Housing_Price

Yeshwanthi

6/6/2020

##Data Loading##

Loading crime data

```
crimedata <- read.table("C:/Users/Deepak Yesh/Documents/Final-Project-
DA/crime.csv", header=TRUE,sep=",")
```

Loading distance data

```
distancedata <- read.table("C:/Users/Deepak Yesh/Documents/Final-Project-
DA/amazon.csv",header=TRUE,sep=",")
```

Loading housing data

```
housingdata <- read.table("C:/Users/Deepak Yesh/Documents/Final-Project-
DA/initial-housingdata.csv",header=TRUE,sep=",")
```

##Data Merging##

Merging crime and distance data and storing in distancecrime dataframe

```
distancecrime <- merge(distancedata,crimedata, by ="zipcode")
```

Merging distance crimem data with housing data and storing in fulldata data frame

```
fulldata <- merge(distancecrime,housingdata, by="zipcode")
```

##Data Cleaning##

Stripping out comma and $ form Income

```
fulldata$Income= as.numeric(gsub("\\$|,","",as.character(fulldata$Income)))
```

Stripping out comma from population

```
fulldata$population=
as.numeric(gsub("\\$|,","",as.character(fulldata$population)))
```

##Derived variables##

Deriving year from date

```
fulldata$year<-substr(fulldata$date,1,4)
```

Deriving new living area square feet based on year of renovation

```
fulldata$newsqft_living <- 
ifelse(fulldata$year==2015,fulldata$sqft_living15,fulldata$sqft_living)
```

Deriving new lot area square feet based on year of renovation

```
fulldata$newsqft_lot <- 
ifelse(fulldata$year==2015,fulldata$sqft_lot15,fulldata$sqft_lot)
```

Deriving crime rate based on population

```
fulldata$crimerate <-fulldata$crime.count/fulldata$population
fulldata$crimerate <- round(fulldata$crimerate,2)
```

##Analysing data issue problem##

```
Rooms_Issue = fulldata$bedrooms
Rooms_Issue.freq = table(Rooms_Issue)
Rooms_Issue.freq

## Rooms_Issue
##    0    1    2    3    4    5    6    7    8    9   10   11   33 
##   13  199 2760 9824 6882 1601  272   38   13    6    3    1    1 
```

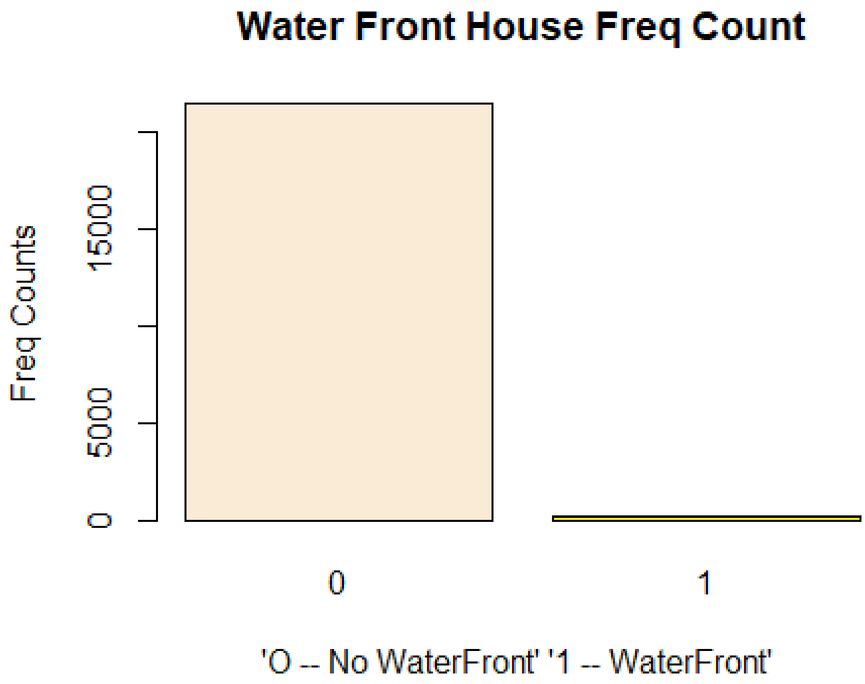
Removed data entry issue from the data set Bed room is 33 but square feet is very less

```
fulldata[which(fulldata$bedrooms == 33),]

##       zipcode Distance Income time population crime.count burglary theft
b.t
## 13247   98103     3.23  84069    8      45911          31        3     5
8
##                id             date Date.only  price bedrooms bathrooms
## 13247 2402100895 20140625T000000  20140625 640000       33      1.75
##       sqft_living sqft_lot floors waterfront view condition grade
sqft_above
## 13247        1620     6000      1          0    0         5     7
1040
##       sqft_basement yr_built yr_renovated     lat     long sqft_living15
## 13247           580     1947            0 47.6878 -122.331          1330
##       sqft_lot15 Age year newsqft_living newsqft_lot crimerate
## 13247       4700  71 2014           1620        6000         0

fulldata <- fulldata[-13247,]
```

##Exploratory Analysis for X Variables##

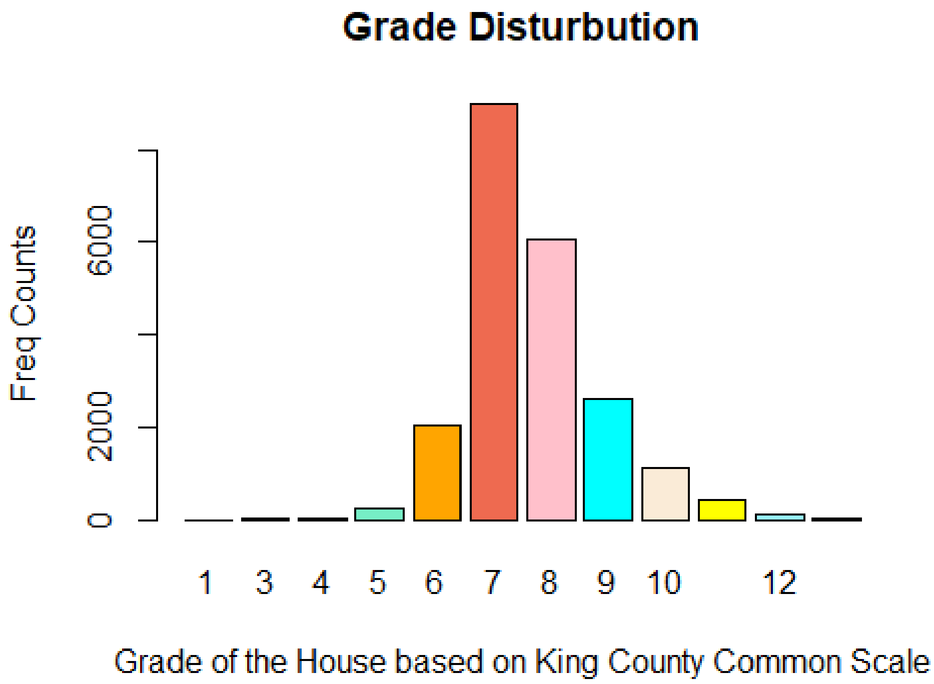
Water front

```
library(plyr)
library(MASS)
waterfrong = fulldata$waterfront
waterfrong.freq = table(waterfrong)
colors = c("antiquewhite", "yellow", "green", "violet",
           "orange", "blue", "pink", "cyan")
barplot((waterfrong.freq),main="Water Front House Freq Count",
        xlab="'O -- No WaterFront' '1 -- WaterFront' ",
        ylab="Freq Counts",col=colors)
```



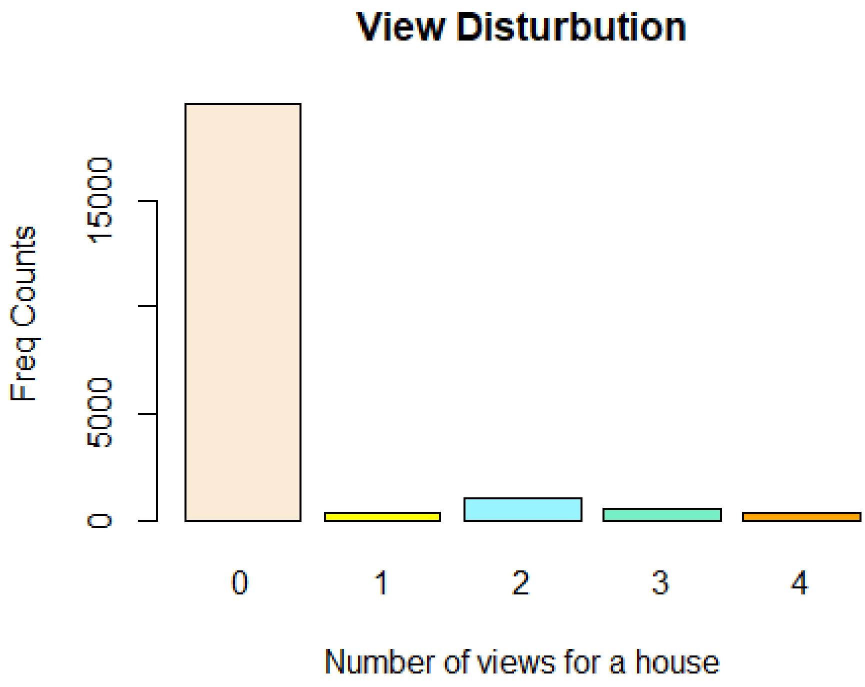
Grade

```
Gradeb = fulldata$grade
Gradeb.freq = table(Gradeb)
Gradeb.freq

## Gradeb
##    1    3    4    5    6    7    8    9   10   11   12   13
##    1    3   29  242 2038 8980 6068 2615 1134  399   90   13

colors = c("antiquewhite", "yellow", "cadetblue1", "aquamarine2",
           "orange", "coral2", "pink", "cyan")
barplot((Gradeb.freq),main="Grade Disturbution",
        xlab="Grade of the House based on King County Common Scale",
      ylab="Freq Counts",col=colors)
```



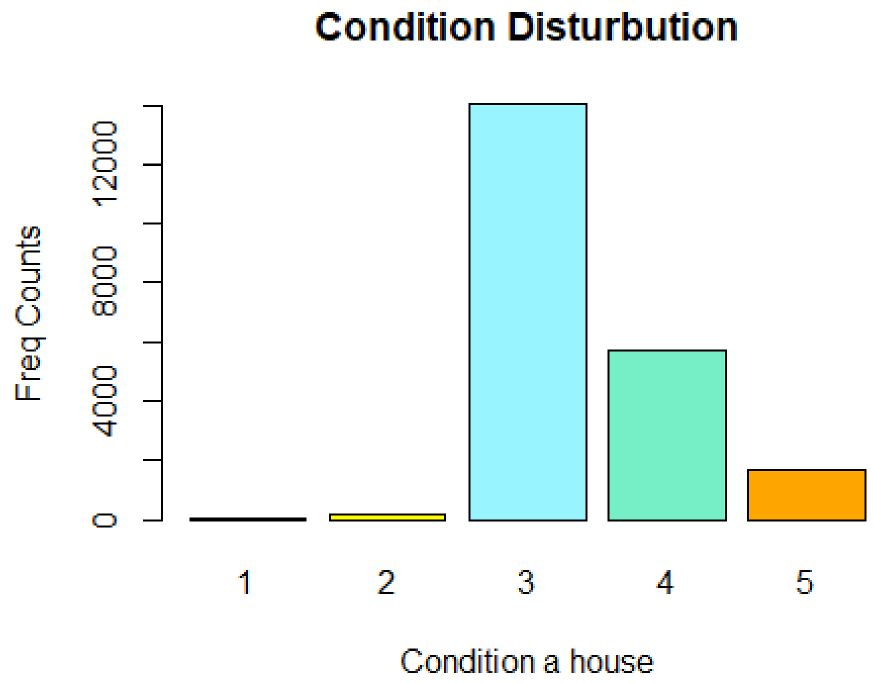
View

```
View1 = fulldata$view
View1.freq = table(View1)
View1.freq

## View1
##     0     1     2     3     4 
## 19488   332   963   510   319

colors = c("antiquewhite", "yellow", "cadetblue1", "aquamarine2",
           "orange")
barplot((View1.freq),main="View Disturbution",
        xlab="Number of views for a house",
        ylab="Freq Counts",col=colors)
```



Condition

```
condition1 = fulldata$condition
condition1.freq = table(condition1)
condition1.freq

## condition1
##     1     2     3     4     5 
##    30   172 14031  5679  1700

colors = c("antiquewhite", "yellow", "cadetblue1", "aquamarine2",
           "orange")
barplot((condition1.freq),main="Condition Disturbution",
        xlab="Condition a house",
        ylab="Freq Counts",col=colors)
```



##Analysing the frequency of ordinal variables##

```
library(car)

## Loading required package: carData

count(mtcars,'fulldata$view')

##   fulldata.view  freq
## 1             0 19488
## 2             1   332
## 3             2   963
## 4             3   510
## 5             4   319

count(mtcars,'fulldata$waterfront')

##   fulldata.waterfront  freq
## 1                   0 21449
## 2                   1   163

count(mtcars,'fulldata$condition')

##   fulldata.condition  freq
## 1                  1    30
## 2                  2   172
## 3                  3 14031
## 4                  4  5679
## 5                  5  1700
```

```
count(mtcars,'fulldata$grade')
```

```
##    fulldata.grade freq
## 1               1    1
## 2               3    3
## 3               4   29
## 4               5  242
## 5               6 2038
## 6               7 8980
## 7               8 6068
## 8               9 2615
## 9              10 1134
## 10             11  399
## 11             12   90
## 12             13   13
```

```
count(mtcars,'fulldata$floors')
```

```
##   fulldata.floors  freq
## 1             1.0 10679
## 2             1.5  1910
## 3             2.0  8241
## 4             2.5   161
## 5             3.0   613
## 6             3.5     8
```

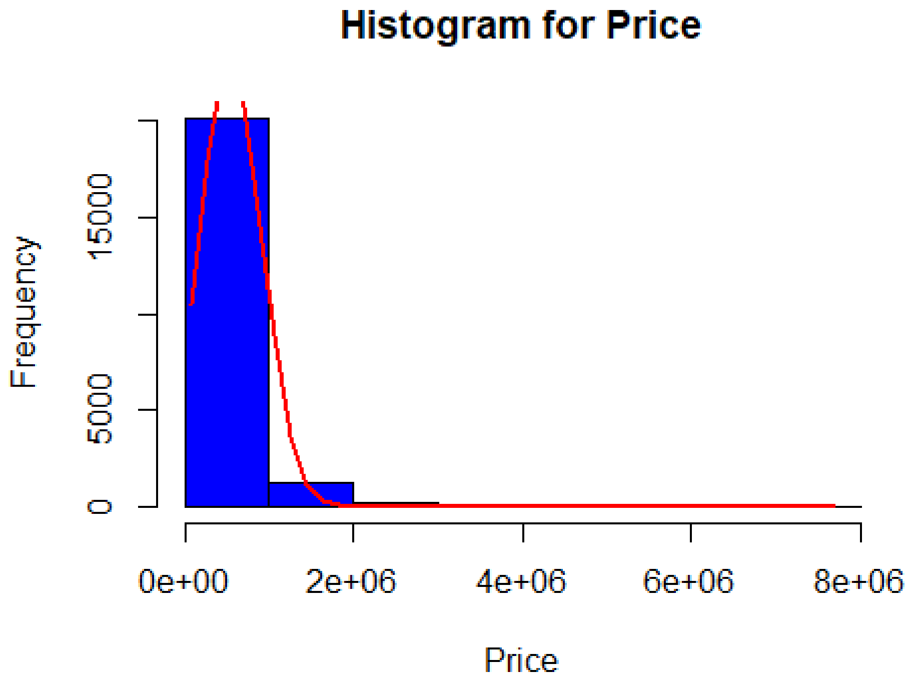
```
count(mtcars,'fulldata$bedroom')
```

```
##    fulldata.bedroom freq
## 1                 0   13
## 2                 1  199
## 3                 2 2760
## 4                 3 9824
## 5                 4 6882
## 6                 5 1601
## 7                 6  272
## 8                 7   38
## 9                 8   13
## 10                9    6
## 11               10    3
## 12               11    1
```

##Exploratory Anlysis for y variable price##

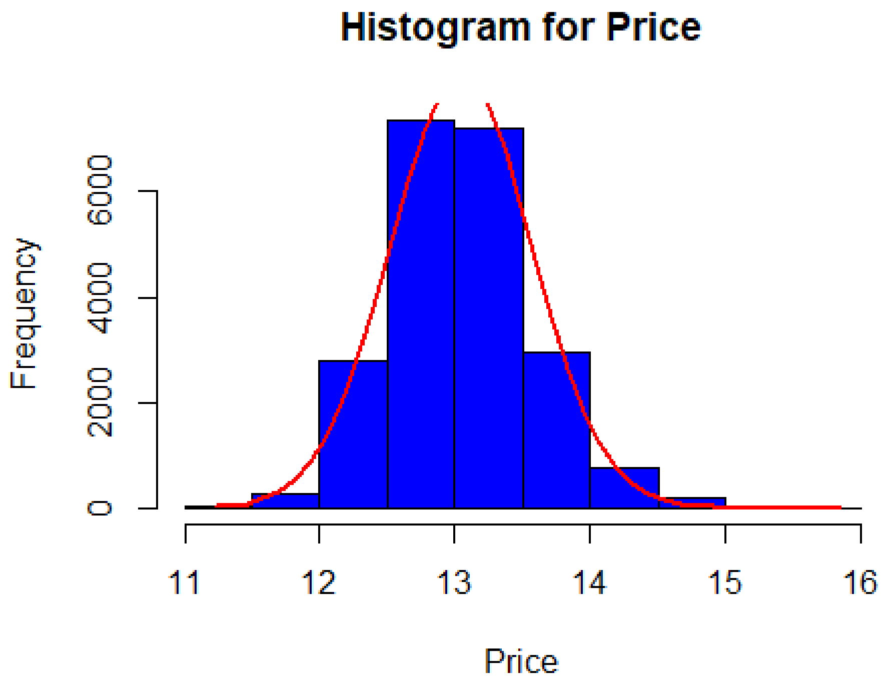
```
x<-fulldata$price
h<-hist(x, breaks=10, col="Blue", xlab="Price",
        main="Histogram for Price")
xfit<-seq(min(x),max(x),length=40)
yfit<-dnorm(xfit,mean=mean(x),sd=sd(x))
```

```
yfit <- yfit*diff(h$mids[1:2])*length(x)
lines(xfit, yfit, col="Red", lwd=2)
```



Applying log transformations on y variable

```
x<-log(fulldata$price)
h<-hist(x, breaks=10, col="Blue", xlab="Price",
        main="Histogram for Price")
xfit<-seq(min(x),max(x),length=40)
yfit<-dnorm(xfit,mean=mean(x),sd=sd(x))
yfit <- yfit*diff(h$mids[1:2])*length(x)
lines(xfit, yfit, col="Red", lwd=2)
```

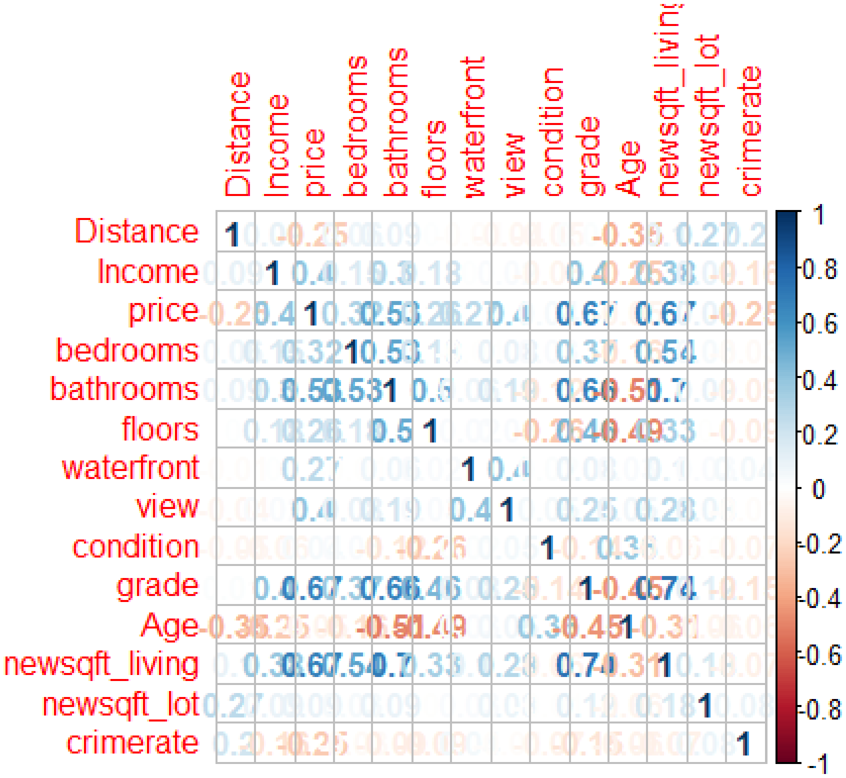


##Extracting the required variables to fulldata##

```
data1<-fulldata[c(2,3,13,14,15,18,19,20,21,22,31,33,34,35)]
```

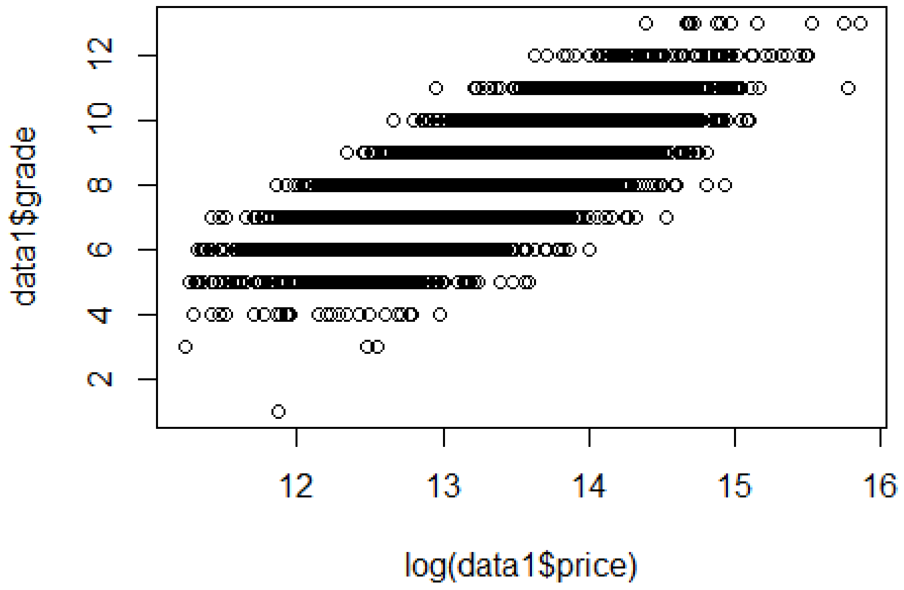
##Pearson correlation matrix##

```
library(corrplot)

## corrplot 0.84 loaded

c<-cor(data1)
corrplot(c,method="number")
```

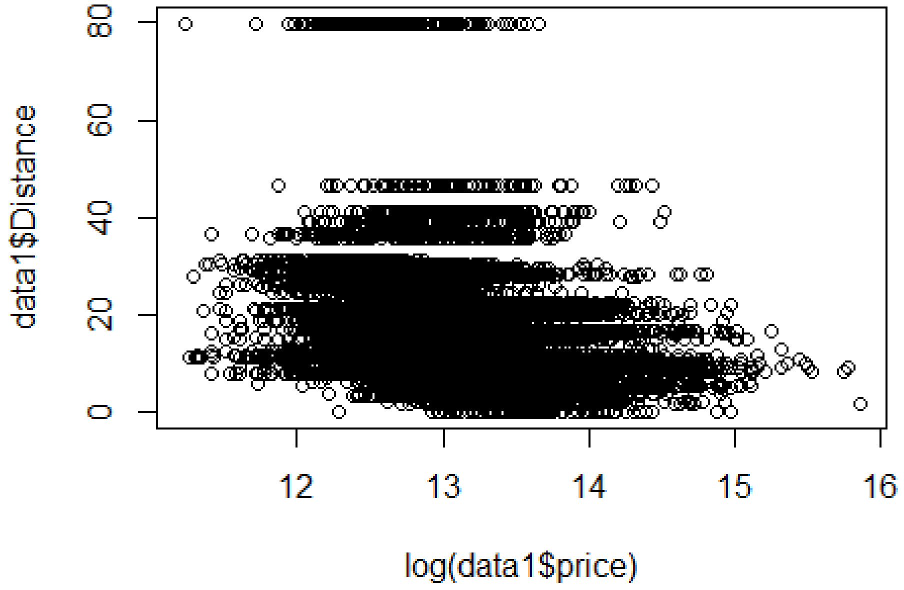


##Scatter plot for each X and Variable ##

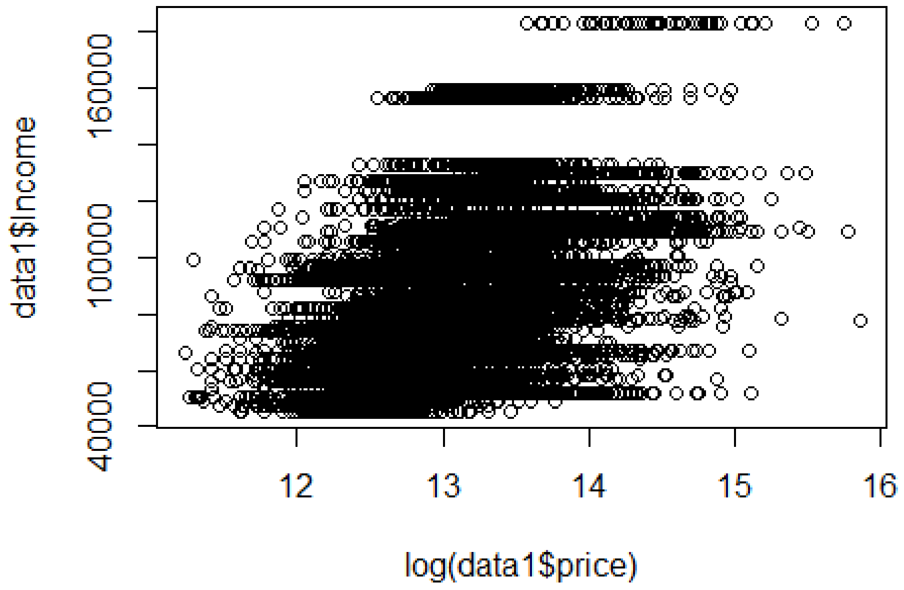
```
library(car)
plot(log(data1$price),data1$grade)
```



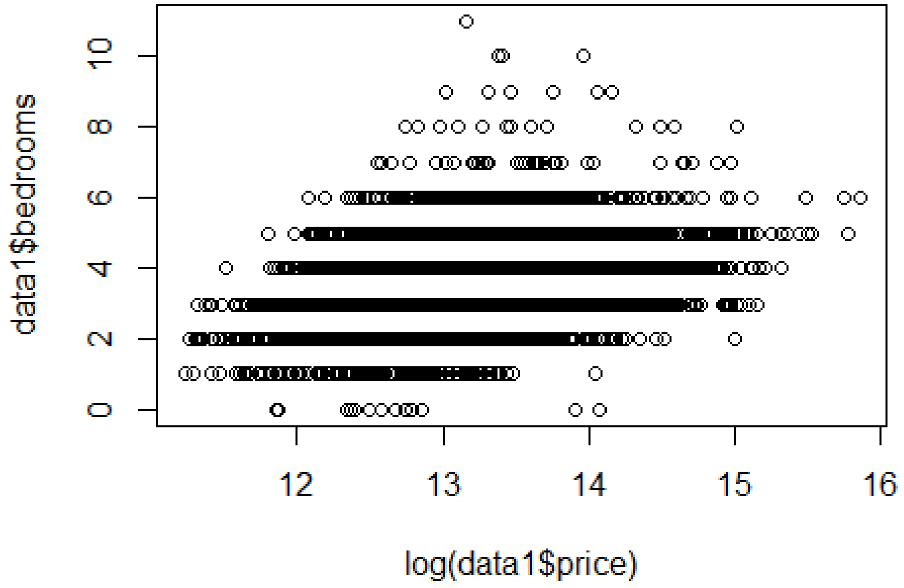
```
plot(log(data1$price),data1$Distance)
```



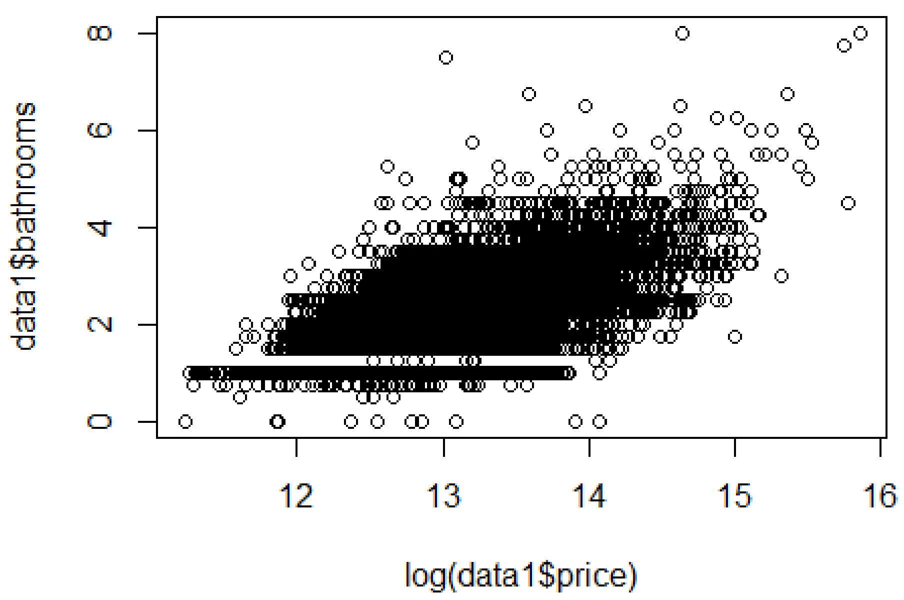
```
plot(log(data1$price),data1$Income)
```



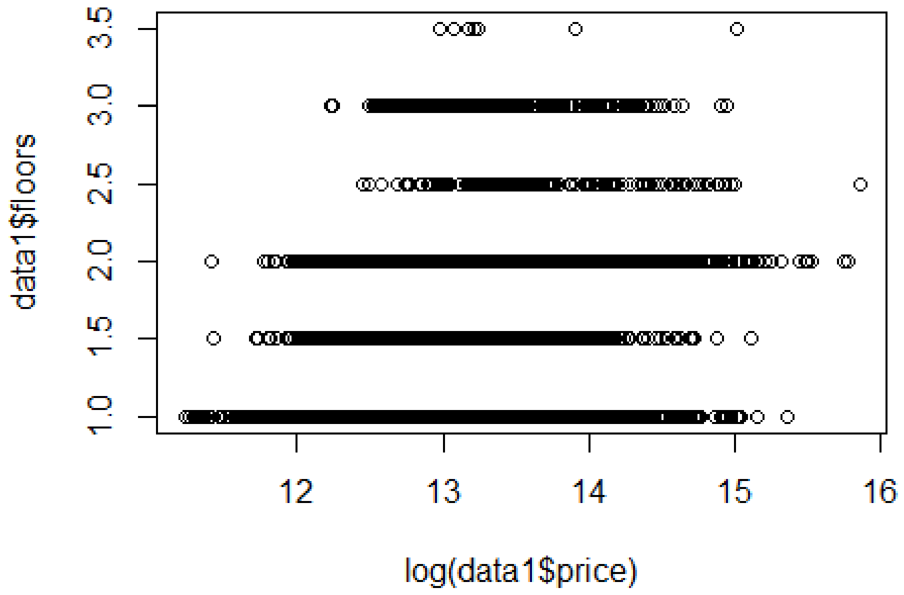
```
plot(log(data1$price),data1$bedrooms)
```



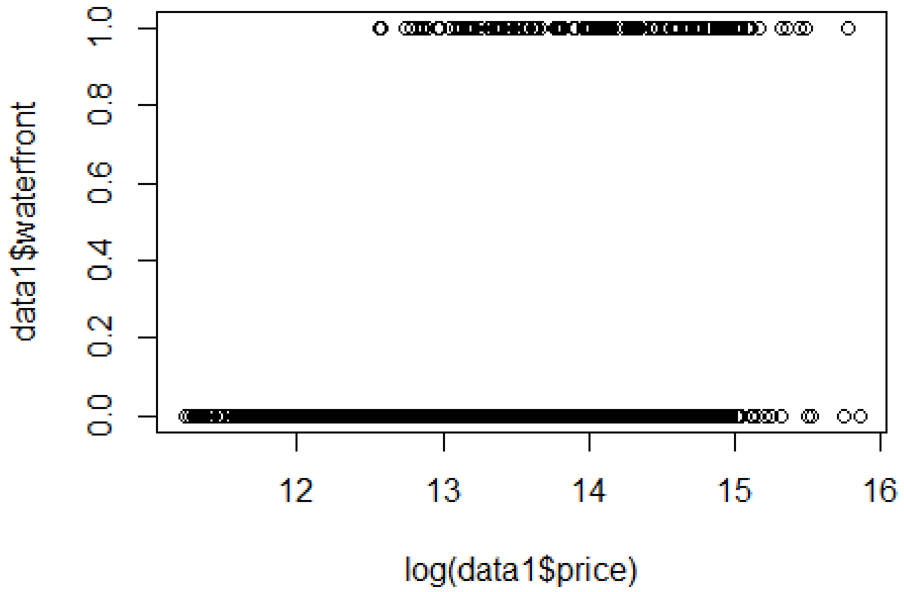
```
plot(log(data1$price),data1$bathrooms)
```



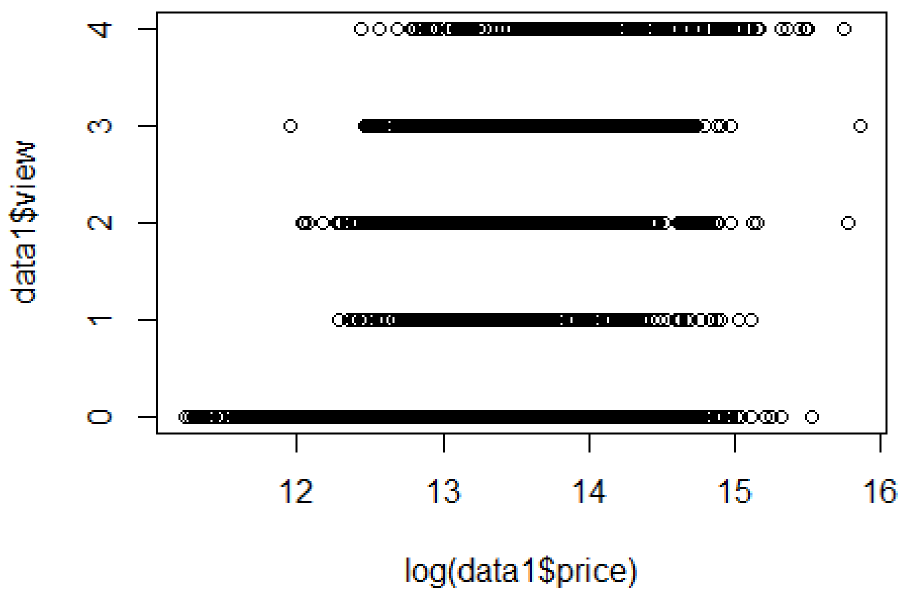
```
plot(log(data1$price),data1$floors)
```



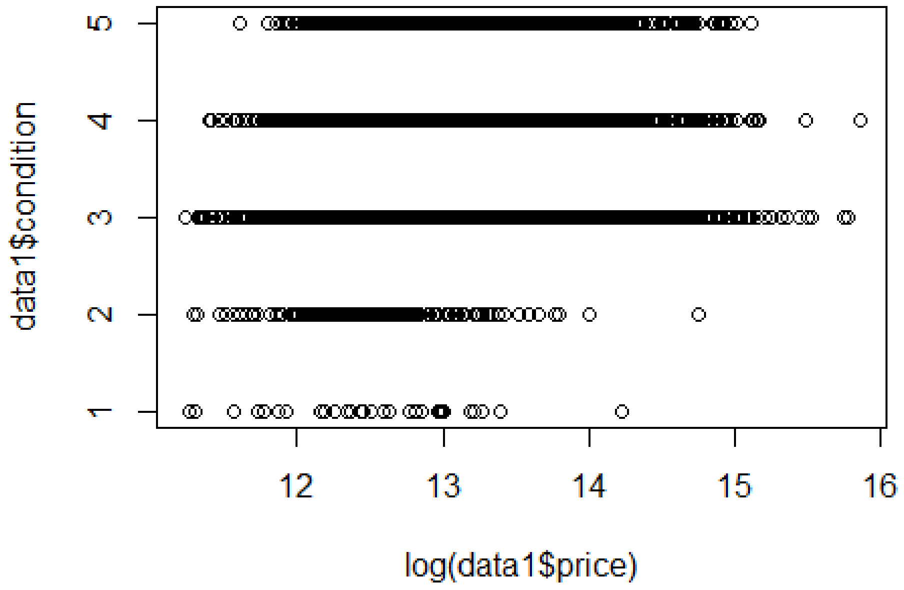
```
plot(log(data1$price),data1$waterfront)
```



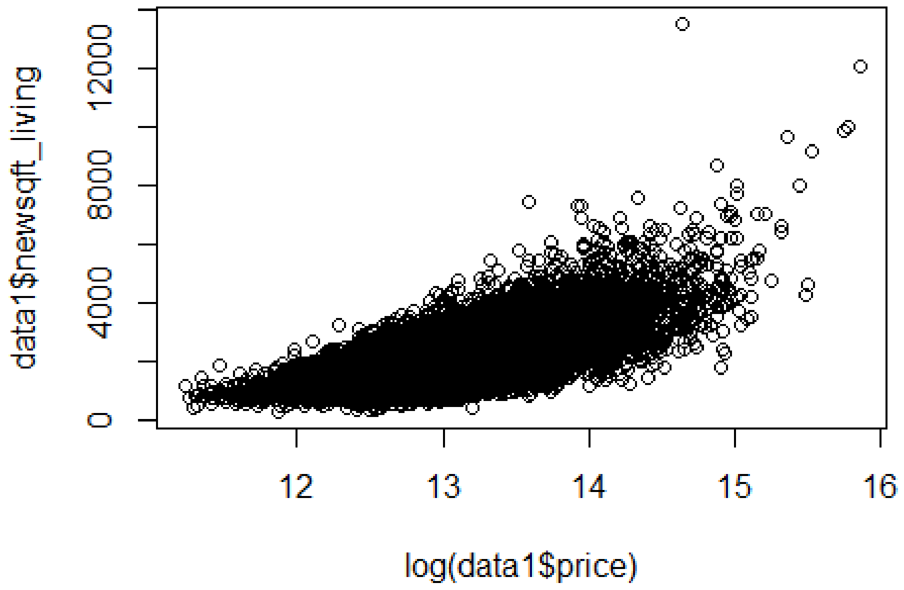
```
plot(log(data1$price),data1$view)
```



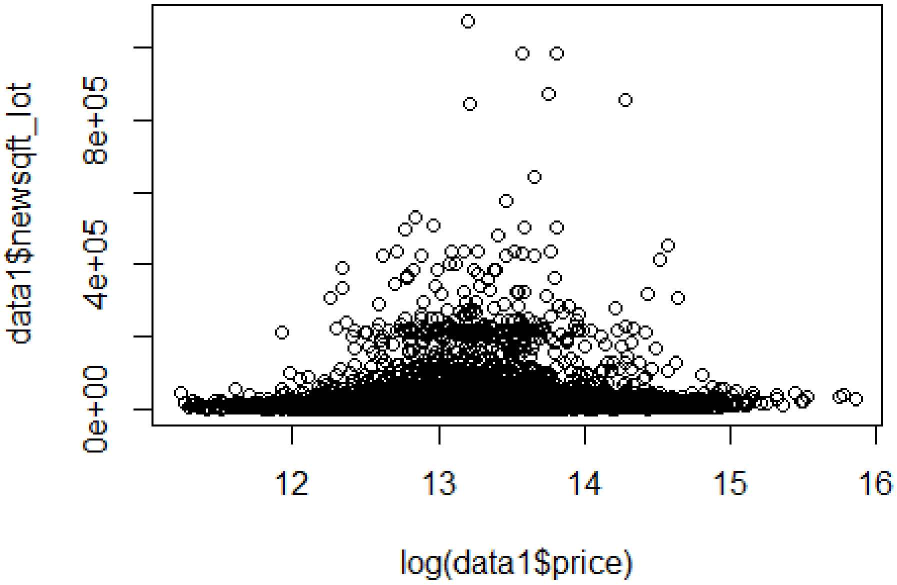
```
plot(log(data1$price),data1$condition)
```



```
plot(log(data1$price),data1$newsqft_living)
```



```
plot(log(data1$price),data1$newsqft_lot)
```



##Splitting Training and Testing Data##

```
selectfulldata = sample(1:nrow(fulldata),0.80*nrow(fulldata));
train1.data = data1[selectfulldata,];
test1.data = data1[-selectfulldata,];
str(train1.data)

## 'data.frame':    17289 obs. of  14 variables:
##  $ Distance      : num  3.23 9.13 18.49 10.06 21.41 ...
##  $ Income        : num  84069 56745 75695 129348 95717 ...
##  $ price         : int  615000 276000 359000 800000 302000 307635 147000
250000 457000 354000 ...
##  $ bedrooms      : int  4 2 4 4 4 3 3 4 4 3 ...
##  $ bathrooms     : num  1 2 2.5 2.25 3 2.5 1.5 1.75 2.5 2.5 ...
##  $ floors        : num  1.5 1 2 1 1 2 1 1 2 2 ...
##  $ waterfront    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ view          : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ condition     : int  4 5 3 2 3 3 4 4 3 3 ...
##  $ grade         : int  6 7 7 7 7 8 7 7 9 8 ...
##  $ Age           : int  117 99 16 66 55 4 54 41 5 19 ...
##  $ newsqft_living: int  1330 1480 2160 2350 3320 1820 1600 1860 2820 1990
...
##  $ newsqft_lot   : int  4400 6075 4496 10140 13500 4200 9619 7350 6983
15817 ...
##  $ crimerate     : num  0 0 0 0 0.02 0.04 0 0 0.03 0 ...
```

##Linear Regression Model##

Removed non significiant variables -condition,squareftlot,view,waterfront

```
model1 <- lm(formula=log(price)~bathrooms+grade+bedrooms+floors
         +newsqft_living+crimerate+Distance,data=train1.data)
summary(model1)

##
## Call:
## lm(formula = log(price) ~ bathrooms + grade + bedrooms + floors +
##     newsqft_living + crimerate + Distance, data = train1.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.28738 -0.18449  0.00514  0.17653  1.53848
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.156e+01  1.929e-02 599.528  < 2e-16 ***
## bathrooms       4.815e-02  4.668e-03  10.314  < 2e-16 ***
## grade           1.676e-01  3.101e-03  54.053  < 2e-16 ***
## bedrooms       -2.467e-03  3.028e-03  -0.815    0.415
## floors         -3.621e-02  4.896e-03  -7.396 1.47e-13 ***
## newsqft_living  2.333e-04  4.485e-06  52.008  < 2e-16 ***
## crimerate      -4.645e+00  1.174e-01 -39.549  < 2e-16 ***
## Distance       -1.447e-02  1.958e-04 -73.916  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2908 on 17281 degrees of freedom
## Multiple R-squared:  0.6935, Adjusted R-squared:  0.6934
## F-statistic:  5587 on 7 and 17281 DF,  p-value: < 2.2e-16
```

Removing bedroms as it is not significant

```
model2 <- lm(formula=log(price)~bathrooms+grade+floors
           +newsqft_living+crimerate+Distance,data=train1.data)
summary(model2)

##
## Call:
## lm(formula = log(price) ~ bathrooms + grade + floors + newsqft_living +
##     crimerate + Distance, data = train1.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.28626 -0.18410  0.00527  0.17666  1.53680
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.156e+01  1.757e-02 657.814  < 2e-16 ***
## bathrooms       4.704e-02  4.465e-03  10.534  < 2e-16 ***
## grade           1.679e-01  3.080e-03  54.525  < 2e-16 ***
## floors         -3.591e-02  4.882e-03  -7.356 1.99e-13 ***
## newsqft_living  2.322e-04  4.288e-06  54.145  < 2e-16 ***
## crimerate      -4.646e+00  1.174e-01 -39.557  < 2e-16 ***
## Distance       -1.447e-02  1.958e-04 -73.913  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2908 on 17282 degrees of freedom
## Multiple R-squared:  0.6935, Adjusted R-squared:  0.6934
## F-statistic:  6518 on 6 and 17282 DF,  p-value: < 2.2e-16
```

##Models using step wise regression##

Backward stepwise regression

```
step(model1,direction="backward",trace=T)
```

```
## Start:  AIC=-42697.42
## log(price) ~ bathrooms + grade + bedrooms + floors + newsqft_living +
##     crimerate + Distance
##
##                  Df Sum of Sq    RSS    AIC
## - bedrooms        1      0.06 1461.6 -42699
## <none>                        1461.6 -42697
## - floors          1      4.63 1466.2 -42645
## - bathrooms       1      9.00 1470.6 -42593
## - crimerate       1    132.29 1593.9 -41201
## - newsqft_living  1    228.77 1690.3 -40185
## - grade           1    247.11 1708.7 -39999
## - Distance        1    462.09 1923.7 -37950
##
## Step:  AIC=-42698.76
## log(price) ~ bathrooms + grade + floors + newsqft_living + crimerate +
##     Distance
##
##                  Df Sum of Sq    RSS    AIC
## <none>                        1461.6 -42699
## - floors          1      4.58 1466.2 -42647
## - bathrooms       1      9.39 1471.0 -42590
## - crimerate       1    132.34 1594.0 -41202
## - newsqft_living  1    247.95 1709.6 -39992
## - grade           1    251.44 1713.1 -39956
## - Distance        1    462.05 1923.7 -37952

##
## Call:
## lm(formula = log(price) ~ bathrooms + grade + floors + newsqft_living +
##     crimerate + Distance, data = train1.data)
##
## Coefficients:
```

```
##    (Intercept)       bathrooms           grade          floors
newsqft_living
##     11.5569906       0.0470397       0.1679098      -0.0359110
0.0002322
##      crimerate        Distance
##     -4.6455965      -0.0144686

#Backward suggested to remove bedrooms
#This model is same as model3
```

Forward stepwise regression

```
step(model1,direction="forward",trace=T)

## Start:  AIC=-42697.42
## log(price) ~ bathrooms + grade + bedrooms + floors + newsqft_living +
##     crimerate + Distance

##
## Call:
## lm(formula = log(price) ~ bathrooms + grade + bedrooms + floors +
##     newsqft_living + crimerate + Distance, data = train1.data)
##
## Coefficients:
##    (Intercept)       bathrooms           grade        bedrooms
floors
##     11.5634747       0.0481480       0.1676137      -0.0024669      -
0.0362053
## newsqft_living       crimerate        Distance
##      0.0002333      -4.6448206      -0.0144698

#Forward did not suggest to remove any variables
#This model is same as model2
```

Both stepwise regression

```
step(model1,direction="both",trace=T)

## Start:  AIC=-42697.42
## log(price) ~ bathrooms + grade + bedrooms + floors + newsqft_living +
##     crimerate + Distance
##
##                  Df Sum of Sq    RSS    AIC
## - bedrooms        1      0.06 1461.6 -42699
## <none>                        1461.6 -42697
## - floors          1      4.63 1466.2 -42645
## - bathrooms       1      9.00 1470.6 -42593
## - crimerate       1    132.29 1593.9 -41201
## - newsqft_living  1    228.77 1690.3 -40185
## - grade           1    247.11 1708.7 -39999
## - Distance        1    462.09 1923.7 -37950
##
```

```
## Step:  AIC=-42698.76
## log(price) ~ bathrooms + grade + floors + newsqft_living + crimerate + 
##     Distance
## 
##                  Df Sum of Sq    RSS    AIC
## <none>                        1461.6 -42699
## + bedrooms        1      0.06 1461.6 -42697
## - floors          1      4.58 1466.2 -42647
## - bathrooms       1      9.39 1471.0 -42590
## - crimerate       1    132.34 1594.0 -41202
## - newsqft_living  1    247.95 1709.6 -39992
## - grade           1    251.44 1713.1 -39956
## - Distance        1    462.05 1923.7 -37952

## 
## Call:
## lm(formula = log(price) ~ bathrooms + grade + floors + newsqft_living + 
##     crimerate + Distance, data = train1.data)
## 
## Coefficients:
##    (Intercept)       bathrooms           grade          floors  
newsqft_living  
##     11.5569906       0.0470397       0.1679098      -0.0359110  
0.0002322  
##      crimerate        Distance  
##     -4.6455965      -0.0144686

#Both stepwiseregression suggested to remove bedrooms
#This model is same as model3
```

##Best subset Regression ##

Assigning x and y variable to leaps functions and using stpewise regression with cp as metric for feature selection

```
library(leaps)
leaps(x=train1.data[c(1,2,4,5,6,10,12,14)],y=train1.data[,3],
      names=names(train1.data)[c(1,2,4,5,6,10,12,14
)],method="Cp")

## $which
##   Distance Income bedrooms bathrooms floors grade newsqft_living crimerate
## 1    FALSE  FALSE    FALSE     FALSE  FALSE FALSE           TRUE     FALSE
## 1    FALSE  FALSE    FALSE     FALSE  FALSE  TRUE          FALSE     FALSE
## 1    FALSE  FALSE    FALSE      TRUE  FALSE FALSE          FALSE     FALSE
## 1    FALSE   TRUE    FALSE     FALSE  FALSE FALSE          FALSE     FALSE
## 1    FALSE  FALSE     TRUE     FALSE  FALSE FALSE          FALSE     FALSE
## 1    FALSE  FALSE    FALSE     FALSE   TRUE FALSE          FALSE     FALSE
## 1    FALSE  FALSE    FALSE     FALSE  FALSE FALSE          FALSE      TRUE
## 1     TRUE  FALSE    FALSE     FALSE  FALSE FALSE          FALSE     FALSE
## 2     TRUE  FALSE    FALSE     FALSE  FALSE FALSE           TRUE     FALSE
## 2    FALSE  FALSE    FALSE     FALSE  FALSE  TRUE           TRUE     FALSE
## 2     TRUE  FALSE    FALSE     FALSE  FALSE  TRUE          FALSE     FALSE
## 2    FALSE  FALSE    FALSE     FALSE  FALSE FALSE           TRUE      TRUE
## 2    FALSE   TRUE    FALSE     FALSE  FALSE FALSE           TRUE     FALSE
## 2    FALSE  FALSE    FALSE     FALSE  FALSE  TRUE          FALSE      TRUE
## 2    FALSE   TRUE    FALSE     FALSE  FALSE  TRUE          FALSE     FALSE
## 2    FALSE  FALSE    FALSE      TRUE  FALSE FALSE           TRUE     FALSE
## 2    FALSE  FALSE    FALSE      TRUE  FALSE  TRUE          FALSE     FALSE
## 2    FALSE  FALSE     TRUE     FALSE  FALSE FALSE           TRUE     FALSE
## 3     TRUE  FALSE    FALSE     FALSE  FALSE  TRUE           TRUE     FALSE
## 3     TRUE   TRUE    FALSE     FALSE  FALSE FALSE           TRUE     FALSE
## 3     TRUE  FALSE    FALSE     FALSE  FALSE FALSE           TRUE      TRUE
## 3     TRUE  FALSE    FALSE      TRUE  FALSE FALSE           TRUE     FALSE
## 3     TRUE  FALSE     TRUE     FALSE  FALSE FALSE           TRUE     FALSE
## 3     TRUE  FALSE    FALSE     FALSE   TRUE FALSE           TRUE     FALSE
## 3    FALSE  FALSE    FALSE     FALSE  FALSE  TRUE           TRUE      TRUE
## 3     TRUE   TRUE    FALSE     FALSE  FALSE  TRUE          FALSE     FALSE
## 3     TRUE  FALSE    FALSE      TRUE  FALSE  TRUE          FALSE     FALSE
## 3    FALSE   TRUE    FALSE     FALSE  FALSE  TRUE           TRUE     FALSE
## 4     TRUE   TRUE    FALSE     FALSE  FALSE  TRUE           TRUE     FALSE
## 4     TRUE  FALSE    FALSE     FALSE  FALSE  TRUE           TRUE      TRUE
## 4     TRUE  FALSE    FALSE     FALSE   TRUE  TRUE           TRUE     FALSE
## 4     TRUE  FALSE     TRUE     FALSE  FALSE  TRUE           TRUE     FALSE
## 4     TRUE  FALSE    FALSE      TRUE  FALSE  TRUE           TRUE     FALSE
## 4     TRUE   TRUE    FALSE     FALSE  FALSE FALSE           TRUE      TRUE
## 4     TRUE   TRUE    FALSE      TRUE  FALSE FALSE           TRUE     FALSE
## 4     TRUE   TRUE     TRUE     FALSE  FALSE FALSE           TRUE     FALSE
## 4     TRUE   TRUE    FALSE     FALSE   TRUE FALSE           TRUE     FALSE
## 4     TRUE  FALSE    FALSE      TRUE  FALSE FALSE           TRUE      TRUE
## 5     TRUE   TRUE    FALSE     FALSE  FALSE  TRUE           TRUE      TRUE
## 5     TRUE   TRUE    FALSE     FALSE   TRUE  TRUE           TRUE     FALSE
## 5     TRUE   TRUE     TRUE     FALSE  FALSE  TRUE           TRUE     FALSE
## 5     TRUE   TRUE    FALSE      TRUE  FALSE  TRUE           TRUE     FALSE
## 5     TRUE  FALSE    FALSE     FALSE   TRUE  TRUE           TRUE      TRUE
## 5     TRUE  FALSE     TRUE     FALSE  FALSE  TRUE           TRUE      TRUE
## 5     TRUE  FALSE    FALSE      TRUE  FALSE  TRUE           TRUE      TRUE
## 5     TRUE  FALSE     TRUE     FALSE   TRUE  TRUE           TRUE     FALSE
## 5     TRUE  FALSE    FALSE      TRUE   TRUE  TRUE           TRUE     FALSE
## 5     TRUE  FALSE     TRUE      TRUE  FALSE  TRUE           TRUE     FALSE
## 6     TRUE   TRUE    FALSE     FALSE   TRUE  TRUE           TRUE      TRUE
## 6     TRUE   TRUE     TRUE     FALSE  FALSE  TRUE           TRUE      TRUE
## 6     TRUE   TRUE    FALSE      TRUE  FALSE  TRUE           TRUE      TRUE
## 6     TRUE   TRUE    FALSE      TRUE   TRUE  TRUE           TRUE     FALSE
## 6     TRUE   TRUE     TRUE     FALSE   TRUE  TRUE           TRUE     FALSE
## 6     TRUE   TRUE     TRUE      TRUE  FALSE  TRUE           TRUE     FALSE
## 6     TRUE  FALSE     TRUE     FALSE   TRUE  TRUE           TRUE      TRUE
## 6     TRUE  FALSE    FALSE      TRUE   TRUE  TRUE           TRUE      TRUE
## 6     TRUE  FALSE     TRUE      TRUE  FALSE  TRUE           TRUE      TRUE
## 6     TRUE  FALSE     TRUE      TRUE   TRUE  TRUE           TRUE     FALSE
## 7     TRUE   TRUE    FALSE      TRUE   TRUE  TRUE           TRUE      TRUE
## 7     TRUE   TRUE     TRUE     FALSE   TRUE  TRUE           TRUE      TRUE
## 7     TRUE   TRUE     TRUE      TRUE  FALSE  TRUE           TRUE      TRUE
## 7     TRUE   TRUE     TRUE      TRUE   TRUE  TRUE           TRUE     FALSE
## 7     TRUE  FALSE     TRUE      TRUE   TRUE  TRUE           TRUE      TRUE
## 7     TRUE   TRUE     TRUE      TRUE   TRUE FALSE           TRUE      TRUE
## 7     TRUE   TRUE     TRUE      TRUE   TRUE  TRUE          FALSE      TRUE
## 7    FALSE   TRUE     TRUE      TRUE   TRUE  TRUE           TRUE      TRUE
## 8     TRUE   TRUE     TRUE      TRUE   TRUE  TRUE           TRUE      TRUE
##
## $label
## [1] "(Intercept)"    "Distance"       "Income"         "bedrooms"
## [5] "bathrooms"      "floors"         "grade"          "newsqft_living"
## [9] "crimerate"
##
## $size
##  [1] 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5
5 5 5
## [39] 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 9
##
## $Cp
##  [1]  8605.04056  8968.29234 17179.74402 22678.09468 25360.62076
26805.43485
##  [7] 26949.77001 27053.28198  3924.99582  5612.03184  5816.55193
6590.35454
## [13]  7527.48445  7792.49076  8057.80364  8338.98091  8465.70389
8467.97387
## [19]  1512.04679  2558.31541  2946.74703  3589.06990  3813.97481
3886.00473
## [25]  4195.62869  4552.59546  5007.36551  5096.33194   741.41617
869.94437
## [31]  1390.96698  1446.00780  1505.36881  1894.05866  2281.53610
2493.99321
## [37]  2544.58141  2675.14691   268.41111   621.58744   699.70823
731.50130
## [43]   732.19805   805.26694   866.17918  1329.37755  1345.04922
1417.63497
## [49]   134.10721   225.28794   261.85344   568.26309   583.40885
670.90595
## [55]   672.23012   690.77125   783.06128  1237.96989    85.79403
94.75561
## [61]   201.96534   491.30782   587.98102  1479.59469  2933.00763
3635.21796
## [67]     9.00000

#This suggested variables sames as model2
```

Assigning x and y variable to leaps functions and using stpewise regression with adjusted R squared squared as metric for feature selection

```
leaps(x=train1.data[c(1,4,5,6,10,12,14)],y=train1.data[,3],
      names=names(train1.data)[c(1,4,5,6,10,12,14
      )],method="adjr2")
```

```
## $which
##   Distance bedrooms bathrooms floors grade newsqft_living crimerate
## 1    FALSE    FALSE     FALSE  FALSE FALSE           TRUE     FALSE
## 1    FALSE    FALSE     FALSE  FALSE  TRUE          FALSE     FALSE
## 1    FALSE    FALSE      TRUE  FALSE FALSE          FALSE     FALSE
## 1    FALSE     TRUE     FALSE  FALSE FALSE          FALSE     FALSE
## 1    FALSE    FALSE     FALSE   TRUE FALSE          FALSE     FALSE
## 1    FALSE    FALSE     FALSE  FALSE FALSE          FALSE      TRUE
## 1     TRUE    FALSE     FALSE  FALSE FALSE          FALSE     FALSE
## 2     TRUE    FALSE     FALSE  FALSE FALSE           TRUE     FALSE
## 2    FALSE    FALSE     FALSE  FALSE  TRUE           TRUE     FALSE
## 2     TRUE    FALSE     FALSE  FALSE  TRUE          FALSE     FALSE
## 2    FALSE    FALSE     FALSE  FALSE FALSE           TRUE      TRUE
## 2    FALSE    FALSE     FALSE  FALSE  TRUE          FALSE      TRUE
## 2    FALSE    FALSE      TRUE  FALSE FALSE           TRUE     FALSE
## 2    FALSE    FALSE      TRUE  FALSE  TRUE          FALSE     FALSE
## 2    FALSE     TRUE     FALSE  FALSE FALSE           TRUE     FALSE
## 2    FALSE    FALSE     FALSE   TRUE FALSE           TRUE     FALSE
## 2    FALSE     TRUE     FALSE  FALSE  TRUE          FALSE     FALSE
## 3     TRUE    FALSE     FALSE  FALSE  TRUE           TRUE     FALSE
## 3     TRUE    FALSE     FALSE  FALSE FALSE           TRUE      TRUE
## 3     TRUE    FALSE      TRUE  FALSE FALSE           TRUE     FALSE
## 3     TRUE     TRUE     FALSE  FALSE FALSE           TRUE     FALSE
## 3     TRUE    FALSE     FALSE   TRUE FALSE           TRUE     FALSE
## 3    FALSE    FALSE     FALSE  FALSE  TRUE           TRUE      TRUE
## 3     TRUE    FALSE      TRUE  FALSE  TRUE          FALSE     FALSE
## 3     TRUE    FALSE     FALSE  FALSE  TRUE          FALSE      TRUE
## 3     TRUE     TRUE     FALSE  FALSE  TRUE          FALSE     FALSE
## 3    FALSE    FALSE     FALSE   TRUE  TRUE           TRUE     FALSE
## 4     TRUE    FALSE     FALSE  FALSE  TRUE           TRUE      TRUE
## 4     TRUE    FALSE     FALSE   TRUE  TRUE           TRUE     FALSE
## 4     TRUE     TRUE     FALSE  FALSE  TRUE           TRUE     FALSE
## 4     TRUE    FALSE      TRUE  FALSE  TRUE           TRUE     FALSE
## 4     TRUE    FALSE      TRUE  FALSE FALSE           TRUE      TRUE
## 4     TRUE     TRUE     FALSE  FALSE FALSE           TRUE      TRUE
## 4     TRUE    FALSE     FALSE   TRUE FALSE           TRUE      TRUE
## 4     TRUE     TRUE      TRUE  FALSE FALSE           TRUE     FALSE
## 4     TRUE    FALSE      TRUE   TRUE FALSE           TRUE     FALSE
## 4     TRUE     TRUE     FALSE   TRUE FALSE           TRUE     FALSE
## 5     TRUE    FALSE     FALSE   TRUE  TRUE           TRUE      TRUE
## 5     TRUE     TRUE     FALSE  FALSE  TRUE           TRUE      TRUE
## 5     TRUE    FALSE      TRUE  FALSE  TRUE           TRUE      TRUE
## 5     TRUE     TRUE     FALSE   TRUE  TRUE           TRUE     FALSE
## 5     TRUE    FALSE      TRUE   TRUE  TRUE           TRUE     FALSE
## 5     TRUE     TRUE      TRUE  FALSE  TRUE           TRUE     FALSE
## 5     TRUE     TRUE      TRUE  FALSE FALSE           TRUE      TRUE
## 5     TRUE    FALSE      TRUE   TRUE FALSE           TRUE      TRUE
## 5     TRUE     TRUE     FALSE   TRUE FALSE           TRUE      TRUE
## 5     TRUE     TRUE      TRUE   TRUE FALSE           TRUE     FALSE
## 6     TRUE     TRUE     FALSE   TRUE  TRUE           TRUE      TRUE
## 6     TRUE    FALSE      TRUE   TRUE  TRUE           TRUE      TRUE
## 6     TRUE     TRUE      TRUE  FALSE  TRUE           TRUE      TRUE
## 6     TRUE     TRUE      TRUE   TRUE  TRUE           TRUE     FALSE
## 6     TRUE     TRUE      TRUE   TRUE FALSE           TRUE      TRUE
## 6     TRUE     TRUE      TRUE   TRUE  TRUE          FALSE      TRUE
## 6    FALSE     TRUE      TRUE   TRUE  TRUE           TRUE      TRUE
## 7     TRUE     TRUE      TRUE   TRUE  TRUE           TRUE      TRUE
##
## $label
## [1] "(Intercept)"    "Distance"       "bedrooms"       "bathrooms"
## [5] "floors"         "grade"          "newsqft_living" "crimerate"
##
## $size
##  [1] 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5
5 5 6
## [39] 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 8
##
## $adjr2
##  [1] 0.45201885 0.44433037 0.27052915 0.09737507 0.06679455 0.06373959
##  [7] 0.06154869 0.55109170 0.51538231 0.51105325 0.49467422 0.46922868
## [13] 0.45766115 0.45497882 0.45493077 0.45354234 0.44973775 0.60218576
## [19] 0.57181579 0.55821900 0.55345817 0.55193343 0.54537926 0.52819629
## [25] 0.52251729 0.51925082 0.51786965 0.61579799 0.60476826 0.60360308
## [31] 0.60234645 0.57758296 0.57405438 0.57217248 0.56347798 0.55821434
## [37] 0.55431967 0.61873428 0.61718736 0.61589782 0.60609162 0.60575984
## [43] 0.60422316 0.58234638 0.57769070 0.57442513 0.56360498 0.62002419
## [49] 0.61963164 0.61767769 0.60804644 0.58267764 0.55095368 0.55055118
## [55] 0.62182836

#This is suggesting distance,grade and square feet living as variables
```

Building a new model with variables suggested using adjusted R squared metric

```
model3 <- lm(formula=log(price)~grade+
           newsqft_living+Distance,data=train1.data)
summary(model3)

##
## Call:
## lm(formula = log(price) ~ grade + newsqft_living + Distance,
##     data = train1.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.44977 -0.20331 -0.00125  0.19424  1.58393
##
## Coefficients:
```

```
##                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)     1.141e+01  1.795e-02  635.43   <2e-16 ***
## grade           1.844e-01  2.948e-03   62.55   <2e-16 ***
## newsqft_living  2.465e-04  4.082e-06   60.38   <2e-16 ***
## Distance       -1.588e-02  2.006e-04  -79.19   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.3047 on 17285 degrees of freedom
## Multiple R-squared:  0.6635, Adjusted R-squared:  0.6634 
## F-statistic: 1.136e+04 on 3 and 17285 DF,  p-value: < 2.2e-16
```

Assigning x and y variable to leaps functions and using stpewise regression with R squared as metric for feature selection

```
leaps(x=train1.data[c(1,4,5,6,10,12,14)],y=train1.data[,3],
      names=names(train1.data)[c(1,4,5,6,10,12,14
      )],method="r2")
```

```
## $which
##   Distance bedrooms bathrooms floors grade newsqft_living crimerate
## 1    FALSE    FALSE     FALSE  FALSE FALSE           TRUE     FALSE
## 1    FALSE    FALSE     FALSE  FALSE  TRUE          FALSE     FALSE
## 1    FALSE    FALSE      TRUE  FALSE FALSE          FALSE     FALSE
## 1    FALSE     TRUE     FALSE  FALSE FALSE          FALSE     FALSE
## 1    FALSE    FALSE     FALSE   TRUE FALSE          FALSE     FALSE
## 1    FALSE    FALSE     FALSE  FALSE FALSE          FALSE      TRUE
## 1     TRUE    FALSE     FALSE  FALSE FALSE          FALSE     FALSE
## 2     TRUE    FALSE     FALSE  FALSE FALSE           TRUE     FALSE
## 2    FALSE    FALSE     FALSE  FALSE  TRUE           TRUE     FALSE
## 2     TRUE    FALSE     FALSE  FALSE  TRUE          FALSE     FALSE
## 2    FALSE    FALSE     FALSE  FALSE FALSE           TRUE      TRUE
## 2    FALSE    FALSE     FALSE  FALSE  TRUE          FALSE      TRUE
## 2    FALSE    FALSE      TRUE  FALSE FALSE           TRUE     FALSE
## 2    FALSE    FALSE      TRUE  FALSE  TRUE          FALSE     FALSE
## 2    FALSE     TRUE     FALSE  FALSE FALSE           TRUE     FALSE
## 2    FALSE    FALSE     FALSE   TRUE FALSE           TRUE     FALSE
## 2    FALSE     TRUE     FALSE  FALSE  TRUE          FALSE     FALSE
## 3     TRUE    FALSE     FALSE  FALSE  TRUE           TRUE     FALSE
## 3     TRUE    FALSE     FALSE  FALSE FALSE           TRUE      TRUE
## 3     TRUE    FALSE      TRUE  FALSE FALSE           TRUE     FALSE
## 3     TRUE     TRUE     FALSE  FALSE FALSE           TRUE     FALSE
## 3     TRUE    FALSE     FALSE   TRUE FALSE           TRUE     FALSE
## 3    FALSE    FALSE     FALSE  FALSE  TRUE           TRUE      TRUE
## 3     TRUE    FALSE      TRUE  FALSE  TRUE          FALSE     FALSE
## 3     TRUE    FALSE     FALSE  FALSE  TRUE          FALSE      TRUE
## 3     TRUE     TRUE     FALSE  FALSE  TRUE          FALSE     FALSE
## 3    FALSE    FALSE     FALSE   TRUE  TRUE           TRUE     FALSE
## 4     TRUE    FALSE     FALSE  FALSE  TRUE           TRUE      TRUE
## 4     TRUE    FALSE     FALSE   TRUE  TRUE           TRUE     FALSE
## 4     TRUE     TRUE      FALSE  FALSE  TRUE           TRUE     FALSE
## 4     TRUE    FALSE       TRUE  FALSE  TRUE           TRUE     FALSE
## 4     TRUE    FALSE       TRUE  FALSE FALSE           TRUE      TRUE
## 4     TRUE     TRUE      FALSE  FALSE FALSE           TRUE      TRUE
## 4     TRUE    FALSE      FALSE   TRUE FALSE           TRUE      TRUE
## 4     TRUE     TRUE       TRUE  FALSE FALSE           TRUE     FALSE
## 4     TRUE    FALSE       TRUE   TRUE FALSE           TRUE     FALSE
## 4     TRUE     TRUE      FALSE   TRUE FALSE           TRUE     FALSE
## 5     TRUE    FALSE      FALSE   TRUE  TRUE           TRUE      TRUE
## 5     TRUE     TRUE      FALSE  FALSE  TRUE           TRUE      TRUE
## 5     TRUE    FALSE       TRUE  FALSE  TRUE           TRUE      TRUE
## 5     TRUE     TRUE      FALSE   TRUE  TRUE           TRUE     FALSE
## 5     TRUE    FALSE       TRUE   TRUE  TRUE           TRUE     FALSE
## 5     TRUE     TRUE       TRUE  FALSE  TRUE           TRUE     FALSE
## 5     TRUE     TRUE       TRUE  FALSE FALSE           TRUE      TRUE
## 5     TRUE    FALSE       TRUE   TRUE FALSE           TRUE      TRUE
## 5     TRUE     TRUE      FALSE   TRUE FALSE           TRUE      TRUE
## 5     TRUE     TRUE       TRUE   TRUE FALSE           TRUE     FALSE
## 6     TRUE     TRUE      FALSE   TRUE  TRUE           TRUE      TRUE
## 6     TRUE    FALSE       TRUE   TRUE  TRUE           TRUE      TRUE
## 6     TRUE     TRUE       TRUE  FALSE  TRUE           TRUE      TRUE
## 6     TRUE     TRUE       TRUE   TRUE  TRUE           TRUE     FALSE
## 6     TRUE     TRUE       TRUE   TRUE FALSE           TRUE      TRUE
## 6     TRUE     TRUE       TRUE   TRUE  TRUE          FALSE      TRUE
## 6    FALSE     TRUE       TRUE   TRUE  TRUE           TRUE      TRUE
## 7     TRUE     TRUE       TRUE   TRUE  TRUE           TRUE      TRUE
## 
## $label
## [1] "(Intercept)"    "Distance"       "bedrooms"       "bathrooms"     
## [5] "floors"         "grade"          "newsqft_living" "crimerate"     
## 
## $size
##  [1] 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5
5 5 6
## [39] 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 8
## 
## $r2
##  [1] 0.45205055 0.44436251 0.27057134 0.09742728 0.06684853 0.06379375
##  [7] 0.06160297 0.55114363 0.51543838 0.51110981 0.49473268 0.46929008
## [13] 0.45772390 0.45504187 0.45499383 0.45360556 0.44980141 0.60225479
## [19] 0.57189009 0.55829566 0.55353566 0.55201119 0.54545815 0.52827816
## [25] 0.52260015 0.51933424 0.51795331 0.61588689 0.60485971 0.60369480
## [31] 0.60243845 0.57768070 0.57415293 0.57227147 0.56357898 0.55831655
## [37] 0.55442279 0.61884455 0.61729808 0.61600890 0.60620555 0.60587387
## [43] 0.60433763 0.58246718 0.57781284 0.57454821 0.56373119 0.62015607
## [49] 0.61976365 0.61781038 0.60818247 0.58282248 0.55110953 0.55070717
## [55] 0.62198148
```

```
#This suggest variables similar to m4
```

##Multicollinearity##

```
library(car)
#Testing multicollinearity for model 1
vif(model1)

##      bathrooms          grade       bedrooms         floors newsqft_living 
##       2.615030       2.697341       1.530456       1.430074       2.971639 
##      crimerate       Distance 
##       1.068324       1.063413

#Result - No Multi Colliearity issues in model1

#Testing multicollinearity for model 2
vif(model2)

##      bathrooms          grade         floors newsqft_living      crimerate 
##       2.392949       2.660275       1.422287       2.716635       1.068254 
##       Distance 
##       1.063357

#Result - No Multi Colliearity issues in model2

#Testing multicollinearity for model 3
vif(model3)

##          grade newsqft_living       Distance 
##       2.221183       2.241694       1.016669

#Result - No Multi Colliearity issues in model3
```

##Predicted Values and rmse##

Assigning price as y

```
y=test1.data[,3]
```

predicting the price for test data using model1

```
m1y1<-exp(predict.glm(model1,test1.data))
```

Calculating rmse for predicted and observed values using model1

```
py=test1.data[,3]
rmsem1 = sqrt((py-m1y1)%*%(py-m1y1))/nrow(test1.data)
rmsem1

##          [,1]
## [1,] 4532.372
```

predicting the price for test data using model1

```
m2y2<-exp(predict.glm(model2,test1.data))
```

Calculating rmse for predicted and observed values using model1

```
py=test1.data[,3]
rmsem2 = sqrt((py-m2y2)%*%(py-m2y2))/nrow(test1.data)
rmsem2

##          [,1]
## [1,] 4507.944
```

predicting the price for test data using model1

```
m3y3<-exp(predict.glm(model3,test1.data))
```

Calculating rmse for predicted and observed values using model1

```
py=test1.data[,3]
rmsem3 = sqrt((py-m3y3)%*%(py-m3y3))/nrow(test1.data)
rmsem3

##          [,1]
## [1,] 4673.479
```

Model 1

```
rmsem1

##          [,1]
## [1,] 4532.372
```

Model 2

```
rmsem2

##          [,1]
## [1,] 4507.944
```

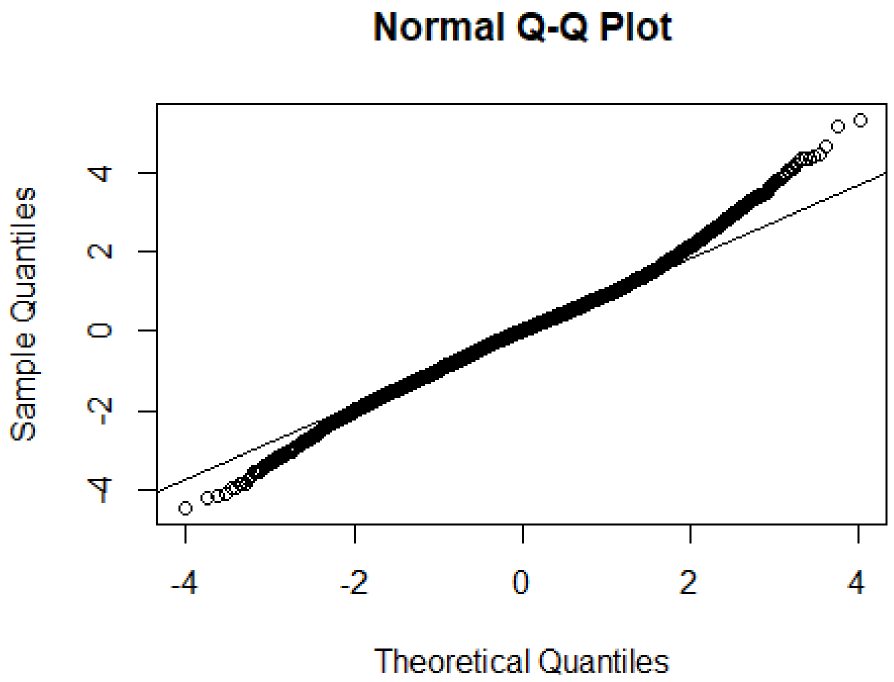
Model 3

```
rmsem3

##          [,1]
## [1,] 4673.479
```

##Residuals Assumptions##

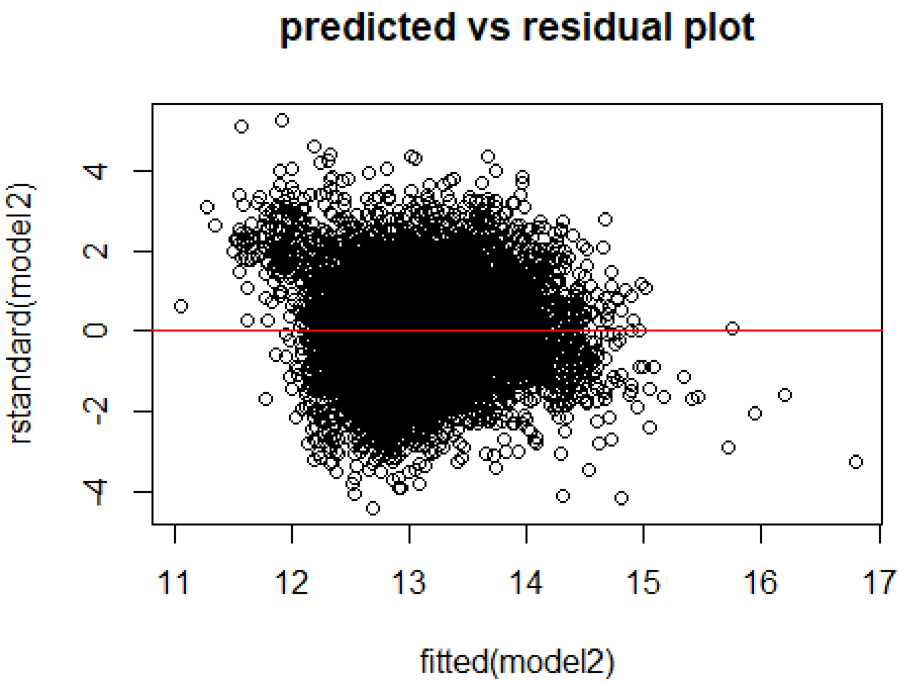
model1 Residual plot normality

```
qqnorm(rstandard(model1))
qqline(rstandard(model1,col="red"))
```



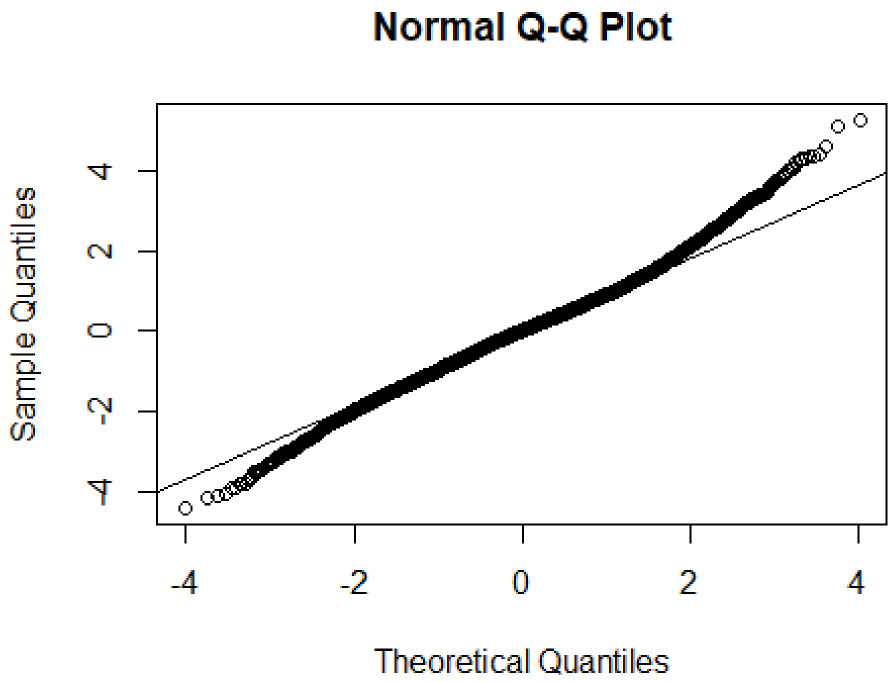
plot between residual and predicted

```
plot(fitted(model2),rstandard(model2),main="predicted vs residual plot")
abline(a=0, b=0, col='red')
```



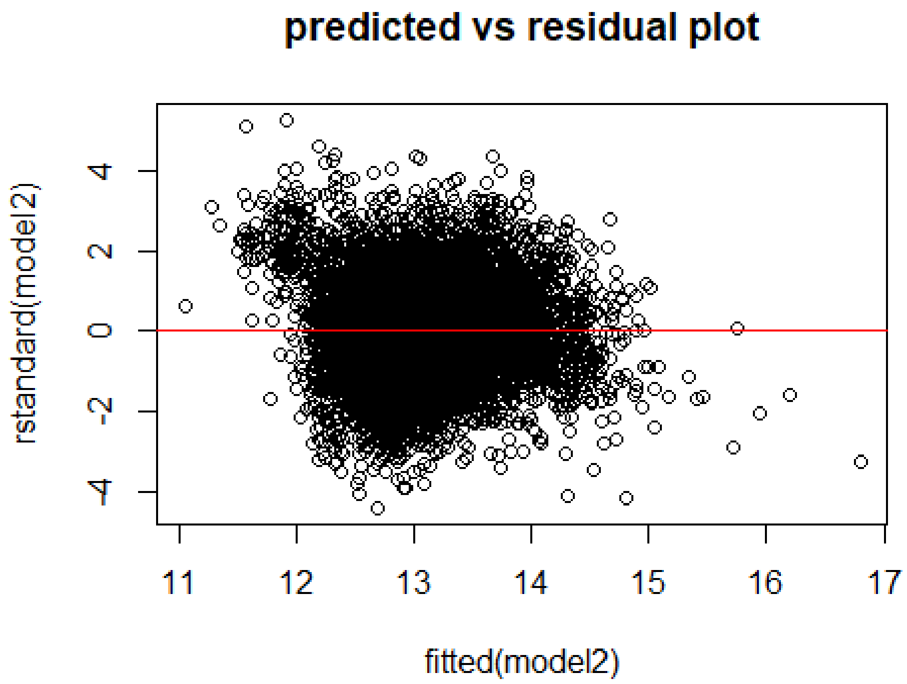
model2 Residual plot normality

```
qqnorm(rstandard(model2))
qqline(rstandard(model2,col="red"))
```



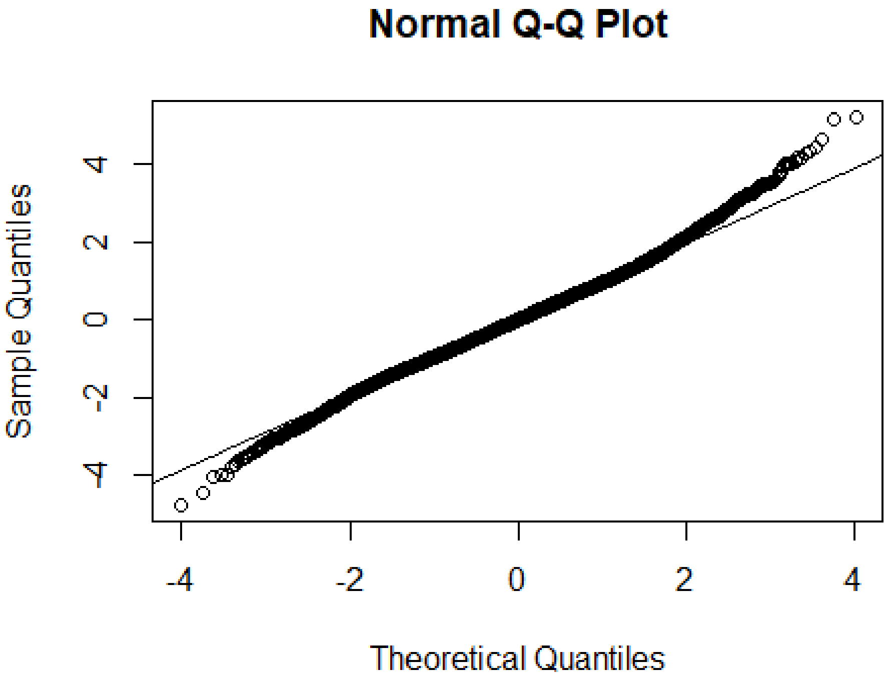
plot between residual and predicted

```
plot(fitted(model2),rstandard(model2),main="predicted vs residual plot")
abline(a=0, b=0, col='red')
```



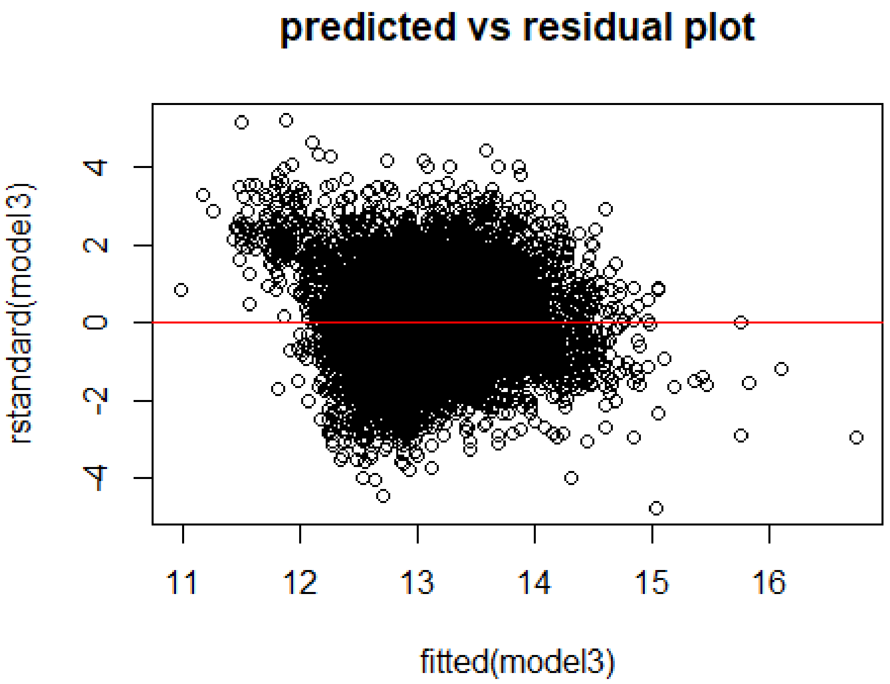
model3 Residual plot normality

```
qqnorm(rstandard(model3))
qqline(rstandard(model3,col="red"))
```



plot between residual and predicted

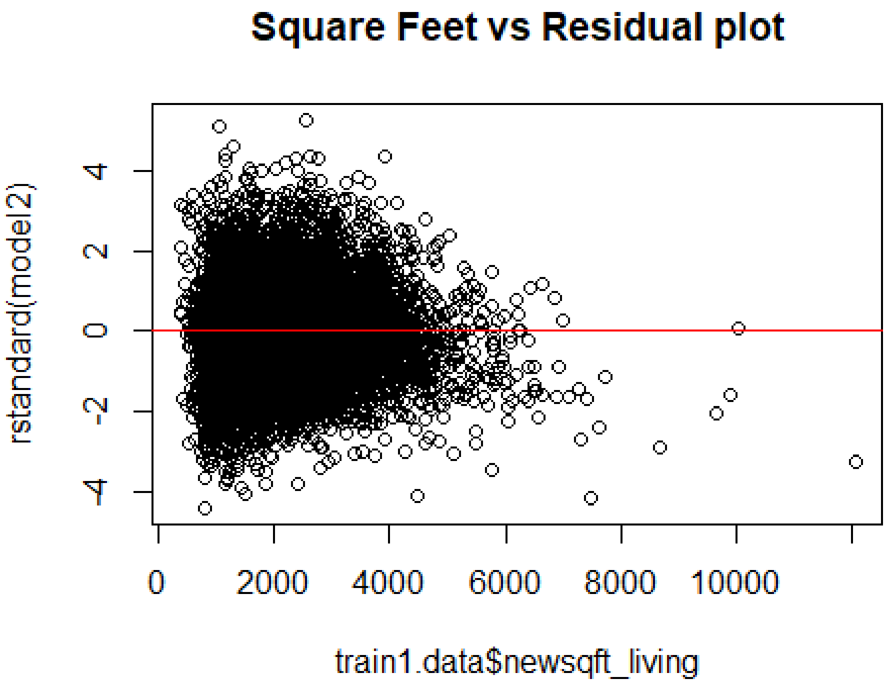
```
plot(fitted(model3),rstandard(model3),main="predicted vs residual plot")
abline(a=0, b=0, col='red')
```



##Residual vs independent variables for model3##

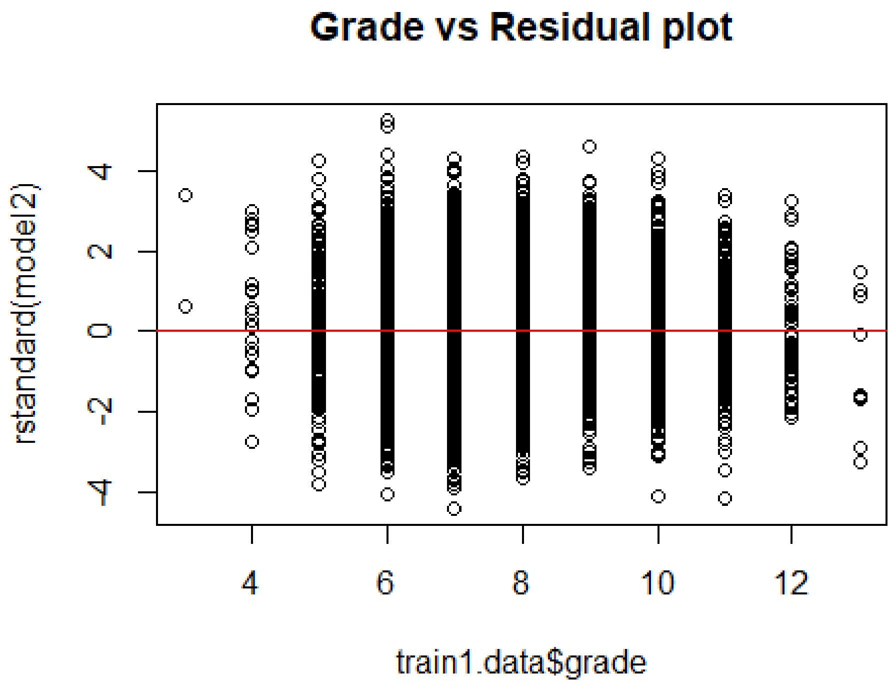
plot for residual and sqft living variable

```
plot(train1.data$newsqft_living,rstandard(model2),main="Square Feet vs
Residual plot")
abline(a=0, b=0, col='red')
```



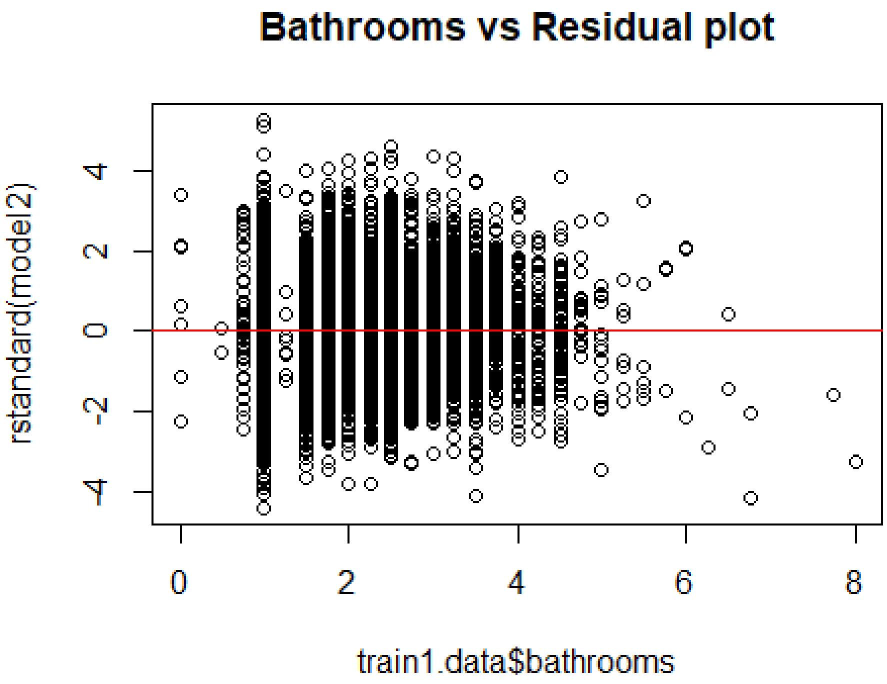
plot for residual and grade variable

```
plot(train1.data$grade,rstandard(model2),main="Grade vs Residual plot")
abline(a=0, b=0, col='red')
```



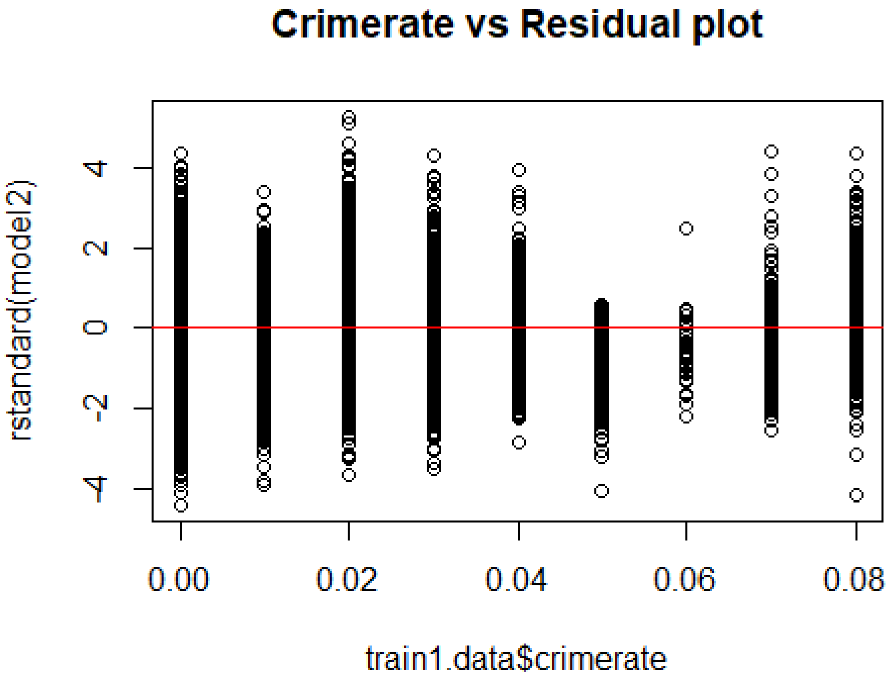
plot for residual and bedrooms variable

```
plot(train1.data$bathrooms,rstandard(model2),main="Bathrooms vs Residual
plot")
abline(a=0, b=0, col='red')
```



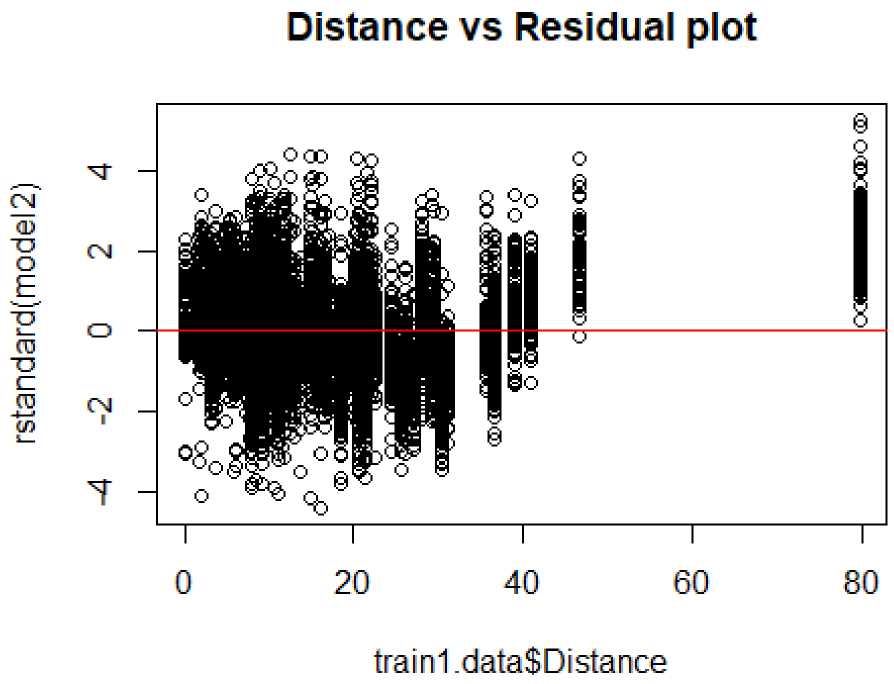
plot for residual and crime variable

```
plot(train1.data$crimerate,rstandard(model2),main="Crimerate vs Residual
plot")
abline(a=0, b=0, col='red')
```



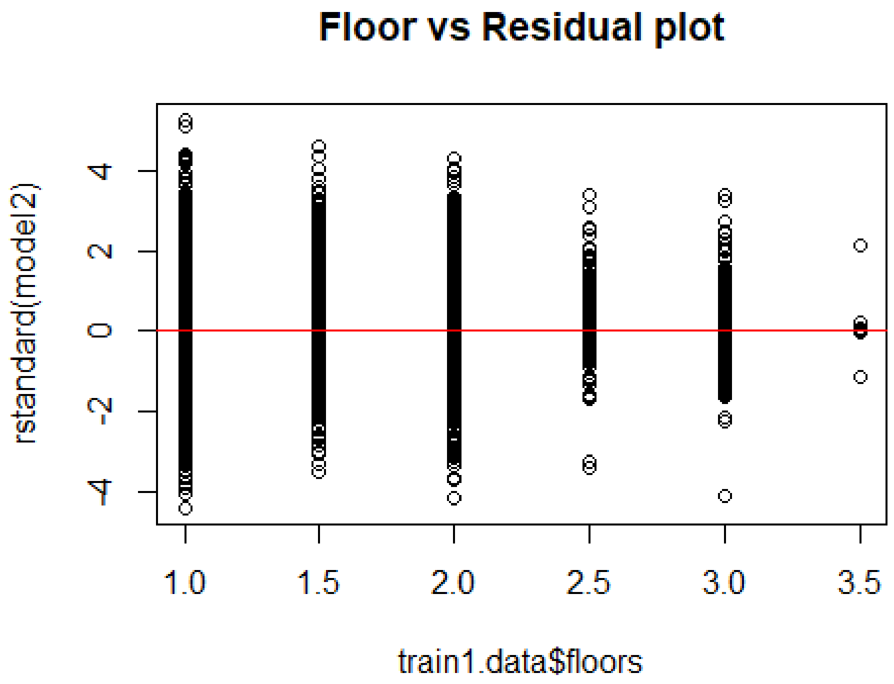
plot for residual and distance variable

```
plot(train1.data$Distance,rstandard(model2),main="Distance vs Residual plot")
abline(a=0, b=0, col='red')
```



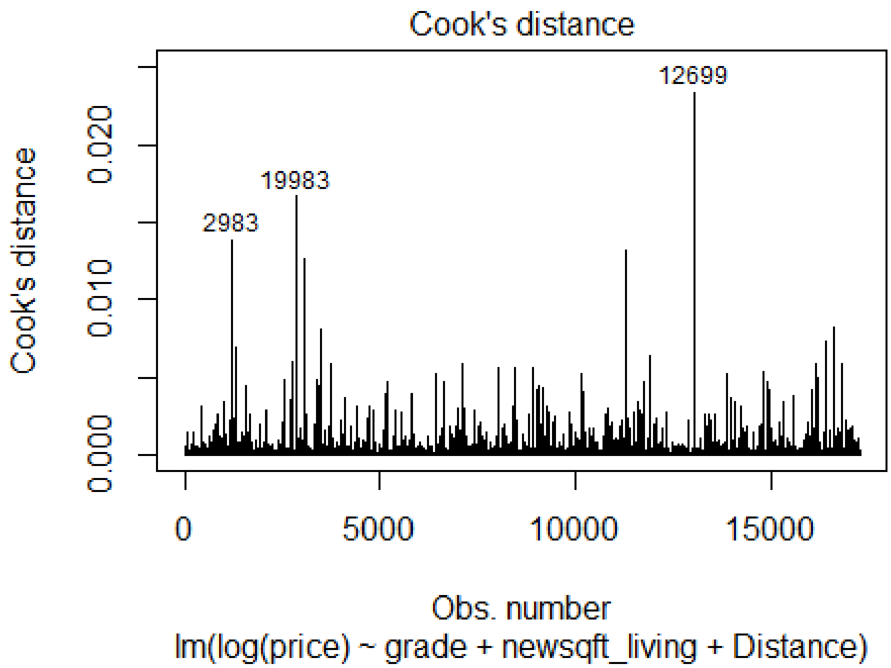
plot for residual and floors variable

```
plot(train1.data$floors,rstandard(model2),main="Floor vs Residual plot")
abline(a=0, b=0, col='red')
```



##Influential Points##

```
cutoff <- 4/((nrow(train1.data)-length(model3$coefficients)-2))
plot(model3, which=4, cook.levels=cutoff)
```



Removing influential points

```
fulldata2 <- fulldata[- 19983,]
fulldata2 <- fulldata[- 12699,]
fulldata2 <- fulldata[- 8845,]
data2<-fulldata2[c(2,3,13,14,15,18,19,20,21,22,31,33,34,35)]
```

Building the model again using model3 after removing influentail points

```
selectdata2 = sample(1:nrow(data2),0.80*nrow(data2));
train2.data = data2[selectdata2,];
test2.data = data2[-selectdata2,];

model5 <- lm(formula=log(price)~bathrooms+grade+floors+
           newsqft_living+crimerate+Distance,data=train2.data)
summary(model5)

##
## Call:
## lm(formula = log(price) ~ bathrooms + grade + floors + newsqft_living +
##     crimerate + Distance, data = train2.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.28337 -0.18335  0.00369  0.17535  1.52663
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.154e+01  1.755e-02 657.634  < 2e-16 ***
## bathrooms       4.750e-02  4.484e-03  10.594  < 2e-16 ***
## grade           1.687e-01  3.061e-03  55.096  < 2e-16 ***
## floors         -3.376e-02  4.878e-03  -6.921 4.64e-12 ***
## newsqft_living  2.325e-04  4.266e-06  54.513  < 2e-16 ***
## crimerate      -4.557e+00  1.172e-01 -38.889  < 2e-16 ***
## Distance       -1.429e-02  1.958e-04 -72.969  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2907 on 17281 degrees of freedom
## Multiple R-squared:  0.6951, Adjusted R-squared:  0.695
## F-statistic:  6567 on 6 and 17281 DF,  p-value: < 2.2e-16
```

predicting the price for test data using model5

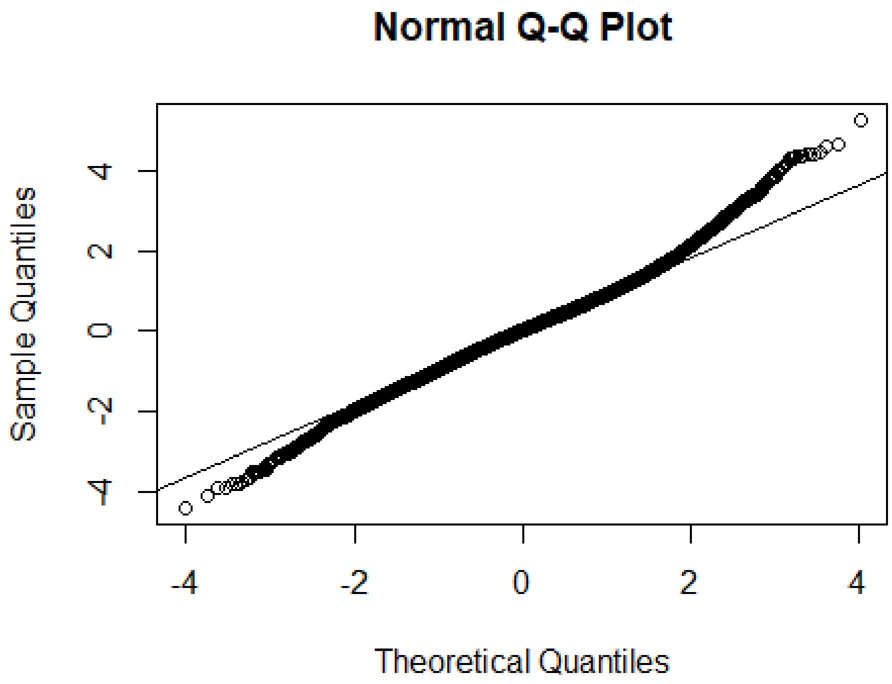
```
y5<-exp(predict.glm(model5,test2.data))
```

Calculating rmse for predicted and observed values using model5

```
y=test2.data[,3]
rmse5= sqrt((y-y5)%*%(y-y5))/nrow(test2.data)
rmse5

##          [,1]
## [1,] 3162.197
```

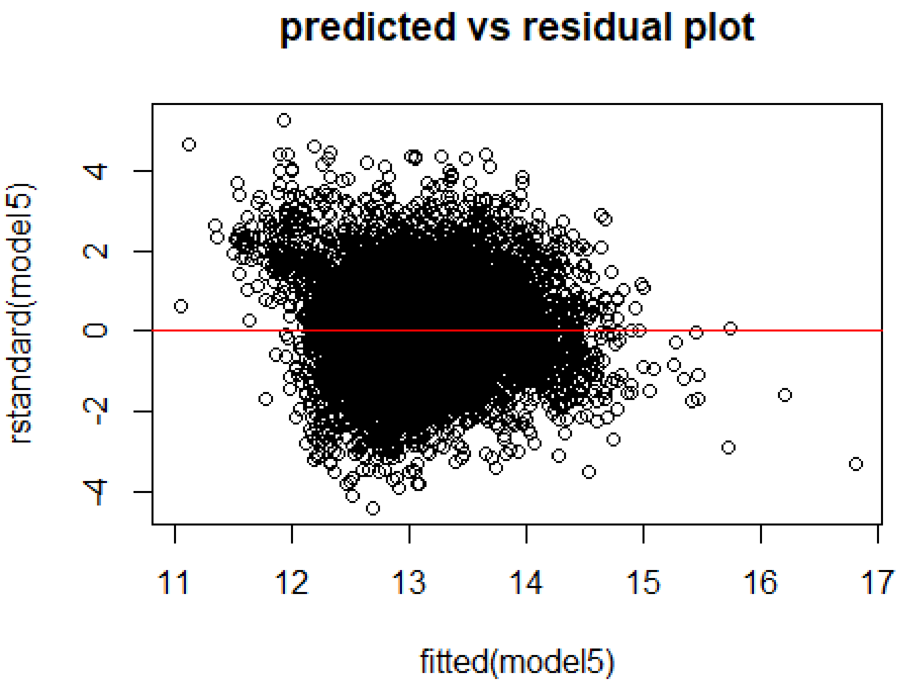
Residual Plots model5 Residual plot normality

```
qqnorm(rstandard(model5))
qqline(rstandard(model5,col="red"))
```



plot between residual and predicted

```
plot(fitted(model5),rstandard(model5),main="predicted vs residual plot")
abline(a=0, b=0, col='red')
```



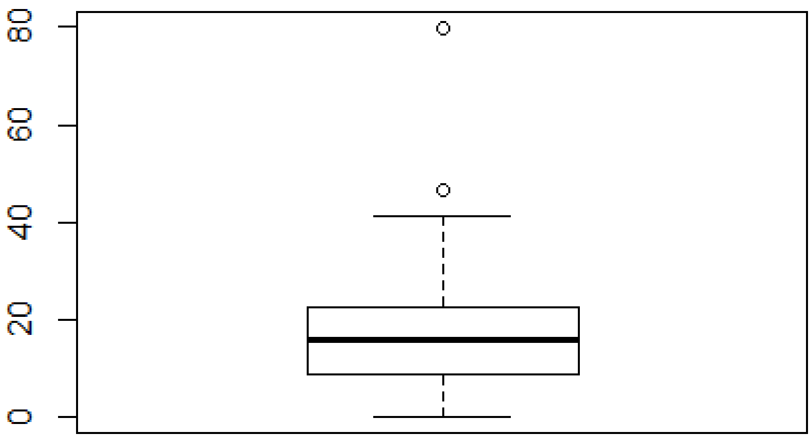
##########################Hypothesis Testing the effect of distance on price##### ##Two Sample Z test for Distance##

Calculating price persqft

```
fulldata$pricepersqft <- fulldata$price/fulldata$newsqft_living
names(fulldata)

##  [1] "zipcode"        "Distance"       "Income"         "time"
##  [5] "population"     "crime.count"    "burglary"       "theft"
##  [9] "b.t"            "id"             "date"           "Date.only"
## [13] "price"          "bedrooms"       "bathrooms"      "sqft_living"
## [17] "sqft_lot"       "floors"         "waterfront"     "view"
## [21] "condition"      "grade"          "sqft_above"     "sqft_basement"
## [25] "yr_built"       "yr_renovated"   "lat"            "long"
## [29] "sqft_living15"  "sqft_lot15"     "Age"            "year"
## [33] "newsqft_living" "newsqft_lot"    "crimerate"      "pricepersqft"
```

Analysing distance

```
summary(fulldata$Distance)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    8.50   15.85   17.31   22.49   79.83

boxplot(fulldata$Distance)
```



```
names(fulldata)
```

```
##  [1] "zipcode"       "Distance"      "Income"        "time"
##  [5] "population"    "crime.count"   "burglary"      "theft"
##  [9] "b.t"           "id"            "date"          "Date.only"
## [13] "price"         "bedrooms"      "bathrooms"     "sqft_living"
## [17] "sqft_lot"      "floors"        "waterfront"    "view"
## [21] "condition"     "grade"         "sqft_above"    "sqft_basement"
## [25] "yr_built"      "yr_renovated"  "lat"           "long"
## [29] "sqft_living15" "sqft_lot15"    "Age"           "year"
## [33] "newsqft_living" "newsqft_lot"  "crimerate"     "pricepersqft"
```

Storing distance and price in a new dataset

```
hypdistance<- fulldata[c(2,36)]
```

First sample with zipcodes less then median distance from Amazon head quartes

```
x<- subset(hypdistance,Distance <= 15.85)
```

Second sample with zipcodes gretera then median distance from Amazon head quartes

```
y<- subset(hypdistance,Distance > 15.85)
```

Assiging pricepersqft of two samples to x and y varaibles

```
x <- x$pricepersqft
y<- y$pricepersqft
```

Hypothesis for distance Null Hypothesis -No Average differnece in house price based on Distance Alternate Hypothesis -There is Average difference in house pricesummary

Standard deviation of X

```
sd(x)

## [1] 121.7353
```

Standard deviation of Y

```
sd(y)

## [1] 76.98242
```

Two sample Ztest

```
t.test(x,sigma.x=121.73 ,y,sigma.y=76.98 ,alternative="two.sided"
,conf.level=0.95)

##
##  Welch Two Sample t-test
##
## data:  x and y
## t = 77.53, df = 18873, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  104.1533 109.5563
## sample estimates:
## mean of x mean of y
##  317.3277  210.4729
```

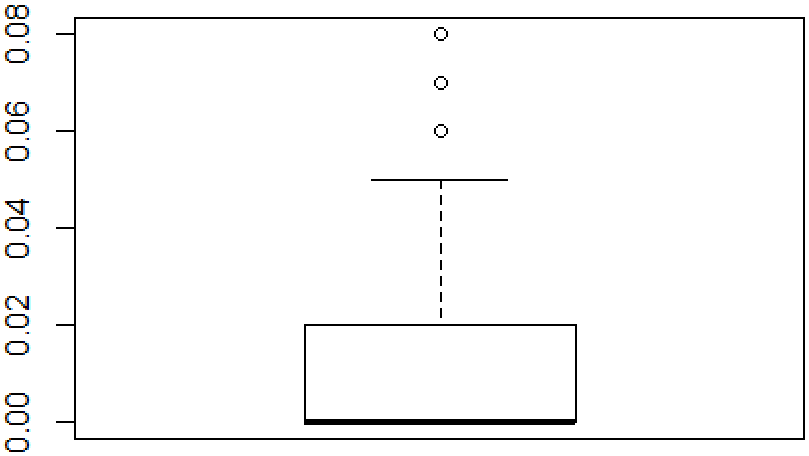
##Hypothesis Testing the effect of crime on price## Two Sample Z test for crime Calculating price persqft

```
fulldata$pricepersqft <- fulldata$price/fulldata$newsqft_living
names(fulldata)

##  [1] "zipcode"        "Distance"       "Income"         "time"
##  [5] "population"     "crime.count"    "burglary"       "theft"
##  [9] "b.t"            "id"             "date"           "Date.only"
## [13] "price"          "bedrooms"       "bathrooms"      "sqft_living"
## [17] "sqft_lot"       "floors"         "waterfront"     "view"
## [21] "condition"      "grade"          "sqft_above"     "sqft_basement"
## [25] "yr_built"       "yr_renovated"   "lat"            "long"
## [29] "sqft_living15"  "sqft_lot15"     "Age"            "year"
## [33] "newsqft_living" "newsqft_lot"    "crimerate"      "pricepersqft"
```

Analysing crimerate

```
summary(fulldata$crimerate)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## 0.00000 0.00000 0.00000 0.01437 0.02000 0.08000

boxplot(fulldata$crimerate)
```



```
names(fulldata)

##  [1] "zipcode"        "Distance"       "Income"         "time"          
##  [5] "population"     "crime.count"    "burglary"       "theft"         
##  [9] "b.t"            "id"             "date"           "Date.only"     
## [13] "price"          "bedrooms"       "bathrooms"      "sqft_living"   
## [17] "sqft_lot"       "floors"         "waterfront"     "view"          
## [21] "condition"      "grade"          "sqft_above"     "sqft_basement" 
## [25] "yr_built"       "yr_renovated"   "lat"            "long"          
## [29] "sqft_living15"  "sqft_lot15"     "Age"            "year"          
## [33] "newsqft_living" "newsqft_lot"    "crimerate"      "pricepersqft"
```

Storing price per square feet and crime in a data frame

```
hypdistance<- fulldata[c(35,36)]
```

First sample with zipcodes having crime rates lesser than median crime rate

```
x<- subset(hypdistance,crimerate <= 0.004)
```

Second sample with zipcodes having crime rates lesser than median crime rate

```
y<- subset(hypdistance,crimerate > 0.004)
```

Assigning the price per squaree feet of two samples to x and y

```
x<- x$pricepersqft
y<- y$pricepersqft
```

Hypothesis for crime Null Hypothesis -Crime has no effect on average price per square feet Alternate Hypothesis- Crime is contributing on average price per square feet

Standard deviation of X

```
sd(x)

## [1] 127.7349
```

Standard deviation of Y

```
sd(y)

## [1] 78.91234
```

Two sample Ztest

```
t.test(x,sigma.x=127.7317 ,y,sigma.y=78.9124 ,conf.level=0.95)

## 
##  Welch Two Sample t-test
## 
## data:  x and y
## t = 62.033, df = 18371, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  86.47902 92.12237
## sample estimates:
## mean of x mean of y 
##  309.2689  219.9682
```