollinearity can be taken care later by checking VIF value of the model.

Building the models and Residual Analysis of each model:





Taking log of price normalizes the data.

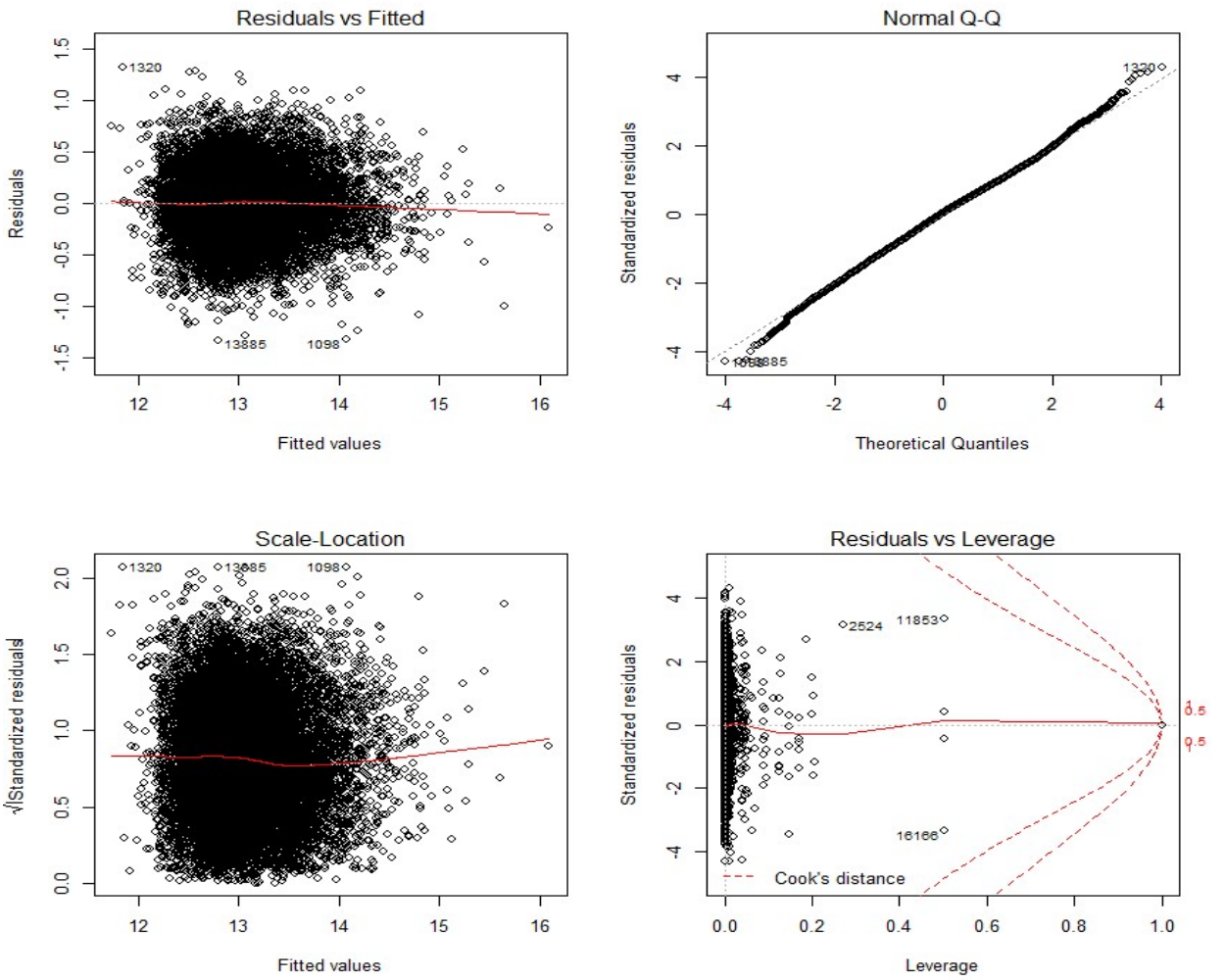
Model 1 – Elimination by P value:

Taken all variables into account in the first model and then eliminating the variables one by one depending on the p value.

```
> full=lm(log(price)~date+bedrooms+bathrooms+sqft_lot+floors+waterfront+view+condition+grade+sqft_above+yr_built+yr_renovated
+long+sqft_living15+sqft_lot15, data=train)
> summary(full)

Call:
lm(formula = log(price) ~ date + bedrooms + bathrooms + sqft_lot +
    floors + waterfront + view + condition + grade + sqft_above +
    yr_built + yr_renovated + long + sqft_living15 + sqft_lot15,
    data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-1.33207 -0.20792  0.01313  0.20703  1.32336

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    7.136e+00  2.662e+00   2.681 0.007351 **
date           1.546e-04  2.094e-05   7.383 1.61e-13 ***
bedrooms1      1.168e-02  1.216e-01   0.096 0.923489
bedrooms2      3.856e-02  1.194e-01   0.323 0.746719
bedrooms3     -3.584e-02  1.193e-01  -0.300 0.763899
bedrooms4     -3.078e-02  1.194e-01  -0.258 0.796615
bedrooms5      3.994e-03  1.198e-01   0.033 0.973401
bedrooms6     -5.245e-02  1.214e-01  -0.432 0.665692
bedrooms7     -7.545e-02  1.335e-01  -0.565 0.571991
bedrooms8      1.051e-01  1.626e-01   0.646 0.518274
bedrooms9     -1.267e-01  1.839e-01  -0.689 0.490820
bedrooms10     8.930e-02  2.502e-01   0.357 0.721153
bedrooms11    -1.715e-01  3.327e-01  -0.516 0.606167
bedrooms33     4.331e-01  3.325e-01   1.303 0.192676
bathrooms      1.397e-01  5.199e-03  26.872  < 2e-16 ***
sqft_lot       3.123e-07  8.009e-08   3.899 9.68e-05 ***
floors1.5      4.350e-02  9.319e-03   4.667 3.08e-06 ***
```

```
grade10         1.008e+00  3.384e-01   2.979 0.002895 **
grade11         1.138e+00  3.389e-01   3.358 0.000786 ***
grade12         1.256e+00  3.409e-01   3.686 0.000229 ***
grade13         1.485e+00  3.513e-01   4.228 2.37e-05 ***
sqft_above      6.297e-05  6.089e-06  10.341  < 2e-16 ***
yr_built       -5.863e-03  1.350e-04 -43.435  < 2e-16 ***
yr_renovated    1.138e-05  6.356e-06   1.791 0.073358 .
long           -1.104e-01  2.069e-02  -5.336 9.60e-08 ***
sqft_living15   1.472e-04  5.879e-06  25.042  < 2e-16 ***
sqft_lot15     -4.459e-07  1.225e-07  -3.640 0.000273 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3102 on 17243 degrees of freedom
Multiple R-squared:  0.6509,    Adjusted R-squared:   0.65
F-statistic: 698.9 on 46 and 17243 DF,  p-value: < 2.2e-16

> vif(full)
                   GVIF Df GVIF^(1/(2*Df))
date           1.007892  1        1.003938
bedrooms       2.399391 12        1.037140
bathrooms      2.873787  1        1.695225
sqft_lot       2.110084  1        1.452613
floors         3.162676  5        1.122033
waterfront     1.498125  1        1.223979
view           1.781590  4        1.074858
condition      1.428906  4        1.045624
grade          5.744928 11        1.082712
sqft_above     4.573728  1        2.138628
yr_built       2.839015  1        1.684938
yr_renovated   1.174091  1        1.083555
long           1.537701  1        1.240041
sqft_living15  2.905789  1        1.704638
sqft_lot15     2.147883  1        1.465566
>
```

Residual Analysis of the above model: The residuals are normally distributed from QQ Plot.



Below is RMSE and prediction for model1:
The predictions are close and the **RMSE value is 3181.399**. This RMSE will be compared with other models to determine the best model with lowest RMSE.

```
> y1=predict.glm(full,test)
> y1=exp(y1)
> y=test$price
> rmse_1 = sqrt((y-y1)%*%(y-y1))/nrow(test)
> rmse_1
         [,1]
[1,] 3181.399
> head(y1)
       8       16       17       25       28       31
661174.0 317097.9 351262.3 665358.6 503166.0 761677.2
> head(y)
[1] 482000 242500 419000 612500 615000 790000
>
```

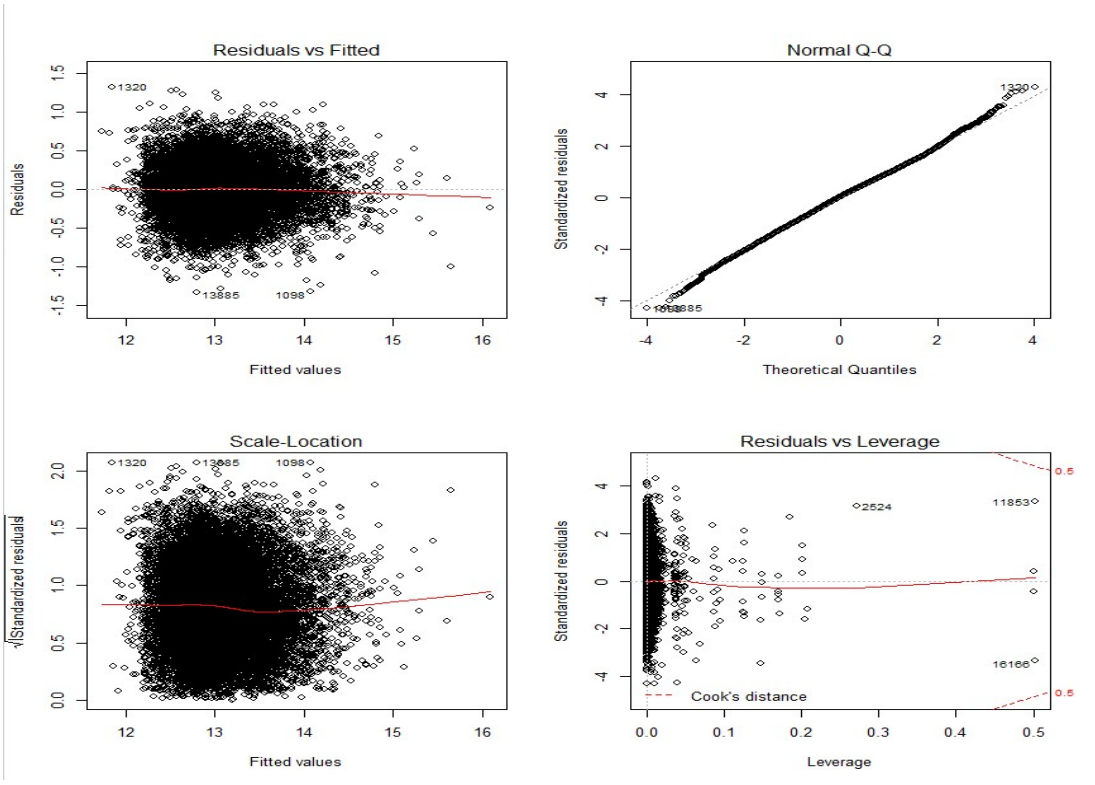
Model2: Model by using step function “forward” selection and applying log on price.

```
> model3=step(base, scope=list(upper=full, lower=~1), direction="forward", trace=F)
> summary(model3)

Call:
lm(formula = log(price) ~ date + grade + yr_built + bathrooms +
    sqft_living15 + view + floors + condition + waterfront +
    sqft_above + bedrooms + long + sqft_lot + sqft_lot15 + yr_renovated,
    data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-1.33207 -0.20792  0.01313  0.20703  1.32336

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    7.136e+00  2.662e+00   2.681 0.007351 **
date           1.546e-04  2.094e-05   7.383 1.61e-13 ***
grade3        -7.381e-02  4.037e-01  -0.183 0.854944
grade4        -2.796e-01  3.426e-01  -0.816 0.414313
grade5        -1.194e-01  3.381e-01  -0.353 0.723903
grade6         9.676e-02  3.379e-01   0.286 0.774585
grade7         3.945e-01  3.378e-01   1.168 0.242929
grade8         6.140e-01  3.380e-01   1.817 0.069266 .
grade9         8.422e-01  3.381e-01   2.491 0.012762 *
grade10        1.008e+00  3.384e-01   2.979 0.002895 **
grade11        1.138e+00  3.389e-01   3.358 0.000786 ***
grade12        1.256e+00  3.409e-01   3.686 0.000229 ***
grade13        1.485e+00  3.513e-01   4.228 2.37e-05 ***
yr_built      -5.863e-03  1.350e-04 -43.435  < 2e-16 ***
bathrooms      1.397e-01  5.199e-03  26.872  < 2e-16 ***
sqft_living15  1.472e-04  5.879e-06  25.042  < 2e-16 ***
view1          1.796e-01  1.912e-02   9.394  < 2e-16 ***
view2          8.321e-02  1.172e-02   7.099 1.30e-12 ***
view3          1.027e-01  1.628e-02   6.309 2.88e-10 ***
view4          2.311e-01  2.475e-02   9.339  < 2e-16 ***
```

```
bedrooms9      -1.267e-01  1.839e-01  -0.689 0.490820
bedrooms10      8.930e-02  2.502e-01   0.357 0.721153
bedrooms11     -1.715e-01  3.327e-01  -0.516 0.606167
bedrooms33      4.331e-01  3.325e-01   1.303 0.192676
bathrooms       1.397e-01  5.199e-03  26.872  < 2e-16 ***
sqft_lot        3.123e-07  8.009e-08   3.899 9.68e-05 ***
sqft_living15   1.472e-04  5.879e-06  25.042  < 2e-16 ***
long           -1.104e-01  2.069e-02  -5.336 9.60e-08 ***
sqft_lot15     -4.459e-07  1.225e-07  -3.640 0.000273 ***
yr_built       -5.863e-03  1.350e-04 -43.435  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3102 on 17243 degrees of freedom
Multiple R-squared:  0.6509,    Adjusted R-squared:   0.65
F-statistic: 698.9 on 46 and 17243 DF,  p-value: < 2.2e-16

> vif(model3)
                 GVIF Df GVIF^(1/(2*Df))
date          1.007892  1        1.003938
grade         5.744928 11        1.082712
view          1.781590  4        1.074858
waterfront    1.498125  1        1.223979
condition     1.428906  4        1.045624
yr_renovated  1.174091  1        1.083555
floors        3.162676  5        1.122033
sqft_above    4.573728  1        2.138628
bedrooms      2.399391 12        1.037140
bathrooms     2.873787  1        1.695225
sqft_lot      2.110084  1        1.452613
sqft_living15 2.905789  1        1.704638
long          1.537701  1        1.240041
sqft_lot15    2.147883  1        1.465566
yr_built      2.839015  1        1.684938
>
```

Residual Analysis of the above model: The residuals are normally distributed from QQ Plot.
Below is RMSE and prediction for model1:
The predictions are close and the **RMSE value is 3181.399**. This RMSE will be compared with other models to determine the best model with lowest RMSE.

Forward selection with step function yields the same model as model1.



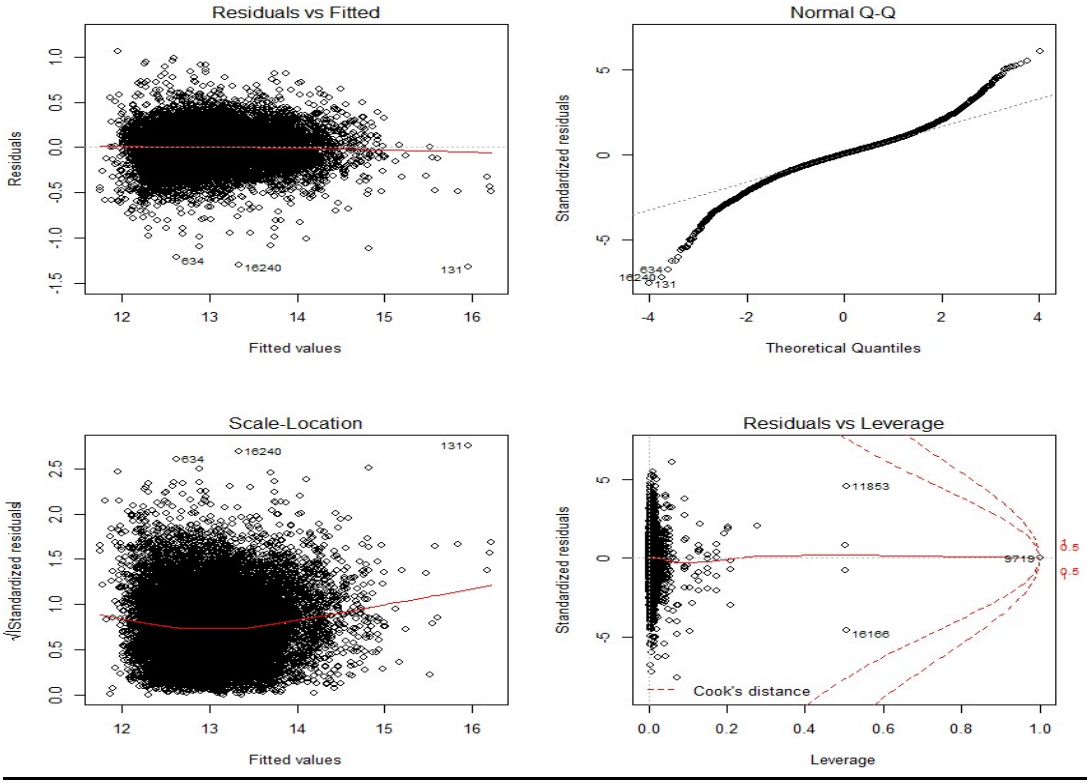

```
> par(mfrow = c(2, 2))
> plot(model3)
Warning messages:
1: not plotting observations with leverage one:
  5366, 6978, 10707
2: not plotting observations with leverage one:
  5366, 6978, 10707
> y3=predict.glm(model3,test)
> y=test$price
> y3=exp(y3)
> rmse_3 = sqrt((y-y3)%*%(y-y3))/nrow(test)
> rmse_3
         [,1]
[1,] 3181.399
> head(y3)
       8       16       17       25       28       31
661174.0 317097.9 351262.3 665358.6 503166.0 761677.2
> head(y)
[1] 482000 242500 419000 612500 615000 790000
>
```

Model3: Using step function “both” with AIC metric for selection- Best model with lowest RMSE.

```
> library(MASS)
> modelAIC<-stepAIC(model1, direction="both")
Start:  AIC=411747.5
price ~ date + bedrooms + bathrooms + sqft_living + sqft_lot +
    floors + waterfront + view + condition + grade + sqft_above +
    yr_built + yr_renovated + zipcode + lat + long + sqft_living15 +
    sqft_lot15

               Df  Sum of Sq        RSS    AIC
<none>                       3.7518e+14 411748
- yr_built      1 1.6695e+11 3.7535e+14 411753
- sqft_lot15    1 1.9849e+11 3.7538e+14 411755
- long          1 3.7544e+11 3.7556e+14 411763
- lat           1 5.0482e+11 3.7569e+14 411769
- sqft_living15 1 7.3158e+11 3.7591e+14 411779
- sqft_lot      1 9.9999e+11 3.7618e+14 411792
- bedrooms     12 2.0250e+12 3.7721e+14 411817
- yr_renovated  1 1.7685e+12 3.7695e+14 411827
- bathrooms     1 1.8478e+12 3.7703e+14 411830
- date          1 2.7953e+12 3.7798e+14 411874
- floors        5 3.3261e+12 3.7851e+14 411890
- sqft_above    1 3.7333e+12 3.7892e+14 411917
- condition     4 6.1405e+12 3.8132e+14 412020
- view          4 1.8139e+13 3.9332e+14 412556
- sqft_living   1 2.0719e+13 3.9590e+14 412675
- waterfront    1 2.6656e+13 4.0184e+14 412932
- grade        11 8.2423e+13 4.5761e+14 415159
- zipcode      69 2.5788e+14 6.3306e+14 420655
>
```

```
zipcode98115   5.531e-01  3.293e-02  16.795  < 2e-16 ***
zipcode98116   5.237e-01  2.670e-02  19.615  < 2e-16 ***
zipcode98117   5.207e-01  3.334e-02  15.617  < 2e-16 ***
zipcode98118   3.000e-01  2.345e-02  12.792  < 2e-16 ***
zipcode98119   7.002e-01  3.228e-02  21.690  < 2e-16 ***
zipcode98122   5.982e-01  2.899e-02  20.636  < 2e-16 ***
zipcode98125   2.979e-01  3.550e-02   8.391  < 2e-16 ***
zipcode98126   3.543e-01  2.452e-02  14.447  < 2e-16 ***
zipcode98133   1.595e-01  3.670e-02   4.347 1.39e-05 ***
zipcode98136   4.700e-01  2.505e-02  18.758  < 2e-16 ***
zipcode98144   4.655e-01  2.695e-02  17.269  < 2e-16 ***
zipcode98146   1.057e-01  2.271e-02   4.654 3.27e-06 ***
zipcode98148   5.196e-02  3.027e-02   1.716 0.086113 .
zipcode98155   1.272e-01  3.817e-02   3.333 0.000862 ***
zipcode98166   1.843e-01  2.061e-02   8.942  < 2e-16 ***
zipcode98168  -4.606e-02  2.191e-02  -2.102 0.035556 *
zipcode98177   2.839e-01  3.829e-02   7.414 1.28e-13 ***
zipcode98178   3.816e-02  2.246e-02   1.699 0.089408 .
zipcode98188   3.015e-03  2.327e-02   0.130 0.896903
zipcode98198  -6.350e-03  1.744e-02  -0.364 0.715855
zipcode98199   5.769e-01  3.164e-02  18.235  < 2e-16 ***
lat            6.217e-01  7.958e-02   7.812 5.96e-15 ***
long          -3.971e-01  5.762e-02  -6.892 5.68e-12 ***
sqft_living15  8.110e-05  3.658e-06  22.172  < 2e-16 ***
sqft_lot15     6.778e-08  7.347e-08   0.923 0.356253
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1801 on 17172 degrees of freedom
Multiple R-squared:  0.8828,    Adjusted R-squared:  0.882
F-statistic:  1105 on 117 and 17172 DF,  p-value: < 2.2e-16
```

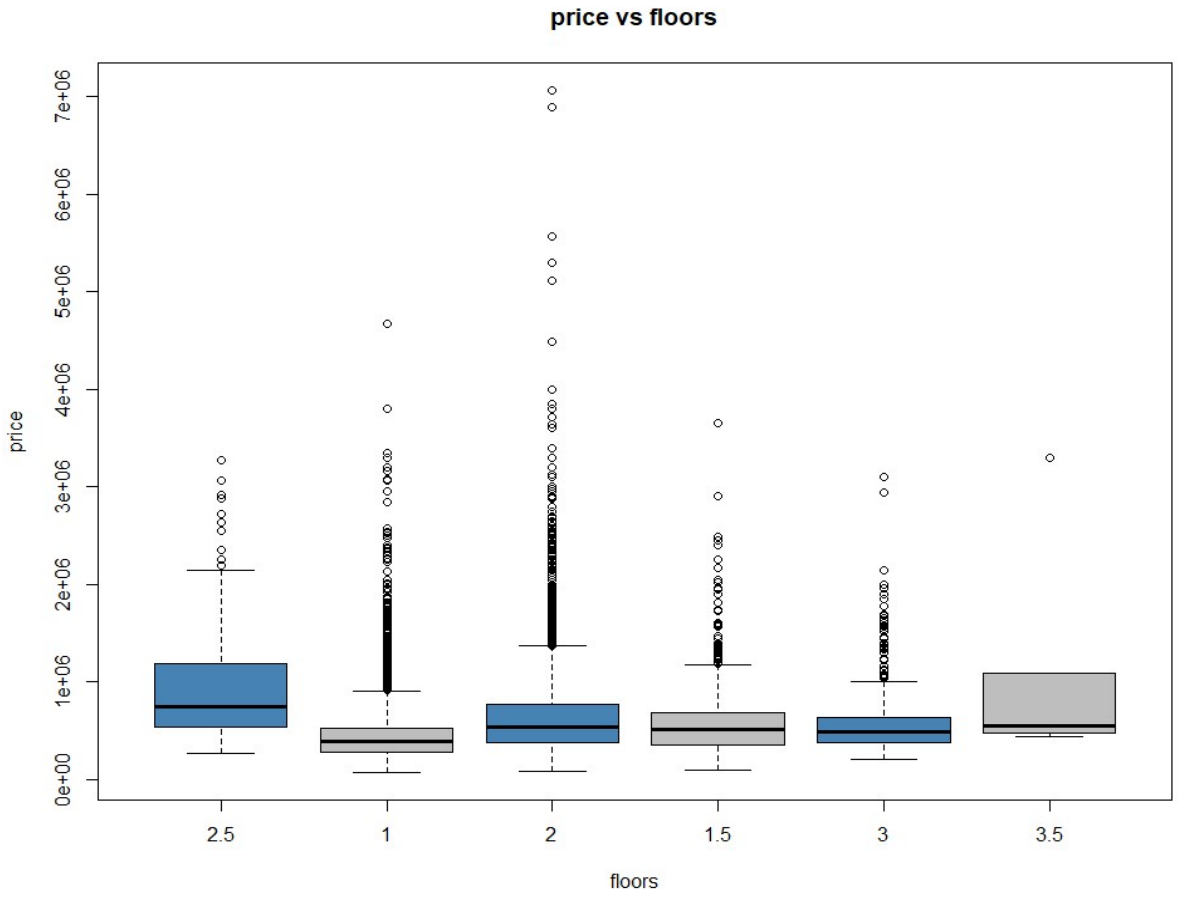
Residual Analysis and Predictions of Model3: Predictions are very close, and this has been tested on test dataset. This model has lowest RMSE.




```
> yaic=predict.glm(modelAIClog,test)
> y=test$price
> yaic=exp(yaic)
> rmse_aic = sqrt((y-yaic)%*%(y-yaic))/nrow(test)
> rmse_aic
         [,1]
[1,] 1907.606
> head(yaic)
       8       16       17       25       28       31
462370.5 240506.9 452722.2 597538.6 566037.3 831184.7
> head(y)
[1] 482000 242500 419000 612500 615000 790000
>
```

Model3 has the lowest RMSE value 1907.606 among the three models as stated above. So model3 is the best model by multiple linear regression and can be used for prediction of house prices.

ANOVA: Extension to Linear Regression



Null Hypothesis: Group means of price of all houses with different floors are equal
Alternative Hypothesis: Group means of price of all houses with different floors are not equal

```
> anova1=lm(price~floors)
> summary(anova1)

Call:
lm(formula = price ~ floors)

Residuals:
    Min      1Q  Median      3Q     Max
-699608 -201581  -71641  101327 6412827

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   442700       3796 116.616  < 2e-16 ***
floors1.5     116047       9824  11.812  < 2e-16 ***
floors2       206973       5770  35.871  < 2e-16 ***
floors2.5     524407      31827  16.477  < 2e-16 ***
floors3       140908      16158   8.721  < 2e-16 ***
floors3.5     626716     143706   4.361  1.3e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 351900 on 17284 degrees of freedom
Multiple R-squared:  0.07996,   Adjusted R-squared:  0.0797
F-statistic: 300.4 on 5 and 17284 DF,  p-value: < 2.2e-16

>
```

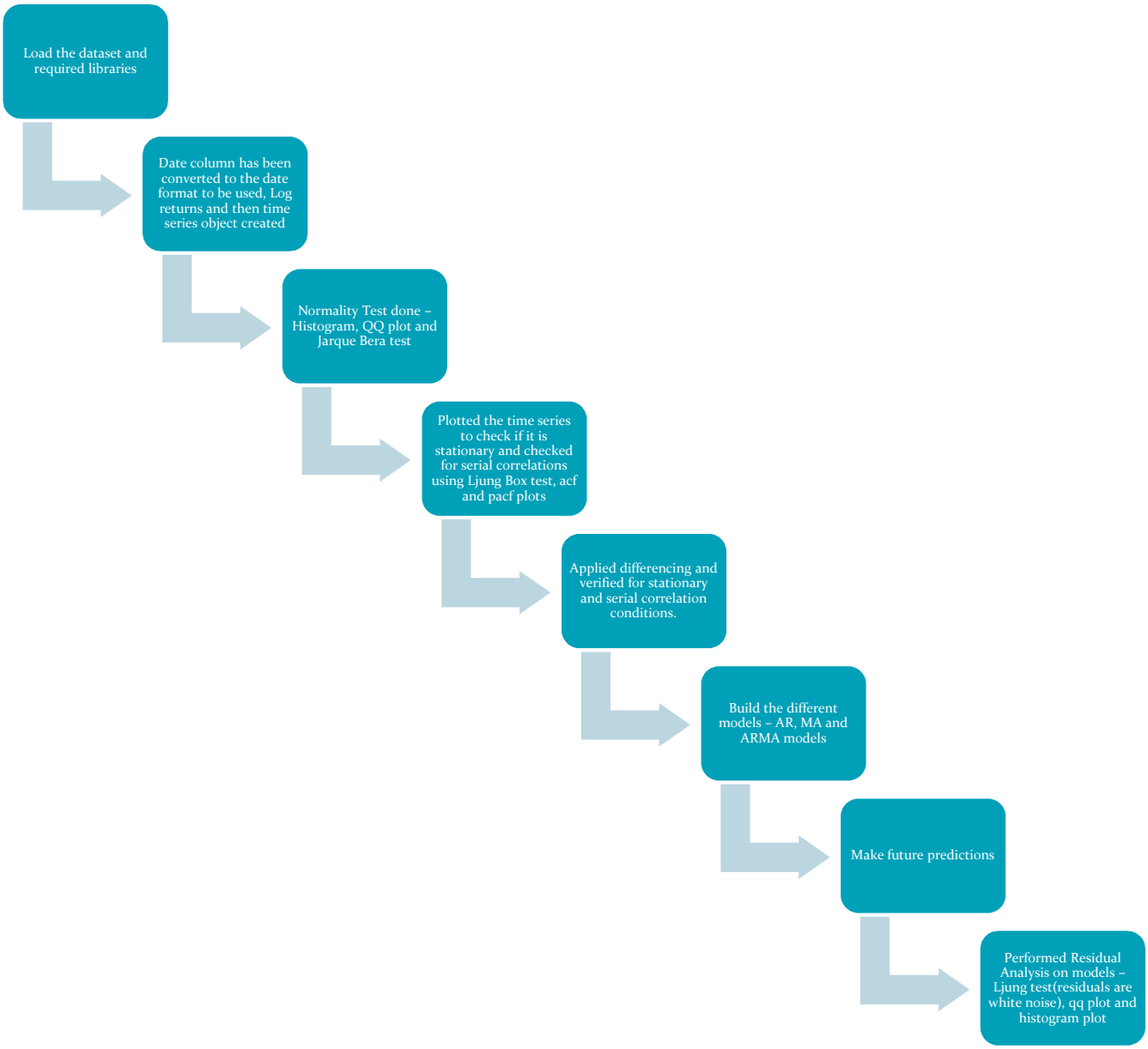
After building the model, we find that F test is satisfied as p value for model is less than 0.05 at 95% confidence which means that at least one of the group means in price for different types of floors are different.

Here, the baseline has been taken as floor1 in the model and group means of all houses differ for different kinds of floors from floor1.

Releveling the baseline of floor to 2.5 . Before floor=1 is taken as baseline and other floor's group price means are compared with floor1 group mean.
Rebuilding the anova model and we can find that other types of floors have different group price means from floor2.5 except floor3.5 which has same group mean for price as floor2.5(because pvalue>0.05).

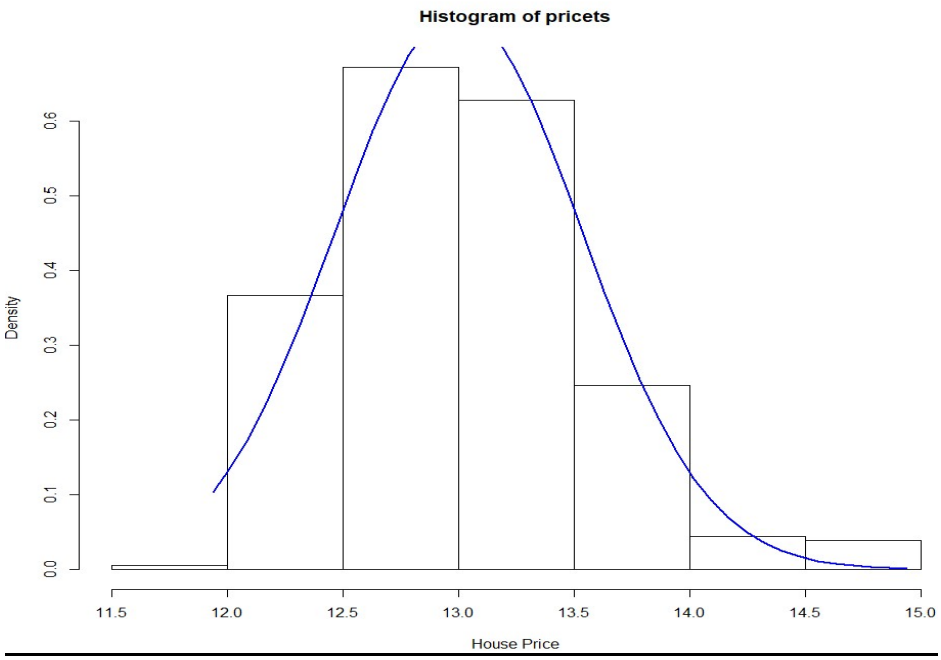
```
> floors = relevel(floors, ref=5) #floors=2.5 taken as baseline
> anova2=lm(price~floors)
> summary(anova2)

Call:
lm(formula = price ~ floors)

Residuals:
    Min      1Q  Median      3Q     Max
-699608 -201581  -71641  101327 6412827

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   967108      31600  30.605   <2e-16 ***
floors1      -524407      31827 -16.477   <2e-16 ***
floors2      -317435      31898  -9.952   <2e-16 ***
floors1.5    -408360      32874 -12.422   <2e-16 ***
floors3      -383499      35288 -10.868   <2e-16 ***
floors3.5     102309     147091   0.696    0.487
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 351900 on 17284 degrees of freedom
Multiple R-squared:  0.07996,   Adjusted R-squared:  0.0797
F-statistic: 300.4 on 5 and 17284 DF,  p-value: < 2.2e-16

>
```

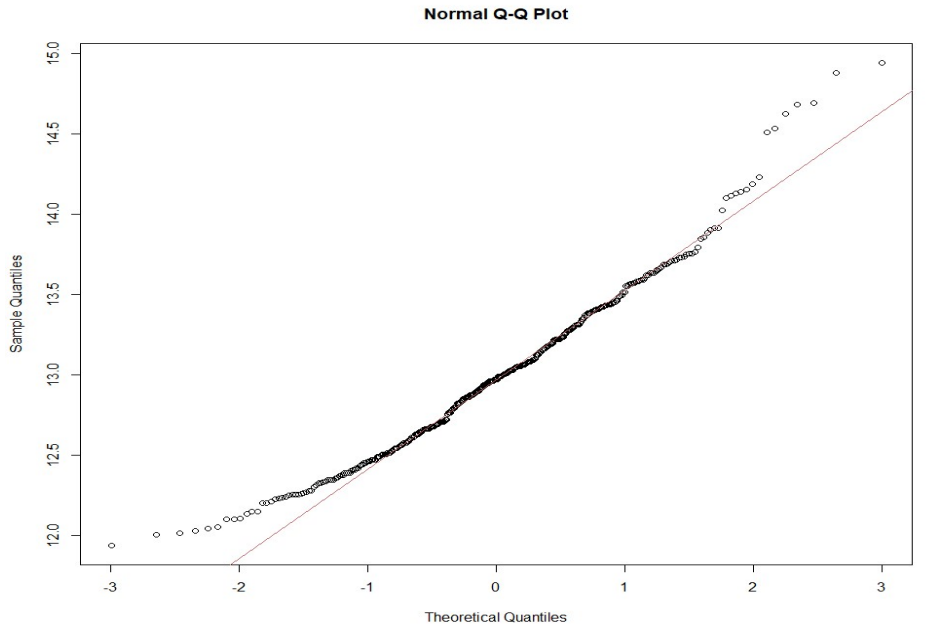
Time Series Analysis and Forecasting:



Time Series Object Normality Tests: Histogram, QQ Plot and Jarque Bera test show that price is coming from almost normal distribution and is slightly right skewed.

```
> rt= log(housedata$price+1)
> pricets=ts(rt,start =c(2014,05), end=c(2015,05), frequency=365)
>
```

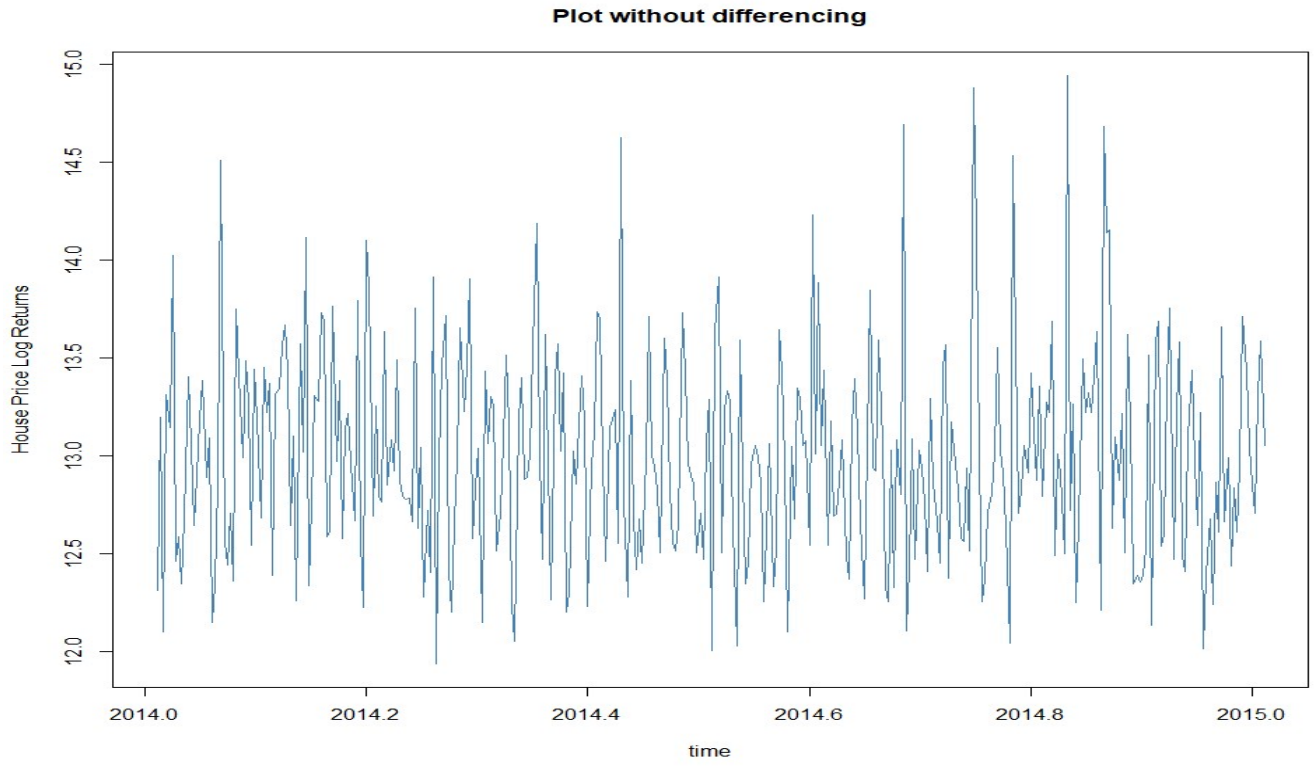


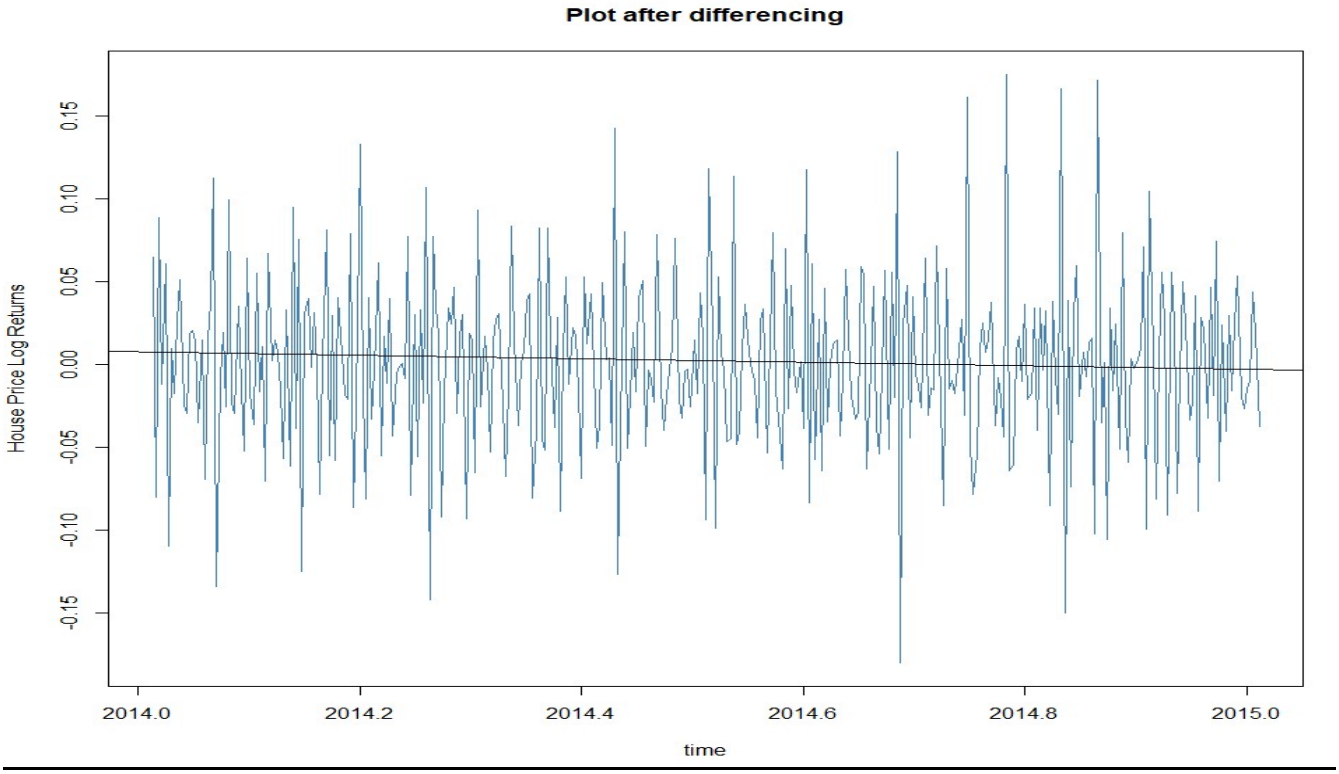



```
> jarque.bera.test(housedata$price) #price is coming from normal distribution

        Jarque Bera Test

data:  housedata$price
X-squared = 1131000, df = 2, p-value < 2.2e-16

>
```

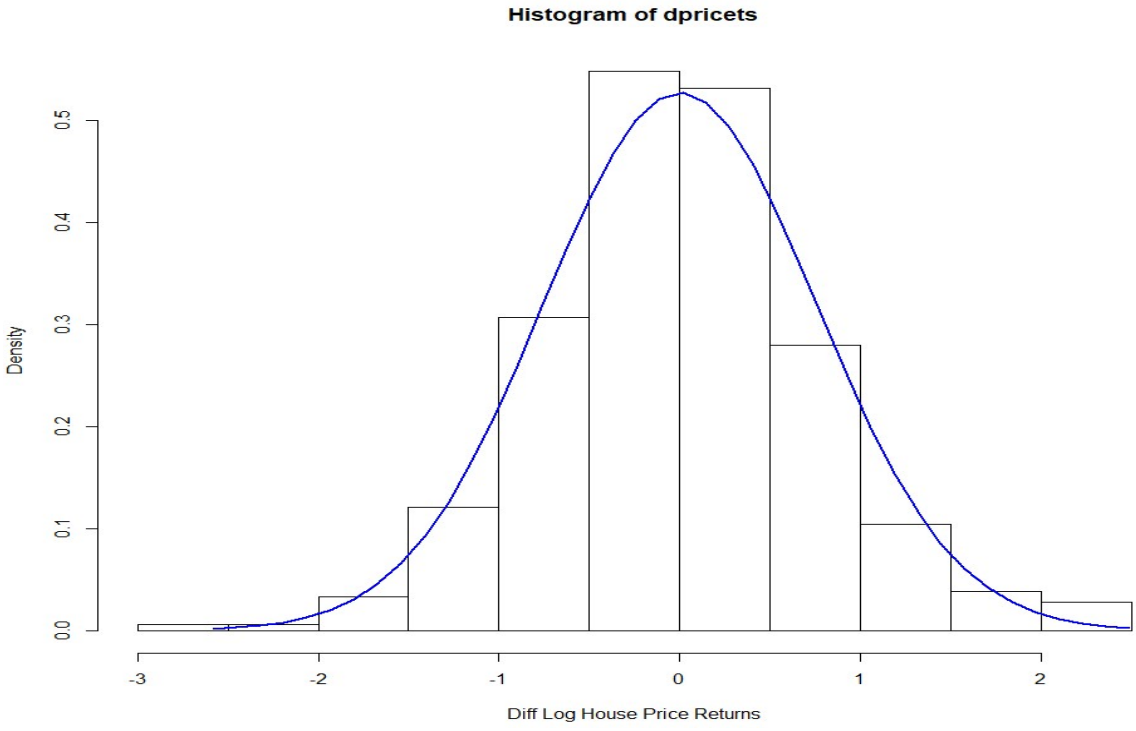
Time Series Plots:

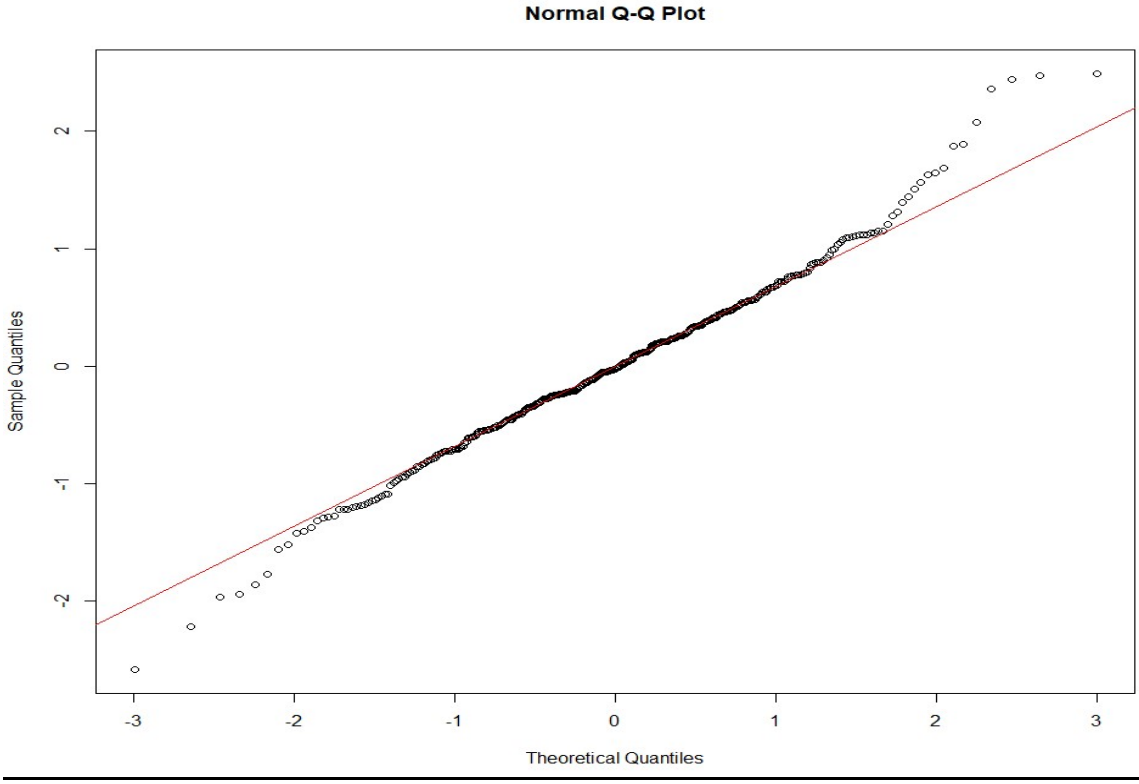
Mean and variance looks almost constant with time: Stationary series





Differenced Time Series Object Normality Tests: Histogram, QQ Plot and Jarque Bera test show that price is coming from almost normal distribution.



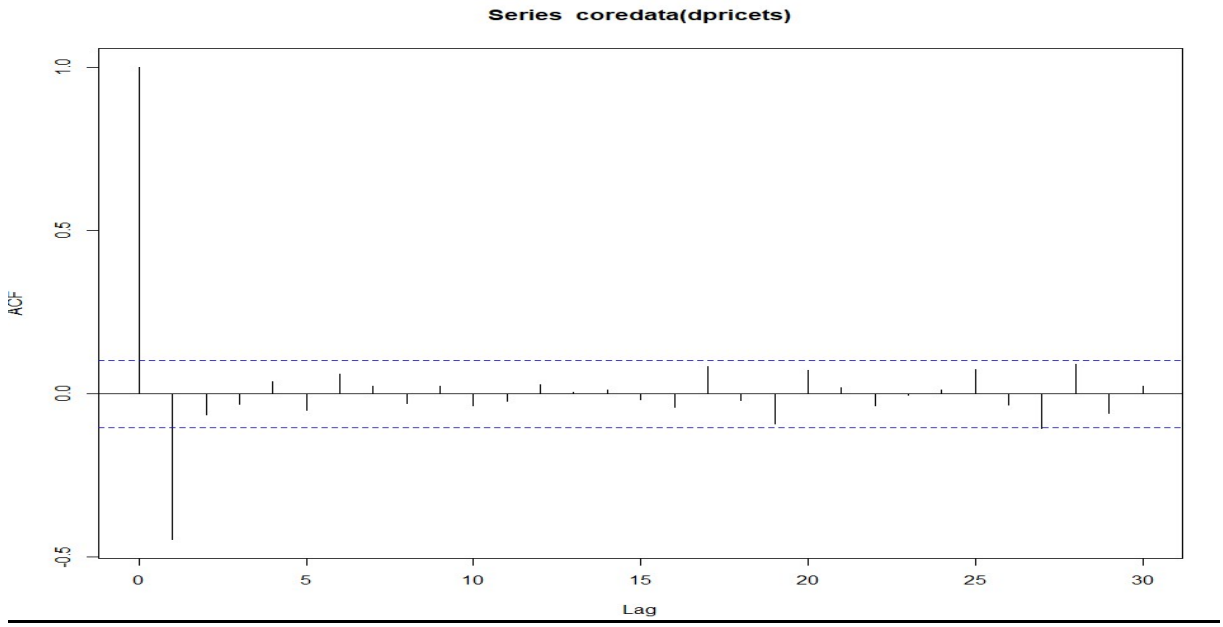


```
> normalTest(dpricets, method=c("jb"))

Title:
 Jarque - Bera Normalality Test

Test Results:
  STATISTIC:
    X-squared: 16.1513
  P VALUE:
    Asymptotic p Value: 0.000311

Description:
 Wed Nov 29 02:58:43 2017 by user: Shephalika
```

Plots and test show that serial correlations exist
Series is not a white noise: Confirmed by Ljung Box test as p value is less than 0.05 – Performed on diff time series object
#H0: Series is not correlated, and autocorrelations of time series object is zero
#H1: Series is correlated

```
> Box.test(dpricets,lag=6,type='Ljung')

        Box-Ljung test

data:  dpricets
X-squared = 77.559, df = 6, p-value = 1.144e-14

> Box.test(dpricets,lag=12,type='Ljung')

        Box-Ljung test

data:  dpricets
X-squared = 79.379, df = 12, p-value = 5.42e-12
```

ACF plot decays quickly which shows that series is serially correlated.



Model1: Model Using AR (1,0,0) and auto.arima model:

Both are same models with same AIC value.

```
> model_AR=arima(dpricets, order=c(1,0,0))
> model_AR

Call:
arima(x = dpricets, order = c(1, 0, 0))

Coefficients:
          ar1  intercept
      -0.4460     0.0017
s.e.   0.0469     0.0245

sigma^2 estimated as 0.458:  log likelihood = -375.5,  aic = 756.99
>
```

```
> library(forecast)
> model_arima=auto.arima(coredata(dpricets), max.p = 20,max.q = 20 ,stationary = TRUE,ic = c("aic"), stepwise =
 TRUE)
> model_arima
Series: coredata(dpricets)
ARIMA(1,0,0) with non-zero mean

Coefficients:
         ar1    mean
      -0.4460  0.0017
s.e.   0.0469  0.0245

sigma^2 estimated as 0.4605:  log likelihood=-375.5
AIC=756.99   AICc=757.06   BIC=768.69
>
```

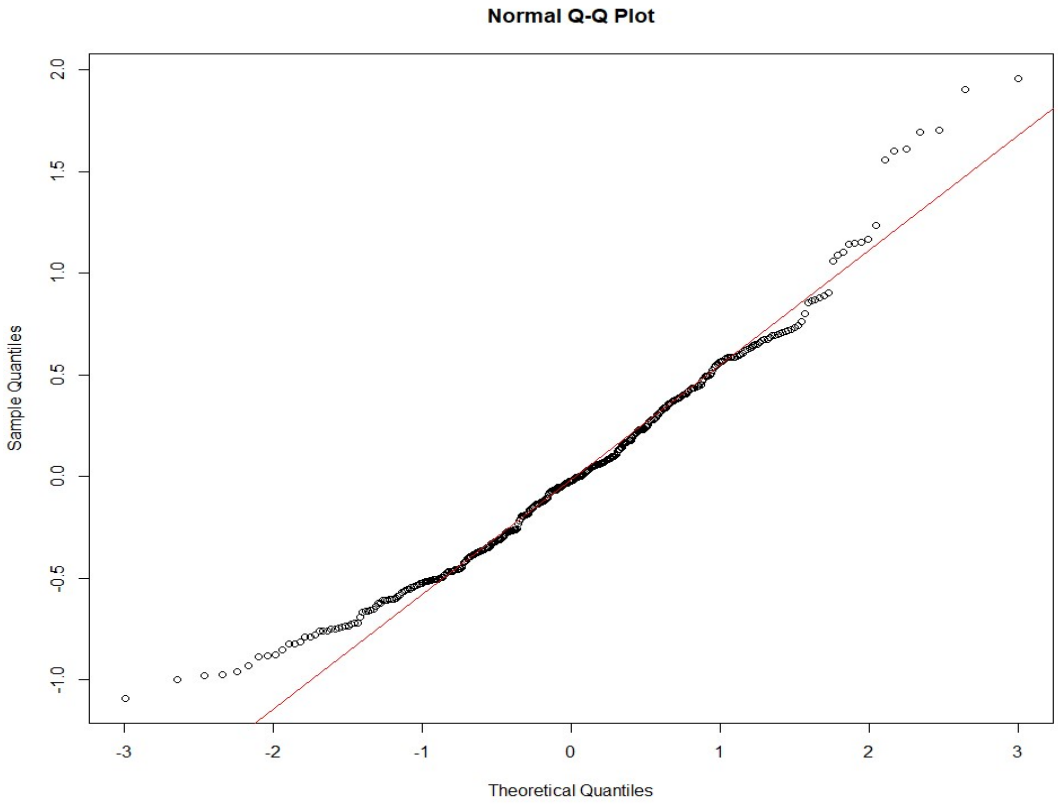
Model 2: Model using EACF

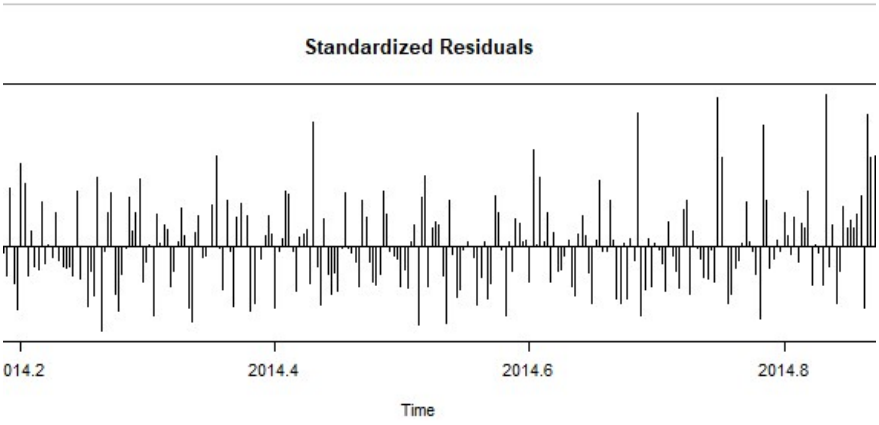
```
> source("EACF.R")
> EACF(dpricets) #p=1 and q=2
[1] "EACF table"
      [,1]   [,2]    [,3]     [,4]    [,5]   [,6]
[1,] -0.44 -0.063 -0.0316  0.03869 -0.0493 0.0608
[2,] -0.51  0.089 -0.0807 -0.00390 -0.0012 0.0511
[3,] -0.52 -0.320 -0.0615 -0.01288 -0.0019 0.0585
[4,] -0.50 -0.492 -0.0344 -0.00091 -0.0097 0.0493
[5,] -0.51 -0.511 -0.0064  0.09224 -0.2506 0.0064
[6,] -0.49  0.133 -0.1651  0.26354 -0.2195 0.0612
[1] " "
[1] "Simplified EACF: 2 denotes significance"
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    2    0    0    0    0    0
[2,]    2    0    0    0    0    0
[3,]    2    2    0    0    0    0
[4,]    2    2    0    0    0    0
[5,]    2    2    0    0    2    0
[6,]    2    2    2    2    2    0
> model_ARMA=arima(dpricets, order=c(1,0,2), method='ML', include.mean = T)
> model_ARMA

Call:
arima(x = dpricets, order = c(1, 0, 2), include.mean = T, method = "ML")

Coefficients:
          ar1      ma1      ma2  intercept
      -0.3989  -0.5783  -0.4217     -1e-04
s.e.   0.6184   0.6147   0.6147      3e-04

sigma^2 estimated as 0.2852:  log likelihood = -291.87,  aic = 593.74
>
```

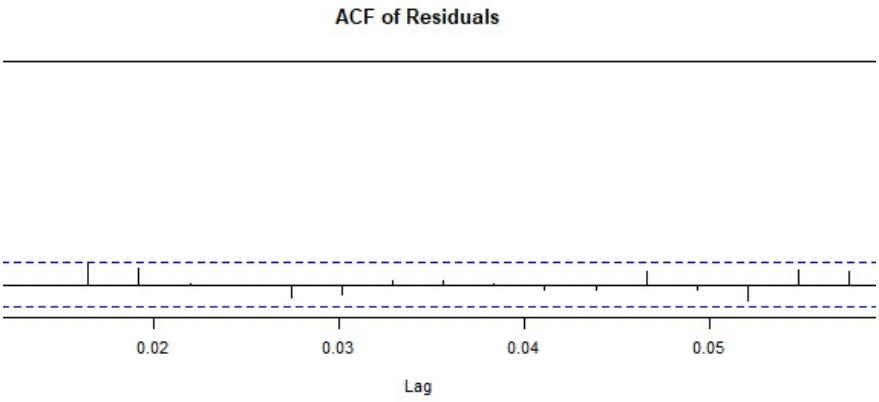
Model 3: Model Using MA – Model with lowest AIC value

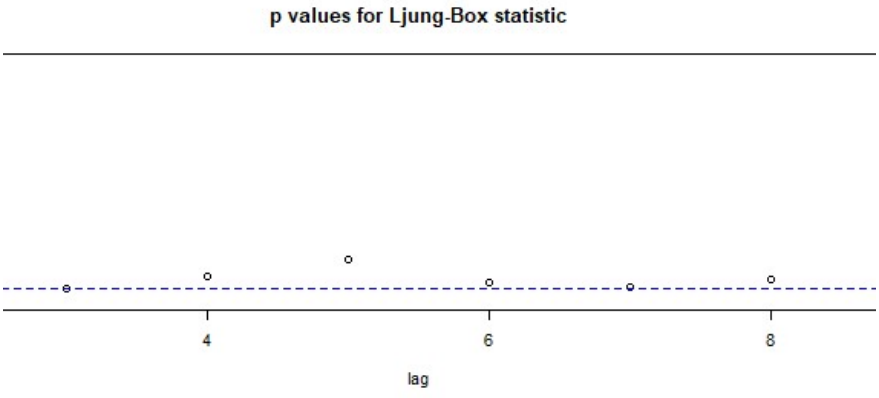
```
> model_MA=arima(dpricets, order=c(0,0,1))
> model_MA #lowest AIC value

Call:
arima(x = dpricets, order = c(0, 0, 1))

Coefficients:
         ma1  intercept
      -1.000     -1e-04
s.e.   0.008      3e-04

sigma^2 estimated as 0.2853:  log likelihood = -291.98,  aic = 589.96
>
```

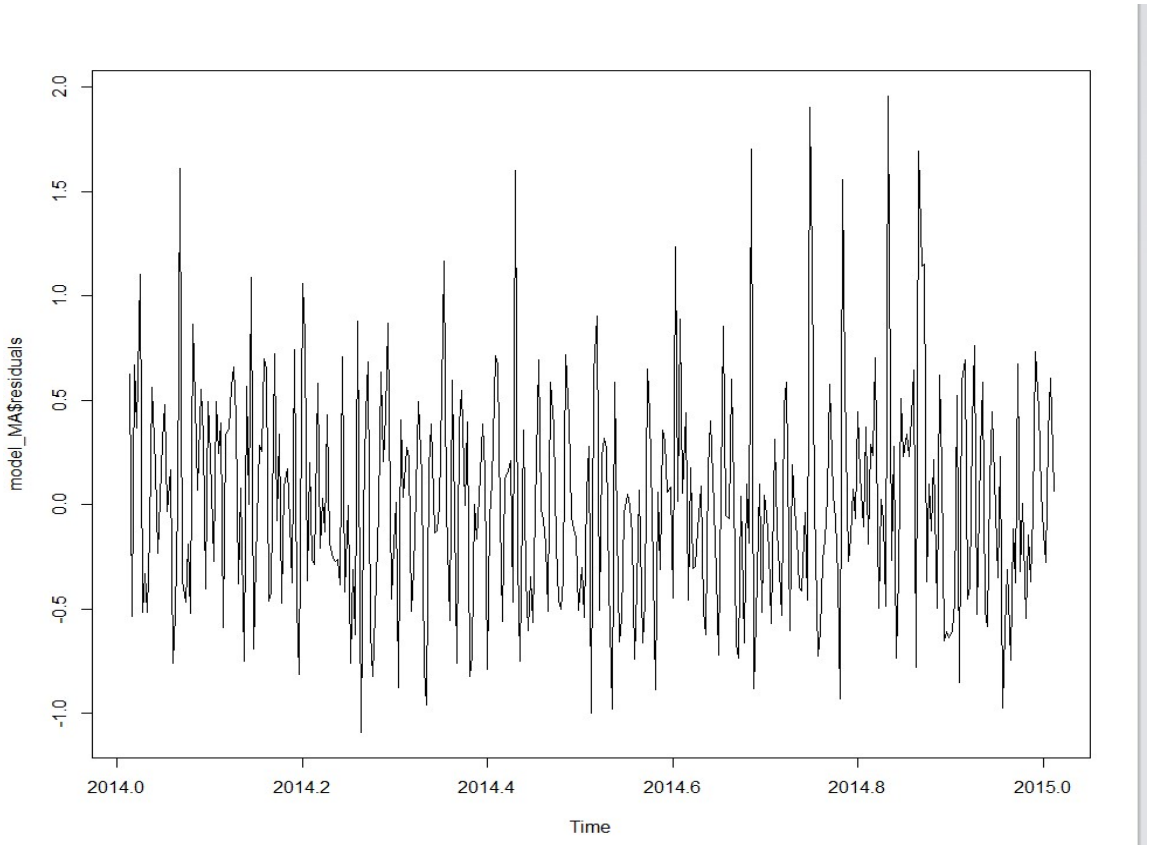
Residual Analysis for MA model:




Standardized Residuals
014.2
2014.4
2014.6
2014.8
Time


ACF of Residuals
0.02
0.03
0.04
0.05
Lag


p values for Ljung-Box statistic
4
6
8
lag



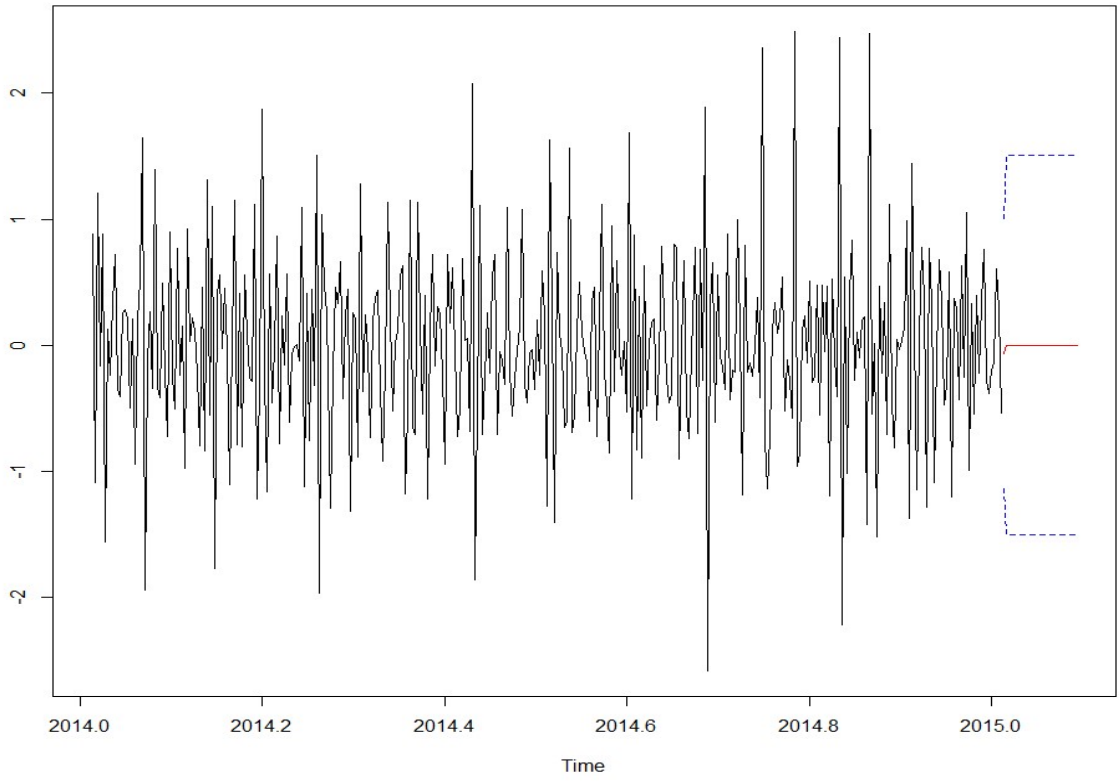
Residual Analysis for MA model: Ljung test result says that the residuals is white noise(independent). Assumption met.

```
> Box.test(model_MA$resid, lag=6, type='Ljung') #Residuals are white noise

        Box-Ljung test

data:  model_MA$resid
X-squared = 11.361, df = 6, p-value = 0.07783

> Box.test(model_MA$resid, lag=12, type='Ljung')

        Box-Ljung test

data:  model_MA$resid
X-squared = 15.934, df = 12, p-value = 0.1943

> Box.test(model_MA$resid, lag=18, type='Ljung')

        Box-Ljung test

data:  model_MA$resid
X-squared = 18.245, df = 18, p-value = 0.4397
```

6. Results and Findings:

Evaluations for best model for linear regression has been made based on lowest RMSE value.
Prediction from Linear Regression:
Best model is model3 and predictions from model3 are almost close to the actual values. The factors used to build model 3 are useful variables for price prediction.

Predictions for future with time series:



```
$pred
Time Series:
Start = c(2015, 6)
End = c(2015, 35)
Frequency = 365
 [1]  0.2417201197 -0.1053081288  0.0494796607 -0.0195615769  0.0112334393 -0.0025023084  0.0036243572
 [8]  0.0008916314  0.0021105310  0.0015668554  0.0018093554  0.0017011912  0.0017494365  0.0017279173
[15]  0.0017375157  0.0017332344  0.0017351440  0.0017342923  0.0017346722  0.0017345027  0.0017345783
[22]  0.0017345446  0.0017345597  0.0017345529  0.0017345559  0.0017345546  0.0017345552  0.0017345549
[29]  0.0017345551  0.0017345550

$se
Time Series:
Start = c(2015, 6)
End = c(2015, 35)
Frequency = 365
 [1] 0.6767278 0.7409938 0.7531257 0.7555161 0.7559908 0.7560852 0.7561040 0.7561077 0.7561084 0.7561086
[11] 0.7561086 0.7561086 0.7561086 0.7561086 0.7561086 0.7561086 0.7561086 0.7561086 0.7561086 0.7561086
[21] 0.7561086 0.7561086 0.7561086 0.7561086 0.7561086 0.7561086 0.7561086 0.7561086 0.7561086 0.7561086

> U = x.fore$pred+ 2*x.fore$se
> L = x.fore$pred-2*x.fore$se
> minx=min(dpricets,L)
> maxx=max(dpricets,U)
> ts.plot(dpricets, x.fore$pred,col=1:2, ylim=c(minx,maxx))
> lines(U, col="blue", lty="dashed")
> lines(L, col="blue", lty="dashed")
>
```

7. Conclusion and Future Work:

Significant relationship has been established between the factors and the price. Models has been developed and evaluated to closely predict the house prices. Time series analysis show that the future price can be predicted based on the past values.

Limitations:

More attributes can be added to understand and predict house prices in a improved manner.

Potential Improvements or Future Work:

Time series analysis can be done in a better way. Other algorithms can be used on the dataset and more models can be built and then these models can be compared with other ones to find the best model for predicting the price of a house.