Final Project for Math 342W Data Science at Queens College

May 25,2021

By: Shoara Chowdhury

In collaboration with: Nasif Khan

**Abstract:** The ideal purpose of this paper is to build a predictive model for the sales price apartment in Queens, New York. Machine learning is where it comes in play, that’s where the beauty begins, and a way of we can use in everyday life anywhere you go. Using the data, on the Queens housing market, we used machine learning algorithms for the sales prediction. Those features in the dataset have been classified that are Random Forest, Linear Regression and Regression tree model algorithm are used to predict the selling price of the houses. The houses that are selling prices shows the regression of the, so it advises the algorithm to handle both regression and, classification problem. The dataset in this case study is taken from MLSI. Furthermore, in the dataset, is in the raw format and needs too preprocessing. The supervised machine learning has shown us a significant result of R2 and MSE. Random forest and decision fit is suitable over the dataset and didn’t show any overfitting neither underfitting.

**Introduction:** The analysis is very important and necessary for the housing sale data. These model later can be used to predict the outcome in advance for the future data. After dataset finished cleaning processes, models can be created. Among all Random Forest appears to be the accurate term by accessing accurate and stable prediction.

**2. Data:**

**2.1. Description:**

Dataset is extracted from MLSL website, which can be access and update the raw information of the sales price based on the variable. The dataset contains 57 columns with 2230 instances. This dataset contains a lot of features that are not relevant, and the dataset contains a lot of missing values. Each attribute contains different types of invalid values. Data cleaning will process to perform many features especially to sale\_price variable. It will be the dependent variable of this case study.

**2.2. Featurization**

This is very important step to perform on the attribute before passing to machine learning model. The common mistake people make is making the same attributes of the data. They converted the attributes that does not remain the same. However, there are many features that are removed from the dataset which is not relevant to the model.

**2.3. Errors and Missingness**

In this dataset, which contains the errors, misspell or missing information. The first method we used is the one who has small missing values, and we remove them on the record and their attributes. Then the second method we used, that contains has large missing values. Then we use the mean method of the second method. To verify, we check again of the dataset, if it’s placed in the correct dataset.

**3. Modeling:**

In this model, there are three algorithm we used in this project. Regression tree modeling, Linear Regression model, and Random Forest. Random forest used for classification and regression based on a forest of trees using random inputs. They are caret for data splitting and generating ranked bootstrap samples. Regression tree modeling also have ability to the classification and regression problems. Regression tree modeling perform well for sale prices of the houses. Used the Regression tree modeling technique for exploring we use the relationship between house prices and housing characteristics.

**Regression Tree model:**

Regression tree is a tree building technique which divide into smaller, subgroups datasets. Its divided into, sale\_price, num\_total\_rooms, num\_full\_bathrooms, num\_Oloors\_in\_building, num\_bedrooms. The variable is plotted as shown in which features are important for predicting the model.

**Linear Regression Model:**

Linear regression model has the knowledge to solve the regression problem. It’s very important to solve a regression task. In my model linear regression model is training on the following features kitchen\_type, num\_Oloors\_in\_building, num\_bedrooms, coop\_condo, total\_taxes.

**Random Forest Model:**

Random forest is a combination of tree predictors. It can handle regression and classification tasks efficiently. The model shows it is very fitted to the model, over the traning data, and test data. This model does not have overfitting, and unfitting.

**Conclusion/Discussion:**

From my understanding, we got to handle raw dataset, how to handles the attributes of this model. We also got and more to learn on machine learning are how to be trained on regression problem. Our algorithm outperformed on such raw dataset and get insight from the dataset. Random Forest is better to perform than any tree decisions for prediction of sale prices. At some point I was very lost in this class, I had to re watch lectures, thankfully I have saved recording of the lecture, and I want to give some credit to my friend that helped me correct my mistakes and learn through machine learning.

Project Final

setwd("C:/Users/Megazone/Downloads")  
  
houseing\_data<- read.csv("housing\_data\_2016\_2017.csv")  
str(houseing\_data)

## 'data.frame': 2230 obs. of 57 variables:  
## $ HITId : Factor w/ 1472 levels "301KG0KX9CLCDSCMMO1Y70TF2R6H21",..: 969 900 540 20 619 904 1450 887 1089 57 ...  
## $ HITTypeId : Factor w/ 2 levels "310F0WGLWJ9S9EY8QG59KVO3KZZTLH",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ Title : Factor w/ 1 level "Find Information about Housing To Help a Student Project -- Very easy": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Description : Factor w/ 2 levels "Go to a link and copy information into the HIT",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ Keywords : logi NA NA NA NA NA NA ...  
## $ Reward : Factor w/ 1 level "$0.05 ": 1 1 1 1 1 1 1 1 1 1 ...  
## $ CreationTime : Factor w/ 62 levels "Thu Feb 09 11:58:04 PST 2017",..: 48 48 52 44 49 50 53 48 62 44 ...  
## $ MaxAssignments : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ RequesterAnnotation : Factor w/ 2 levels "BatchId:2682081;OriginalHitTemplateId:920937336;",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ AssignmentDurationInSeconds : int 900 900 900 900 900 900 900 900 900 900 ...  
## $ AutoApprovalDelayInSeconds : int 60 60 60 60 60 60 60 60 60 60 ...  
## $ Expiration : Factor w/ 62 levels "Thu Feb 16 11:58:04 PST 2017",..: 48 48 52 44 49 50 53 48 62 44 ...  
## $ NumberOfSimilarHITs : logi NA NA NA NA NA NA ...  
## $ LifetimeInSeconds : logi NA NA NA NA NA NA ...  
## $ AssignmentId : Factor w/ 1472 levels "3018Q3ZVOIQHU9JNLAB9F6A4BNNRAQ",..: 88 225 577 1133 732 482 330 468 231 411 ...  
## $ WorkerId : Factor w/ 73 levels "A116EAQ6S8M9PL",..: 21 44 21 64 21 27 21 57 17 12 ...  
## $ AssignmentStatus : Factor w/ 1 level "Approved": 1 1 1 1 1 1 1 1 1 1 ...  
## $ AcceptTime : Factor w/ 1457 levels "Fri Feb 10 00:00:40 PST 2017",..: 1239 1431 1037 1170 1454 955 1188 1396 1409 1134 ...  
## $ SubmitTime : Factor w/ 1460 levels "Fri Feb 10 00:00:15 PST 2017",..: 1248 1437 1043 1177 1458 962 1194 1402 1417 1141 ...  
## $ AutoApprovalTime : Factor w/ 1460 levels "Fri Feb 10 00:01:15 PST 2017",..: 1249 1438 1044 1178 1459 963 1195 1403 1418 1142 ...  
## $ ApprovalTime : Factor w/ 929 levels "2017-02-09 20:12:11 UTC",..: 828 595 713 794 616 637 804 916 922 778 ...  
## $ RejectionTime : logi NA NA NA NA NA NA ...  
## $ RequesterFeedback : logi NA NA NA NA NA NA ...  
## $ WorkTimeInSeconds : int 181 121 120 160 136 249 85 132 198 130 ...  
## $ LifetimeApprovalRate : Factor w/ 32 levels "100% (1/1)","100% (10/10)",..: 13 28 13 4 13 10 13 27 8 29 ...  
## $ Last30DaysApprovalRate : Factor w/ 32 levels "100% (1/1)","100% (10/10)",..: 13 28 13 4 13 10 13 27 8 29 ...  
## $ Last7DaysApprovalRate : Factor w/ 32 levels "100% (1/1)","100% (10/10)",..: 12 28 12 3 12 9 12 26 19 29 ...  
## $ URL : Factor w/ 1450 levels "http://www.mlsli.com/homes-for-sale/1-Bay-Club-Dr-Bayside-NY-11360-195059673",..: 1413 689 12 263 599 141 474 1150 501 718 ...  
## $ approx\_year\_built : int 1955 1955 2004 2002 1949 1938 1950 1960 1960 2005 ...  
## $ cats\_allowed : Factor w/ 3 levels "no","y","yes": 1 1 1 1 3 3 1 1 1 1 ...  
## $ common\_charges : int 767 0 167 275 0 0 0 0 0 0 ...  
## $ community\_district\_num : int 25 25 24 25 26 28 29 28 25 30 ...  
## $ coop\_condo : Factor w/ 2 levels "co-op","condo": 1 1 2 2 1 1 1 1 1 2 ...  
## $ date\_of\_sale : Factor w/ 222 levels "1/10/2017","1/11/2017",..: 71 71 72 72 73 73 74 74 76 76 ...  
## $ dining\_room\_type : Factor w/ 5 levels "combo","dining area",..: 1 3 1 1 1 1 1 NA NA 5 ...  
## $ dogs\_allowed : Factor w/ 3 levels "no","yes","yes89": 1 1 1 1 2 2 1 1 1 1 ...  
## $ fuel\_type : Factor w/ 6 levels "electric","gas",..: 2 4 NA 2 2 4 2 2 4 NA ...  
## $ full\_address\_or\_zip\_code : Factor w/ 1176 levels " Bayside NY, 11360",..: 1158 562 24 223 497 121 391 941 415 586 ...  
## $ garage\_exists : Factor w/ 6 levels "1","eys","UG",..: NA NA NA NA NA NA NA NA NA NA ...  
## $ kitchen\_type : Factor w/ 4 levels "combo","eat in",..: 2 2 3 2 2 2 3 3 2 2 ...  
## $ maintenance\_cost : int 0 604 0 0 660 932 660 514 781 0 ...  
## $ model\_type : Factor w/ 875 levels "-","\"A\" Unit",..: 651 539 253 56 310 754 448 866 555 350 ...  
## $ num\_bedrooms : int 2 1 1 3 2 2 1 0 1 1 ...  
## $ num\_floors\_in\_building : int 6 7 1 NA 2 6 NA 2 NA 4 ...  
## $ num\_full\_bathrooms : int 1 1 1 2 1 1 1 1 1 1 ...  
## $ num\_half\_bathrooms : int NA NA NA NA NA NA NA NA NA NA ...  
## $ num\_total\_rooms : int 5 4 3 5 4 4 3 2 4 3 ...  
## $ parking\_charges : Factor w/ 90 levels " NA ","100","105",..: 1 1 1 1 1 1 1 1 41 1 ...  
## $ pct\_tax\_deductibl : int NA NA NA NA 39 NA NA NA NA NA ...  
## $ sale\_price : Factor w/ 316 levels " NA ","100000",..: 107 113 33 252 119 126 38 8 94 250 ...  
## $ sq\_footage : int NA 890 550 NA 675 1000 NA 375 NA 681 ...  
## $ total\_taxes : Factor w/ 294 levels " NA ","100","1024",..: 1 1 255 68 1 1 1 1 1 19 ...  
## $ walk\_score : int 82 89 90 94 71 90 72 93 70 98 ...  
## $ listing\_price\_to\_nearest\_1000: int NA NA NA NA NA NA NA NA NA NA ...  
## $ url : Factor w/ 758 levels "http://www.mlsli.com/homes-for-sale/10-01-162nd-St-Beechhurst-NY-11357-194398973",..: NA NA NA NA NA NA NA NA NA NA ...  
## $ lat : num 40.7 40.8 40.7 40.8 40.7 ...  
## $ lon : num -73.8 -73.8 -73.9 -73.8 -73.7 ...

dim(houseing\_data)

## [1] 2230 57

names(houseing\_data)

## [1] "HITId" "HITTypeId"   
## [3] "Title" "Description"   
## [5] "Keywords" "Reward"   
## [7] "CreationTime" "MaxAssignments"   
## [9] "RequesterAnnotation" "AssignmentDurationInSeconds"   
## [11] "AutoApprovalDelayInSeconds" "Expiration"   
## [13] "NumberOfSimilarHITs" "LifetimeInSeconds"   
## [15] "AssignmentId" "WorkerId"   
## [17] "AssignmentStatus" "AcceptTime"   
## [19] "SubmitTime" "AutoApprovalTime"   
## [21] "ApprovalTime" "RejectionTime"   
## [23] "RequesterFeedback" "WorkTimeInSeconds"   
## [25] "LifetimeApprovalRate" "Last30DaysApprovalRate"   
## [27] "Last7DaysApprovalRate" "URL"   
## [29] "approx\_year\_built" "cats\_allowed"   
## [31] "common\_charges" "community\_district\_num"   
## [33] "coop\_condo" "date\_of\_sale"   
## [35] "dining\_room\_type" "dogs\_allowed"   
## [37] "fuel\_type" "full\_address\_or\_zip\_code"   
## [39] "garage\_exists" "kitchen\_type"   
## [41] "maintenance\_cost" "model\_type"   
## [43] "num\_bedrooms" "num\_floors\_in\_building"   
## [45] "num\_full\_bathrooms" "num\_half\_bathrooms"   
## [47] "num\_total\_rooms" "parking\_charges"   
## [49] "pct\_tax\_deductibl" "sale\_price"   
## [51] "sq\_footage" "total\_taxes"   
## [53] "walk\_score" "listing\_price\_to\_nearest\_1000"  
## [55] "url" "lat"   
## [57] "lon"

summary(houseing\_data)

## HITId HITTypeId   
## 301KG0KX9CLCDSCMMO1Y70TF2R6H21: 1 310F0WGLWJ9S9EY8QG59KVO3KZZTLH:944   
## 301KG0KX9CLCDSCMMO1Y70TF2R72HN: 1 36BILMLQB75QQNBTYKGYCZWDN8TVAU:528   
## 301KG0KX9CLCDSCMMO1Y70TF2XO2HG: 1 NA's :758   
## 302OLP89DZ7A1NWADBGC2RFX1NTACS: 1   
## 302OLP89DZ7A1NWADBGC2RFX1NTCAU: 1   
## (Other) :1467   
## NA's : 758   
## Title   
## Find Information about Housing To Help a Student Project -- Very easy:1472   
## NA's : 758   
##   
##   
##   
##   
##   
## Description Keywords   
## Go to a link and copy information into the HIT :528 Mode:logical   
## Got to a link and copy information into the HIT:944 NA's:2230   
## NA's :758   
##   
##   
##   
##   
## Reward CreationTime MaxAssignments  
## $0.05 :1472 Thu Feb 09 11:58:29 PST 2017: 43 Min. :1   
## NA's : 758 Thu Feb 09 11:58:24 PST 2017: 40 1st Qu.:1   
## Wed Feb 15 22:13:50 PST 2017: 39 Median :1   
## Thu Feb 09 11:58:18 PST 2017: 37 Mean :1   
## Thu Feb 09 11:58:07 PST 2017: 36 3rd Qu.:1   
## (Other) :1277 Max. :1   
## NA's : 758 NA's :758   
## RequesterAnnotation  
## BatchId:2682081;OriginalHitTemplateId:920937336;:944   
## BatchId:2689947;OriginalHitTemplateId:920937336;:528   
## NA's :758   
##   
##   
##   
##   
## AssignmentDurationInSeconds AutoApprovalDelayInSeconds  
## Min. :900 Min. :60   
## 1st Qu.:900 1st Qu.:60   
## Median :900 Median :60   
## Mean :900 Mean :60   
## 3rd Qu.:900 3rd Qu.:60   
## Max. :900 Max. :60   
## NA's :758 NA's :758   
## Expiration NumberOfSimilarHITs LifetimeInSeconds  
## Thu Feb 16 11:58:29 PST 2017: 43 Mode:logical Mode:logical   
## Thu Feb 16 11:58:24 PST 2017: 40 NA's:2230 NA's:2230   
## Wed Feb 22 22:13:50 PST 2017: 39   
## Thu Feb 16 11:58:18 PST 2017: 37   
## Thu Feb 16 11:58:07 PST 2017: 36   
## (Other) :1277   
## NA's : 758   
## AssignmentId WorkerId AssignmentStatus  
## 3018Q3ZVOIQHU9JNLAB9F6A4BNNRAQ: 1 A231MNJJDDF3LS:187 Approved:1472   
## 3018Q3ZVOIQHU9JNLAB9F6A4BNSARE: 1 A1SAMLI9AUPHEJ:129 NA's : 758   
## 304SM51WA34YEYOS6DBA0RZ6FW5SB2: 1 A3CA5SLHQTNDHU:124   
## 304SM51WA34YEYOS6DBA0RZ6FWBBSR: 1 AHXBZXWIZJSVG :114   
## 304SM51WA34YEYOS6DBA0RZ6FWISBF: 1 A3B3W8ZOII4D0T:106   
## (Other) :1467 (Other) :812   
## NA's : 758 NA's :758   
## AcceptTime SubmitTime   
## Thu Feb 09 14:09:03 PST 2017: 2 Thu Feb 09 13:43:34 PST 2017: 2   
## Thu Feb 09 14:20:18 PST 2017: 2 Thu Feb 09 13:55:45 PST 2017: 2   
## Thu Feb 09 16:30:51 PST 2017: 2 Thu Feb 09 14:20:17 PST 2017: 2   
## Thu Feb 09 16:45:26 PST 2017: 2 Thu Feb 09 16:53:36 PST 2017: 2   
## Thu Feb 09 17:09:51 PST 2017: 2 Thu Feb 09 17:08:31 PST 2017: 2   
## (Other) :1462 (Other) :1462   
## NA's : 758 NA's : 758   
## AutoApprovalTime ApprovalTime   
## Thu Feb 09 13:44:34 PST 2017: 2 2017-02-10 01:10:11 UTC: 6   
## Thu Feb 09 13:56:45 PST 2017: 2 2017-02-10 01:38:11 UTC: 6   
## Thu Feb 09 14:21:17 PST 2017: 2 2017-02-09 22:12:11 UTC: 5   
## Thu Feb 09 16:54:36 PST 2017: 2 2017-02-10 01:11:11 UTC: 5   
## Thu Feb 09 17:09:31 PST 2017: 2 2017-02-10 01:19:11 UTC: 5   
## (Other) :1462 (Other) :1445   
## NA's : 758 NA's : 758   
## RejectionTime RequesterFeedback WorkTimeInSeconds LifetimeApprovalRate  
## Mode:logical Mode:logical Min. : 22.0 100% (187/187):187   
## NA's:2230 NA's:2230 1st Qu.: 89.0 100% (126/126):126   
## Median :127.0 100% (139/139):124   
## Mean :162.4 100% (116/116):106   
## 3rd Qu.:197.0 100% (115/115):102   
## Max. :815.0 (Other) :827   
## NA's :758 NA's :758   
## Last30DaysApprovalRate Last7DaysApprovalRate  
## 100% (187/187):187 100% (187/187):187   
## 100% (126/126):126 100% (126/126):126   
## 100% (139/139):124 100% (139/139):124   
## 100% (116/116):106 100% (116/116):106   
## 100% (115/115):102 100% (103/103):102   
## (Other) :827 (Other) :827   
## NA's :758 NA's :758   
## URL   
## http://www.mlsli.com/homes-for-sale/102-30-66-Rd-Forest-Hills-NY-11375-192665803 : 2   
## http://www.mlsli.com/homes-for-sale/110-11-Queens-Blvd-Forest-Hills-NY-11375-191269365: 2   
## http://www.mlsli.com/homes-for-sale/135-17-Northern-Blvd-Flushing-NY-11354-196168687 : 2   
## http://www.mlsli.com/homes-for-sale/138-10-Franklin-Ave-Flushing-NY-11355-196309288 : 2   
## http://www.mlsli.com/homes-for-sale/138-70-Elder-Ave-Flushing-NY-11355-186802214 : 2   
## (Other) :1462   
## NA's : 758   
## approx\_year\_built cats\_allowed common\_charges community\_district\_num  
## Min. :1893 no :1402 Min. : 0.0 Min. : 3.00   
## 1st Qu.:1950 y : 2 1st Qu.: 0.0 1st Qu.:25.00   
## Median :1958 yes: 826 Median : 0.0 Median :26.00   
## Mean :1963 Mean : 108.2 Mean :26.33   
## 3rd Qu.:1970 3rd Qu.: 0.0 3rd Qu.:28.00   
## Max. :2017 Max. :2499.0 Max. :32.00   
## NA's :39 NA's :19   
## coop\_condo date\_of\_sale dining\_room\_type dogs\_allowed  
## co-op:1661 6/30/2016 : 7 combo :957 no :1684   
## condo: 569 10/14/2016: 6 dining area: 2 yes : 544   
## 12/27/2016: 6 formal :620 yes89: 2   
## 2/26/2016 : 6 none : 2   
## 8/10/2016 : 6 other :201   
## (Other) : 497 NA's :448   
## NA's :1702   
## fuel\_type full\_address\_or\_zip\_code  
## electric: 62 70-25 Yellowstone Blvd, Forest Hills NY, 11375: 22   
## gas :1348 26910 Grand Central Pky, Floral Park NY, 11005: 17   
## none : 3 27010 Grand Central Pky, Floral Park NY, 11005: 16   
## oil : 664 73-12 35th Ave, Jackson Heights NY, 11372 : 14   
## other : 40 110-11 Queens Blvd, Forest Hills NY, 11375 : 12   
## Other : 1 166-25 Powells Cove Blvd, Beechhurst NY, 11357: 12   
## NA's : 112 (Other) :2137   
## garage\_exists kitchen\_type maintenance\_cost model\_type   
## 1 : 1 combo :399 Min. : 0.0 1 Bedroom : 63   
## eys : 1 eat in :942 1st Qu.: 0.0 One Bedroom: 59   
## UG : 1 efficiency:849 Median : 659.0 2 Bedroom : 50   
## Underground: 1 none : 23 Mean : 618.9 Hi-Rise : 41   
## yes : 361 NA's : 17 3rd Qu.: 880.0 Co-Op : 33   
## Yes : 39 Max. :4659.0 (Other) :1944   
## NA's :1826 NA's : 40   
## num\_bedrooms num\_floors\_in\_building num\_full\_bathrooms num\_half\_bathrooms  
## Min. :0.000 Min. : 1.000 Min. :1.000 Min. :0.0000   
## 1st Qu.:1.000 1st Qu.: 3.000 1st Qu.:1.000 1st Qu.:1.0000   
## Median :2.000 Median : 6.000 Median :1.000 Median :1.0000   
## Mean :1.653 Mean : 7.785 Mean :1.001 Mean :0.9535   
## 3rd Qu.:2.000 3rd Qu.: 7.000 3rd Qu.:1.000 3rd Qu.:1.0000   
## Max. :6.000 Max. :34.000 Max. :2.000 Max. :2.0000   
## NA's :115 NA's :650 NA's :2058   
## num\_total\_rooms parking\_charges pct\_tax\_deductibl sale\_price   
## Min. : 0.000 NA :1671 Min. :20.0 NA :1702   
## 1st Qu.: 3.000 150 : 42 1st Qu.:40.0 155000 : 11   
## Median : 4.000 60 : 41 Median :50.0 175000 : 10   
## Mean : 4.139 75 : 27 Mean :45.4 130000 : 7   
## 3rd Qu.: 5.000 135 : 23 3rd Qu.:50.0 220000 : 7   
## Max. :14.000 100 : 21 Max. :75.0 235000 : 7   
## NA's :2 (Other): 405 NA's :1754 (Other): 486   
## sq\_footage total\_taxes walk\_score listing\_price\_to\_nearest\_1000  
## Min. : 100.0 NA :1646 Min. : 7.00 Min. : 65.0   
## 1st Qu.: 743.0 13 : 13 1st Qu.:77.00 1st Qu.: 229.8   
## Median : 881.0 250 : 12 Median :89.00 Median : 329.5   
## Mean : 955.4 4800 : 11 Mean :83.92 Mean : 385.6   
## 3rd Qu.:1100.0 2800 : 10 3rd Qu.:95.00 3rd Qu.: 525.0   
## Max. :6215.0 3200 : 8 Max. :99.00 Max. :1000.0   
## NA's :1210 (Other): 530 NA's :534   
## url   
## http://www.mlsli.com/homes-for-sale/10-01-162nd-St-Beechhurst-NY-11357-194398973 : 1   
## http://www.mlsli.com/homes-for-sale/10-11-162nd-St-Beechhurst-NY-11357-190650328 : 1   
## http://www.mlsli.com/homes-for-sale/10-24-166-St-Beechhurst-NY-11357-195381023 : 1   
## http://www.mlsli.com/homes-for-sale/10-Station-Sq-Forest-Hills-NY-11375-187747422 : 1   
## http://www.mlsli.com/homes-for-sale/100-10-67th-Rd-Forest-Hills-NY-11375-189635859: 1   
## (Other) : 753   
## NA's :1472   
## lat lon   
## Min. :40.65 Min. :-73.99   
## 1st Qu.:40.72 1st Qu.:-73.86   
## Median :40.74 Median :-73.83   
## Mean :40.74 Mean :-73.82   
## 3rd Qu.:40.76 3rd Qu.:-73.79   
## Max. :40.80 Max. :-73.70   
##

# selecting the variables which are going to affect the housing price,  
  
data\_select = names(houseing\_data) %in%  
 c("sale\_price","num\_total\_rooms","kitchen\_type","num\_bedrooms", "num\_floors\_in\_building","sq\_footage"  
 , "maintenance\_cost" ,"total\_taxes" ,"num\_full\_bathrooms" ,"dining\_room\_type","common\_charges",   
 "fuel\_type", "parking\_charges" )  
  
# combining the data set with the selected variables and removing rest of them  
Final\_data = houseing\_data[data\_select]  
  
# checkinf type of final data set  
str(Final\_data)

## 'data.frame': 2230 obs. of 13 variables:  
## $ common\_charges : int 767 0 167 275 0 0 0 0 0 0 ...  
## $ dining\_room\_type : Factor w/ 5 levels "combo","dining area",..: 1 3 1 1 1 1 1 NA NA 5 ...  
## $ fuel\_type : Factor w/ 6 levels "electric","gas",..: 2 4 NA 2 2 4 2 2 4 NA ...  
## $ kitchen\_type : Factor w/ 4 levels "combo","eat in",..: 2 2 3 2 2 2 3 3 2 2 ...  
## $ maintenance\_cost : int 0 604 0 0 660 932 660 514 781 0 ...  
## $ num\_bedrooms : int 2 1 1 3 2 2 1 0 1 1 ...  
## $ num\_floors\_in\_building: int 6 7 1 NA 2 6 NA 2 NA 4 ...  
## $ num\_full\_bathrooms : int 1 1 1 2 1 1 1 1 1 1 ...  
## $ num\_total\_rooms : int 5 4 3 5 4 4 3 2 4 3 ...  
## $ parking\_charges : Factor w/ 90 levels " NA ","100","105",..: 1 1 1 1 1 1 1 1 41 1 ...  
## $ sale\_price : Factor w/ 316 levels " NA ","100000",..: 107 113 33 252 119 126 38 8 94 250 ...  
## $ sq\_footage : int NA 890 550 NA 675 1000 NA 375 NA 681 ...  
## $ total\_taxes : Factor w/ 294 levels " NA ","100","1024",..: 1 1 255 68 1 1 1 1 1 19 ...

#converting the factor variables to numeric for prices  
Final\_data$parking\_charges = gsub("[\\$,]", "", Final\_data$parking\_charges)  
Final\_data$maintenance\_cost = gsub("[\\$,]", "", Final\_data$maintenance\_cost)  
Final\_data$sale\_price = gsub("[\\$,]", "", Final\_data$sale\_price)  
Final\_data$total\_taxes = gsub("[\\$,]", "", Final\_data$total\_taxes)  
  
# checking for Na's values  
sapply(Final\_data, function(x) sum(is.na(x)))

## common\_charges dining\_room\_type fuel\_type   
## 0 448 112   
## kitchen\_type maintenance\_cost num\_bedrooms   
## 17 0 115   
## num\_floors\_in\_building num\_full\_bathrooms num\_total\_rooms   
## 650 0 2   
## parking\_charges sale\_price sq\_footage   
## 0 0 1210   
## total\_taxes   
## 0

# converting data types to numerics  
Final\_data$dining\_room\_type = as.numeric(Final\_data$dining\_room\_type)  
Final\_data$fuel\_type = as.numeric(Final\_data$fuel\_type)  
Final\_data$kitchen\_type = as.numeric(Final\_data$kitchen\_type)  
Final\_data$num\_bedrooms = as.numeric(Final\_data$num\_bedrooms)  
Final\_data$num\_floors\_in\_building= as.numeric(Final\_data$num\_floors\_in\_building)  
Final\_data$num\_total\_rooms= as.numeric(Final\_data$num\_total\_rooms)  
Final\_data$sq\_footage= as.numeric(Final\_data$sq\_footage)  
Final\_data$parking\_charges = as.numeric(Final\_data$parking\_charges )

## Warning: NAs introduced by coercion

Final\_data$sale\_price= as.numeric(Final\_data$sale\_price)

## Warning: NAs introduced by coercion

Final\_data$total\_taxes= as.numeric(Final\_data$total\_taxes)

## Warning: NAs introduced by coercion

# replacing all Na's values with the median values  
dining = median(Final\_data$dining\_room\_type, na.rm = T)  
Final\_data$dining\_room\_type = ifelse(is.na(Final\_data$dining\_room\_type),  
 dining,Final\_data$dining\_room\_type)  
  
fuel\_type = median(Final\_data$fuel\_type, na.rm = T)  
Final\_data$fuel\_type = ifelse(is.na(Final\_data$fuel\_type),  
 fuel\_type,Final\_data$fuel\_type)  
  
kitchen\_type = median(Final\_data$kitchen\_type , na.rm = T)  
Final\_data$kitchen\_type = ifelse(is.na(Final\_data$kitchen\_type ),  
 kitchen\_type,Final\_data$kitchen\_type )  
  
num\_bedrooms = median(Final\_data$num\_bedrooms , na.rm = T)  
Final\_data$num\_bedrooms = ifelse(is.na(Final\_data$num\_bedrooms ),  
 num\_bedrooms,Final\_data$num\_bedrooms )  
  
num\_floors\_in\_building = median(Final\_data$num\_floors\_in\_building , na.rm = T)  
Final\_data$num\_floors\_in\_building = ifelse(is.na(Final\_data$num\_floors\_in\_building ),  
 num\_floors\_in\_building,Final\_data$num\_floors\_in\_building )  
  
num\_total\_rooms = median(Final\_data$num\_total\_rooms , na.rm = T)  
Final\_data$num\_total\_rooms = ifelse(is.na(Final\_data$num\_total\_rooms ),  
 num\_total\_rooms,Final\_data$num\_total\_rooms )  
  
parking\_charges = median(Final\_data$parking\_charges , na.rm = T)  
Final\_data$parking\_charges = ifelse(is.na(Final\_data$parking\_charges ),  
 parking\_charges,Final\_data$parking\_charges )  
  
sale\_price = median(Final\_data$sale\_price , na.rm = T)  
Final\_data$sale\_price = ifelse(is.na(Final\_data$sale\_price ),  
 sale\_price,Final\_data$sale\_price )  
  
sq\_footage = median(Final\_data$sq\_footage , na.rm = T)  
Final\_data$sq\_footage = ifelse(is.na(Final\_data$sq\_footage ),  
 sq\_footage,Final\_data$sq\_footage)  
  
total\_taxes = median(Final\_data$total\_taxes , na.rm = T)  
Final\_data$total\_taxes = ifelse(is.na(Final\_data$total\_taxes ),  
 total\_taxes,Final\_data$total\_taxes)  
  
# checking for Na's values after adding median values   
sapply(Final\_data, function(x) sum(is.na(x)))

## common\_charges dining\_room\_type fuel\_type   
## 0 0 0   
## kitchen\_type maintenance\_cost num\_bedrooms   
## 0 0 0   
## num\_floors\_in\_building num\_full\_bathrooms num\_total\_rooms   
## 0 0 0   
## parking\_charges sale\_price sq\_footage   
## 0 0 0   
## total\_taxes   
## 0

# summary of the data set  
summary(Final\_data)

## common\_charges dining\_room\_type fuel\_type kitchen\_type   
## Min. : 0.0 Min. :1.00 Min. :1.000 Min. :1.000   
## 1st Qu.: 0.0 1st Qu.:1.00 1st Qu.:2.000 1st Qu.:2.000   
## Median : 0.0 Median :1.00 Median :2.000 Median :2.000   
## Mean : 108.2 Mean :1.92 Mean :2.625 Mean :2.222   
## 3rd Qu.: 0.0 3rd Qu.:3.00 3rd Qu.:4.000 3rd Qu.:3.000   
## Max. :2499.0 Max. :5.00 Max. :6.000 Max. :4.000   
## maintenance\_cost num\_bedrooms num\_floors\_in\_building num\_full\_bathrooms  
## Length:2230 Min. :0.000 Min. : 1.000 Min. :1.000   
## Class :character 1st Qu.:1.000 1st Qu.: 5.000 1st Qu.:1.000   
## Mode :character Median :2.000 Median : 6.000 Median :1.000   
## Mean :1.671 Mean : 7.265 Mean :1.001   
## 3rd Qu.:2.000 3rd Qu.: 6.000 3rd Qu.:1.000   
## Max. :6.000 Max. :34.000 Max. :2.000   
## num\_total\_rooms parking\_charges sale\_price sq\_footage   
## Min. : 0.000 Min. : 6.0 Min. : 55000 Min. : 100   
## 1st Qu.: 3.000 1st Qu.: 99.0 1st Qu.:259500 1st Qu.: 881   
## Median : 4.000 Median : 99.0 Median :259500 Median : 881   
## Mean : 4.139 Mean :101.1 Mean :272631 Mean : 915   
## 3rd Qu.: 5.000 3rd Qu.: 99.0 3rd Qu.:259500 3rd Qu.: 881   
## Max. :14.000 Max. :837.0 Max. :999999 Max. :6215   
## total\_taxes   
## Min. : 11   
## 1st Qu.:2411   
## Median :2411   
## Mean :2363   
## 3rd Qu.:2411   
## Max. :9300

# converting still faactor to numerica data  
Final\_data$maintenance\_cost = as.numeric(Final\_data$maintenance\_cost)  
Final\_data$parking\_charges = as.numeric(Final\_data$parking\_charges)  
Final\_data$sale\_price = as.numeric(Final\_data$sale\_price)  
Final\_data$total\_taxes = as.numeric(Final\_data$total\_taxes)  
  
# final summary of the data set after doing all the possible conversions  
summary(Final\_data)

## common\_charges dining\_room\_type fuel\_type kitchen\_type   
## Min. : 0.0 Min. :1.00 Min. :1.000 Min. :1.000   
## 1st Qu.: 0.0 1st Qu.:1.00 1st Qu.:2.000 1st Qu.:2.000   
## Median : 0.0 Median :1.00 Median :2.000 Median :2.000   
## Mean : 108.2 Mean :1.92 Mean :2.625 Mean :2.222   
## 3rd Qu.: 0.0 3rd Qu.:3.00 3rd Qu.:4.000 3rd Qu.:3.000   
## Max. :2499.0 Max. :5.00 Max. :6.000 Max. :4.000   
## maintenance\_cost num\_bedrooms num\_floors\_in\_building num\_full\_bathrooms  
## Min. : 0.0 Min. :0.000 Min. : 1.000 Min. :1.000   
## 1st Qu.: 0.0 1st Qu.:1.000 1st Qu.: 5.000 1st Qu.:1.000   
## Median : 659.0 Median :2.000 Median : 6.000 Median :1.000   
## Mean : 618.9 Mean :1.671 Mean : 7.265 Mean :1.001   
## 3rd Qu.: 880.0 3rd Qu.:2.000 3rd Qu.: 6.000 3rd Qu.:1.000   
## Max. :4659.0 Max. :6.000 Max. :34.000 Max. :2.000   
## num\_total\_rooms parking\_charges sale\_price sq\_footage   
## Min. : 0.000 Min. : 6.0 Min. : 55000 Min. : 100   
## 1st Qu.: 3.000 1st Qu.: 99.0 1st Qu.:259500 1st Qu.: 881   
## Median : 4.000 Median : 99.0 Median :259500 Median : 881   
## Mean : 4.139 Mean :101.1 Mean :272631 Mean : 915   
## 3rd Qu.: 5.000 3rd Qu.: 99.0 3rd Qu.:259500 3rd Qu.: 881   
## Max. :14.000 Max. :837.0 Max. :999999 Max. :6215   
## total\_taxes   
## Min. : 11   
## 1st Qu.:2411   
## Median :2411   
## Mean :2363   
## 3rd Qu.:2411   
## Max. :9300

str(Final\_data)

## 'data.frame': 2230 obs. of 13 variables:  
## $ common\_charges : int 767 0 167 275 0 0 0 0 0 0 ...  
## $ dining\_room\_type : num 1 3 1 1 1 1 1 1 1 5 ...  
## $ fuel\_type : num 2 4 2 2 2 4 2 2 4 2 ...  
## $ kitchen\_type : num 2 2 3 2 2 2 3 3 2 2 ...  
## $ maintenance\_cost : num 0 604 0 0 660 932 660 514 781 0 ...  
## $ num\_bedrooms : num 2 1 1 3 2 2 1 0 1 1 ...  
## $ num\_floors\_in\_building: num 6 7 1 6 2 6 6 2 6 4 ...  
## $ num\_full\_bathrooms : int 1 1 1 2 1 1 1 1 1 1 ...  
## $ num\_total\_rooms : num 5 4 3 5 4 4 3 2 4 3 ...  
## $ parking\_charges : num 99 99 99 99 99 99 99 99 20 99 ...  
## $ sale\_price : num 228000 235500 137550 545000 241700 ...  
## $ sq\_footage : num 881 890 550 881 675 1000 881 375 881 681 ...  
## $ total\_taxes : num 2411 2411 5500 2260 2411 ...

# partitioning the data sets into two parts training andtesting   
library(caret)

## Loading required package: lattice

## Warning: package 'lattice' was built under R version 3.6.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 3.6.3

set.seed(200)  
trainIndex <- createDataPartition(Final\_data$sale\_price, p = 0.7,list = FALSE)  
train <- Final\_data[trainIndex, ]  
test <- Final\_data[-trainIndex, ]  
  
###########################Regression Tree Model ###################  
library(rpart)  
Tree <- rpart(sale\_price ~ ., Final\_data)  
Tree$variable.importance

## maintenance\_cost common\_charges total\_taxes   
## 1.650293e+12 1.171413e+12 9.419430e+11   
## sq\_footage num\_floors\_in\_building num\_bedrooms   
## 8.834023e+11 5.311867e+11 4.825814e+11   
## num\_total\_rooms kitchen\_type fuel\_type   
## 1.223379e+11 9.015394e+09 2.043422e+09

plot(Tree$variable.importance)

Chart, scatter chart

Description automatically generated

summary(Tree)

## Call:  
## rpart(formula = sale\_price ~ ., data = Final\_data)  
## n= 2230   
##   
## CP nsplit rel error xerror xstd  
## 1 0.06427699 0 1.0000000 1.0006911 0.08579965  
## 2 0.03523877 1 0.9357230 0.9424917 0.08132666  
## 3 0.02284063 3 0.8652455 0.9609578 0.07992761  
## 4 0.01438507 4 0.8424048 0.9091466 0.07237486  
## 5 0.01428081 5 0.8280198 0.9128499 0.07274855  
## 6 0.01078736 7 0.7994581 0.9189205 0.07242182  
## 7 0.01000000 8 0.7886708 0.8980089 0.07005557  
##   
## Variable importance  
## maintenance\_cost common\_charges total\_taxes   
## 28 20 16   
## sq\_footage num\_floors\_in\_building num\_bedrooms   
## 15 9 8   
## num\_total\_rooms   
## 2   
##   
## Node number 1: 2230 observations, complexity param=0.06427699  
## mean=272630.5, MSE=8.172402e+09   
## left son=2 (1738 obs) right son=3 (492 obs)  
## Primary splits:  
## common\_charges < 207 to the left, improve=0.06427699, (0 missing)  
## maintenance\_cost < 345.5 to the right, improve=0.05362392, (0 missing)  
## num\_total\_rooms < 4.5 to the left, improve=0.04214775, (0 missing)  
## num\_bedrooms < 1.5 to the left, improve=0.04101316, (0 missing)  
## total\_taxes < 2573.5 to the left, improve=0.03268324, (0 missing)  
## Surrogate splits:  
## maintenance\_cost < 158.5 to the right, agree=0.934, adj=0.699, (0 split)  
## total\_taxes < 2418 to the left, agree=0.859, adj=0.360, (0 split)  
## sq\_footage < 1975 to the left, agree=0.782, adj=0.010, (0 split)  
## num\_bedrooms < 4.5 to the left, agree=0.781, adj=0.008, (0 split)  
## num\_total\_rooms < 8.5 to the left, agree=0.781, adj=0.006, (0 split)  
##   
## Node number 2: 1738 observations, complexity param=0.03523877  
## mean=260436.1, MSE=5.41386e+09   
## left son=4 (1449 obs) right son=5 (289 obs)  
## Primary splits:  
## maintenance\_cost < 1087.5 to the left, improve=0.06276233, (0 missing)  
## sq\_footage < 1875 to the left, improve=0.05766499, (0 missing)  
## num\_bedrooms < 1.5 to the left, improve=0.05732530, (0 missing)  
## num\_total\_rooms < 4.5 to the left, improve=0.05242858, (0 missing)  
## num\_floors\_in\_building < 28 to the left, improve=0.02662562, (0 missing)  
## Surrogate splits:  
## sq\_footage < 1115.5 to the left, agree=0.873, adj=0.235, (0 split)  
## num\_floors\_in\_building < 29.5 to the left, agree=0.860, adj=0.156, (0 split)  
## num\_bedrooms < 2.5 to the left, agree=0.838, adj=0.024, (0 split)  
## num\_total\_rooms < 7.5 to the left, agree=0.836, adj=0.014, (0 split)  
## fuel\_type < 4.5 to the left, agree=0.834, adj=0.003, (0 split)  
##   
## Node number 3: 492 observations, complexity param=0.02284063  
## mean=315707.5, MSE=1.553609e+10   
## left son=6 (477 obs) right son=7 (15 obs)  
## Primary splits:  
## num\_floors\_in\_building < 20.5 to the left, improve=0.05445728, (0 missing)  
## total\_taxes < 102 to the right, improve=0.02401668, (0 missing)  
## num\_total\_rooms < 6.5 to the left, improve=0.01495783, (0 missing)  
## dining\_room\_type < 2 to the right, improve=0.01364689, (0 missing)  
## sq\_footage < 740 to the left, improve=0.01141150, (0 missing)  
##   
## Node number 4: 1449 observations, complexity param=0.01438507  
## mean=252203.9, MSE=3.166248e+09   
## left son=8 (25 obs) right son=9 (1424 obs)  
## Primary splits:  
## num\_bedrooms < 0.5 to the left, improve=0.05714169, (0 missing)  
## num\_total\_rooms < 4.5 to the left, improve=0.03419156, (0 missing)  
## sq\_footage < 895 to the left, improve=0.02950284, (0 missing)  
## maintenance\_cost < 818.5 to the left, improve=0.02617607, (0 missing)  
## total\_taxes < 3900 to the left, improve=0.02435603, (0 missing)  
##   
## Node number 5: 289 observations, complexity param=0.03523877  
## mean=301711.2, MSE=1.46396e+10   
## left son=10 (282 obs) right son=11 (7 obs)  
## Primary splits:  
## sq\_footage < 1750 to the left, improve=0.16400180, (0 missing)  
## maintenance\_cost < 2285.5 to the left, improve=0.05209679, (0 missing)  
## num\_bedrooms < 2.5 to the left, improve=0.03720352, (0 missing)  
## parking\_charges < 87.5 to the left, improve=0.02037878, (0 missing)  
## num\_total\_rooms < 5.5 to the left, improve=0.01691183, (0 missing)  
## Surrogate splits:  
## maintenance\_cost < 2654.5 to the left, agree=0.983, adj=0.286, (0 split)  
##   
## Node number 6: 477 observations, complexity param=0.01428081  
## mean=310549.4, MSE=1.397321e+10   
## left son=12 (430 obs) right son=13 (47 obs)  
## Primary splits:  
## total\_taxes < 102 to the right, improve=0.032638820, (0 missing)  
## dining\_room\_type < 2 to the right, improve=0.014475980, (0 missing)  
## num\_total\_rooms < 6.5 to the left, improve=0.012339190, (0 missing)  
## sq\_footage < 882 to the left, improve=0.010095980, (0 missing)  
## common\_charges < 535.5 to the right, improve=0.007989041, (0 missing)  
##   
## Node number 7: 15 observations  
## mean=479733.3, MSE=3.748493e+10   
##   
## Node number 8: 25 observations  
## mean=150688, MSE=1.4746e+09   
##   
## Node number 9: 1424 observations, complexity param=0.01078736  
## mean=253986.1, MSE=3.011846e+09   
## left son=18 (748 obs) right son=19 (676 obs)  
## Primary splits:  
## num\_bedrooms < 1.5 to the left, improve=0.04583814, (0 missing)  
## num\_total\_rooms < 4.5 to the left, improve=0.03007128, (0 missing)  
## sq\_footage < 895 to the left, improve=0.02735916, (0 missing)  
## total\_taxes < 3900 to the left, improve=0.02465219, (0 missing)  
## maintenance\_cost < 820.5 to the left, improve=0.02200345, (0 missing)  
## Surrogate splits:  
## num\_total\_rooms < 3.5 to the left, agree=0.784, adj=0.544, (0 split)  
## maintenance\_cost < 761.5 to the left, agree=0.628, adj=0.216, (0 split)  
## sq\_footage < 895 to the left, agree=0.619, adj=0.197, (0 split)  
## num\_floors\_in\_building < 5.5 to the right, agree=0.581, adj=0.117, (0 split)  
## kitchen\_type < 2.5 to the right, agree=0.547, adj=0.046, (0 split)  
##   
## Node number 10: 282 observations  
## mean=293991.3, MSE=1.058628e+10   
##   
## Node number 11: 7 observations  
## mean=612714.1, MSE=7.880688e+10   
##   
## Node number 12: 430 observations  
## mean=303489, MSE=1.170387e+10   
##   
## Node number 13: 47 observations, complexity param=0.01428081  
## mean=375144.7, MSE=3.010669e+10   
## left son=26 (40 obs) right son=27 (7 obs)  
## Primary splits:  
## total\_taxes < 67.5 to the left, improve=0.21411450, (0 missing)  
## sq\_footage < 1083 to the right, improve=0.08545282, (0 missing)  
## common\_charges < 297.5 to the left, improve=0.08207958, (0 missing)  
## num\_bedrooms < 1.5 to the right, improve=0.07456540, (0 missing)  
## dining\_room\_type < 2 to the right, improve=0.05545256, (0 missing)  
##   
## Node number 18: 748 observations  
## mean=242816.1, MSE=2.798518e+09   
##   
## Node number 19: 676 observations  
## mean=266345.8, MSE=2.957077e+09   
##   
## Node number 26: 40 observations  
## mean=341557.5, MSE=2.211472e+10   
##   
## Node number 27: 7 observations  
## mean=567071.4, MSE=3.249289e+10

library(rpart.plot)  
rpart.plot(Tree)

Diagram

Description automatically generated

pred = predict(Tree, test)  
pred\_re <- cbind(pred , test$sale\_price)  
colnames(pred\_re) <- c('pred','original')  
pred\_re <- as.data.frame(pred\_re)  
  
#mean Square error  
mse <- mean((pred\_re$original - pred\_re$pred) ^ 2)  
print(mse)

## [1] 6285897603

#Root mean squared error   
mse ^ 0.5

## [1] 79283.65

# R - square Value  
SSE = sum((pred\_re$pred - pred\_re$original) ^ 2)  
SST = sum((mean(Final\_data$sale\_price) - pred\_re$original) ^ 2)  
R2 = 1 - SSE/SST  
R2

## [1] 0.1570063

# linear regression model  
model = lm(sale\_price~. , Final\_data)  
summary(model)

##   
## Call:  
## lm(formula = sale\_price ~ ., data = Final\_data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -205619 -36879 -8682 13368 619951   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -74599.431 61376.873 -1.215 0.22433   
## common\_charges 68.125 10.282 6.626 4.33e-11 \*\*\*  
## dining\_room\_type 213.223 1390.882 0.153 0.87818   
## fuel\_type -771.399 1859.817 -0.415 0.67835   
## kitchen\_type -6292.865 2450.058 -2.568 0.01028 \*   
## maintenance\_cost -1.392 5.074 -0.274 0.78393   
## num\_bedrooms 19116.795 3592.028 5.322 1.13e-07 \*\*\*  
## num\_floors\_in\_building 1996.563 324.538 6.152 9.05e-10 \*\*\*  
## num\_full\_bathrooms 281195.328 60128.937 4.677 3.09e-06 \*\*\*  
## num\_total\_rooms 3264.914 2000.226 1.632 0.10276   
## parking\_charges 81.020 51.330 1.578 0.11461   
## sq\_footage 20.575 8.037 2.560 0.01053 \*   
## total\_taxes -5.127 1.933 -2.653 0.00803 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 84620 on 2217 degrees of freedom  
## Multiple R-squared: 0.1288, Adjusted R-squared: 0.1241   
## F-statistic: 27.32 on 12 and 2217 DF, p-value: < 2.2e-16

plot(model, 1)