House Pricing

Shashank Semwal

10 August 2018

```
house_data<-read.csv("C:/Users/Shashank/Documents/R/dataset/home credit default/kc_house_data.csv")
```

```
#ID column not necessary
house_data<-house_data[,-1]
# Structure
str(house_data)
```

```
## 'data.frame':    21613 obs. of  20 variables:
##  $ 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 ...
```

Calculating Variance Inflation Factor inorder to calculate multicollanity among the independent variable

```
house_data<-house_data %>% select(-sqft_living)
vif(lm(price~., house_data))
```

```
##                   GVIF  Df GVIF^(1/(2*Df))
## date          1.368161 371        1.000423
## bedrooms      1.681363   1        1.296674
## bathrooms     3.420949   1        1.849581
## sqft_lot      2.149613   1        1.466156
## floors        2.054294   1        1.433281
## waterfront    1.221874   1        1.105384
## view          1.460767   1        1.208622
## condition     1.274641   1        1.129000
## grade         3.481012   1        1.865747
## sqft_above    5.016523   1        2.239759
## sqft_basement 2.047548   1        1.430926
## yr_built      2.475763   1        1.573456
## yr_renovated  1.171242   1        1.082240
## zipcode       1.686680   1        1.298723
## lat           1.202523   1        1.096596
## long          1.854812   1        1.361915
## sqft_living15 3.029455   1        1.740533
## sqft_lot15    2.183443   1        1.477648
```

```
alias(lm(price~., house_data))
```

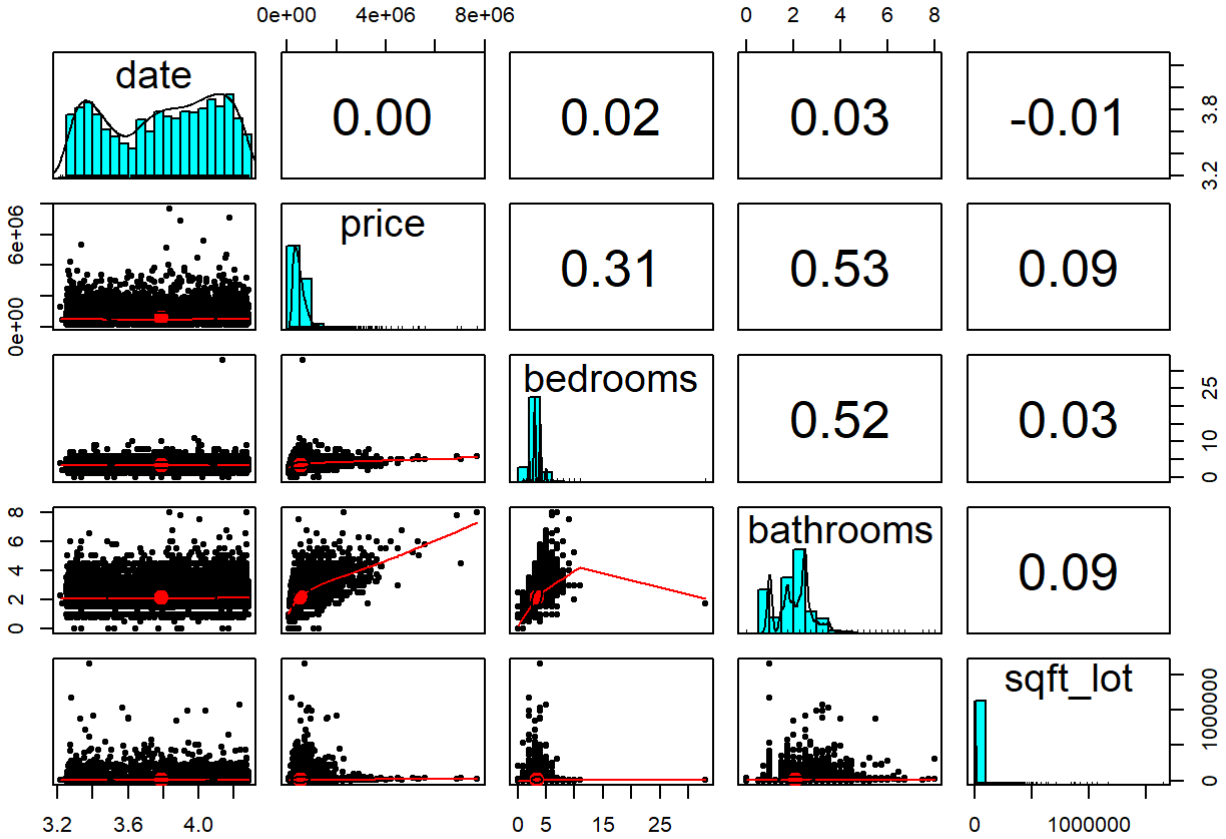
```
## Model :
## price ~ date + bedrooms + bathrooms + sqft_lot + floors + waterfront +
##     view + condition + grade + sqft_above + sqft_basement + yr_built +
##     yr_renovated + zipcode + lat + long + sqft_living15 + sqft_lot15
```

```
# we conclude that sqft_living is the variable giving us low variance hence we will remove it
```

```
# convert date to date datatype
house_data$date<-as.Date(strtrim(house_data$date,8),format="%Y%m%d")
# Age from Date
house_data$date<-as.numeric((Sys.Date()-house_data$date)/365)
```

```
# Sampling
sample<-sample(nrow(house_data),size = 0.7*nrow(house_data))
house_train<-house_data[sample,]
house_test<-house_data[-sample,]
```

```
pairs.panels(house_data[1:5])
```

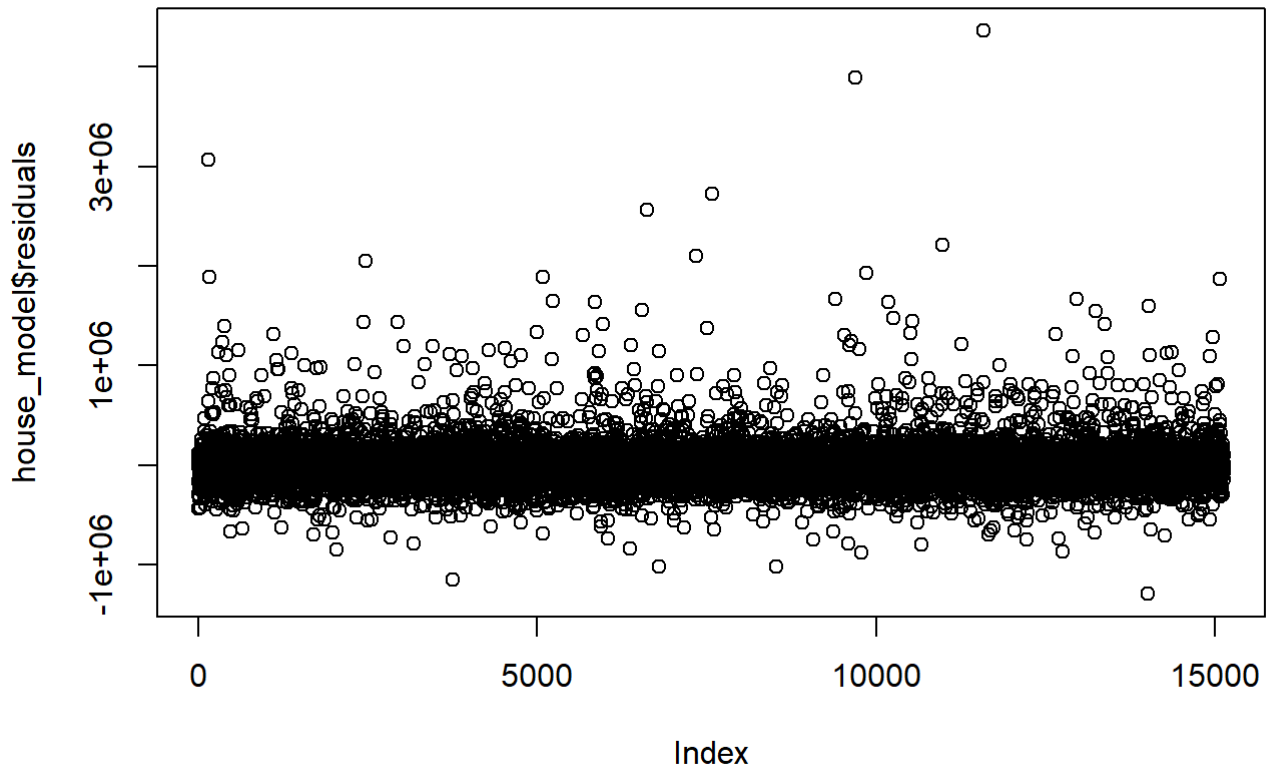



```
# MODEL
house_model<-lm(price~.,data = house_train)
house_predict<-predict(house_model,house_test[,-2])
```

```
summary(house_model)
```

```
##
## Call:
## lm(formula = price ~ ., data = house_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1295738   -99563    -9343    78597  4363178
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    8.896e+06  3.474e+06   2.561  0.01045 *
## date          -4.888e+04  5.286e+03  -9.246  < 2e-16 ***
## bedrooms      -3.300e+04  2.217e+03 -14.889  < 2e-16 ***
## bathrooms      3.944e+04  3.891e+03  10.136  < 2e-16 ***
## sqft_lot       1.084e-01  6.043e-02   1.794  0.07285 .
## floors         1.011e+04  4.279e+03   2.363  0.01814 *
## waterfront     5.825e+05  2.038e+04  28.580  < 2e-16 ***
## view           4.744e+04  2.567e+03  18.484  < 2e-16 ***
## condition      2.612e+04  2.779e+03   9.399  < 2e-16 ***
## grade          9.711e+04  2.566e+03  37.845  < 2e-16 ***
## sqft_above     1.779e+02  4.331e+00  41.077  < 2e-16 ***
## sqft_basement  1.493e+02  5.247e+00  28.457  < 2e-16 ***
## yr_built      -2.645e+03  8.624e+01 -30.671  < 2e-16 ***
## yr_renovated   2.013e+01  4.354e+00   4.622 3.83e-06 ***
## zipcode       -6.070e+02  3.923e+01 -15.475  < 2e-16 ***
## lat            5.983e+05  1.278e+04  46.819  < 2e-16 ***
## long          -2.199e+05  1.547e+04 -14.221  < 2e-16 ***
## sqft_living15  2.376e+01  4.118e+00   5.769 8.13e-09 ***
## sqft_lot15    -2.770e-01  8.953e-02  -3.094  0.00198 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 200200 on 15110 degrees of freedom
## Multiple R-squared:  0.7004, Adjusted R-squared:    0.7
## F-statistic:  1962 on 18 and 15110 DF,  p-value: < 2.2e-16
```

```
# Check for Heteroskedasticity
plot(house_model$residuals)
```

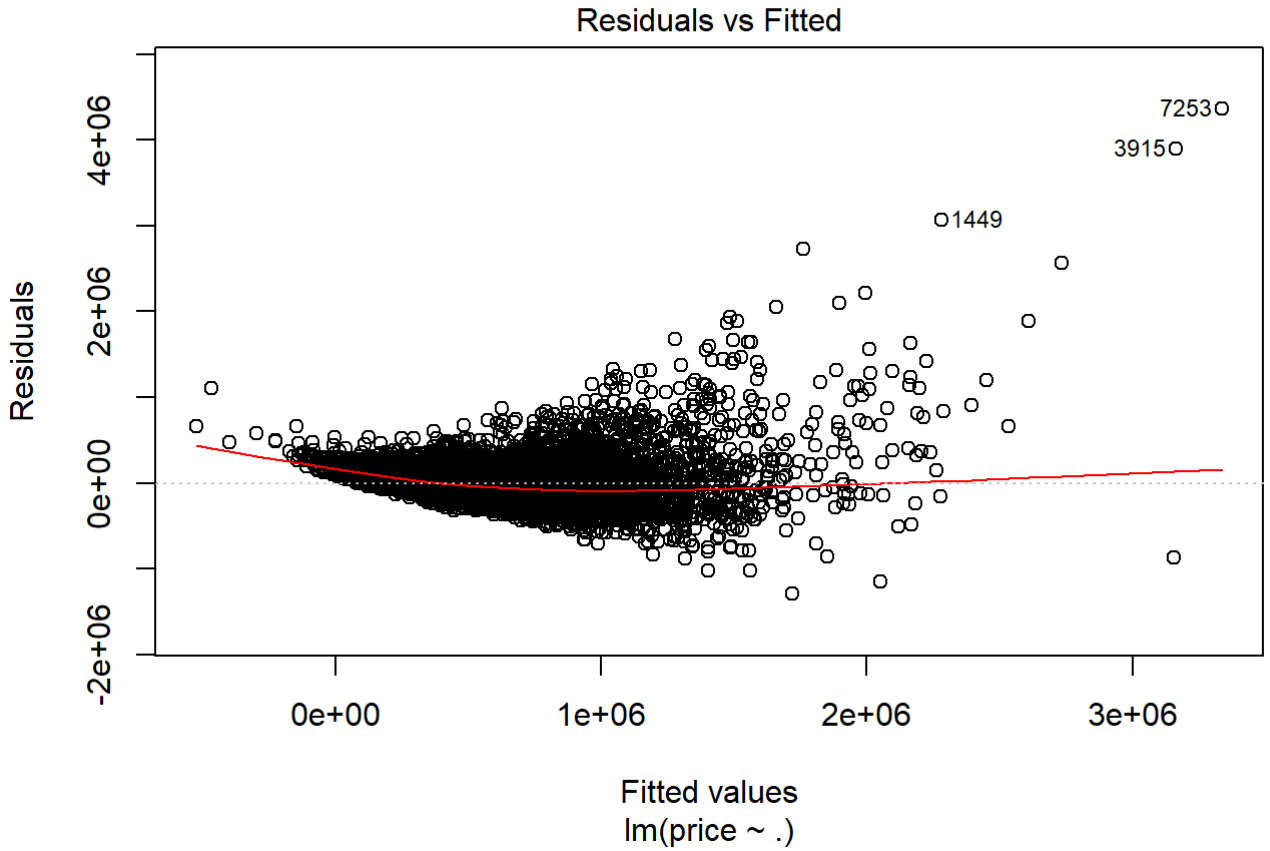


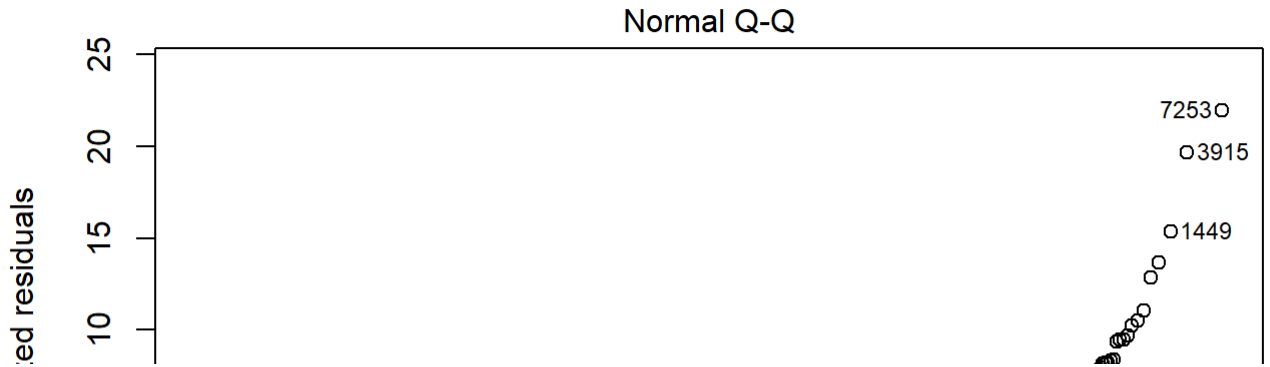

```
#we can see there is no Hetroskedasticity
```

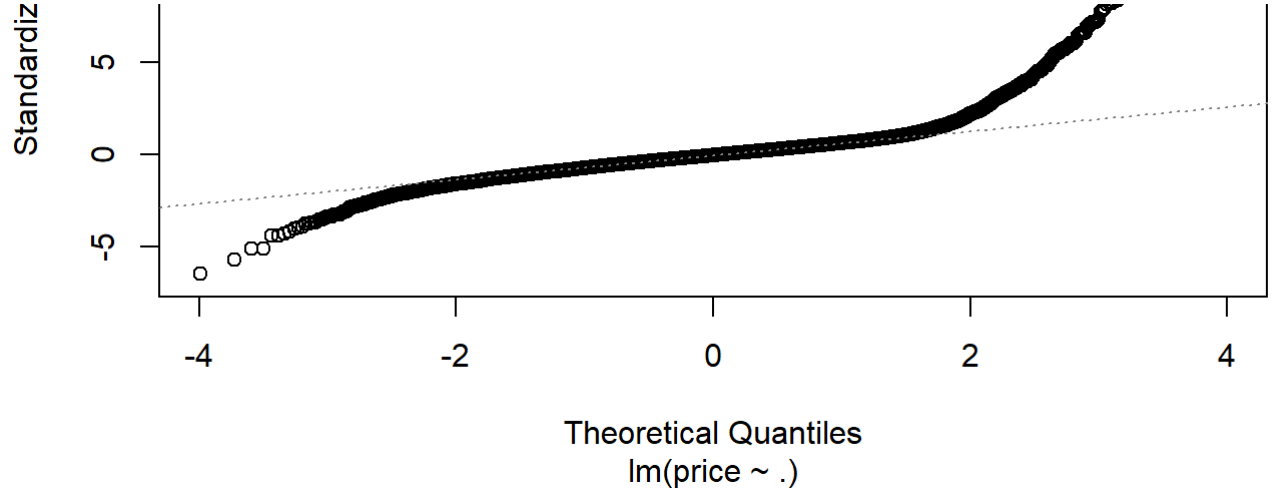
```
house_rmse<-sqrt(sum((house_model$residuals^2))/nrow(house_train))
house_rmse
```

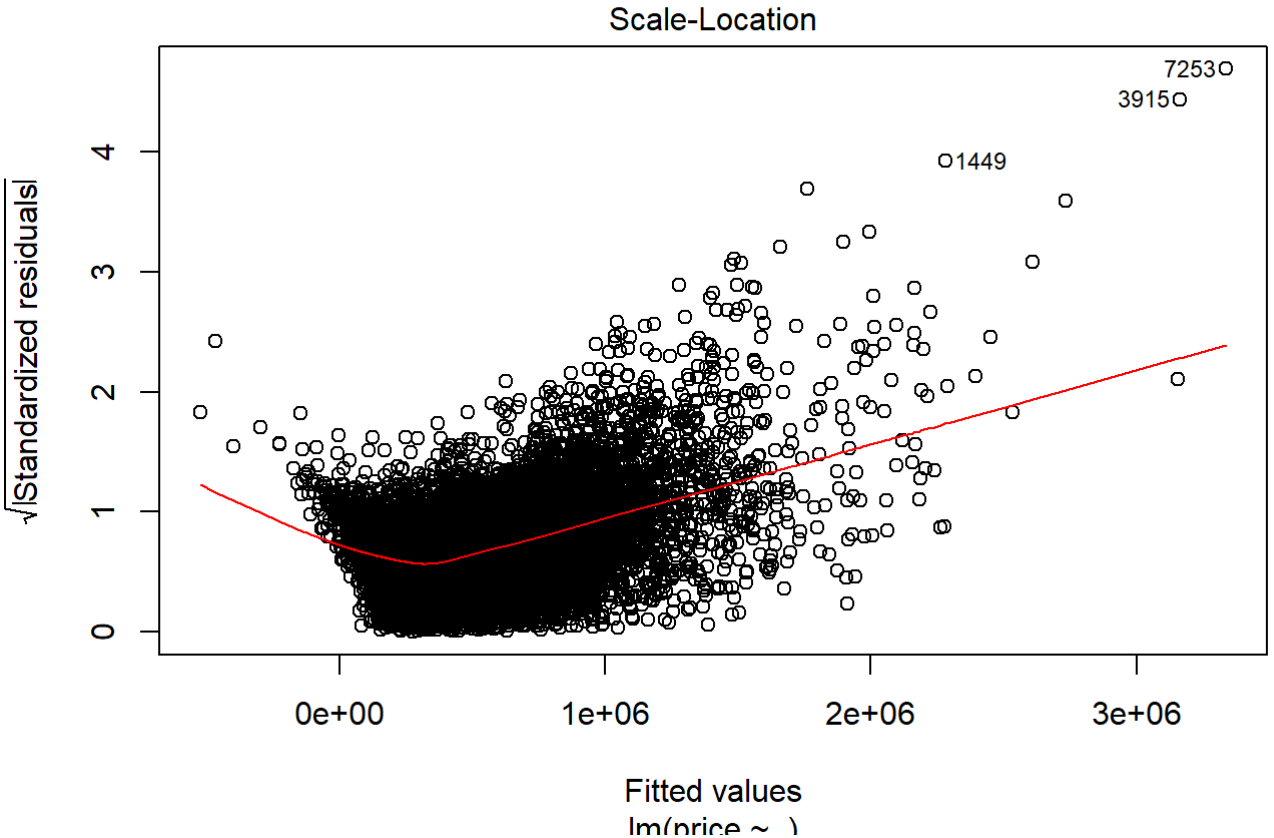
```
## [1] 200085.9
```

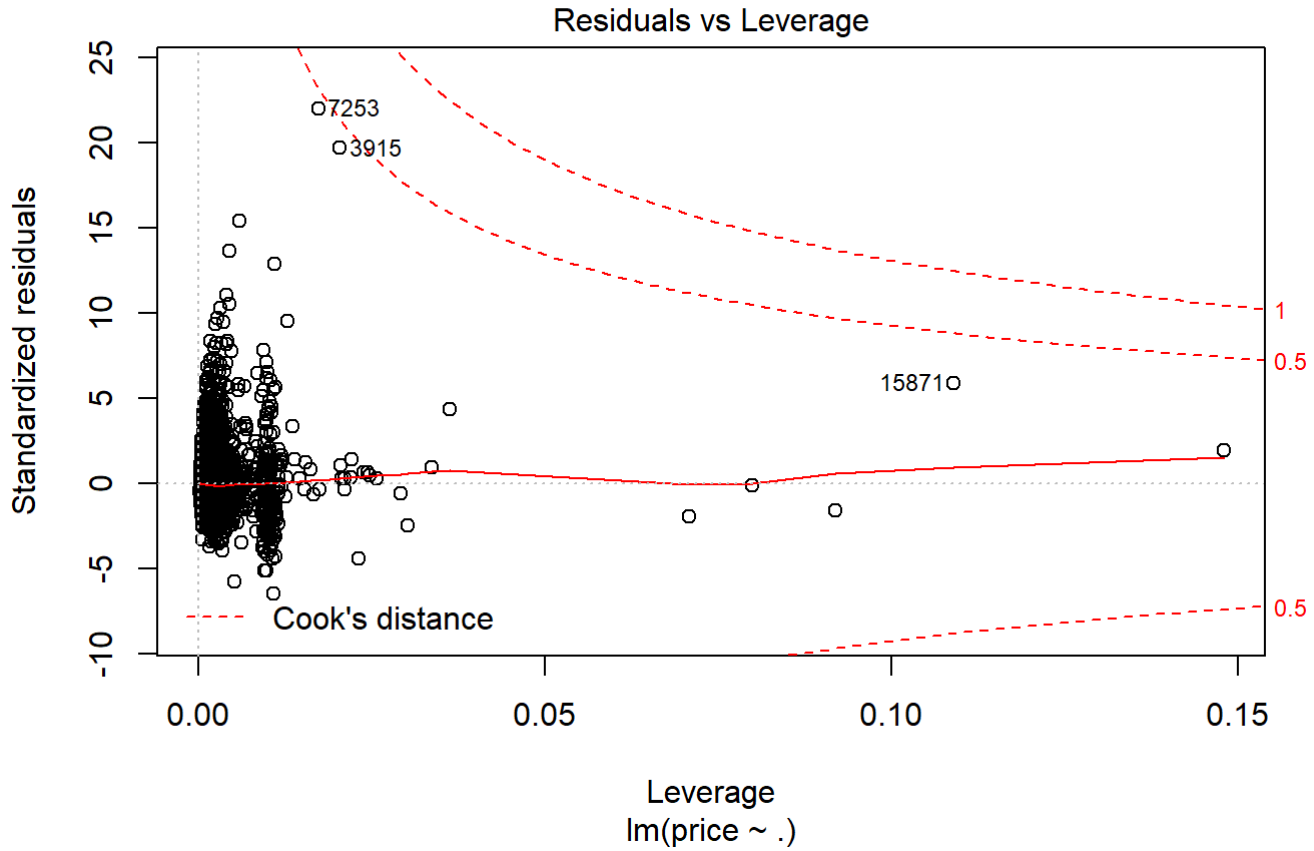
```
plot(house_model)
```


Residuals vs Fitted
Residuals
Fitted values
lm(price ~ .)
7253
3915
1449
4e+06
2e+06
0e+00
-2e+06
0e+00
1e+06
2e+06
3e+06


Normal Q-Q
7253
3915
1449
25
20
15
10


Standardiz
5
0
-5
-4
-2
0
2
4
Theoretical Quantiles
lm(price ~ .)


Scale-Location
7253
3915
1449
√|Standardized residuals|
0
1
2
3
4
0e+00
1e+06
2e+06
3e+06
Fitted values
lm(price ~ .)




```
# R Square
house_rsquare<- 1 - sum((house_model$residuals^2))/sum((house_train$price-mean(house_train$price))^2)
```

```
#Adjusted R Square
house_adjustRSquare<-(1-house_rsquare)*17/(nrow(house_train)-17)
```

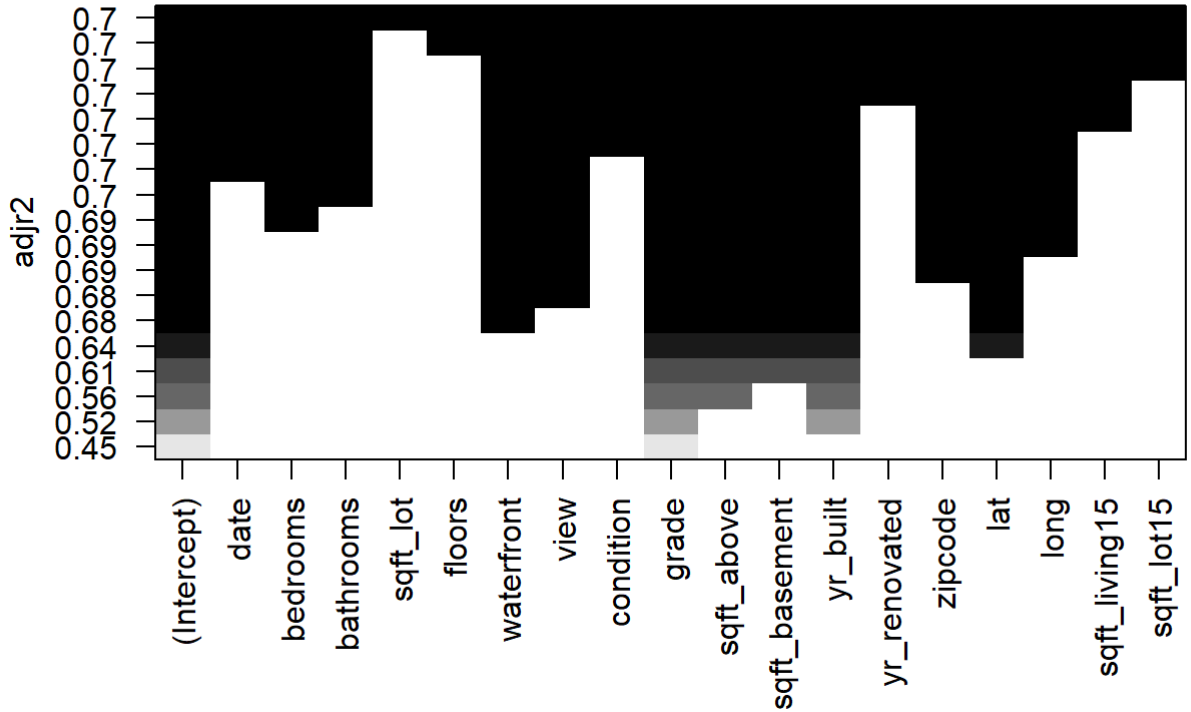
REGULARIZATION

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 3.4.4
```

```
house_reg<-regsubsets(x = price~.,data = house_train,nvmax = 19,method = "backward")

plot(house_reg,scale = 'adjr2')
```




```
# SO WE CONCLUDE THAT OUT REGULARIZATION ON TRAINING SET INCREASES THE MODEL ACCURACY
```