Loading the cleaned housing data and also the required packages:

```
housing <- read.csv("C:\\Users\\dell\\Downloads\\house_2.csv", header=TRUE, stringsAsFactors=FALSE)
library(ggcorrplot)
```

```
## Loading required package: ggplot2
```

```
library(Hmisc)
```

```
## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(car)
```

```
## Loading required package: carData
```

```
library(caret)
```

```
##
## Attaching package: 'caret'

## The following object is masked from 'package:survival':
##
##     cluster
```

```
library(glmnet)
```

```
## Loading required package: Matrix

## Loaded glmnet 4.1-4
```

```
library(leaps)
library(xgboost)
```

```
## Warning: package 'xgboost' was built under R version 4.2.2
```

Viewing the data and the dattype of each column

```
str(housing)
```

```
## 'data.frame':    151951 obs. of  24 variables:
##  $ Lng                : num  116 116 117 116 116 ...
##  $ Lat                : num  40 39.9 39.9 40.1 39.9 ...
##  $ tradeTime          : chr  "09-08-16" "28-07-16" "11-12-16" "30-09-16" ...
##  $ DOM                : int  1464 903 1271 965 927 861 851 904 873 865 ...
##  $ followers          : int  106 126 48 138 286 57 167 138 218 134 ...
##  $ totalPrice         : num  415 575 1030 298 392 ...
##  $ price              : int  31680 43436 52021 22202 48396 52000 37672 49521 27917 55883 ...
##  $ square             : num  131 132 198 134 81 ...
##  $ livingRoom         : int  2 2 3 3 2 1 2 3 1 1 ...
##  $ drawingRoom        : int  1 2 2 1 1 0 1 2 0 0 ...
##  $ kitchen            : int  1 1 1 1 1 1 1 1 1 0 ...
##  $ bathRoom           : int  1 2 3 1 1 1 1 2 1 0 ...
##  $ buildingType       : int  1 1 4 1 4 4 4 1 3 1 ...
##  $ constructionTime   : int  2005 2004 2005 2008 1960 2005 1997 2004 2009 2009 ...
##  $ renovationCondition: int  3 4 3 1 2 3 4 4 1 4 ...
##  $ buildingStructure  : int  6 6 6 6 2 6 2 6 2 6 ...
##  $ ladderRatio        : num  0.217 0.667 0.5 0.273 0.333 0.333 0.5 0.667 0.333 0.308 ...
##  $ elevator           : int  1 1 1 1 0 1 0 1 0 1 ...
##  $ fiveYearsProperty  : int  0 1 0 0 1 1 0 1 0 1 ...
##  $ subway             : int  1 0 0 0 1 0 0 0 0 1 ...
##  $ district           : int  7 7 7 6 1 7 7 7 13 1 ...
##  $ communityAverage   : int  56021 71539 48160 51238 62588 67738 50112 71539 44235 78590 ...
##  $ floorType          : int  1 1 2 3 2 2 1 1 1 2 ...
##  $ floorNumber        : int  26 22 4 21 6 8 6 22 10 23 ...
```

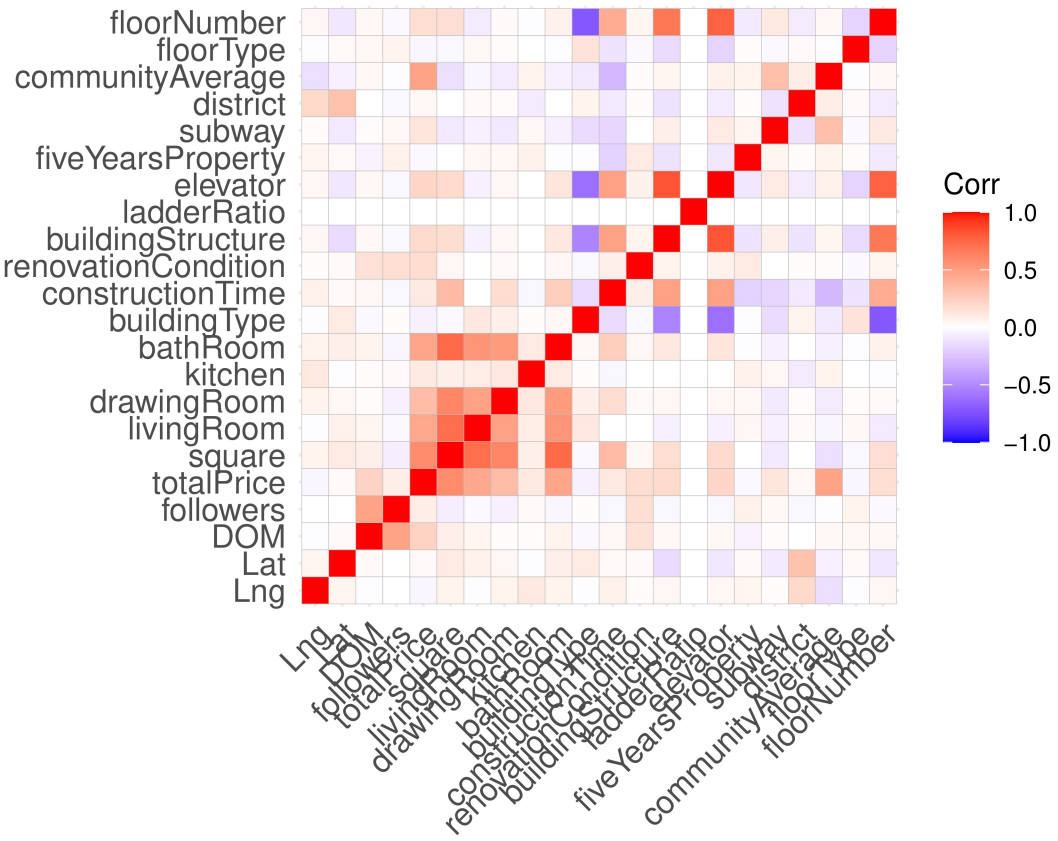
```
View(housing)
dim(housing)
```

```
## [1] 151951     24
```

```
#the histogram of the variables
#hist.data.frame(housing)
```

```
#Removing the tradetime and price data from the data frame

housing$tradeTime <- NULL
housing$price <- NULL
```

```
#this heatmap shows the relationship of the different variables to each other

ggcorrplot(cor(housing), lab=F)
```



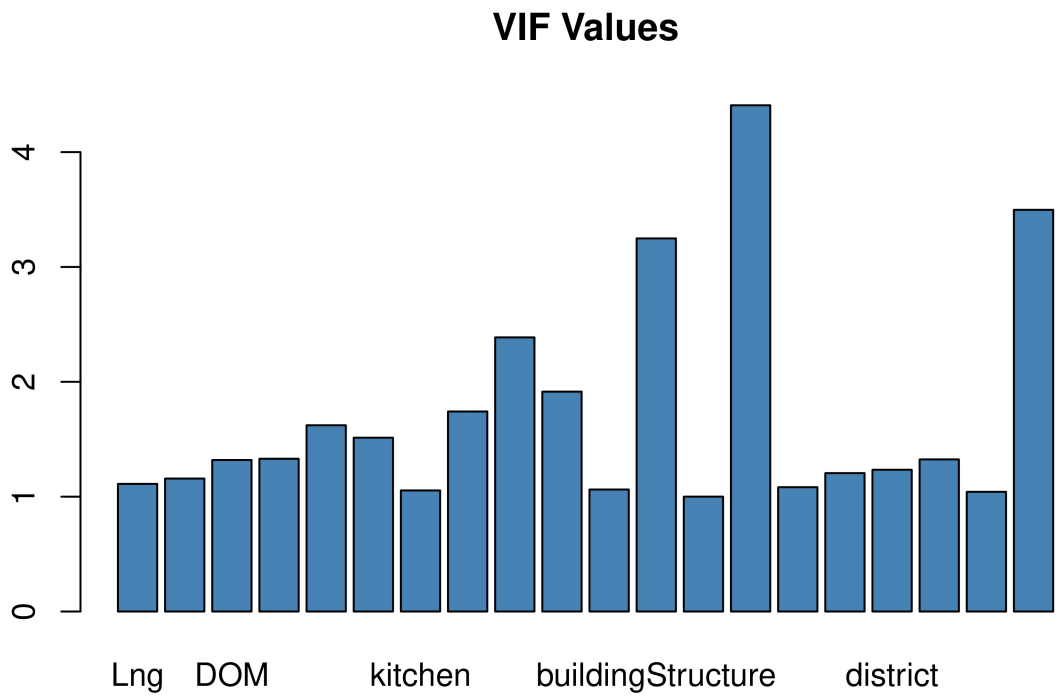

```
#Removing the square variable as it is redundant

housing$square <- NULL
```

```
#fitting a regreesion model to find out the vif values of different variables

linear.regression.model <- lm(totalPrice~.,data = housing)
```

```
#Code for VIF values and their bar plot

vif(linear.regression.model)
```

```
##                 Lng                 Lat                 DOM           followers
##            1.111095            1.157560            1.319191            1.330059
##          livingRoom         drawingRoom             kitchen            bathRoom
##            1.621412            1.513435            1.054139            1.741337
##        buildingType    constructionTime renovationCondition   buildingStructure
##            2.386727            1.914461            1.062230            3.248399
##         ladderRatio            elevator   fiveYearsProperty              subway
##            1.000079            4.407605            1.082585            1.205047
##            district   communityAverage           floorType         floorNumber
##            1.234266            1.324479            1.042120            3.497194
```

```r
vif_values <- vif(linear.regression.model)
barplot(vif_values, main = "VIF Values", col = "steelblue",names.arg = c("Lng","Lat","DOM","followers",
```




```r
#Splitting the data into training and testing

index = sample(1:nrow(housing), 0.7*nrow(housing))


train = housing[index,] # Create the training data

test = housing[-index,] # Create the test data


dim(train)
```

```
## [1] 106365     21
```

```r
dim(test)
```

```
## [1] 45586    21
```

```r
#Scaling and centering the different variables

housing$floorType <- as.factor(housing$floorType)
```

```
cols = c("Lng","Lat","DOM","followers","livingRoom","drawingRoom","kitchen","bathRoom","buildingType","

pre_proc_val <- preProcess(train[,cols], method = c("center", "scale"))

train[,cols] = predict(pre_proc_val, train[,cols])

test[,cols] = predict(pre_proc_val, test[,cols])


summary(train)
```

```
##       Lng                 Lat                DOM             followers
##  Min.   :-3.01342   Min.   :-3.3850   Min.   :-0.5666   Min.   :-0.6024
##  1st Qu.:-0.62721   1st Qu.:-0.5953   1st Qu.:-0.5666   1st Qu.:-0.5576
##  Median : 0.01543   Median :-0.1514   Median :-0.4470   Median :-0.3338
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.54624   3rd Qu.: 0.5684   3rd Qu.: 0.1710   3rd Qu.: 0.1361
##  Max.   : 2.61715   Max.   : 3.1945   Max.   :32.8440   Max.   :24.9749
##    totalPrice       livingRoom        drawingRoom         kitchen
##  Min.   :   0.1   Min.   :-2.60581   Min.   :-2.2536   Min.   :-8.43924
##  1st Qu.: 248.0   1st Qu.:-1.31068   1st Qu.:-0.2855   1st Qu.: 0.07121
##  Median : 355.0   Median :-0.01554   Median :-0.2855   Median : 0.07121
##  Mean   : 411.4   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.00000
##  3rd Qu.: 502.0   3rd Qu.:-0.01554   3rd Qu.:-0.2855   3rd Qu.: 0.07121
##  Max.   :4800.0   Max.   : 6.46015   Max.   : 7.5870   Max.   :17.09211
##     bathRoom        buildingType     constructionTime  renovationCondition
##  Min.   :-2.7705   Min.   :-1.6177   Min.   :-5.4932   Min.   :-1.77021
##  1st Qu.:-0.4262   1st Qu.:-1.6177   1st Qu.:-0.5986   1st Qu.:-0.01826
##  Median :-0.4262   Median : 0.7631   Median : 0.1801   Median :-0.01826
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000
##  3rd Qu.:-0.4262   3rd Qu.: 0.7631   3rd Qu.: 0.7363   3rd Qu.: 0.85772
##  Max.   :11.2952   Max.   : 0.7631   Max.   : 1.8487   Max.   : 0.85772
##  buildingStructure  ladderRatio          elevator       fiveYearsProperty
##  Min.   :-1.8059   Min.   : -0.0031   Min.   :-1.1621   Min.   :-1.2565
##  1st Qu.:-1.2811   1st Qu.: -0.0031   1st Qu.:-1.1621   1st Qu.:-1.2565
##  Median : 0.8183   Median : -0.0031   Median : 0.8605   Median : 0.7959
##  Mean   : 0.0000   Mean   :  0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.8183   3rd Qu.: -0.0031   3rd Qu.: 0.8605   3rd Qu.: 0.7959
##  Max.   : 0.8183   Max.   :326.1334   Max.   : 0.8605   Max.   : 0.7959
##      subway           district       communityAverage    floorType
##  Min.   :-1.2185   Min.   :-2.0002   Min.   :-2.1856   Min.   :-1.3827
##  1st Qu.:-1.2185   1st Qu.:-0.2420   1st Qu.:-0.7813   1st Qu.:-0.3211
##  Median : 0.8206   Median : 0.1096   Median :-0.1961   Median :-0.3211
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.8206   3rd Qu.: 0.4612   3rd Qu.: 0.5531   3rd Qu.: 0.7405
##  Max.   : 0.8206   Max.   : 2.2194   Max.   : 5.3947   Max.   : 1.8020
##   floorNumber
##  Min.   :-1.5697
##  1st Qu.:-0.9294
##  Median :-0.2891
##  Mean   : 0.0000
##  3rd Qu.: 0.7354
##  Max.   : 6.3702
```

```r
#fitting the linear regression model on the training data

linear.model <- lm(totalPrice~.,data = train)
summary(linear.model)
```

```
##
## Call:
## lm(formula = totalPrice ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1201.3   -87.1    -7.1    75.7  3360.1
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         411.3766     0.4858 846.854  < 2e-16 ***
## Lng                   0.1493     0.5117   0.292 0.770460
## Lat                   3.8594     0.5225   7.386 1.52e-13 ***
## DOM                  35.4706     0.5594  63.409  < 2e-16 ***
## followers             9.3217     0.5613  16.608  < 2e-16 ***
## livingRoom           77.9611     0.6185 126.043  < 2e-16 ***
## drawingRoom          23.6572     0.5979  39.564  < 2e-16 ***
## kitchen               6.0031     0.4990  12.030  < 2e-16 ***
## bathRoom             54.1894     0.6412  84.511  < 2e-16 ***
## buildingType         12.3608     0.7497  16.489  < 2e-16 ***
## constructionTime     31.8252     0.6730  47.287  < 2e-16 ***
## renovationCondition  31.6375     0.5006  63.204  < 2e-16 ***
## buildingStructure     6.3170     0.8760   7.211 5.58e-13 ***
## ladderRatio           0.3284     0.4858   0.676 0.499067
## elevator              7.0279     1.0204   6.888 5.70e-12 ***
## fiveYearsProperty    -9.6709     0.5054 -19.134  < 2e-16 ***
## subway                1.4384     0.5336   2.696 0.007023 **
## district             -1.9270     0.5397  -3.571 0.000356 ***
## communityAverage    136.1412     0.5590 243.529  < 2e-16 ***
## floorType            -5.6077     0.4960 -11.306  < 2e-16 ***
## floorNumber          21.0945     0.9084  23.222  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 158.4 on 106344 degrees of freedom
## Multiple R-squared:  0.6146, Adjusted R-squared:  0.6145
## F-statistic:  8478 on 20 and 106344 DF,  p-value: < 2.2e-16
```

Based on the p-values and significance level = 0.05, lng and ladder ratio are not statistically not significant.

```r
#R2 and rmse on training data

summary(linear.model)$r.squared
```

```
## [1] 0.6145682
```

```r
predictions = matrix(predict(linear.model,train))
MSE = mean((predictions-train$totalPrice)^2)
MSE
```

```
## [1] 25094.33
```

```r
#rmse on testing data

predictions = matrix(predict(linear.model,test))
MSE = mean((predictions-test$totalPrice)^2)
MSE
```

```
## [1] 24533.29
```

```r
#Using the step function to find out the best model based on AIC

step(linear.model, direction="both", k = 2)
```

```
## Start:  AIC=1077562
## totalPrice ~ Lng + Lat + DOM + followers + livingRoom + drawingRoom +
##     kitchen + bathRoom + buildingType + constructionTime + renovationCondition +
##     buildingStructure + ladderRatio + elevator + fiveYearsProperty +
##     subway + district + communityAverage + floorType + floorNumber
##
##                        Df  Sum of Sq        RSS     AIC
## - Lng                   1       2137 2669160154 1077560
## - ladderRatio           1      11468 2669169485 1077560
## <none>                               2669158017 1077562
## - subway                1     182404 2669340421 1077567
## - district              1     320015 2669478032 1077572
## - elevator              1    1190712 2670348729 1077607
## - buildingStructure     1    1305221 2670463238 1077612
## - Lat                   1    1369251 2670527268 1077614
## - floorType             1    3208277 2672366294 1077687
## - kitchen               1    3632128 2672790145 1077704
## - buildingType          1    6823833 2675981850 1077831
## - followers             1    6922808 2676080825 1077835
## - fiveYearsProperty     1    9189229 2678347246 1077925
## - floorNumber           1   13534970 2682692988 1078098
## - drawingRoom           1   39288740 2708446757 1079114
## - constructionTime      1   56124135 2725282152 1079773
## - renovationCondition   1  100266349 2769424366 1081482
## - DOM                   1  100917552 2770075569 1081507
## - bathRoom              1  179261305 2848419322 1084474
## - livingRoom            1  398747537 3067905554 1092369
## - communityAverage      1 1488546842 4157704859 1124701
##
## Step:  AIC=1077560
## totalPrice ~ Lat + DOM + followers + livingRoom + drawingRoom +
##     kitchen + bathRoom + buildingType + constructionTime + renovationCondition +
##     buildingStructure + ladderRatio + elevator + fiveYearsProperty +
##     subway + district + communityAverage + floorType + floorNumber
```

```
##
##                       Df  Sum of Sq        RSS     AIC
## - ladderRatio          1      11491 2669171645 1077558
## <none>                              2669160154 1077560
## + Lng                  1       2137 2669158017 1077562
## - subway               1     187458 2669347612 1077565
## - district             1     325842 2669485995 1077571
## - elevator             1    1193043 2670353196 1077605
## - buildingStructure    1    1306826 2670466980 1077610
## - Lat                  1    1367342 2670527495 1077612
## - floorType            1    3209057 2672369210 1077686
## - kitchen              1    3728719 2672888873 1077706
## - buildingType         1    6828149 2675988303 1077830
## - followers            1    6925714 2676085868 1077833
## - fiveYearsProperty    1    9202202 2678362355 1077924
## - floorNumber          1   13536493 2682696647 1078096
## - drawingRoom          1   39354183 2708514337 1079115
## - constructionTime     1   56142711 2725302865 1079772
## - renovationCondition  1  100265985 2769426138 1081480
## - DOM                  1  100922830 2770082984 1081505
## - bathRoom             1  179611445 2848771599 1084485
## - livingRoom           1  400114114 3069274267 1092415
## - communityAverage     1 1530345978 4199506131 1125763
##
## Step:  AIC=1077558
## totalPrice ~ Lat + DOM + followers + livingRoom + drawingRoom +
##     kitchen + bathRoom + buildingType + constructionTime + renovationCondition +
##     buildingStructure + elevator + fiveYearsProperty + subway +
##     district + communityAverage + floorType + floorNumber
##
##                       Df  Sum of Sq        RSS     AIC
## <none>                              2669171645 1077558
## + ladderRatio          1      11491 2669160154 1077560
## + Lng                  1       2159 2669169485 1077560
## - subway               1     187041 2669358686 1077564
## - district             1     325850 2669497495 1077569
## - elevator             1    1193922 2670365567 1077604
## - buildingStructure    1    1307332 2670478977 1077608
## - Lat                  1    1367422 2670539067 1077611
## - floorType            1    3209338 2672380983 1077684
## - kitchen              1    3728889 2672900533 1077705
## - buildingType         1    6829403 2676001048 1077828
## - followers            1    6925511 2676097156 1077832
## - fiveYearsProperty    1    9200905 2678372550 1077922
## - floorNumber          1   13535460 2682707104 1078094
## - drawingRoom          1   39355530 2708527175 1079113
## - constructionTime     1   56134985 2725306630 1079770
## - renovationCondition  1  100272539 2769444184 1081479
## - DOM                  1  100920907 2770092552 1081504
## - bathRoom             1  179614552 2848786197 1084483
## - livingRoom           1  400102899 3069274544 1092413
## - communityAverage     1 1530334945 4199506590 1125761

##
```

```
## Call:
## lm(formula = totalPrice ~ Lat + DOM + followers + livingRoom +
##     drawingRoom + kitchen + bathRoom + buildingType + constructionTime +
##     renovationCondition + buildingStructure + elevator + fiveYearsProperty +
##     subway + district + communityAverage + floorType + floorNumber,
##     data = train)
##
## Coefficients:
##         (Intercept)                  Lat                  DOM
##             411.377                3.855               35.468
##           followers           livingRoom          drawingRoom
##               9.323               77.948               23.664
##             kitchen             bathRoom         buildingType
##               6.023               54.198               12.365
##    constructionTime  renovationCondition    buildingStructure
##              31.825               31.638                6.322
##            elevator    fiveYearsProperty               subway
##               7.036               -9.662                1.451
##            district     communityAverage            floorType
##              -1.890              136.113               -5.609
##         floorNumber
##              21.095
```

Based on AIC: Lng and ladder ratio are not significant.

```
#Finding the best model using regsubsets function based on mallows Cp

RSSleaps = regsubsets(x = as.matrix(train[, -5]), y = train[, 5], nvmax = 21)
sumleaps = summary(RSSleaps, matrix = T)
sumleaps$which
```

```
##    (Intercept)   Lng   Lat   DOM followers livingRoom drawingRoom kitchen
## 1         TRUE FALSE FALSE FALSE     FALSE      FALSE       FALSE   FALSE
## 2         TRUE FALSE FALSE FALSE     FALSE      FALSE       FALSE   FALSE
## 3         TRUE FALSE FALSE FALSE     FALSE       TRUE       FALSE   FALSE
## 4         TRUE FALSE FALSE FALSE     FALSE       TRUE       FALSE   FALSE
## 5         TRUE FALSE FALSE  TRUE     FALSE       TRUE       FALSE   FALSE
## 6         TRUE FALSE FALSE  TRUE     FALSE       TRUE       FALSE   FALSE
## 7         TRUE FALSE FALSE  TRUE     FALSE       TRUE        TRUE   FALSE
## 8         TRUE FALSE FALSE  TRUE     FALSE       TRUE        TRUE   FALSE
## 9         TRUE FALSE FALSE  TRUE     FALSE       TRUE        TRUE   FALSE
## 10        TRUE FALSE FALSE  TRUE      TRUE       TRUE        TRUE   FALSE
## 11        TRUE FALSE FALSE  TRUE      TRUE       TRUE        TRUE   FALSE
## 12        TRUE FALSE FALSE  TRUE      TRUE       TRUE        TRUE   FALSE
## 13        TRUE FALSE FALSE  TRUE      TRUE       TRUE        TRUE    TRUE
## 14        TRUE FALSE FALSE  TRUE      TRUE       TRUE        TRUE    TRUE
## 15        TRUE FALSE FALSE  TRUE      TRUE       TRUE        TRUE    TRUE
## 16        TRUE FALSE  TRUE  TRUE      TRUE       TRUE        TRUE    TRUE
## 17        TRUE FALSE  TRUE  TRUE      TRUE       TRUE        TRUE    TRUE
## 18        TRUE FALSE  TRUE  TRUE      TRUE       TRUE        TRUE    TRUE
## 19        TRUE FALSE  TRUE  TRUE      TRUE       TRUE        TRUE    TRUE
## 20        TRUE  TRUE  TRUE  TRUE      TRUE       TRUE        TRUE    TRUE
##    bathRoom buildingType constructionTime renovationCondition buildingStructure
```

```
## 1     FALSE        FALSE            FALSE               FALSE             FALSE
## 2      TRUE        FALSE            FALSE               FALSE             FALSE
## 3      TRUE        FALSE            FALSE               FALSE             FALSE
## 4      TRUE        FALSE             TRUE               FALSE             FALSE
## 5      TRUE        FALSE             TRUE               FALSE             FALSE
## 6      TRUE        FALSE             TRUE                TRUE             FALSE
## 7      TRUE        FALSE             TRUE                TRUE             FALSE
## 8      TRUE        FALSE             TRUE                TRUE             FALSE
## 9      TRUE        FALSE             TRUE                TRUE             FALSE
## 10     TRUE        FALSE             TRUE                TRUE             FALSE
## 11     TRUE         TRUE             TRUE                TRUE             FALSE
## 12     TRUE         TRUE             TRUE                TRUE             FALSE
## 13     TRUE         TRUE             TRUE                TRUE             FALSE
## 14     TRUE         TRUE             TRUE                TRUE             FALSE
## 15     TRUE         TRUE             TRUE                TRUE              TRUE
## 16     TRUE         TRUE             TRUE                TRUE              TRUE
## 17     TRUE         TRUE             TRUE                TRUE              TRUE
## 18     TRUE         TRUE             TRUE                TRUE              TRUE
## 19     TRUE         TRUE             TRUE                TRUE              TRUE
## 20     TRUE         TRUE             TRUE                TRUE              TRUE
##    ladderRatio elevator fiveYearsProperty subway district communityAverage
## 1        FALSE    FALSE             FALSE  FALSE    FALSE             TRUE
## 2        FALSE    FALSE             FALSE  FALSE    FALSE             TRUE
## 3        FALSE    FALSE             FALSE  FALSE    FALSE             TRUE
## 4        FALSE    FALSE             FALSE  FALSE    FALSE             TRUE
## 5        FALSE    FALSE             FALSE  FALSE    FALSE             TRUE
## 6        FALSE    FALSE             FALSE  FALSE    FALSE             TRUE
## 7        FALSE    FALSE             FALSE  FALSE    FALSE             TRUE
## 8        FALSE    FALSE             FALSE  FALSE    FALSE             TRUE
## 9        FALSE    FALSE              TRUE  FALSE    FALSE             TRUE
## 10       FALSE    FALSE              TRUE  FALSE    FALSE             TRUE
## 11       FALSE    FALSE              TRUE  FALSE    FALSE             TRUE
## 12       FALSE     TRUE              TRUE  FALSE    FALSE             TRUE
## 13       FALSE     TRUE              TRUE  FALSE    FALSE             TRUE
## 14       FALSE     TRUE              TRUE  FALSE    FALSE             TRUE
## 15       FALSE     TRUE              TRUE  FALSE    FALSE             TRUE
## 16       FALSE     TRUE              TRUE  FALSE    FALSE             TRUE
## 17       FALSE     TRUE              TRUE  FALSE     TRUE             TRUE
## 18       FALSE     TRUE              TRUE   TRUE     TRUE             TRUE
## 19        TRUE     TRUE              TRUE   TRUE     TRUE             TRUE
## 20        TRUE     TRUE              TRUE   TRUE     TRUE             TRUE
##    floorType floorNumber
## 1      FALSE       FALSE
## 2      FALSE       FALSE
## 3      FALSE       FALSE
## 4      FALSE       FALSE
## 5      FALSE       FALSE
## 6      FALSE       FALSE
## 7      FALSE       FALSE
## 8      FALSE        TRUE
## 9      FALSE        TRUE
## 10     FALSE        TRUE
## 11     FALSE        TRUE
## 12     FALSE        TRUE
```

```
## 13     FALSE        TRUE
## 14      TRUE        TRUE
## 15      TRUE        TRUE
## 16      TRUE        TRUE
## 17      TRUE        TRUE
## 18      TRUE        TRUE
## 19      TRUE        TRUE
## 20      TRUE        TRUE
```

```
sumleaps$cp
```

```
##  [1] 108173.02352  43765.28203  27440.33386  16706.85313   8407.63788
##  [6]   4200.47126   2731.16108   1375.73959   1039.08698    768.58571
## [11]    595.26236    404.52001    246.43601    119.04249     76.43697
## [16]     37.06711     22.99501     17.54295     19.08513     21.00000
```

Based on mallows cp: lng and ladder ratio are statistically not significant.

```
#Building a linear regression model without ladder ratio and lng

linear.model.new <- lm(formula = totalPrice ~ Lat + DOM + followers + livingRoom +
    drawingRoom + kitchen + bathRoom + buildingType + constructionTime +
    renovationCondition + buildingStructure + elevator + fiveYearsProperty +
    subway + district + communityAverage + floorType + floorNumber,
    data = train)

predictions = matrix(predict(linear.model.new,train))
MSE = mean((predictions-train$totalPrice)^2, na.rm = T)
MSE
```

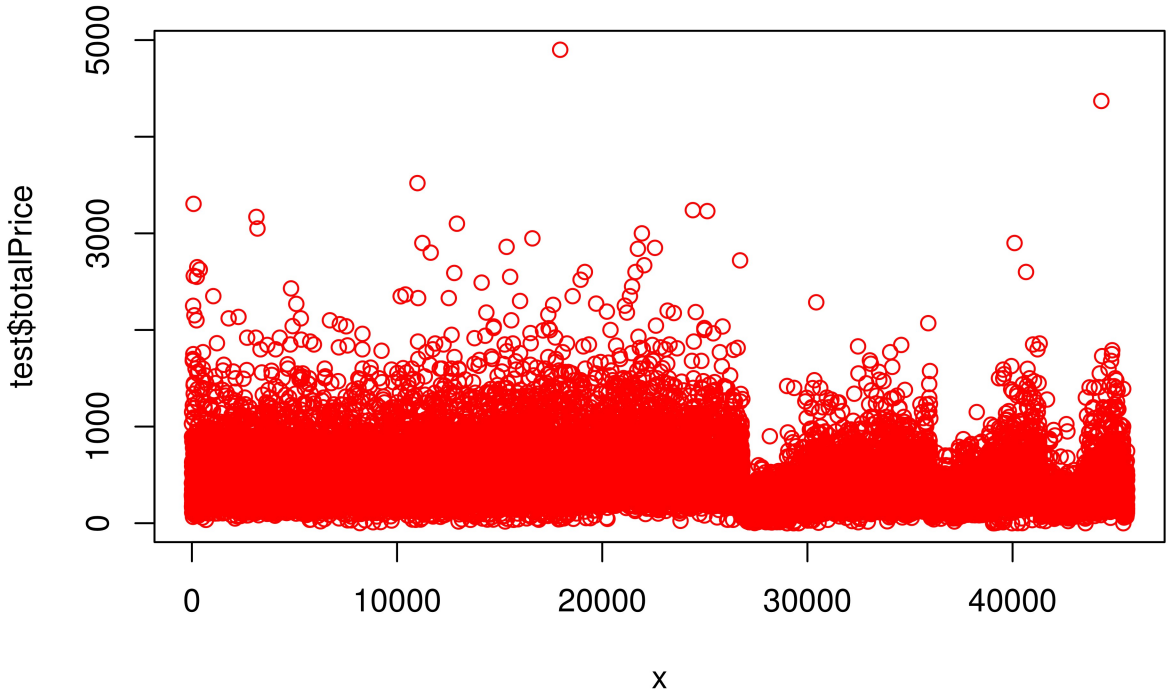
```
## [1] 25094.45
```

```
predictions = matrix(predict(linear.model.new,test))
MSE = mean((predictions-test$totalPrice)^2, na.rm = T)
MSE
```

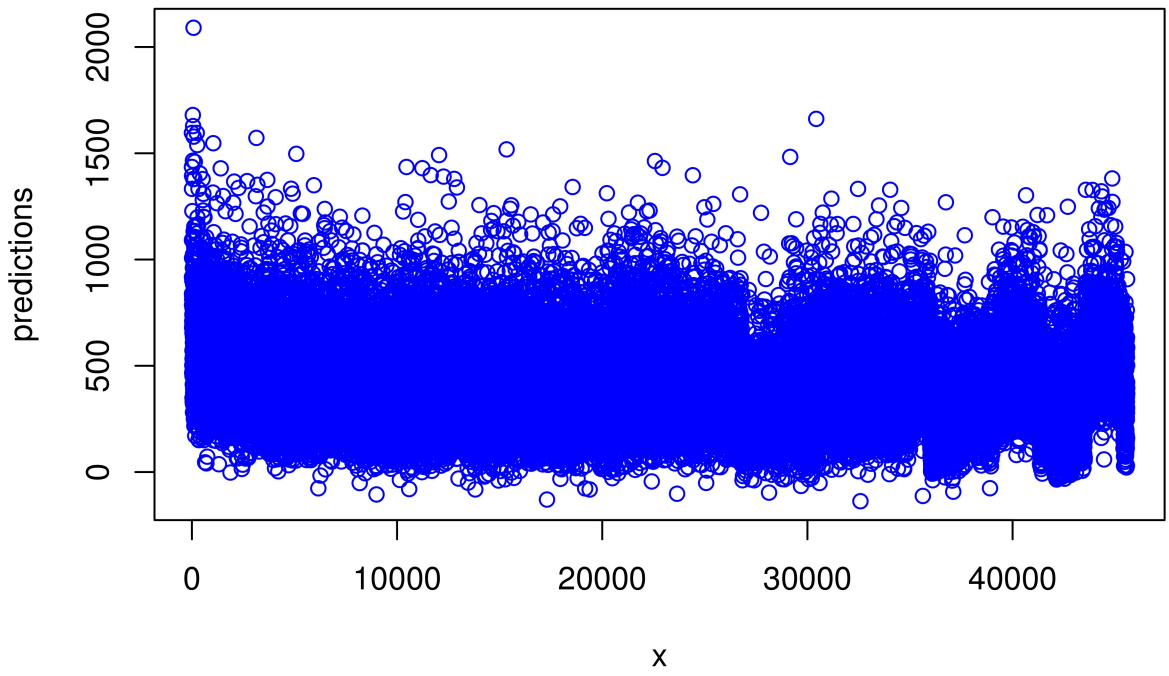
```
## [1] 24533.48
```

This new model gives about the same mse.

```
x = 1:length(test$totalPrice)
plot(x, test$totalPrice, col = "red")
```




```
plot(x, predictions, col = "blue")
```



Building the rig regression model:

```
x_train = as.matrix(train[,-5])

y_train = train$totalPrice


x_test = as.matrix(test[,-5])

y_test = test$totalPrice


lambdas <- seq(0.001, 0.009, by = 0.0001)
```

```
cv_ridge <- cv.glmnet(x_train, y_train, alpha = 0, nfolds = 10,lambda = lambdas)

optimal_lambda <- cv_ridge$lambda.min

optimal_lambda
```

```
## [1] 0.0083
```

```
# Prediction and evaluation on train data

predictions_train <- predict(cv_ridge, s = optimal_lambda, newx = x_train)
```

```
mse_train = mean((y_train-predictions_train)^2)
mse_train
```

```
## [1] 25094.33
```

```
SSE <- sum((predictions_train - y_train)^2)
SST <- sum((y_train - mean(y_train))^2)
R_square <- 1 - (SSE / SST)
R_square
```

```
## [1] 0.6145682
```

```
# Prediction and evaluation on test data

predictions_test <- predict(cv_ridge, s = optimal_lambda, newx = x_test)
mse_test = mean((y_test-predictions_test)^2)
mse_test
```

```
## [1] 24533.24
```

Ridge regression gives about the same error as well as the same R^2 value.

Building the lasso regression model:

```
cv_lasso <- cv.glmnet(x_train, y_train, alpha = 1, nfolds = 10,lambda = lambdas)
optimal_lambda <- cv_lasso$lambda.min

optimal_lambda
```

```
## [1] 0.0085
```

```
# Prediction and evaluation on train data

predictions_train <- predict(cv_lasso, s = optimal_lambda, newx = x_train)

mse_train = mean((y_train-predictions_train)^2)
mse_train
```

```
## [1] 25094.33
```

```
SSE <- sum((predictions_train - y_train)^2)
SST <- sum((y_train - mean(y_train))^2)
R_square <- 1 - (SSE / SST)
R_square
```

```
## [1] 0.6145682
```

```
# Prediction and evaluation on test data

predictions_test <- predict(cv_lasso, s = optimal_lambda, newx = x_test)
mse_test = mean((y_test-predictions_test)^2)
mse_test
```

```
## [1] 24533.22
```

The error and R^2 with lasso regression is also the same.

Creating the Xgb model:

```
set.seed(7)
xgb_train = xgb.DMatrix(data = x_train, label = y_train)
xgb_test = xgb.DMatrix(data = x_test, label = y_test)

watchlist = list(train=xgb_train, test=xgb_test)
model = xgb.train(data = xgb_train, max.depth = 7, watchlist=watchlist, nrounds = 400, lambda = 3)
```

```
## [1]  train-rmse:350.140914   test-rmse:346.908592
## [2]  train-rmse:259.493804   test-rmse:256.932859
## [3]  train-rmse:199.163479   test-rmse:197.442558
## [4]  train-rmse:159.898596   test-rmse:159.354452
## [5]  train-rmse:135.204654   test-rmse:135.763134
## [6]  train-rmse:120.150253   test-rmse:121.750998
## [7]  train-rmse:111.181324   test-rmse:113.901024
## [8]  train-rmse:105.527687   test-rmse:109.189216
## [9]  train-rmse:101.902605   test-rmse:106.499547
## [10] train-rmse:99.668896    test-rmse:105.013779
## [11] train-rmse:97.765162    test-rmse:103.654499
## [12] train-rmse:96.456580    test-rmse:102.742249
## [13] train-rmse:95.439431    test-rmse:102.015086
## [14] train-rmse:94.686742    test-rmse:101.555385
## [15] train-rmse:94.259820    test-rmse:101.335602
## [16] train-rmse:93.789188    test-rmse:101.094436
## [17] train-rmse:93.083266    test-rmse:100.640058
## [18] train-rmse:92.621592    test-rmse:100.444706
## [19] train-rmse:92.032768    test-rmse:100.113883
## [20] train-rmse:91.363942    test-rmse:99.897626
## [21] train-rmse:90.879895    test-rmse:99.522385
## [22] train-rmse:90.761030    test-rmse:99.427539
## [23] train-rmse:90.512395    test-rmse:99.219076
## [24] train-rmse:90.165929    test-rmse:99.108259
## [25] train-rmse:89.840635    test-rmse:98.886807
## [26] train-rmse:89.677423    test-rmse:98.804994
## [27] train-rmse:89.134291    test-rmse:98.447912
## [28] train-rmse:88.776721    test-rmse:98.249028
## [29] train-rmse:88.455978    test-rmse:98.207610
## [30] train-rmse:88.253397    test-rmse:98.118581
## [31] train-rmse:87.785091    test-rmse:97.851065
## [32] train-rmse:87.214733    test-rmse:97.565672
## [33] train-rmse:86.821638    test-rmse:97.358224
## [34] train-rmse:86.523959    test-rmse:97.194608
```

```
## [35] train-rmse:86.303777    test-rmse:97.180279
## [36] train-rmse:86.113529    test-rmse:97.064163
## [37] train-rmse:85.899189    test-rmse:96.981470
## [38] train-rmse:85.524495    test-rmse:96.751414
## [39] train-rmse:85.249933    test-rmse:96.548198
## [40] train-rmse:84.974941    test-rmse:96.467673
## [41] train-rmse:84.711370    test-rmse:96.417990
## [42] train-rmse:84.365317    test-rmse:96.241436
## [43] train-rmse:84.146300    test-rmse:96.251882
## [44] train-rmse:83.881095    test-rmse:96.160026
## [45] train-rmse:83.609047    test-rmse:96.100354
## [46] train-rmse:83.515137    test-rmse:96.062282
## [47] train-rmse:83.252925    test-rmse:95.942673
## [48] train-rmse:83.092211    test-rmse:95.860022
## [49] train-rmse:82.873140    test-rmse:95.708301
## [50] train-rmse:82.582212    test-rmse:95.539837
## [51] train-rmse:82.418356    test-rmse:95.467594
## [52] train-rmse:82.155529    test-rmse:95.318417
## [53] train-rmse:82.013817    test-rmse:95.246796
## [54] train-rmse:81.688895    test-rmse:95.134911
## [55] train-rmse:81.409961    test-rmse:95.117188
## [56] train-rmse:81.208487    test-rmse:95.091786
## [57] train-rmse:81.034962    test-rmse:95.032435
## [58] train-rmse:80.884861    test-rmse:95.007925
## [59] train-rmse:80.822052    test-rmse:94.988780
## [60] train-rmse:80.564392    test-rmse:94.821571
## [61] train-rmse:80.370998    test-rmse:94.839372
## [62] train-rmse:80.308914    test-rmse:94.840249
## [63] train-rmse:80.060372    test-rmse:94.745288
## [64] train-rmse:79.927190    test-rmse:94.676487
## [65] train-rmse:79.765575    test-rmse:94.590844
## [66] train-rmse:79.501890    test-rmse:94.519372
## [67] train-rmse:79.320368    test-rmse:94.481234
## [68] train-rmse:79.288777    test-rmse:94.470120
## [69] train-rmse:79.115573    test-rmse:94.387779
## [70] train-rmse:78.999320    test-rmse:94.339492
## [71] train-rmse:78.811703    test-rmse:94.298374
## [72] train-rmse:78.708006    test-rmse:94.237572
## [73] train-rmse:78.607168    test-rmse:94.204795
## [74] train-rmse:78.355470    test-rmse:94.161745
## [75] train-rmse:78.247845    test-rmse:94.103568
## [76] train-rmse:78.082213    test-rmse:94.028744
## [77] train-rmse:77.904817    test-rmse:93.958197
## [78] train-rmse:77.839999    test-rmse:93.939305
## [79] train-rmse:77.733036    test-rmse:93.893638
## [80] train-rmse:77.690008    test-rmse:93.895952
## [81] train-rmse:77.616301    test-rmse:93.873132
## [82] train-rmse:77.485944    test-rmse:93.822334
## [83] train-rmse:77.351067    test-rmse:93.788823
## [84] train-rmse:77.228921    test-rmse:93.711339
## [85] train-rmse:77.075961    test-rmse:93.636548
## [86] train-rmse:76.935435    test-rmse:93.618614
## [87] train-rmse:76.856902    test-rmse:93.606150
## [88] train-rmse:76.704442    test-rmse:93.523623
```

```
## [89] train-rmse:76.507259    test-rmse:93.490387
## [90] train-rmse:76.359936    test-rmse:93.436363
## [91] train-rmse:76.202361    test-rmse:93.395514
## [92] train-rmse:76.148897    test-rmse:93.391955
## [93] train-rmse:75.998543    test-rmse:93.311599
## [94] train-rmse:75.825756    test-rmse:93.293436
## [95] train-rmse:75.583801    test-rmse:93.308045
## [96] train-rmse:75.430961    test-rmse:93.280042
## [97] train-rmse:75.205119    test-rmse:93.253406
## [98] train-rmse:75.057914    test-rmse:93.228784
## [99] train-rmse:75.031132    test-rmse:93.227965
## [100]    train-rmse:74.960843    test-rmse:93.197477
## [101]    train-rmse:74.899956    test-rmse:93.189073
## [102]    train-rmse:74.740485    test-rmse:93.158756
## [103]    train-rmse:74.699939    test-rmse:93.141697
## [104]    train-rmse:74.550463    test-rmse:93.146323
## [105]    train-rmse:74.437868    test-rmse:93.120969
## [106]    train-rmse:74.355986    test-rmse:93.108903
## [107]    train-rmse:74.216937    test-rmse:93.074297
## [108]    train-rmse:74.083086    test-rmse:93.063008
## [109]    train-rmse:74.031093    test-rmse:93.040974
## [110]    train-rmse:73.925893    test-rmse:93.007871
## [111]    train-rmse:73.864044    test-rmse:93.035428
## [112]    train-rmse:73.711207    test-rmse:93.044613
## [113]    train-rmse:73.661212    test-rmse:93.028175
## [114]    train-rmse:73.584701    test-rmse:93.006891
## [115]    train-rmse:73.407001    test-rmse:92.919152
## [116]    train-rmse:73.262209    test-rmse:92.897542
## [117]    train-rmse:73.216926    test-rmse:92.887693
## [118]    train-rmse:73.163927    test-rmse:92.875455
## [119]    train-rmse:73.089982    test-rmse:92.853124
## [120]    train-rmse:73.041134    test-rmse:92.833625
## [121]    train-rmse:72.994957    test-rmse:92.846373
## [122]    train-rmse:72.844091    test-rmse:92.792264
## [123]    train-rmse:72.709308    test-rmse:92.758125
## [124]    train-rmse:72.662978    test-rmse:92.757132
## [125]    train-rmse:72.582239    test-rmse:92.722014
## [126]    train-rmse:72.391583    test-rmse:92.681702
## [127]    train-rmse:72.269200    test-rmse:92.685242
## [128]    train-rmse:72.200475    test-rmse:92.673909
## [129]    train-rmse:72.097280    test-rmse:92.668185
## [130]    train-rmse:71.952873    test-rmse:92.625064
## [131]    train-rmse:71.921142    test-rmse:92.612635
## [132]    train-rmse:71.800148    test-rmse:92.607819
## [133]    train-rmse:71.709340    test-rmse:92.580356
## [134]    train-rmse:71.568264    test-rmse:92.563911
## [135]    train-rmse:71.446355    test-rmse:92.545259
## [136]    train-rmse:71.354896    test-rmse:92.542191
## [137]    train-rmse:71.307352    test-rmse:92.519369
## [138]    train-rmse:71.238071    test-rmse:92.499369
## [139]    train-rmse:71.150735    test-rmse:92.513500
## [140]    train-rmse:71.067189    test-rmse:92.509066
## [141]    train-rmse:71.006942    test-rmse:92.470358
## [142]    train-rmse:70.887122    test-rmse:92.401871
```

```
## [143]    train-rmse:70.830733    test-rmse:92.393424
## [144]    train-rmse:70.736263    test-rmse:92.383417
## [145]    train-rmse:70.647308    test-rmse:92.397913
## [146]    train-rmse:70.590839    test-rmse:92.400002
## [147]    train-rmse:70.521400    test-rmse:92.371227
## [148]    train-rmse:70.425453    test-rmse:92.341259
## [149]    train-rmse:70.392408    test-rmse:92.348500
## [150]    train-rmse:70.344601    test-rmse:92.343531
## [151]    train-rmse:70.316777    test-rmse:92.347949
## [152]    train-rmse:70.147775    test-rmse:92.356068
## [153]    train-rmse:70.113665    test-rmse:92.349902
## [154]    train-rmse:70.005724    test-rmse:92.354771
## [155]    train-rmse:69.852574    test-rmse:92.372036
## [156]    train-rmse:69.784024    test-rmse:92.380085
## [157]    train-rmse:69.729510    test-rmse:92.360114
## [158]    train-rmse:69.629517    test-rmse:92.334201
## [159]    train-rmse:69.512724    test-rmse:92.319195
## [160]    train-rmse:69.462585    test-rmse:92.308731
## [161]    train-rmse:69.427656    test-rmse:92.301953
## [162]    train-rmse:69.363820    test-rmse:92.284158
## [163]    train-rmse:69.300527    test-rmse:92.291567
## [164]    train-rmse:69.220412    test-rmse:92.301529
## [165]    train-rmse:69.180362    test-rmse:92.293140
## [166]    train-rmse:69.017859    test-rmse:92.264001
## [167]    train-rmse:68.953462    test-rmse:92.251531
## [168]    train-rmse:68.893551    test-rmse:92.249520
## [169]    train-rmse:68.826739    test-rmse:92.223779
## [170]    train-rmse:68.743483    test-rmse:92.209093
## [171]    train-rmse:68.677486    test-rmse:92.206996
## [172]    train-rmse:68.544140    test-rmse:92.186729
## [173]    train-rmse:68.493034    test-rmse:92.168081
## [174]    train-rmse:68.433108    test-rmse:92.170767
## [175]    train-rmse:68.363222    test-rmse:92.164499
## [176]    train-rmse:68.289953    test-rmse:92.138663
## [177]    train-rmse:68.258351    test-rmse:92.132680
## [178]    train-rmse:68.152467    test-rmse:92.116435
## [179]    train-rmse:68.051497    test-rmse:92.114525
## [180]    train-rmse:67.945763    test-rmse:92.092353
## [181]    train-rmse:67.889644    test-rmse:92.086394
## [182]    train-rmse:67.849536    test-rmse:92.071191
## [183]    train-rmse:67.740515    test-rmse:92.048972
## [184]    train-rmse:67.635757    test-rmse:92.044500
## [185]    train-rmse:67.515827    test-rmse:92.006467
## [186]    train-rmse:67.431718    test-rmse:92.010381
## [187]    train-rmse:67.411611    test-rmse:92.004981
## [188]    train-rmse:67.308912    test-rmse:91.994089
## [189]    train-rmse:67.232700    test-rmse:91.970739
## [190]    train-rmse:67.173437    test-rmse:91.975150
## [191]    train-rmse:67.112911    test-rmse:91.973380
## [192]    train-rmse:67.082665    test-rmse:91.969704
## [193]    train-rmse:66.999910    test-rmse:91.968008
## [194]    train-rmse:66.970827    test-rmse:91.963753
## [195]    train-rmse:66.901979    test-rmse:91.980743
## [196]    train-rmse:66.824803    test-rmse:91.967031
```

```
## [197]    train-rmse:66.780903    test-rmse:91.952830
## [198]    train-rmse:66.762174    test-rmse:91.948042
## [199]    train-rmse:66.683478    test-rmse:91.948666
## [200]    train-rmse:66.619670    test-rmse:91.959912
## [201]    train-rmse:66.573265    test-rmse:91.954376
## [202]    train-rmse:66.531070    test-rmse:91.940767
## [203]    train-rmse:66.472679    test-rmse:91.944922
## [204]    train-rmse:66.411866    test-rmse:91.946456
## [205]    train-rmse:66.352505    test-rmse:91.947581
## [206]    train-rmse:66.263495    test-rmse:91.942872
## [207]    train-rmse:66.189700    test-rmse:91.933611
## [208]    train-rmse:66.138906    test-rmse:91.929359
## [209]    train-rmse:66.102339    test-rmse:91.929687
## [210]    train-rmse:66.051734    test-rmse:91.948391
## [211]    train-rmse:65.983679    test-rmse:91.966183
## [212]    train-rmse:65.946779    test-rmse:91.959390
## [213]    train-rmse:65.863989    test-rmse:91.943963
## [214]    train-rmse:65.827578    test-rmse:91.938046
## [215]    train-rmse:65.755960    test-rmse:91.931602
## [216]    train-rmse:65.647554    test-rmse:91.907263
## [217]    train-rmse:65.589770    test-rmse:91.873054
## [218]    train-rmse:65.539052    test-rmse:91.869762
## [219]    train-rmse:65.444211    test-rmse:91.847667
## [220]    train-rmse:65.400110    test-rmse:91.847899
## [221]    train-rmse:65.323168    test-rmse:91.845545
## [222]    train-rmse:65.248596    test-rmse:91.843809
## [223]    train-rmse:65.182953    test-rmse:91.819508
## [224]    train-rmse:65.129125    test-rmse:91.832659
## [225]    train-rmse:65.098916    test-rmse:91.834880
## [226]    train-rmse:65.032658    test-rmse:91.830213
## [227]    train-rmse:65.001383    test-rmse:91.819026
## [228]    train-rmse:64.902684    test-rmse:91.800700
## [229]    train-rmse:64.864741    test-rmse:91.777795
## [230]    train-rmse:64.845696    test-rmse:91.780147
## [231]    train-rmse:64.709986    test-rmse:91.729624
## [232]    train-rmse:64.642582    test-rmse:91.707679
## [233]    train-rmse:64.587187    test-rmse:91.676584
## [234]    train-rmse:64.552054    test-rmse:91.667813
## [235]    train-rmse:64.477169    test-rmse:91.650588
## [236]    train-rmse:64.437828    test-rmse:91.644909
## [237]    train-rmse:64.400257    test-rmse:91.640061
## [238]    train-rmse:64.340255    test-rmse:91.639969
## [239]    train-rmse:64.317022    test-rmse:91.647129
## [240]    train-rmse:64.282567    test-rmse:91.652235
## [241]    train-rmse:64.249496    test-rmse:91.650611
## [242]    train-rmse:64.137109    test-rmse:91.674076
## [243]    train-rmse:64.027718    test-rmse:91.709907
## [244]    train-rmse:64.019724    test-rmse:91.705580
## [245]    train-rmse:64.004693    test-rmse:91.709087
## [246]    train-rmse:63.888252    test-rmse:91.676896
## [247]    train-rmse:63.803504    test-rmse:91.656942
## [248]    train-rmse:63.741529    test-rmse:91.632851
## [249]    train-rmse:63.674160    test-rmse:91.638041
## [250]    train-rmse:63.604775    test-rmse:91.641581
```

```
## [251]    train-rmse:63.553956    test-rmse:91.632357
## [252]    train-rmse:63.486745    test-rmse:91.613430
## [253]    train-rmse:63.443614    test-rmse:91.601212
## [254]    train-rmse:63.425149    test-rmse:91.597360
## [255]    train-rmse:63.415602    test-rmse:91.591143
## [256]    train-rmse:63.331224    test-rmse:91.572366
## [257]    train-rmse:63.302782    test-rmse:91.565684
## [258]    train-rmse:63.279062    test-rmse:91.571115
## [259]    train-rmse:63.245345    test-rmse:91.584715
## [260]    train-rmse:63.122009    test-rmse:91.591840
## [261]    train-rmse:63.060993    test-rmse:91.593016
## [262]    train-rmse:62.968889    test-rmse:91.573578
## [263]    train-rmse:62.950609    test-rmse:91.571943
## [264]    train-rmse:62.925657    test-rmse:91.563345
## [265]    train-rmse:62.890161    test-rmse:91.547706
## [266]    train-rmse:62.826924    test-rmse:91.548778
## [267]    train-rmse:62.731570    test-rmse:91.559904
## [268]    train-rmse:62.682523    test-rmse:91.555728
## [269]    train-rmse:62.609948    test-rmse:91.568409
## [270]    train-rmse:62.563569    test-rmse:91.573130
## [271]    train-rmse:62.469234    test-rmse:91.574977
## [272]    train-rmse:62.446227    test-rmse:91.575065
## [273]    train-rmse:62.342831    test-rmse:91.563724
## [274]    train-rmse:62.322414    test-rmse:91.554694
## [275]    train-rmse:62.291538    test-rmse:91.558212
## [276]    train-rmse:62.249295    test-rmse:91.565464
## [277]    train-rmse:62.178338    test-rmse:91.557394
## [278]    train-rmse:62.146876    test-rmse:91.553269
## [279]    train-rmse:62.113958    test-rmse:91.556027
## [280]    train-rmse:62.067459    test-rmse:91.554347
## [281]    train-rmse:61.970658    test-rmse:91.538938
## [282]    train-rmse:61.934781    test-rmse:91.536733
## [283]    train-rmse:61.885719    test-rmse:91.548386
## [284]    train-rmse:61.857709    test-rmse:91.551502
## [285]    train-rmse:61.812982    test-rmse:91.535088
## [286]    train-rmse:61.786763    test-rmse:91.538414
## [287]    train-rmse:61.777696    test-rmse:91.537086
## [288]    train-rmse:61.723774    test-rmse:91.538373
## [289]    train-rmse:61.688894    test-rmse:91.535001
## [290]    train-rmse:61.608904    test-rmse:91.506558
## [291]    train-rmse:61.545163    test-rmse:91.530143
## [292]    train-rmse:61.499225    test-rmse:91.527078
## [293]    train-rmse:61.469522    test-rmse:91.523223
## [294]    train-rmse:61.437190    test-rmse:91.516945
## [295]    train-rmse:61.363845    test-rmse:91.520603
## [296]    train-rmse:61.275714    test-rmse:91.520505
## [297]    train-rmse:61.214755    test-rmse:91.516856
## [298]    train-rmse:61.159570    test-rmse:91.501439
## [299]    train-rmse:61.112387    test-rmse:91.496916
## [300]    train-rmse:61.081782    test-rmse:91.496543
## [301]    train-rmse:61.033074    test-rmse:91.498698
## [302]    train-rmse:61.019103    test-rmse:91.500933
## [303]    train-rmse:60.920131    test-rmse:91.434717
## [304]    train-rmse:60.896393    test-rmse:91.439765
```

```
## [305]    train-rmse:60.844622    test-rmse:91.430016
## [306]    train-rmse:60.810375    test-rmse:91.425075
## [307]    train-rmse:60.774740    test-rmse:91.419681
## [308]    train-rmse:60.702379    test-rmse:91.423500
## [309]    train-rmse:60.616904    test-rmse:91.428378
## [310]    train-rmse:60.553394    test-rmse:91.407064
## [311]    train-rmse:60.485924    test-rmse:91.407650
## [312]    train-rmse:60.417280    test-rmse:91.401499
## [313]    train-rmse:60.380299    test-rmse:91.394038
## [314]    train-rmse:60.348561    test-rmse:91.392291
## [315]    train-rmse:60.287464    test-rmse:91.387944
## [316]    train-rmse:60.249094    test-rmse:91.399454
## [317]    train-rmse:60.223531    test-rmse:91.383942
## [318]    train-rmse:60.191251    test-rmse:91.394055
## [319]    train-rmse:60.133892    test-rmse:91.406497
## [320]    train-rmse:60.120715    test-rmse:91.406244
## [321]    train-rmse:60.109156    test-rmse:91.413710
## [322]    train-rmse:60.049434    test-rmse:91.407047
## [323]    train-rmse:60.027967    test-rmse:91.406230
## [324]    train-rmse:59.992484    test-rmse:91.389543
## [325]    train-rmse:59.915982    test-rmse:91.373342
## [326]    train-rmse:59.850476    test-rmse:91.363188
## [327]    train-rmse:59.799071    test-rmse:91.358569
## [328]    train-rmse:59.738549    test-rmse:91.371088
## [329]    train-rmse:59.661357    test-rmse:91.389683
## [330]    train-rmse:59.622185    test-rmse:91.393791
## [331]    train-rmse:59.611486    test-rmse:91.395137
## [332]    train-rmse:59.558460    test-rmse:91.405936
## [333]    train-rmse:59.518738    test-rmse:91.397397
## [334]    train-rmse:59.482809    test-rmse:91.402197
## [335]    train-rmse:59.449630    test-rmse:91.404656
## [336]    train-rmse:59.375365    test-rmse:91.390014
## [337]    train-rmse:59.346948    test-rmse:91.386891
## [338]    train-rmse:59.288331    test-rmse:91.372444
## [339]    train-rmse:59.260794    test-rmse:91.370711
## [340]    train-rmse:59.210893    test-rmse:91.375175
## [341]    train-rmse:59.189329    test-rmse:91.373132
## [342]    train-rmse:59.135351    test-rmse:91.374952
## [343]    train-rmse:59.098746    test-rmse:91.360553
## [344]    train-rmse:59.019094    test-rmse:91.351501
## [345]    train-rmse:58.953988    test-rmse:91.365434
## [346]    train-rmse:58.918501    test-rmse:91.365692
## [347]    train-rmse:58.884252    test-rmse:91.367347
## [348]    train-rmse:58.843253    test-rmse:91.372781
## [349]    train-rmse:58.796771    test-rmse:91.381785
## [350]    train-rmse:58.770050    test-rmse:91.382102
## [351]    train-rmse:58.736226    test-rmse:91.378357
## [352]    train-rmse:58.694186    test-rmse:91.388217
## [353]    train-rmse:58.642106    test-rmse:91.382339
## [354]    train-rmse:58.589262    test-rmse:91.381225
## [355]    train-rmse:58.560402    test-rmse:91.381918
## [356]    train-rmse:58.540076    test-rmse:91.382909
## [357]    train-rmse:58.490219    test-rmse:91.396399
## [358]    train-rmse:58.457764    test-rmse:91.389421
```

```
## [359]    train-rmse:58.382063    test-rmse:91.412980
## [360]    train-rmse:58.350132    test-rmse:91.425145
## [361]    train-rmse:58.319665    test-rmse:91.421483
## [362]    train-rmse:58.284348    test-rmse:91.411937
## [363]    train-rmse:58.247088    test-rmse:91.422899
## [364]    train-rmse:58.195641    test-rmse:91.431868
## [365]    train-rmse:58.158278    test-rmse:91.426485
## [366]    train-rmse:58.063573    test-rmse:91.427626
## [367]    train-rmse:58.050883    test-rmse:91.425013
## [368]    train-rmse:58.001992    test-rmse:91.430730
## [369]    train-rmse:57.920211    test-rmse:91.439622
## [370]    train-rmse:57.898905    test-rmse:91.439155
## [371]    train-rmse:57.824150    test-rmse:91.442733
## [372]    train-rmse:57.757074    test-rmse:91.448627
## [373]    train-rmse:57.689839    test-rmse:91.447157
## [374]    train-rmse:57.667735    test-rmse:91.440195
## [375]    train-rmse:57.646050    test-rmse:91.442010
## [376]    train-rmse:57.606105    test-rmse:91.440518
## [377]    train-rmse:57.595232    test-rmse:91.447833
## [378]    train-rmse:57.532886    test-rmse:91.461415
## [379]    train-rmse:57.494431    test-rmse:91.466244
## [380]    train-rmse:57.470055    test-rmse:91.459984
## [381]    train-rmse:57.460552    test-rmse:91.457601
## [382]    train-rmse:57.436341    test-rmse:91.460520
## [383]    train-rmse:57.411485    test-rmse:91.467583
## [384]    train-rmse:57.364390    test-rmse:91.462109
## [385]    train-rmse:57.297830    test-rmse:91.464338
## [386]    train-rmse:57.277476    test-rmse:91.460578
## [387]    train-rmse:57.245777    test-rmse:91.452693
## [388]    train-rmse:57.219917    test-rmse:91.454282
## [389]    train-rmse:57.179295    test-rmse:91.459750
## [390]    train-rmse:57.161114    test-rmse:91.460218
## [391]    train-rmse:57.130666    test-rmse:91.469132
## [392]    train-rmse:57.090035    test-rmse:91.473285
## [393]    train-rmse:57.065221    test-rmse:91.471109
## [394]    train-rmse:57.048740    test-rmse:91.473209
## [395]    train-rmse:57.016466    test-rmse:91.465610
## [396]    train-rmse:56.968731    test-rmse:91.478077
## [397]    train-rmse:56.884888    test-rmse:91.461528
## [398]    train-rmse:56.850521    test-rmse:91.455905
## [399]    train-rmse:56.817204    test-rmse:91.456202
## [400]    train-rmse:56.790765    test-rmse:91.455443
```

```
which.min(model$evaluation_log$test_rmse)
```

```
## [1] 344
```

```
min(model$evaluation_log$train_rmse)
```

```
## [1] 56.79077
```

395th round is the best round.

```
model_xgboost = xgboost(data = xgb_train, max.depth = 7, nrounds = 395, verbose = 0, lambda = 3)

summary(model_xgboost)
```

```
##                Length  Class              Mode
## handle              1 xgb.Booster.handle externalptr
## raw           2935003 -none-             raw
## niter               1 -none-             numeric
## evaluation_log      2 data.table         list
## call               15 -none-             call
## params              3 -none-             list
## callbacks           1 -none-             list
## feature_names      20 -none-             character
## nfeatures           1 -none-             numeric
```

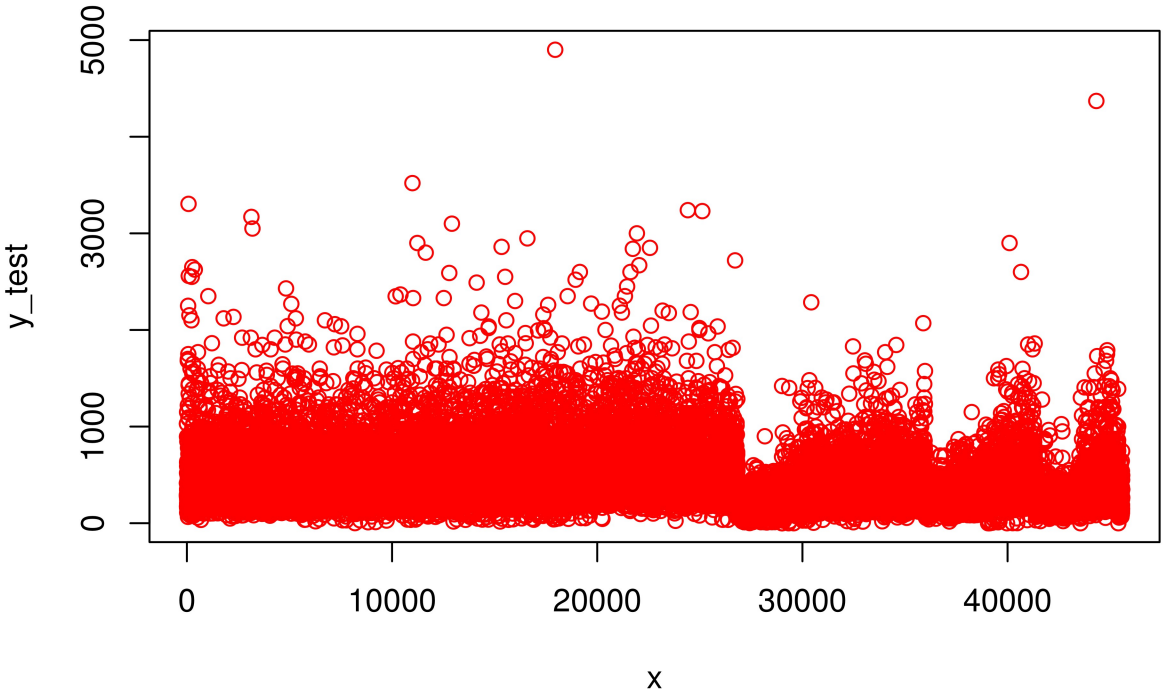
```
pred_y = predict(model_xgboost, xgb_test)
mean((y_test - pred_y)^2)
```

```
## [1] 8365.958
```

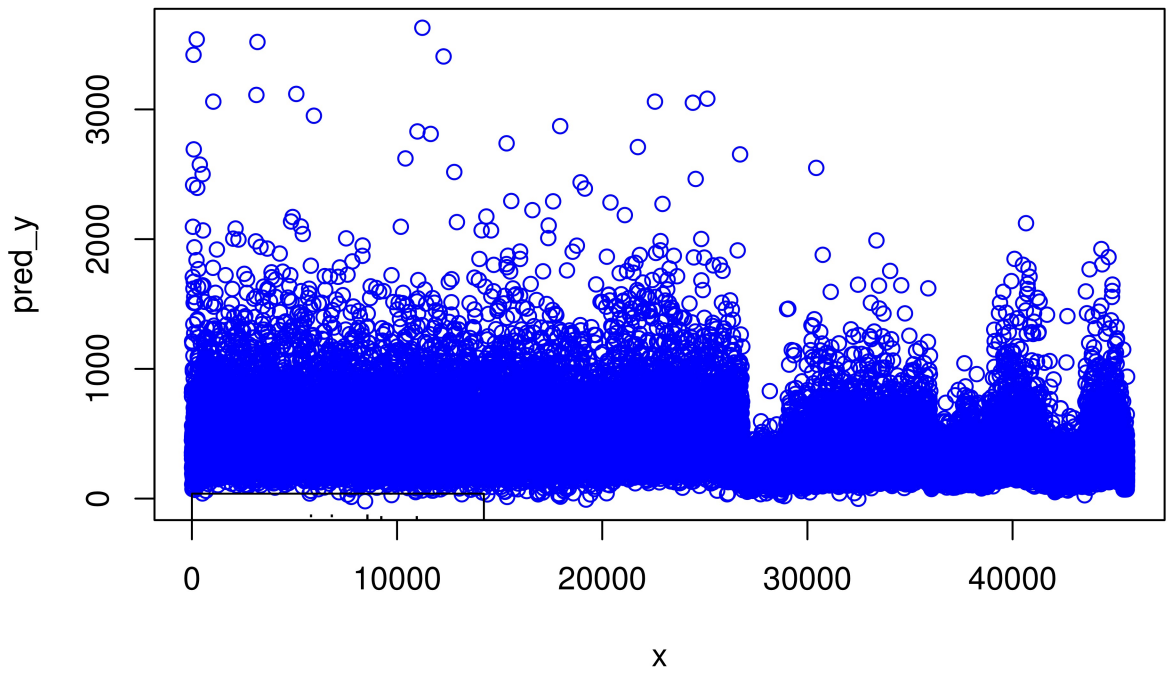
```
residuals = y_test - pred_y
y_test_mean = mean(y_test)
TSS = sum((y_test - y_test_mean)^2)
RSS = sum(residuals^2)
R_squared = 1 - (RSS/TSS)
R_squared
```

```
## [1] 0.8662235
```

```
x = 1:length(y_test)
plot(x, y_test, col = "red")
```




```
plot(x, pred_y, col = "blue")
legend(x = 1, y = 38,  legend = c("original test_y", "predicted test_y"),
       col = c("red", "blue"), box.lty = 1, cex = 0.8, lty = c(1, 1))
```




```
set.seed(7)

knnmodel = knnreg(train[,-5], train$totalPrice, k=8)
pred_y = predict(knnmodel, test[,-5])
error = mean((pred_y - test$totalPrice)^2)
error
```

```
## [1] 19209.56
```

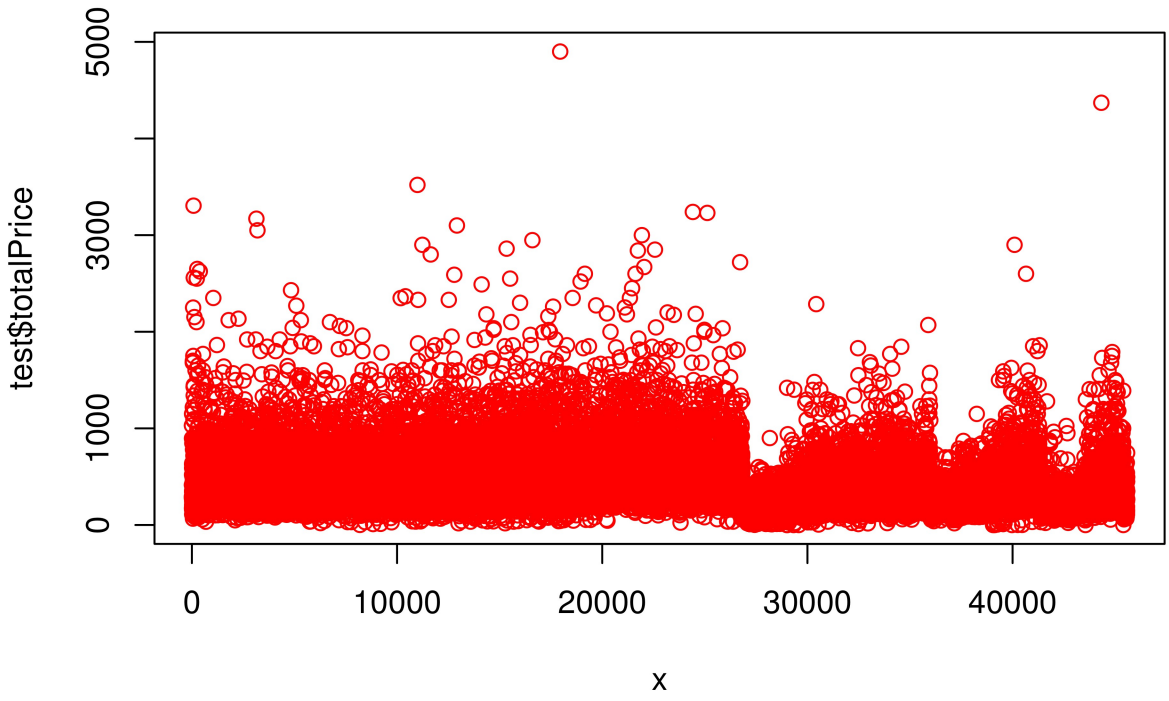
```
pred = predict(knnmodel, train[,-5])
error_train = mean((pred - train$totalPrice)^2)
error_train
```

```
## [1] 15211.36
```

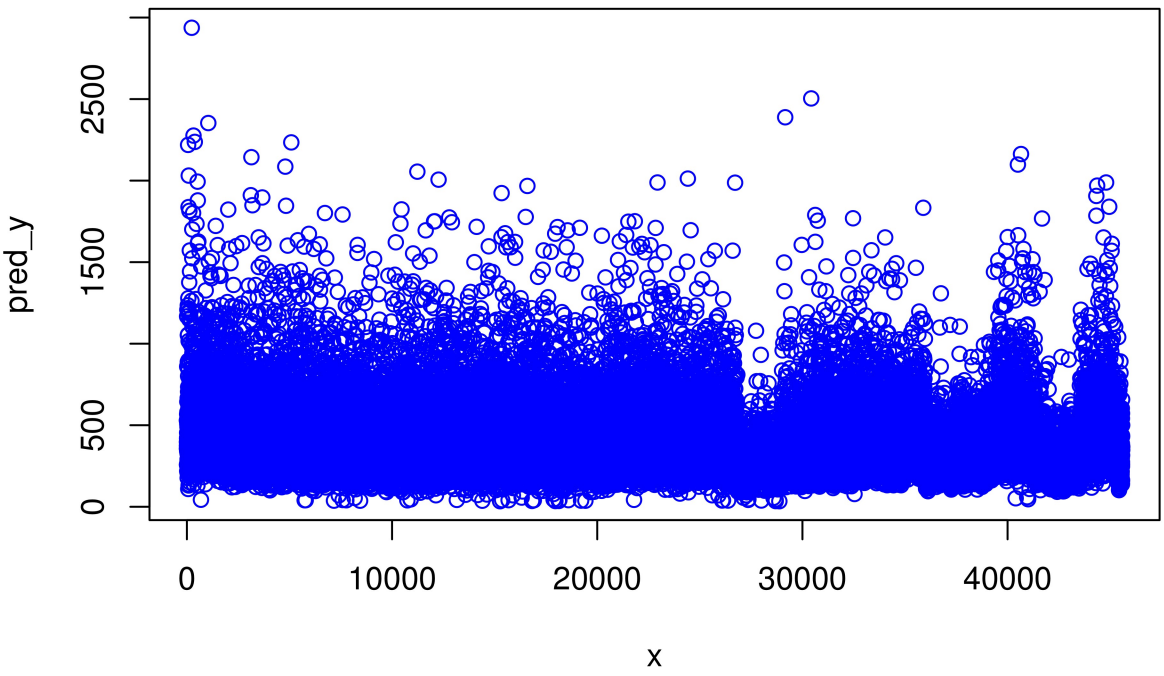
```
rss <- sum((pred_y - test$totalPrice) ^ 2)
tss <- sum((test$totalPrice - mean(test$totalPrice)) ^ 2)
r_square = 1 - (rss/tss)
r_square
```

```
## [1] 0.6928279
```

```
x = 1:length(test$totalPrice)
plot(x, test$totalPrice, col = "red")
```




```
plot(x, pred_y, col = "blue")
```


pred_y
0
500
1500
2500
0
10000
20000
30000
40000
x