SURVEY OF PROGRAMMING LANGUAGES

ITCS 5102 [FALL2016]

HOUSE PRICE ANALYSIS

PROJECT DOCUMENTATION

PRESENTED BY:

ADITYA GUPTA (800966229)
HOZEFA HAVELIWALA (800936900)
REKHANSH PANCHAL (800970541)

TABLE OF CONTENTS

OVERVIEW

The project focuses on predicting house prices by analyzing a data set using linear regression technique in R programming language. A house has several properties such as area, number of floors, condition, etc. upon which house prices are dependent. The values of these properties are given in the dataset. The cost of the house indicates how expensive the house is. Backward elimination is used to eliminate the insignificant variables and improve the model accuracy. Finally, we get all the properties on which the house prices depend upon.

LANGUAGE SELECTION

We have selected R programming language to proceed with this project because it provides a powerful and efficient platform for data analysis. It also allows the user to visualize the output in the form of graphs. Also, R is open source and easy to implement.

FEATURES OF THE LANGUAGE

- R is an interpreted language.
- It focuses on user-friendly data analysis and graphical models.
- It is possible to accomplish even complex analysis using only a few lines of code.
- R programming is easy to learn.
- It is used for data-wrangling.
- It is used to implement statistical techniques such as linear and nonlinear modelling.

FEATURES USED IN THE PROJECT

- Data cleaning.
- Random splitting of data.
- Generation of a linear regression model.
- Prediction of data using the above model.
- Data Visualization: Graphs to show relationships between different variables.

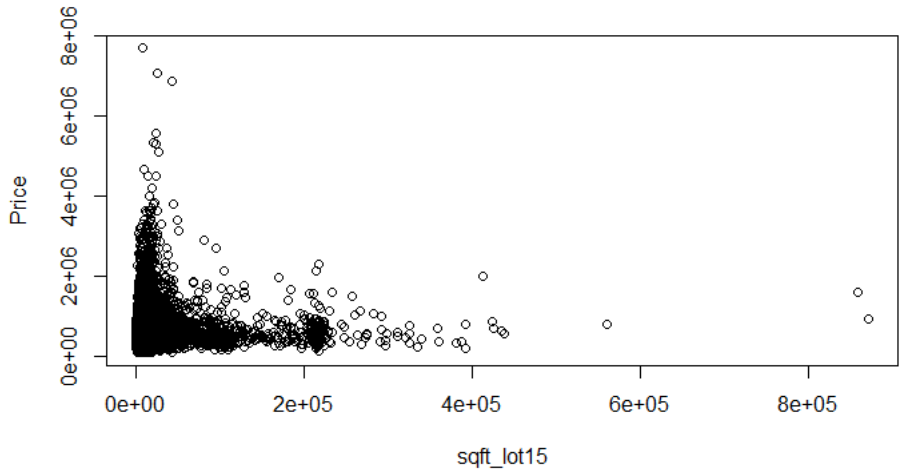
STEPS FOLLOWED FOR DATA ANALYSIS

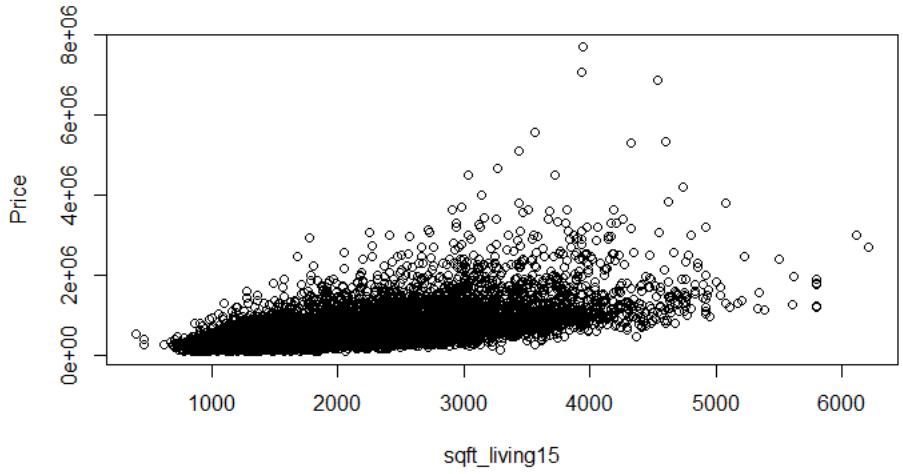
- Data input
- Data Cleaning
- Plotting Scatterplots
- Dataset splitting
- Correlation check
- Regression model
- Variable elimination
- Value Prediction
- Accuracy Check
- Visualization

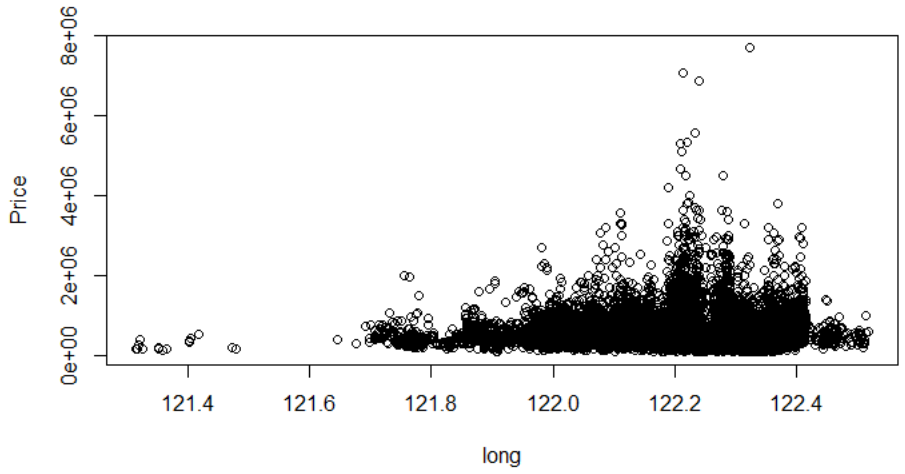
OUTPUT

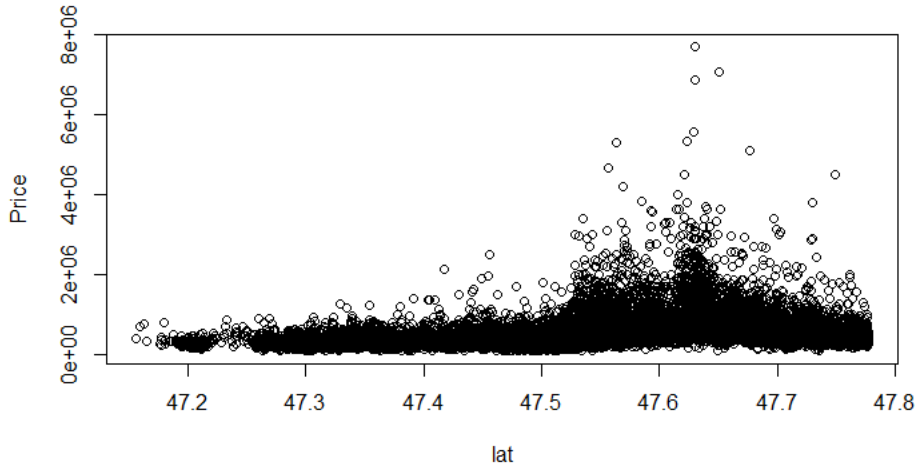
The following are the outputs of the analysis performed:

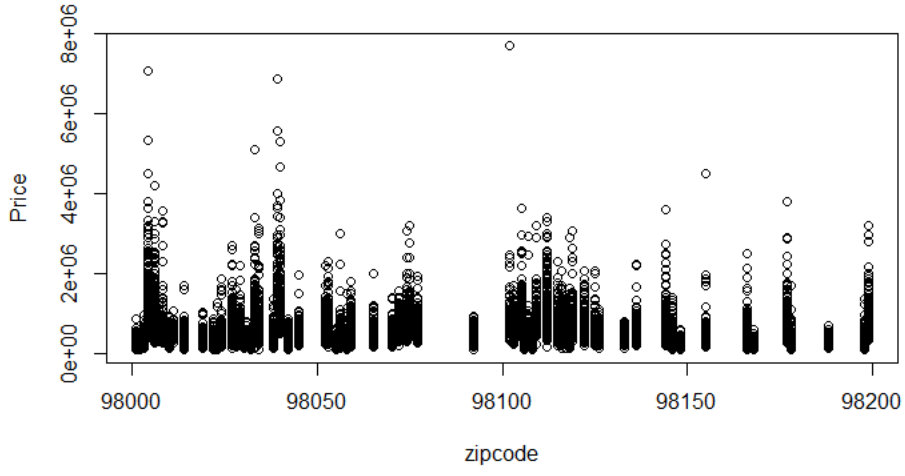
Scatterplots

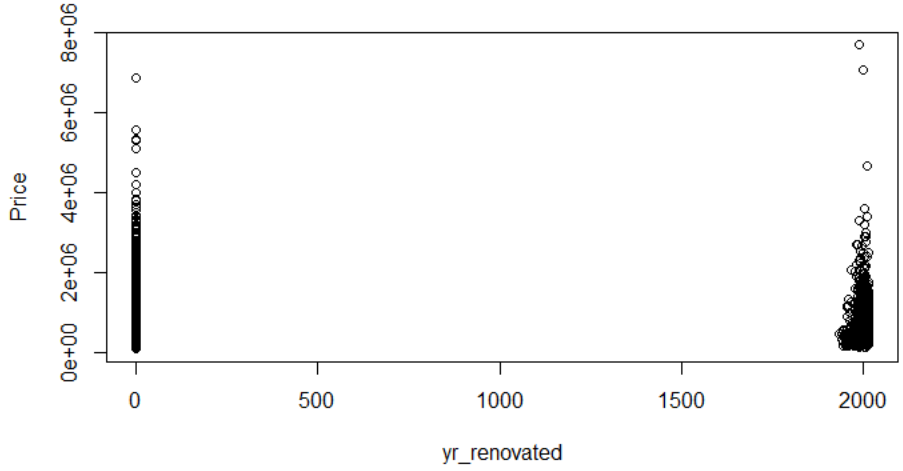


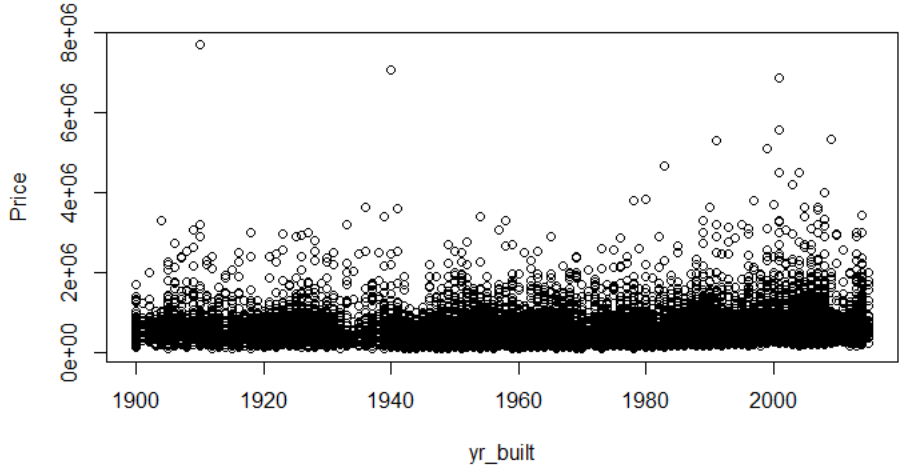


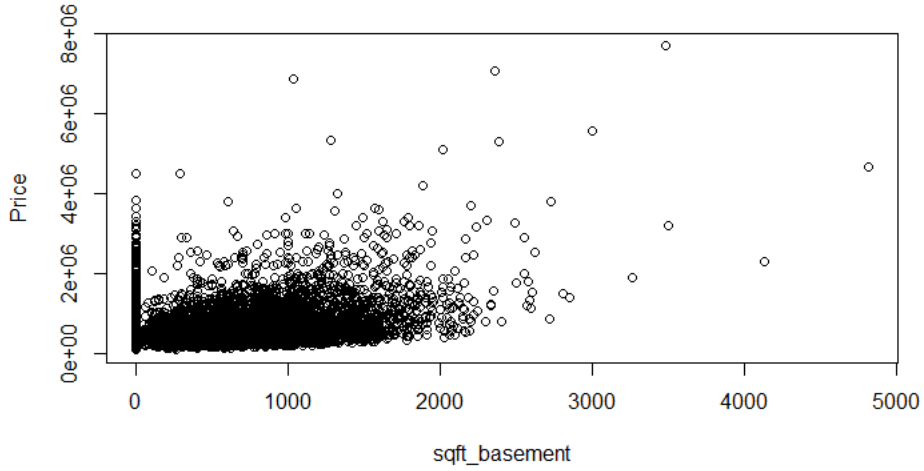

Price
0e+00
2e+06
4e+06
6e+06
8e+06
121.4
121.6
121.8
122.0
122.2
122.4
long

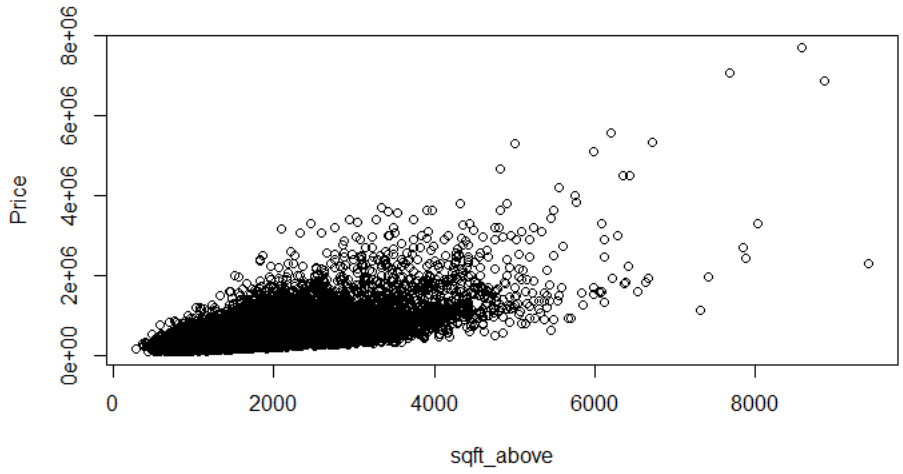

Price
0e+00
2e+06
4e+06
6e+06
8e+06
47.2
47.3
47.4
47.5
47.6
47.7
47.8
lat

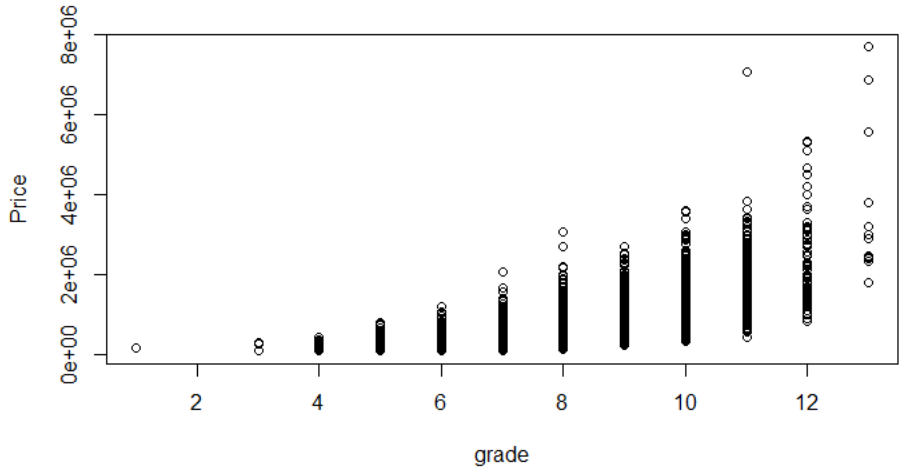

Price
0e+00
2e+06
4e+06
6e+06
8e+06
98000
98050
98100
98150
98200
zipcode

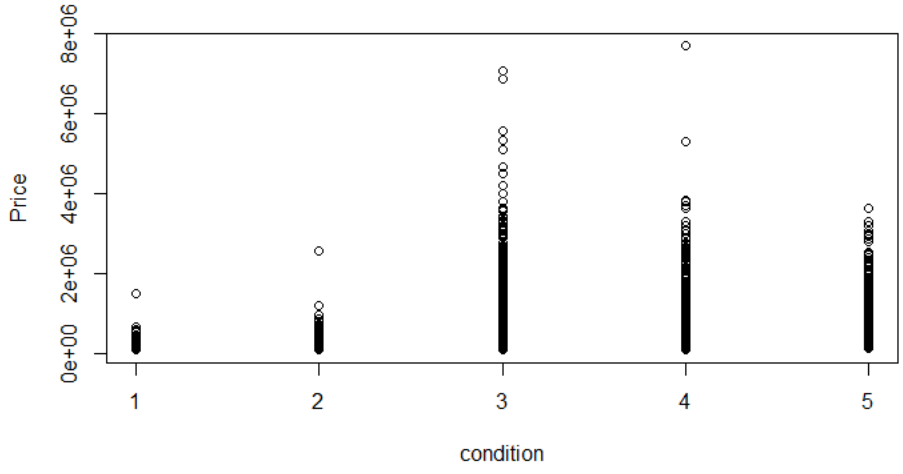

Price
0e+00
2e+06
4e+06
6e+06
8e+06
0
500
1000
1500
2000
yr_renovated

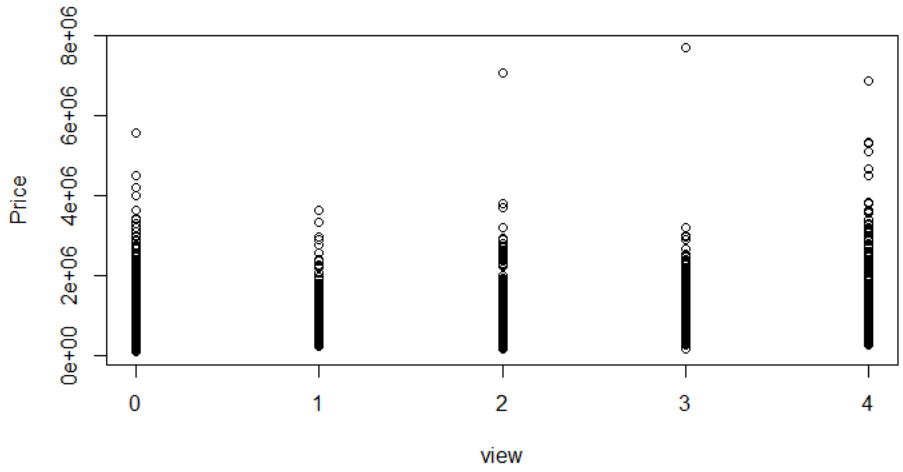

Price
0e+00
2e+06
4e+06
6e+06
8e+06
1900
1920
1940
1960
1980
2000
yr_built

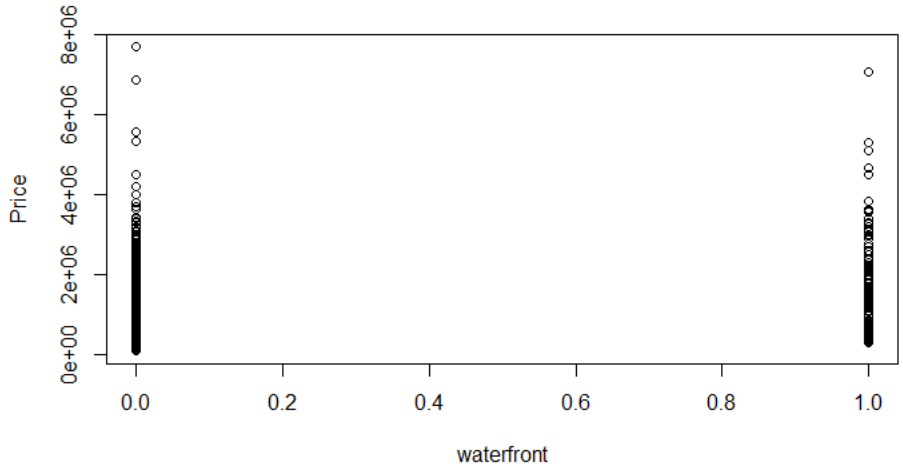

Price
0e+00
2e+06
4e+06
6e+06
8e+06
0
1000
2000
3000
4000
5000
sqft_basement

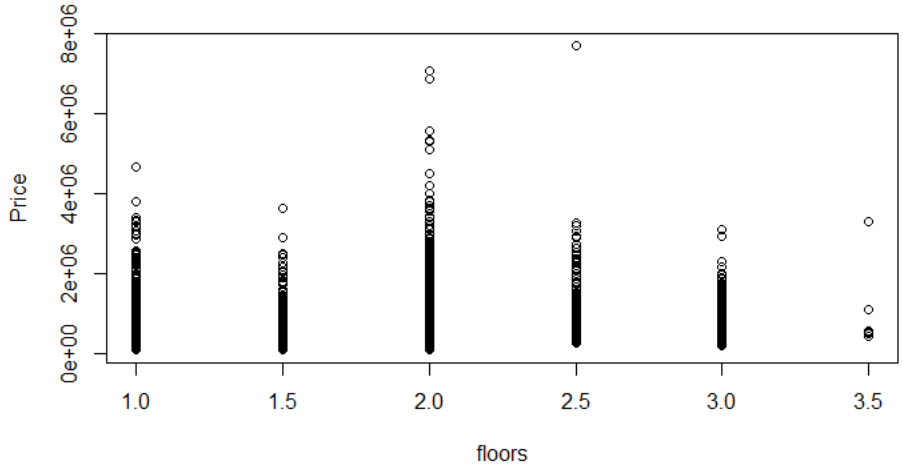

Price
0e+00
2e+06
4e+06
6e+06
8e+06
0
2000
4000
6000
8000
sqft_above

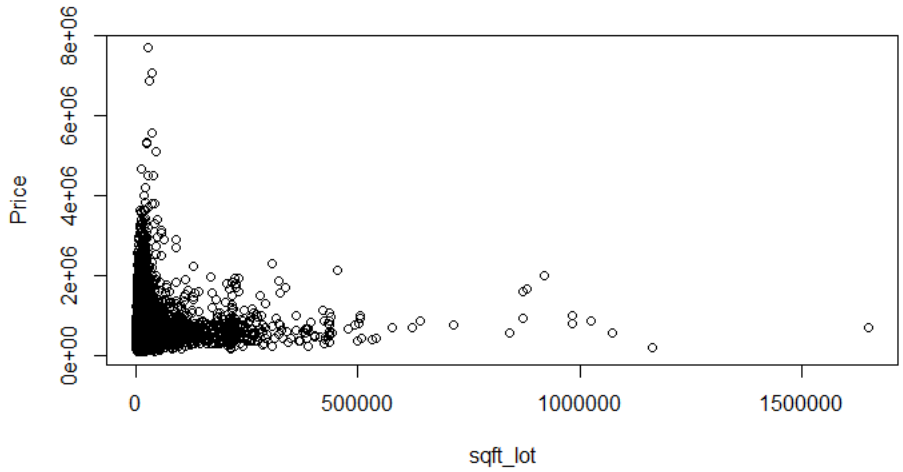

Price
0e+00
2e+06
4e+06
6e+06
8e+06
2
4
6
8
10
12
grade

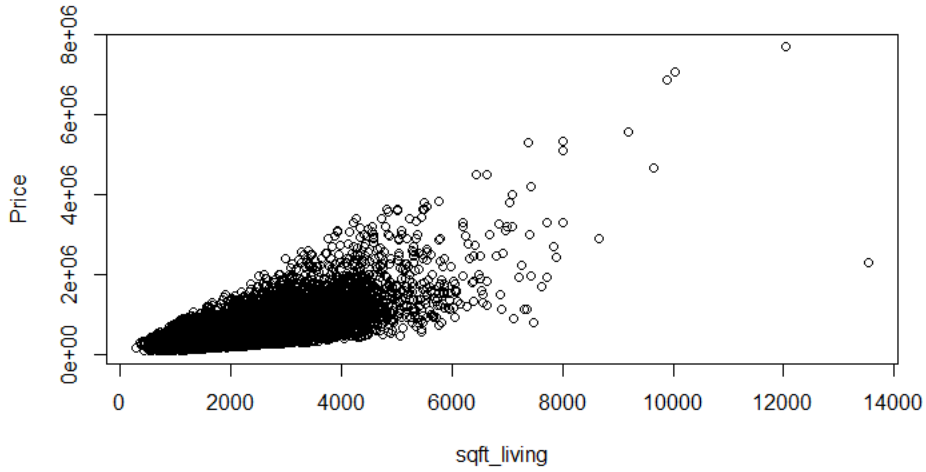

Price
0e+00
2e+06
4e+06
6e+06
8e+06
1
2
3
4
5
condition

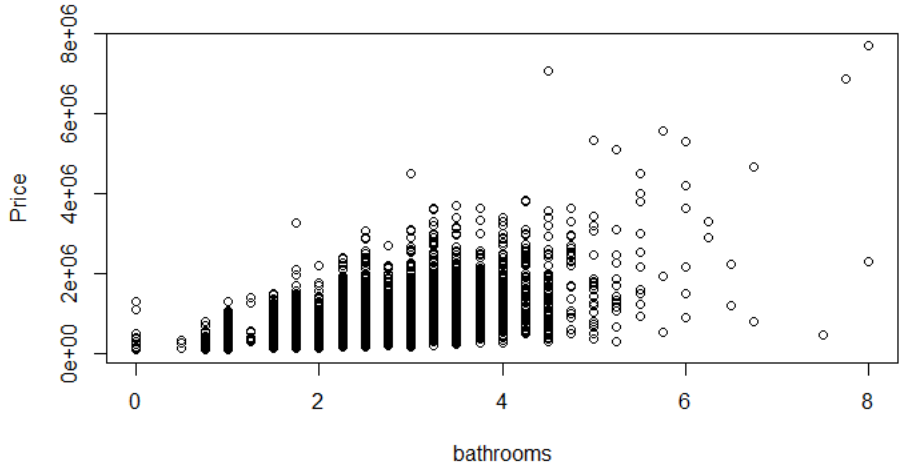

Price
0e+00
2e+06
4e+06
6e+06
8e+06
0
1
2
3
4
view

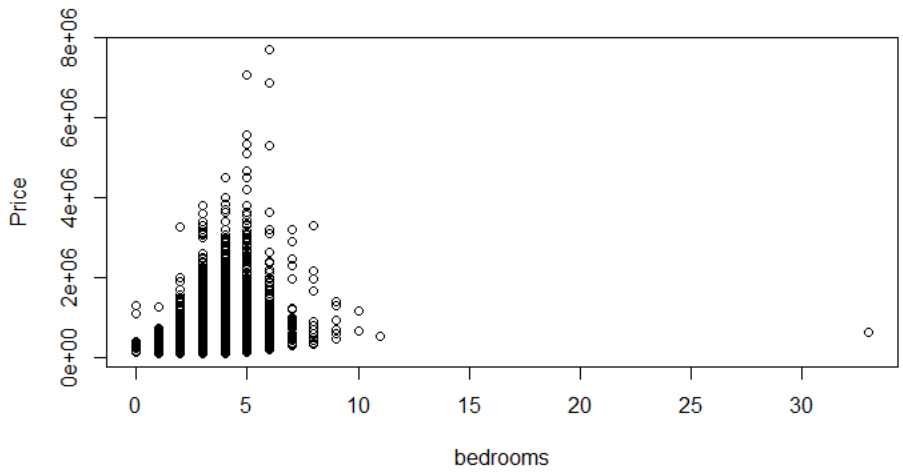

Price
0e+00
2e+06
4e+06
6e+06
8e+06
0.0
0.2
0.4
0.6
0.8
1.0
waterfront


Price
0e+00
2e+06
4e+06
6e+06
8e+06
1.0
1.5
2.0
2.5
3.0
3.5
floors


Price
0e+00
2e+06
4e+06
6e+06
8e+06
0
500000
1000000
1500000
sqft_lot


Price
0e+00
2e+06
4e+06
6e+06
8e+06
0
2000
4000
6000
8000
10000
12000
14000
sqft_living


Price
0e+00
2e+06
4e+06
6e+06
8e+06
0
2
4
6
8
bathrooms



Correlation Matrix

```
> cor(mydatasetcla)[1,]
                                  # Correlation matrix for Price
      price      bedrooms     bathrooms   sqft_living      sqft_lot        floors
 1.00000000    0.30834960    0.52513750    0.70203505    0.08966086    0.25679388
 waterfront          view     condition         grade    sqft_above sqft_basement
 0.26636943    0.39729349    0.03636179    0.66743425    0.60556730    0.32381602
   yr_built  yr_renovated       zipcode           lat          long sqft_living15
 0.05401153    0.12643379   -0.05320285    0.30700348   -0.02162624    0.58537890
 sqft_lot15
 0.08244715
```

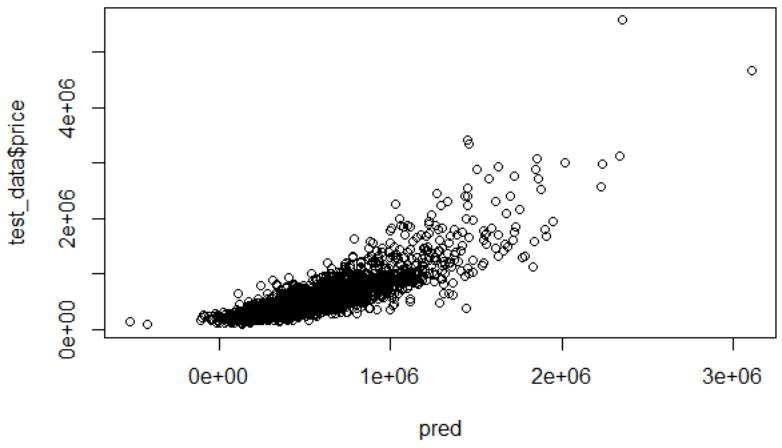
Summary of Linear Model

```
> summary(linearModel)

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

Residuals:
     Min       1Q   Median       3Q      Max
-1299082   -98454    -9610    77288  4325710

Coefficients: (1 not defined because of singularities)
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    6.100e+06  3.190e+06   1.912  0.05588 .
bedrooms      -3.523e+04  2.053e+03 -17.161  < 2e-16 ***
bathrooms      4.046e+04  3.531e+03  11.458  < 2e-16 ***
sqft_living    1.475e+02  4.784e+00  30.826  < 2e-16 ***
sqft_lot       1.413e-01  5.221e-02   2.706  0.00681 **
floors         5.016e+03  3.896e+03   1.287  0.19801
waterfront     5.808e+05  1.874e+04  31.001  < 2e-16 ***
view           5.415e+04  2.325e+03  23.287  < 2e-16 ***
condition      2.743e+04  2.549e+03  10.762  < 2e-16 ***
grade          9.648e+04  2.340e+03  41.237  < 2e-16 ***
sqft_above     3.504e+01  4.730e+00   7.408 1.34e-13 ***
sqft_basement         NA         NA      NA       NA
yr_built      -2.605e+03  7.912e+01 -32.928  < 2e-16 ***
yr_renovated   2.278e+01  4.010e+00   5.682 1.35e-08 ***
zipcode       -5.832e+02  3.590e+01 -16.246  < 2e-16 ***
lat            6.051e+05  1.165e+04  51.939  < 2e-16 ***
long           2.189e+05  1.429e+04  15.323  < 2e-16 ***
sqft_living15  2.074e+01  3.743e+00   5.540 3.07e-08 ***
sqft_lot15    -3.947e-01  7.913e-02  -4.987 6.18e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 201500 on 18397 degrees of freedom
Multiple R-squared:  0.6994,    Adjusted R-squared:  0.6992
F-statistic:  2518 on 17 and 18397 DF,  p-value: < 2.2e-16
```

Updated Linear Model

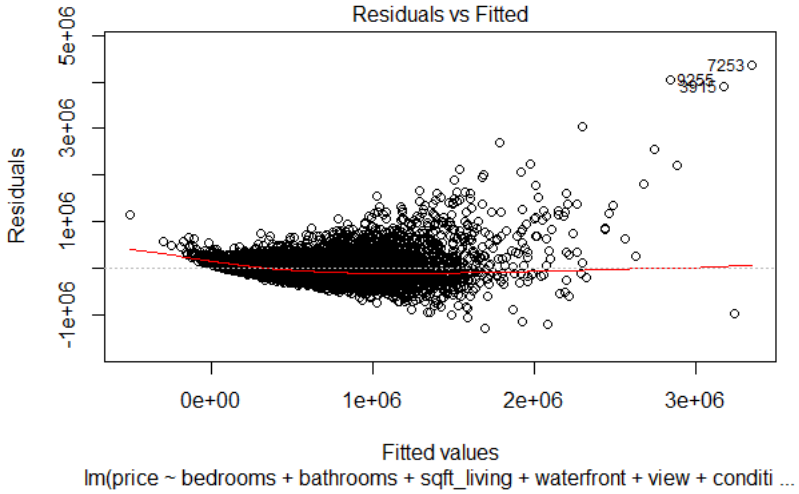
```
> linearModel <- update(linearModel, .~.-sqft_basement-floors-sqft_lot-sqft_lot15-yr_renovated)
> summary(linearModel)

Call:
lm(formula = price ~ bedrooms + bathrooms + sqft_living + waterfront +
    view + condition + grade + sqft_above + yr_built + zipcode +
    lat + long + sqft_living15, data = training_data)

Residuals:
     Min       1Q   Median       3Q      Max
-1268482   -98949    -9804    77540  4356203

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    5.381e+06  3.115e+06   1.727   0.0841 .
bedrooms      -3.499e+04  2.046e+03 -17.099  < 2e-16 ***
bathrooms      4.541e+04  3.371e+03  13.470  < 2e-16 ***
sqft_living    1.437e+02  4.546e+00  31.616  < 2e-16 ***
waterfront     5.853e+05  1.873e+04  31.249  < 2e-16 ***
view           5.440e+04  2.324e+03  23.414  < 2e-16 ***
condition      2.447e+04  2.508e+03   9.758  < 2e-16 ***
grade          9.767e+04  2.331e+03  41.901  < 2e-16 ***
sqft_above     3.802e+01  4.249e+00   8.947  < 2e-16 ***
yr_built      -2.721e+03  7.308e+01 -37.239  < 2e-16 ***
zipcode       -5.837e+02  3.578e+01 -16.313  < 2e-16 ***
lat            6.061e+05  1.159e+04  52.307  < 2e-16 ***
long           2.268e+05  1.396e+04  16.247  < 2e-16 ***
sqft_living15  1.836e+01  3.713e+00   4.945 7.69e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 201800 on 18401 degrees of freedom
Multiple R-squared:  0.6984,    Adjusted R-squared:  0.6982
F-statistic:  3278 on 13 and 18401 DF,  p-value: < 2.2e-16
```

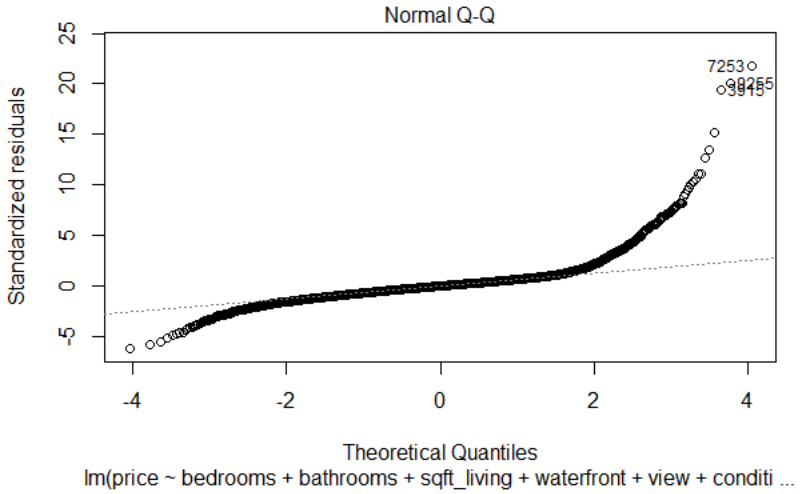
```
> plot(pred, test_data$price)
```

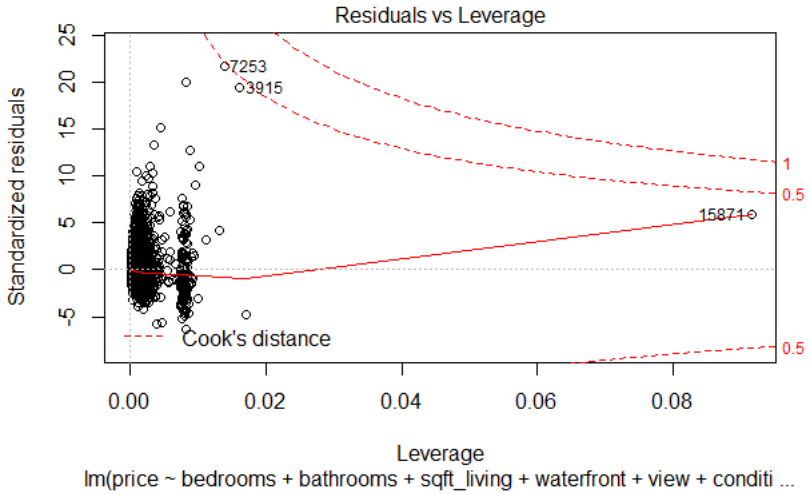


Plots of Linear Model properties

```
> plot(linearModel)
```

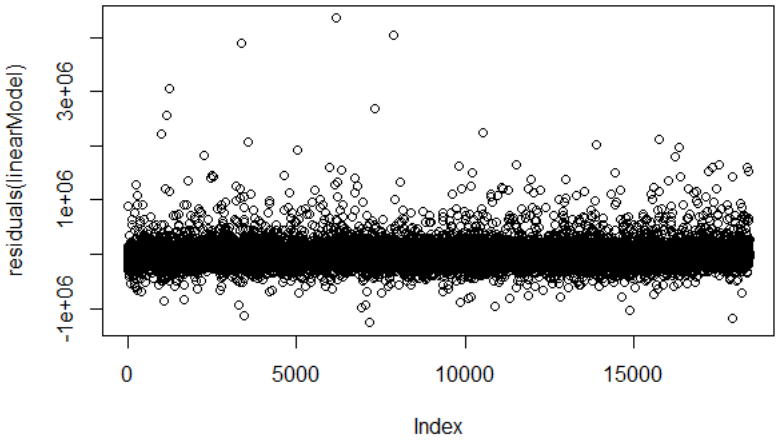



Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(price ~ bedrooms + bathrooms + sqft_living + waterfront + view + conditi ...
7253
3915

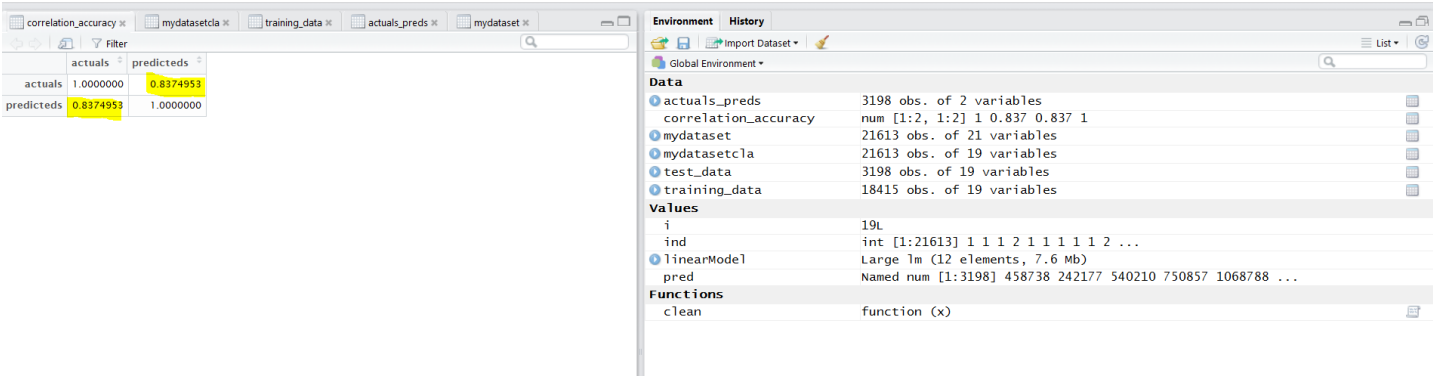

Residuals vs Leverage
Standardized residuals
Leverage
lm(price ~ bedrooms + bathrooms + sqft_living + waterfront + view + conditi ...
Cook's distance
7253
3915
15871
1
0.5
0.5

Plotting Residuals of Linear Model

```
> plot(residuals(linearModel))
```



Accuracy with Environment Variables



CONCLUSION

After analyzing the data taken, we found that out of the 18 variables, 5 variables were not significant enough to predict the house prices. The final equation to predict the house price is as below:

House Price = 5.381e+06 -3.499e+04(bedrooms) + 4.541e+04(bathrooms) + 1.437e+02(sqft_living) + 5.853e+05(waterfront) + 5.440e+04(view) + 2.447e+04(condition) + 9.767e+04(grade) + 3.802e+01(sqft_above) – 2.721e+03(year_built) – 5.837e+02(zip_code) + 6.061e+05(lat) + 2.268e+05(long) + 1.836e+01(sqft_living15)

RISK AND ISSUE MANAGEMENT

POTENTIAL EXCEPTIONS AND PROBLEMS

- Linear regression model is not always applicable.
- Dataset may include more than one dependent/repeated collinear-variables leading to singularities.

APPROPRIATE CORRECTIVE MEASURES

- One needs to check for non-linear regression model to improve accuracy.
- Dependent/Repeated collinear-variables can be predicted using another variable. Hence, one can ignore those values to avoid singularity.

REFERENCES

- https://d396qusza40orc.cloudfront.net/phoenixassets/home_data.csv